COMBINING REASONING AND LEARNING FOR MULTI-STAGE INFERENCE IN COMPUTATIONAL SUSTAINABILITY AND SCIENTIFIC DISCOVERY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Yexiang Xue

May 2018

COMBINING REASONING AND LEARNING FOR MULTI-STAGE INFERENCE IN COMPUTATIONAL SUSTAINABILITY AND SCIENTIFIC DISCOVERY

Yexiang Xue, Ph.D.
Cornell University 2018

Problems at the intersection of reasoning, optimization, and learning often involve multi-stage inference. For example, making decisions based on machine learning models often leads to multi-stage inference problems where probabilistic models learned from data are embedded as first-stage subproblems within a global second-stage problem for decision-making. Multi-agent reasoning also involves multi-stage inference, since the reasoning of any given agent has to incorporate the goals of the other agents. As a result, the decision processes of the other agents are embedded as first-stage subproblems within the overall decision-making problem of that agent. With formal complexities beyond NP, multi-stage inference problems are often highly intractable.

In this thesis, I introduce a novel computational framework, based on embeddings, to tackle multi-stage inference problems. Our embedding technique approximates the intractable sub-problems of a multi-stage inference problem through a series of novel representations. We then embed these representations into the global problem, effectively reducing a multi-stage inference to a single-stage inference. As one example, I present a novel way to encode the reward allocation problem for a two-stage organizer–agent game as a single-stage optimization. The encoding embeds an approximation of the agents' decision-making process into the organizer's problem in the form of linear constraints.

We apply this methodology to eBird, a well-established citizen-science program for collecting bird observations, in a game called Avicaching. Our AI-based reward allocation was shown to be highly effective, surpassing the expectations of the eBird organizers and bird conservation experts. As another example, I present a novel constant approximation algorithm to solve stochastic optimization problems which identify the optimal policy that maximizes the expectation of a stochastic objective. To tackle this problem, I propose the embedding of its intractable counting subproblems as queries to NP oracles subject to additional XOR constraints. As a result, the entire problem is encoded as a single NP-equivalent optimization. This approach outperforms state-of-the-art solvers based on variational inference and MCMC sampling, on probabilistic inference benchmarks, deep learning applications, and a novel decision-making application in network design for wildlife conservation.

I also apply the embedding technique to automated reasoning and machine learning for dimensionality reduction in scientific discovery. As one example, I propose the use of embeddings based on Fourier analysis as a compact representation of high-dimensional probability distributions. As a second example, I show how human computation, crowdsourcing, and parallel computation can identify key backdoor information, thereby drastically reducing the computation time from days to minutes in a dimensionality reduction application with complex physical constraints. Our novel integration of reasoning and learning has led to the discovery of new solar light absorbers by solving a dimensionality reduction problem to characterize the crystal structure of metal oxide materials using X-ray diffraction data.

BIOGRAPHICAL SKETCH

Yexiang Xue works with Professor Carla P. Gomes and Professor Bart Selman for his Ph.D. study in the Department of Computer Science at Cornell University. His research aims at combining large-scale constraint-based reasoning and optimization with state-of-the-art machine learning techniques to enable intelligent agents to make optimal decisions in high-dimensional and uncertain real-world applications. More specifically, his research focuses on scalable and accurate probabilistic reasoning techniques, statistical modeling of data, and robust decision-making under uncertainty. Mr. Xue's work is motivated by key problems across multiple scientific domains, including artificial intelligence, machine learning, renewable energy, materials science, citizen science, urban computing, and ecology, with an emphasis on developing cross-cutting computational methods for applications in the areas of computational sustainability and scientific discovery. Mr. Xue's work received the Innovative Application Award at IAAI-17 and was featured as the cover article and the Editor's Choice in the journal Combinatorial Science of the American Chemical Society. Prior to coming to Cornell, Mr. Xue earned his Bachelor's degree in science from the school of Electronic Engineering and Computer Science, Peking University, China in 2011.

Dedicated to my parents, Xue, L. and Zhai, X. and my darling, Yin, M.

ACKNOWLEDGEMENTS

First, let me express my sincere gratitude to my advisers, Professor Carla P. Gomes and Professor Bart Selman, for their invaluable guidance, endless passion, and grand vision. They helped me discover my academic interest and offered an enormous amount of support through this academic journey. I would like to thank my committee members, Professor John Hopcroft and Professor Thorsten Joachims, for their helpful comments and suggestions. Professor Hopcroft has inspired me to make contributions to the overall prosperity of mankind by his dedication to the improvement of education in my home country.

I would like to thank my close collaborators, Professor Stefano Ermon, Professor Bistra Dilkina, Dr. Ronan Le Bras, Dr. John Gregoire, Richard Bernstein, Dr. Xiaojian Wu, Dr. Arthur Choi, Professor Adnan Darwiche, and Dr. Ashish Sabharwal, for leading me through various stages of my Ph.D. studies. In addition, my research would not have been possible, were it not for my collaborators across multiple academic disciplines. I will forever remember these colleagues: Elizabeth Anderson, Chris Barrett, Eddy Chebelyon, Patrick Clark, Theodoros Damoulas, Ian Davies, Steve DeGloria, Alexander Flecker, Daniel Fink, Angela Fuller, Roosevelt Garcia-Villacorta, Nathan Jensen, Steve Kelling, Andrew Mude, Oscar Naibei, J. Andrew Royle, Suresh Sethi, Santosh K. Suram, Scott Steinchneider, Russell Toth, Bruce van Dover, Christopher Wood, and Yu Zheng. I would like to give special thanks to Junwen Bai, Di Chen, Anmol Kabra, Zhiyuan Li, Luming Tang, and Runzhe Yang for becoming a special part of my life as my mentees. I would also like to thank Yiwei Bai, Johan Bjorck, Wen Cao, Shuo Chen, Xilun Chen, Thomas G. Dietterich, Fei Fang, Dieqiao Feng, Gao Huang, Xun Huang, Ningzi Li, Dana Morin, Brendan Rappazzo, Qinru Shi, Tianze Shi, Adith Swaminathan, Milind Tambe, Chenhao Tan, Willem-Jan Van Hoeve, Yilun Wang, Christianne White, Wenlei Xie, Jianmin Yin, Yang Yuan, and Duhan Zhang for their inspiration.

My Ph.D. would also not have been possible without the encouragement of my father, Lening Xue, and my mother, Xiaohua Zhai, who dealt with sudden challenges to our family with unbelievable courage and optimism. Last but not least, I am indebted to my darling, Ming Yin, for her unfailing support throughout my studies.

My research was supported by the National Science Foundation (Grant Numbers OCI-0830944, CCF-0832782, CCF-1522054, ITR-0427914, DBI-1049363, DBI-0542868, DUE-0734857, IIS-0748626, IIS-0844546, IIS-0612031, IIS-1050422, IIS-0905385, IIS-0746500, IIS-1209589, AGS-0835821, CNS-0751152, CNS-0855167, CNS-1059284, DEB-110008, DBI-1356308, IIS-1344201); also, by ARO grant W911-NF-14-1-0498 as well as by the Leon Levy Foundation and the Wolf Creek Foundation. I acknowledge research collaboration and field work support from colleagues at the International Livestock Research Institute (Nairobi, Kenya), Cornell University, and UC San Diego. The materials science experiments were supported through the Office of Science of the U.S. Department of Energy under Award No. DESC0004993. Use of the Stanford Synchrotron Radiation Lightsource, SLAC National Accelerator Laboratory, was supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DEAC02-76SF00515. I also acknowledge support by the Academic Venture Fund of the Atkinson Center for a Sustainable Future. In addition, I thank the eBird citizen science program, thousands of eBird participants, and the Cornell Lab of Ornithology for their contributions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Over the last decade, artificial intelligence (AI) has achieved tremendous success in computer vision, speech recognition, language understanding, and game playing. In terms of learning, the latest machine learning technologies enable an AI system to perceive the world better than what human beings are capable of (Krizhevsky, Sutskever, & Hinton, 2012; He, Zhang, Ren, & Sun, 2016). In terms of reasoning, modern inference engines can handle sophisticated reasoning tasks that involve millions of variables and constraints (Biere, 2013). With the great progress that has been made in both fields, the time has come to address the "last mile" of AI, which is *to bridge learning and reasoning, building fully automated systems that are capable of making optimal decisions based on high-dimensional and uncertain machine learning models*.

Nevertheless, problems at the intersection of learning and reasoning pose unique challenges and require fundamentally novel thinking. First, many problems at this intersection are high dimensional in nature and require dimensionality reduction tools to extract meaningful patterns from data. Second, many problems at this intersection involve multi-stage inference which is highly intractable. For example, making decisions based on machine learning models often leads to multi-stage inference problems where probabilistic models learned from data are embedded as first-stage subproblems within a global second-stage problem for decision-making. Multi-agent reasoning also involves multi-stage inference where the reasoning of any given agent has to incorporate the reasoning of the other agents, hence the decision processes of the other agents are embedded as first-stage subproblems within the overall decision-making problem of that agent. Overall, multi-stage reasoning generally leads to highly intractable problems whose complexity lies beyond NP.

Formally, a decision problem is said to be within the NP-class if given a solution to the problem, one can verify its correctness (i.e., check whether it satisfies all the constraints or not) in polynomial time. NP-complete problems are the most difficult problems within the NP-class. A problem is NP-hard if it is at least as challenging as solving an NP-complete problem. For example, a #P-complete problem, which counts the number of feasible solutions to a NP-complete problem, is NP-hard, because knowing the number of solutions implies the feasibility of the problem.

Many problems at the intersection of learning and reasoning are beyond the NP-complexity class, because they have NP-hard problems embedded as subproblems. As an example, consider the stochastic optimization problem which finds the optimal policy intervention that maximizes the expectation of a probabilistic outcome. Typical applications of this problem include managing an asset portfolio that maximizes future return, or protecting landscape to facilitate the movement of wild animals over a large landscape. The stochastic relationship between the outcome and the policy is often modeled using a probabilistic model which is learned from data. For a given policy, computing the expectation of the probabilistic outcome requires averaging over exponentially many probabilistic outcomes, which is #P-complete in terms of formal complexity. The decision-making problem on top of the probabilistic model subsumes the #P-complete problem as a first-level subproblem, thereby constituting a highly intractable multi-level inference.

As another example, consider a multi-agent setting where each agent searches for the best action that maximizes his or her own utility function. Even when the actions of all but one of the agents are fixed, the optimization problem for that one agent is often NP-hard, because it requires optimizing over an action space whose size is exponential in the size of the input. In a more complex setting, where the utility function of one agent depends on the utility functions of the others, the reasoning of that one agent has to take into account the reasoning of the other agents as well. As a result, this leads to a multi-level inference problem with NP-hard problems of the other agents embedded as first-level subproblems. Over the years, there has been tremendous progress in solving NP-complete problems. Nevertheless, little progress has been made for highly intractable multi-level inference problems.

In this thesis, I introduce a new computational framework based on so-called *embeddings* to tackle highly intractable problems at the intersection of learning and reasoning. For multi-level inference problems beyond the NP complexity class, I propose *embedding of approximations of the intractable sub-problems into the global optimization task through a series of novel representations.* As a result, the entire multi-level inference problem can be *approximated with guarantees by a single optimization*. This novel embedding framework allows us to take advantage of recent progress in solving NP-complete problems which enables us to tackle problems of higher complexity.

I also apply the embedding technique in automated reasoning and machine learning for dimensionality reduction in scientific discovery. As one example, I propose the use of embeddings based on Fourier analysis as a compact representation of high-dimensional probability distributions. As a second example, I show how human computation, crowdsourcing, and parallel computation can identify key backdoor information, thereby drastically reducing the computation time from days to minutes in a dimensionality reduction application with complex physical constraints, motivated by the crystal structure identification problem in high-throughput materials discovery.

Our novel embedding technique is motivated by real-world problems in *computational sustainability*, a new interdisciplinary field that aims to develop inference techniques and decision-support tools for tackling high-dimensional computational problems that arise in the quest for sustainability of human, animal, and plant life far into the future.

My research would not have been possible without my collaboration with the *eBird* team of the Cornell Lab of Ornithology and the Joint Center for Artificial Photosynthesis (JCAP) at Caltech. *eBird* is a successful citizen science program of the Cornell Lab of Ornithology, which engages the general public in bird conservation. To understand the distribution and migration of birds, *eBird* enlists bird watchers to identify bird species. To date, more than 360,000 individuals have volunteered more than 400 million bird observations, which in terms of man-hours is equivalent to the work required to build several Empire State buildings. Since 2006, eBird data have been used to study a variety of scientific questions, from highlighting the impact of climate change to designing plans for conservation (Sullivan et al., 2014; Kelling et al., 2012).

Our embedding technique was also motivated by a collaboration with the Joint Center for Artificial Photosynthesis (JCAP) at Caltech to interpret X-ray diffraction data in the presence of physical constraints. We were very fortunate to deploy our system at JCAP. Materials scientists have been able to analyze thousands of X-ray diffraction patterns with our system, and the results have yielded the discovery of new materials for energy applications (Y. Xue, Bai, et al., 2017; Bai et al., 2017). Our work was featured as the cover article and the Editors' Choice in the journal *Combinatorial Science* of the American Chemical Society (Suram et al., 2016). It also received recognition with the IAAI-2017 Innovative Application Award (Y. Xue, Bai, et al., 2017).

1.1 Embedding for Multi-stage Inference beyond NP

Many problems at the intersection of machine learning, optimization, and decision-making have intractable sub-problems embedded in the global optimization problem, leading to multi-stage inference problems whose computational complexity lies beyond NP. Computing optimal strategies in a multi-agent setting is a good example: As the individual optimization problems depend on one another, each agent effectively needs to maximize his or her own utility while taking into account the fact that the other agents are solving their own optimization problems as well. Finding optimal strategies in multi-agent systems is therefore generally significantly harder than solving a single-agent optimization problem.

Many types of stochastic optimization problems are also good examples of reaching beyond NP. Such problems arise naturally in a variety of settings with decision-making under uncertainty, where the objective is to find the policy interventions that yield the best stochastic outcomes. Such stochastic decision problems have intractable counting problems embedded in the global optimization challenge. High-complexity problems can be found within machine learning as well; for example, learning a statistical model with latent variables lies beyond NP, since it involves optimizing over model parameters while marginal-

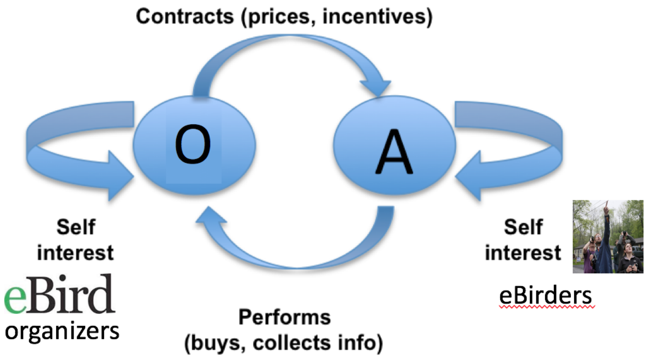

Contracts (prices, incentives)
O
A
Self interest
eBird organizers
Performs (buys, collects info)
Self interest
eBirders

Figure 1.1: The organizer–agent interaction

izing over the hidden variables.

As mentioned earlier, I propose a novel computational framework based on so-called embeddings to tackle these highly intractable problems. The general idea is to embed approximations of the intractable sub-problems into the global optimization task through a series of novel representations. As a result, the entire problem can be approximated with guarantees by a single optimization problem. The approach is quite general and can be applied in different contexts. I will illustrate the embedding idea using the following two examples. In the first example, I demonstrate the effectiveness of the embedding approach by encoding the reward allocation problem in organizer–agent games as a single optimization problem in the citizen science domain. In the second example, I provide a *constant factor approximation algorithm* to solve stochastic optimization problems, by embedding the intractable counting sub-problems into the global optimization problem as queries to NP oracles subject to additional XOR constraints.

Embedding for Optimal Reward Allocation in Organizer-Agent Games

In an organizer–agent incentive game, the organizer allocates external incentives selectively on a few tasks to encourage agents to complete a subset of crucial tasks. The agents complete tasks that maximize their own utilities, taking into consideration the external incentives offered by the organizer, subject to resource constraints. The optimal reward allocation problem is to determine an optimal plan for the organizer to allocate the rewards under a fixed budget—a plan which most effectively drives the agents to complete the crucial tasks.

As a concrete example, consider the citizen science program *eBird*, which analyzes bird distributions and migration patterns by collecting observational data from bird enthusiasts all over the world. We developed a new game, called *Avicaching*, in which the organizer, *eBird*, uses additional bonus points to incentivize avid bird watchers to go to undersampled locations, where observational data for statistical modeling are needed the most. The optimal reward allocation problem is to determine how many bonus points we should allocate for each location in order to most effectively drive bird watchers to undersampled locations. In many scenarios, it is already NP-hard to solve an individual agent's utility optimization problem. The reward allocation problem, to be solved by the organizer, embeds the agents' decision processes as subtasks, and is therefore even more challenging.

Our novel solution to the reward allocation problem (Y. Xue, Davies, Fink, Wood, & Gomes, 2016b, 2016a) considers a setting in which agents have bounded rationality and their own decision processes can be captured by a polynomial approximation scheme. We convert the polynomial approximation scheme from a procedural encoding into a declarative encoding, and then embed the declarative encoding in the form of linear constraints into the bi-level optimization. As a result, the reward allocation problem can be encoded into a single optimization problem, and can be solved using an off-the-shelf optimization package.

The effectiveness of our novel approach has been demonstrated with the field deployment of *Avicaching*, to reduce the bias in data collected within the well-established *eBird* citizen science program. Under our novel reward allocation, we were able to shift 19% of the effort in two counties in upstate New York from oversampled locations to undersampled locations during the period from April to August in 2015, thereby significantly reducing the data bias.

Embedding for Stochastic Optimization using NP Oracles and XOR Constraints

Stochastic optimization, also known as the Marginal Maximum-A-Posteriori (MMAP) problem in the probabilistic graphical model community, searches for an optimal policy that performs the best, in expectation, across multiple probabilistic scenarios. Stochastic optimization arises in a broad range of applications at the intersection of decision-making and machine learning under uncertainty, ranging from machine learning to financial engineering, computer vision, and conservation. This problem unifies two main classes of probabilistic inference, namely *maximization* (optimization) and *marginal inference* (counting), and is believed to have higher complexity than both of them (NP^{PP}).

We propose a novel approach `XOR_MMAP` (Y. Xue, Li, Ermon, Gomes, & Selman, 2016) , which gives a constant factor approximation for the stochastic opti-

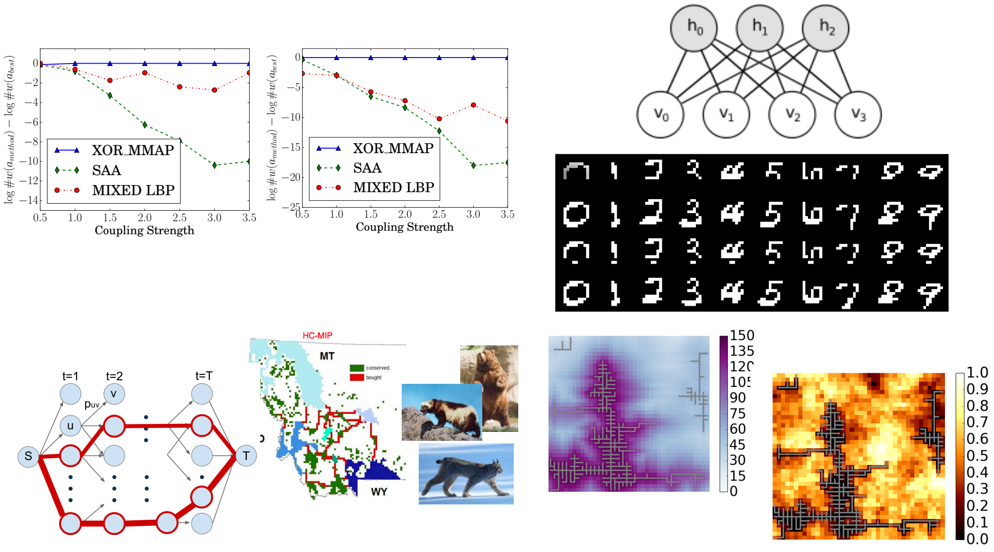


Figure 1.2: Multiple applications of solving stochastic optimization problems with XOR constraints. Probabilistic inference (upper left). Deep belief network (upper right). Network design (bottom).

mization problem. In this approach, we represent the intractable counting subproblem as queries to NP oracles, subject to additional XOR constraints. We then embed the NP oracles as sub-optimization problems into the global problem, and the entire problem becomes a single optimization. We can prove that XOR_MMAP provides a constant factor approximation for the stochastic optimization problem. We evaluated our approach on classical probabilistic inference benchmarks and on deep learning applications, as well as on a novel decision-making application in network design. We show that our approach outperforms state-of-the-art solvers based on variational inference as well as MCMC sampling.

Application to Network Design In follow-up work (Y. Xue, Wu, et al., 2017; Wu, Xue, Selman, & Gomes, 2017), we embed a spatial-capture-recapture model that estimates the density, space usage, and landscape connectivity of a given species into a dynamic landscape connectivity optimization problem. In order to scale up our encoding, we propose a sampling scheme via random partitioning of the search space with XOR constraints, closely related to the novel embedding proposed in `XOR_MMAP`. We show that our method scales to real-world-size problems and dramatically outperforms the solution quality of an expectation maximization approach and a sample average approximation approach.

1.2 Embedding for Dimensionality Reduction in Scientific Discovery

The area of automated scientific discovery presents unique challenges for the integration of learning and reasoning. I have developed several dimensionality reduction tools to facilitate the discovery of useful patterns in data with complicated physical constraints.

Dimensionality Reduction with Discrete Fourier Representation

Finding good representations of high-dimensional probability distributions is a core challenge for probabilistic methods. I introduced a novel compact representation to encode probabilistic information using the discrete Fourier transform of weighted Boolean functions (Y. Xue, Ermon, Bras, Gomes, & Selman, 2016). This approach complements the classical approach based on conditional independence. We show that a large class of probabilistic graphical models have compact Fourier representations. A variable elimination strategy that uses the Fourier representation shows superior performance on a series of probabilistic inference challenge instances.

Dimensionality Reduction with Combinatorial Constraints

Motivated by an application in materials discovery with complex physical constraints, we work on decomposing signals in a high-dimensional space into a linear combination of a few basis patterns, subject to additional physical rules. The decomposed signals are used to interpret the crystal structures of new materials.

We first encountered the phase map identification problem as part of our computational sustainability effort to address pressing problems in renewable energy. Collaborating with materials scientists, we made significant progress in this domain. We first model the phase map identification problem using a constraint reasoning approach (Ermon, Le Bras, Gomes, Selman, & van Dover, 2012). In (Le Bras, Xue, Bernstein, Gomes, & Selman, 2014), we demonstrate how human computation and crowdsourcing can speed up pattern decomposition with complex combinatorial constraints, by identifying key backdoor information.

I will introduce a novel way to boost dimensionality reduction solvers with parallel problem-solving (Y. Xue, Ermon, Gomes, & Selman, 2015). In our approach, we use parallelism to exploit hidden structure of dimensionality reduction problems with combinatorial constraints. Our approach complements divide-and-conquer and portfolio approaches to parallel problem-solving. In our scheme, a set of parallel processes are first deployed to solve a series of related subproblems. Next, the solutions to these subproblems are aggregated to obtain an initial guess for a candidate solution to the original problem. The aggregation is based on a key empirical observation that solutions to the subproblems, when properly aggregated, provide information about solutions for the original problem. Lastly, a global sequential solver searches for a solution in an iterative deepening manner, starting from the promising portion of the search space identified by the previous aggregation step.

We show that the solution time can be reduced drastically from days to minutes when we initialize a combinatorial solver with the backdoor information discovered by parallel problem-solving. To show the broad impact of this approach, we also demonstrate the effectiveness of our parallel problem decomposition on the set basis problem in combinatorial optimization.

1.3 Summary

In this thesis, I introduce a novel computational framework based on embedding to tackle problems that are at the intersection of constraint-based reasoning and machine learning, with high-dimensional and uncertain real-world applications. For multi-level inference problems beyond the NP complexity class, I propose to embed approximations of the intractable sub-problems into the global optimization task through a series of novel representations. As a result, the entire multi-level inference problem can be approximated with guarantees by a single optimization. I also apply the embedding technique to automated reasoning and machine learning in order to achieve dimensionality reduction in scientific discovery. I propose novel embeddings based on Fourier analysis as a compact representation of high-dimensional probability distributions. For each embedding strategy, I show that the resulting framework leads to a significant computational advance over previous methods. The increase in efficiency makes a new range of applications feasible. My research was motivated by key problems across multiple scientific domains, focusing on developing cross-cutting computational methods in the areas of computational sustainability and scientific discovery. Results in this thesis have been reported in the following peer-reviewed publications:

1. **Xue, Y.**, Bai, J., Le Bras, R., Rappazzo, B., Bernstein, R., Bjorck, J., Longpre, L., Suram, S. K., van Dover, R. B., Gregoire, J., Gomes, C. P. (2017). Phase-mapper: An AI platform to accelerate high throughput materials discovery. In *Proceedings of the 29th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI)*; IAAI Innovation Application Award.
2. Wu, X.*, **Xue, Y.***, Selman, B., Gomes, C. P. (2017). XOR-sampling for network design with correlated stochastic events. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*. *equal contributions.
3. **Xue, Y.**, Wu, X., Morin, D., Dilkina, B., Fuller, A., Royle, J. A., Gomes, C. P. (2017). Dynamic optimization of landscape connectivity embedding spatial-capture-recapture information. In *Proceedings of the 31th AAAI Conference on Artificial Intelligence (AAAI).*
4. Bai, J., Bjorck, J., **Xue, Y.**, Suram, S. K., Gregoire, J., Gomes, C. P. (2017). Relaxation methods for constrained matrix factorization problems: Solving the phase mapping problem in materials discovery. In *Proceedings of the 14th International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming (CPAIOR).*
5. **Xue, Y.**, Li, Z., Ermon, S., Gomes, C. P., Selman, B. (2016). Solving marginal map problems with NP oracles and parity constraints. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS).*

6. **Xue, Y.**, Davies, I., Fink, D., Wood, C., Gomes, C. P. (2016a). Avicaching: A two stage game for bias reduction in citizen science. In *Proceedings of the 15th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

7. **Xue, Y.**, Davies, I., Fink, D., Wood, C., Gomes, C. P. (2016b). Behavior identification in two-stage games for incentivizing citizen science exploration. In *Proceedings of the 22nd International Conference on Principles and Practice of Constraint Programming (CP)*.

8. **Xue, Y.**, Ermon, S., Bras, R. L., Gomes, C. P., Selman, B. (2016). Variable elimination in the Fourier domain. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*.

9. Suram, S. K., **Xue, Y.**, Bai, J., Le Bras, R., Rappazzo, B., Bernstein, R., Bjorck, J., Zhou, L., van Dover, R. B., Gomes, C. P., Gregoire, J. (2016). Automated phase mapping with AgileFD and its application to light absorber discovery in the V–MN–NB oxide system. In *American Chemical Society Combinatorial Science*; Editors Choice and Cover Story.

10. **Xue, Y.**, Ermon, S., Gomes, C. P., Selman, B. (2015). Uncovering hidden structure through parallel problem decomposition for the set basis problem: Application to materials discovery. In *Proceedings of the Twenty-fourth International Joint Conference on Artificial Intelligence (IJCAI)*.

11. Le Bras, R., **Xue, Y.**, Bernstein, R., Gomes, C. P., Selman, B. (2014). A human computation framework for boosting combinatorial solvers. In *Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*.

CHAPTER 2

EMBEDDING FOR MULTI-STAGE INFERENCE BEYOND NP

Many problems at the intersection of machine learning, optimization, and decision-making have intractable sub-problems embedded in the global optimization problem, leading to highly intractable multi-stage inference problems.

Multi-stage inference problems often arise in a multi-agent setting, in which the optimal strategy of any given agent depends on those of others. Hence, the optimal decision of the given agent has to take into account of other agents. For example, in the following two-stage game, both Alice and Bob are optimizing their objective functions, subject to their own constraints.

$$\begin{array}{lll} \textbf{(Alice)} & \text{maximize}_{\,x} & U(x,y), \\ & \text{subject to} & C_a(x), \\ \textbf{(Bob)} & & y = \text{argmax}_{\,y}\ \ V(x,y), \\ & & \qquad\qquad \text{subject to}\ \ C_b(y). \end{array} \tag{2.1}$$

Here, x and y are Alice and Bob's actions, respectively. $U(x,y)$ and $V(x,y)$ are Alice's and Bob's objective functions, both of which depend on their joint actions. $C_a(x)$ and $C_b(x)$ are their personal constraints.

Multi-agent games often lead to highly intractable problems whose complexity lies beyond NP. For example, we can reduce a Σ_2^p-complete problem to the aforementioned two-stage game in Equation. 2.1.

Theorem 2.0.1. *Assuming that $U(x,y)$ and $V(x,y)$ can encode arbitrary Boolean functions, then the two-stage game in (2.1) is Σ_2^p-hard*[1].

[1] Under restrictive cases where $U(x,y)$ and $V(x,y)$ are boolean functions with binary outputs, $C_a(x)$ and $C_b(y)$ are boolean constraints, we can prove that the decision version of the two-stage

Here, Σ_2^p denotes the complexity class of the second-level polynomial hierarchy. We prove Theorem 2.0.1 with a reduction from the following Σ_2^p-complete problem, which is to decide the truth of the following logic statement:

$$\exists x \in \{0,1\}^m \;\; \forall y \in \{0,1\}^n, \;\; F(x,y) = 1. \tag{2.2}$$

Here x and y represent two sets of disjoint binary variables. $F(x,y) : \{0,1\}^{m+n} \rightarrow \{0,1\}$ is a Boolean function with binary variables as both its input and output.

Proof. (Theorem 2.0.1) Suppose we have an oracle to solve an arbitrary two-stage game as in Equation 2.1, then we can set $U(x,y) = F(x,y)$ and $V(x,y) = -F(x,y)$. Constraint sets $C_a(x)$ and $C_b(y)$ are both set to be empty. Suppose statement (2.2) is true, then Alice can find an x such that $F(x,y)$ evaluates to 1, regardless of what Bob chooses. Hence, $V(x,y)$ is 1 in the optimal play of the two-stage game in (2.1). Conversely, if statement (2.2) is false, then for all x played by Alice, Bob is able to find a y such that $F(x,y)$ evaluates to 0. Hence, $V(x,y)$ is 0 in the optimal play of the two-stage game in (2.1). In summary, the truth value of (2.2) is equivalent to deciding if $V(x,y)$ is 1 in the optimal play of the two-stage game (2.1). □

Multi-stage inference problems also arise in stochastic optimization, which encompasses a wide variety of applications in financial engineering, optimal control, computer vision, and conservation planning. The goal of stochastic optimization is to find the best policy interventions that maximize a stochastic

game in (2.1) is in Σ_2^p, therefore it is Σ_2^p-complete. Notice that the difference between the two-stage game and the classical Nash Equilibrium setting is that Alice commits to a strategy first, anticipating all possible actions of Bob.

outcome. In mathematical language, such problem becomes

$$\max_{a \in \mathcal{A}} \; \mathbb{E}_x \, f(x, a). \tag{2.3}$$

In other words, we would like to search for the optimal assignment to variables a, such that the *expectation* of function f is maximized. The complexity of stochastic optimization problems in general lies beyond NP, because it requires to solve a counting sub-problem[2] to compute the expectation.

The challenge of tackling highly intractable problems beyond the NP is often due to the intractable sub-problems embedded in the global problem. For example, the reasoning process of other agents serve as the intractable subproblems in the two stage game in (2.1). The expectation computation is the intractable subproblem in the case of stochastic optimization (2.3).

To solve these highly intractable problems, I propose a novel computational framework to *embed approximations of the intractable sub-problems through a series of novel representations into the global optimization task.* As a result, the whole problem can be *approximated with guarantees by a single optimization problem* that can be solved by an off-the-shelf optimization package. This approach is general and can be applied in different contexts.

For example, to solve the two-stage game (2.1), we embed (an approximation of) the agents' intractable sub-problems into the global problem. The core idea is to approximate the agents' reasoning process with a tractable algorithm. We then compile this algorithm into a declarative encoding as a set of linear constraints. The compilation process mimics the execution of the approximation algorithm, introducing one constraint for each operation. After obtaining

[2] A counting problem is to count the number of solutions to an NP-complete problem. Probabilistic inference tasks, such as computing marginals and expectations, can be formulated as special cases of counting problems.

the declarative representation in the form of linear constraints, we embed them into the two-stage game, collapsing the entire problem as a single optimization.

In the first part of this section, we illustrate this idea with an example of solving a real two-stage game called *Avicaching* to reduce the data bias problem in citizen science domain. In the *Avicaching* game, the organizer, which corresponds to experts who can influence the citizen science program, uses additional bonus points to stimulate agents towards undersampled locations, where observational data is needed the most for statistical modeling. The optimal reward allocation problem is to determine how many bonus points that the organizer should allocate for each location in order to motivate the agents to go to undersampled locations the most effectively. We show that the *Avicaching* game can be encoded exactly as a two-stage game in (2.1), where the optimal reward allocation can be computed using the embedding technique. We further demonstrate the effectiveness of our reward allocation by deploying it to eBird, a well-known citizen science program. We show that our AI-based reward allocation is highly effective, surpassing the expectations of eBird organizers and conservation scientists.

Highly intractable stochastic optimization problems can be also solved using the embedding technique. In this case, we represent the intractable expectation operation as queries to optimization problems with additional XOR constraints, which in turn can be *embedded* into the global optimization task. As a result, we effectively collapse a two-stage stochastic optimization problem to a single joint inference of polynomial size of the original problem, and obtain a *constant factor approximation* guarantee.

In the second part of this section, We discuss methodologies using the embedding technique to solve stochastic optimization problems, with applications in probabilistic inference, deep learning, and network design.

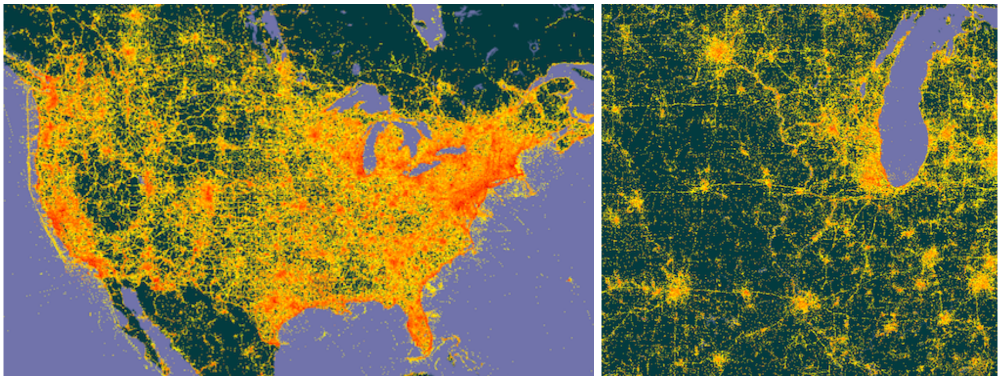


Figure 2.1: Highly biased distribution of *eBird* observations until 2014. (Left) continental US (Right) Zoom in Midwest US. Submissions coincide with urban areas.

2.1 Embedding for Bias Reduction in Citizen Science

Over the past decade, along with the emergence of the *big data* era, the data collection process for scientific discovery has evolved dramatically. One effective way of collecting large datasets is to engage the public through citizen science projects, such as *Zooniverse*, *Cicada Hunt* and *eBird* (Lintott, Schawinski, Slosar, et al., 2008; Zilli, Parson, Merrett, & Rogers, 2014; Sullivan et al., 2014). The success of these projects relies on the ability to tap into the intrinsic motivations of the volunteers to make participation enjoyable (Bonney et al., 2009). Thus in order to engage large groups of participants, citizen science projects often have few restrictions, leaving many decisions about where, when, and how to collect data up to the participants. As a result, the data collected by volunteers are often biased, more aligned with their preferences, rather than providing systematic observations across various experimental settings. Moreover, since participants volunteer their effort, personal convenience is an important factor that often determines how data are collected. For spatial data, this means more searches occur in areas close to urban areas and roads (Fig. 2.1).

We provide a general methodology to mitigate the data bias problem, as a two-stage game in which the game organizer, e.g., a citizen-science program, provides incentives to the agents, the citizen scientists, to perform more crucial scientific tasks. We apply it to *eBird*, a well-established citizen-science program for collecting bird observations, as a game called *Avicaching*.

Our proposed two-stage game is related to the Principal-Agent framework, originally studied in economics (Shavell, 1979), and more recently in computer science (Aggarwal, Feder, Motwani, & Zhu, 2004; Guruswami et al., 2005; Endriss, Kraus, Lang, & Wooldridge, 2011), and to the Stackelberg games (Conitzer & Sandholm, 2006; Conitzer & Garera, 2006; Paruchuri et al., 2008; Fang, Stone, & Tambe, 2015), which also involves e.g., a *principal* or a *leader* and *agents* or *followers*. These games have been studied under different assumptions regarding the agents' preferences and computational abilities (Hartline & Koltun, 2005; Briest, Hoefer, Gualà, & Ventre, 2009). In crowdsourcing, there has been related work on mechanisms to improve the crowd performance (Radanovic & Faltings, 2015; Li et al., 2015; Jain, Chen, & Parkes, 2014; Kawajiri, Shimosaka, & Kashima, 2014; Singla et al., 2015; Minder, Seuken, Bernstein, & Zollinger, 2012; Tran-Thanh, Huynh, Rosenfeld, Ramchurn, & Jennings, 2015; Bragg, Mausam, & Weld, 2013). Notable works include using incentives to promote exploration activities (Frazier, Kempe, Kleinberg, & Kleinberg, 2014), and steering user participation with badges (Anderson, Huttenlocher, Kleinberg, & Leskovec, 2013). (Singer & Mittal, 2013; Chen, Lin, & Zhou, 2013; Cai, Daskalakis, & Papadimitriou, 2014) discuss the optimal reward allocation to reduce the empirical risk of machine learning models.

In our two-stage game setting, the *agents* are citizen scientists maximizing their intrinsic utilities, as well as the incentives distributed by the game organizer, subject to a budget constraint. The organizer corresponds to an organization with notable influence on the citizen scientists. The *organizer* factors in the reasoning process of the citizen scientists to optimize an incentive scheme. In our setting, the game organizer's goal is to optimize an incentive scheme in order to **induce a uniform data collection process**.

We considered several models both for the organizer and the agents. Herein we present a model in which the organizer **explicitly models each agent's choice as a knapsack problem.** We refer to this two-stage game as the **Optimal Incentives for Knapsack Agents (OptIKA)** game. In a follow-up work, we provide an alternative agent behavior model based on the discrete choice model in behavioral economics (Y. Xue, Davies, et al., 2016b), which can be solved with similar embedding techniques as presented here.

We provide several novel algorithms to solve the **OptIKA** game. In particular, we propose a novel embedding technique to convert the two-stage game into a single optimization problem by embedding (an approximation of) the agents' problems into the organizer's problem. The core idea is to approximate the agents' reasoning process with a tractable algorithm. We then compile this algorithm as a set of linear constraints. The compilation process mimics exactly the execution of the algorithm, introducing one constraint for each operation. After obtaining all the linear constraints, we embed them in the bi-level optimization, collapsing the entire problem to a single optimization.

We consider **(1) different objectives** for the organizer, corresponding to different measures of data uniformity using *Mixed Integer Programming* and *Mixed Integer Quadratic Programming* formulations. We also consider **(2) different lev-**

els of rationality for the *agents*, which result in different approaches to fold the agents' knapsack problems into the organizer's problem. For the scenario in which the agents have **unbounded rationality**, we developed an **iterative, row generation method**, given the exponential number of constraints induced by agents' knapsack problems. We also consider two scenarios in which the agents have **bounded rationality**: one in which the agents use a **greedy heuristic** and another one based on a **dynamic programming (DP), polynomial time approximation scheme** for the knapsack problem. For **(3) scalability**, we use the Taylor expansion of the L2-norm and develop a novel approach based on the *Alternating Direction Method of Multipliers*. **(4)** We propose **a novel structural SVM** framework to solve the so-called **identification problem**, which learns agents' behaviors under different incentive schemes.

We applied our methodology to *eBird* as a game called **Avicaching, deploying it as a pilot study in two New York counties.** Since the inception in March 2015, **19%** of the *eBird* observations in our pilot counties shifted from traditional locations to *Avicaching* locations with no previous observations. Our field results show that our **Avicaching incentives are remarkably effective at encouraging the bird watchers to explore under-sampled areas and hence it alleviates the data bias problem in *eBird*.** We also showed that **under our Avicaching scheme, agents can cover the area more uniformly**, which leads **to higher performance on a predictive model for bird occurrence** than the no-incentive case, with the same amount of effort devoted. Our methodology is general and can be applied to other citizen science applications as well as similar scenarios, beyond citizen science.

2.1.1 Problem Formulation

We consider the setting in which citizen scientists are encouraged to conduct *scientific surveys*. For example in *eBird*, bird watchers survey a given area, and record all the interesting species observed in that area. This setting can be generalized to other scientific exploration activities. The general formulation of the two-stage game is:

$$\begin{aligned} &\textbf{(Organizer)} \text{ maximize}_{r} \quad U_o(v,r), \\ &\qquad\qquad\quad \text{subject to} \quad B_o(r), \\ &\qquad\qquad\quad \textbf{(Agents)} \quad v = \text{argmax}_{v} \;\; U_a(v,r), \\ &\qquad\qquad\qquad\qquad\qquad \text{subject to} \;\; B_a(v), \end{aligned} \tag{2.4}$$

where r is the external reward that the organizer (e.g., a citizen science program) uses to steer the agents (e.g., citizen scientists), and v are the reactions from the agents, which is the result of optimizing their own utilities. $U_o(v,r)$ and $U_a(v,r)$ are the utility functions of the organizer and agents, respectively, and $B_o(r)$ and $B_a(v)$ are their respective budget constraints.

The Organizer's Objective is to promote a balanced exploration activity, which corresponds to sending people to under-sampled areas. Let $L = \{l_1, l_2, \ldots, l_n\}$ be the set of locations, and $X_{0,i}$ the number of historical visits at location l_i at the beginning of a time period T. Suppose there are m citizen scientists $b_1, b_2, \ldots, b_m$. During time period T, each citizen scientist b_j chooses a set $L_j \subseteq L$ of locations to explore. At the end of time period T, location i received a net amount of visits $V_i = |\{l_i \in L_j : j = 1, \ldots, m\}|$ and its total number Y_i of visits corresponds to $Y_i = X_{0,i} + V_i$. We denote by $\mathbf{Y}$ the column vector $(Y_1, \ldots, Y_n)^T$ and by $\overline{\mathbf{Y}}$ the constant column vector $(\overline{Y}, \ldots, \overline{Y})^T$ where $\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$. As the organizer aims to promote a more uniform sampling effort among different locations, this objective can be expressed as the reduction $D_p = \frac{1}{n}||\mathbf{Y} - \overline{\mathbf{Y}}||_p^p$ of the difference between $\mathbf{Y}$ and $\overline{\mathbf{Y}}$. Given this definition, D_1 corresponds to the *mean absolute deviation*, while D_2 corresponds to the *sample variance*. Other objectives could be used, e.g., maximizing the entropy of the sample distribution in order to minimize the distance to a uniform distribution.[3]

The Agents' Model – Each agent is maximizing her own utility subject to her budget constraint. Namely, if a citizen scientist b_j chooses to visit location l_i, she will receive an intrinsic utility $u_{j,i}$, at a cost $c_{j,i}$. We assume that agent b_j has a given budget C_j, so the total cost of all the places explored by b_j cannot exceed C_j.

To incentivize citizen scientists to visit undersampled areas, the organizer introduces an extra incentive r_i for each location l_i. Every citizen scientist visiting location l_i receives an extra reward r_i, besides their internal utility $u_{j,i}$. For the sake of fairness, we require that these rewards only vary across locations and are the same for all agents. In addition, to make it easier to communicate with the agents, we assume that all rewards come from a fixed discrete set: $r_i \in R = \{r_1^*, \ldots, r_k^*\}$. Taking into account intrinsic utilities, external rewards and the budget constraint, the citizen scientist b_j's planning problem becomes:

$$\begin{aligned} \underset{L_j \subseteq L}{\text{maximize}} \quad & \sum_{l_i \in L_j} u_{j,i} + w_r \cdot r_i, \\ \text{subject to} \quad & \sum_{l_i \in L_j} c_{j,i} \leq C_j. \end{aligned} \tag{2.5}$$

In this formulation, $u_{j,i}$ is the intrinsic utility, r_i is the external reward, w_r is the relative importance ratio between the intrinsic utilities and the external rewards,

[3] Note: uncertainty measures, often used in active learning (Settles, 2010), are typically tied to a particular predictive model and therefore do not serve our goal of meeting multiple scientific objectives with balanced sampling. We cannot commit to improving one particular predictive model.

$c_{j,i}$ is the cost, and C_j is the total budget for an agent. Overall, combining the organizer's goal and the agents' models, the pricing problem of the **Opt**imal **I**ncentives for **K**napsack **A**gents (**OptIKA**) game is:

$$\begin{aligned} \textbf{(OptIKA)}\ \ \underset{r}{\text{minimize}}\ \ & \frac{1}{n}||\mathbf{Y} - \overline{\mathbf{Y}}||_p^p \\ \text{subject to}\ \ & L_j = \underset{L_j \subseteq L}{\text{argmax}} \sum_{i \in L_j} u_{j,i} + w_r \cdot r_i, \\ & \sum_{l_i \in L_j} c_{j,i} \leq C_j,\ j \in \{1, \ldots, m\}, \\ & r_i \in R,\ i \in \{1, \ldots, n\}. \end{aligned} \tag{2.6}$$

The Identification Problem Citizen scientists do not reveal their reward preferences to the organizer directly. Instead, the organizer must infer the agents' utility functions based on their response to different reward treatments. In our application, the identification problem is to capture the values of $u_{j,i}$ and w_r in agents' behavior model. The identification problem is related to Inverse Reinforcement Learning (Ng & Russell, 2000; Syed, Bowling, & Schapire, 2008; Hu & Wellman, 1998), in which one also assumes that the agents are optimizing for long-term rewards.

The Pricing Problem is to solve agents' decision problem in Equation. 2.6 of determining the optimal rewards to induce the desired behavior from the agents, namely sending them to undersampled areas. The challenge in solving the pricing problem is mainly that it is a two-stage game, in which has the agents' decision problems embedded as sub-problems of the bi-level optimization.

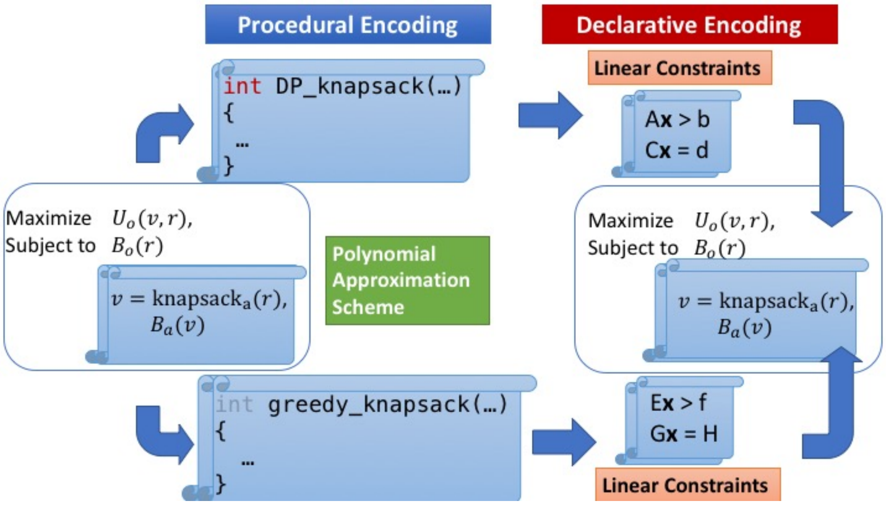


Figure 2.2: Illustration of the embedding idea to solve the pricing problem of the Avicaching game.

2.1.2 Algorithms

Pricing Problem

Embedding Technique to Solve the Pricing Problem Our novel contribution is to embed (an approximation of) the agents' problems into the organizer's problem. The core idea is to approximate the agents' reasoning process with a tractable algorithm. We then compile this algorithm as a set of linear constraints. The compilation process mimics exactly the execution of the algorithm, introducing one constraint for each operation. After obtaining all the linear constraints, we embed them in the bi-level optimization, collapsing the entire problem to a single optimization. The high-level idea is illustrated in Figure 2.2.

More specifically, we assume that agents use polynomial-time approximation schemes to solve the intractable sub-problems. In our case that the agents solve knapsack problems, we assume they use dynamic programming or greedy algorithms to come up with near optimal plans. It is valid to assume that agents in our game have bounded rationality. These polynomial approximation schemes are in *procedural form*, which cannot be embed directly into the bi-level optimization problem. We therefore compile these polynomial approximation schemes from to a *declarative form*, by mimicking their execution (detailed below). The compilation process allows us to transform agents' decision process into a set of mixed-integer linear constraints, which then can be embedded into the bi-level optimization problem.

Embedding for Dynamic Programming Agents Suppose citizen scientist b_j solves knapsack problems using a dynamic algorithm up to certain precision. We first discretize the budget C_j into N_b equal-sized units. Let $\mathcal{D}_j = \{kC_j/N_b | k = 0, \ldots, N_b\}$ be the set of all discrete units. We further round the cost $c_{j,i}$ to its nearest discrete unit from above in $\mathcal{D}_j$. We introduce extra variables $opt(j, i, c)$, for $i \in \{1, \ldots, n\}$ and $c \in \mathcal{D}_j$, to denote the optimal utility for agent b_j if we only consider the first i locations $l_1, \ldots, l_i$ and the total cost cannot exceed c. Consider the Dynamic Programming recursion to solve the Knapsack Problem:

$$opt(j,i,c) = \begin{cases} \max\{opt(j,i-1,c-c_{j,i}) + u_{j,i} + w_r \cdot r_i, & \\ \qquad opt(j,i-1,c)\}, & \text{if } i > 1, c \geq c_{j,i}, \\ opt(j,i-1,c), & \text{if } i > 1, c < c_{j,i}, \\ 0, & \text{otherwise}. \end{cases} \tag{2.7}$$

The key insight is that this recursion can be translated as a set of linear inequalities. As an example, when $i > 1$ and $c \geq c_{j,i}$, the recursion can be encoded as,

$$opt(j,i,c) \geq opt(j,i-1,c), \tag{2.8}$$

$$opt(j,i,c) \geq opt(j,i-1,c-c_{j,i}) + u_{j,i} + w_r \cdot r_i. \tag{2.9}$$

There are similar inequalities to capture other cases in Equation 2.7. Denote $u_{\mathcal{D}_j}^{knap}$ as the optimal utility for solving the knapsack problem for citizen scientist b_j. We must have $u_{\mathcal{D}_j}^{knap} \geq opt(j,n,c)$, for all $c \in \mathcal{D}_j$. Let $v_{j,i}$ be a binary variable, which is 1 if and only if $l_i \in L_j$. Using vector representations, we set $\mathbf{v}_j = (v_{j,1}, \ldots, v_{j,n})^T$, $\mathbf{u}_j = (u_{j,1}, \ldots, u_{j,n})^T$, $\mathbf{c}_j = (c_{j,1}, c_{j,2}, \ldots, c_{j,n})^T$, $\mathbf{r} = (r_1, \ldots, r_n)^T$, $\mathbf{s} = (s_1, \ldots, s_n)^T$. In summary, agent b_j's knapsack problem can be encoded as:

$$\begin{aligned} &(\mathbf{u}_j + w_r \cdot \mathbf{r})^T \cdot \mathbf{v}_j \geq u_{\mathcal{D}_j}^{knap}, \\ &\mathbf{c}_j^T \cdot \mathbf{v}_j \leq C_j \text{ and } u_{\mathcal{D}_j}^{knap} \geq opt(j,n,c), \ \forall c \in \mathcal{D}_j, \end{aligned} \tag{2.10}$$

Here, $opt(j,n,c)$ is encoded by linear inequalities similar to the ones in Equations 2.8 and 2.9. There is a multiplication in Equation 2.10. We use a big-M notation to linearize this inequality.

If we bound N_b, this encoding introduces $O(mnN_b)$ extra variables and $O(mnN_b)$ extra constraints. Notice that this encoding can be combined with the row generation approach. We can first solve the problem under limited precision using this dynamic programming encoding, then further refine the solution using the row generation approach.

Embedding for Greedy Agents Suppose each agent follows a simple greedy heuristic: first, rank all the locations based on their efficiency, i.e. the ratio between the utility (including the external reward) and the cost; then greedily select locations with the highest efficiency, without exceeding the budget limit.

This simple heuristic is a 2-approximation for the Knapsack problem, and works well in practice. Define $\psi_{j,i} = (w_r \cdot r_i + u_{j,i})/c_{j,i}$ as the efficiency of location l_i according to agent b_j. Our formulation is based on the following theorem:

Theorem 2.1.1. *Assume for all $i \neq i'$, $\psi_{j,i} \neq \psi_{j,i'}$*[4]*, then $\{v_{j,1}, \ldots, v_{j,n}\}$ is a decision made by the greedy algorithm if and only if the following two constraints hold:*

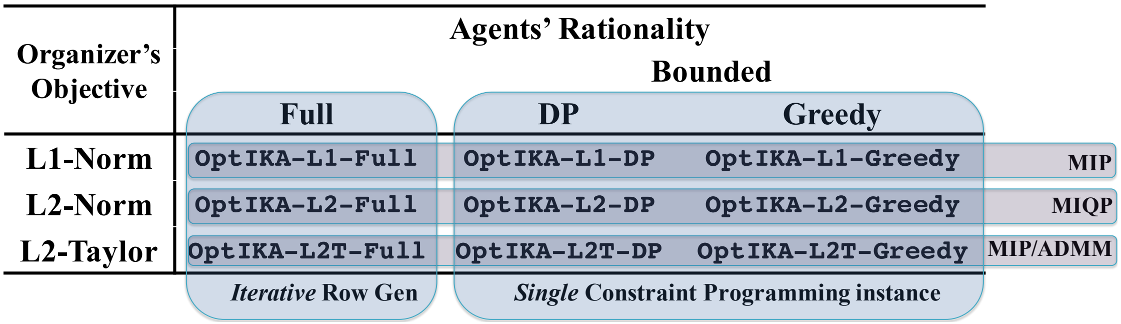
$$v_{j,i} = 0 \Rightarrow c_{j,i} > C_j - \sum_{i' \neq i} v_{j,i'} c_{j,i'} \mathbf{1}\left(\psi_{j,i'} \geq \psi_{j,i}\right), \tag{2.11}$$

for all $i \in 1, .., n$, and $\sum_{i=1}^{n} c_{j,i} \cdot v_{j,i} \leq C_j$.

In this theorem, $\mathbf{1}\left(\psi_{j,i'} \geq \psi_{j,i}\right)$ is an indicator variable, which is one if and only if $\psi_{j,i'} \geq \psi_{j,i}$. Theorem 2.1.1 translates the greedy process into a set of constraints. The intuitive meaning of inequality (2.11) says that if location l_i is not in the knapsack ($v_{j,i} = 0$), then it must be the case that some locations with higher efficiency than l_i has already taken up its space. We can use big-M notation to transform Inequality (2.11) into a set of linear constraints.

Modeling the Organizer's Objective A first measure of sample uniformity is the mean absolute deviation D_1, which allows us to formulate the objective function as a MIP. For every location l_i, introduce a variable Z_i such that $Z_i \geq |Y_i - \overline{Y}|$. Overall, the organizer's objective function can be captured as: min $\sum_{i=1}^{n} Z_i$, s.t. $Z_i \geq Y_i - \overline{Y}, Z_i \geq \overline{Y} - Y_i$. We refer to this formulation as `OptIKA-L1`. A second formulation (`OptIKA-L2`) uses the L2-norm sample variance (D_2). In this case, the organizer's objective is quadratic, and hence the entire problem becomes a Mixed Integer Quadratic Program (MIQP). As a third option, we model the organizer's objective using the Taylor approximation of the sample variance (`OptIKA-L2T`), in which case the organizer's problem translates into minimizing $S = \sum_{i=1}^{n} s_i V_i$, where $s_i = \frac{2}{n}\left(X_{0,i} - \overline{X_0}\right)$. Notice that the

[4] In practice, efficiencies almost always differ when they are learned from data.



Organizer's Objective	Agents' Rationality			
		Bounded		
	Full	DP	Greedy	
L1-Norm	OptIKA-L1-Full	OptIKA-L1-DP	OptIKA-L1-Greedy	MIP
L2-Norm	OptIKA-L2-Full	OptIKA-L2-DP	OptIKA-L2-Greedy	MIQP
L2-Taylor	OptIKA-L2T-Full	OptIKA-L2T-DP	OptIKA-L2T-Greedy	MIP/ADMM
	Iterative Row Gen	*Single* Constraint Programming instance		

Figure 2.3: Two stage game scenarios and corresponding algorithms for the pricing problem described in Section 2.1.2.

Stackelberg pricing games studied in (Guruswami et al., 2005) is a special case of **OptIKA** in this form, therefore **OptIKA** is APX-hard.

Other Algorithms for the Pricing Problem Aside from making an assumption that the agents solve their optimization problems using polynomial approximation schemes, we also developed a variety of algorithms to solve the pricing problem, capturing different organizer's objectives and agents' computational capabilities, as summarized in Fig. 2.3. More specifically, the computational capability of the agents impacts how one can embed the constraints of the agents into the organizer's problem. We consider in the previous section the case in which agents have *bounded* rationality, whether the agent solves her knapsack using a *dynamic programming*-based approach or in a *greedy* fashion. There the polynomial number of linear constraints to consider can be encoded in a single Constraint Programming instance. In this section, we further consider the case in which the agent solves her knapsack problem optimally with *full* rationality. We will see that it yields an exponential number of constraints to be handled by the organizer, thus raising scalability issues and requiring an iterative approach (see the *Row Generation* encoding below). Furthermore, in order to scale up with the number of agents, we improve our approach with a decomposition method

Algorithm 1: Row Generation-`OptIKA-LX-Full`

```
$\Phi \leftarrow \emptyset$;
$OptimalFlag \leftarrow False$;
while $OptimalFlag = False$ do
    $(\mathbf{r}_\dagger, \mathbf{v}_{1\dagger}, \ldots, \mathbf{v}_{m\dagger}) \leftarrow$ Rowgen-Relax$(\Phi)$;
    $OptimalFlag \leftarrow True$;
    for $j \in \{1, \ldots, m\}$ do
        $\mathbf{v}_j^* \leftarrow$ argmax $(\mathbf{u}_j + w_r \cdot \mathbf{r}_\dagger)^T \cdot \mathbf{v}_j,$
                subject to $\mathbf{c}_j^T \cdot \mathbf{v}_j \leq C_j$;
        if $(\mathbf{u}_j + w_r \cdot \mathbf{r}_\dagger)^T \cdot \mathbf{v}_j^* > (\mathbf{u}_j + w_r \cdot \mathbf{r}_\dagger)^T \cdot \mathbf{v}_{j\dagger}$ then
            $\Phi \leftarrow \Phi \cup \{ \ (\mathbf{u}_j + w_r \cdot \mathbf{r})^T \cdot \mathbf{v}_j \geq (\mathbf{u}_j + w_r \cdot \mathbf{r})^T \cdot \mathbf{v}_j^* \ \}$;
            $OptimalFlag = False$;
        end
    end
end
```

that decouples the agents' optimization problems (see *ADMM*). Fig. 2.3 reports all different algorithm variants in additional to the three organizer's objective functions that we considered in the previous section.

Row Generation Encoding We present the algorithm `OptIKA-LX-Full`, (where `X` is either `1`, `2` or `2T`) in which we assume the agents have full rationality, and their reasoning process is captured by an iterative row generation method. This algorithm can be combined with any of the three organizer's objectives.

The agents' optimization problem can be transformed into an exponential number of constraints of the type:

$$(\mathbf{u}_j + w_r \cdot \mathbf{r})^T \cdot \mathbf{v}_j \geq (\mathbf{u}_j + w_r \cdot \mathbf{r})^T \cdot \mathbf{v}'_j, \tag{2.12}$$

in which $\mathbf{v}'_j$ ranges over all vectors in $\{0, 1\}^n$, which satisfies $\mathbf{c}_j^T \cdot \mathbf{v}'_j \leq C_j$, for all $j \in \{1, \ldots, m\}$. The intuitive meaning of Inequality 2.12 is that the location set that the agent chooses is better in terms of utility values than any other location set within the budget constraint. We use Φ to denote a set of constraints of this form, and we write $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\} \in \Phi$ to mean that $\mathbf{v}_1, \ldots, \mathbf{v}_m$ satisfy all the constraints in Φ.

We cannot add all the constraints upfront, as there are exponentially many of them. Instead, we add them in an iterative manner until proving optimality. The row generation scheme starts by solving a relaxation of the original pricing problem: $\texttt{OptIKA-LX-Full-Relax}(\Phi)$, with a small initial constraint set Φ of constraints as shown in Inequality 2.12:

$$\begin{aligned}\texttt{OptIKA-LX-Full-Relax}(\Phi) : \text{Min: (organizer's obj) } & D_p \text{ or } S,\\ \text{s. t. } & \mathbf{c}_j^T \cdot \mathbf{v}_j \leq C_j,\ j \in \{1, .., m\},\\ & \{\mathbf{v}_1, \ldots, \mathbf{v}_m\} \in \Phi,\\ & r_i \in R,\ i \in \{1, \ldots, n\}.\end{aligned}$$

Then the algorithm seeks to enlarge the set Φ with new constraints of the form in Inequality 2.12 to further improve the objective function. This step is done by solving the Knapsack problem for each agent. If the current response of one agent b_j is not the optimal response to the Knapsack problem, then it implies that at least one constraint of the form shown in Inequality 2.12 is violated. We then add in the constraint into Φ and solve again. The whole algorithm iterates until no new constraints can be added to Φ, at which point we can prove optimality. The algorithm is shown as Algorithm 1.

There is one last subtlety: the Inequality (2.12) is not a linear one, because both $\mathbf{r}$ and $\mathbf{v}_j$ are variables. To linearize it, we bring in an extra variable $ur_{j,i}$, and we add constraints to ensure that $ur_{j,i}$ is always equal to $v_{j,i} \cdot (u_{j,i} + w_r \cdot r_i)$.

The constraints needed are:

$$
\begin{aligned}
& ur_{j,i} \geq 0, \\
& ur_{j,i} \leq u_{j,i} + w_r \cdot r_i, \\
& v_{j,i} = 0 \Rightarrow ur_{j,i} \leq 0, \qquad (2.13)
\end{aligned}
$$

In this case, Inequality (2.12) can be rewritten as $\sum_{i=1}^{n} ur_{j,i} \geq (\mathbf{u}_j + w_r \cdot \mathbf{r})^T \cdot \mathbf{v}'_j$. Eq. (2.13) is an indicator constraint, which can be linearized with the big-M formulation (Chvatal, 1983).

Scaling to Many Agents with ADMM In order to model a large number of citizen scientists, the pricing algorithm needs to be able to scale. To that end, we develop `OptIKA-L2T-ADMM`, harnessing a variant of the Alternating Direction Method of Multipliers (Boyd, Parikh, Chu, Peleato, & Eckstein, 2011; Martins, Figueiredo, Aguiar, Smith, & Xing, 2011). This approach decomposes the global problem of designing the rewards for all agents to a series of subproblems, each of which designs the optimal rewards for one agent. Then the algorithm matches the local rewards for all agents using an iterative approach. To the best of our knowledge, this is the first time that a decomposition based method is introduced to solve the optimal pricing problem. Because ADMM requires a decomposable objective function, this variant only applies to the third organizer's objective function that uses the Taylor expansion (`OptIKA-L2T`). We introduce a local copy of the reward vector for each agent b_j: $\mathbf{r}_j = (r_{j,1}, \ldots, r_{j,n})^T$, and we rewrite the global problem as:

$$
\begin{aligned}
\text{min} \quad & S = \sum_{j=1}^{m} \mathbf{s}^T \cdot \mathbf{v}_j, \\
\text{s.t.} \quad & (\mathbf{r}_j, \mathbf{v}_j) \in \Sigma_j, \quad \mathbf{r}_j = \mathbf{r}, \quad \forall j \in \{1, \ldots, m\}.
\end{aligned}
$$

In this formulation, we use $(\mathbf{r}_j, \mathbf{v}_j) \in \Sigma_j$ to mean that $\mathbf{v}_j$ is optimal for agent b_j

given rewards $\mathbf{r}_j$:

$$\begin{aligned}(\mathbf{r}_j, \mathbf{v}_j) \in \Sigma_j \iff r_{j,i} \in R,\ \forall i \in \{1, \ldots, n\},\\ \mathbf{v}_j = \text{argmax } (\mathbf{u}_j + w_r \cdot \mathbf{r}_j)^T \cdot \mathbf{v}_j,\\ \text{s.t. } \mathbf{c}_j^T \cdot \mathbf{v}_j \leq C_j.\end{aligned}$$

Our variant of the ADMM can be derived via the Augmented Lagrangian:

$$L_\rho = \sum_{j=1}^{m} \mathbf{s}^T \cdot \mathbf{v}_j + \boldsymbol{\lambda}_j^T \cdot (\mathbf{r}_j - \mathbf{r}) + (\rho/2)||\mathbf{r}_j - \mathbf{r}||_2^2.$$

in which $\boldsymbol{\lambda}_j$'s are Lagrangian multipliers, $\rho > 0$ is the penalty parameter. Our variant starts with an initial $\mathbf{r}_j^0$, $\mathbf{v}_j^0$, $\boldsymbol{\lambda}_j^0$ and $\mathbf{r}^0$, and updates the Lagrangian in an alternating manner for T steps. At the k-th step, $(\mathbf{v}_j^{k+1}, \mathbf{r}_j^{k+1})$ and $\mathbf{r}^{k+1}$ are obtained by minimizing $L_\rho(.)$ w.r.t. $(\mathbf{v}_j, \mathbf{r}_j)$ and $\mathbf{r}$, respectively. $\boldsymbol{\lambda}_j^{k+1}$ is updated by taking a subgradient step in the dual. The updates of `OptIKA-L2T-ADMM` are:

$$\begin{aligned}(\mathbf{v}_j^{k+1}, \mathbf{r}_j^{k+1}) =& \text{argmin}_{(\mathbf{v}_j, \mathbf{r}_j) \in \Sigma_j} \mathbf{s}^T \mathbf{v}_j + \boldsymbol{\lambda}_j^{kT} (\mathbf{r}_j - \mathbf{r}^k)\\ &+ (\rho/2)||\mathbf{r}_j - \mathbf{r}^k||_2^2, \qquad (2.14)\\ \mathbf{r}^{k+1} =& \frac{1}{m} \sum_{j=1}^{m} (1/\rho) \boldsymbol{\lambda}_j^k + \mathbf{r}_j^{k+1}, \qquad (2.15)\\ \boldsymbol{\lambda}_j^{k+1} =& \boldsymbol{\lambda}_j^k + \rho(\mathbf{r}_j^{k+1} - \mathbf{r}^{k+1}). \qquad (2.16)\end{aligned}$$

The difference of our variant with classical ADMM is that we impose extra constraints $(\mathbf{v}_j, \mathbf{r}_j) \in \Sigma_j$ in the first optimization step in Equation 2.14. This makes it computationally hard. In practice, we solve it via MIP, using the three encodings as described above.[5] However, the benefit of this algorithm is that the optimization problem for agent b_j is *localized*: it only involves variables and constraints for agent b_j herself, which represents a significant improvement over the previous algorithms, in which we need to consider all m agents all together in one encoding.

[5] In the case of `OptIKA-L2T-DP` (or `greedy`), Σ_j is then a relaxed constraint set, which only has constraints specified by the dynamic programming (or greedy) encoding. We use a big-M notation to handle the quadratic term $||\mathbf{r}_j - \mathbf{r}^k||_2^2$.

ADMM allows us to derive a series of interesting properties about the obtained solution. The Lagrange dual function $g(\{\boldsymbol{\lambda_j}\})$ is defined as:

$$g(\{\boldsymbol{\lambda}_j\}) = \inf_{\mathbf{r},\mathbf{v}_j,\mathbf{r}_j:(\mathbf{v}_j,\mathbf{r}_j)\in\Sigma_j} L_\rho\left(\{\mathbf{r}_j\},\{\mathbf{v}_j\},\{\boldsymbol{\lambda}_j\},\mathbf{r}\right).$$

We can view `OptIKA-L2T-ADMM` as an alternating direction descend algorithm trying to find the optimum of the optimization problem $max_{\{\boldsymbol{\lambda}_j\}}\ g(\{\boldsymbol{\lambda}_j\})$.We have the following theorectic results:

Theorem 2.1.2. *(Weak Duality) The optimal objective value to the problem* $\max_{\{\boldsymbol{\lambda}_j\}}\ g(\{\boldsymbol{\lambda}_j\})$ *is a lower bound on the optimal value of the global problem (**OptIKA**).*

Theorem 2.1.3. $g(\{\boldsymbol{\lambda}_j\})$ *is concave for* $\{\boldsymbol{\lambda}_j | j = 1 \ldots m\}$*.*

Theorem 2.1.4. *If* `OptIKA-L2T-ADMM` *converges, then* $\mathbf{r}_1, .., \mathbf{r}_m$ *and* $\mathbf{r}$ *all converge to the same vector.*

Identification Problem

In practice, parameters governing agents' preferences, such as $\mathbf{u}_j$, w_r, are unknown to us. The identification problem therefore is to learn these parameters by observing agents' reactions under different reward schemes. In our setting, we use road distance as cost $c_{j,i}$ (which is the main factor for accessibility) and we learn each agent's budget C_j from the historical mean. The variables left to estimate are the intrinsic utilities $u_{j,i}$ and the elasticity of external rewards w_r. We further assume that the intrinsic utility $u_{j,i}$ is parameterized by a set of features: $u_{j,i} = \mathbf{w}_u^T \cdot \mathbf{f}_{j,i}$, in which $\mathbf{f}_{j,i}$ includes both personal features related to agent b_j and environmental features related to location i. We assume agents are rational, therefore, their choices should always maximize the overall utility. In other words, suppose one agent chooses location set L_j, then

$\sum_{i \in L_j} \mathbf{w}_u^T \cdot \mathbf{f}_{j,i} + w_r \cdot r_i \geq \sum_{i \in L'} \mathbf{w}_u^T \cdot \mathbf{f}_{j,i} + w_r \cdot r_i$, holds for any other set of locations L', when the total distance to reach all locations in L' is within the budget. The identification problem then corresponds to finding $(\mathbf{w}_u, w_r)$ to satisfy all inequalities of this type. Because of the trivial solution $\mathbf{w}_u = \mathbf{0}$, $w_r = 0$, we aim to maximize the margin:

$$\begin{aligned} &\text{Min } ||\mathbf{w}_u||^2 + w_r^2, \\ &\text{s.t. } \sum_{i \in L_j} \mathbf{w}_u^T \cdot \mathbf{f}_{j,i} + w_r \cdot r_i \geq \sum_{i \in L'} \mathbf{w}_u^T \cdot \mathbf{f}_{j,i} + w_r \cdot r_i + \\ &\quad \Phi(L_j, L'), \quad \forall L' : \sum_{i \in L'} c_{j,i} \leq C_j. \end{aligned} \tag{2.17}$$

Here $\Phi(L_j, L')$ is a loss function, which applies different levels of penalties to location set L', depending on how similar L' and L_j are. We choose $\Phi(L_j, L') = |L_j \setminus L'| + |L' \setminus L_j|$ in the experiment. In practice, not all constraints shown in Equation 2.17 can be satisfied. Therefore, we introduce linear slack variables, and the whole identification problem becomes:

$$\begin{aligned} &\text{Min } ||\mathbf{w}_u||^2 + w_r^2 + C \sum_{j=1}^{m} \xi_j, \\ &\text{s.t. } \sum_{i \in L_j} \mathbf{w}_u^T \cdot \mathbf{f}_{j,i} + w_r \cdot r_i \geq \sum_{i \in L'} \mathbf{w}_u^T \cdot \mathbf{f}_{j,i} + w_r \cdot r_i + \\ &\quad \Phi(L_j, L') - \xi_j, \quad \forall L' : \sum_{i \in L'} c_{j,i} \leq C_j. \end{aligned} \tag{2.18}$$

This is a novel application of structural SVM (Tsochantaridis, Joachims, Hofmann, & Altun, 2005). As another contribution, we developed a modified delayed constraint generation approach to solve the optimization problem as shown in Equation 2.18, which involves solving knapsack-type problems for both the prediction and the separation problem within the structural SVM.

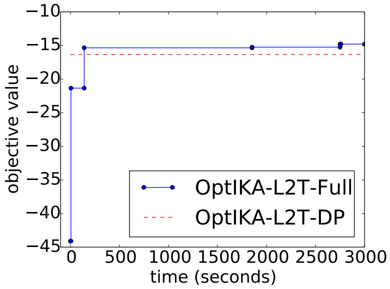


Figure 2.4: Comparison between `OptIKA-L2T-Full` and `OptIKA-L2T-DP`. (blue) Improvement of the objective function for `OptIKA-L2T-Full` over time. (red) Approximate solution value found by `OptIKA-L2T-DP` from solving a single MIP (very close to optimal, and much faster).

Method	*Red.*	$\delta\mathbf{r}$
`OptIKA-L2-Full`	44%	0
`OptIKA-L2-DP`(50)	41%	1.36
`OptIKA-L2-DP`(100)	42%	1.13
`OptIKA-L2-Greedy`	41%	1.30

Method	$\delta\mathbf{r}$
`OptIKA-L2-Full`	0
`OptIKA-L1-Full`	1.20
`OptIKA-L2T-Full`	0.74

Table 2.1: **(Left)** Comparison of different agents' rationality level. *Red.* is the L2-norm reduction w.r.t. the non-incentive case, while $\delta\mathbf{r}$ is the average hamming distance of the reward vector w.r.t. the *Full*-rational case. **(Right)** Comparison of different organizer's objectives, where $\delta\mathbf{r}$ is the average hamming dist. of the reward vector w.r.t. the L2 case.

2.1.3 Experiments

Algorithm Performance

We first compare algorithms assuming different levels of rationalities for the organizer and agents, on synthetically generated benchmarks, in which the initial number of visits $X_{0,i}$ is drawn from a geometric distribution in order to introduce some spatial bias, and other variables are drawn from uniform distributions. All the experiments are run using IBM CPLEX 12.6, on machines with a 12-core Intel x5690 3.46GHz CPU, and 48GB of memory. We implement the distributed version of the `ADMM`-based algorithms with 12 cores, in which each agent problem is allocated to one core.

Comparing Organizer's Objectives & Agents' Rationality Levels: We test our algorithms with 300 synthetic benchmarks. We fix the organizer's goal (L2-norm), and consider the case where agents are planning with different levels of rationality. The left panel of Table 2.1 reports the reduction in terms of the organizer's objective, and the mean hamming distance of the reward vectors obtained (i.e. the total number of locations in which the two reward vectors differ). Regarding the solution quality, the performance of the different approaches is similar in terms of the reduction in $L2$ and, while these approaches recommend 5 locations with positive rewards in the median case, the hamming distance between the reward vectors is barely more than 1. This suggests that the different models for the agents yield very similar results.

On the other hand, they largely differ in terms of computational complexity. When assuming full rationality of the agents, the row generation approach needs to solve multiple CPLEX instances iteratively. Fig. 2.4 depicts the running time for `OptIKA-L2T-Full` for one instance, compared with `OptIKA-L2T-DP`. As we can see, it takes the row generation algorithm a very long time to prove optimality, while `OptIKA-L2T-DP` finds a solution, close to optimal, and it is much faster. For a set of instances with 10 locations and up to 10 agents, the median completion time for the `Full` case is $1,251$ seconds, while it corresponds to 80, 93 and 38 seconds for the single MIP in the `DP` case with $N_b = 50$, $N_b = 100$ and in the `Greedy` case, respectively.

Second, we study the impact of choosing different organizer's objectives. As shown in the right panel of Table 2.1, the difference in terms of the solution quality is again very small. However, the running times vary significantly. The median completion times are $1,251$, 11 and 10 seconds for `L2`, `L1` and `L2T` objec-

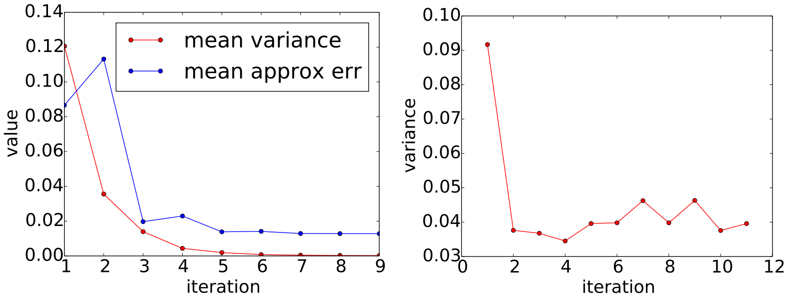


Figure 2.5: **(Left)** The mean variance and the relative error of `OptIKA-ADMM-L2T-DP` vs. iteration on small instances. **(Right)** Mean variance vs. iteration on a real *eBird* instance with 3,000 observers. ADMM converges very quickly.

tives, respectively.

Convergence of `ADMM`: In order to measure how fast `OptIKA-ADMM-L2T-DP` converges, we first run `OptIKA-ADMM-L2T-DP` for a set of small benchmarks with 20 or 30 agents and 20 locations. Although `OptIKA-ADMM-L2T-DP` works for problems of much larger scale, we still experiment with small benchmarks in order to compare it with non-decomposition based methods. In this experiment, the ADMM algorithm allocates one subproblem per agent. Because the main goal is to examine the decomposition method, each subproblem is solved by an `OptIKA-L2T-DP` module, and we compare the result with another `OptIKA-L2T-DP` which considers all agents at once. The two `OptIKA-L2T-DP` algorithms share a common discretization. The ρ for `OptIKA-ADMM-L2T-DP` is selected to be 1.

The blue line (top curve) in the left plot of Fig. 2.5 shows the relative error in the objective function as a function of the iteration number. The relative error is defined as $|S_{dp} - S_{admm}|/|S_{dp}|$, in which S_{dp} and S_{admm} are the objective values found by `OptIKA-L2T-DP` and `OptIKA-ADMM-L2T-DP`, respectively. The mean relative error is averaged among all benchmarks. As we can see, the error quickly drops from 10% to about 2% in only 3 iterations. At the same time, the red line

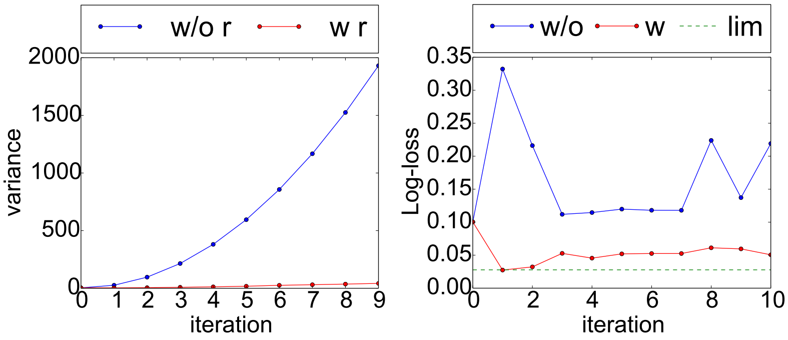


Figure 2.6: **(Left)** The change of spatial variance with and without incentives in the simulation study. **(Right)** The change of Log-loss to predict the occurrence of Horned Lark (with and without incentives). Dashed line is the performance limit.

shows how quickly the local copies $\mathbf{r}_j$ converge towards a common $\mathbf{r}$. For one benchmark, the variance is defined as: $\frac{1}{nm}\sum_{j=1}^{m}||\bar{\mathbf{r}}-\mathbf{r}_j||^2$, in which $\bar{\mathbf{r}}$ is the mean of $\mathbf{r}_1,\ldots,\mathbf{r}_m$. The mean variance is taken among all benchmarks. As we can see, the variance drops to close to zero after 3 iterations.

Next we show the performance of `OptIKA-ADMM-L2T-DP` on an instance with 63 locations and 3,000 agents, which cannot be solved by non-decomposition methods at all. The agents' behavior parameters come from real *eBird* data. We would like to emphasize that $3,000$ is enough for real use, since there are in total $2,626$ bird observers in New York State who submitted 3 or more observations in the past 10 years. The right plot of Fig. 2.5 shows the mean variance w.r.t. different iterations. Again we see that the local copies $\mathbf{r}_j$ almost converge to a common $\mathbf{r}$ in a few iterations.

Avicaching in eBird

eBird is a well-established citizen-science program for collecting bird observations. In its first years of existence, *eBird* mainly focused on appealing to birders to help address science objectives. The participation rates were disap-

Metric	SVM-struct	#Species	Popularity
Percentage Loss	**3.9%**	10.6%	8.2%
Utility Percentile	**2.3%**	46.0%	20.3%
Environmental Diff	**2.0**	5.9	4.9

Table 2.2: The Structural SVM model outperforms two other models on identifying people's behavior (#Species: model based on estimated species num; Popularity: model based on location popularity).

Year	norm D_2
2015	**0.010**
2014	0.016
2013	0.018

Treatment	norm D_2
OptIKA	**0.015**
B1: Inv-correlate #visits	0.021
B2: Uniform-in-Avicache	0.017
Manual (Expert's)	0.020

Table 2.3: **(Left)** Visits are more uniform (in normalized D_2) from April to August, 2015, when *Avicaching* is introduced, compared to previous years. **(Right)** Visits are more uniform under rewards designed by **OptIKA** against baseline B1 which assigns rewards inversely correlates to the number of visits to locations, B2 which assigns uniform rewards to all Avicaching locations, zero to others, and manually designed rewards (average over weeks of each treatment) in summer 2015.

pointing. After 3 years, in order to make participation more fun and engaging, in the spirit of "friendly competition" and "cooperation", *eBird* started providing tools to allow birders to track and rank their submissions (e.g., leaderboards by region, number of species, and number of checklists). This approach resulted in an exponential increase of submissions (Sullivan et al., 2014). Nevertheless, like most citizen-science programs, *eBird* suffers from sampling bias. Birders tend to visit locations aligned with their preferences, leading to gaps in remote areas and areas perceived as uninteresting, as shown in Fig. 2.1.

In order to address this data bias, we gamified our methodology via a web-based application called ***Avicaching***, explaining to birders that the goal of *Avi-*

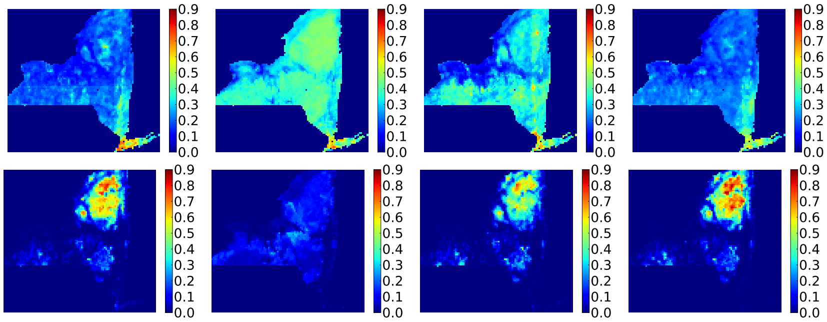


Figure 2.7: Heatmaps for the prediction of the White-throated Sparrow. (Upper 4 figures) Models for April. (Lower 4 figures) Models for July. A model trained on small, only 5% of the original data, but spatially uniform dataset (*Grid*, 2 in the leftmost column) has comparable accuracy with a model trained on the whole, big dataset that experts consider close to the ground truth (*Complete*, 2 in rightmost column), while other biased datasets have lower accuracy (*Urban*, 2nd column, *Random Subsample*, 3rd column).

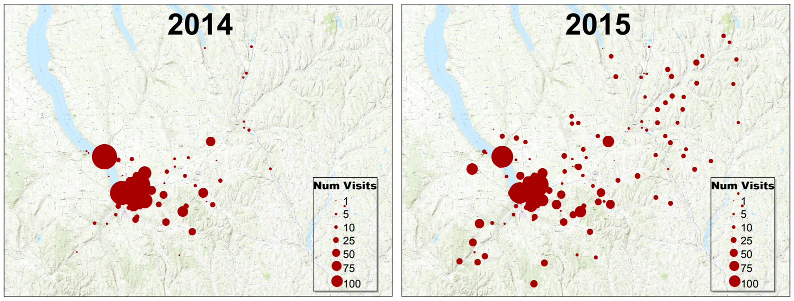


Figure 2.8: The number of eBird submissions in Tompkins and Cortland County in New York State. The circle sizes represent the number of submissions in each location. (Left) from Mar 28 to Oct 31, 2014 before *Avicaching*. (Right) from Mar 28 to Oct 31, 2015 when *Avicaching* is in the field. 19% effort is shifted to undersampled *Avicaching* locations.

caching is to “increase eBird data density in habitats that are generally underrepresented by normal eBirding”. We deployed *Avicaching* as a pilot study in Tompkins and Cortland counties, NY, starting in March 2015. Tompkins is known for a high participation rate for *eBird*, while surprisingly, Cortland, a county adjacent to Tompkins, receives much fewer observations. We identified a set of locations with no prior observations and defined them as *Avicache* locations: bird watchers receive extra *avicache* points for every checklist submitted in those locations. The locations were selected around undercovered regions, emphasizing on important yet undersampled land types. We also ensure that all locations are publicly accessible. *Avicache* points have intrinsic values to bird watchers, because these points mark their contribution to science. In addition, participants have a chance to win a pair of binoculars from a lottery drawn based on their *avicache* points.

Pricing and Agents' Model We update the *Avicaching* points every week. In the first few weeks, the allocation of points is manually assigned, based only on the number of previous visits to locations. This phase is designed to collect data to fit participants' behavior model. After the initial phase, the points are assigned by the pricing algorithm (`OptIKA-L1-DP`). We fit the agents' model using data from the two counties in 2015 (with Avicaching rewards), as well as data from the same season in 2013 and 2014 (without rewards). The results of the structural model are shown in Table 2.2, in which we predict the location set that people will visit per week. We randomly split all the data into a 90% training set and a 10% test set. The scores shown in the table are evaluated on the separate test set. The first measure is the the percentage loss: $\frac{1}{n}(|L_{pred} \setminus L_{true}| + |L_{true} \setminus L_{pred}|)$, which is the difference between the predicted location set with respect to the ground truth set. As shown in the table, **the mean percentage loss is merely 4%. This result is remarkably good, especially taking into account of the fact that we are modeling complex and noisy human behaviors.** We also look at how good our model is in terms of capturing people's rationality. Ideally, we would like to see that our model always ranks the ground truth behavior the highest in terms of the utility score. Yet, this is impossible, because human beings occasionally take suboptimal actions. In the utility percentile row, we show the percentile of the ground truth actions in terms of the utility scores among all valid actions. For example, the score 2.3% means that on average the utility scores of the ground truth actions are ranked at top 2.3% among all valid actions. Because the action set is big, we sample 10,000 location sets per test point. The low rank indicates that people are indeed motivated by the utilities defined in our model. Finally, the third row shows the difference of the environmental variables (NLCD values (Homer et al., 2007), normalized) between the predicted location set and the ground truth set. We compare our learned model with two other models. One chooses the set of locations which maximizes the estimated number of species (column #Species), and the other maximizes the total popularity of locations (column Popularity). These are the *two main* factors when planning a trip, according to expert opinions and birders' surveys. In summary, our model is quite good at capturing agents' preferences.

Field Results and Simulation We are delighted to see that people's behaviors are changing with *Avicaching*. These results compared citizen scientists' participation during the same period of time in the year 2015, when *Avicaching* was in the field, and in the year of 2014, before the introduction of *Avicaching*. Results include weeks during which **OptIKA**, which assumes that agents solve knapsack problems as discussed in this thesis, were deployed, as well as weeks during which the discrete choice behavior model of the followup work (Y. Xue, Davies, et al., 2016b), were deployed.

1. Between Mar 28, 2015 and Aug 31, 2015, there have been 1,021 observations submitted from *Avicaching* locations, out of the 5,376 observations in total for these two counties: **19% birding effort has shifted from tradi-**

tional locations to *Avicaching* locations, which received zero visits before. A few new birders also became more active, motivated by the *Avicaching* game.

2. In terms of locations, Cortland, an undersampled county, received only 128 observations from April to August in 2013 and 2014 combined. This year during the same period of time, with *Avicaching*, it received **452** observations, over **3.5 times the total number of observations of the previous 2 years**!

3. Serious bird watchers are motivated to participate in *Avicaching*. **14 out of the Top 20 bird watchers in Tompkins and 15 out of the Top 20 bird watchers in Cortland (ranked by the number of species discovered since 2015) participated in *Avicaching*.** People who participated in *Avicaching* submitted 64% of total observations in Tompkins and Cortland, from April to August, 2015.

4. In terms of whether *Avicaching* is useful to motivate people to visit undersampled areas, we compare **OptIKA** against two baselines and a manually designed scheme. To eliminate time effects, we ensure that the results against baselines were all collected in summertime, with treatments interleaved. All baselines and **OptIKA** were given two weeks time. The numbers of locations receiving each level of rewards were kept the same for B1, **OptIKA**, and manual. The non-zero reward in B2 matches the mean of other treatments. Table 2.3 shows the comparison on the normalized D_2 score, which is $\frac{1}{n}||\mathbf{Y} - \overline{\mathbf{Y}}||_2/\overline{Y}$. The visits are more uniform in 2015, when *Avicaching* is introduced. Moreover, **OptIKA** wins against baselines in terms of uniformity. Figure 2.8 provides a visual confirmation on the map. The success of *Avicaching* is the combination of the gamification

and the algorithm. It is difficult to isolate the algorithm's contribution, because field implementation is time-consuming and we cannot afford to alienate the community with drastic or complicated experiments. The **OptIKA** algorithm is better in our experiment, but simpler algorithms may also work, especially at a small scale. However, they are likely to perform worse for new scenarios or over a large scale.

5. We further simulate, for a longer period, a set of virtual agents whose behaviors are learned from the ***real bird watchers***. At the end of each round, we fit a predictive model based on all the data virtually collected so far, to see how much the species distribution model can be affected by agents' shifts of exploration efforts. We use the collected data to predict the occurrence of the Horned Lark in Spring. The left plot of Fig. 2.6 illustrates the sample variance D_2 as a function of the number of iterations with and without extra incentives. The right plot of Fig. 2.6 shows the Log-loss of the fitted predictive model in the first few iterations. This simulation shows that ***under the Avicaching scheme, agents cover the area more uniformly*** than the no-incentive case, which leads ***to higher performance on a predictive model for bird occurrence***, with the same amount of effort devoted.

Power of Uniform Sampling Finally, we also illustrate the benefit of incentivizing people to sample areas uniformly, by comparing the performance of a random forest classifier trained on four datasets, subsampled in different ways from the real *eBird* dataset. In Fig. 2.7, we show that the predictive model fit on a small, but spatially uniformly subsampled dataset is close to the ground truth, and outperforms the model fit on biased datasets.

2.1.4 Discussion

We introduced a methodology to improve the scientific quality of data collected by citizen scientists, by providing incentives to shift their efforts to more crucial scientific tasks. We formulated the problem of Optimal Incentives for Knapsack Agents (**OptIKA**) as a two-stage game and provided novel algorithms on optimal reward design and on behavior modeling. Our algorithms are based on a novel embedding idea to convert the two-stage game into a single optimization problem by embedding (an approximation of) the agents' problems into the organizer's problem.

In our follow-up work (Y. Xue, Davies, et al., 2016b), we develop a probabilistic agent behavioral model that takes into account variable patterns of human behavior and suboptimal actions, adapting ideas from discrete choice modeling in behavioral economics. By modeling deviations from baseline behavior, we are able to accurately predict future agent behavior based on limited, sparse data. Similar to the knapsack model, we provide a novel scheme to embed the agent model into a bi-level optimization as a single Mixed Integer Program, and scale up our approach by adding redundant constraints, based on insights of an easy-hard-easy phase transition phenomenon.

We applied our methodology to *eBird* as a gamified application called *Avicaching*, deploying it in two NY counties. Our results show that our incentives are remarkably effective at steering the bird watchers' efforts to explore undersampled areas, which alleviates the data bias problem and improves species distribution modeling.

2.2 Embedding for Stochastic Optimization with XOR Constraints

Stochastic optimization problems arise naturally in the context of decision-making under uncertainty, where the goal is to find a decision that maximizes the expectation of a stochastic function across multiple probabilistic scenarios. It is challenging because it integrates optimization into probabilistic inference. Stochastic optimization arises in a broad selection of applications at the intersection of machine learning and decision-making, ranging from financial engineering, optimal control, computer vision, and conservation planning.

In machine learning, stochastic optimization is often formulated as the so-called Marginal Maximum a Posteriori (MMAP) problem, which unifies the *maximum a posteriori* (MAP) inference, which computes the most likely assignment of a set of variables, as well as the *marginal* inference, which computes the marginal probability of an event. MMAP problems arise naturally in many machine learning applications. For example, learning latent variable models can be formulated as a MMAP inference, where the goal is to optimize over the model's parameters while marginalizing all the hidden variables.

Stochastic optimization is known to be $\mathrm{NP}^{\mathrm{PP}}$-complete (Park & Darwiche, 2004), which is commonly believed to be harder than both MAP inference (NP-hard) and marginal inference (#P-complete). As supporting evidence, stochastic optimization are NP-hard even on tree structured probabilistic graphical models (Liu & Ihler, 2013). Aside from attempts to solve this problem exactly (Park & Darwiche, 2003; Marinescu, Dechter, & Ihler, 2014, 2015; Mauá & de Campos, 2012), previous approximate approaches fall into two categories, in general. The core idea of approaches in both categories is to effectively approximate the intractable marginalization step, which often involves averaging over an exponentially large number of terms for the expectation. One class of approaches (Liu & Ihler, 2013; Jiang, Rai, & III, 2011; Ping, Liu, & Ihler, 2015; Lee, Marinescu, Dechter, & Ihler, 2016) use variational forms to represent the intractable expectation. Then the entire problem can be solved with message passing algorithms, which correspond to searching for the best variational approximation in an iterative manner. As another family of approaches, Sample Average Approximation (SAA) (Sheldon et al., 2010; S. Xue, Fern, & Sheldon, 2015) uses a fixed set of samples to represent the intractable expectation, which then transforms the entire problem into a restricted optimization, only considering a finite number of samples. Both approaches treat the optimization and marginalization components separately. However, we will show that by solving these two tasks in an integrated manner, we can obtain significant computational benefits.

Ermon et al. (Ermon, Gomes, Sabharwal, & Selman, 2013c, 2014) recently proposed an alternative approach to approximate intractable counting and marginalization problems. Their key idea is a mechanism to transform a counting problem into a series of optimization problems, each corresponding to the original problem subject to randomly generated XOR constraints. Based on this mechanism, they developed an algorithm providing a constant-factor approximation to the counting (marginalization) problem.

We propose a novel algorithm, called `XOR_MMAP`, again using our embedding idea. Our approach approximates the intractable expectation with a series of optimization problems. Then we *embed* these optimization problems into the global optimization task, and effectively reduce the stochastic optimization inference to *a single joint optimization* of polynomial size of the original problem.

We show that XOR_MMAP provides a constant factor approximation to the original stochastic optimization problem. Our approach also provides upper and lower bounds on the final result. The quality of the bounds can be improved incrementally with increased computational effort.

We evaluate our algorithm on unweighted SAT instances and on weighted Markov Random Field models, comparing our algorithm with variational methods and sample average approximation. We also show the effectiveness of our algorithm on applications in computer vision with deep neural networks and in computational sustainability. Our sustainability application shows how stochastic optimization is also found in scenarios of searching for optimal policy interventions to maximize the outcomes of probabilistic models. As an example, we consider a network design application to maximize the spread of cascades (Sheldon et al., 2010), which include modeling animal movements or information diffusion in social networks. In this setting, the marginals of a probabilistic decision model represent the probabilities for a cascade to reach certain target states (marginalization), and the overall network design problem is to make optimal policy interventions on the network structure to maximize the spread of the cascade (optimization). We show that XOR_MMAP is able to find considerably better solutions than those found by previous methods, as well as provide tighter bounds.

2.2.1 Preliminaries

Problem Definition Let $a = (a_1, \dots, a_m)$ be a set of binary *control variables* whose possible assignments are from $\mathcal{A} = \{0,1\}^m$. $x = (x_1, \dots, x_n)$ are a set of binary *random variables* whose possible assignments are from $\mathcal{X} = \{0,1\}^n$. $f(x, a)$ is a *stochastic function*, which depends on not only control variables a but also random variables x. A stochastic optimization problem searches for an optimal policy intervention that maximizes the *expectation* of a stochastic function. In other words, the problem becomes

$$\max_{a \in \mathcal{A}} \ \mathbb{E}_x \, f(x, a). \tag{2.19}$$

Stochastic optimization has many natural applications. For example, in robotics, a denotes a control sequence, while x represents the stochastic response of the environment. We therefore look for an optimal control sequence a so that the robot performs well *in expectation* over multiple probabilistic scenarios.

In machine learning, stochastic optimization is often formulated as a *Marginal Maximum A Posteriori* (MMAP) problem. Let $w(x, a) : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^+$ be a function that computes $\mathbb{E}_x \, f(x, a)$. In other words,

$$w(x, a) = \mathbb{E}_x \, f(x, a) = \sum_{x \in \mathcal{X}} f(x, a) Pr(x).$$

We can write a stochastic optimization problem in the following general form:

$$\max_{a \in \mathcal{A}} \ \sum_{x \in \mathcal{X}} w(x, a). \tag{2.20}$$

which is often denoted as a *Marginal Maximum A Posteriori* (MMAP) problem. We consider the case where the counting problem $\sum_{x \in \mathcal{X}} w(x, a)$ and the maximization problem $\max_{a \in \mathcal{A}} \#w(a)$ are defined over sets of exponential size, therefore both are intractable in general.

Typical queries over a probabilistic model include the *optimization* task, which requires the computation of $\max_{a \in \mathcal{A}} w(a)$, and the *marginal inference* task

$\sum_{x\in\mathcal{X}} w(x)$, which sums over $\mathcal{X}$. optimization with marginal inference, and therefore is believed to have a higher complexity (NP^{PP}-complete, proved in (Park & Darwiche, 2004)).

Counting by Hashing and Optimization Our approach is based on a recent theoretical result that transforms a counting problem to a series of optimization problems (Ermon et al., 2013c, 2014; Belle, Van den Broeck, & Passerini, 2015; Achlioptas & Jiang, 2015). A family of functions $\mathcal{H} = \{h : \{0,1\}^n \rightarrow \{0,1\}^k\}$ is said to be *pairwise independent* if the following two conditions hold for any function h randomly chosen from the family $\mathcal{H}$: (1) $\forall x \in \{0,1\}^n$, the random variable $h(x)$ is uniformly distributed in $\{0,1\}^k$ and (2) $\forall x_1, x_2 \in \{0,1\}^n$, $x_1 \neq x_2$, the random variables $h(x_1)$ and $h(x_2)$ are independent.

We sample matrices $A \in \{0,1\}^{k\times n}$ and vector $b \in \{0,1\}^k$ uniformly at random to form the function family $\mathcal{H}_{A,b} = \{h_{A,b} : h_{A,b}(x) = Ax + b \bmod 2\}$. It is possible to show that $\mathcal{H}_{A,b}$ is pairwise independent (Ermon et al., 2013c, 2014). Notice that in this case, each function $h_{A,b}(x) = Ax + b \bmod 2$ corresponds to k parity constraints. One useful way to think about pairwise independent functions is to imagine them as functions that randomly project elements in $\{0,1\}^n$ into 2^k buckets. Define $B_h(g) = \{x \in \{0,1\}^n : h_{A,b}(x) = g\}$ to be a "bucket" that includes all elements in $\{0,1\}^n$ whose mapped value $h_{A,b}(x)$ is vector g ($g \in \{0,1\}^k$). Intuitively, if we randomly sample a function $h_{A,b}$ from a pairwise independent family, then we get the following: $x \in \{0,1\}^n$ has an equal probability to be in any bucket $B(g)$, and the bucket locations of any two different elements x, y are independent.

```
Algorithm 2: XOR_Binary($w : \mathcal{A} \times \mathcal{X} \to \{0,1\}$, $a_0$, $k$)
Sample function $h_k : \mathcal{X} \to \{0,1\}^k$ from a pair-wise independent function
  family;
Query an NP Oracle on whether
    $\mathcal{W}(a_0, h_k) = \{x \in \mathcal{X} : w(a_0, x) = 1, h_k(x) = \mathbf{0}\}$ is empty;
Return true if $\mathcal{W}(a_0, h_k) \neq \emptyset$, otherwise return false.
```

2.2.2 Algorithms

Binary Case We first solve the stochastic optimization problem for the binary case, in which the function $w : \mathcal{A} \times \mathcal{X} \to \{0,1\}$ outputs either 0 or 1. We will extend the result to the weighted case in the next section. Since $a \in \mathcal{A}$ often represent decision variables, we call a fixed assignment to vector $a = a_0$ a "solution strategy". To simplify the notation, we use $\mathcal{W}(a_0)$ to represent the set $\{x \in \mathcal{X} : w(a_0, x) = 1\}$, and use $\mathcal{W}(a_0, h_k)$ to represent the set $\{x \in \mathcal{X} : w(a_0, x) = 1$ and $h_k(x) = \mathbf{0}\}$, in which h_k is sampled from a pairwise independent function family that maps $\mathcal{X}$ to $\{0,1\}^k$. We write $\#w(a_0)$ as shorthand for the count $|\{x \in \mathcal{X} : w(a_0, x) = 1\}| = \sum_{x \in \mathcal{X}} w(a_0, x)$. Our algorithm depends on the following result:

Theorem 2.2.1. *((Ermon et al., 2013c)) For a fixed solution strategy $a_0 \in \mathcal{A}$,*

- *Suppose $\#w(a_0) \geq 2^{k_0}$, then for any $k \leq k_0$, with probability $1 - \frac{2^c}{(2^c-1)^2}$, Algorithm* XOR_Binary*($w, a_0, k - c$)=**true**.*
- *Suppose $\#w(a_0) < 2^{k_0}$, then for any $k \geq k_0$, with probability $1 - \frac{2^c}{(2^c-1)^2}$, Algorithm* XOR_Binary*($w, a_0, k + c$)=**false**.*

To understand Theorem 2.2.1 intuitively, we can think of h_k as a function that maps every element in set $\mathcal{W}(a_0)$ into 2^k buckets. Because h_k comes from a pairwise independent function family, each element in $\mathcal{W}(a_0)$ will have an equal probability to be in any one of the 2^k buckets, and the buckets in which any two elements end up are mutually independent. Suppose the count of solutions for a fixed strategy $\#w(a_0)$ is 2^{k_0}, then with high probability, there will be at least one element located in a randomly selected bucket if the number of buckets 2^k is less than 2^{k_0}. Otherwise, with high probability there will be no element in a randomly selected bucket.

Theorem 2.2.1 provides us with a way to obtain a rough count on $\#w(a_0)$ via a series of tests on whether $\mathcal{W}(a_0, h_k)$ is empty, subject to extra parity functions h_k. This transforms a counting problem to a series of NP queries, which can also be thought of as optimization queries. This transformation is extremely helpful for the stochastic optimization problem. As noted earlier, the main challenge for the stochastic optimization problem is the intractable sum embedded in the maximization. Nevertheless, the whole problem can be re-written as a single optimization if the intractable sum can be approximated well by solving an optimization problem over the same domain.

We therefore design Algorithm XOR_MMAP, which is able to provide a constant factor approximation to the stochastic optimization problem. The whole algorithm is shown in Algorithm 4. In its main procedure XOR_K, the algorithm transforms the stochastic optimization problem into an optimization over the sum of T replicates of the original function w. Here, $x^{(i)} \in \mathcal{X}$ is a replicate of the original x, and $w(a, x^{(i)})$ is the original function w but takes $x^{(i)}$ as one of the inputs. All replicates share common input a. In addition, each replicate is subject to an independent set of parity constraints on $x^{(i)}$. Theorem 2.2.2 states that XOR_MMAP provides a constant-factor approximation to the stochastic optimization problem:

Theorem 2.2.2. *For $T \geq \frac{m \ln 2 + \ln(n/\delta)}{\alpha^*(c)}$, with probability $1 - \delta$,* `XOR_MMAP`*($w, \log_2 |\mathcal{X}|$, $\log_2 |\mathcal{A}|, T$) outputs a 2^{2c}-approximation to the stochastic optimization problem: $\max_{a \in \mathcal{A}} \#w(a)$. $\alpha^*(c)$ is a constant.*

Let us first understand the theorem in an intuitive way. Without losing generality, suppose the optimal value $\max_{a \in \mathcal{A}} \#w(a) = 2^{k_0}$. Denote a^* as the optimal solution, ie, $\#w(a^*) = 2^{k_0}$. According to Theorem 2.2.1, the set $\mathcal{W}(a^*, h_k)$ has a high probability to be non-empty, for any function h_k that contains $k < k_0$ parity constraints. In this case, the optimization problem $\max_{x^{(i)} \in \mathcal{X}, h_k^{(i)}(x^{(i)}) = \mathbf{0}} w(a^*, x^{(i)})$ for one replicate $x^{(i)}$ almost always returns 1. Because $h_k^{(i)}$ ($i = 1 \ldots T$) are sampled independently, the sum $\sum_{i=1}^{T} w(a^*, x^{(i)})$ is likely to be larger than $\lceil T/2 \rceil$, since each term in the sum is likely to be 1 (under the fixed a^*). Furthermore, since `XOR_K` maximizes this sum over all possible strategies $a \in \mathcal{A}$, the sum it finds will be at least as good as the one attained at a^*, which is already over $\lceil T/2 \rceil$. Therefore, we conclude that when $k < k_0$, `XOR_K` will return **true** with high probability.

We can develop similar arguments to conclude that `XOR_K` will return **false** with high probability when more than k_0 XOR constraints are added. Notice that replications and an additional union bound argument are necessary to establish the probabilistic guarantee in this case. As a counter-example, suppose function $w(x, a) = 1$ if and only if $x = a$, otherwise $w(x, a) = 0$ ($m = n$ in this case). If we set the number of replicates $T = 1$, then `XOR_K` will almost always return 1 when $k < n$, which suggests that there are 2^n solutions to the stochastic optimization problem. Nevertheless, in this case the true optimal value of $\max_x \#w(x, a)$ is 1, which is far away from 2^n. This suggests that at least two

Algorithm 3: XOR_K($w : \mathcal{A} \times \mathcal{X} \to \{0,1\}, k, T$)

```
Sample $T$ pair-wise independent hash functions
    $h_k^{(1)}, h_k^{(2)}, \dots, h_k^{(T)} : \mathcal{X} \to \{0,1\}^k$;
Query Oracle
    $$\max_{a \in \mathcal{A}, x^{(i)} \in \mathcal{X}} \sum_{i=1}^{T} w(a, x^{(i)}) \quad \text{s.t.} \quad h_k^{(i)}(x^{(i)}) = \mathbf{0}, i = 1, \dots, T. \qquad (2.21)$$
    Return **true** if the max value is larger than $\lceil T/2 \rceil$, otherwise return **false**.
```

Algorithm 4: XOR_MMAP($w : \mathcal{A} \times \mathcal{X} \to \{0,1\}$,$n = \log_2 |\mathcal{X}|$,$m = \log_2 |\mathcal{A}|$,$T$)

```
$k = n$;
while $k > 0$ do
    if XOR_K($w, k, T$) then
        Return $2^k$;
    end
    $k \leftarrow k - 1$;
end
Return 1;
```

replicates are needed.

Lemma 2.2.3. *For $T \geq \frac{\ln 2 \cdot m + \ln(n/\delta)}{\alpha^*(c)}$, procedure* XOR_K($w$,$k$,$T$) *satisfies:*

- *Suppose $\exists a^* \in \mathcal{A}$, s.t. $\#w(a^*) \geq 2^k$, then with probability $1 - \frac{\delta}{n 2^m}$,* XOR_K($w, k - c, T$) *returns* **true***.*
- *Suppose $\forall a_0 \in \mathcal{A}$, s.t. $\#w(a_0) < 2^k$, then with probability $1 - \frac{\delta}{n}$,* XOR_K($w, k + c, T$) *returns* **false***.*

Proof. **Claim 1:** If there exists such a^* satisfying $\#w(a^*) \geq 2^k$, pick $a_0 = a^*$. Let $X^{(i)}(a_0) = \max_{x^{(i)} \in \mathcal{X}, h_{k-c}^{(i)}(x^{(i)}) = \mathbf{0}} w(a_0, x^{(i)})$, for $i = 1 \dots, T$. From Theorem 2.2.1, $X^{(i)}(a_0) = 1$ holds with probability $1 - \frac{2^c}{(2^c-1)^2}$. Let $\alpha^*(c) = D(\frac{1}{2} \| \frac{2^c}{(2^c-1)^2})$. By Chernoff bound, we have

$$\Pr\left[\max_{a \in \mathcal{A}} \sum_{i=1}^{T} X^{(i)}(a) \leq T/2\right] \leq \Pr\left[\sum_{i=1}^{T} X^{(i)}(a_0) \leq T/2\right] \leq e^{-D(\frac{1}{2} \| \frac{2^c}{(2^c-1)^2})T} = e^{-\alpha^*(c)T}, \tag{2.22}$$

where

$$D\left(\frac{1}{2}\|\frac{2^c}{(2^c-1)^2}\right) = 2\ln(2^c-1) - \ln 2 - \frac{1}{2}\ln(2^c) - \frac{1}{2}\ln((2^c-1)^2 - 2^c) \geq (\frac{c}{2} - 2)\ln 2.$$

For $T \geq \frac{\ln 2 \cdot m + \ln(n/\delta)}{\alpha^*(c)}$, we have $e^{-\alpha^*(c)T} \leq \frac{\delta}{n2^m}$. Thus, with probability $1 - \frac{\delta}{n2^m}$, we have $\max\limits_{a\in\mathcal{A}} \sum_{i=1}^{T} X^{(i)}(a) > T/2$, which implies that XOR_K$(w, k-c, T)$ returns **true**.

Claim 2: The proof is almost the same as Claim 1, except that we need to use a union bound to let the property hold for all $a \in \mathcal{A}$ simultaneously. As a result, the success probability will be $1-\frac{\delta}{n}$ instead of $1-\frac{\delta}{n2^m}$. The full proof is in (Y. Xue, Li, et al., 2016). □

Proof. (Theorem 2.2.2) With probability $1 - n\frac{\delta}{n} = 1 - \delta$, the output of n calls of XOR_K(w, k, T) (with different $k = 1 \ldots n$) all satisfy the two claims in Lemma 2.2.3 simultaneously. Suppose $\max\limits_{a\in\mathcal{A}} \#w(a) \in [2^{k_0}, 2^{k_0+1})$, we have (i) $\forall k \geq k_0+c+1$, XOR_K(w, k, T) returns **false**, (ii) $\forall k \leq k_0-c$, XOR_K(w, k, T) returns **true**. Therefore, with probability $1-\delta$, the output of XOR_MMAP is guaranteed to be among 2^{k_0-c} and 2^{k_0+c}. □

The approximation bound in Theorem 2.2.2 is a worst-case guarantee. We can obtain a tight bound (e.g. 16-approx) with a large number of T replicates. Nevertheless, we keep a small T, therefore a loose bound, in our experiments, after trading between the formal guarantee and the empirical complexity. In practice, our method performs well, even with loose bounds. Moreover, XOR_K procedures with different input k are not uniformly hard. We therefore can run them in parallel. We can obtain a looser bound at any given time, based on all completed XOR_K procedures. Finally, if we have access to a polynomial approximation algorithm for the optimization problem in XOR_K, we can propagate this bound through the analysis, and again get a guaranteed bound, albeit looser for the stochastic optimization problem.

Implementation We solve the optimization problem in XOR_K using Mixed Integer Programming (MIP). Without losing generality, we assume $w(a, x)$ is an indicator variable, which is 1 iff (a, x) satisfies constraints represented in Conjunctive Normal Form (CNF). We introduce extra variables to represent the sum $\sum_i w(a, x^{(i)})$. The XORs in Equation 2.21 are encoded as MIP constraints using the Yannakakis encoding, similar as in (Ermon, Gomes, Sabharwal, & Selman, 2013b).

Extension to the Weighted Case In this section, we study the more general case, where $w(a, x)$ takes non-negative real numbers instead of integers in $\{0, 1\}$. Unlike in (Ermon et al., 2013c), we choose to build our proof from the unweighted case because it can effectively avoid modeling the median of an array of numbers (Ermon, Gomes, Sabharwal, & Selman, 2013a), which is difficult to encode in integer programming. We noticed recent work (Chakraborty, Fried, Meel, & Vardi, 2015). It is related but different from our approach. Let $w : \mathcal{A} \times \mathcal{X} \to \mathbb{R}^+$, and $M = \max_{a,x} w(a, x)$.

Definition 2.2.4. *We define the embedding $\mathcal{S}_a(w, l)$ of $\mathcal{X}$ in $\mathcal{X} \times \{0, 1\}^l$ as:*

$$\mathcal{S}_a(w, l) = \left\{ (x, y) | \forall 1 \leq i \leq l, \frac{w(a, x)}{M} \leq \frac{2^{i-1}}{2^l} \Rightarrow y_i = 0 \right\}. \tag{2.23}$$

Lemma 2.2.5. *Let $w'_l(a, x, y)$ be an indicator variable which is 1 if and only if (x, y) is in $\mathcal{S}_a(w, l)$, i.e., $w'_l(a, x, y) = \mathbf{1}_{(x,y) \in \mathcal{S}_a(w,l)}$. We claim that*

$$\max_a \sum_x w(a, x) \leq \frac{M}{2^l} \max_a \sum_{(x,y)} w'_l(a, x, y) \leq 2 \max_a \sum_x w(a, x) + M 2^{n-l}.^{6} \tag{2.24}$$

[6] If w satisfy the property that $\min_{a,x} w(a, x) \geq 2^{-l-1} M$, we do not have the $M 2^{n-l}$ term.

Proof. Define $S_a(w, l, x_0)$ as the set of (x, y) pairs within the set $S_a(w, l)$ and $x = x_0$, ie, $S_a(w, l, x_0) = \{(x, y) \in S_a(w, l) : x = x_0\}$. It is not hard to see that $\sum_{(x,y)} w'_l(a, x, y) = \sum_x |S_a(w, l, x)|$. In the following, first we are going to establish the relationship between $|S_a(w, l, x)|$ and $w(a, x)$. Then we use the result to show the relationship between $\sum_x |S_a(w, l, x)|$ and $\sum_x w(x, a)$. Case (i): If $w(a, x)$ is sandwiched between two exponential levels: $\frac{M}{2^l} 2^{i-1} < w(a, x) \le \frac{M}{2^l} 2^i$ for $i \in \{0, 1, \dots, l\}$, according to Definition 2.2.4, for any $(x, y) \in S_a(w, l, x)$, we have $y_{i+1} = y_{i+2} = \dots = y_l = 0$. This makes $|S_a(w, l, x)| = 2^i$, which further implies that

$$\frac{M}{2^l} \cdot \frac{|S_a(w, l, x)|}{2} < w(a, x) \le \frac{M}{2^l} \cdot |S_a(w, l, x)|, \tag{2.25}$$

or equivalently,

$$w(a, x) \le \frac{M}{2^l} \cdot |S_a(w, l, x)| < 2w(a, x). \tag{2.26}$$

Case (ii): If $w(a, x) \le \frac{M}{2^{l+1}}$, we have $|S_a(w, l, x)| = 1$. In other words,

$$w(a, x) \le 2w(a, x) \le 2\frac{M}{2^{l+1}} |S_a(w, l, x)| = \frac{M}{2^l} |S_a(w, l, x)|. \tag{2.27}$$

Also, $M2^{-l}|S_a(w, l, x)| = M2^{-l} \le 2w(a, x) + M2^{-l}$. Hence, the following bound holds in both cases (i) and (ii):

$$w(a, x) \le \frac{M}{2^l} |S_a(w, l, x)| \le 2w(a, x) + M2^{-l}. \tag{2.28}$$

The lemma holds by summing up over $\mathcal{X}$ and maximizing over $\mathcal{A}$ on all sides of Inequality 2.28. □

With the result of Lemma 2.2.5, we are ready to prove the following approximation result:

Theorem 2.2.6. *Suppose there is an algorithm that gives a* c*-approximation to solve the unweighted problem:* $\max_a \sum_{(x,y)} w'_l(a,x,y)$*, then we have a* $3c$*-approximation algorithm to solve the weighted stochastic optimization problem* $\max_a \sum_x w(a,x)$*.*

Proof. Let $l = n$ in Lemma 2.2.5. By definition $M = \max_{a,x} w(a,x) \leq \max_a \sum_x w(a,x)$, we have:

$$\max_a \sum_x w(a,x) \leq \frac{M}{2^l} \max_a \sum_{(x,y)} w'_l(a,x,y) \leq 2 \max_a \sum_x w(a,x) + M \leq 3 \max_a \sum_x w(a,x).$$

This is equivalent to:

$$\frac{1}{3} \cdot \frac{M}{2^l} \max_a \sum_{(x,y)} w'_l(a,x,y) \leq \max_a \sum_x w(a,x) \leq \frac{M}{2^l} \max_a \sum_{(x,y)} w'_l(a,x,y).$$

□

2.2.3 Experiments

We evaluate our proposed algorithm XOR_MMAP against two baselines – the Sample Average Approximation (SAA) (Sheldon et al., 2010) and the Mixed Loopy Belief Propagation (Mixed LBP) (Liu & Ihler, 2013). These two baselines are selected to represent the two most widely used classes of methods that approximate the embedded sum in stochastic optimization problems in two different ways. SAA approximates the intractable sum with a finite number of samples, while the Mixed LBP uses a variational approximation. We obtained the Mixed LBP implementation from the author of (Liu & Ihler, 2013) and we use their default parameter settings. Since stochastic optimization problems are in general very hard and there is currently no exact solver that scales to reasonably large instances, our main comparison is on the relative optimality gap: we first obtain the solution a_{method} for each approach. Then we compare the difference in

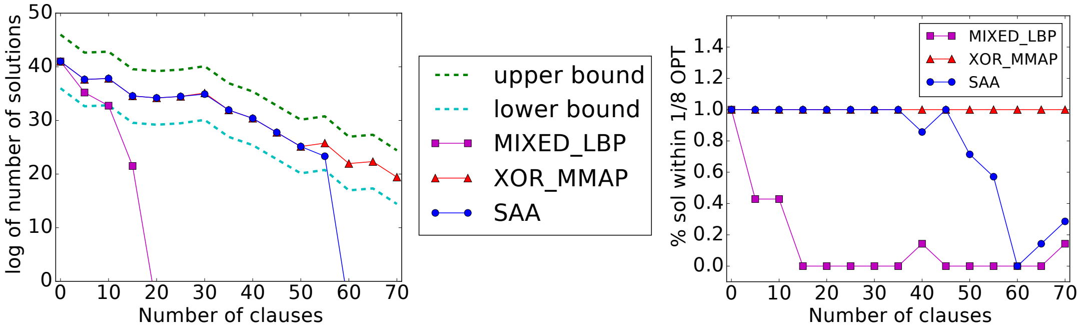


Figure 2.9: (Left) On median case, the solutions a_0 found by the proposed Algorithm XOR_MMAP have higher objective $\sum_{x\in\mathcal{X}} w(a_0, x)$ than the solutions found by SAA and Mixed LBP, on random 2-SAT instances with 60 variables and various number of clauses. Dashed lines represent the proved bounds from XOR_MMAP. (Right) The percentage of instances that each algorithm can find a solution that is at least 1/8 value of the best solutions among 3 algorithms, with different number of clauses.

objective function $\log \sum_{x\in\mathcal{X}} w(a_{method}, x) - \log \sum_{x\in\mathcal{X}} w(a_{best}, x)$, in which a_{best} is the best solution among the three methods. Clearly a better algorithm will find a vector a which yields a larger objective function. The counting problem under a fixed solution a is solved using an exact counter ACE (Chavira, Darwiche, & Jaeger, 2006), which is only used for comparing the results of different MMAP solvers.

Our first experiment is on unweighted random 2-SAT instances. Here, $w(a, x)$ is an indicator variable on whether the 2-SAT instance is satisfiable. The SAT instances have 60 variables, 20 of which are randomly selected to form set $\mathcal{A}$, and the remaining ones form set $\mathcal{X}$. The number of clauses varies from 1 to 70. For a fixed number of clauses, we randomly generate 20 instances, and the left panel of Figure 2.9 shows the median objective function $\sum_{x\in\mathcal{X}} w(a_{method}, x)$ of the solutions found by the three approaches. We tune the constants of our XOR_MMAP so it gives a $2^{10} = 1024$-approximation ($2^{-5} \cdot sol \leq OPT \leq 2^5 \cdot sol$,

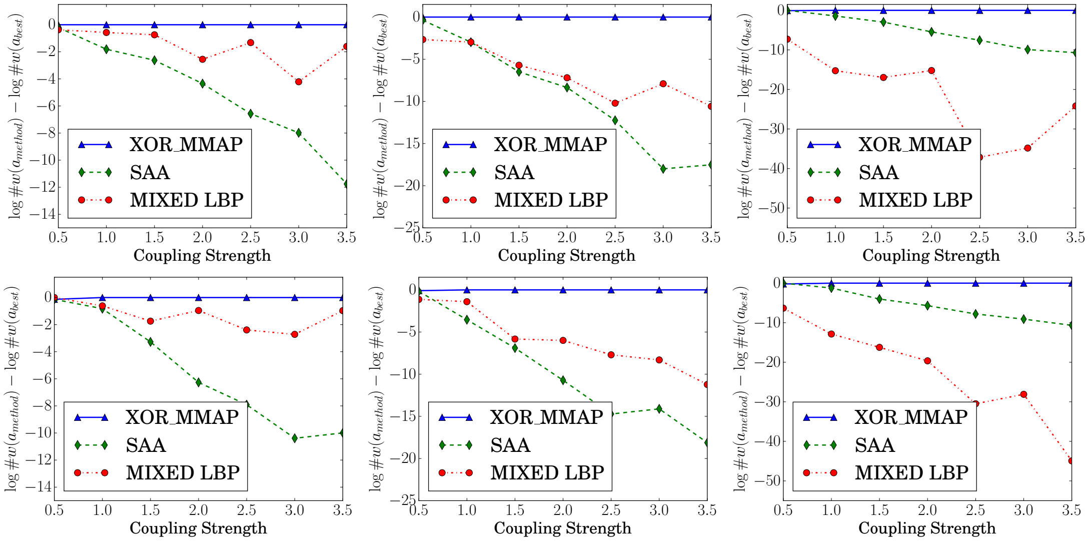


Figure 2.10: On median case, the solutions a_0 found by the proposed Algorithm XOR_MMAP are better than the solutions found by SAA and Mixed LBP, on weighted 12-by-12 Ising models with mixed coupling strength. (Up) Field strength 0.01. (Down) Field strength 0.1. (Left) 20% variables are randomly selected for maximization. (Mid) 50% for maximization. (Right) 80% for maximization.

$\delta = 10^{-3}$). The upper and lower bounds are shown in dashed lines. SAA uses 10,000 samples. On average, the running time of our algorithm is reasonable. When enforcing the 1024-approximation bound, the median time for a single XOR_k procedure is in seconds, although we occasionally have long runs (no more than 30-minute timeout).

As we can see from the left panel of Figure 2.9, both Mixed LBP and SAA match the performance of our proposed XOR_MMAP on easy instances. However, as the number of clauses increases, their performance quickly deteriorates. In fact, for instances with more than 20 (60) clauses, typically the a vectors returned by Mixed LBP (SAA) do not yield non-zero solution values. Therefore we are not able to plot their performance beyond the two values. At the same time, our algorithm XOR_MMAP can still find a vector a yielding over 2^{20} solutions on larger instances with more than 60 clauses, while providing a 1024-approximation.

Next, we look at the performance of the three algorithms on weighted instances. Here, we set the number of replicates $T = 3$ for our algorithm XOR_MMAP, and we repeatedly start the algorithm with an increasing number of XOR constraints k, until it completes for all k or times out in an hour. For SAA, we use 1,000 samples, which is the largest we can use within the memory limit. All algorithms are given a one-hour time and a 4G memory limit.

The solutions found by XOR_MMAP are considerably better than the ones found by Mixed LBP and SAA on weighted instances. Figure 2.10 shows the performance of the three algorithms on 12-by-12 Ising models with mixed coupling strength, different field strengths and number of variables to form set $\mathcal{A}$. All values in the figure are median values across 20 instances (in $\log_{10}$). In all 6 cases in Figure 2.10, our algorithm XOR_MMAP is the best among the three approximate algorithms. In general, the difference in performance increases as the coupling strength increases. These instances are challenging for the state-of-the-art complete solvers. For example, the state-of-the-art exact solver AOBB with mini-bucket heuristics and moment matching (Marinescu et al., 2015) runs out of 4G memory on 60% of instances with 20% variables randomly selected as max variables. We also notice that the solution found by our XOR_MMAP is already close to the ground-truth. On smaller 10-by-10 Ising models which the exact AOBB solver can complete within the memory limit, the median difference between the log10 count of the solutions found by XOR_MMAP and those found by the exact solver is 0.3, while the differences between the solution values of XOR_MMAP against those of the Mixed BP or SAA are on the order of 10.

We also apply our solver to an image completion task. We first learn a

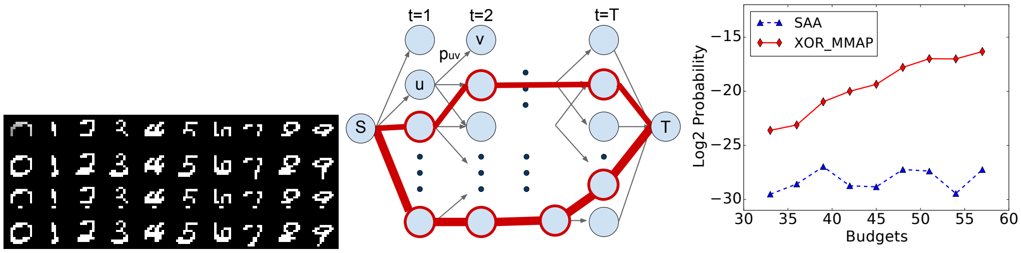


Figure 2.11: (Left) The image completion task. Solvers are given digits of the upper part as shown in the first row. Solvers need to complete the digits based on a two-layer deep belief network and the upper part. (2nd Row) completion given by XOR_MMAP. (3rd Row) SAA. (4th Row) Mixed Loopy Belief Propagation. (Middle) Graphical illustration of the network cascade problem. Red circles are nodes to purchase. Lines represent cascade probabilities. See main text. (Right) Our XOR_MMAP performs better than SAA on a set of network cascade benchmarks, with different budgets.

two-layer deep belief network (Bengio, Lamblin, Popovici, & Larochelle, 2006; G. Hinton & Salakhutdinov, 2006) from a 14-by-14 MNIST dataset. Then for a binary image that only contains the upper part of a digit, we ask the solver to complete the lower part, based on the learned model. This is a Marginal MAP task, since one needs to integrate over the states of the hidden variables, and query the most likely states of the lower part of the image. Figure 2.11 shows the result of a few digits. As we can see, SAA performs poorly. In most cases, it only manages to come up with a light dot for all 10 different digits. Mixed Loopy Belief Propagation and our proposed XOR_MMAP perform well. The good performance of Mixed LBP may be due to the fact that the weights on pairwise factors in the learned deep belief network are not very combinatorial.

Finally, we consider a stochastic optimization application that applies decision-making into machine learning models. This network design application maximizes the spread of cascades in networks, which is important in the domain of social networks and computational sustainability. In this application, we are given a stochastic graph, in which the source node at time $t = 0$ is affected. For a node v at time t, it will be affected if one of its ancestor nodes at time $t-1$ is affected, and the configuration of the edge connecting the two nodes is "on". An edge connecting node u and v has probability $p_{u,v}$ to be turned on. A node will not be affected if it is not purchased. Our goal is to purchase a set of nodes within a finite budget, so as to maximize the probability that the target node is affected. We refer the reader to (Sheldon et al., 2010) for more background knowledge. This application cannot be captured by graphical models due to global constraints. Therefore, we are not able to run mixed LBP on this problem. We consider a set of synthetic networks, and compare the performance of SAA and our `XOR_MMAP` with different budgets. As we can see from the right panel of Figure 2.11, the nodes that our `XOR_MMAP` decides to purchase result in higher probabilities of the target node being affected, compared to SAA. Each dot in the figure is the median value over 30 networks generated in a similar way.

Dynamic Optimization of Landscape Connectivity Embedding Spatial-Capture-Recapture Information Motivated by the network design application, we further propose a novel approach to dynamically optimize landscape connectivity, which an application in computational sustainability (Y. Xue, Wu, et al., 2017). Our approach is based on a mixed integer program formulation, embedding a spatial capture-recapture model that estimates the density, space usage, and landscape connectivity for a given species. Our method takes into account the fact that local animal density and connectivity change dynamically and non-linearly with different habitat protection plans. In order to scale up our encoding, we propose a sampling scheme via *random partitioning of the search* *space using parity functions*. We show that our method scales to real-world size problems and dramatically outperforms the solution quality of an expectation maximization approach and a sample average approximation approach.

2.2.4 Discussion

We propose a novel constant approximation algorithm to solve the stochastic optimization problem. Our approach represents the intractable counting subproblem with queries to NP oracles, subject to additional parity constraints. The NP queries then are embedded as optimization problems into the global problem, therefore reducing the entire problem into a single optimization. We evaluate our approach on several machine learning and decision-making applications. We are able to show that `XOR_MMAP` outperforms several state-of-the-art solvers. `XOR_MMAP` provides a new angle to solving the stochastic optimization problem, opening the door to new research directions and applications in real world domains. Future work in this direction include devising strategies to scale up this approach using shorter XOR constraints and considering iterative approaches that gradually refine solutions.

CHAPTER 3

EMBEDDING FOR DIMENSIONALITY REDUCTION IN SCIENTIFIC DISCOVERY

Many problems at the intersection of reasoning and learning are high dimensional in nature, which requires effective dimensionality reduction tools to navigate the solution space and extract meaningful information from data.

Our first dimensionality reduction application is in the domain of probabilistic inference, a key computational challenge in statistical machine learning. Inference methods have a wide range of applications, from learning models to making predictions and informing decision-making using statistical models. Unfortunately, the inference problem is computationally intractable, and standard exact inference algorithms have worst-case exponential complexity. The key to address the challenges of probabilistic inference is to *reduce high-dimensional complex probability distributions into compact forms*.

In this chapter, we first explore a novel compact representation of high-dimensional distributions based on *discrete Fourier embedding*, complementing the classical factored representation based on conditional independencies. We show that a large class of probabilistic distributions have a compact Fourier representation. This theoretical result opens up an entirely new way of approximating a high-dimensional probability distribution. We demonstrate the significance of this approach by applying it to the variable elimination algorithm for probabilistic inference. Compared with the traditional bucket representation and other approximate inference algorithms, we obtain significant improvements.

Our second dimensionality reduction application focuses on the phase map identification problem, a central task in combinatorial materials discovery, to identify the crystalline phases of inorganic compounds based on an analysis of high-intensity X-ray patterns. New materials will help us address some of the key challenges our society faces today, in terms finding a path towards a sustainable planet (White, 2012; Patel, 2011). In combinatorial materials discovery, scientists experimentally explore large numbers of combinations of different elements with the hope of finding new compounds with interesting properties, e.g., for efficient fuel cells or solar cell arrays. We are collaborating with two teams of materials scientists, one at the Department of Materials Science at Cornell and the other in the Joint Center for Artificial Photosynthesis (JCAP) at Caltech. An overall goal is to develop the capability of analyzing data from over one million new materials samples per day. Automated data analysis tools will be key to the success of this project.

The phase map identification problem is a dimensionality reduction problem. It focuses on discovering meaningful patterns corresponding to true material structures out of many X-ray diffraction patterns generated from the high-throughput experimental pipeline, which are mixed with other patterns and corrupted with noise. Unlike dimensionality reduction problems from other domains, our problem is subject to hard physical constraints, which bring additional challenges.

We first encountered the phase map identification problem as part of our Computational Sustainability effort to address pressing problems in renewable energy (Le Bras et al., 2011). Collaborating with materials scientists, we made progress in this domain with a fruitful line of research. We first model the phase map identification problem using a constraint reasoning approach (Ermon et al., 2012). In (Le Bras et al., 2014), we further integrate a state-of-the-art optimization framework based on constraint reasoning with subtle human insights, therefore drastically reducing the solution time. In (Y. Xue, Bai, et al., 2017), we developed Phase-Mapper, a comprehensive platform that tightly integrates materials science experimentation, AI problem solving, and human intelligence for combinatorial materials discovery. We have deployed our approaches at JCAP. Since its deployment, thousands of X-ray diffraction patterns have been processed and the results are yielding discovery of new materials for energy applications. Our work (Suram et al., 2016) was featured as the cover article and the Editors' Choice in the journal *Combinatorial Science* of the American Chemical Society. The work of (Y. Xue, Bai, et al., 2017) also received recognition with the IAAI-2017 Innovative Application Award.

In this thesis, we highlight a novel way to boost dimensionality reduction solvers with parallel problem solving. In our approach, we use parallelism to exploit hidden structure of dimensionality reduction problems with combinatorial constraints. Our approach complements divide-and-conquer and portfolio approaches for parallel problem solving. We first illustrate our approach on the minimum set basis problem: a core combinatorial problem with a range of applications in optimization, machine learning, and system security. Then we highlight the application of our parallel approach on combinatorial materials discovery for renewable energy sources. In our approach, a large number of smaller sub-problems are identified and solved concurrently. We then aggregate the information from those solutions, and use this information to initialize the search of a global, complete solver. We show that this strategy leads to a substantial speed-up over a sequential approach, since the aggregated subproblem solution information often provides key structural insights to the complete solver. Our approach also greatly outperforms state-of-the-art incomplete solvers in terms of solution quality. Our work opens up a novel angle for using parallelism to solve hard dimensionality reduction problems with complex constraints.

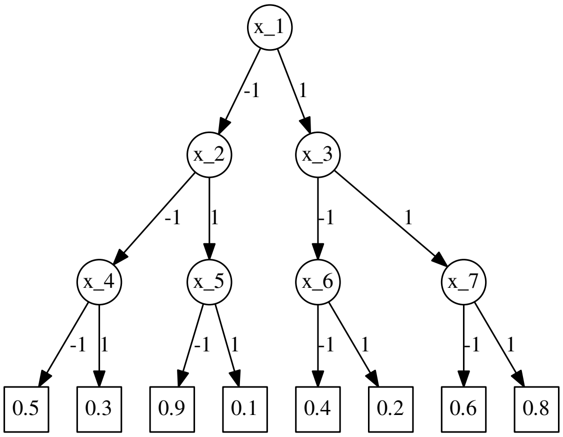


Figure 3.1: An example of a decision tree representing function $f : \{x_1, \ldots, x_7\} \to \mathcal{R}^+$.

3.1 Dimensionality Reduction with Discrete Fourier Representation

The ability to represent complex high dimensional probability distributions in a compact form is perhaps the most important insight in the field of probabilistic inference. The fundamental idea is to exploit (conditional) independencies between the variables to achieve compact *factored* representations, where a complex global model is represented as a product of simpler, local models. Similar ideas have been considered in the analysis of Boolean functions and logical forms (Dechter, 1997), as well as in physics with low rank tensor decompositions and matrix product states representations (Jordan, Ghahramani, Jaakkola, & Saul, 1999; Linden, Smith, & York, 2003; Sontag, Meltzer, Globerson, Jaakkola, & Weiss, 2008; Friesen & Domingos, 2015).

Compact representations are also key for the development of efficient inference algorithms, including message-passing ones. Efficient algorithms can be developed when messages representing the interaction among many variables can be decomposed or approximated with the product of several smaller messages, each involving a subset of the original variables. Numerous approximate and exact inference algorithms are based on this idea (Bahar et al., 1993; Flerova, Ihler, Dechter, & Otten, 2011; Mateescu, Kask, Gogate, & Dechter, 2010; Gogate & Domingos, 2013; Wainwright, Jaakkola, & Willsky, 2003; Darwiche & Marquis, 2002; Ihler, Flerova, Dechter, & Otten, 2012; Hazan & Jaakkola, 2012).

Conditional independence (and related factorizations) is not the only type of structure that can be exploited to achieve compactness. For example, consider the weighted decision tree in Figure 3.1. No two variables in the probability distribution in Figure 3.1 are independent of each other. The probability distribution cannot be represented by the product of simpler terms of disjoint domains and hence we cannot take advantage of independencies. The full probability table needs $2^7 = 128$ entries to be represented exactly. Nevertheless, this table can be described exactly by 8 simple decision rules, each corresponding to a path from the root to a leaf in the tree.

We explore a novel way to exploit compact representations of high-dimensional probability tables in (approximate) probabilistic inference algorithms. Our approach is based on a (discrete) Fourier embedding of the tables, which can be interpreted as a change of basis. Crucially, tables that are dense in the canonical basis can have a sparse Fourier representation. In particular, under certain conditions, probability tables can be represented (or well approximated) using a small number of Fourier coefficients. The Fourier representation has found numerous recent applications, including modeling stochastic processes (Rogers, 2000; Abbring & Salimans, 2012), manifolds (Cohen & Welling, 2015), and permutations (Huang, Guestrin, & Guibas, 2009). Our approach is based on Fourier representation on Boolean functions, which has found tremendous success in PAC learning (O'Donnell, 2008; Mansour, 1994; Blum, Burch, & Langford, 1998; Buchman, Schmidt, Mohamed, Poole, & de Freitas, 2012), but these ideas have not been fully exploited in the fields of probabilistic inference and graphical models.

In general, a factor over n Boolean variables requires $O(2^n)$ entries to be specified, and similarly the corresponding Fourier representation is dense in general, i.e., it has $O(2^n)$ non-zero coefficients. However, a rather surprising fact which was first discovered by Linial (Linial, Mansour, & Nisan, 1993) is that factors corresponding to fairly general classes of logical forms admit a compact Fourier representation. Linial discovered that formulas in Conjunctive Normal Form (CNF) and Disjunctive Normal Form (DNF) with bounded width (the number of variables in each clause) have compact Fourier representations.

We introduce a novel approach for using approximate Fourier representations in the field of probabilistic inference. We generalize the work of Linial to the case of probability distributions (the weighted case where the entries are not necessarily 0 or 1), showing that a large class of probabilistic graphical models have compact Fourier representation. The proof extends the Hastad's Switching Lemma (Hrastad, 1987) to the weighted case. At a high level, a compact Fourier representation often means the weighted probabilistic distribution can be captured by a small set of critical decision rules. Hence, this notion is closely related to decision trees with bounded depth.

Sparse (low-degree) Fourier representations provide an entirely new way of approximating a probability distribution. We demonstrate the power of this idea by applying it to the variable elimination algorithm. Despite that it is conceptually simple, we show in Table 3.2 that the variable elimination algorithm with Fourier representation outperforms Minibucket, Belief Propagation and MCMC, and is competitive and even outperforms an award winning solver HAK on several categories of the UAI Inference Challenge.

3.1.1 Preliminaries

Inference in Graphical Models

We consider a Boolean graphical model over N Boolean variables $\{x_1, x_2, \ldots, x_N\}$. We use bold typed variables to represent a vector of variables. For example, the vector of all Boolean variables $\mathbf{x}$ is written as $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$. We also use $\mathbf{x}_S$ to represent the image of vector $\mathbf{x}$ *projected* onto a subset of variables: $\mathbf{x}_S = (x_{i_1}, x_{i_2}, \ldots, x_{i_k})^T$ where $S = \{i_1, \ldots, i_k\}$. A probabilistic graphical model is defined as:

$$Pr(\mathbf{x}) = \frac{1}{Z} f(\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{K} \psi_i(\mathbf{x}_{S_i}).$$

where each $\psi_i : \{-1, 1\}^{|S_i|} \to \mathbb{R}^+$ is called a *factor*, and is a function that depends on a subset of variables whose indices are in S_i. $Z = \sum_{\mathbf{x}} \prod_{i=1}^{K} \psi_i(\mathbf{x}_{S_i})$ is the normalization factor, and is often called the *partition function*. We will use -1 and 1 to represent false and true. We consider two key probabilistic inference tasks: the computation of the partition function Z (PR) and marginal probabilities $Pr(e) = \frac{1}{Z} \sum_{\mathbf{x} \sim e} f(\mathbf{x})$ (Marginal), in which $\mathbf{x} \sim e$ means that $\mathbf{x}$ is consistent with the evidence e.

The Variable Elimination Algorithm is an exact algorithm to compute marginals and the partition function for general graphical models. It starts with a variable ordering π. In each iteration, it eliminates one variable by multiplying all factors involving that variable, and then summing that variable out. When all variables are eliminated, the factor remaining is a singleton, whose value corresponds to the partition function. The complexity of the VE algorithm depends on the size of the largest factors generated during the elimination process, and is known to be exponential in the tree-width (Gogate & Dechter, 2004).

Detcher proposed the Mini-bucket Elimination Algorithm (Dechter, 1997), which dynamically decomposes and approximates factors (when the domain of a product exceeds a threshold) with the product of smaller factors during the elimination process. Mini-bucket can provide upper and lower bounds on the partition function. The authors of (van Rooij, Bodlaender, & Rossmanith, 2009; Smith & Gogate, 2013) develop fast operations similar to the Fast Fourier transformation, and use it to speed up the exact inference. Their approaches do not approximate the probability distribution, which is different from our approach.

Hadamard-Fourier Transformation

Hadamard-Fourier transformation has attracted a lot of attention in PAC Learning Theory. Table 3.1 provides an example where a function $\phi(x, y)$ is transformed into its Fourier representation. The transformation works by writing $\phi(x, y)$ using interpolation, then re-arranging the terms to get a canonical term. The example can be generalized, and it can be shown that any function defined on a Boolean hypercube has an equivalent Fourier representation.

Theorem 3.1.1. *(Hadamard-Fourier Transformation) Every* $f : \{-1, 1\}^n \to \mathbb{R}$ *can be* *uniquely expressed as a multilinear polynomial,*

$$f(\mathbf{x}) = \sum_{S \subseteq [n]} c_S \prod_{i \in S} x_i.$$

where each $c_S \in \mathbb{R}$. This polynomial is referred to as the Hadamard-Fourier expansion of f.

Here, $[n]$ is the power set of $\{1, \ldots, n\}$. Following standard notation, we will write $\hat{f}(S)$ to denote the coefficient c_S and $\chi_S(\mathbf{x})$ for the basis function $\prod_{i \in S} x_i$. As a special case, $\chi_\emptyset = 1$. Notice these basis functions are parity functions. We also call $\hat{f}(S)$ a degree-k coefficient of f iff $|S| = k$. In our example in Table 3.1, the coefficient for basis function xy is $\hat{\phi}(\{x, y\}) = \frac{1}{4}(\phi_1 - \phi_2 - \phi_3 + \phi_4)$, which is a degree-2 coefficient.

We re-iterate some classical results on Fourier expansion. First, as with the classical (inverse) Fast Fourier Transformation (FFT) in the continuous domain, there are similar divide-and-conquer algorithms (`FFT` and `invFFT`) which connect the table representation of f (e.g., upper left table, Table 3.1) with its Fourier representation (e.g., bottom representation, Table 3.1). Both `FFT` and `invFFT` run in time $O(n \cdot 2^n)$ for a function involving n variables. In fact, the length 2^n vector of all function values and the length 2^n vector of Fourier coefficients are connected by a 2^n-by-2^n matrix H_n, which is often called the n-th Hadamard-Fourier matrix. In addition, we have the Parseval's identity for Boolean Functions as well: $\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})^2] = \sum_S \hat{f}(S)^2$.

x	y	$\phi(x,y)$
-1	-1	ϕ_1
-1	1	ϕ_2
1	-1	ϕ_3
1	1	ϕ_4

$$\begin{aligned}\phi(x,y) =&\frac{1-x}{2}\cdot\frac{1-y}{2}\cdot\phi_1+\\ &\frac{1-x}{2}\cdot\frac{1+y}{2}\cdot\phi_2+\\ &\frac{1+x}{2}\cdot\frac{1-y}{2}\cdot\phi_3+\\ &\frac{1+x}{2}\cdot\frac{1+y}{2}\cdot\phi_4.\end{aligned}$$

$$\begin{aligned}\phi(x,y) =&\frac{1}{4}(\phi_1+\phi_2+\phi_3+\phi_4)+\frac{1}{4}(-\phi_1-\phi_2+\phi_3+\phi_4)x\\ &+\frac{1}{4}(-\phi_1+\phi_2-\phi_3+\phi_4)y+\frac{1}{4}(\phi_1-\phi_2-\phi_3+\phi_4)xy.\end{aligned}$$

Table 3.1: (Upper Left) Function $\phi : \{-1,1\}^2 \to \mathbb{R}$ is represented in a table. (Upper Right) ϕ is re-written using interpolation. (Bottom) The terms of the upper-right equation are re-arranged, which yields the Fourier expansion of function ϕ.

3.1.2 Low Degree Concentration of Fourier Coefficients

Fourier expansion replaces the table representation of a weighted function with its Fourier coefficients. For a function with n Boolean variables, the complete table representation requires 2^n entries, and so does the full Fourier expansion. Interestingly, many natural functions can be approximated well with only a few Fourier coefficients. This raises a natural question: *what type of functions can be well approximated with a compact Fourier expansion?*

We first discuss which functions can be represented *exactly* in the Fourier domain with coefficients up to degree d. To answer this question, we show a tight connection between Fourier representations with bounded degree and decision trees with bounded depth. A decision tree for a weighted function $f : \{-1,1\}^n \to \mathbb{R}$ is a tree in which each inner node is labelled with one variable, and has two out-going edges, one labelled with -1, and other one with 1. The leaf nodes are labelled with real values. When evaluating the value on an input $\mathbf{x} = x_1 x_2 \ldots x_n$, we start from the root node, and follow the corresponding outgoing edges by inspecting the value of one variable at each step, until we reach one of the leaf nodes. The value at the leaf node is the output for $f(\mathbf{x})$. The *depth* of the decision tree is defined as the longest path from the root node to one of the leaf nodes. Figure 3.1 provides a decision tree representation for a weighted Boolean function. One classical result (O'Donnell, 2008) states that if a function can be captured by a decision tree with depth d, then it can be represented with Fourier coefficients up to degree d:

Theorem 3.1.2. *Suppose $f : \{-1, 1\}^n \to \mathbb{R}$ can be represented by a decision tree of depth d, then all the coefficients whose degree are larger than d is zero in f's Fourier expansion: $\hat{f}(S) = 0$ for all S such that $|S| > d$.*

We can also provide the converse of Theorem 3.1.2:

Theorem 3.1.3. *Suppose $f : \{-1, 1\}^n \to \mathbb{R}$ can be represented by a Fourier expansion with non-zero coefficients up to degree d, then f can be represented by the sum of several decision trees, each of which has depth at most d.*

Theorem 3.1.2 and Theorem 3.1.3 provide a tight connection between the Fourier expansion and the decision trees. This is also part of the reason why the Fourier representation is a powerful tool in PAC learning. Notice that the Fourier representation complements the classical way of approximating weighted functions exploiting independencies. To see this, suppose there is a decision tree of the same structure as in Figure 3.1, but has depth d. According to Theorem 3.1.2, it can be represented exactly with Fourier coefficients up to degree d. In this specific example, the number of non-zero Fourier coefficients is $O(2^{2d})$. Nonetheless, no two variables in figure 3.1 are independent with each other. Therefore, it's not possible to decompose this factor into a product of smaller factors with disjoint domains (exploiting independencies). Notice that the full table representation of this factor has $O(2^{2^d})$ entries, because different nodes in the decision tree have different variables and there are $O(2^d)$ variables in total in this example.

If we are willing to accept an approximate representation, low degree Fourier coefficients can capture an even wider class of functions. We follow the standard notion of ϵ-concentration:

Definition 3.1.4. *The Fourier spectrum of $f : \{-1,1\}^n \to \mathbb{R}$ is ϵ-concentrated on degree up to k if and only if $\mathcal{W}_{>k}[f] = \sum_{S \subseteq [n], |S|>k} \hat{f}(S)^2 < \epsilon$.*

We say a CNF (DNF) formula has bounded width w if and only if every clause (term) of the CNF (DNF) has at most w literals. In the literatures outside of PAC Learning, this is also referred to as a CNF (DNF) with clause (term) length w. Linial (Linial et al., 1993) proved the following result:

Theorem 3.1.5 (Linial). *Suppose $f : \{-1,1\}^n \to \{-1,1\}$ is computable by a DNF (or CNF) of width w, then f's Fourier spectrum is ϵ-concentrated on degree up to $O(w \log(1/\epsilon))$.*

Linial's result demonstrates the power of Fourier representations, since bounded width CNF's (or DNF's) include a very rich class of functions. Interestingly, the bound does not depend on the number of clauses, even though the clause-variable ratio is believed to characterize the hardness of satisfiability problems.

We extend Linial's results to a class of weighted probabilistic graphical models, which are contractive with gap $1-\eta$ and have bounded width w. To our knowledge, this extension from the deterministic case to the probabilistic case is novel.

Definition 3.1.6. *Suppose $f(\mathbf{x}) : \{-1,1\}^n \to \mathbb{R}^+$ is a weighted function, we say $f(\mathbf{x})$ has bounded width w iff the number of variables in the domain of f is no more than w. We say $f(\mathbf{x})$ is contractive with gap $1-\eta$ ($0 \leq \eta < 1$) if and only if (1) for all $\mathbf{x}$, $f(\mathbf{x}) \leq 1$; (2) $\max_{\mathbf{x}} f(\mathbf{x}) = 1$; (3) if $f(\mathbf{x}_0) < 1$, then $f(\mathbf{x}_0) \leq \eta$.*

The first and second conditions are mild restrictions. For a graphical model, we can always rescale each factor properly to ensure its range is within $[0,1]$ and the largest element is 1. The approximation bound we are going to prove depends on the gap $1-\eta$. Ideally, we want η to be small. The class of contractive functions with gap $1-\eta$ still captures a wide class of interesting graphical models. For example, it captures Markov Logic Networks (Richardson & Domingos, 2006), when the weight of each clause is large. Notice that this is one of the possible necessary conditions we found success in proving the weight concentration result. In practice, because compact Fourier representation is more about the structure of the weighted distribution (captured by a series of decision trees of given depth), graphical models with large η could also have concentrated weights. The main theorem we are going to prove is as follows:

Theorem 3.1.7. *(Main) Suppose $f(\mathbf{x}) = \prod_{i=1}^{m} f_i(\mathbf{x}_i)$, in which every f_i is a contractive function with width w and gap $1-\eta$, then f's Fourier spectrum is ϵ-concentrated on degree up to $O(w \log(1/\epsilon) \log_\eta \epsilon)$ when $\eta > 0$ and $O(w \log(1/\epsilon))$ when $\eta = 0$.*

The proof of theorem 3.1.7 relies on the notion of random restriction and our own extension to the Hastad's Switching Lemma (Hrastad, 1987).

Definition 3.1.8. *Let $f(\mathbf{x}) : \{-1,1\}^n \rightarrow \mathbb{R}$ and J be subset of all the variables $x_1, \ldots, x_n$. Let $\mathbf{z}$ be an assignment to remaining variables $\overline{J} = \{-1,1\}^n \setminus J$. Define $f|_{J|\mathbf{z}} : \{-1,1\}^J \rightarrow \mathbb{R}$ to be the restricted function of f on J by setting all the remaining variables in $\overline{J}$ according to $\mathbf{z}$.*

Definition 3.1.9. *(δ-random restriction) A δ-random restriction of $f(\mathbf{x}) : \{-1,1\}^n \rightarrow \mathbb{R}$ is defined as $f|_{J|\mathbf{z}}$, when elements in J are selected randomly with probability δ, and $\mathbf{z}$ is formed by randomly setting variables in $\overline{J}$ to either -1 or 1. We also say $J|\mathbf{z}$ is a δ-random restriction set.*

With these definitions, we proved our weighted extension to the Hastad's Switching Lemma:

Lemma 3.1.10. *(Weighted Hastad's Switching Lemma) Suppose $f(\mathbf{x}) = \prod_{i=1}^{m} f_i(\mathbf{x}_i)$, in which every f_i is a contractive function with width w and gap $1-\eta$. Suppose $J|\mathbf{z}$ is a δ-random restriction set, then*

$$Pr\left(\exists \textit{ decision tree } h \textit{ with depth } t,\ ||h - f_{J|\mathbf{z}}||_\infty \leq \gamma\right) \geq 1 - \frac{1}{2}\left(\frac{\delta}{1-\delta}8uw\right)^t.$$

in which $u = \lceil \log_\eta \gamma \rceil + 1$ if $0 < \eta < 1$ or $u = 1$ if $\eta = 0$ and $||.||_\infty$ means $\max|.|$.

The formal proof of Lemma 3.1.10 is based on a clever generalization of the proof by Razborov for the unweighted case (Razborov, 1995). We refer readers to the full version of (Y. Xue, Ermon, et al., 2016) for the detailed proof.

Lemma 3.1.11. *Suppose $f(\mathbf{x}) : \{-1,1\}^n \rightarrow \mathbb{R}$ and $|f(\mathbf{x})| \leq 1$. $J|\mathbf{z}$ is a δ-random restriction set. $t \in \mathbb{N}$, $\gamma > 0$ and let*

$$\epsilon_0 = Pr\{\neg\exists \textit{ decision tree } h \textit{ with depth } t \textit{ such that } ||f|_{J|\mathbf{z}} - h||_\infty \leq \gamma\},$$

then the Fourier spectrum of f is $4\left(\epsilon_0 + (1-\epsilon_0)\gamma^2\right)$-concentrated on degree up to $2t/\delta$.

Proof. We first bound $\mathbb{E}_{J|\mathbf{z}}\left[\sum_{S\subseteq[n],|S|>t}\hat{f}|_{J|\mathbf{z}}(S)^2\right]$. With probability $1-\epsilon_0$, there is a decision tree h with depth t such that $||f|_{J|\mathbf{z}}(\mathbf{x})-h(\mathbf{x})||_\infty \le \gamma$. In this scenario,

$$\sum_{S\subseteq[n],|S|>t}\hat{f}|_{J|\mathbf{z}}(S)^2 = \sum_{S\subseteq[n],|S|>t}\left(\hat{f}|_{J|\mathbf{z}}(S)-\hat{h}(S)\right)^2. \tag{3.1}$$

This is because due to Theorem 3.1.2, $\hat{h}(S)=0$ for all S such that $|S|>t$. Because $|f|_{J|\mathbf{z}}(\mathbf{x})-h(\mathbf{x})|\le\gamma$ for all x, hence the right side of Equation 3.1 must satisfy

$$\sum_{S\subseteq[n],|S|>t}\left(\hat{f}|_{J|\mathbf{z}}(S)-\hat{h}(S)\right)^2 \le \sum_{S\subseteq[n]}\left(\hat{f}|_{J|\mathbf{z}}(S)-\hat{h}(S)\right)^2 = \mathbb{E}\left[\left(f|_{J|\mathbf{z}}(\mathbf{x})-h(\mathbf{x})\right)^2\right] \le \gamma^2. \tag{3.2}$$

The second to the last equality of Equation 3.2 is due to the Parseval's Identity. With probability ϵ_0, there are no decision trees close to $f|_{J|\mathbf{z}}$. However, because $|f|_{J|\mathbf{z}}| \le 1$, we must have $\sum_{S\subseteq[n],|S|>t}\hat{f}|_{J|\mathbf{z}}(S)^2 \le 1$. Summarizing these two points, we have:

$$\mathbb{E}_{J|\mathbf{z}}\left[\sum_{S\subseteq[n],|S|>t}\hat{f}|_{J|\mathbf{z}}(S)^2\right] \le (1-\epsilon_0)\gamma^2+\epsilon_0.$$

Using a known result $\mathbb{E}_{J|\mathbf{z}}\left[\hat{f}|_{J|\mathbf{z}}(S)^2\right] = \sum_{U\subseteq[n]} Pr\{U\cap J = S\}\cdot\hat{f}(U)^2$, we have:

$$\mathbb{E}_{J|\mathbf{z}}\left[\sum_{S\subseteq[n],|S|>t}\hat{f}|_{J|\mathbf{z}}(S)^2\right] = \sum_{S\subseteq[n],|S|>t}\mathbb{E}_{J|\mathbf{z}}\left[\hat{f}|_{J|\mathbf{z}}(S)^2\right] = \sum_{U\subseteq[n]} Pr\{|U\cap J|>t\}\cdot\hat{f}(U)^2. \tag{3.3}$$

The distribution of random variable $|U\cap J|$ is Binomial($|U|$, δ). When $|U|\ge 2t/\delta$, this variable has mean at least $2t$, using Chernoff bound, $Pr\{|U\cap J|\le t\} \le (2/e)^t < 3/4$. Therefore,

$$\begin{aligned}(1-\epsilon_0)\gamma^2+\epsilon_0 &\ge \sum_{U\subseteq[n]} Pr\{|U\cap J|>t\}\cdot\hat{f}(U)^2 \ge \sum_{U\subseteq[n],|U|\ge 2t/\delta} Pr\{|U\cap J|>t\}\cdot\hat{f}(U)^2\\ &\ge \sum_{U\subseteq[n],|U|\ge 2t/\delta}\left(1-\frac{3}{4}\right)\cdot\hat{f}(U)^2.\end{aligned}$$

We get our claim $\sum_{|U|\ge 2t/\delta}\hat{f}(U)^2 \le 4((1-\epsilon_0)\gamma^2+\epsilon_0)$. □

Now we are ready to prove Theorem 3.1.7. Firstly suppose $\eta > 0$, choose $\gamma = \sqrt{\epsilon/8}$, which ensures $4(1-\epsilon_0)\gamma^2 \leq 1/2 \cdot \epsilon$. Next choose $\delta = 1/(16uw+1)$, $t = C\log(1/\epsilon)$, which ensures

$$\epsilon_0 = \frac{1}{2}\left(\frac{\delta}{1-\delta}8uw\right)^t = \frac{1}{2}\epsilon^C.$$

Choose C large enough, such that $4 \cdot 1/2 \cdot \epsilon^C \leq 1/2 \cdot \epsilon$. Now we have $4((1-\epsilon_0)\gamma^2 + \epsilon_0) \leq \epsilon$. At the same time, $2t/\delta = C\log(1/\epsilon)(16uw+1) = O(w\log(1/\epsilon)\log_\eta \epsilon)$.[1]

3.1.3 Variable Elimination in the Fourier Domain

We have seen above that a Fourier representation can provide a useful compact representation of certain complex probability distributions. In particular, this is the case for distributions that can be captured with a relatively sparse set of Fourier coefficients. We will now show the practical impact of this new representation by using it in an inference setting. In this section, we propose an inference algorithm which works like the classic Variable Elimination (VE) Algorithm, except for passing messages represented in the Fourier domain.

The classical VE algorithm consists of two basic steps – the multiplication step and the elimination step. The multiplication step takes f and g, and returns $f \cdot g$, while the elimination step sums out one variable x_i from f by returning $\sum_{x_i} f$. Hence, the success of the VE procedure in the Fourier domain depends on efficient algorithms to carry out the aforementioned two steps. A naive approach is to transform the representation back to the value domain, carry out the two steps there, then transform it back to Fourier space. While correct, this

[1] $\eta = 0$ corresponds to the classical CNF (or DNF) case.

strategy would eliminate all the benefits of Fourier representations. Luckily, the elimination step can be carried out in the Fourier domain as follows:

Theorem 3.1.12. *Suppose f has a Fourier expansion: $f(\mathbf{x}) = \sum_{S\subseteq[n]} \hat{f}(S)\chi_S(\mathbf{x})$. Then the Fourier expansion for $f' = \sum_{x_i} f$ when x_i is summed out is: $\sum_{S\subseteq[n]} \hat{f}'(S)\chi_S(\mathbf{x})$, where $\hat{f}'(S) = 2\hat{f}(S)$ if $i \notin S$ and $\hat{f}'(S) = 0$ if $i \in S$.*

From Theorem 3.1.12, one only needs a linear scan of all the Fourier coefficients of f in order to compute the Fourier expansion for $\sum_{x_0} f$. Suppose f has m non-zero coefficients in its Fourier representation, this linear scan takes time $O(m)$.

There are several ways to implement the multiplication step. The first option is to use the school book multiplication. To multiply functions f and g, one multiplies every pair of their Fourier coefficients, and then combines similar terms. If f and g have m_f and m_g terms in their Fourier representations respectively, this operation takes time $O(m_f m_g)$. As a second option for multiplication, one can convert f and g to their value domain, multiply corresponding entries, and then convert the result back to the Fourier domain. Suppose the union of the domains of f and g has n variables (2^n Fourier terms), the conversion between the two domains dominates the complexity, which is $O(n \cdot 2^n)$. Nonetheless, when f and g are relatively dense, this method could have a better time complexity than the school book multiplication. In our implementation, we trade the complexity between the aforementioned two options, and always use the one with lower time complexity.

Because we are working on models in which exact inference is intractable, sometimes we need to truncate the Fourier representation to prevent an exponential explosion. We implement two variants for truncation. One is to keep

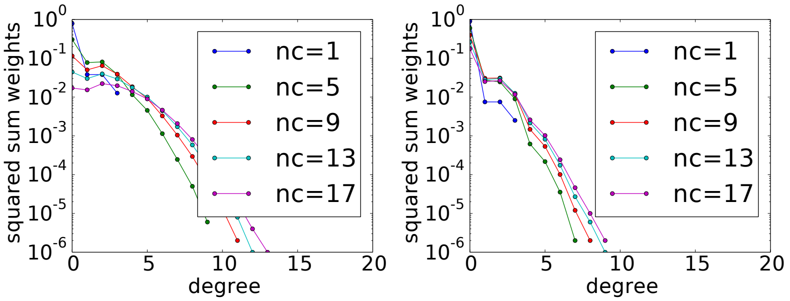


Figure 3.2: Weight concentration on low degree coefficients in the Fourier domain. Weight random 3-SAT instances, with 20 variables and nc clauses (Left) $\eta = 0.1$, (Right) $\eta = 0.6$.

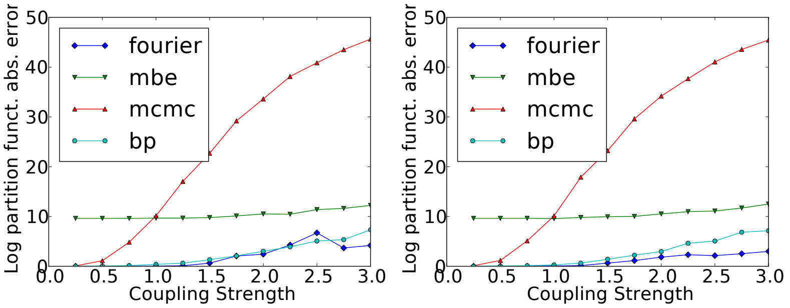


Figure 3.3: Log-partition function absolute errors for 15×15 small scale mixed weights Ising Grids. Fourier is for the VE Algorithm in the Fourier domain. mbe is for Mini-bucket Elimination. BP is for Belief Propagation. (Left) Field 0.01. (Right) Field 0.1.

low degree Fourier coefficients, which is inspired by our theoretical observations. The other one is to keep Fourier coefficients with large absolute values, which offers us a little bit extra flexibility, especially when the whole graphical model is dominated by a few key variables and we would like to go over the degree limitations occasionally. We found both variants work equally well.

3.1.4 Experiments

Weight Concentration on Low Degree Coefficients

We first validate our theoretical results on the weight concentration on low-degree coefficients in Fourier representations. We evaluate our results on random weighted 3-SAT instances with 20 variables. Small instances are chosen because we have to compute the full Fourier spectrum. The weighted 3-SAT instances is specified by a CNF and a weight η. Each factor corresponds to a clause in the CNF. When the clause is satisfied, the corresponding factor evaluates to 1, otherwise evaluates to η. For each η and the number of clauses nc, we randomly generate 100 instances. For each instance, we compute the squared sum weight at each degree: $\mathcal{W}_k[f] = \sum_{S \subseteq [n], |S|=k} \hat{f}(S)^2$. Figure 3.2 shows the median value of the squared sum weight over 100 instances for given η and nc in log scale. As seen from the figure, although the full representation involves coefficients up to degree 20 (20 variables), the weights are concentrated on low degree coefficients (up to 5), regardless of η, which is in line with the theoretical result.

Applying Fourier Representation in Variable Elimination

We integrate the Fourier representation into the variable elimination algorithm, and evaluate its performance as an approximate probabilistic inference scheme to estimate the partition function of undirected graphical models. We implemented two versions of the Fourier Variable Elimination Algorithm. One version always keeps coefficients with the largest absolute values when we truncate the representation. The other version keeps coefficients with the lowest degree. Our main comparison is against Mini-Bucket Elimination, since the two

Category	#ins	Minibucket	Fourier (max coef)	Fourier (min deg)	BP	MCMC	HAK
bn2o-30-*	18	3.91	$1.21 \cdot 10^{-2}$	$1.36 \cdot 10^{-2}$	$0.94 \cdot 10^{-2}$	0.34	$\mathbf{8.3 \cdot 10^{-4}}$
grids2/50-*	72	5.12	$\mathbf{3.67 \cdot 10^{-6}}$	$7.81 \cdot 10^{-6}$	$1.53 \cdot 10^{-2}$	–	$1.53 \cdot 10^{-2}$
grids2/75-*	103	18.34	$\mathbf{5.41 \cdot 10^{-4}}$	$6.87 \cdot 10^{-4}$	$2.94 \cdot 10^{-2}$	–	$2.94 \cdot 10^{-2}$
grids2/90-*	105	26.16	$\mathbf{2.23 \cdot 10^{-3}}$	$5.71 \cdot 10^{-3}$	$5.59 \cdot 10^{-2}$	–	$5.22 \cdot 10^{-2}$
blockmap_05*	48	$1.25 \cdot 10^{-6}$	$\mathbf{4.34 \cdot 10^{-9}}$	$\mathbf{4.34 \cdot 10^{-9}}$	0.11	–	$8.73 \cdot 10^{-9}$
students_03*	16	$2.85 \cdot 10^{-6}$	$\mathbf{1.67 \cdot 10^{-7}}$	$\mathbf{1.67 \cdot 10^{-7}}$	2.20	–	$3.17 \cdot 10^{-6}$
mastermind_03*	48	7.83	0.47	0.36	27.69	–	$\mathbf{4.35 \cdot 10^{-5}}$
mastermind_04*	32	12.30	$\mathbf{3.63 \cdot 10^{-7}}$	$\mathbf{3.63 \cdot 10^{-7}}$	20.59	–	$4.03 \cdot 10^{-5}$
mastermind_05*	16	4.06	$\mathbf{2.56 \cdot 10^{-7}}$	$\mathbf{2.56 \cdot 10^{-7}}$	22.47	–	$3.02 \cdot 10^{-5}$
mastermind_06*	16	22.34	$\mathbf{3.89 \cdot 10^{-7}}$	$\mathbf{3.89 \cdot 10^{-7}}$	17.18	–	$4.5 \cdot 10^{-5}$
mastermind_10*	16	275.82	5.63	2.98	26.32	–	$\mathbf{0.14}$

Table 3.2: The comparison of various inference algorithms on several categories in UAI 2010 Inference Challenge. The median differences in log partition function $|\log_{10} Z_{\text{approx}} - \log_{10} Z_{\text{true}}|$ averaged over benchmarks in each category are shown. Fourier VE algorithms outperform Belief Propagation, MCMC and Minibucket Algorithm. #ins is the number of instances in each category.

algorithms are both based on variable elimination, with the only difference being the way in which the messages are approximated. We obtained the source code from the author of Mini-Bucket Elimination, which includes sophisticated heuristics for splitting factors. The versions we obtained are used for Maximum A Posteriori Estimation (MAP). We augment this version to compute the partition function by replacing the maximization operators by summation operators. We also compare our VE algorithm with MCMC and Loopy Belief Propagation. We implemented the classical Ogata-Tanemura scheme (Ogata & Tanemura, 1981) with Gibbs transitions in MCMC to estimate the partition function. We use the implementation in LibDAI (Mooij, 2010) for belief propagation, with random updates, damping rate of 0.1 and the maximal number of iterations 1,000,000. Throughout the experiment, we control the number of MCMC steps, the i-bound of Minibucket and the message size of Fourier VE to make sure that the algorithms complete in reasonable time (several minutes).

We first compare on small instances for which we can compute ground truth using the state-of-the-art exact inference algorithm ACE (Darwiche & Marquis, 2002). We run on 15-by-15 Ising models with mixed coupling strengths and various field strengths. We run 20 instances for each coupling strength. For a fair comparison, we fix the size of the messages for both Fourier VE and Mini-bucket to $2^{10} = 1,024$. Under this message size VE algorithms cannot handle the instances exactly. Figure 3.3 shows the results. The performance of the two versions of the Fourier VE algorithm are almost the same, so we only show one curve. Clearly the Fourier VE Algorithm outperforms the MCMC and the Mini-bucket Elimination. It also outperforms Belief Propagation when the field strength is relatively strong.

In addition, we compare our inference algorithms on large benchmarks from the UAI 2010 Approximate Inference Challenge (*UAI 2010 Approximate Inference Challenge*, n.d.). Because we need the ground truth to compare with, we only consider benchmarks that can be solved by ACE (Darwiche & Marquis, 2002) in 2 hours time, and 8GB of memory. The second column of Table 3.2 shows the number of instances that ACE completes with the exact answer. The 3rd to the 7th column of Table 3.2 shows the result for several inference algorithms, including the Minibucket algorithm with i-bound of 20, two versions of the Fourier Variable Elimination algorithms, belief propagation and MCMC. To be fair with Minibucket, we set the message size for Fourier VE to be 1,048,576 (2^{20}). Because the complexity of the multiplication step in Fourier VE is quadratic in the number of coefficients, we further shrink the message size to 1,024 (2^{10}) during multiplication. We allow 1,000,000 steps for burn in and another 1,000,000 steps for sampling in the MCMC approach. The same with the inference challenge, we compare inference algorithms on the difference in the log partition function $|\log Z_{\text{approx}} - \log Z_{\text{true}}|$. The table reports the median differences, which are averaged over all benchmarks in each category. If one algorithm fails to complete on one instance, we count the difference in partition function as $+\infty$, so it is counted as the worst case when computing the median. For MCMC, "–" means that the Ogata-Tanemura scheme did not find a belief state with substantial probability mass, so the result is way off when taking the logarithm. The results in Table 3.2 show that Fourier Variable Elimination algorithms outperform MCMC, BP and Minibucket on many categories in the Inference challenge. In particular, Fourier VE works well on grid and structural instances. We also listed the performance of a Double-loop Generalized Belief Propagation (Heskes, Albers, & Kappen, 2003) in the last column of Table 3.2. This implementation won one category in the Inference challenge, and contains various improvements besides the techniques presented in the paper. We used the parameter settings for high precision in the Inference challenge for HAK. As we can see, Fourier VE matches or outperforms this implementation in some categories. Unlike fully optimized HAK, Fourier VE is a simple variable elimination algorithm, which involves passing messages only once. Indeed, the median time for Fourier VE to complete on bn2o instances is about 40 seconds,

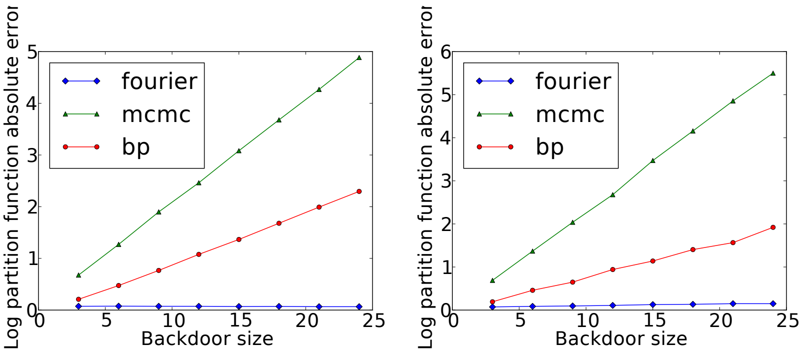


Figure 3.4: Log-partition function absolute errors for Weighted Models with Backdoor Structure. (Left) Independent backdoors. (Right) Linked backdoors.

while HAK takes 1800 seconds. We are researching on incorporating the Fourier representation into message passing algorithms.

Next we evaluate their performance on a synthetically generated benchmark beyond the capability of exact inference algorithms. For one instance of this benchmark, we randomly generate factors of size 3 with low coupling weights. We then add a backdoor structure to each instance, by enforcing coupling factors of size 3 in which the 3 variables of the factor must take the same value. For these instances, we can compute the expected value of the partition function and compare it with the output of the algorithms. We report the results on Figure 3.4. Here the experimental setup for each inference algorithm is kept the same as the previous algorithm. The Mini-bucket approach is not reported, as it performs very poorly on these instances. The performance of the two implementations of Fourier VE are again similar, so they are combined into one curve. These results show that the Fourier approach outperforms both MCMC and Belief Propagation, and suggest that it can perform arbitrarily better than both approaches as the size of the backdoor increases.

Finally, we compare different inference algorithms on a machine learning application. Here we learn a grid Ising model from data. The computation of the partition function is beyond any exact inference methods. Hence in order to compare the performance of different inference algorithms, we have to control the training data that are fit into the Ising Model, to be able to predict what the learned model looks like. To generate training pictures, we start with a template with nine boxes (shown in Figure 3.5(a)). The training pictures are of size 25×25, so the partition function cannot be computed exactly by variable elimination algorithms with message size $2^{20} = 1,048,576$. Each of the nine boxes in the template will have a 50% opportunity to appear in a training picture, and the occurrences of the nine boxes are independent of each other. We further blur the training images with 5% white noise. Figures 3.5(b) and 3.5(c) show two examples of the generated training images. We then use these training images to learn a grid Ising Model:

$$Pr(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i \in V} a_i x_i + \sum_{(i,j) \in E} b_{i,j} x_i x_j\right),$$

where V, E are the node and edge set of a grid, respectively. We train the model using contrastive divergence (G. E. Hinton, 2002), with $k = 15$ steps of blocked Gibbs updates, on $20,000$ such training images. (As we will see, *vanilla* Gibbs sampling, which updates one pixel at a time, does not work well on this problem.) We further encourage a sparse model by using a L1 regularizer. Once the model is learned, we use inference algorithms to compute the marginal probability of each pixel. Figure 3.5(d,e,f,g) show the marginals computed for the Fourier VE, MCMC, Minibucket Elimination, and the Mean Field on the learned model (white means the probability is close to 1, black means close to 0). Both the Minibucket and the Fourier VE keep a message size of $2^{20} = 1,048,576$, so they cannot compute the marginals exactly. Fourier VE keeps coefficients

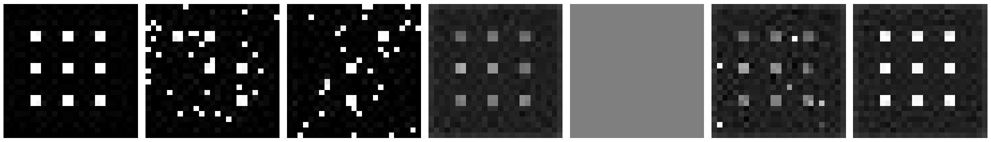


Figure 3.5: Comparison of several inference algorithms on computing the marginal probabilities of an Ising model learned from synthetic data. From left to right (a to g): (a) The template to generate training images and (b,c) two example images in the training set. (d,e,f,g) The marginal probabilities obtained via four inference algorithms. (d) Fourier, (e) MCMC, (f) mbe, (g) mean field. Only the Fourier algorithm captures the fact that the 9 boxes are presented half of the time independently in the training data.

with largest absolute value during multiplication. For pixels outside of the nine boxes, in most circumstances they are black in the training images. Therefore, their marginals in the learned model should be close to 0. For pixels within the nine boxes, half of the time they are white in the training images. Hence, the marginal probabilities of these pixels in the learned model should be roughly 0.5. We validated the two aforementioned empirical observations on images with small size which we can compute the marginals exactly. As we can see, only the Fourier Variable Elimination Algorithm is able to predict a marginal close to 0.5 on these pixels. The performance of the MCMC algorithm (a Gibbs sampler, updating one pixel at a time) is poor. The Minibucket Algorithm has noise on some pixels. The marginals of the nine boxes predicted by mean field are close to 1, a clearly wrong answer.

3.1.5 Discussion

We explore a novel way to exploit compact representations of high-dimensional probability distributions in approximate probabilistic inference. Our approach is based on discrete Fourier embedding of weighted Boolean Functions, complementing the classical method of exploiting conditional independence between the variables. We show that a large class of weighted probabilistic graphical models have a compact Fourier representation. This theoretical result opens up a novel way of approximating probability distributions. We demonstrate the significance of this approach by applying it to the variable elimination algorithm, obtaining very encouraging results.

3.2 Dimensionality Reduction using Parallel Problem Decomposition

Exploiting parallelism and multi-core architectures is a natural way to speed up computations in many domains. Recently, there has been great success in parallel computation in fields such as scientific computing and information retrieval (Dean & Ghemawat, 2008; C. Chu et al., 2007).

Parallelism has also been taken into account as a promising way to solve hard combinatorial problems. However, it remains challenging to exploit parallelism to speed up combinatorial search due to the intricate non-local nature of the interactions between variables in hard problems (Hamadi & Wintersteiger, 2013). One class of approaches in this domain is divide-and-conquer, which dynamically splits the search space into sub-spaces, and allocates each sub-space to a parallel node (Chrabakh & Wolski, 2003; G. Chu, Stuckey, & Harwood, 2008; Rao & Kumar, 1993; Regin, Rezgui, & Malapert, 2013; Moisan, Gaudreault, & Quimper, 2013; Fischetti, Monaci, & Salvagnin, 2014). A key challenge in this approach is that the solution time for subproblems can vary by several orders of magnitude and is highly unpredictable. Frequent load re-balancing is required to keep all processors busy, but the load re-balancing process can result in a substantial overhead cost. Another class of approaches harnesses portfolio strategies, which runs a portfolio of solvers (of different type or with different randomization) in parallel, and terminates as soon as one of the algorithms completes. (Xu, Hutter, Hoos, & Leyton-Brown, 2008; Leyton-Brown, Nudelman, Andrew, Mcfadden, & Shoham, 2003; Malitsky, Sabharwal, Samulowitz, & Sellmann, 2011; Kadioglu, Malitsky, Sabharwal, Samulowitz, & Sellmann, 2011;

Hamadi & Sais, 2009; Biere, 2010; Kottler & Kaufmann, 2011; Schubert, Lewis, & Becker, 2010; O'Mahony, Hebrard, Holland, & Nugent, 2008). Parallel portfolio approaches can be highly effective. They do require however the use of a collection of effective solvers that each excel at different types of problem instances. In certain areas, such as SAT/SMT solving, we have such collections of solvers but for other combinatorial tasks, we do not have many different solvers available.

We exploit parallelism to boost dimensionality reduction with combinatorial constraints. Our framework complements the two parallel approaches discussed before. In our approach parallelism is used as a preprocessing step to identify a promising portion of the search space to be explored by a complete sequential solver. In our scheme, a set of parallel processes are first deployed to solve a series of related subproblems. Next, the solutions to these subproblems are aggregated to obtain an initial guess for a candidate solution to the original problem. The aggregation is based on a key empirical observation that solutions to the subproblems, when properly aggregated, provide information about solutions for the original problem. Lastly, a global sequential solver searches for a solution in an iterative deepening manner, starting from the promising portion of the search space identified by the previous aggregation step. At a high level, the initial guess obtained by aggregating solutions to subproblems provides the so-called backdoor information to the sequential solver, by forcing it to start from the most promising portion of the search space. A backdoor set is a set of variables, such that once their values are set correctly, the remaining problem can be solved in polynomial time (Williams, Gomes, & Selman, 2003; Dilkina, Gomes, Malitsky, Sabharwal, & Sellmann, 2009; Hamadi, Marques-Silva, & Wintersteiger, 2011).

We empirically show that a global solver, when initialized with proper information obtained by solving the sub-problems, can solve a set of instances in seconds, while it takes for the same solver hours to days to find the solution without initialization. The strategy also outperforms state-of-the-art incomplete solvers in terms of solution quality.

Our parallel approach compliments our previous work (Le Bras et al., 2014), which integrates subtle human insights with combinatorial solvers for dimensionality reduction. The parallel scheme presented here can be seen as to replace crowdsourced human inputs with fully automatic parallel processes in searching for backdoor information to initialize a combinatorial solver.

We first apply the parallel scheme to an NP-complete problem called the Set Basis Problem, in which we are given a collection of subsets $\mathcal{C}$ of a finite set U. The task is to find another, hopefully smaller, collection of subsets of U, called a "set basis", such that each subset in $\mathcal{C}$ can be represented exactly by a union of sets from the set basis. Intuitively, the set basis provides a compact representation of the original collection of sets. The set basis problem occurs in a range of applications, most prominently in machine learning, e.g., used as a special type of matrix factorization technique (Miettinen, Mielikainen, Gionis, Das, & Mannila, 2008). It also has applications in system security and protection, where it is referred to as the role mining problem in access control (Vaidya, Atluri, & Guo, 2007). It also has applications in secure broadcasting (Shu, Lee, & Yannakakis, 2006) and computational biology (Nau, Markowsky, Woodbury, & Amos, 1978).

While having many natural applications, our work is motivated by a novel application in the field of computational sustainability (Gomes, Winter 2009), concerning the discovery of new materials for renewable energy sources such as improved fuel cell catalysts (Le Bras et al., 2011). In this domain, the set basis problem is used to find a succinct explanation of a large set of measurements (X-ray diffraction patterns) that are represented in a discrete way as sets. Mathematically, this corresponds to a generalized version of the set basis problem with extra constraints. Our parallel solver can be applied to this generalized version as well, and we demonstrate significant speedups on a set of challenging benchmarks. Our work opens up a novel angle for using parallelism to solve a set of dimensionality reduction problems with complex physical constraints.

3.2.1 Set Basis Problem

Sets will be denoted by uppercase letters, while members of a set will be denoted by lowercase letters. A collection of sets will be denoted using calligraphic letters.

The classic Set Basis Problem is defined as follows:

- **Given:** a collection $\mathcal{C}$ of subsets of a finite universe U, $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ and a positive integer K;
- **Find:** a collection $\mathcal{B} = \{B_1, \ldots, B_K\}$ where each B_i is a subset of U, and for each $C_i \in \mathcal{C}$, there exists a sub-collection $\mathcal{B}_i \subseteq \mathcal{B}$, such that $C_i = \cup_{B \in \mathcal{B}_i} B$. In this case, we say $\mathcal{B}_i$ *covers* C_i, and we say $\mathcal{C}$ is *collectively covered* by $\mathcal{B}$. Following common notations, $\mathcal{B}$ is referred to as a *basis*, and we call each $B_j \in \mathcal{B}$ a *basis set*. If $B_j \in \mathcal{B}_i$ and $\mathcal{B}_i$ covers C_i, we call B_j a *contributor* of C_i, and call C_i a *sponsor* of B_j. $C_1, C_2, \ldots, C_m$ are referred to as *original sets*.

C_0	$\{x_0, x_1, x_2, x_3\}$	$B_0 \cup B_2$
C_1	$\{x_0, x_2, x_3\}$	B_0
C_2	$\{x_0, x_1, x_4\}$	$B_2 \cup B_4$
C_3	$\{x_2, x_3, x_4\}$	$B_1 \cup B_4$
C_4	$\{x_0, x_1, x_3\}$	$B_2 \cup B_3$
C_5	$\{x_3, x_4\}$	$B_3 \cup B_4$
C_6	$\{x_2, x_3\}$	B_1

B_0	$\{x_0, x_2, x_3\}$	$C_0 \cap C_1$
B_1	$\{x_2, x_3\}$	$C_3 \cap C_6$
B_2	$\{x_0, x_1\}$	$C_0 \cap C_2 \cap C_4$
B_3	$\{x_3\}$	$C_4 \cap C_5$
B_4	$\{x_4\}$	$C_2 \cap C_3 \cap C_5$

Table 3.3: (An example of set basis problem) $C_0, \ldots, C_6$ are the original sets. A basis of size 5 that cover these sets is given by $B_0, \ldots, B_4$. The rightmost column at the top shows how each original set can be obtained from the union of one or more basis sets. The given cover is minimum (i.e., containing a minimum number of basis sets). The rightmost column at the bottom shows the duality property: each basis set can be written as an intersection of several original sets.

Intuitively, similar to the basis vectors in linear algebra, which provides a succinct representation of a linear space, a set basis with smallest cardinality K plays the role of a compact representation of a collection of sets. The Set Basis Problem is shown to be NP-hard in (Stockmeyer, 1975). We use $\mathcal{I}(\mathcal{C})$ to denote an instance of the set basis problem which finds the basis for $\mathcal{C}$. A simple instance and its solution is reported in Table 3.3.

Most algorithms used in solving set basis problems are incomplete algorithms. These algorithms are based on heuristics that work well in certain domains, but often fail at covering sets exactly. For a survey, see Molloy et al (Molloy et al., 2009). The authors of (Ene et al., 2008) implement the only complete solver we are aware of. The idea is to translate the set basis problem as a graph coloring problem, and then use existing graph coloring solvers. They also develop a useful prepossessing technique, which can significantly reduce the problem complexity.

The Set Basis Problem has a useful dual property, which has been implicitly used by previous researchers (Vaidya, Atluri, & Warner, 2006; Ene et al., 2008). We formalize the idea by introducing Theorem 3.2.2.

Definition 3.2.1. *(Closure) For a collection of sets $\mathcal{C}$, define the closure of $\mathcal{C}$, denoted as $\overline{\mathcal{C}}$, which includes the collection of all possible intersections of sets in $\mathcal{C}$:*

- $\forall C_i \in \mathcal{C}$, $C_i \in \overline{\mathcal{C}}$.
- *For $A \in \overline{\mathcal{C}}$ and $B \in \overline{\mathcal{C}}$, $A \cap B \in \overline{\mathcal{C}}$.*

Theorem 3.2.2. *For original sets $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$, suppose $\{B_1, \ldots, B_K\}$ is a basis that collectively covers $\mathcal{C}$. Define $\mathcal{C}_i = \{C_j \in \mathcal{C} | B_i \subseteq C_j\}$. Then $B'_i = \cap_{C \in \mathcal{C}_i} C$ ($i = 1 \ldots K$) collectively covers $\mathcal{C}$ as well. Note for every B'_i ($i = 1 \ldots K$), $B'_i \in \overline{\mathcal{C}}$.*

One can check Theorem 3.2.2 by examining the example in Table 3.3. The full proof is available in (Y. Xue et al., 2015). From the theorem, any set basis problem has a solution of minimum cardinality, where each basis set is in $\overline{\mathcal{C}}$. Therefore, it is sufficient to only search for basis within the closure $\overline{\mathcal{C}}$. Hence throughout this paper, we assume all basis sets are within its closure for any solutions to set basis problems. Theorem 3.2.2 also implies that *each basis set $B_i \in \overline{C}$ is an intersection of all its sponsor sets.* One can observe this fact in Table 3.3. It motivates our dual approach to solve the set basis problem, in which we search for possible sponsors for each basis set.

3.2.2 Motivating Application in Materials Discovery

Our parallel scheme is motivated by a central task in combinatorial materials discovery, namely the problem of identifying the crystalline phases of inorganic

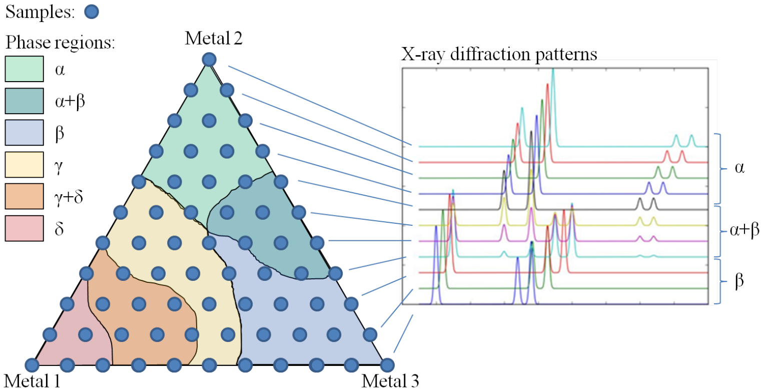


Figure 3.6: Demonstration of a phase identification problem. (Left) A set of of sample points (blue circles) on a silicon wafer (triangle). Colored areas show the regions where phase (basis pattern) α, β, γ, δ exist. (Right) the X-ray diffraction pattern (XRD) for sample points on the right edge of the triangle. The XRD patterns transform from single phase region α to composite phase region $\alpha+\beta$ to single phase region β, with small shiftings along neighboring sample locations.

compounds based on an analysis of high-intensity X-ray patterns. Many industrial and technological innovations, from steam engines to silicon circuits and solar panels, have been enabled through the discovery of advanced materials. Accelerating the pace of the discovery cycle of new materials is essential to fostering innovative advances, improving human welfare and achieving sustainable development.

In order to effectively assess many candidate materials, materials scientists have developed high-throughput deposition techniques capable of generating large composition-spread libraries (Takeuchi, Dover, & Koinuma, 2002). Once synthesized, the promising libraries are characterized through X-ray diffraction and fluorescence (Gregoire, Dale, Kazimirov, DiSalvo, & van Dover, 2009). The goal of this characterization is to map the composition and the structure of each library. This is called the *phase identification problem* and is the motivating application of our work. This problem aims to provide composition and structure maps that can be correlated with interesting physical properties within an inorganic library, such as conductivity, catalytic properties or light absorbency. Yet, solving this problem remains a laborious time-consuming manual task that relies on experts in materials science. The contribution of this work is to propose a principled approach to solving this problem, accelerating the pace of analysis of the composition libraries and alleviating the need for human experts.

In combinatorial materials discovery, a *thin film* is obtained by depositing three metals onto a silicon wafer using guns pointing at three locations. As metals are sputtered on the silicon wafer, different locations have different combinations of the metals, due to their distances from the gun points. As a result, various crystal structures are formed across locations. Researchers then analyze the X-ray diffraction patterns (XRD) at a selected set of sample points. The XRD pattern at one sample point reflects the crystal structure of the underlying material, and is a mixture of one or more basis patterns, each of which characterizes one crystal structure. The overall goal of the phase identification problem is to explain all the XRD patterns using a small number of basis patterns.

The phase identification problem can be formulated as an extended version of the set basis problem. We begin by introducing some terminologies. Similar to (Ermon et al., 2012), we use discrete representations of the XRD signals, where we characterize each XRD pattern with the locations of its *peaks*. In this model, we define a *peak* q as a set of (sample point, location) pairs: $q = \{(s_i, l_i) | i = 1, \ldots, n_q\}$, where $\{s_i | i = 1, \ldots, n_q\}$ is a set of sample points where peak q is present, and l_i is the location of peak q at sample point s_i, respectively. We use the term *phase* to refer to a basis XRD pattern. Precisely, a *phase* comprises set of peaks that occur in the same set of sample points. We use the term *partial phase* to refer to a subset of the peaks and/or a subset of the sample points of a phase. We use lower-case letters p, q, r to represent peaks, and use upper-case letters P, Q, R to represent phases. Given these definitions, the *Phase Identification Problem* is:

Given A set of X-ray diffraction patterns representing different material compositions and a set of detected peaks for each pattern; and K, the expected number of phases.

Find A set of K phases, characterized as a set of peaks and the sample points in which they are involved.

Subject to Physical constraints that govern the underlying crystallographic process. We use all the constraints in (Ermon et al., 2012). For example, one physical constraint is that a phase must span a continuous region in the silicon wafer.

Figure 3.6 shows an illustrative example. In this example, there are 4 peaks for phase α, and 3 peaks for phase β. Peaks in phase α exist in all sample points in the green region, and peaks in phase β exist in purple region. They co-exist in several sample points in the mid-right region of the triangle.

There is an analogy between the Phase Identification Problem and the classical Set Basis Problem. In the Set Basis Problem, each original set is the union of some basis sets. In the Phase Identification Problem, the XRD pattern at a given sample point is a mixture of several phases. Here, the phase is analogous to the basis set, and the XRD pattern at a given sample point is analogous to the original set.

3.2.3 Parallel Scheme

Set Basis Problem

We first illustrate of the idea of our parallel scheme in the context of the set basis problem. The main intuition of our parallel scheme comes from an empirical observation on the structure of the solutions of the benchmark problems we considered. For each benchmark, we solve a series of related simplified subproblems, where we restrict ourselves to finding basis for *a subset of the original collection of sets $\mathcal{C}$*. Interestingly, the solutions found by solving these subproblems are connected to the basis in the global solution. Although strictly speaking, the basis found for one sub-problem can only be expected to be a solution for that particular sub-problem, we observe empirically that *almost all basis sets from subproblems are supersets of one or more basis sets for the original, global problem.* One intuitive explanation is as follows: Recall that from Theorem 3.2.2 each basis set can be obtained as the intersection of its sponsors. This fact applies both to the original global problem and its relaxed versions (subproblems). Since there are fewer sets to be covered in the subproblems, basis sets for the subproblems are likely to have fewer sponsors, compared to the ones for the global problem. When we take the intersection of fewer sets, we get a larger intersection. Hence we observe that it is often the case that a basis set for a subproblem is a superset of a basis set for the global problem.

Now suppose two subproblem basis sets A and B are both supersets of one basis set C in the global solution. If we intersect A with B, then the elements of C will remain in the intersection, but other elements from A or B will likely be removed. In practice, *we can often obtain a basis set in the global solution by*

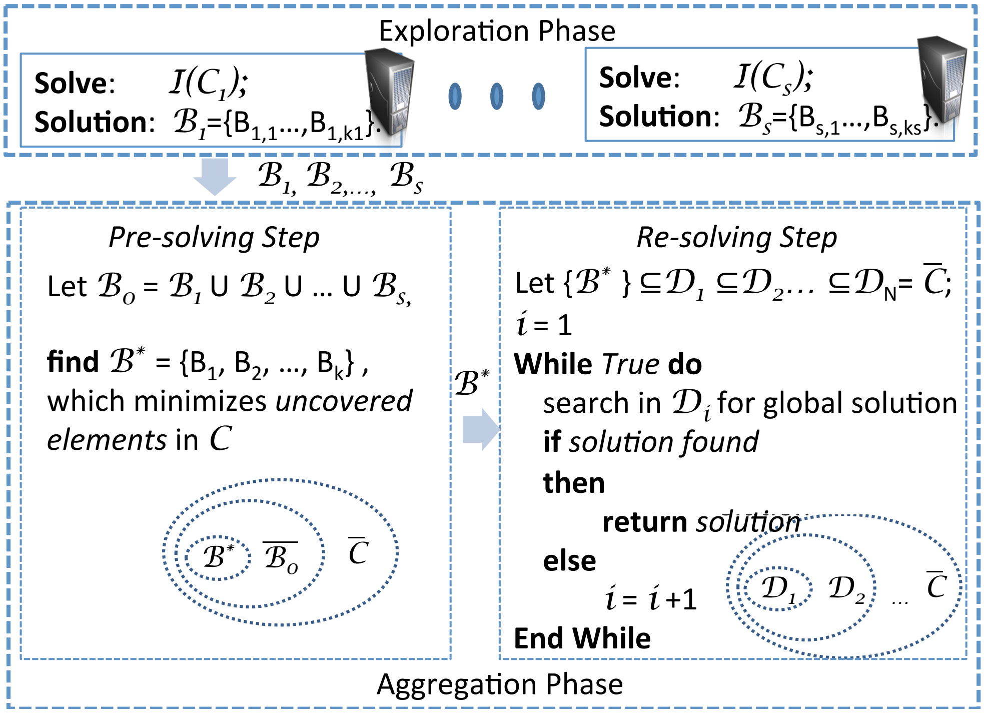

Exploration Phase
Solve: I(C_1);
Solution: ℬ_1={B_{1,1}…,B_{1,k1}}.
Solve: I(C_s);
Solution: ℬ_s={B_{s,1}…,B_{s,ks}}.
ℬ_1, ℬ_2,…, ℬ_s
Pre-solving Step
Let ℬ_0 = ℬ_1 U ℬ_2 U … U ℬ_s,
find ℬ* = {B_1, B_2, …, B_k} , which minimizes uncovered elements in C
ℬ* ℬ_0 C
ℬ*
Re-solving Step
Let {ℬ* } ⊆𝒟_1 ⊆𝒟_2… ⊆𝒟_N= C;
i = 1
While True do
search in 𝒟_i for global solution
if solution found
then
return solution
else
i = i +1
End While
𝒟_1 𝒟_2 … C
Aggregation Phase

Figure 3.7: A diagram showing the parallel scheme.

intersecting only a few basis sets from the solutions to subproblems.

Let us walk through the example in Table 3.3. First consider the subproblem consisting of the first 5 original sets, $C_0 ... C_4$. It can be shown that a minimum set basis is $B_{1,1} = \{x_0, x_1\}$, $B_{1,2} = \{x_0, x_3\}$, $B_{1,3} = \{x_2, x_3\}$, $B_{1,4} = \{x_4\}$. As another subproblem we consider the collection of all original sets except for C_0 and C_2. We obtain a minimum basis $B_{2,1} = \{x_0, x_3\}$, $B_{2,2} = \{x_2, x_3\}$, $B_{2,3} = \{x_3, x_4\}$, $B_{2,4} = \{x_0, x_1, x_3\}$. We see that each basis set of these two subproblems contains at least one of the basis sets of the original, full set basis problem. For example, $B_2 = \{x_0, x_1\} \subseteq B_{1,1}$ and $B_2 \subseteq B_{2,4}$. Moreover, one can obtain all basis sets except B_0 for the original problem by intersecting these basis sets. For example, $B_3 = \{x_3\} = B_{1,2} \cap B_{2,2}$.

Given this observation, we design a parallel scheme that works in two phases – an exploration phase, followed by an aggregation phase. The whole process is shown in Figure 3.7, and the two phases are detailed in subsequent subsections.

Exploration Phase: we use a set of parallel processes. Each one solves a sub-problem obtained by restricting the global problem to finding the *minimum basis* for a subset of the original collection of sets $\mathcal{C}$.

Aggregation Phase: we first identify an initial solution candidate by looking at all the possible intersections among basis sets found by solving sub-problems in the exploration phase. We then use this candidate to initialize a complete solver, which expands the search in an iterative deepening way to achieve completeness, iteratively adding portions of the search space that are "close" to the initial candidate.

Exploration Phase The exploration phase utilizes parallel processes to solve a series of sub-problems. Recall the global problem is to find a basis of size K for the collection $\mathcal{C}$. Let $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s\}$ be a decomposition of $\mathcal{C}$, which satisfies $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \ldots \cup \mathcal{C}_s$. The sub-problem $\mathcal{I}(\mathcal{C}_i)$ restricted on $\mathcal{C}_i$ is defined as:

- **Given:** $\mathcal{C}_i \subseteq \mathcal{C}$;
- **Find:** a basis $\mathcal{B}_i = \{B_{i,1}, B_{i,2}, \ldots, B_{i,k_i}\}$ with smallest cardinality, such that every set $C' \in \mathcal{C}_i$ is covered by the union of a sub-collection of $\mathcal{B}_i$.

The sub-problem is similar to the global problem, however, with one key difference: we are solving an *optimization problem* where we look for a minimum basis, as opposed to the global problem, which is the decision version of the Set Basis Problem. In practice, the optimization is done by repeatedly solving the decision problem, with increasing values of K. We observe empirically that the optimization is crucial for us to get meaningful basis sets to be used in later aggregation steps. If we do not enforce the minimum cardinality constraint, the problem becomes under-constrained and there could be redundant basis sets found in this phase, which have no connections with the ones in the global solution.

Sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s$ need not be mutually exclusive in the decomposition. We group similar sets into one subproblem in our algorithm, so the resulting subproblem will have a small basis. To obtain collection $\mathcal{C}_i$ for the i-th subproblem, we start from an initial collection of a singleton $\mathcal{C}_i = \{C_{i_1}\}$, where C_{i_1} is a randomly picked element from $\mathcal{C}$. We then add $T-1$ sets most similar to C_{i_1}, using the Jaccard similarity coefficient $\frac{|A \cap B|}{|A \cup B|}$. This results in a collection of T sets which look similar. Notice that this is our method to find a collection of similar sets. We expect other approach can work equally well.

Aggregation Phase In the aggregation phase, a centralized process searches for the exact solution, starting from basis sets that are "close" to a candidate solution selected from the closure of basis sets found by solving sub-problems, and then expands its search space iteratively to achieve completeness.

To obtain a good initial candidate solution, we begin with a pre-solving step, in which we restrict ourselves to find a good global solution only within the closure of basis sets found by solving sub-problems. This is of course an incomplete procedure, because the solution might lie outside the closure. However, due to the empirical connections between the basis sets found by parallel subproblem solving and the ones in the final solution, we often find the global solution at this step.

If we cannot find a solution in the pre-solving step, the algorithm continues with a re-solving step, in which an iterative deepening process is applied. It starts with the best K basis sets found in the pre-solving step[2], and iteratively expands the search space until it finds a global solution. The two steps are detailed as follows.

Pre-solving Step Suppose $\mathcal{B}_i$ is the basis found for sub-problem $\mathcal{I}(\mathcal{C}_i)$, let $\mathcal{B}_0 = \cup_{i=1}^{s} \mathcal{B}_i$ and $\overline{\mathcal{B}_0}$ be the closure of $\mathcal{B}_0$. The algorithm solves the following problem in the pre-solving step:

- **Given:** $\mathcal{B}_0$;
- **Find:** Basis $\mathcal{B}^* = \{B_1^*, \ldots, B_K^*\}$ from $\overline{\mathcal{B}_0}$, such that $B_1^*, \ldots, B_K^*$ minimizes the total number of uncovered elements and falsely covered elements in $\mathcal{C}$[3].

In practice $\overline{\mathcal{B}_0}$ is still a huge space, so this optimization problem is hard to solve. We thus apply an incomplete algorithm, which only samples a small subset $\mathcal{U} \subseteq \overline{\mathcal{B}_0}$ and then select the best K basis sets from $\mathcal{U}$. It does not affect the later re-solving step, since it can start the iterative deepening process from any $\mathcal{B}^*$, whether optimal in $\overline{\mathcal{B}_0}$ or not.

The incomplete algorithm first forms $\mathcal{U}$ by conducting multiple random walks in the space of $\overline{\mathcal{B}_0}$. Each random walk starts with a random basis set $B \in \mathcal{B}_0$, and randomly intersects it with other basis sets in $\mathcal{B}_0$ to obtain a new member in $\overline{\mathcal{B}_0}$. All these sets are collected to form $\mathcal{U}$. With probability p, the algorithm chooses to intersect with the basis which maximizes the cardinality of

[2] Best in terms of the coverage of the initial set collection.

[3] An *uncovered* element of set C_j is one element contained in C_j, but is not covered by any basis set that are contributors to C_j. A *falsely covered* element of set C_j is one element that is in one basis set that is a contributor to set C_j, but is not contained in C_j.

the intersection. With probability $(1-p)$, the algorithm intersects with a random set. In our experiment, p is set to 0.95, and we repeat this random walk several times with different initial sets to make $\mathcal{U}$ large enough. Next the algorithm selects the optimal basis of size K from $\mathcal{U}$ which maximizes the coverage of the initial set collection, using a Mixed Integer Programming (MIP) formulation.

Re-solving Step The final step is the re-solving step. It takes as input the basis $\mathcal{B}^* = \{B_1^*, B_2^*, \ldots, B_K^*\}$ from the pre-solving step, and searches for a complete solution to $\mathcal{I}(C)$ in an iterative deepening manner. The algorithm starts from a highly restricted space $\mathcal{D}_1$, which is a small space close to $\mathcal{B}^*$. If the algorithm can find a global solution in $\mathcal{D}_1$, then it terminates and returns the solution. Otherwise, it expands its search space to $\mathcal{D}_2$, and searches again in this expanded space, and so on. At the last step, searching in $\mathcal{D}_n$ is equivalent to searching in the original unconstrained space $\overline{\mathcal{C}}$, which is equivalent to solving the global set-basis problem without initialization at all. However, this situation is rarely seen in our experiments.

In practice, $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are specified by adding extra constraints to the original MIP formulation for the global problem, then iteratively removing them. $\mathcal{D}_n$ corresponds to the case where all extra constraints are removed.

The actual design of $\mathcal{D}_1, \ldots, \mathcal{D}_n$ relies on the MIP formulation. In our MIP formulation, there are indicator variables $y_{i,k}$ ($1 \leq i \leq n$ and $1 \leq k \leq K$), where $y_{i,k} = 1$ if and only if the i-th element is contained in the k-th basis set B_k. We also have indicator variables $z_{k,j}$, where $z_{k,j}$ is one if and only if the basis set B_k is a contributor of the original set C_j (or equivalently, C_j is a sponsor set for B_k).

Because we empirically observe that $B_1^*, B_2^*, \ldots, B_K^*$ are often super-sets of the basis sets in the exact solution, we construct the constrained space $\mathcal{D}_1, \ldots, \mathcal{D}_n$ by enforcing the sponsor sets of certain basis sets. Notice that this is a straightforward step in the MIP formulation, since we only need to fix the corresponding indicator variables $z_{k,j}$ to 1 to enforce C_j as a sponsor set for B_k. The hope is that these clamped variables will include a subset of backdoor variables for the original search problem (Williams et al., 2003; Dilkina et al., 2009; Hamadi et al., 2011). The runtime of the sequential solver is dramatically reduced when the aggregation phase is successful in identifying a promising portion of the search space.

As pointed out by Theorem 3.2.2, we can represent $B_1^*, B_2^*, \ldots, B_K^*$ in terms of their sponsors:

$$\begin{aligned} B_1^* &= C_{11} \cap C_{12} \cap \ldots \cap C_{1s_1} \\ B_2^* &= C_{21} \cap C_{22} \cap \ldots \cap C_{2s_2} \\ &\ldots \\ B_K^* &= C_{K1} \cap C_{K2} \cap \ldots \cap C_{Ks_K} \end{aligned}$$

in which $C_{11}, C_{12}, \ldots, C_{21}, \ldots, C_{Ks_K}$ are all original sets in collection $\mathcal{C}$. For the first restricted search space $\mathcal{D}_1$, we enforce the constraint that the sponsors for the i-th basis set B_i must contain all the sponsors of B_i^* for all $i \in \{1, \ldots, K\}$. Notice this implies $B_i \subseteq B_i^*$.

In later steps, we gradually relax these extra constraints, by freeing some of the indicator variables $z_{k,j}$'s which were clamped to 1 in previous steps. $\mathcal{D}_n$ denotes the search space when all these constraints are removed, which is equivalent to searching the entire space. The last thing is to decide the order used to remove these sponsor constraints. Intuitively, if one particular set is discovered many times as a sponsor set in the solutions to subproblems, then it should have a high chance to be the sponsor set in the global solution, because it fits in the solutions to many subproblems. Given this intuition, we associate each sponsor set with a *confidence score*, and define n thresholds: $0 = p_1 < \ldots < p_n = +\infty$. In the k-th round (search in $\mathcal{D}_k$), we remove all the sponsor sets whose confidence score is lower than p_k. We define the confidence score of a particular set as the number of times it appears as a sponsor of a basis set in subproblem solutions, which can be aggregated from the solutions to subproblems.

Phase Identification Problem

We employ a similar parallel scheme to solve the phase identification problem of combinatorial materials discovery, which also includes an exploration phase followed by an aggregation phase.

Exploration Phase In the Exploration Phase, a set of subproblems are solved in parallel. For the Phase Identification Problem, a subproblem is defined as finding the minimal number of phases to explain a *contiguous* region of sample points on the silicon wafer.

This is analogous to the exploration phase defined for set basis problem – finding basis for a subset of sets. The reason why we emphasize a contiguous region is because of the underlying physical constraint: the phase found must span a contiguous region in the silicon wafer. Figure 3.8 shows a sample decomposition into subproblems. Here each colored small region represents a subproblem.

Aggregation Phase The exploration phase produces a set of partial phases from solving subproblems. We call them partial because each of them describes only a subset of sample points.

As in the Set Basis Problem, we find partial phases can be merged together into larger phases. Figure 3.8 shows an illustrative example. Formally, two phases A and B may be *merged* into a new phase C, denoted as $C = A \circ B$, which contains all the peaks from A and B whose locations match across all the sample points they both present. The peaks in C then span the union of sample points of A and B. The merge operator $\circ$ plays the same role as the intersection operator of the Set Basis Problem. Similarly, we define $\overline{S}$ as the closure of (partial) phases S with respect to the merge operator $\circ$, which generates all possible merging of the phases in S.

Suppose $\mathcal{B}_0 = \cup_{i=1}^{s} \mathcal{B}_i$ is the set of all (partial) phases identified by solving subproblems, where $\mathcal{B}_i$ is the set of (partial) phases identified when solving subproblem i. As with the Set Basis Problem, the aggregation phase also has a pre-solving step, and a re-solving step. The pre-solving step takes as input the responses $\mathcal{B}_0$ from all subproblems, and extracts a subset of K partial phases from the closure $\overline{\mathcal{B}_0}$ as the candidate solution, which explains as many peaks on the silicon wafer as possible. The re-solving step searches in an iterative-deepening way for an exact solution, starting from the phases close to the candidate solution from the pre-solving step.

As in the pre-solving step of the Set Basis Problem, $\overline{\mathcal{B}_0}$ could be a large space and we are unable to enumerate all items in $\overline{\mathcal{B}_0}$ to find an exact solution. Instead, we take an approximate approach which first expands $\mathcal{B}_0$ to a larger set $\mathcal{B}' \subseteq \overline{\mathcal{B}_0}$ using a greedy approach. Then we employ a Mixed-Integer Program (MIP) formulation that selects the best K phases from $\mathcal{B}'$ which covers the largest number of peaks. The greedy algorithm and the MIP encoding are similar in concept to

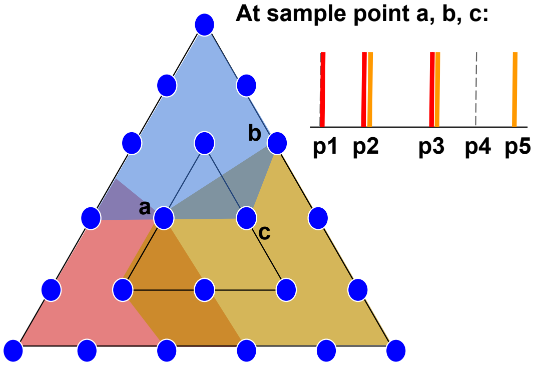


Figure 3.8: An example showing subproblem decomposition and merging of partial phases. (Subproblem decomposition) Each of the red, yellow and blue areas represents a subproblem, which is to find the minimal number of (partial) phases to explain all sample points in a colored area. (Merging of partial phases) Suppose partial phase A and B are discovered by solving the subproblem in the blue and the yellow region, respectively. A has peaks p_1, p_2, p_3 and all these peaks span the entire blue region, while B has peaks p_2, p_3, p_5 and all these peaks span the entire yellow region. Notice peaks p_2, p_3 match on sample points a, b and c, which are all the sample points in the intersection of the blue and yellow regions. Hence, the partial phases A and B can be merged into a larger phase C, which has peaks p_2 and p_3, but span all sample points in both the blue and yellow regions.

the ones used in solving the Set Basis Problem, but take into account extra physical constraints.

The Re-solving step expands the search from the pre-solving step in an iteratively deepening way to achieve completeness. Suppose the pre-solving step produces K phases $P_1^*, P_2^*, \ldots, P_K^*$. In the first round of the re-solving step, the complete solver is initialized such that the first phase must contain all the peaks of P_1^*, the second phase must contain all the peaks of P_2^*, etc. If the solver can find a solution with this initialization, then the solver terminates and returns the results. Otherwise, it usually detects a contradiction very quickly. In this case, we remove some peaks from $P_1^*, \ldots, P_K^*$ and re-solve the problem. We continue this re-solving process, until all the peaks from the Pre-solving step are removed, in which case the solver is free to explore the entire space without any restrictions. Again, this is highly unlikely in practice. In most cases, the solver is able to find solutions in the first one or two iterations.

3.2.4 Experiments

We test the performance of our parallel scheme on both the classic set basis problem, and on the phase identification problem in materials science.

Classic Set Basis Problem

Setup We test the parallel scheme on synthetic instances. We use a random ensemble similar to Molloy et al (Molloy et al., 2009), where every synthetic instance is characterized by n, m, k, e, p. To generate one synthetic instance, we first generate k basis sets. Every set contains $[n\frac{p}{100}]$ objects, uniformly sampled from a finite universe of n elements. We then generate m sets. Each set is a union of e randomly chosen basis sets from those initially generated.

We develop a Mixed Integer Programming (MIP) model to solve the set basis problem. The MIP model takes the original sets $\mathcal{C}$ and an integer K, and either returns a basis of size K that covers $\mathcal{C}$ exactly, or reports failure. We compare the performance of the MIP formulation with and without the initialization obtained using the parallel scheme described in the previous section.

We empirically observe high variability in the running times of the subproblems solved in the exploration phase, as commonly observed for NP-hard problems. Luckily, our parallel scheme can still be used without waiting for every sub-problem to complete. Specifically, we set up a cut-off threshold of 90%, such that the central process waits until 90% of sub-problems are solved before carrying out the aggregation phase. We also run 5 instances of the aggregation phase in parallel, each with a different random initialization, and terminate as soon as the fastest one finds a solution. In our experiment, $n = m = 100$. All MIPs are solved using IBM CPLEX 12.6, on a cluster of Intel x5690 3.46GHz core processors with 4 gigabytes of memory. We let each subproblem contain $T = 15$ sets for all instances.

Results Results obtained with and without initialization from parallel sub-problem solving are reported in Table 3.4. First, we see it takes much less wall-clock time (typically, by several orders of magnitude) for the complete solver to find the exact solution if it is initialized with the information collected from the sub-problems. The improvements are significant even when taking into account the time required for solving sub-problems in the exploration phase. In this case, we obtain several orders of magnitude saving in terms of solving time. For example, it takes about 50 seconds (wall-clock time) to solve A6, but about 5 hours without parallel initialization. Because we run $s = 100$ sub-problems in the exploration phase, another comparison would be based on CPU time, which is given by $(100 \cdot Exploration + 5 \cdot Aggregation)$. Under this measurement, our parallel scheme still outperforms the sequential approach on problem instances A2, A3, A5, A6, A8. Even though our CPU time is longer for some instances, our parallel scheme can be easily applied to thousands of cores. As parallel re-

[4] For this instance, 73 out of 100 subproblem instances complete within 2 hours. Thus the aggregation phase is conducted based on these instances. This exploration time here is calculated based on the slowest of the 73 instances.

Instance		Solution Quality							
		HPe		*Fast-Miner*		*ASSO*		*Complete*	
No.	K	K'	$E\%$	K'	$E\%$	K'	$E\%$	K'	$E\%$
A1	8	65	0	8	100	8	26.56	8	0
A2	8	86	0	8	100	8	18.18	8	0
A3	10	41	0	10	96.67	10	25	10	0
A4	10	47	0	10	100	10	18.92	10	0
A5	10	58	0	10	100	10	52	10	0
A6	12	51	0	12	96.3	12	25	12	0
A7	12	63	0	12	100	12	38.46	12	0
A8	12	93	0	12	100	12	51.06	12	0

Instance		Run-time for Complete Method (seconds)			
				Total Time	*Total Time*
No.	K	*Exploration*	*Aggregation*	*Parallel*	*Sequential*
A1	8	30.61	7.21	**37.82**	2199.07
A2	8	42.32	149.18	**191.5**	11374.34
A3	10	12.09	0.42	**12.51**	1561.46
A4	10	136.82	10.94	**147.76**	332.93
A5	10	55.52	2.60	**58.12**	57004.13
A6	12	46.85	0.75	**47.6**	17774.04
A7	12	6963.42[4]	13.47	**6967.89**	>72 hours
A8	12	176.4	5.77	**182.17**	>72 hours

Table 3.4: Comparison of different methods on classic set basis problems. K is the number basis sets used by the synthetic generator. In the solution quality block, we show the basis size K' and the error rate $E\%$ for incomplete method *HPe*, *FastMiner* and *ASSO* and the complete method. $K' > K$ means more basis sets are used than optimal. $E\% > 0$ means the coverage is not perfect. The running time for incomplete solvers are little, so they are not listed. In the run-time block, *Exploration*, *Aggregation*, *Total Time Parallel* and *Sequential* show the wall times of the corresponding phases in the parallel scheme and the time to solve the instance sequentially (*Total Time Parallel* = *Exploration* + *Aggregation*).

sources are becoming more and more accessible, it is obvious to see the benefit of this scheme. Note that we can also exploit *at the same time* the built-in parallelism of CPLEX to solve these instances. However, because CPLEX cannot explore the problem structure explicitly, it cannot achieve significant speed-ups on many instances. For example, it takes 12813.15, 259100.25 and 113475.12 seconds to solve the largest A6, A7 and A8 instances using CPLEX on 12-cores.

Although our focus is on improving the run-time of exact solvers, Table 3.4 also shows the performance of several state-of-the-art incomplete solvers on these synthetic instances. We implemented *FastMiner* from (Vaidya et al., 2006), *ASSO* from (Miettinen et al., 2008), and *HPe* from (Ene et al., 2008), which are among the most widely used incomplete algorithms. *FastMiner* and *ASSO* take the size of the basis K as input, and output K basis sets. They are incomplete in the sense that their solution may contain false positives and false negatives, which are defined as follows. c is a false positive element if $c \notin C_i$, but c is in one basis set that is a contributor to C_i. c is a false negative element, if $c \in C_i$, but c is not covered by any basis sets contributing for C_i. *FastMiner* does not provide the information about which basis set contributes to an original set. We therefore give the most conservative assignment: B is a contributor to C_i if and only if $B \subseteq C_i$. This assignment introduces no false positives. Both *FastMiner* and *ASSO* have parameters to tune. Our report are based on the best parameters we found. We report the maximum error rate in Table 3.4, which is defined as $\max_{C_i \in \mathcal{C}}\{(ft_i + ff_i)/|C_i|\}$, where ft_i and ff_i are the number of false positive and false negative elements at C_i, respectively. As seen from the table, neither of these two algorithms can recover the exact solution. *ASSO* performs better, but it still has 51.06% error rate on the hardest benchmark. We think the reason why *FastMiner* performs poorly is because it is usually used in situations where certain number of false positives can be tolerated. *HPe* is a graph based incomplete algorithm. It is guaranteed to find a complete cover, however it might require a number of basis sets K' larger than the optimal number K. We implemented both the greedy algorithm and the lattice-based post-improvement for *HPe*, and we used the best heuristic reported by the authors. As we can see from Table 3.4,

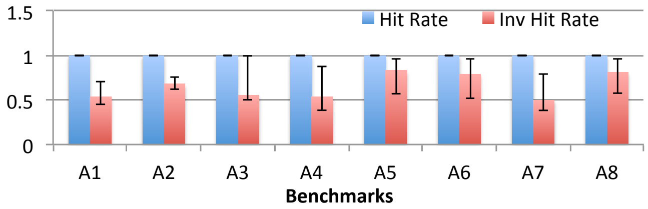


Figure 3.9: The overlap between the s-basis sets and the g-basis sets in each benchmark. The bars show the median value for (inverse) hitting rate, and error bars show the 10-th and 90-th percentile.

HPe often needs five times more basis sets to cover the entire collection.

The authors in (Ene et al., 2008) implemented the only complete solver we are aware of. Unfortunately, we can not obtain their code, so a direct comparison is not possible. However, the parallel scheme we developed does not make any assumption on the specific complete solver used. We expect other complete solvers (in addition to the MIP one we experimented with) will improve from the initialization information provided by solving subproblems.

Discussion We now provide empirical evidence that justifies and explains the surprising empirical effectiveness of our method. For clarity, we call a basis set found by solving subproblems an s-basis set, and a basis set in the global solution a g-basis set. For any set S, we define the *hitting rate* as: $p(S) = \max_{B \in \mathcal{B}} |S \cap B|/|B|$, and the *inverse hitting rate* as: $ip(S) = \max_{B \in \mathcal{B}} |S \cap B|/|S|$, where B is chosen among all g-basis set. Intuitively, $p(S)$ and $ip(S)$ measure the distance between S and the closest g-basis set. Note that $p(S) = 1$ (respectively $ip(S) = 1$) implies the basis S is the superset (respectively subset) of at least one g-basis in the global solution. If $p(S)$ and $ip(S)$ are both 1, then S matches exactly to one g-basis set[5].

[5] Assuming no g-basis set is the subset of another g-basis set, which is the case in instances A1 to A8.

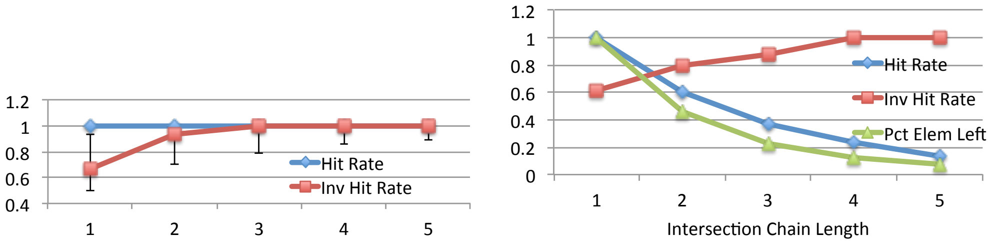


Figure 3.10: The median (inverse) hitting rate of a random intersection of multiple s-basis sets. X-axis shows the number of s-basis sets involved in the intersection. (Top) The basis sets in one intersection are supersets of one g-basis set. (Bottom) The basis sets in one intersection are randomly selected.

First, we study the overlap between a single s-basis set and all g-basis sets. As shown in Figure 3.9, across the benchmarks we considered the hitting rate is almost always one (with the lowest mean is for A2, which is 0.9983). This means that *the s-basis sets are almost always supersets of at least one g-basis set in the global solution*.

Next, we study the relationship between the intersection of multiple s-basis sets and g-basis sets. Figure 3.10 shows the median hitting rate and inverse hitting rate with respect to different number of s-basis sets involved in one intersection. The error bars show the 10-th and 90-th percentile. The result is averaged over all instances A1 through A8, with equal number of samples obtained from each instance. In the top chart, the s-basis sets involved in one intersection are supersets of one common g-basis set. In this case, the hitting rate is always 1. However, by intersecting only a few (2 or 3) s-basis sets, the inverse hitting rate becomes close to 1 as well, which implies the intersection becomes very close to an *exact* match of one g-basis set. This is in contrast with the result in the bottom chart, where the intersection is among randomly selected s-basis sets. In this case, when we increase the size of the intersection, fewer and fewer elements

System	# Points	Parallel (secs)	Sequential (secs)
A1	45	**119.22**	902.99
A2	45	**156.24**	588.85
A3	45	**74.37**	537.55
B1	60	**118.97**	972.8
B2	60	**177.89**	591.66
B3	60	**122.4**	1060.79
B4	60	**133.25**	633.52
C1	45	**3292.44**	17441.39
C2	45	**1186.70**	3948.41
D1	28	**207.92**	622.16
D2	28	**281.4**	2182.23
D3	28	**903.41**	2357.87

Table 3.5: The time for solving phase identification problems. # Points is the number of sample points in the system. Parallel and Sequential show the time to solve the problem with and without parallel initialization, respectively.

remain in the intersection. The bottom chart of Figure 3.10 shows the percentage of elements left, defined as $|\cap_{i=1}^{k} A_i| / \max_{i=1}^{k} |A_i|$. When intersecting 5 basis sets, in median case less than 10% elements still remain in the intersection.

The top and bottom charts of Figure 3.10 provide an empirical explanation for the success of our scheme: as we randomly intersect basis sets from the solutions to the subproblems, some intersections become close to the empty set (as in the bottom chart case), but others converge to one of the g-basis sets in the global solution (as in the upper chart case). In the second case, we obtain good solution candidates for the global problem by intersecting solutions to subproblems.

Phase Identification Problem in Materials Discovery

Setup We augmented the Satisfiability Modulo Theory formulation as described in (Ermon et al., 2012) with our parallel scheme and use the Z3 solver (De Moura & Bjørner, 2008) in the experiments. We use Z3 directly in the exploration phase, and then use it as a component of an iterative deepening search scheme in the aggregation phase. Due to a rather more imbalanced distribution of the running times across different sub-problems, we only wait for 50% of sub-problem solvers to complete before conducting the aggregation phase.

Table 3.5 displays the experimental results for the phase identification problem. We run on the same benchmark instances used in the work of Ermon et al (Ermon et al., 2012). We can see from Table 3.5 that in all cases the solver completes much faster when initialized with information obtained by parallel subproblem solving. This improvement in the run-time allows us to analyze much bigger problems than previously possible in combinatorial materials discovery.

3.2.5 Discussion

We introduced a novel angle for using parallelism to exploit hidden structure of hard dimensionality reduction problems with complex physical constraints. We demonstrated empirical success in solving the Set Basis Problem, obtaining over an order of magnitude speedups on certain problem instances. We also applied our parallel scheme to a novel application area, concerning the discovery of new materials for renewable energy sources. Future directions include applying this approach to other combinatorial optimization problems, and exploring its theoretical foundations.

CHAPTER 4

CONCLUSION

In this thesis, I first introduce a novel computational framework, based on embeddings, to tackle multi-stage inference problems at the intersection of reasoning, optimization, and learning, whose complexity is beyond NP. As a first example, I present a novel way to encode the reward allocation problem for a two-stage organizer–agent game as a single-stage optimization problem. The encoding embeds an approximation of the agents' decision-making process into the organizer's problem. We apply this methodology to eBird, a well-established citizen-science program for collecting bird observations, in a game called Avicaching. Our AI-based reward allocation was shown to be highly effective, surpassing the expectations of the eBird organizers and bird conservation experts. As a second example, I present a novel constant approximation algorithm to solve stochastic optimization problems which identifies the optimal policy that maximizes the expectation of a stochastic objective. To tackle this problem, I propose the embedding of its intractable counting subproblems as queries to NP oracles subject to additional XOR constraints. As a result, the entire problem is encoded as a single NP-equivalent optimization. The approach outperforms state-of-the-art solvers based on variational inference as well as MCMC sampling, on probabilistic inference benchmarks, deep learning applications, and a novel decision-making application in network design for wildlife conservation.

In addition, I apply the embedding technique to automated reasoning and machine learning for dimensionality reduction in scientific discovery. As one example, I propose embeddings based on Fourier analysis as a compact representation of high-dimensional probability distributions. I show that a large class of probabilistic models have a compact Fourier embedding. A simple variable elimination algorithm equipped with the Fourier embedding is able to match the performance of the state-of-the-art solvers in probabilistic inference. Motivated by an application in materials discovery with complex physical constraints, we show that human computation, crowdsourcing, and parallel computation can identify key backdoor information, thereby drastically reducing the computation time from days to minutes in a novel dimensionality reduction application to decompose signals in a high-dimensional space into a linear combination of a few basis patterns, subject to additional physical rules.

My research was made possible through the Computational Sustainability research network, via our collaboration with the *eBird* team of the Cornell Lab of Ornithology and the Joint Center for Artificial Photosynthesis (JCAP) at Caltech. By collaborating with ecologists from Cornell Lab of Ornithology, we were able to deploy our AI-based reward allocation in the field. The reward scheme proved to be highly effective, surpassing the expectations of the eBird organizers and bird conservation experts. We were also fortunate to deploy our materials discovery pipeline at JCAP. Materials scientists were able to analyze thousands of X-ray diffraction patterns with our system, and the results led to the discovery of new materials for energy applications. Our work was featured as the cover article and the Editors' Choice in the journal *Combinatorial Science* of the American Chemical Society. It also received recognition with the IAAI-2017 Innovative Application Award.

REFERENCES

Abbring, J. H., & Salimans, T. (2012). *The likelihood of mixed hitting times* (Tech. Rep.).

Achlioptas, D., & Jiang, P. (2015). Stochastic integration via error-correcting codes. In *Proc. uncertainty in artificial intelligence.*

Aggarwal, G., Feder, T., Motwani, R., & Zhu, A. (2004). Algorithms for multi-product pricing. In *International colloquium on automata, languages and programming.*

Agogino, A. K., & Tumer, K. (2008). Analyzing and visualizing multiagent rewards in dynamic and stochastic environments. *Journal of Autonomous Agents and Multi-Agent Systems*, *17*(2), 320-338.

Anderson, A., Huttenlocher, D. P., Kleinberg, J. M., & Leskovec, J. (2013). Steering user behavior with badges. In *22nd int'l world wide web conference, WWW.*

Bacon, D. F., Parkes, D. C., Chen, Y., Rao, M., Kash, I., & Sridharan, M. (2012). Predicting your own effort. In *Proc. of the 11th int'l conf. on autonomous agents and multiagent systems-volume 2* (pp. 695–702).

Bahar, R., Frohm, E., Gaona, C., Hachtel, G., Macii, E., Pardo, A., & Somenzi, F. (1993). Algebraic decision diagrams and their applications. In *Computer-aided design.*

Bai, J., Bjorck, J., Xue, Y., Suram, S. K., Gregoire, J., & Gomes, C. P. (2017). Relaxation methods for constrained matrix factorization problems: Solving the phase mapping problem in materials discovery. In *Proceedings of the 14th international conference on integration of artificial intelligence and operations research techniques in constraint programming (cpaior).*

Belle, V., Van den Broeck, G., & Passerini, A. (2015). Hashing-based approximate probabilistic inference in hybrid domains. In *Proceedings of the 31st uai conference.*

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems 19.*

Biere, A. (2010). *Lingeling, plingeling, picosat and precosat at sat race 2010* (Tech. Rep.). SAT race.

Biere, A. (2013). Lingeling, plingeling and treengeling entering the sat competition 2013. In *Proceedings of sat competition* (pp. 51–52).

Blum, A., Burch, C., & Langford, J. (1998). On Learning Monotone Boolean Functions. In *Focs* (pp. 408–415).

Bonney, R., Cooper, C. B., Dickinson, J., Kelling, S., Phillips, T., Rosenberg, K. V., & Shirk, J. (2009). Citizen science: a developing tool for expanding science knowledge and scientific literacy. *BioScience*, *59*(11), 977–984.

Boyd, S. P., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, *3*(1), 1–122.

Bragg, J., Mausam, & Weld, D. S. (2013). Crowdsourcing multi-label classification for taxonomy creation. In *Proceedings of the first aaai conference on human computation and crowdsourcing HCOMP.*

Briest, P., Hoefer, M., Gualà, L., & Ventre, C. (2009). On stackelberg pricing with computationally bounded consumers. In *WINE* (pp. 42–54).

Buchman, D., Schmidt, M. W., Mohamed, S., Poole, D., & de Freitas, N. (2012). On sparse, spectral and other parameterizations of binary probabilistic models. In *AISTATS.*

Cai, Y., Daskalakis, C., & Papadimitriou, C. H. (2014). Optimum statistical estimation with strategic data sources. *CoRR 14*.

Chakraborty, S., Fried, D., Meel, K. S., & Vardi, M. Y. (2015). From weighted to unweighted model counting. In *Proceedings of the 24th interational joint conference on ai (ijcai).*

Chavira, M., Darwiche, A., & Jaeger, M. (2006). Compiling relational bayesian networks for exact inference. *Int. J. Approx. Reasoning*.

Chen, X., Lin, Q., & Zhou, D. (2013). Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *ICML.*

Chrabakh, W., & Wolski, R. (2003). *Gradsat: A parallel sat solver for the grid* (Tech. Rep. No. 2003-05). UCSB Computer Science.

Chu, C., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G., Ng, A. Y., & Olukotun, K. (2007). Map-reduce for machine learning on multicore. *NIPS*.

Chu, G., Stuckey, P. J., & Harwood, A. (2008). *Pminisat: A parallelization of minisat 2.0* (Tech. Rep.). SAT race.

Chvatal, V. (1983). *Linear programming*. New York, USA: W. H. Freeman and Company.

Cohen, T., & Welling, M. (2015). Harmonic exponential families on manifolds. In *Proceedings of the 32nd international conference on machine learning, ICML.*

Conitzer, V., & Garera, N. (2006). Learning algorithms for online principal-agent problems (and selling goods online). In *Proc. of the 23rd icml.*

Conitzer, V., & Sandholm, T. (2006). Computing the optimal strategy to commit to. In *Proc. 7th ACM conference on electronic commerce (ec)* (pp. 82–90).

Darwiche, A., & Marquis, P. (2002). A knowledge compilation map. *J. Artif. Int. Res.*.

Dean, J., & Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107–113.

Dechter, R. (1997). Mini-buckets: A general scheme for generating approximations in automated reasoning. In *Proceedings of the fifteenth international joint conference on artificial intelligence.*

De Moura, L., & Bjørner, N. (2008). Z3: An efficient smt solver. In *Proceedings of the theory and practice of software, 14th international conference on tools and algorithms for the construction and analysis of systems* (pp. 337–340).

Dilkina, B., Gomes, C., Malitsky, Y., Sabharwal, A., & Sellmann, M. (2009). Backdoors to combinatorial optimization: Feasibility and optimality. *CPAIOR*, 56–70.

Endriss, U., Kraus, S., Lang, J., & Wooldridge, M. (2011). Incentive engineering for boolean games. *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, *22*(3), 2602.

Ene, A., Horne, W. G., Milosavljevic, N., Rao, P., Schreiber, R., & Tarjan, R. E. (2008). Fast exact and heuristic methods for role minimization problems. In I. Ray & N. Li (Eds.), *Sacmat* (p. 1-10). ACM.

Ermon, S., Gomes, C. P., Sabharwal, A., & Selman, B. (2013a). Embed and project: Discrete sampling with universal hashing. In *Advances in neural information processing systems (nips)* (pp. 2085–2093).

Ermon, S., Gomes, C. P., Sabharwal, A., & Selman, B. (2013b). Optimization with parity constraints: From binary codes to discrete integration. In *Proceedings of the twenty-ninth conference on uncertainty in artificial intelligence, UAI.*

Ermon, S., Gomes, C. P., Sabharwal, A., & Selman, B. (2013c). Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *Proceedings of the 30th international conference on machine learning, ICML.*

Ermon, S., Gomes, C. P., Sabharwal, A., & Selman, B. (2014). Low-density parity constraints for hashing-based discrete integration. In *Proceedings of the 31th international conference on machine learning, ICML.*

Ermon, S., Le Bras, R., Gomes, C. P., Selman, B., & van Dover, R. B. (2012). Smt-aided combinatorial materials discovery. In *Sat'12.*

Fang, F., Stone, P., & Tambe, M. (2015). When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *IJCAI.*

Fischetti, M., Monaci, M., & Salvagnin, D. (2014). Self-splitting of workload in parallel computation. In *Cpaior* (p. 394-404).

Flerova, N., Ihler, E., Dechter, R., & Otten, L. (2011). Mini-bucket elimination with moment matching. In *In nips workshop discml.*

Frazier, P., Kempe, D., Kleinberg, J. M., & Kleinberg, R. (2014). Incentivizing exploration. In *ACM conference on economics and computation, EC* (pp. 5–22).

Friesen, A. L., & Domingos, P. (2015). Recursive decomposition for nonconvex optimization. In *Proceedings of the 24th international joint conference on artificial intelligence.*

Gogate, V., & Dechter, R. (2004). A complete anytime algorithm for treewidth. In *Proceedings of the 20th conference on uncertainty in artificial intelligence.*

Gogate, V., & Domingos, P. M. (2013). Structured message passing. In *Uai.*

Gomes, C. P. (Winter 2009). Computational Sustainability: Computational methods for a sustainable environment, economy, and society. *The Bridge, National Academy of Engineering*, *39*(4).

Gregoire, J. M., Dale, D., Kazimirov, A., DiSalvo, F. J., & van Dover, R. B. (2009). High energy x-ray diffraction/x-ray fluorescence spectroscopy for high-throughput analysis of composition spread thin films. *Review of Scientific*

Instruments, *80*(12), 123905–123905.

Guruswami, V., Hartline, J. D., Karlin, A. R., Kempe, D., Kenyon, C., & McSherry, F. (2005). On profit-maximizing envy-free pricing. In *Soda* (pp. 1164–1173).

Hamadi, Y., Marques-Silva, J., & Wintersteiger, C. M. (2011). Lazy decomposition for distributed decision procedures. In *Pdmc* (p. 43-54).

Hamadi, Y., & Sais, L. (2009). Manysat: a parallel sat solver. *Journal On Satisfiability, Boolean Modeling and Computation (JSAT)*, *6*.

Hamadi, Y., & Wintersteiger, C. M. (2013). Seven challenges in parallel SAT solving. *AI Magazine*, *34*(2), 99–106.

Hartline, J. D., & Koltun, V. (2005). Near-optimal pricing in near-linear time. In *Algorithms and data structures, 9th int'l workshop, WADS* (pp. 422–431).

Hrastad, J. (1987). *Computational limitations of small-depth circuits*. Cambridge, MA, USA: MIT Press.

Hazan, T., & Jaakkola, T. S. (2012). On the partition function and random maximum a-posteriori perturbations. In *ICML.*

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition, CVPR* (pp. 770–778).

Heskes, T., Albers, K., & Kappen, B. (2003). Approximate inference and constrained optimization. In *Proceedings of the nineteenth conference on uncertainty in artificial intelligence.*

Hinton, G., & Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504 - 507.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 1771–1800.

Homer, C., Dewitz, J., Fry, J., Coan, M., Hossain, N., Larson, C., ... Wickham, J. (2007). Completion of the 2001 National Land Cover Database for the conterminous United States. *Photogrammetric Engineering and Remote Sensing*, *73*(4), 337–341.

Hu, J., & Wellman, M. P. (1998). Online learning about other agents in a dynamic multiagent system. In *Proceedings of the second international conference on autonomous agents.*

Huang, J., Guestrin, C., & Guibas, L. J. (2009). Fourier theoretic probabilistic inference over permutations. *Journal of Machine Learning Research*, *10*, 997–1070.

Ihler, A. T., Flerova, N., Dechter, R., & Otten, L. (2012). Join-graph based cost-shifting schemes. In *Uai.*

Jain, S., Chen, Y., & Parkes, D. C. (2014). Designing incentives for online question-and-answer forums. *Games and Economic Behavior*, *86*, 458–474.

Jiang, J., Rai, P., & III, H. D. (2011). Message-passing for approximate MAP inference with latent variables. In *Advances in neural information processing systems 24.*

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Mach. Learn.*.

Kadioglu, S., Malitsky, Y., Sabharwal, A., Samulowitz, H., & Sellmann, M. (2011). Algorithm selection and scheduling. In *Proceedings of the 17th international conference on principles and practice of constraint programming* (pp. 454–469).

Kawajiri, R., Shimosaka, M., & Kashima, H. (2014). Steered crowdsensing: Incentive design towards quality-oriented place-centric crowdsensing. In *Ubicomp.*

Kelling, S., Gerbracht, J., Fink, D., Lagoze, C., Wong, W.-K., Yu, J., … Gomes, C. P. (2012). ebird: A human/computer learning network for biodiversity conservation and research. In *Proceedings of the twenty-fourth conference on innovative applications of artificial intelligence (iaai).*

Kottler, S., & Kaufmann, M. (2011). SArTagnan - A parallel portfolio SAT solver with lockless physical clause sharing. In *Pragmatics of sat.*

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th international conference on neural information processing systems* (pp. 1097–1105).

Le Bras, R., Damoulas, T., Gregoire, J. M., Sabharwal, A., Gomes, C. P., & van Dover, R. B. (2011). Constraint reasoning and kernel clustering for pattern decomposition with scaling. In *Cp'11* (pp. 508–522).

Le Bras, R., Xue, Y., Bernstein, R., Gomes, C. P., & Selman, B. (2014). A human computation framework for boosting combinatorial solvers. In *Proceedings of the second AAAI conference on human computation and crowdsourcing, HCOMP.*

Lee, J., Marinescu, R., Dechter, R., & Ihler, A. T. (2016). From exact to anytime solutions for marginal MAP. In *Proceedings of the thirtieth AAAI conference on artificial intelligence, aaai.*

Leyton-Brown, K., Nudelman, E., Andrew, G., Mcfadden, J., & Shoham, Y. (2003). A portfolio approach to algorithm selection. In *Ijcai'03* (pp. 1542–1543).

Li, H., Tian, F., Chen, W., Qin, T., Ma, Z., & Liu, T. (2015). Generalization analysis for game-theoretic machine learning. In *AAAI.*

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*.

Linial, N., Mansour, Y., & Nisan, N. (1993). Constant depth circuits, fourier transform, and learnability. *J. ACM*, *40*(3).

Lintott, C. J., Schawinski, K., Slosar, A., et al. (2008, September 21). Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, *389*(3), 1179–1189.

Liu, Q., & Ihler, A. T. (2013). Variational algorithms for marginal MAP. *Journal of Machine Learning Research*, *14*.

Malitsky, Y., Sabharwal, A., Samulowitz, H., & Sellmann, M. (2011). Non-model-based algorithm portfolios for sat. In *Proceedings of the 14th international conference on theory and application of satisfiability testing* (pp. 369–370).

Mansour, Y. (1994). Learning Boolean functions via the Fourier transform. *advances in neural computation and learning*, *0*, 1–28.

Marinescu, R., Dechter, R., & Ihler, A. (2015). Pushing forward marginal map with best-first search. In *Proceedings of the 24th international conference on artificial intelligence (ijcai).*

Marinescu, R., Dechter, R., & Ihler, A. T. (2014). AND/OR search for marginal MAP. In *Proceedings of the thirtieth conference on uncertainty in artificial intelligence, UAI.*

Martins, A. F. T., Figueiredo, M. A. T., Aguiar, P. M. Q., Smith, N. A., & Xing, E. P. (2011). An augmented lagrangian approach to constrained MAP inference. In *Proceedings of icml* (pp. 169–176).

Mateescu, R., Kask, K., Gogate, V., & Dechter, R. (2010). Join-graph propagation algorithms. *J. Artif. Intell. Res. (JAIR)*, *37*.

Mauá, D. D., & de Campos, C. P. (2012). Anytime marginal MAP inference. In *Proceedings of the 29th international conference on machine learning, ICML.*

Miettinen, P., Mielikainen, T., Gionis, A., Das, G., & Mannila, H. (2008). The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering*, *20*(10), 1348-1362.

Minder, P., Seuken, S., Bernstein, A., & Zollinger, M. (2012). Crowdmanager-combinatorial allocation and pricing of crowdsourcing tasks with time constraints. In *Workshop on social computing and user generated content in conjunction with acm conference on electronic commerce (acm-ec 2012).*

Moisan, T., Gaudreault, J., & Quimper, C.-G. (2013). Parallel discrepancy-based search. In C. Schulte (Ed.), *Cp* (p. 30-46).

Molloy, I., Li, N., Li, T., Mao, Z., Wang, Q., & Lobo, J. (2009). Evaluating role mining algorithms. In B. Carminati & J. Joshi (Eds.), *Sacmat* (p. 95-104). ACM.

Mooij, J. M. (2010, August). libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, *11*, 2169-2173.

Nau, D. S., Markowsky, G., Woodbury, M. A., & Amos, D. B. (1978). A mathematical analysis of human leukocyte antigen serology. *Mathematical Biosciences*, *40*(34), 243 - 270.

Ng, A. Y., & Russell, S. J. (2000). Algorithms for inverse reinforcement learning. In *Icml* (pp. 663–670).

O'Donnell, R. (2008). Some topics in analysis of boolean functions. *Proceedings of the fourtieth annual ACM symposium on Theory of computing - STOC 08*, 569.

Ogata, Y., & Tanemura, M. (1981). Estimation of interaction potentials of spatial point patterns through the maximum likelihood procedure. *Annals of the Institute of Statistical Mathematics*.

O'Mahony, E., Hebrard, E., Holland, A., & Nugent, C. (2008). Using case-based reasoning in an algorithm portfolio for constraint solving. In *Irish conference on artificial intelligence and cognitive science.*

Park, J. D., & Darwiche, A. (2003). Solving map exactly using systematic search. In *Proceedings of the nineteenth conference on uncertainty in artificial intelligence (uai).*

Park, J. D., & Darwiche, A. (2004). Complexity results and approximation strategies for map explanations. *J. Artif. Int. Res.*.

Paruchuri, P., Pearce, J. P., Marecki, J., Tambe, M., Ordóñez, F., & Kraus, S. (2008). Playing games for security: an efficient exact algorithm for solving bayesian stackelberg games. In *Aamas* (pp. 895–902).

Patel, P. (2011). Materials genome initiative and energy. *MRS bulletin*, *36*(12), 964–966.

Ping, W., Liu, Q., & Ihler, A. T. (2015). Decomposition bounds for marginal MAP. In *Advances in neural information processing systems 28.*

Radanovic, G., & Faltings, B. (2015). Incentive schemes for participatory sensing. In *AAMAS.*

Rao, V., & Kumar, V. (1993, Apr). On the efficiency of parallel backtracking. *Parallel and Distributed Systems, IEEE Transactions on*, *4*(4), 427-437.

Razborov, A. A. (1995). Bounded arithmetic and lower bounds in boolean complexity. In *Feasible mathematics ii.*

Regin, J.-C., Rezgui, M., & Malapert, A. (2013). Embarrassingly parallel search. In *Cp* (Vol. 8124, p. 596-610). Springer.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Mach. Learn.*.

Rogers, L. C. G. (2000). Evaluating first-passage probabilities for spectrally one-sided lévy processes. *Journal of Applied Probability*, 1173–1180.

Schubert, T., Lewis, M., & Becker, B. (2010). *Antom solver description* (Tech. Rep.). SAT race.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, *52*(55-66), 11.

Shavell, S. (1979). Risk sharing and incentives in the principal and agent relationship. *The Bell Journal of Economics*.

Sheldon, D., Dilkina, B. N., Elmachtoub, A. N., Finseth, R., Sabharwal, A., Conrad, J., … Vaughan, W. (2010). Maximizing the spread of cascades using network design. In *UAI.*

Shu, G., Lee, D., & Yannakakis, M. (2006). A note on broadcast encryption key management with applications to large scale emergency alert systems. In *20th international parallel and distributed processing symposium (ipdps)* (p. 8 pp.-).

Singer, Y., & Mittal, M. (2013). Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd internat. conf. on world wide web (www).*

Singla, A., & Krause, A. (2013). Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd international conference on world wide web.*

Singla, A., Santoni, M., Bartók, G., Mukerji, P., Meenen, M., & Krause, A. (2015). Incentivizing users for balancing bike sharing systems. In *AAAI.*

Smith, D., & Gogate, V. (2013). The inclusion-exclusion rule and its application to the junction tree algorithm. In *Proceedings of the twenty-third international joint conference on artificial intelligence.*

Sontag, D., Meltzer, T., Globerson, A., Jaakkola, T., & Weiss, Y. (2008). Tightening lp relaxations for map using message passing. In *Uai* (pp. 503–510).

Stockmeyer, L. J. (1975). *The set basis problem is np-complete* (Technical Report No.

RC-5431). IBM.

Sullivan, B. L., Aycrigg, J. L., Barry, J. H., Bonney, R. E., Bruns, N., Cooper, C. B., . . . Kelling, S. (2014). The ebird enterprise: An integrated approach to development and application of citizen science. *Biological Conservation*, *169*(0), 31 - 40.

Suram, S. K., Xue, Y., Bai, J., Le Bras, R., Rappazzo, B., Bernstein, R., . . . Gregoire, J. (2016). Automated phase mapping with agilefd and its application to light absorber discovery in the v-mn-nb oxide system. *American Chemical Society Combinatorial Science (Editor's Choice and Cover Story)*.

Syed, U., Bowling, M., & Schapire, R. E. (2008). Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on machine learning.*

Takeuchi, I., Dover, R. B. v., & Koinuma, H. (2002). Combinatorial synthesis and evaluation of functional inorganic materials using thin-film techniques. *MRS bulletin*, *27*(04), 301–308.

Tran-Thanh, L., Huynh, T. D., Rosenfeld, A., Ramchurn, S. D., & Jennings, N. R. (2015). Crowdsourcing complex workflows under budget constraints. In *Proceedings of the aaai conference.* AAAI.

Tran-Thanh, L., Stein, S., Rogers, A., & Jennings, N. R. (2014). Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, *214*, 89–111.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005, December). Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, *6*, 1453–1484.

Uai 2010 approximate inference challenge. (n.d.). Retrieved from `http://www.cs.huji.ac.il/project/UAI10`

Vaidya, J., Atluri, V., & Guo, Q. (2007). The role mining problem: Finding a minimal descriptive set of roles. In *In symposium on access control models and technologies (sacmat)* (pp. 175–184).

Vaidya, J., Atluri, V., & Warner, J. (2006). Roleminer: Mining roles using subset enumeration. In *Proceedings of the 13th acm conference on computer and communications security.*

van Rooij, J. M. M., Bodlaender, H. L., & Rossmanith, P. (2009). Dynamic programming on tree decompositions using generalised fast subset convolution. In *Algorithms - ESA, 17th annual european symposium.*

Wainwright, M. J., Jaakkola, T. S., & Willsky, A. S. (2003). Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudomoment matching. In *Proceedings of the ninth international workshop on artificial intelligence and statistics.*

White, A. (2012). The materials genome initiative: One year on. *MRS Bulletin*, *37*(08), 715–716.

Williams, R., Gomes, C., & Selman, B. (2003). Backdoors to typical case complexity. In *Ijcai'03* (Vol. 18, pp. 1173–1178).

Wu, X., Xue, Y., Selman, B., & Gomes, C. P. (2017). Xor-sampling for network design with correlated stochastic events. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI* (pp. 4640–4647).

Xu, L., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2008). Satzilla: Portfolio-based algorithm selection for sat. *J. Artif. Intell. Res. (JAIR)*, *32*, 565-606.

Xue, S., Fern, A., & Sheldon, D. (2015). Scheduling conservation designs for maximum flexibility via network cascade optimization. *J. Artif. Intell. Res. (JAIR).*

Xue, Y., Bai, J., Le Bras, R., Rappazzo, B., Bernstein, R., Bjorck, J., ... Gomes,

C. P. (2017). Phase-mapper: An ai platform to accelerate high throughput materials discovery. In *Proceedings of the 29th annual conference on innovative applications of artificial intelligence (iaai) (iaai innovative application award).*

Xue, Y., Davies, I., Fink, D., Wood, C., & Gomes, C. P. (2016a). Avicaching: A two stage game for bias reduction in citizen science. In *Proceedings of the 15th international conference on autonomous agents and multiagent systems (aamas).*

Xue, Y., Davies, I., Fink, D., Wood, C., & Gomes, C. P. (2016b). Behavior identification in two-stage games for incentivizing citizen science exploration. In *Proceedings of the 22nd international conference on principles and practice of constraint programming (cp).*

Xue, Y., Ermon, S., Bras, R. L., Gomes, C. P., & Selman, B. (2016). Variable elimination in the fourier domain. In *Proceedings of the 33nd international conference on machine learning, ICML.*

Xue, Y., Ermon, S., Gomes, C. P., & Selman, B. (2015). Uncovering hidden structure through parallel problem decomposition for the set basis problem: Application to materials discovery. In *Proceedings of the twenty-fourth international joint conference on artificial intelligence, IJCAI.*

Xue, Y., Li, Z., Ermon, S., Gomes, C. P., & Selman, B. (2016). Solving marginal map problems with np oracles and parity constraints. In *Proceedings of the 29th annual conference on neural information processing systems (nips).*

Xue, Y., Wu, X., Morin, D., Dilkina, B., Fuller, A., Royle, J. A., & Gomes, C. P. (2017). Dynamic optimization of landscape connectivity embedding spatial-capture-recapture information. In *Proceedings of the 31th aaai conference on artificial intelligence (aaai).*

Zhang, H., Law, E., Miller, R., Gajos, K., Parkes, D., & Horvitz, E. (2012). Human computation tasks with global constraints. In *Chi.*

Zilli, D., Parson, O., Merrett, G. V., & Rogers, A. (2014). A hidden markov model-based acoustic cicada detector for crowdsourced smartphone biodiversity monitoring. *J. Artif. Intell. Res. (JAIR)*, *51*, 805–827.