Perfect sampling: Part I

Mark Jerrum

Queen Mary, University of London

ADYN Summer School 27th June – 1st July, 2022

Joint work with Konrad Anand (QMUL), Heng Guo (Edinburgh) and Jingcheng Liu (Nanjing)

Motivation

Problem: produce a realisation of a random variable with a specified probability distribution.

Example: Given an undirected graph G, sample, uniformly at random, a spanning tree in G.

The sample must exactly match the desired probability distribution. This rules out Markov chain simulation, a common approach to sampling.

Why study perfect sampling?

- Theoretical appeal.
- Perfect samplers are 'self clocking'.

The scope of these tutorials

• Perfect sampling is too wide a topic to cover in generality. See, for example,

Mark Huber, *Perfect simulation*. Monographs on Statistics and Applied Probability, 148. CRC Press, 2016.

• We concentrate on samplers that can, in principle, work in time linear in the input size (or in constant time for each 'piece' of the output).

This rules out Coupling from the Past (CFTP), perhaps the best known approach to perfect sampling.

It also opens up the possibility of sampling (portions of) infinite objects.

• Perhaps the first perfect sampler along these lines was Fill and Huber's 'Randomness Recycler'.

The setting for this tutorial

Suppose $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ is a formula in variables X_1, \ldots, X_n . (The variables may be Boolean, and the clauses might be disjunctions of literals, but this is not essential.)

There is an underlying product distribution on the variables $\mathbf{X} = (X_1, \dots, X_n)$. (Perhaps, the values result from independent tosses of a fair coin.)

We wish to sample from the conditional distribution of **X** given $\Phi(\mathbf{X})$.

Reference: Fundamentals of Partial Rejection Sampling, arXiv:2106.07744

イロト イポト イヨト イヨト 二日

Example: independent sets



э

< □ > < 同 >



< □ > < @ >

Exploit connection with the Lovász Local Lemma

Recall $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$. Assume that

- each clause shares variables with at most d other clauses;
- under a uniform random assignment to the variables X_1, \ldots, X_n each clause is false with probability at most p.

Then the Lovász Local Lemma (LLL) asserts that, if $4pd \leq 1$, then Φ is true with non-zero probability.

The LLL guarantees only an exponentially small probability, so simple rejection sampling will not, in general, find a satisfying assignment to Φ efficiently.

イロト 不得下 イヨト イヨト 二日

Moser-Tardos resampling algorithm

A remarkable breakthrough is due to Moser and Tardos (2010), who found an algorithmically efficient version of LLL:

- Initialize variables X_1, \ldots, X_n independently at random.
- While there exists an unsatisfied clause: pick one and resample all its variables.

Moser and Tardos showed that this algorithm is efficient under the same condition as LLL.

Question

Instead of simply finding a satisfying assignment, can we generate one uniformly at random?

< □ > < □ > < □ > < □ > < □ > < □ >

Searching versus sampling: independent sets

Consider the problem of sampling independent sets in graph. Variables correspond to vertices. Clauses correspond to edges. A typical clause has the form $\neg X_i \lor \neg X_j$.

The Moser-Tardos algorithm selects an edge with both endpoints in the current independent set and re-randomises the variables corresponding to the two endpoints.

For a path of length two (i.e., with three vertices) the empty independent set is generated with probability $\frac{2}{9}$ and not $\frac{1}{5}$ as required.

In fact, any efficient algorithm ought to fail, as sampling independent sets uniformly at random is a computationally hard problem (NP-hard).

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

Searching versus sampling: sink-free orientations



Lemma (Cohn, Pemantle and Propp, 2002) Moser-Tardos-style resampling generates a uniform sink-free orientation in expected time O(|V||E|).

What distinguishes sink-free orientations from independent sets?

Definition

We call an instance (formula) $\Phi = C_1 \wedge \cdots \wedge C_m$ extremal if every pair of distinct clauses C_i and C_j are either independent (C_i and C_j have no variables in common) or exhaustive (C_i and C_j cannot both be false).

- The formula encoding sink-free orientations is extremal.
- The formula encoding independent sets is not extremal.

Check this!



э

< □ > < □ > < □ > < □ > < □ > < □ >

Extremal instances and perfect sampling

- Extremal instances Φ (in some precise sense) minimize the probability that the formula is true [Shearer 85].
- Moser-Tardos is slowest on extremal instances.
- However, slowest for searching is best for sampling!

Theorem (Guo, Jerrum and Liu, 2017)

For extremal instances, the output of Moser-Tardos is uniform. More precisely, if Φ is satisfiable, then the output of the resampling algorithm is from the product distribution conditioned on $\Phi(X)$.

We refer to this approach as *Partial Rejection Sampling* (PRS).

A B A A B A

Resampling tables and transcripts (by example)

A formula (extremal)

 $\Phi(\mathbf{X}) = (X_1 \lor X_2) \land (\neg X_1 \lor X_3 \lor \neg X_4) \land (\neg X_2 \lor \neg X_3 \lor X_5) \land (X_4 \lor \neg X_5)$

and a possible resampling table and transcript



э

< □ > < □ > < □ > < □ > < □ > < □ >

The order of resampling does not matter

Lemma

Fix a resampling table. Suppose that for some sequence of non-deterministic choices, PRS run on Φ terminates with a certain transcript. Then for any other sequence of choices, the algorithm will terminate with the same transcript.

Proof by picture. A rigorous proof uses a version of Newman's Lemma. C.f. the Abelian Sandpile model.

Uniqueness of the transcript

Recall the formula

 $\Phi(\mathbf{X}) = (X_1 \vee X_2) \wedge (\neg X_1 \vee X_3 \vee \neg X_4) \wedge (\neg X_2 \vee \neg X_3 \vee X_5) \wedge (X_4 \vee \neg X_5)$ and a possible resampling table



Invariance under change of solution

Changing the content of the 'final frontier' of the resampling table does not change the interior of the transcript.



★ ∃ ► < ∃ ►</p>

Correctness of PRS

Theorem

Suppose Φ is a satisfiable extremal instance. Then PRS on input Φ terminates with probability 1. On termination, X is a realisation of a random variable from the desired distribution.

Proof.

- The number of resamplings is dominated by a geometric random variable.
- Regard two transcripts as equivalent iff they agree except on the satisfying assignment in the final frontier. Each satisfying assignment is possible in this location. So conditioned on the equivalence class, each satisfying assignment is equally likely.

3

Extremal instances: running time

Theorem

For extremal instances, the expected number of (sequential) resampling steps is equal to

 $\Pr(\textit{Exactly one clause in } \Phi \textit{ is false})$

 $Pr(\Phi \text{ is true})$

The probabilities are with respect to the (unconditioned) product distribution on variables.

The upper bound is due to Kolipaka and Szegedy (2011).

It is a simple statement, but there is apparently no simple proof.

4 1 1 1 4 1 1 1

A generating function for transcripts

Recall the formula we have been working with:

 $\Phi(\mathbf{X}) = (X_1 \lor X_2) \land (\neg X_1 \lor X_3 \lor \neg X_4) \land (\neg X_2 \lor \neg X_3 \lor X_5) \land (X_4 \lor \neg X_5)$

Introduce variables z_1, z_2, z_3, z_4 corresponding to the four clauses C_1, C_2, C_3, C_4 . Each transcript can be encoded as a monomial, in this case $z_1^2 z_2^2 z_3 z_4^2$.

Summing these monomials over all transcripts we obtain the generating function $T(z_1, z_2, z_3, z_4)$ for transcripts.

The probability of a transcript

The probability of a particular transcript arising is easy to compute! Denote by p_k the probability of clause C_k being true in the (unconditioned) product distribution. Note that the probability of a transcript with monomial $z_1^{e_1} z_2^{e_2} z_3^{e_3} z_4^{e_4}$ arising is proportional to $p_1^{e_1} p_2^{e_2} p_3^{e_3} p_4^{e_4}$.

Expected number of resamplings

- By differentiating the generating function T(z₁, z₂, z₃, z₄) with respect to z_k and evaluating it at the point
 (Z₁, Z₂, z₃, z₄) = (p₁, p₂, p₃, p₄), we obtain a new sum over transcripts, this time weighted by the number of resamplings of the variables in clause C_k.
- This immediately yields an expression for the expected number of resamplings of clause C_k in a run of PRS.
- Finally, by inclusion-exclusion, we can relate this expression to the quantities $Pr(Exactly \text{ one clause in } \Phi \text{ is false})$ and $Pr(\Phi \text{ is true})$ in the statement of the theorem.

An example: Sink-free orientations

Assume the graph G does have a sink-free orientation. Applying PRS we obtain the following:

while the current orientation of G has at least one sink do Choose any sink vertex vResample the orientations of the edges incident at vend while

- If G has a sink-free orientation, then the above loop terminates with probability 1. (Why?)
- Since the instance is extremal, the output distribution is uniform over all sink-free orientations.

Running time

The expected number of resamplings is given by

 $\frac{\Pr(\text{There is exactly one sink in } O)}{\Pr(O \text{ is sink-free})}.$

where O is a uniform random orientation of G. Since the orientation of each edge is unbiased, this quotient is equal to

The number of orientations with exactly one sink The number of sink-free orientations

In order to bound the latter ratio, we show how to take an orientation with exactly one sink, and 'repair' it to obtain a sink-free orientation.

Running time (continued)

- To obtain a good bound, we require the repair to be minimal.
- Let O be any 'reference' sink-free orientation.
- Choose a function $f: V \to V$ that is consistent with O, in the sense that f(u) = v implies edge uv is directed from u to v in O.
- The repair is effected as in the following slide.

Repairing an orientation with a single sink



э

ヘロト 人間ト 人間ト 人間ト

Finishing up

- We can undo the repair if we know the start an end vertices of the path of reversals.
- Thus, each sink-free orientation corresponds to at most $|V(G)|^2$ orientations with exactly one sink.
- The expected number of resamplings is bounded by $|V(G)|^2$.
- The analysis can be extended to the expected number of edge reversals, which is bounded by 2|V(G)||E(G)|. Guo and He improved this bound to |V(G)|² + |E(G)|.

Another example: root connected subgraphs

- Let G be a directed graph with a distinguished root r.
- A (spanning) subgraph of G in which there is a path from every vertex of G to the root r is called *root connected*.
- Our goal is to sample a root-connected subgraph uniformly at random from G.
- Edge weights can be accommodated, but we shall consider the unweighted case for simplicity.
- Motivation: equivalent to the problem of sampling connected spanning subgraphs of an undirected graph (all-terminal network reliability).

Clusters

Definition

In a directed graph (V,A) with root r, a subset $C \subseteq V$ of vertices is called a *cluster* if $C \neq \emptyset$, $r \notin C$ and there is no arc $u \rightarrow v \in A$ such that $u \in C$ and $v \notin C$. We say that C is a *minimal cluster* if C is a cluster and for any proper subset C' \subset C, C' is not a cluster.

Some observations:

- Minimal clusters are disjoint.
- Minimal clusters are strongly connected.
- G = (V, A) is root connected iff it contains no clusters.

Cluster popping (Gorodezky and Pak)

The algorithm samples a uniform root-connected subgraph of a directed graph G.

The basic approach is simple: keep 'popping' minimal clusters until no clusters exist.

Let S be a subset of arcs obtained by choosing each arc $e \in A$ with probability $\frac{1}{2}$ independently. while there is a cluster in (V, S) do

Let C_1,\ldots,C_k be all minimal clusters in (V,S), and $C=\bigcup_{i=1}^k C_i.$

Re-randomize all arcs with start vertices in C to get a new S. end while

return S

The instance.



2

< □ > < □ > < □ > < □ > < □ >

A random starting configuration (in red).



The unique minimal cluster and its associated edges (dashed).



Re-randomise the cluster.



Image: A matrix

The unique minimal cluster and its associated edges (dashed).



Re-randomise the cluster.



Image: A matrix

The unique minimal cluster and its associated edges (dashed).



Re-randomise the cluster.



Image: A matrix

The two minimal clusters and their associated edges (dashed).



Re-randomise the clusters (say left first then centre). There are no remaining clusters, so halt.



Correctness

Cluster popping fits into the PRS framework.

Key property: for every pair of subsets $U, U' \subseteq V \setminus \{r\}$ either

- $U \cap U' = \emptyset$ (in which case the events that U and U' are minimal clusters are probabilistically independent), or
- U and U' cannot both be minimal clusters.

So the extremal property holds here and guarantees correctness of the partial rejection sampling algorithm: while there exists a minimal cluster U, re-randomise the variables associated with U.

Note that this is exactly what Gorodezky and Pak's cluster-popping algorithm does!

Running time: bidirected graphs

In general, the expected number of cluster poppings is exponential.

An interesting special case is that of *bi-directed* graphs, in which arcs occur in antiparallel pairs.

Gorodezky and Pak conjectured that the running time of cluster popping in bidirected graphs is polynomial in n, the number of vertices of G.

As we saw, the expected number of (sequential) resampling steps is equal to

 $\frac{\Pr(\mathsf{Exactly one clause in } \Phi \text{ is false})}{\Pr(\Phi \text{ is true})},$

where there is a clause φ_u in Φ for each $U \subseteq V \setminus \{r\}$. The clause φ_u asserts that U is not a minimal cluster.

イロト 不得下 イヨト イヨト 二日

Analysis for cluster popping in particular

In this instance,

 $\frac{\Pr(\mathsf{Exactly one clause in } \Phi \text{ is false})}{\Pr(\Phi \text{ is true})}$

is equal to

Number of subgraphs with exactly one minimal cluster Number of root-connected subgraphs

As before, we can bound this ratio by showing how to repair a subgraph with exactly one minimal cluster, to obtain a root-connected subgraph. Repairing is a little more complicated this time.

Circles are strongly connected components. R is the set of vertices from which the root r is reachable.



Light blue shading indicates the unique minimal cluster.



There must exist an edge (u, u') entering R.



Light blue shading indicates vertices reachable from u.



Flip direction of edges between shaded connected components and add the edge (u, u').



No minimal clusters \equiv root-connected graph.



Conclusion

Theorem

There is a perfect sampler for root-connected subgraphs of a bidirected graph with expected running time $O(|E|^2|V|)$. (The number of resampling steps is O(|E||V|).)

Corollary

There is an FPRAS for approximating the number of connected spanning subgraphs of an undirected graph.

As edge weights can be incorporated, we have an efficient approximation algorithm for (undirected) all-terminal reliability.

In a subsequent breakthrough, Anari, Liu, Oveis Gharan and Vinzant showed how to solve a more general sampling problem by Markov chain simulation.

A list of applications of PRS (in the extremal case)

- Sink-free orientations.
- Spanning trees.
- Root-connected subgraphs of a bidirected graph (equivalent to spanning connected subgraphs of an undirected graph.
- Bases of a bicircular matroid.

Where it is applicable, PRS works well. But the range of applications seems limited. Are there any further applications out there?