Dual Hashing-based Algorithms for Discrete Integration (Thesis Summary)

Alexis de Colnet[1] and Kuldeep S. Meel[2]

[1] CNRS, CRIL UMR 8188, Lens, France
[2] School of Computing, National University of Singapore, Singapore

Abstract. This paper is a summary of the master thesis "Dual Hashing-based Algorithms for Discrete Integration" realized by the first author at the National University of Singapore. This thesis led to the publication of an article of same title at the 25th International Conference on Principles and Practice of Constraint Programming (CP2019).

Given a boolean formula F and a weight function ρ, the problem of discrete integration seeks to compute the weight of F, defined as the sum of the weights of satisfying assignments. Discrete integration is a fundamental problem in computer science with wide variety of applications ranging from machine learning and statistics to physics and infrastructure reliability. Given the intractability of the problem, the approximate variant has been subject to intense theoretical and practical investigations over the years. This thesis investigates development of algorithmic approaches for approximate discrete integration. Discrete integration is analyzed through the framework of general integration. Two algorithms emerge from this framework: WISH, which was already discovered by Ermon et al [7], and a new algorithm: SWITCH. These algorithms both approximate the weight of F within a constant factor with high probability and can be seen as dual to each other, in the sense that their complexities differ only by a permutation of certain parameters. Indeed we show that, for F defined over n variables, a weight function ρ that can be represented using p bits, and a confidence parameter δ, there is a function f and an NP oracle such that WISH makes $\mathcal{O}(f(n,p,\delta))$ calls to NP oracle while SWITCH makes $\mathcal{O}(f(p,n,\delta))$ calls. We found $f(n,p,\delta)$ polynomial in n, p and $1/\delta$, more specifically $f(n,p,\delta) = n\log(p)\log(n/\delta)$.

1 Introduction

Given a boolean formula F and a weight function ρ that assigns a non-negative weight to every assignment of values to variables, the problem of discrete integration seeks to compute the weight of F, defined as the sum of weights of its satisfying assignments. Discrete integration is a fundamental problem in computer science. A wide variety of problems such as probabilistic inference [12], partition function of graphical models, permanent of a ma-

trix [15], reliability of a network [11] can be reduced to discrete integration.

In his seminal work, Valiant [15] established the complexity of discrete integration as #P-complete for all polynomially computable weight functions, where #P is the complexity class comprised of counting problems whose decision variant lies in NP. Given the computational intractability of discrete integration, approximate variants have been subject of intense theoretical and practical investigations over the past few decades.

Approaches to discrete integration can be classified into three categories: variational techniques, sampling techniques, and hashing-based techniques. Inspired from statistical physics, variational methods often scale to large instances but do not provide guarantees on the computed estimates [16,14]. Sampling-based techniques focus on approximation of the discrete integral via sampling from the probability distribution induced by the boolean formula and the weight function [9]. The estimation of rigorous bounds, however, require exponentially many samples and therefore, practical implementations such as those based on Markov Chain Monte Carlo methods [1] or randomized branching choices [8] fail to provide rigorous estimates [6,10]. Recently, hashing-based techniques have emerged as a promising alternative to variational and sampling techniques to provide rigorous approximation guarantees [7,5,3]. The hashing-based algorithm WISH seeks to utilize progress made in combinatorial solving over the past two decades and to this end, the problem of discrete integration is reduced to linear number of optimization queries subject to randomly generated parity constraints [7].

The primary contribution of the thesis has been to investigate the development of algorithmic approaches for discrete integration. It sets a framework from which were derived two different algorithms: WISH, which was already discovered by Ermon et al [7], and a new algorithm: SWITCH. In particular, WISH reduces the problem of discrete integration to optimization queries while SWITCH proceeds via reduction to unweighted model counting. Both WISH and SWITCH compute constant factor approximations with high probability $1 - \delta$ via usage of universal hash functions, a concept invented by Carter and Wegman in their seminal work [2]. The thesis has been divided into three parts: the study of discrete integration through the framework of general integration, the analysis of WISH and the analysis of SWITCH. This summary briefly presents the findings and conclusions of each of these parts. After an introduction of the hypothesis that were made for this work in section 2, section 3 explains that discrete integration reduces through our framework to optimization and counting subproblems. Section 4 presents WISH and SWITCH as hashing-based algorithms solving the aforementioned subproblems to approximate a discrete integral. In our work, we proved that both algorithms compute constant factor approximations of the discrete integral with high probability. However we have shown that they have dual time complexities in the sense that,

for F defined over n variables and a weight function ρ that can be represented using p bits, there is a function f and an NP oracle such that WISH makes $\mathcal{O}(f(n,p,\delta))$ calls to NP oracle while SWITCH makes $\mathcal{O}(f(p,n,\delta))$. We find $f(x,y,\delta)$ polynomial in x, y and $1/\delta$, specifically $f(n,p,\delta) = n\log(p)\log(n/\delta)$.

The duality obtained may not seem surprising in retrospect but such has not been the case for the past five years. The prior work has often, without complete evidence, asserted that the corresponding dual approach would be inferior both theoretically and empirically [7,3]. The findings of this thesis, in turn, contradicts such assertions and show that the two approaches indeed have complimentary time complexity from theoretical perspective and empirical analysis will be key in determining their usefulness. Since the work on development of MaxSAT solvers that support XORs and SAT solvers that support XORs and Pseudo-Boolean (PB) constraints is in its infancy; this work provides a strong argument for the need and potential of both of these solvers as queries generated by WISH require MaxSAT solvers with the ability to handle XORs while the queries by SWITCH requires SAT solvers that support XORs and PB constraints.

2 Context

Given a boolean formula F over n variables and a weight function ρ mapping each truth assignment to a non-negative weight, the problem of discrete integration seeks to compute the weight of F, defined as the sum of weights of its satisfying assignments, or witnesses, and denoted by $\rho(F) = \sum_{\sigma \models F} \rho(\sigma)$.

The thesis makes a few hypothesis on the weight function

- for all assignment $\sigma \in \{0,1\}^n$, the weight $\rho(\sigma)$ is computable in polynomial time
- for all assignment $\sigma \in \{0,1\}^n$, the weight $\rho(\sigma)$ is written with p bits in binary representation
- bounds on the minimum and maximum weights are known

Regarding the third hypothesis, the results of the thesis were obtained assuming that all weights belong to $[0,1]$. This makes sense in the context of probability inference, and the results obtained can easily be adapted to any arbitrary, but fixed, upper bound on ρ.

3 A Framework for Discrete Integration

The development of algorithmic approaches for discrete integration led to setting up a framework for discrete integration. Methods from this framework follow a two-steps strategy:

1. reduce discrete integration to an integration problem for a real non-increasing function

2. apply a method to approximate the integral of a real function

In the first step, the function is required to be non-increasing so that constant factor approximations can be ensured when estimating its integral.

Discrete integration is a problem of obvious discrete nature, however it can be lifted into the continuous world using the *tail function* τ associated with the weight distribution of witnesses of F. The tail is a function from $\mathbb{R}_+$ to $\mathbb{N}$. For some positive number u, $\tau(u)$ answers the question "how many witnesses of F have weight heavier than u ?". One can prove that τ is a non-increasing staircase function, as illustrated in figure 1, and that its integral $\int \tau(u)du$ is exactly the discrete integral $\rho(F)$.

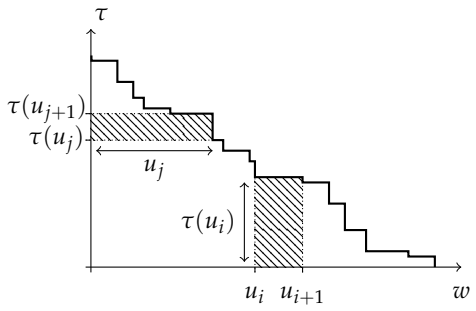

Fig. 1: The tail function

There is a bijection between the set of tail values and the set of all distinct weights of witnesses of F, so that the weight function (restricted to witnesses of F) is also expressed as a function over the tails and then extended it to a function over $\mathbb{R}_+$, denoted here by w. Graphically one gets this function rotating the graph of τ. It is then quite visual that the integral $\int w(t)dt$ is another expression for $\rho(F)$.

The first step of the framework has been to find τ and w such that $\rho(F) = \int \tau(u)du = \int w(t)dt$. These integrals are intractable and are to be approximated, as stated in the second step of the framework. Given their staircase nature, the only method fitted for approximated integration is the rectangles approximation: the integration axis is partitioned into polynomially many intervals and the integral of the non-increasing function on each interval is bounded between two rectangle areas. Choosing a partition such that for each interval, the area of the upper rectangle is twice that of the lower rectangle, enables an approximation of the integral within a factor 2. Following this idea, two estimates approximating $\rho(F)$ within a factor of 2 were designed:

1. The first estimate W_1 derives from the rectangles approximation of τ after partitioning the weight axis at $n+1$ splitting weights

$$W_1 = q_0 + \sum_{i=1}^{n} q_i 2^{i-1}$$

where the q_i are the partitioning weights defined as the 2^i quantiles of the collection of weights of witnesses of F (i.e. q_i is the maximum weight such that there exists 2^i assignment heavier than q_i). We proved that $W_1 \leq \rho(F) \leq 2W_1$. Computation of W_1 requires computation of all q_i ; discrete integration is then reduced to $n+1$ "optimization" sub-problems.
2. The second estimate W_2 derives from the rectangles approximation of w after partitioning the tail axis at $p+1$ splitting weights

$$W_2 = \tau_p 2^{-p} + \sum_{i=0}^{p-1} \tau_i 2^{-(i+1)}$$

where the τ_i are the partitioning tails defined as the tails at values 2^{-i} (i.e. τ_i is the number of witnesses of F of weight heavier than 2^{-i}). We proved that $W_2 \leq \rho(F) \leq 2W_2$. Computation of W_2 requires computation of all τ_i ; discrete integration is then reduced to $p+1$ "counting" sub-problems.

The efforts invested in this framework for discrete integration lead to 2 estimates of $\rho(F)$. Through these estimates, discrete integration is reduced to polynomially many optimization or counting sub-problems. Algorithms solving these sub-problems would then be approximating $\rho(F)$.

4 Hashing-based Algorithms for Approximate Discrete Integration

Following the conclusions of the framework described in the previous section, two algorithms were formulated: WISH, which was already discovered by Ermon et al [7], and a new algorithm: SWITCH. WISH solves approximate variant of the optimization sub-problems previously defined, thus it approximates W_1. On the other hand SWITCH solves the approximate variant of the counting sub-problems to approximate the second estimates W_2. To these ends, both implement hashing-based strategies. They rely on families of universal hashing functions defined by Wegman and Carter in their seminal work, and on an NP-oracle solving satisfiability queries for boolean formula with pseudo-boolean constraint (PB constraints). The complexity of WISH and SWITCH is then expressed in terms of number of calls to NP oracle.

WISH deals with approximate variants of the optimization problems whose solutions are necessary to computation of W_1. These problems aim at finding quantiles weights from the set of weights of witnesses of F. Hashing is used

to reduce finding quantiles to standard optimization (i.e. finding a maximum weight). The hash functions are built upon random parity constraints (XOR constraints). The idea is to keep only witnesses of F satisfying i randomly sampled XOR constraints, thus conserving in expectation $\#F/2^i$ witnesses (where $\#F$ denotes the number of witnesses of F). We prove that the maximum weight of a witness surviving i random constraints is a good approximation of the 2^i quantile weight q_i. Based on this idea, WISH implements a strategy to find, with high probability, high quality approximations of the quantile weights. Using these approximations in W_1, it generates a 8-approximation of $\rho(F)$ with high probability. Several contributions to the original algorithm of Ermon and al. were made. Optimization queries where reduced to NP oracle queries, a few algorithmic improvements on the generation of random hash functions were introduced, and, above all, the quality on the approximation of the discrete integral has been greatly increased: the original analysis used pairwise independence of the hash functions to prove a factor 256 approximation while we exploited 3-wise independence to reach an 8-approximation.

The second algorithm, SWITCH, deals with approximate variants of the counting problems to find the tails τ_i required to compute W_2. The idea is to view tails as cardinals of some subsets of witness of F and use hashing techniques to estimate these cardinalities. For a given subset of size τ_i, we successively apply constraint. Each new randomly constraint halves the remaining subset in expectation, so that the number of constraints necessary to reach the empty set can be viewed as a good approximation of $\log(\tau_i)$ and its power of 2 approaches τ_i. Each time a constraint is applied, we check if the empty set has been reached with one oracle query. Based on this idea, SWITCH implements a strategy to find, with high probability, high quality approximations of the desired tails. Using these approximations in W_2, it generates a 8-approximation of $\rho(F)$ with high probability.

The work done in the thesis shows that WISH and SWITCH can be seen as dual to each other, in the sense that their complexities differ only by a permutation of the parameter. Specifically, for F a boolean formula on n variables whose weights are written with p bits in binary representation (assuming usual encoding), and a confidence parameter δ, there exists a function f such that both WISH and SWITCH generates a 8-approximation of $\rho(F)$ with probability $\geq 1-\delta$, and WISH makes $O(f(n,p,\delta))$ calls to NP oracle against $O(f(p,n,\delta))$ for SWITCH. We have found f to be polynomial in n, p and $1/\delta$, specifically $f(n,p,\delta) = n\log(p)\log(n/\delta)$. Furthermore, the analysis shows that the constants hidden by the $\mathcal{O}$ notation are of same order of magnitude. So depending on the value of n and p, one may prefer one algorithm to the other.

References

1. Brooks S., Gelman A., Jones G., Meng X.L.: Handbook of markov chain monte carlo. Chapman & Hall/CRC (2011)
2. Carter J.L., Wegman M.N.: Universal classes of hash functions. Journal of Computer and System Sciences (1977)
3. Chakraborty S., Fremont D.J., Meel K.S., Seshia S.A., Vardi M.Y.: Distribution-aware sampling and weighted model counting for sat. In: Proc. of AAAI. pp. 1722–1730 (2014)
4. Carla P. Gomes, Ashish Sabharwal, Bart Selman: Near-Uniform Sampling of Combinatorial Spaces Using XOR Constraints. In: Proc. of NIPS. pp. 481–488 (2006)
5. Chakraborty S., Meel K.S., Vardi M.Y.: A scalable approximate model counter. In: Proc. of CP. pp. 200–216 (2013)
6. Ermon S., Gomes C., Sabharwal A., Selman B.: Embed and project: Discrete sampling with universal hashing. In: Proc. of NIPS. pp. 2085–2093 (2013)
7. Ermon S., Gomes C., Sabharwal A., Selman B.: Taming the curse of dimensionality: Discrete integration by hashing and optimization. In: Proc. of ICML. pp. 334–342 (2013)
8. Gogate V., Dechter R.: Approximate counting by sampling the backtrack-free search space. In: Proc. of the AAAI. vol. 22, p. 198 (2007)
9. Jerrum M.R., Sinclair A.: The Markov chain Monte Carlo method: an approach to approximate counting and integration. Approximation algorithms for NP-hard problems pp. 482–520 (1996)
10. Kitchen N., Kuehlmann A.: Stimulus generation for constrained random simulation. In: Proc. of ICCAD. pp. 258–265 (2007)
11. Paredes R., Duenas-Osorio L., Meel K.S., Vardi M.Y.: Network reliability estimation in theory and practice. Reliability Engineering and System Safety (2018)
12. Roth D.: On the hardness of approximate reasoning. Artificial Intelligence (1996)
13. Stockmeyer L.: The complexity of approximate counting. In: Proc. of STOC. pp. 118–126 (1983)
14. Tzikas D.G., Likas A.C., Galatsanos N.P.: The variational approximation for Bayesian inference. In: IEEE Signal Processing Magazine. pp. 131–146 (Nov 2008)
15. Valiant L.G.: The complexity of computing the permanent. Theoretical Computer Science **8**, 189–201 (1977)
16. Wainwright M.J., Jordan M.I.: Graphical models, exponential families, and variational inference. Found. Trends Machine Learning **1**(1-2), 1–305 (2008)