Diss. ETH No. 25007

STATISTICAL MECHANICS AND INFORMATION THEORY IN APPROXIMATE ROBUST INFERENCE

A thesis submitted to attain the degree of
Doctor of Sciences of ETH Zurich
(Dr. sc. ETH Zurich)

presented by

Alexey GRONSKIY
Specialist in Mathematics,
Lomonosov Moscow State University

born 16 November 1989
citizen of the Russian Federation

accepted on the recommendation of

Prof. Dr. Joachim M. Buhmann, examiner
Prof. Dr. Peter Widmayer, co-examiner
Prof. Dr. Wojciech Szpankowski, co-examiner

2018

Abstract

Dealing with noisy inputs to optimization problems has been one of the central tasks in the field of inference since its invention. The reason is clear: no data comes without measurement uncertainty. *Robustly optimizing* under such uncertainty is the first cornerstone of this thesis. The second cornerstone stems from an idea of contemplating uncertain optimization problems — combinatorial ones in our case, — as *large disordered systems* and analyzing their optimization behavior using the methods of statistical mechanics. Questions concerning optimization methods per se — e.g. related to a specific optimization procedure or guarantees thereof, — are not considered in this work.

More precisely, we first address robust optimization under uncertainty by means of an approximation set-based approach to inference. Here, *approximation set-based* approach refers to a family of methods regularizing the Empirical Risk Minimization (ERM). One performs it by sampling from a set of near-optimal solutions instead of returning the ERM optimizer. We will describe this approach and design proof-of-concept experiments which support its usability. In addition, we will address one of the known computational bottlenecks of the approach by proving and testing theoretical results.

Second, we expand the approximation set-based approach to the field of algorithmic combinatorial optimization under uncertain input, picking as an example its application to the Minimum Spanning Tree (MST) problem. In short, the aim is to perform approximation by means of optimal stopping of the algorithm. We will derive a fast computation pipeline which is free of computational bottlenecks mentioned above; we then carry out experiments revealing an ability of our approach not only to robustly solve MST problems, but also to establish a score that ranks various algorithms according to their expected localization error — as in model validation.

Next, we gradually shift our focus to Maximum Entropy inference for combinatorial optimization problems. Considering them as large disordered particle systems (Mezard and Montanari, 2009) driven by energy optimization via Gibbs distributions, we rigorously study a so-called *Gibbs relaxation* of the approximation set-based approach. Surprisingly, this study leads to a prominent mathematical problem — asymptotically-precise computing of *free energy*, — which we then fully solve for a special class of combinatorial optimization problems. Thus, in this part of the thesis we fulfill a twofold task: on the one hand, we study proper-

ties of Gibbs relaxation of approximate robust optimization; the other, we prove a theoretical result, which is of an independent interest in statistical mechanics.

Last, we drift away from approximate inference and remain in statistical mechanical setting. Inspired by the theoretical results described in the previous paragrpah, we ask fundamental questions about statistical mechanics of combinatorial optimization problems and how their properties define their solution structure. We pick a combinatorial optimization problem (sparse Minimum Bisection Problem, sMBP) and compare its asymptotic behavior with the one of well-known Random Energy Model (REM, Derrida, 1981), leading to some interesting conjectures about differences in their search complexity.

Zusammenfassung

Von Anfang an war die Einbeziehung des Rauschens in die Eingabe von Optimierungsproblemen ein zentrales Problem im Bereich der Inferenz. Der Grund dafür ist klar: Daten sind inherent mit gewissen Unsicherheiten in Ihrer Erhebung verbunden. *Robuste Optimierung* unter Berücksichtigung solcher Unsicherheiten ist der erste Eckpfeiler dieser Dissertation. Der zweite Eckpfeiler beruht auf der Idee, dass man verrauschte Optimierungsprobleme — in unserem Fall kombinatorische, — als *grosse ungeordnete Systeme* betrachtet und deren Optimierungsverhalten mittels statistischer Mechanik analysiert. Die Fragen, die sich auf Optimierungsverfahren beziehen (z.B. im Zusammenhang mit einer spezifischen Optimierungsmethode oder deren Effizienz), sind in dieser Arbeit nicht berücksichtigt.

Erstens konzentrieren wir uns auf robuste Optimierung unter Unsicherheit mittels des *auf Approximationsmengen-basierten* (approximation set-based) Ansatzes für Inferenz. Hier bezieht sich der Begriff des Approximationsmengen basierten Ansatzes auf eine Familie von Methoden, welche die Empirische Risiko-Minimierung (Empirical Risk Minimization, ERM) regularisieren. Man macht es, indem man aus einer Menge von nahezu optimalen Lösungen Stichproben zieht, statt eine ERM-Lösung als Antwort zurückzugeben. Wir beschreiben diesen Ansatz und führen Proof-of-Concept-Experimente aus, welche die Anwendbarkeit des Konzepts zeigen. Ausserdem beschäftigen wir uns mit den bekannten rechnerischen Engpässen diesen Ansatzes, indem wir theoretische Einsichten anbieten und testen.

Zweitens, erweitern wir den Approximationsmengen-basierten Ansatz auf dem Gebiet der kombinatorischen Optimierung gegeben Unsicherheit in der Eingabe, indem wir als Beispiel das Minimale-Spannbaum-Problem auswählen (Minimum Spanning Tree, MST). Kurz gesagt besteht unser Ziel darin, eine Approximation durch optimales Anhalten des Algorithmus zu erzielen. Wir präsentieren eine schnelle Berechnungspipeline, die von den oben erwähnten Berechnungsenpässen befreit ist; dazu führen wir Experimente durch, die zeigen dass der oben definierte Ansatz nicht nur in der Lage ist, MST-Probleme zuverlässig zu lösen, sondern auch einen Ranking Wert einführt, der verschiedene Algorithmen nach ihrem Lokalisierungsfehler bewertet — wie beim Modellvalidierungsverfahren.

Im Anschluss richten wir den Fokus auf die Maximum-Entropie-Inferenz für kombinatorische Optimierungsprobleme. Wir betrachten sie als grosse ungeord-

nete Partikelsysteme (Mezard and Montanari, 2009), die ihr Energieniveau mittels Gibbs-Verteilungen optimieren. Wir führen eine thoretische Untersuchung der sogenannten *Gibbs-Relaxation* des Approximationsmengen basierten Ansatzes aus. Unerwartet, führt diese Untersuchung zu einem berühmten mathematischen Problem — der asymptotisch genauen Berechnung der *freien Energie*, — welches wir dann für eine spezielle Klasse von kombinatorischen Optimierungsproblemen vollständig lösen. Somit liefert dieser Teil zwei Beiträge: einerseits untersuchen wir Eigenschaften der Gibbs-Relaxation für robuste Optimierung; andererseits erreichen wir ein theoretisches Ergebnis, das eine unabhängige Bedeutung für die statistische Mechanik hat.

Zuletzt entfernen wir uns von robuster Inferenz und bleiben im statistisch-mechanischen Bereich. Inspiriert durch die oben erwähnten theoretischen Ergebnisse, stellen wir fundamentale Fragen zur statistischen Mechanik kombinatorischer Optimierungsprobleme und wie deren Eigenschaften die Lösungsstruktur definieren. Wir wählen ein kombinatorisches Optimierungsproblem (sparse Minimum Bisection Problem, sMBP) und vergleichen sein asymptotisches Verhalten mit dem des bekannten Random Energy Models (REM, Derrida, 1981), was zu einigen spannenden Vermutungen über Unterschiede in ihrer Suchkomplexität führt.

Acknowledgments

First of all, I am deeply grateful to my advisor, Prof. Dr. Joachim M. Buhmann, who gave me an opportunity to be part of his group and who, although allowing me much freedom in choosing topics, always guaranteed me necessary help and provided guidance. Besides infallible scientific interest, he always supported me in non-professional matters.

I would also like to thank my co-examiners, Prof. Dr. Peter Widmayer from the ETH Zurich and Prof. Dr. Wojciech Szpankowski from Purdue University for taking time to read through this opus, give valuable comments and examine me. Throughout my studies, Peter Widmayer kept encouraging me and was involved in numerous discussions on a regular basis. Wojciech Szpankowski became my major collaborator for a large part of the time I spent on my PhD. He shared valuable ideas, supported me, taught me a word "prodding" and showed what it meant in practice. He also organized my and my student's visit to Purdue, during which we made significant progress.

I will always be grateful to my first university lecturer and scientific advisor in Moscow, Prof. Dr. Alexander Ugolnikov, who is, unfortunately, no longer with us. He introduced me to the field of discrete mathematics and inspired me to combine humor and seriousness, not forgetting about life while doing research.

Apart from mentioned above, thanks to my close collaborators with whom I had lots of fruitful discussions on the topics of this thesis: Tobias Pröger (who also kindly agreed to proofread this thesis), Rastislav Šrámek, Paolo Penna, Matús Mihalák, An Bian, Nico Gorbach and Stefan Bauer.

The Information Science and Engineering group (formerly the Machine Learning group) at the ETH Zurich accompanied me from the very first days in Zurich, and I enjoyed being part of it. Colleagues from my group, as well as from the ones of Prof. Dr. Andreas Krause (LAS group) and Prof. Dr. Thomas Hofmann (DALab), taught me a lot during the numerous occasions we had to interact. Thanks to David, Brian, Sharon, Patrick, Gabriel, Hasta, Morteza, Alberto, Ludwig, Peter, Dwarikanath, Dima, Kate, Judith, Rebekka, Luis, Nico, Stefan, Djordje, An, Luca, Viktor, Alina and Aytunç for a wonderful time.

If one sees a scientific group as an organism, then heads and bodies have already been mentioned. But a group is still nothing without its heart which actually pumps life into it: thanks to our great administrators Rita Klute and Marianna Berger for their friendliness, support, help and organizing skills.

I thank my Master's students Julien, George and Edouard, with whom we worked a lot and learned much together. I also constantly learned from numerous students of the ETH who attended our Machine Learning and Statistical Learning Theory courses.

I was lucky to spend a summer with Google Research in Zurich, and I am grateful to my host Neil Houlsby, who was supporting and patient towards me as I slowly progressed through my internship, and to the rest of the hosting group: Massimiliano, Jannis, Christian and Wojciech.

The life of a PhD student is, of course, not all about studying and research, but also about *simply* life — in all its variety. Thanks to my Russian friends Dima, Valya, Valera, Masha, Kate, Roma, Martin, Nikolay, Alexander, Arseniy, Anna, Vita and Iuliya for their support, good mood and for the great time we spent already and still spend together. Further, while doing research, one must periodically clear head with hobbies, so thanks to SWISS Flying Club, whose instructors introduced me to the art of flying an airplane and helped to see Switzerland from a bird's-eye view; thanks to the Zurich Rescue and Ambulance Service, whose instructors accepted me as a volunteer and persevered to train me, despite my awful Swiss German in the very beginning. Thanks also to ASVZ and Kim Dojo Karate Clubs whose members actively helped me not to become overly self-confident at times of my research euphoria.

Thanks to my loving parents Yury and Natalya, who raised me not mechanistically, but rather showed by their own example what it means — to be genuinely interested in life. To always be curious, open, to keep pushing and not give up. They are both mathematicians, and I acquired a large part of my early interest in programming and mathematics from them. My younger brothers Dima and Ivan are my first and best friends, and we keep learning from each other till today.

And most importantly, infinite (as in "∞") gratitude goes to my loving wife Lena. The level of encouragement and support I get from her is more than enormous. She laughs and makes me laugh even when everything seems ruined and frustrating. She is supporting in whatever we do together. Holding PhD herself, she understands me as nobody else ever did. Thank you, my darling, for being there. The fact that this text is eventually out is your achievement no less than mine.

UPD on the 10^{th} of March 2018. Four days after the defense of this thesis, my wife Lena gave birth to our wonderful son Andrey. Existence of such miracles as an emerging human life is the greatest support, motivation, and reward for me.

To our parents, who gave us the best they had.

To my wife, with whom we try to transform it into something even better.

To our coming children, who will in turn receive it from us
and hopefully take over this ever-recurring task.

Contents

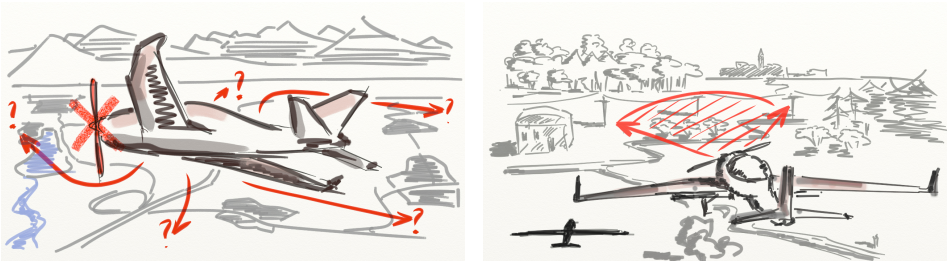
1
Informal Introduction

"Se jeunesse savoit; si viellesse pouvoit."
(fr. "If youth knew; if age could.")

— Henri ESTIENNE

1.1 Information vs. Freedom of Choice

When a light aircraft, while flying on a sunny day in a relatively flat area around Langenthal in Switzerland, gets a sudden engine failure at a high altitude, it is usually not the landing moment itself that constitutes the difficulty — a well-trained pilot can land his engineless airplane anywhere, provided some conditions are met (flatness, no major obstacles, good ground quality). The problem of surviving an engine failure actually becomes an optimization problem of finding the best field. And two contradictory processes are happening at the same time: while the airplane glides down, the pilot gets more and more information about the approaching terrain, as he sees it better (trees, power lines, small rivers). But at the same time he "zooms in" closer to the terrain, loses altitude and has fewer and fewer fields available in his reach (Figure 1.1).



(a) Little information, many choices **(b)** Much information, few choices

Figure 1.1 So much of a problem...

A field which looks great from a high altitude can later turn out to be crossed by a small creek with a power line on the front side, and pilot cannot climb any more to reach "that another nice one I've seen over that hill". It is an inevitable trade-off: *information almost always comes at a cost of reduced freedom of action* (hence the epigraph to this chapter). One cannot infinitely search for the best solution and has to stop and commit at some point. How to choose that point? A good question for a pilot.

But the very same question: "when to stop (commit)?" — can be applied to the field of robust algorithmic optimization. To bring an example, let us consider some stepwise algorithm, e.g. Prim's algorithm for finding the Minimum Spanning Tree (Prim, 1957). It will be discussed in detail in Chapter 4, but for now we concentrate on a simplistic explanation of its main property (Algorithm 1.1): gradually building the optimal solution by reducing the scope of possible decisions.

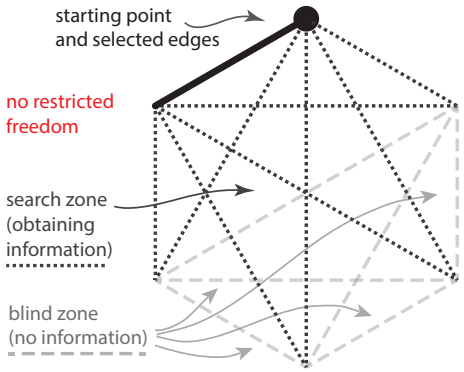
It is obvious that, at each further step, the Prim's algorithm: a) has fewer possible (we call them *feasible*) solutions left, and b) it explores more and more detailed information about these remaining solutions. Since "a picture is worth a thousand words", let's refer to Figure 1.2 to illustrate this concept. The figure brings an artificial example of four consecutive steps of Prim's MST algorithm applied to a 6-vertex weighted complete graph (weights are not shown) and pays attention to three "zones" of edges: a "restricted freedom" zone (edges which will never be considered in the solution), a "search zone" (currently considered edges), and a "blind zone" (edges which are not yet considered and thus their information does not contribute at the current step). One can see the following:

- On step 2, the algorithm still expects all edges to be feasible, but searches through only a fraction of them. The blind zone is large, which yields definite lack of information.
- On step 3, the algorithm starts to create some prohibited edges, since no cycles are allowed in a solution. The blind zone is reducing.

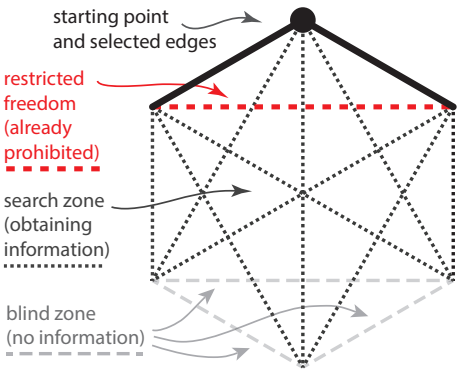
Algorithm 1.1: Prim's Algorithm for finding Minimum Spanning Tree

Input: undirected graph with non-negative weights
Output: spanning tree with minimum total weight

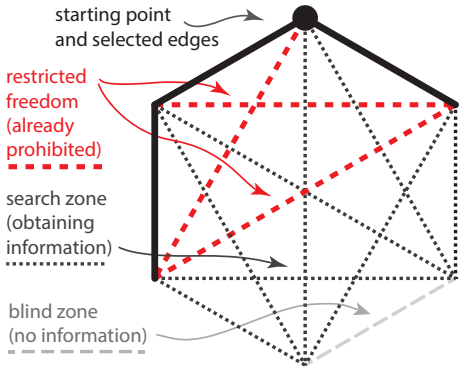
```
initialize current tree by choosing the first vertex randomly;
while not all vertices are in tree do
    find the minimal edge from current tree to the rest;
    add it to the tree;
return current tree
```



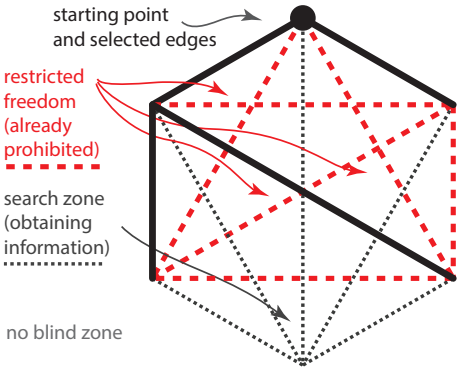
(a) Step 2: blind zone is large, no restricted zone yet



(b) Step 3: blind zone gets smaller, restricted zone is small



(c) Step 4: blind zone gets even smaller, restricted zone is growing



(d) Step 5: no blind zone vanishes, restricted zone is large

Figure 1.2 Prim's MST Algorithm: freedom reduces as more of the graph is explored. Edge weights are not shown.

- On step 4, the algorithm continues creating prohibited edges. The blind zone is reduced to one edge, but still exists.
- On step 5, the algorithm has created lots of prohibited edges, and has no blind zones: it has explored all of the graph by now.

To build a connection to the previous fictional example of the real-world decision making, we refer to Table 1.1. The above property of "increasing information and reducing freedom" is characteristic to many algorithms and, more generally, to systems where the notion of "being optimal" is defined.

But why are we talking about that? Because having all the possible information sometimes does not help to select the best solution, as well as having all the

Pilot in trouble (Figure 1.1)	Prim's MST (Figure 1.2)
— Pilot sees no small details of the terrain and still can literally fly wherever he wants.	— On step 2, there do not yet exist any restrictions regarding which edges can be included into the solution, but searches through only a fraction of them. Blind zone is large, which yields definite lack of information.
— As he descends, the pilot leaves himself less and less freedom of choice, but sees more and more of the details, e.g. large power-lines.	— On step 3, the algorithm starts to create some prohibited edges, since no cycles are allowed. Blind zone is reducing, which reflects the fact, that algorithm explores more and more.
— As he descends further, the pilot continues to lose the freedom of choice, which is yet not fully lost. He continues obtaining more and more details.	— On step 4, the algorithm continues creating prohibited edges. Blind zone is reduced to one edge, but still exists.
— Finally, the pilot has no chance to divert a lot, having to land, but he now has a lot of information about the terrain, e.g. small creeks, trees and terrain roughness.	— On step 5, the algorithm has created lots of prohibited edges, and has no blind zones: it basically explored all of the graph by now.

Table 1.1 An analogy between real and algorithmic worlds.

freedom to choose solutions sometimes fails — there exists an *informativeness vs. robustness trade-off*. We revisit this statement in the next section.

1.2 Overly Informed Decisions Are Bad

Information can be evil under some circumstances. To see that, revisit the case of Prim's MST Algorithm (Figure 1.2 and Algorithm 1.1): one can easily prove that despite all the fancy reasoning above, Prim's algorithm always *finds the optimal tree*. Why are then all the explanations of the previous section important?

The answer is that this optimal tree is only optimal in current instance of graph weights, which most likely contain *noise*. In fact, all the real-world observations are contaminated by noise, making it impossible to judge truth without uncertainty.

Pilot in trouble (Figure 1.1)	Prim's MST (Figure 1.2)
— At each altitude, the pilot sees not all the details because there are limitations to his vision (or visibility).	— At each step, the algorithm observes noised weights of edges, i.e. some randomness exists.
— As a consequence, certain decisions can eliminate possibilities to land on fields which seem bad from current altitude but in fact are the best.	— As a consequence, certain decisions can eliminate edges which seem bad in the current instance, but are very good on the average.

Table 1.2 Noise in real and algorithmic worlds.

In terms of our two examples, a very informal Table 1.2 explains the analogy. In fact, it turns out, that in presence of uncertainty, overly informative environments can deceive the algorithm and lure it into a solution, which is, although optimal in current instance of noise, still far from being optimal in some ground truth, noise-free instance.

Intuitive (we will come to that in a more formal Chapter 3) reasoning for this phenomena of overfitting is as follows: an agent (pilot, algorithm) uses information contained in the environment to solve the task. While this information contains a useful component, it keeps a bit[1] of noise-induced "garbage". Filtering the latter out is thus the key to decent performance. There is a group of methods called *regularization* to fulfill this task.

1.3 Regularization by Stochastic Approximation

While there are lots and lots of ways to regularize solutions to various optimization problems, we will start the story of this thesis from a point which we already touched above: approximating solutions. This concept will be central to the first three chapters of the thesis, leading to surprising conjectures in the last chapter.

But first things first: informally speaking, stochastic approximation allows us to avoid optimizing the given objective till the global minimum — remember from above that this global minimum might be not the best one in expectation! — but instead *stops* optimizing at some moment before the end, leaving us some room near the optimum to choose from. In most cases, one will then choose randomly from a set of these near-optimal solutions, called an *approximation set*.

Remark. Revisiting the above example of our pilot for the last time (and

[1] "Bit" here is not the Shannon's information-theoretic bit yet. We come to it later.

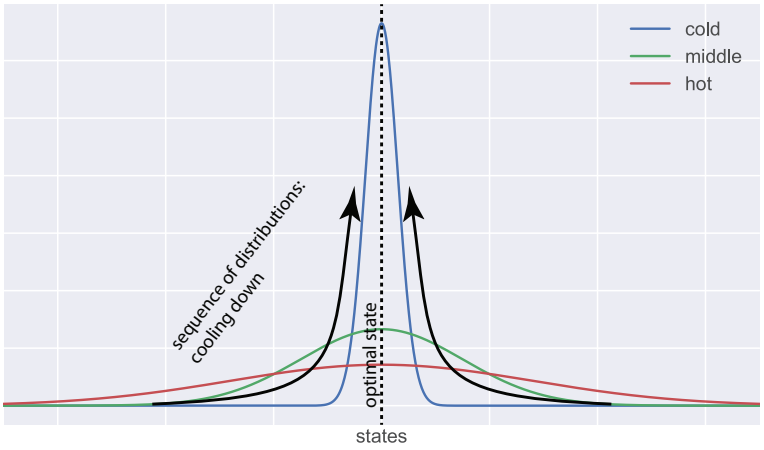


Figure 1.3 Sequence of Gibbs distributions as temperature decreases ("cooling down")

let him land safely), this means: make him descend to some predefined altitude $h > 0$ (it can be low, but not zero), and then stop his (no longer reliable) cognitive process, choose a field randomly from the *approximation set* of nearby fields and commit to landing on that one.

We now would like to make a small side-leap and connect the above approximation intuition with the field of statistical physics. An important observation is to note that Mother Nature essentially does the same stochastic approximation when optimizing energy of states in large particle systems. This phenomenon is largely studied in, for example, statistical mechanics. In fact, real systems never find the global optimum of energy, but sample their states from some distribution (sometimes called *Gibbs distribution*), which *concentrates* around optimal states, but assigns some probability to near-optimal states.

In fact, what we call *temperature* is exactly the parameter which controls how closely the sampled states approximate the optimal state: an H_2O molecule of water steam at 120 C° has more freedom to move around than a molecule of water liquid at 40 C°, which in turn still has more freedom than a molecule of frozen ice at -40 C°. This concept is illustrated by Figure 1.3, in which case temperature is the "width" of distribution.

While the idea of approximate solutions is, of course, not new, the cornerstone of the thesis is to offer a novel answer to the question on *how to choose* the level of approximation, at which one should sample to obtain robust solutions. Throughout the thesis, we will formulate this question and answer it from several

prospectives, which we explain in the next section.

1.4 Thesis Contributions and Outline

Precise objectives and contributions are stated separately in each chapter. Here, we describe them in a survey manner.

- **(Chapter 3)** To start, we revisit an approximate set-based approach to robust solving, called *Approximation Set Coding*. We provide its theoretical background. We then introduce and experimentally evaluate an abstract, but still interpretable proof-of-concept model, and show the superiority of the approximation set-based approach in terms of error. However, this approach involves a computationally expensive step. To address it, we prove a theoretical result which partially solves the computational bottleneck problem. We provide experiments for that as well.

- **(Chapter 4)** As an expansion of the above approach, we adapt and apply it in algorithmic setting, specifically for algorithms which solve the Minimum Spanning Tree problem. We show that the approach allows us not only to robustly solve such problems, but also ranks various algorithms according to their robustness. Finally, our algorithmic adaptation is free of the computational bottleneck exhibited in the general setting.

- **(Chapter 5)** Further, we study a thermodynamical Gibbs relaxation of the approximation set-based approach in combinatorial optimization setting and discover its deep connection with a prominent task in statistical mechanics, namely the task of computing the *free energy density* (for definition and details, refer to the respective chapter). The contribution is thus twofold: first, we devise and apply the Gibbs relaxation of approximation set-based approach, and second, we prove a mathematical result associated with it, which has its own importance.

- **(Chapter 6)** In the last chapter, we drift away from the approximating solutions. Inspired by the theoretical results of the previous chapter, we ask fundamental questions about statistical mechanics of combinatorial optimization and how it defines our ability to efficiently solve them. We pick a combinatorial optimization problem and compare its *large system size* behavior with the one of well-known Random Energy Model (REM), leading to some interesting conjectures about search complexity.

Final notes and possible directions of further work are discussed in Chapter 7.

1.5 Statement of Publications and Joint Contributions

Publications

The following publications appeared while working on this thesis, or were under review by the time of completing this thesis. I was either the main contributor or jointly contributing with my co-authors:

- (Gronskiy and Buhmann, 2014)
  Gronskiy, A., Buhmann, J. M., 2014. How informative are minimum spanning tree algorithms? In: 2014 IEEE International Symposium on Information Theory (ISIT) 2014;

- (Buhmann et al., 2014)
  Buhmann, J. M., **Gronskiy, A.**, Szpankowski, W., 2014. Free energy rates for a class of very noisy optimization problems. In: Analysis of Algorithms (AofA) 2014;

- (Buhmann et al., 2017b)
  Buhmann, J. M., Dumazert, J., **Gronskiy, A.**, Szpankowski, W., 2017b. Phase transitions in parameter rich optimization problems. In: Proceedings of the Fourteenth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2017;

- (Buhmann et al., 2017a)
  Buhmann, J., **Gronskiy, A.**, Mihalák, M., Pröger, T., Šrámek, R., Widmayer, P., 2017. Robust optimization in the presence of uncertainty: A generic approach. Journal of Computer and System Sciences.

- (under review)
  Gronskiy, A., Buhmann, J. M., Szpankowski, W., 2018. Free Energy Asymptotics for Problems with Weak Solution Dependencies. In: 2018 IEEE International Symposium on Information Theory (ISIT) 2018;

- (under review)
  Buhmann, J. M., Dumazert, J., **Gronskiy, A.**, Szpankowski, W., 2018. Posterior Agreement for Large Parameter-Rich Optimization Problems. In: Journal of Theoretical Computer Science.

The following publication is aligned with the topic of the thesis, however I did not contribute to it significantly:

- (Bian et al., 2016)
  Bian, Y., **Gronskiy, A.**, Buhmann, J. M., 2016. Information-theoretic analysis of max-cut algorithms. In: 2016 Information Theory and Applications Workshop, (ITW), 2016.

The following Master's theses were completed under my advisorship:

- Dumazert, J., 2015. Free Energy for a Class of Combinatorial Optimization Problems and its Asymptotics. ETH Zurich;
- Negulescu, G., 2015. Theoretical and Numerical Analysis of an Information-Theoretic Approach to Feature Selection. ETH Zurich;
- Gence, E., 2017. Electrocardiogram Feature Extraction for Takotsubo Syndrome Detection. ETH Zurich (I acted as co-advisor).

Joint Contributions

The ideas in this thesis were mostly developed by me, but some of them were developed jointly with collaborators. Here, I identify major collaborations and contributions which are not due to me. For detailed statements of contributions and notes on presentation, see introductions to the respective chapters (Sections 3.1.2, 4.1.2, 5.1.3 and 6.1.2).

- **(Chapter 3)** The ASC approach (Buhmann, 2010a) and Similarity approach (Šrámek, 2013) served as a starting point for this thesis, explained in Sections 3.4 and 3.5 respectively. We devised the proof-of-concept model of Section 3.6 jointly with T. Pröger, but he implemented the vast majority of the experiments. Theorem 3.5 is my contribution.
- **(Chapter 4)** Contributions, design and implementation are fully due to me.
- **(Chapter 5)** This was a joint work with W. Szpankowski and, partially, with J. Dumazert. Contributions are mainly due to me. The main idea behind proof of Theorem 5.3 is due to J. Dumazert.
- **(Chapter 6)** Contributions are fully due to me.

In general, I appreciate all the inputs I have ever obtained. It is hard to extract which specific bit comes from whom, because most of theoretical work was accompanied by lots of discussions, e-mail exchanges, marker-and-whiteboard as well as chalk-and-blackboard sessions or coffee brainstorming with lots of people to whom I am incredibly grateful.

2
Background

"Мы все учились понемногу
Чему-нибудь и как-нибудь,
Так воспитаньем, слава богу,
У нас немудрено блеснуть."

(rus. "We all meandered through our schooling
haphazard; so, to God be thanks,
it's easy, without too much fooling,
to pass for cultured in our ranks.")

— Alexander PUSHKIN, "Eugene Onegin"

In this chapter, we gather some standard background assumptions, definitions, theorems, tools and approaches to which we refer throughout the thesis.

Remark (On the notation). Although the thesis involves a broad range of topics and fields (coding, combinatorics, learning, algorithms and statistical mechanics), we made an effort to unify the notation and keep it consistent *across these fields*. While it is not always fully possible, the reader can assume that from now on he/she will be exposed to this unified notation. This is done on purpose to allow a simpler grasping of analogies between the same notion in different contexts. At the cost of this, sometimes the notation might seem to be unconventional w.r.t. each specific topic.

2.1 Probability Theory

It is hard to find a field nowadays where probability theory is *not* used. Being no exception, we are going to utilize the standard axiomatization of probability theory, developed by A.N. Kolmogorov. While we refer the reader (in case of

interest) to Shiryaev (1995) for details, here we list just basic information and our assumptions to keep the notation of the thesis self-contained.

Due to the discrete nature of the problems considered in this thesis, we will mostly work with discrete distributions (except for data generation sources, which will be e.g. Gaussians), hence almost all the theorems and derivations will be stated using $\sum$ sign.

For brevity, we will omit subscripts of expectations $\mathbb{E}[\cdot]$, variances $\mathrm{Var}[\cdot]$ and covariances $\mathrm{Cov}(\cdot,\cdot)$, except for the cases where there are several randomness sources and we need to distinguish between them.

We will utilize the term *moment-generating function* defined as follows.

Definition 2.1 (Moment-generating function). *For a random variable X, we call a function*

$$G_X(t) \coloneqq \mathbb{E}[e^{tX}], \quad t \in \mathbb{R} \tag{2.1}$$

a moment-generating function of X.

Moment-generating functions might not exist, and in some derivations we will pay particular attention to that.

2.2 Information Theory

In this thesis, we are going to utilize some terminology of information theory. While one of the best books on the topic remains to be (Cover and Thomas, 2006), we give here some basic definitions and theorems.

2.2.1 Bits, Nats and the Usage of Logarithm

It is common to use logarithms to base 2 in information theory since they yield an easily interpretable measure of information called *bit*, which quantifies information in *binary outcomes*.

However, one can resort, without loss of generality, to utilizing the natural logarithm: in this case all the results of information theory hold, however, the information measure is changed to *nats*, which, informally speaking, quantifies information in *base-e outcomes*.

Generally, the base-e approach is characteristic for problems with a dominant physical or general mathematical viewpoint, while the base 2 approach is reserved by information theory.

With a slight abuse of notation (but w.l.o.g.), we will utilize same notation $\log(\cdot)$ for both binary and natural logarithm, and it is made clear from the context, which of them is currently used. For a very general reference, a binary logarithm

is mainly used in Chapters 3 and 4, and a natural logarithm is used in Chapters 5 and 6, since the latter ones are more inclined into statistical mechanics.

2.2.2 Entropy and KL-Divergence

One of the central points of this thesis is *uncertainty*. This concept obtained its rigorous definition[1] through the notion of *entropy*.

Definition 2.2 (Shannon entropy)**.** *Assume a discrete random variable X is distributed according to $p(x)$. Then, the Shannon entropy of X is defined as*

$$H(X) \equiv H(p) \coloneqq -\sum_x p(x) \log p(x). \tag{2.2}$$

Shannon entropy quantifies the average *surprise of the outcome* $h(x)$:

$$H(X) \equiv \mathbb{E}[h(x)] \equiv \mathbb{E}[-\log p(x)]. \tag{2.3}$$

The term *surprise* refers here to self-contained information contained in the outcome, hence the entropy refers to the average self-contained information. The higher is the entropy, the higher is the average self-contained information of outcomes, hence the higher is uncertainty of the distribution. A classical example is a coin toss experiment, where the heads probability euqals p. Consider two characteristic cases:

a) for a highly unfair coin (i.e. $p = 0$ or $p = 1$), the outcome of each toss does not bring much information ("surprise") as it is known beforehand. The entropy is the lowest (zero bits) in this case;

b) for a fair coin (i.e. $p = 1/2$), the outcome brings a lot of information, as we cannot have any expectations about it beforehand. The entropy is the highest (one bit) in this case.

The joint entropy of two random variables X and Y is defined similarly:

$$H(X, Y) \coloneqq -\sum_{x,y} p(x, y) \log p(x, y). \tag{2.4}$$

Next, we will use the term *Kullback-Leibler divergence* (KL-divergence).

Definition 2.3 (KL-divergence)**.** *Assuming two discrete distributions $p(x)$ and $q(x)$,*

[1] This very phrase "rigorous definition of uncertainty" may already surprise!

a relative entropy, or Kullback-Leibler (KL) divergence is defined as

$$\mathcal{D}^{KL}(p\|q) \coloneqq \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right). \tag{2.5}$$

KL-divergence is one of the possible measures of "distance" between distributions, and it quantifies how much additional bits (nats) of information is required for stable communication, if the receiver assumes source distribution q instead of the true source distribution p.

The next important quantity is *mutual information*.

Definition 2.4 (Mutual information)**.** *Assuming two random variables X and Y, the mutual information between them is defined as*

$$\mathrm{MI}(X,Y) \coloneqq \mathcal{D}^{KL}(p(x,y)\|p(x)p(y)) = \sum_{x,y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right). \tag{2.6}$$

Mutual information is the information-theoretic measure of independence. In other words, one can show that $\mathrm{MI}(X,Y) = 0 \Leftrightarrow X \perp\!\!\!\perp Y$ (symbol "$\perp\!\!\!\perp$" denotes stochastic independence).

We will also utilize the fact that

$$\mathrm{MI}(X,Y) = H(X) + H(Y) - H(X,Y), \tag{2.7}$$

which can be easily observed.

2.2.3 Coding Theory, Channels and Code Rates

Another concept of information theory, whose importance is exceptionally high for our purposes, is *coding theory*. For a detailed explanation of the following definitions we refer the reader to (Cover and Thomas, 2006, Chapter 7), while here we give, as usual, only necessary basics.

Definition 2.5 (Channel)**.** *A channel is defined by an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$, an input distribution $p(x)$ as well as a transition probability matrix $p(y|x)$. We denote such channel as $(\mathcal{X}, p(y|x), \mathcal{Y})$. A message X sent through it turns into a message Y received according to the transition matrix $p(y|x)$.*

Definition 2.6 (Memoryless channel). *A channel is called memoryless if the probability distribution of the output depends only on the input at that time and is conditionally independent of previous channel inputs or outputs.*

Definition 2.7 (Code). *An (M, n) code for channel $(\mathcal{X}, p(y|x), \mathcal{Y})$ consists of*

- *an index set $\{1, \dots, M\}$;*
- *an encoding rule $f^n \colon \{1, \dots, M\} \to \mathcal{X}^n$, where each $f^n(i)$ is called codebook vector. Set $\{f^n(i)\}_{i=1}^M$ is called codebook.*
- *a deterministic decoding rule $g \colon \mathcal{Y}^n \to \{1, \dots, M\}$.*

This definition will become highly important for understanding of Chapter 3, in conjunction with the definition of channel capacity and code rate.

Definition 2.8 (Channel capacity). *The channel capacity of the channel $(\mathcal{X}, p(y|x), \mathcal{Y})$ is defined as*

$$C \coloneqq \max_{p(x)} \mathrm{MI}(X, Y). \tag{2.8}$$

Definition 2.9 (Code rate). *The rate R_{code} of an (M, n) code is defined as*

$$R_{\mathrm{code}} \coloneqq \frac{\log M}{n} \tag{2.9}$$

bits (nats) per transmission.

The rate is called *achievable*, if there exists a sequence of $(2^{nR_{\mathrm{code}}}, n)$ codes such that the maximal probability of decoding error goes to zero.

Finally, it is important to realize that the capacity C is the supremum of all achievable rates, according to Shannon's Channel Coding Theorem which will be stated later. This fact establishes an important connection between the mutual information and the bitrate of the channel.

2.3 Disordered Systems and Statistical Mechanics

Large part of the thesis works with the terminology of statistical mechanics. A good overview is given in (Bovier, 2012), and here we summarize the basics specific to our work.

2.3.1 Disordered Systems

Statistical mechanics, which will play a major role in the second half of this thesis (Chapters 5 and 6), focuses on systems consisting of a large number of *particles*, which can find themselves in certain aggregated *states*, or *configurations*.

As an example, let us consider a so-called n-spin system. It is represented by some physical matter which consists of n particles (atoms) s_i, each of which exhibits one of the two possible directions of the magnetic moment vector of the atom, called *spin*

$$\sigma_i \in \mathcal{S} \equiv \{-1, +1\} \tag{2.10}$$

(i.e. atom's magnetic field is directed "up" or "down"). In this system, a configuration $c_{\boldsymbol{\sigma}} \in \mathcal{C}$ is defined as an ensemble of spins of its particles, i.e. a vector

$$c_{\boldsymbol{\sigma}} = \langle \sigma_1, \ldots, \sigma_n \rangle. \tag{2.11}$$

An object used in conjunction with a configuration is its *Hamiltonian*, or *energy*:

$$R(c_{\boldsymbol{\sigma}}) = -\sum_{i,j} J_{ij} \sigma_i \sigma_j, \tag{2.12}$$

where J_{ij} is a predefined interaction value between spin i and spin j[2].

However, there is no randomness yet in this system, which would bring the *disorder*. To introduce it, assume that there is a random variable $X \in \mathcal{X}$ which is distributed according to a certain law. In general case, its nature should not necessarily be aligned with the nature of other model elements (particles, configurations), i.e. from a model design point of view, X is a source of randomness and anything more than that. If we now assume that the interaction matrix J_{ij} depends on it, i.e. $J_{ij} = J_{ij}(X)$, the Hamiltonian becomes random:

$$R(c_{\boldsymbol{\sigma}}, X) = -\sum_{i,j} J_{ij}(X)\, \sigma_i \sigma_j. \tag{2.13}$$

The described system which consists of

a) a source of randomness $X \in \mathcal{X}$, and

b) a set $\mathcal{C}$ of configurations[3] $c_{\boldsymbol{\sigma}} \in \mathcal{C}$, and

[2]Other Hamiltonians are possible (e.g. the ones with external field, etc.) which depends on the model; describing them is beyond the scope of the thesis. A very good overview is given in (Bovier, 2012).

[3]In the main part of the thesis, we will call them "solutions", highlighting an optimization prospective.

c) a random Hamiltonian[4] $R(c_{\boldsymbol{\sigma}}, X)$

defines a *disordered system*. Its nature is very closely related to the nature of randomized optimization, and as we will see already in Chapter 3, the ingredients of the latter are the same.

2.3.2 Gibbs Measures

The Gibbs measures are an important class of measures which arise naturally in conjunction with statistical mechanics of disordered systems. It is defined as follows.

Definition 2.10. *The Gibbs measure over the configurations is the distribution*

$$p_\beta(c|X) = \frac{1}{Z(\beta, X)} \exp(-\beta R(c, X)), \tag{2.14}$$

where the normalization term

$$Z(\beta, X) = \sum_{c' \in \mathcal{C}} \exp(-\beta R(c', X)) \tag{2.15}$$

is called partition function.

Remark. We remind the reader that the above quantities are random due to their the parametrization by random source X. If it is dropped, then the Hamiltonians turn from random into deterministic ones (see (2.12) above).

2.3.3 Thermodynamic Properties of Disordered Systems

Large disordered systems are characterized by macroscopic thermodynamic properties. One of the most important for us will be *Helmholtz free energy*. While we discuss its altered (adjusted to our needs) version in detail later in Chapters 5 and 6, here we state its classical definition and provide motivation for it.

Definition 2.11. *Helmholtz free energy is originally defined as*

$$\mathcal{F}(\beta, X) = -\frac{1}{\beta} \log[Z(\beta, X)] \tag{2.16}$$

Due to parametrization by a random source X, Helmholtz free energy is itself a random variable. In this situation, studying its moments (e.g. the expected

[4] In the main part of the thesis, we will call it a "cost", highlighting an optimization prospective.

value) becomes a central point (Bovier, 2012, Chapter 9). The same will hold for our case, as explained in much more detail in Chapter 5 where we will define

$$\mathcal{F}(\beta) = \text{const} \cdot \mathbb{E}_X \mathcal{F}(\beta, X). \tag{2.17}$$

The constant is not important for now.

The importance of the log-partition function is hard to overestimate: it implicitly contains many other parameters of the system. For example, Shannon entropy (see Definition 2.2) of the Gibbs measure can be expressed through it:

$$\begin{aligned} H(p_\beta) &= -\sum_{c \in \mathcal{C}} p_\beta(c) \log p_\beta(c) \\ &= \log Z(\beta) - \mathbb{E}_{p_\beta(c)}[-\beta R(c)] \\ &= \log Z(\beta) - \sum_{c \in \mathcal{C}} \frac{-\beta R(c) e^{-\beta R(c)}}{Z(\beta)} \\ &= \log Z(\beta) - \beta \frac{\partial}{\partial \beta} \log Z(\beta). \end{aligned} \tag{2.18}$$

2.3.4 Maximum Entropy Principle

In the following, we will refer to Jaynes' *maximum entropy* principle (Jaynes, 1957a,b). This principle states, informally speaking, that the current state of a physical system is represented by a distribution which has the highest possible entropy under the given constraints. This concept is related to a commonly used Occam's razor principle which states that everything should be explained in the simplest possible way given the observed constraints.

More formally speaking, under the observed data X, the distribution which is followed by a physical system is the following:

$$\widehat{p}(c) = \arg \max_{p\colon \operatorname{obs}(p) = X} H(p), \tag{2.19}$$

where $\operatorname{obs}(p)$ is the function which yields observed data (also-called *testable information*; for example, it extracts moments of the distribution — and thus in some sense "tests" it).

We will largely utilize the fact that the maximum entropy distribution under certain constraints is the Gibbs distribution:

$$p_\beta(c|X) = \frac{1}{Z(\beta, X)} \exp(-\beta R(c, X)) \quad \text{with}$$

$$Z(\beta, X) = \sum_{c' \in \mathcal{C}} \exp(-\beta R(c', X)) \,, \tag{2.20}$$

where $R(c, X)$ is determined explicitly by the context, and β comes from the method of Lagrange multipliers when optimizing (2.19) and has a semantics of an *inverse temperature*.

3

Approximation-Based Regularization for Robust Optimization

"Truth is much too complicated to allow anything but approximations."

— John von NEUMANN, "The Mathematician"

3.1 Introduction

3.1.1 Motivation

Within a given data set, not all information is useful — given measurement error, some part of it explains noise, not the true functional dependencies. Hence, there exists an inevitable limitation on the amount of information bits one should use in order to avoid overfitting. However, another curse — underfitting — happens if for some reason the solver decides to play on the safe side and uses less information than would be optimal. We refer to this phenomenon as the *informativeness vs. robustness trade-off*.

In the line of research started by Buhmann (2010a, 2011), an approach of utilizing self-calibrating[1] optimization procedures is devised and advocated. In its essential part, this approach aims at maximizing (through a set of tools discussed below) the amount of useful information, thereby making solutions both statistically robust *and* informative.

This approach features (cf. Busetto, 2012; Buhmann et al., 2017a), among others, the following key properties:

- it does not require any knowledge on probability distributions of input instances, particularly not whether the noise is systematic or random;

[1] Sometimes "self-calibrating" is replaced by "context-sensitive".

- it allows to quantify the quality of the obtained solution w.r.t. unseen data instances;
- moreover, it makes possible to rank different models.

In this chapter, we will provide justification of this approach, as well as introduce some prototypic examples proving its usability. We will also discuss possible generalizations and adjustments of this approach, which will be addressed in the next chapters.

3.1.2 Contributions and Outline of the Chapter

As main contributions of this chapter, we

- provide a justification and revisit the approach for robust optimization called Approximation Set Coding;
- introduce and evaluate the simplest, yet interpretable proof-of-concept model which shows experimentally the superiority of the ASC-based approach and refers to clear intuitions about the mechanism thereof;
- prove a theoretical result which suggests one step further in the direction of eliminating the computational bottleneck of the ASC — computing the ASC score;
- introduce a so-called Gibbs relaxation of the ASC approach, give a rationale behind it and experimentally evaluate its performance.

The chapter is outlined as follows. First, a background and related work are given in Section 3.2. We then provide technical preliminaries (setting and model assumptions) in Section 3.3. A comprehensive introduction into the original approximation set-based approach is then given in Section 3.4. Section 3.5 presents an analogical approach to robust optimization. We then give a proof-of-concept experimental confirmation of the validity of approximation set-based methods in Section 3.6. Further, we address one problem which is very characteristic bottleneck for the most of applications of approximation set-based approaches and solve it in Section 3.7. We then explain how a relaxation of the approach (called Gibbs relaxation) works in Section 3.8 and show experimental results for it. Finally, concluding remarks follow in Section 3.9.

Remark. Section 3.5 is included into the thesis for the sake of "self-containedness", and presents an approach not due to the author of this thesis. For smooth integration into this thesis, we provided necessary terminological and

notational adaptation, but nevertheless, some small parts of the text of Section 3.5, as well as Section 3.9 may still be similar to that of Šrámek (2013). Sections 3.6, 3.7 and 3.8 are a result of joint work hence their textual presentation may be partially similar to that of Buhmann et al. (2017a).

3.2 Background and Related Work Overview

When dealing with uncertain (noisy) inputs, the model designer always confronts with two related questions:

- For a given model, how to provide a well-generalizing regularization?
- For a predefined set of models, how to establish an ordering of them which would reflect their generalization capability?

While we introduce an approach which solves both tasks, in this chapter we will mainly concentrate on the first application of it, while the next chapter (Chapter 4) addresses the second one. For now, however, we summarize an overview of both tasks.

3.2.1 Generalization and Stability in Learning

When dealing with noisy inputs, a designer of an empirical risk minimization (ERM) algorithm is always confronting with how well the learned solution generalizes to the unknown test sets. In fact, the whole field of statistical learning theory (Vapnik and Chervonenkis, 1971; Vapnik, 1982) has been in its core posing this questions since 70s of the last century. It focuses on the question: for a given algorithm, can we derive bounds of its generalization error (Bishop, 2006)?

The ways of bounding generalization error can be, in our view, split into three[2] classes: to a more classical one belongs, e.g. bounding generalization error via considering properties of *hypothesis space* such as VC-dimension (Vapnik and Chervonenkis, 1971; Vapnik, 1982) or Rademacher complexity (Shalev-Shwartz and Ben-David, 2014).

Another class of research directions encompasses approaches where one derives such bounds using the so-called (hypothesis, error or uniform) stability (Devroye and Wagner, 1979; Bousquet and Elisseeff, 2002) property of the ERM algorithm, which, in essence, reflects how critical are the fluctuations of the input data for the outcome of an algorithm. As a side remark, we can note that from the technical

[2]These classes are very much interrelated. We brought such classification for simplicity and don't pretend it to be the only division possible.

standpoint, the mentioned bounds utilized various concentration inequalities (Raginsky and Sason, 2015).

Lastly, stability (and thus generalization) properties have recently enjoyed research from the information theoretic prospective, considering a learning algorithm as a channel from the input to output (Russo and Zou, 2015; Xu and Raginsky, 2017) and relating stability to the mutual information between the input and the output. To the advantages of this third class of approaches belongs the fact, that the bounds provided by it, involve *both* the properties of the hypothesis space and the learning algorithm (as opposed to the aforementioned methods). To the same "information theory-inspired" class can we assign a recent work by Alabdulmohsin (2015) which relates generalization to a total variation information.

Remark. The approach via input-output mutual information comes very close to the one introduced and advocated in this chapter. However, we should note that both approaches stem from different definitions of the communication channel used to derive error bounds.

Remark. It should be noted here that all the above methods only provide ways to guarantee certain performance when the learning algorithm is fixed. They do not answer the question how to regularize its solutions for a more robust performance. In contrast, the approach presented and tested in this chapter does exactly this.

3.2.2 Model Selection

Besides quantifying the quality of a given ERM solution (overview for which was given above), a modeler can ask another question: how to choose between two possible models, taking into account various properties such as e.g. complexity? A long line research which addressed this question is presented by methods such as the Minimum Description Length (MDL) principle (Rissanen, 1978), the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC) or the Generalized Information Criterion (for an overview, see Konishi and Kitagawa, 2007).

3.2.3 Robust Optimization

Besides the approaches characteristic for statistical learning theory, there exists a methodological direction called *robust optimization*. Being very close to the approaches above, robust optimization deals with models for the uncertain input — however, contrary to the approaches traditional for statistical learning theory, in the field of robust optimization, it is explicitly discouraged to assume the knowledge of the input data distribution, although some information (for example, if the

data comes form certain interval domain or not) might be available. For a comprehensive overview of robust optimization approaches, we recommend a recent survey by Goerigk and Schöbel (2016).

The closest point of contact between the robust optimization approaches and approaches of the above Section 3.2.1 is, in out view, *optimization for stable inputs* which makes an attempt to understand the connection between fluctuations of the input and the output, i.e. some sort of stability (Bilu and Linial, 2012; Bilò et al., 2009; Gatto and Widmayer, 2011; Mihalák et al., 2011).

3.3 Setting and Generative Model Assumptions

3.3.1 Optimization Problem

In the setting we are going to analyze in this and the next chapters, the following components are assumed to be defined:

- A set $\mathcal{X}$ of possible *data instances* X:

$$\mathcal{X} \ni X, \tag{3.1}$$

  on which no further assumptions (e.g. structure, finiteness, countability) are imposed in the most general case (see below in Section 3.3.2 for possible specifications of such assumptions).

- A set $\mathcal{C}$ of possible *solutions*, or *hypotheses* c:

$$\mathcal{C} \ni c, \tag{3.2}$$

  where again no further structural or finiteness assumptions are imposed.

- An *objective function* $R(c, X)$ representing the value of a given solution solution c for a data instance X:

$$R(c, X) \colon \mathcal{C} \times \mathcal{X} \to \mathbb{R}. \tag{3.3}$$

  If not stated otherwise, we will assume a minimization (i.e. "cost", "error" or "energy") semantics of $R(c, X)$ and call it a *cost function*.

Definition 3.1. *Provided that a solution feasibility assumption is fulfilled, i.e. any solution $c \in \mathcal{C}$ is feasible (i.e. valid) for any data instance $X \in \mathcal{X}$, then we can say that these three components define a valid optimization problem denoted by a triplet $\mathcal{P} = (\mathcal{X}, \mathcal{C}, R)$.*

The optimization goal consists in finding the set of those solutions which minimize the cost function:

$$\mathcal{C}^{\perp}(X) \coloneqq \arg\min_{c\in\mathcal{C}} R(c, X). \tag{3.4}$$

With a bit of notation abuse we will also write

$$c^{\perp}(X) \coloneqq \arg\min_{c\in\mathcal{C}} R(c, X) \in \mathcal{C}^{\perp}(X), \tag{3.5}$$

meaning a *one* (out of many possible) optimal solution (global minimizer). We will denote the optimal cost as:

$$R^{\perp}(X) \coloneqq \min_{c\in\mathcal{C}} R(c, X). \tag{3.6}$$

3.3.2 Data Generation Model

Dealing with *uncertainty* in optimization requires to define a data generation process.

In the following, we will simply assume that there is exists *true (ground, signal) data instance* X^0, from which the *noise-contaminated data instances* are obtained independently, i.e.:

$$X', X'', X''', \cdots \sim PG(X|X^0) \tag{3.7}$$

through a *problem generating* process $PG(\cdot|X^0)$. Note that the problem generating process is parametrized through the ground truth X^0. Note also that the obtained data instances are independent, conditioned on the X^0:

$$\text{for any data instances } X', X'' \sim PG(X|X^0): \quad X' \perp\!\!\!\perp X''|X^0. \tag{3.8}$$

Remark. Although this notation might seem complex, it is actually very straightforward. In most cases we consider, $X \in \mathcal{X}$ will be just a vector of random (generated by $PG(\cdot)$) weights from which the costs $R(c, X)$ are constructed.

3.4 Approximation Set-Based Approach

In this section, we will introduce the notions related to Approximation Set Coding framework — a successful way of regularizing solutions to cost-driven optimization

problems.

3.4.1 Approximation Sets

We introduce the notion of *approximation sets*, which are intended to address the question: how to avoid the risk of overfitting in those frequent cases, when the solver is not aware of precise noise conditions $PG(\cdot)$ imposed on the dataset?

Consider the following thought experiment: datasets $X', X'', \ldots$ are drawn according to the random data generation process $PG(\cdot|X^0)$ as given in Section 3.3.2. As all the datasets stem from the same "ground" dataset X^0 (in some sense, which we leave undefined for the sake of keeping things simple for now), they contain both useful and irrelevant information. In other words: only *some* information the one obtains from the dataset "explains signal" X^0 (e.g. has low condition entropy), while the rest of the information "explains noise".

Utilizing an optimization model defined by cost $R(\cdot,\cdot)$, as in (3.3) and thus obtaining optimal solutions $c^\perp(X'), c^\perp(X''), \ldots$, we inevitably absorb both useful and irrelevant information and overfit, making solutions unstable w.r.t. each other. To regularize the optimization process, one might want to relax the optimal solution by including all the solutions located in the vicinity (in some topology we define in a second) of it. A natural way to define such topology is to utilize the level surfaces of the cost function $R(\cdot,\cdot)$ itself! The method proposed by Buhmann (2010a) suggests the following definition of the approximation set.

Definition 3.2 (Buhmann (2010a)). *For a given real number $\gamma \geq 0$, an approximation set is defined as follows:*

$$\mathcal{C}_\gamma(X, R) \coloneqq \{c \in \mathcal{C} \mid R(c, X) - R^\perp(X) \leq \gamma\}, \tag{3.9}$$

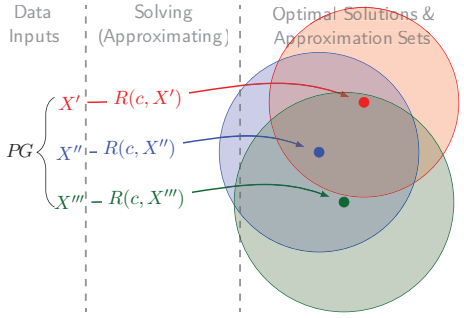
and the solutions belonging to it will be called γ-optimal. For the sake of notation brevity, we will drop the parameter(s) X and/or R where it is clear from the context, which dataset and cost function are meant.

Properties of the parameter γ are crucial for understanding its role. On one hand, it is obvious that infinite γ yields the whole set of feasible solutions:

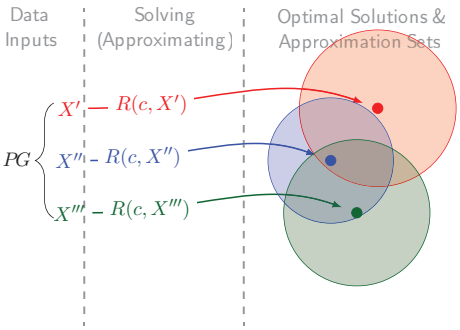
$$\mathcal{C}_\gamma|_{\gamma=\infty}(X) \equiv \mathcal{C}.$$

On the other hand, it holds

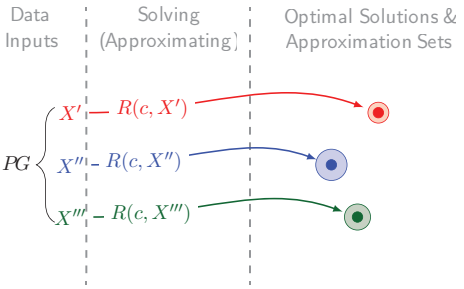
$$\mathcal{C}_\gamma|_{\gamma=0}(X) \equiv \mathcal{C}^\perp(X) \equiv \{c^\perp(X)\},$$



(a) Large γ: approximation sets are in a great agreement, but non-informative at all.



(b) Decreasing γ: approximation sets get distinguished, and more information is extracted.



(c) Small γ: almost all the information is extracted, but solutions are in poor agreement.

Figure 3.1 Intuitive illustration of informativeness vs. stability. Approximation sets are parametrized by γ. The data inputs X', X'' and X''' come from the same generative source. Decreasing the parameter γ leads to extracting more information from the given data, but at the same time making solutions less stable.

i.e. zero γ yields only optimal solutions. Selection of the parameter γ allows to trade-off stability of the solutions (extreme case: $\gamma = \infty$) and the their informativeness (extreme case: $\gamma = 0$). This raises a very important question: does there exist a way to choose this parameter?

3.4.2 Communication and Learning Stability

In this section, we will be working under definitions of Section 2.2.3. The approximation set-based approach has clear analogies in communication, featuring the idea of communication by means of data and solutions. To illustrate this relation, we first will refer to information theory and coding. As established by Shannon (1948); Shannon and Weaver (1963), all the rates up to channel capacity are

achievable with vanishing error.

Shannon's Channel Coding Theorem (e.g. Theorem 7.7.1, Cover and Thomas, 2006). *For a discrete memoryless channel, all rates below capacity C are achievable. Specifically, for every rate $R_{code} < C$, there exists a sequence of $(2^{nR_{code}}, n)$ codes with maximum probability of error $\lambda^{(n)} \to 0$. Conversely, any sequence of $(2^{nR_{code}}, n)$ codes with $\lambda^{(n)} \to 0$ must have $R_{code} \leq C$.*

This important theoretical statement has a non-constructive proof resting on the idea of random coding with code length n going to infinity, and thus it does not provide a practical way of building such codes of finite length. It turns out, that an attempt to design a finite-sized code faces the trade-off between its error-correcting capability and its rate. An example of this idea is the simplest Hamming code of length 3 which we are going to briefly illustrate due to its importance for the next steps.

Figures 3.2 and 3.3 to some extent explain this trade-off in the simplest possible setting, thus preparing the reader for introducing the communication channel by means of datasets. Figure 3.2 shows that one can vary the codebook vector set by, for instance, expanding "neighborhoods" of two vertices (000) and (111) by including all the adjacent vectors, while Figure 3.3 demonstrates that although the above process reduces the code rate from $R_{\text{code}} = 1$ down to $R_{\text{code}} = 1/3$, it increases its error-correcting capability so that the code can now tolerate all the

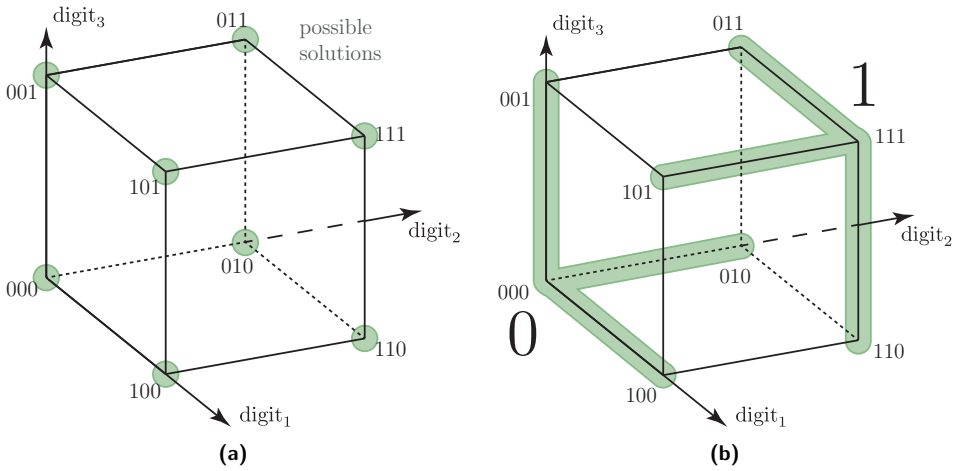
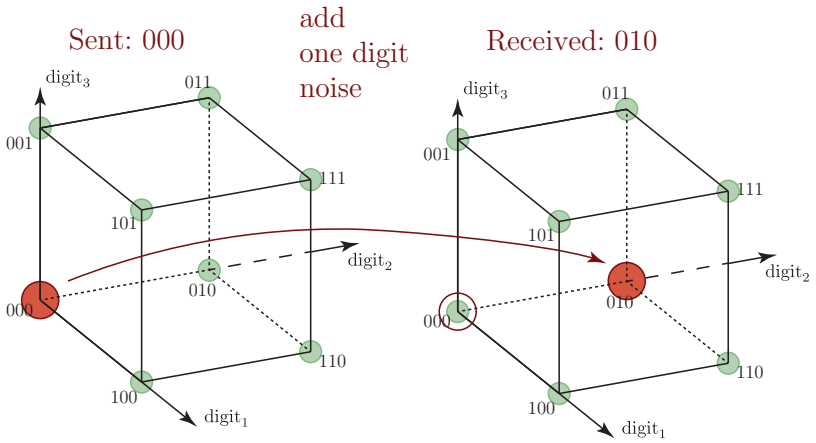
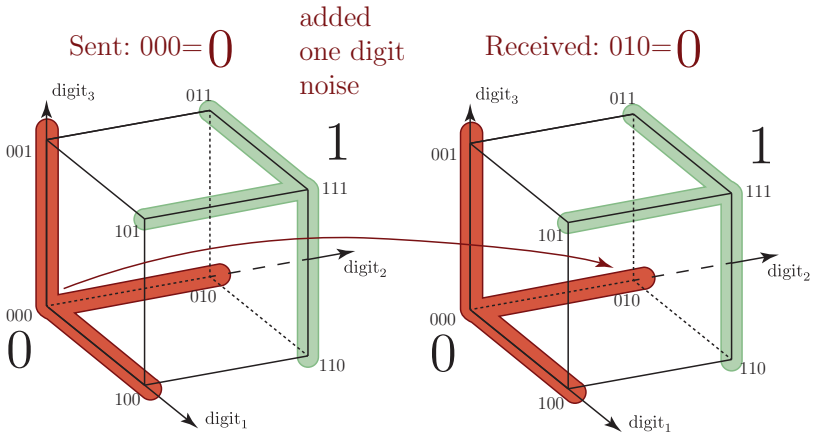


Figure 3.2 Placing codebook vectors of length 3 on a Boolean cube. Case **(a)** is a "mean" option, when we use all the eight vertices as codebook vectors. Case **(b)** is a "lean" option, when we use some of vertices as neighborhoods to the two codebook vectors 000 and 111 denoted as big 0 and big 1.



(a) High rate ($R_{\text{code}} = 1$), but no way to correct the error (red: sent and received codes).



(b) Lower rate ($R_{\text{code}} = 1/3$), correcting one digit error (red: sent and received codes).

Figure 3.3 Dealing with one digit error. Case **(a)** is high rate option with eight codebook vectors, leading to a low error-correcting capacity (in fact, no error can be tolerated). Case **(b)** is lower rate option with two codebook vectors leading to a higher error-correcting capacity (one digit error can be tolerated, two digits not).

one-digit errors. One can also imagine an extreme (not shown in figures) case of high two-digit noise: it is easy to see that under this condition, a reliable, stable communication is only possible with *only one* codebook vector and the *zero rate*

($R_{\text{code}} = 0$). In other words, the code gets less informative, but more robust.

Algorithm 3.1: Establishing the Communication

Data:
instance of the dataset $X' \in \mathcal{X}$,
solution set $\mathcal{C} = \{c\}$,
cost function $R(c, X)$,
set of transformations $\mathbb{T} = \{\tau\}$, where $\tau \colon \mathcal{X} \to \mathcal{X}$
parameter γ

Result: established communication scheme

Sender and Receiver agree on $R(c, X)$;
Sender and Receiver agree on X';
Sender and Receiver agree on $\mathbb{T}$;
Sender and Receiver agree on γ;
`// Then, a coverage by approximation sets is generated:`
foreach $\tau \in \mathbb{T}$ **do**
    both Sender and Receiver generate a transformed dataset $\tau \circ X'$;
    both Sender and Receiver compute γ-approximation set $\mathcal{C}_\gamma(\tau \circ X')$;

As well as in the coding scenario described above, the learning process can be viewed as a noisy communication, where the model is a *decoder* which tries to figure out the solution to *true (useful) signal* contained in the noisy data. Thus, the following rough analogies can be pointed out:

- The role of codebook vectors is played by solutions c to the optimization problem $R(c, X)$.
- The role of errors is played by the noise generating process $PG()$ (see. (3.7)), which injects uncertainty into data X.
- The role of "neighborhoods" of codebook vectors from Figures 3.2(b) and 3.3(b) is played by approximation sets.

We are now going, following Buhmann (2010a), to introduce an artificial communication scenario (Algorithms 3.1, 3.2 and 3.3). We advise the reader to compare the textual explanation with the pictorial one in Figure. 3.4.

Encoding step (Algorithm 3.1 and 3.2; Fig. 3.4(a)). Very briefly, the Sender-Receiver analogy consists in distinguishing individual solutions by means of the noisy datasets: Sender sends a message (defined below) encoded by the first dataset, and Receiver receives this message, but perturbed by means of the second dataset. More precisely, assuming the generative process PG (see (3.7)), the

transmitted "messages" are the transformations $\tau \in \mathbb{T}$ of the datasets, so

$$\tau \in \mathbb{T}, \quad \tau \colon \mathcal{X} \to \mathcal{X}. \tag{3.10}$$

Now, both Sender and Receiver are agreeing on the dataset X', which will play the role of the encoding "benchmark". Sender then picks a transformation τ_{send} and encodes the message by means of X' via applying one to the other:

$$X_{\text{send}} \coloneqq \tau_{\text{send}} \circ X', \tag{3.11}$$

and sends it out. Remember that Receiver does not know τ_{send}, but knows "codebook approximation sets" $\{\tau \circ X'\}_{\tau \in \mathbb{T}}$.

Hypothetic noise-free transmission. If there were no noise, Receiver, having obtained $X_{\text{received}} = X_{\text{send}}$, and knowing both $\mathbb{T}$ and X', could just recover the τ_{send} by enumerating:

$$\widehat{\tau} \coloneqq \arg\max_{\tau \in \mathbb{T}} \mathbb{1}\{X_{\text{received}} = \tau \circ X'\}. \tag{3.12}$$

Algorithm 3.2: Encoding and Transmission

Data:
instance of the dataset $X' \in \mathcal{X}$,
instance $X'' \in \mathcal{X}$ not known to receiver,
solution set $\mathcal{C} = \{c\}$,
cost function $R(c, X)$,
set of transformations $\mathbb{T} = \{\tau\}$, where $\tau \colon \mathcal{X} \to \mathcal{X}$
parameter γ

Result: a received message

1 Sender picks a $\tau_{\text{send}} \in \mathbb{T}$ and sends it;
2 Sender encodes it by generating a transformed dataset $\tau_{\text{send}} \circ X'$ and sends it;
3 Sender sends $\tau_{\text{send}} \circ X'$;
4 `// Channel noise comes in the next line:`
5 Channel introduces error by applying transformation τ_{send} to a X'';
6 Receiver receives $\tau_{\text{send}} \circ X''$ without knowing either τ_{send} or X'';

Algorithm 3.3: Decoding

Data:
instance of the dataset $X' \in \mathcal{X}$,
instance $X'' \in \mathcal{X}$ not known to receiver,
solution set $\mathcal{C} = \{c\}$,
cost function $R(c, X)$,
set of transformations $\mathbb{T} = \{\tau\}$, where $\tau \colon \mathcal{X} \to \mathcal{X}$
parameter γ

Result: Transformation $\widehat{\tau}$ which is estimate for τ_{send}

1 Receiver computes a γ-approximation set of the received dataset: $\mathcal{C}_\gamma(\tau_{\text{send}} \circ X'')$;

2 Receiver maximizes its overlap with known γ-approximation sets:

$$\widehat{\tau} = \arg\max_{\tau \in \mathbb{T}} \left| \mathcal{C}_\gamma(\tau \circ X') \cap \mathcal{C}_\gamma(\tau_{\text{send}} \circ X'') \right| \tag{3.15}$$

Actual noisy transmission (Algorithm 3.2; Fig. 3.4(b)). However, the noise is injected by replacing X' by X'', which is a noisy version of the initial dataset:

$$X_{\text{received}} \coloneqq \tau_{\text{send}} \circ X'', \tag{3.13}$$

which makes it impossible for Receiver to perfectly match obtained message to any of the "benchmarked ones" like in Eq. (3.12).

Remark. It it important to realize, that there are two manifestations of noise in this scenario. One is the original source of noise generated by $PG(\cdot)$ and resulting in replacing X' by X''. The other is the transmission error caused by difference between the sent and received messages.

Decoding (Algorithm 3.3; Fig. 3.4(c) and 3.4(d)). Just the same as the Hamming channel performs decoding the received vector by finding the closest codebook vector, our Receiver tries to find the closest codebook dataset out of all the possible datasets $\{\tau \circ X'\}_{\tau \in \mathbb{T}}$. Closeness is measured as the size of the intersection of their approximation sets:

$$\widehat{\tau} = \arg\max_{\tau \in \mathbb{T}} \left| \mathcal{C}_\gamma(\tau \circ X') \cap \mathcal{C}_\gamma(\tau_{\text{send}} \circ X'') \right|, \tag{3.14}$$

thus, approximation sets play the role of parity check regions here.

It is crucially important to realize that this decoding rule is very similar to that of the Hamming code (and thus very natural), because in the Hamming coding, the

closeness is measured by *minimizing* the Hamming distance between the received vector and the codebook vectors, which is the same as *maximizing* the intersection between them:

$$\begin{aligned}\widehat{\mathbf{x}} &= \arg\min_{\mathbf{x}\in\mathbb{B}^3} \|\mathbf{x}_{\text{received}} \oplus \mathbf{x}\| \\ &= \arg\min_{\mathbf{x}\in\mathbb{B}^3} \big(n - \|\mathbf{x}_{\text{received}} \cap \mathbf{x}\|\big) \\ &= \arg\max_{\mathbf{x}\in\mathbb{B}^3} \|\mathbf{x}_{\text{received}} \cap \mathbf{x}\|. \end{aligned} \tag{3.16}$$

Decoding error and its probability. When $\widehat{\tau} \neq \tau_{\text{send}}$, we say that a decoding error occurs. Obviously, the noise in our channel (Algorithm 3.2), acting via $PG(\cdot|X^0)$, is the reason for that. Transferring robust optimization problem into a robust decoding problem, we now will answer, following Buhmann (2010a), a natural question: how can we bound this probability?

We are interested in bounding the probability

$$\mathbb{P}(\widehat{\tau} \neq \tau_{\text{send}}|\tau_{\text{send}}). \tag{3.17}$$

Before we proceed, we will denote the intersection in (3.15) as follows:

$$\Delta\mathcal{C}_\gamma^\tau \coloneqq \mathcal{C}_\gamma(\tau \circ X') \cap \mathcal{C}_\gamma(\tau_{\text{send}} \circ X''). \tag{3.18}$$

Due to the union bound, it holds that

$$\mathbb{P}(\widehat{\tau} \neq \tau_{\text{send}}|\tau_{\text{send}}) \leq \sum_{\tau\in\mathbb{T}} \mathbb{P}\big(|\Delta\mathcal{C}_\gamma^\tau| \geq |\Delta\mathcal{C}_\gamma^{\tau_{\text{send}}}| \;\big|\; \tau_{\text{send}}\big), \tag{3.19}$$

i.e. for decoding error to occur, one has to encounter an approximation set which is yielded by a wrong transformation, but happens to be closer to the received approximation set (this is illustrated in Figure 3.5(c)). The last bound can be rewritten via the indicator function:

$$\mathbb{P}(\widehat{\tau} \neq \tau_{\text{send}}|\tau_{\text{send}}) \leq \sum_{\tau\in\mathbb{T}} \mathbb{E}_{PG}\big[\mathbb{1}\{|\Delta\mathcal{C}_\gamma^\tau| \geq |\Delta\mathcal{C}_\gamma^{\tau_{\text{send}}}|\} \;\big|\; \tau_{\text{send}}\big], \tag{3.20}$$

where the expectation is taken w.r.t. the problem generation process $X', X'' \sim PG(\cdot|X^0)$. We further utilize the monotonicity of log function:

$$\mathbb{1}\{|\Delta\mathcal{C}_\gamma^\tau| \geq |\Delta\mathcal{C}_\gamma^{\tau_{\text{send}}}|\} = \mathbb{1}\{\log|\Delta\mathcal{C}_\gamma^\tau| \geq \log|\Delta\mathcal{C}_\gamma^{\tau_{\text{send}}}|\} \tag{3.21}$$

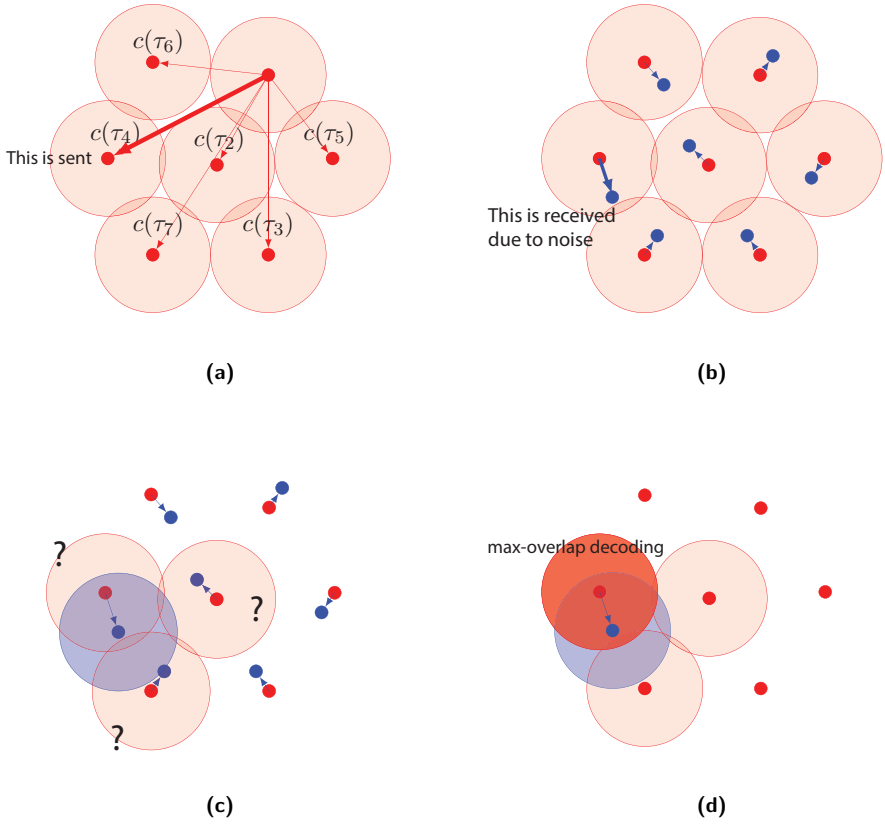


Figure 3.4 Process of correct decoding by approximation sets in the solution space: **(a)** X' is set and sender sends τ_4; **(b)** due to noise which replaces X' by X'', all the minimizers move around (red to blue) in the solution space; **(c)** the received solution is surrounded by its approximation set (blue) and overlaps are considered; **(d)** decoded solution (dark red) happens to be τ_4 which was initially sent (correct decoding).

and the fact that $\mathbb{1}\{x \geq 0\} \leq \exp(x)$ to come to the following:

$$\mathbb{E}_{PG}\Big(\mathbb{1}\{|\Delta\mathcal{C}_\gamma^\tau| \geq |\Delta\mathcal{C}_\gamma^{\tau_{\text{send}}}|\} \;\Big|\; \tau_{\text{send}}\Big) \leq \frac{|\mathcal{C}_\gamma(X')|\,|\mathcal{C}_\gamma(X'')|}{|\mathbb{T}|\,|\Delta\mathcal{C}_\gamma^{\tau_{\text{send}}}|}, \tag{3.22}$$

where the product in the nominator comes from the fact that, under our generation process, the data instances X' and X'' are independent given X^0, see (3.8).

In the spirit of Shannon (1948), we use the random coding argument here: all the τ are identically distributed and independent, hence the above can be

rewritten:

$$\mathbb{P}(\widehat{\tau} \neq \tau_{\text{send}} | \tau_{\text{send}}) \leq (|\mathbb{T}| - 1) \exp(-I_\gamma(\tau_{\text{send}}, \widehat{\tau})), \tag{3.23}$$

where

$$I_\gamma(\tau_{\text{send}}, \widehat{\tau}) \coloneqq \mathbb{E} \log\Big(\frac{|\mathbb{T}| \, |\Delta \mathcal{C}_\gamma^{\tau_{\text{send}}}|}{|\mathcal{C}_\gamma(X')| \, |\mathcal{C}_\gamma(X'')|} \Big). \tag{3.24}$$

Optimizing approximation parameter γ. At this point, we can determine the optimal γ^* as follows: the optimal approximation threshold is chosen as

$$\gamma^* = \arg\max_{\gamma \geq 0} I_\gamma(\tau_{\text{send}}, \widehat{\tau}). \tag{3.25}$$

In practical applications and in the spirit of the Shannon's random coding argument, it is often assumed that that $\tau_{\text{send}} = \text{Id}$, i.e. one computes

$$I_\gamma(\tau_{\text{send}}, \widehat{\tau}) \coloneqq \mathbb{E} \log\Big(\frac{|\mathbb{T}| \, |\Delta \mathcal{C}_\gamma(X', X'')|}{|\mathcal{C}_\gamma(X')| \, |\mathcal{C}_\gamma(X'')|} \Big), \tag{3.26}$$

where

$$\Delta \mathcal{C}_\gamma \coloneqq \mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X''). \tag{3.27}$$

In practice, one often replaces $|\mathbb{T}|$ with the cardinality of the full solution set (Chehreghani et al., 2012), reflecting a specific choice of possible transformations[3]:

$$I_\gamma(\tau_{\text{send}}, \widehat{\tau}) \coloneqq \mathbb{E} \log\Big(\frac{|\mathcal{C}| \, |\Delta \mathcal{C}_\gamma(X', X'')|}{|\mathcal{C}_\gamma(X')| \, |\mathcal{C}_\gamma(X'')|} \Big), \tag{3.28}$$

Definition 3.3. *We will call the above quantity ASC γ-score. We will call its maximum simply ASC score or Approximation Capacity (AC):*

$$C \coloneqq \max_\gamma I_\gamma. \tag{3.29}$$

Remark. It is interesting to note that the semantics of $I_\gamma(\tau_{\text{send}}, \widehat{\tau})$ is surprisingly similar to that of mutual information (Definition 2.4). First, both are related to the maximum rate of certain channel. Second, both can be decomposed in quite a similar way: recall from (2.7) that, for random variables X and Y,

$$\text{MI}(X, Y) = H(X) + H(Y) - H(X, Y).$$

[3]Since the proof of error probability rests on the argument of random coding and the codebook messages are chosen randomly, all the considerations remain valid.

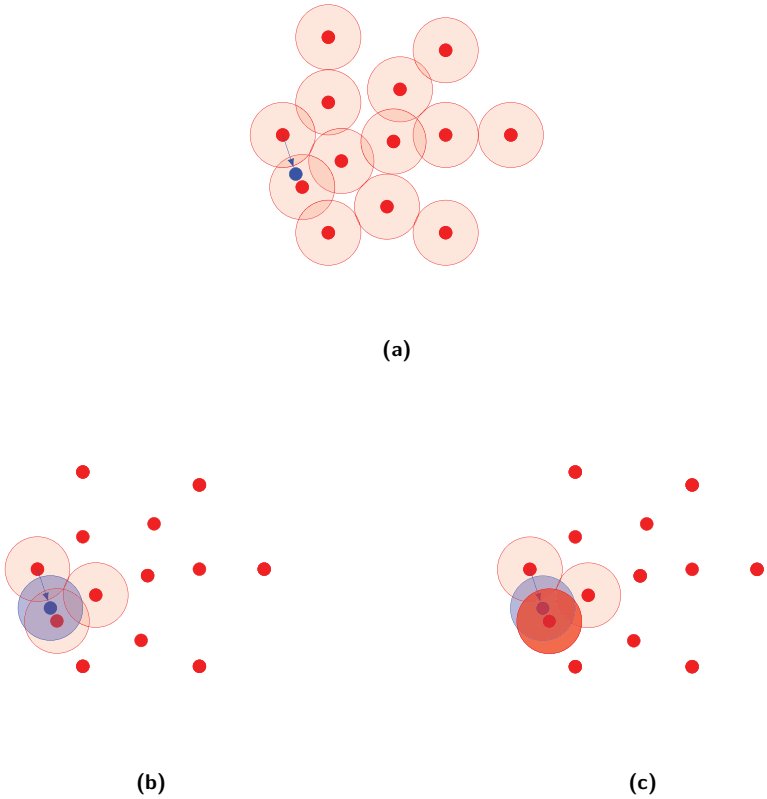


Figure 3.5 Decreased γ and increased code rate leads to incorrect decoding: **(a)** same setting (i.e. same noise) as in Figure 3.4, but added more codebook vectors; **(b)** due to noise which replaces X' by X'', all the minimizers move around (red to blue) in the solution space, **(c)** decoded solution (dark red) happens to be wrong (incorrect decoding).

In a same way one may observe, that (3.28) can be very easily decomposed into three logarithms:

$$I_\gamma(\tau_{\text{send}}, \widehat{\tau}) = \overbrace{-\mathbb{E}\log\Big(\frac{|\mathcal{C}_\gamma(X')|}{|\mathcal{C}|}\Big)}^{\text{single entropy}} \overbrace{-\mathbb{E}\log\Big(\frac{|\mathcal{C}_\gamma(X'')|}{|\mathcal{C}|}\Big)}^{\text{single entropy}} \underbrace{+\mathbb{E}\log\Big(\frac{|\Delta\mathcal{C}_\gamma(X', X'')|}{|\mathcal{C}|}\Big)}_{\text{joint entropy}}, \tag{3.30}$$

where first two terms can be contemplated as single entropies of uniform distributions over approximation sets, and the third term corresponds to the joint entropy.

Remark. In practical applications, when there are only two data points X', X'' and no information about $PG(\cdot)$ is available, one can use an empirical version of (3.28)

$$\widehat{I}_\gamma(X', X'') \coloneqq \log\Big(\frac{|\mathcal{C}| \; |\Delta\mathcal{C}_\gamma(X', X'')|}{|\mathcal{C}_\gamma(X')| \; |\mathcal{C}_\gamma(X'')|}\Big), \tag{3.31}$$

as an estimator without the expectation sign. More on that will be given in Chapter 4 when describing the application.

3.5 Another View: Similarity Approach

In this section, we briefly visit, with sufficient adaptation of notation[4], another view on finding a robust approximation, which is called *similarity approach* and was introduced and developed, e.g., in (Šrámek, 2013; Pröger, 2016; Buhmann et al., 2017a). Yielding the same quantity as in (3.25), this approach arose from a specific interpretation of the ASC (Buhmann, 2010a).

For the sake of itegration into the thesis, in this section we use additive approximation set notation (i.e. the same as everywhere in this thesis), while Šrámek (2013) used a multiplicative one[5]. Assume there are two *data instances* X' and X'', coming from the same source $PG(\cdot|X^0)$, see (3.7). Šrámek (2013) identifies two cases:

- If the generation process PG is very noisy, resulting in two non-similar instances X' and X'' it is obvious that the intersection of two approximation sets $\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')$ will contain some solutions when γ is large enough. Šrámek (2013) calls such solutions *expected due to* γ. At this point, the reader can start building analogies to the above by revisiting Figure 3.1(a), where large approximation sets yield a lot of solutions in the intersection.

- On the other hand, if the two instances X' and X'' are more similar, which is the case for a low-noise PG, the intersection $\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')$, taken at the same γ value, will contain, in addition to the above-mentioned (expected due to γ) ones, some solutions due to the similarity of the instances. Šrámek (2013) calls them *unexpected*. In terms coined later (Buhmann et al., 2017a) it is called *expected due to similarity*.

[4]The two interpretations of the same idea have been developed in parallel, hence there are two consistent systems of notation.

[5]For the definiton of multiplicative approximation sets, refer to e.g. (Šrámek, 2013) or (Buhmann et al., 2017a).

The point of introducing such cases consists in the following: these latter solutions, — i.e. the ones expected due to similarity — are likely to be good choices for possible test instance X''' that comes from the same source.

The goal is thus shifted to finding the γ that maximizes the ratio of the number of solutions that are expected due to similarity over the size of the intersection (compare to the ASC approach (3.25); the comparison will be summarized in conclusion, Section 3.9). To fulfill the task, several definitions are required. Figure 3.7 illustrates these definitions.

Definition 3.4 (Feasible approximation set). *A set of solutions $F \subseteq \mathcal{C}$ is called a* feasible approximation set *if there exists some instance $\tilde{X}$ and some number $\tilde{\gamma}$ such that F is the $\tilde{\gamma}$-approximation set of $\tilde{X}$.*

Definition 3.5 (Expected intersection sizes due to γ). *Given γ and the sizes $|\mathcal{C}_\gamma(X')| =: k(\gamma)$ and $|\mathcal{C}_\gamma(X'')| =: l(\gamma)$, let $es(\gamma, k(\gamma), l(\gamma))$ denote the expected size of the intersection of two feasible approximation sets A and B of sizes $k(\gamma)$ and $l(\gamma)$, respectively.*

Definition 3.6 (Expected intersection due to similarity). *Given γ and the sizes $|\mathcal{C}_\gamma(X')| =: k(\gamma)$ and $|\mathcal{C}_\gamma(X'')| =: l(\gamma)$, if the intersection of $\mathcal{C}_\gamma(X')$ and $\mathcal{C}_\gamma(X'')$ is larger than the expected size $es(\gamma, k(\gamma), l(\gamma))$, then it contains some solutions that are expected due to similarity, and we will denote them $sim(\gamma)$.*

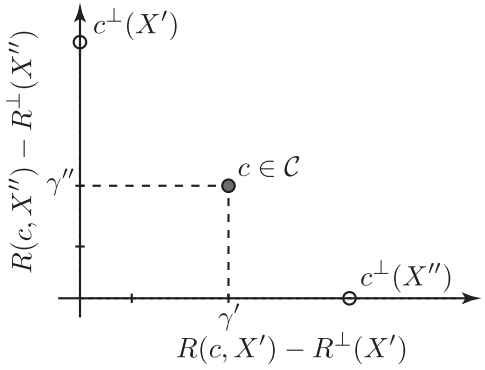
Thus we have

$$|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')| = sim(\gamma) + es(\gamma, k(\gamma), l(\gamma)), \tag{3.32}$$

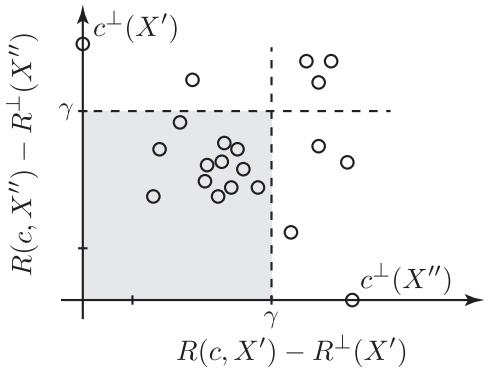
and, to maximize the probability that the uniformly randomly chosen solution from the intersection is stable, we want to find the value γ that maximizes $\frac{sim(\gamma)}{sim(\gamma)+es(\gamma,k(\gamma),l(\gamma))}$. The following about maximization objectives holds:

$$\begin{aligned}
\arg\max_{\gamma>0} \ & \frac{sim(\gamma)}{sim(\gamma) + es(\gamma, k(\gamma), l(\gamma))} \\
&= \arg\max_{\gamma>0} \left(1 - \frac{es(\gamma, k(\gamma), l(\gamma))}{sim(\gamma) + es(\gamma, k(\gamma), l(\gamma))}\right) \\
&= \arg\min_{\gamma>0} \ \frac{es(\gamma, k(\gamma), l(\gamma))}{sim(\gamma) + es(\gamma, k(\gamma), l(\gamma))} \\
&= \arg\max_{\gamma>0} \ \frac{sim(\gamma) + es(\gamma, k(\gamma), l(\gamma))}{es(\gamma, k(\gamma), l(\gamma))},
\end{aligned} \tag{3.33}$$

hence we can reformulate (for the sake of clarity) the objective of the similarity-



(a) Placing a solution $c \in \mathcal{C}$



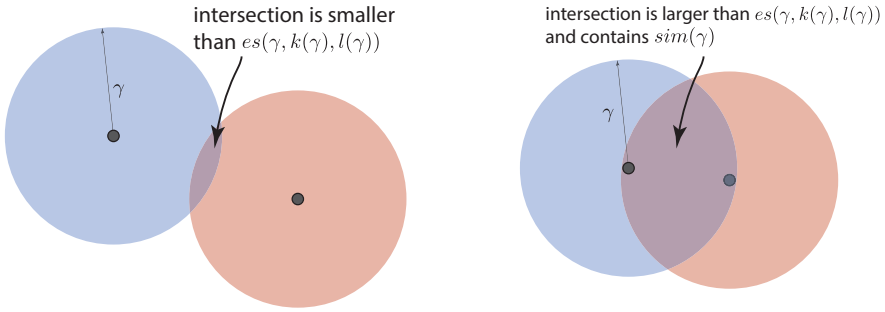
(b) Intersection $\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')$

Figure 3.6 Approximation sets for the instances X' and X''. By $c^\perp(X)$ we denote the solution whose cost is minimum in X. **(a)**: We place each solution $c \in \mathcal{C}$ at position (γ', γ''), where $\gamma' = R(c, X') - R^\perp(X')$ and $\gamma'' = R(c, X'') - R^\perp(X'')$. **(b)**: Example of intersection of approximation sets $\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')$ (this view on approximation sets was originally suggested by Tobias Pröger (cf. Buhmann et al., 2017a), figure labels adapted for additive notation).

based approach as maximizing the value

$$S_\gamma(X', X'') \coloneqq \frac{|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{es(\gamma, k(\gamma), l(\gamma))} = \frac{sim(\gamma) + es(\gamma, k(\gamma), l(\gamma))}{es(\gamma, k(\gamma), l(\gamma))}. \tag{3.34}$$

Problem-based instance similarity. In equation (3.34), the expected size of the intersection is w.r.t. the problem specific probability distribution over all feasi-



(a) Two non-similar approximation sets. **(b)** Two similar approximation sets

Figure 3.7 Illustration of ideas contained in Definitions 3.4, 3.5 and 3.6: as opposed to randomly chosen approximation sets **(a)**, two related (similar) approximation sets **(b)** have a $sim(\gamma)$ component of the intersection, which we naturally seek to maximize.

ble approximation sets of size $|\mathcal{C}_\gamma(X')|$ and $|\mathcal{C}_\gamma(X'')|$, respectively. However, this distribution is hard to estimate, so, Šrámek (2013) introduced a problem-based based instance similarity, which approximates the denominator by a uniformly chosen pair of approximation sets.

Definition 3.7 (Problem-based instance similarity)**.** *Let X' and X'' be two input instances of a combinatorial optimization problem $\mathcal{P}$ with solution space $\mathcal{C}$. For a given γ, let $\mathcal{C}_\gamma(X')$ and $\mathcal{C}_\gamma(X'')$ be γ-approximation sets for X' and X''. Further, let $\mathcal{F}_k$ denote the set of all feasible approximation sets of size k, i.e., the set of all such sets $F \subseteq \mathcal{C}$ of size k for which there exists an instance I' and a value $\tilde{\gamma}$ such that $F = \mathcal{C}_{\tilde{\gamma}}(\tilde{X})$. Then, the expression*

$$S_\gamma(X', X'') = \frac{|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{\mathbb{E}_{A \in \mathcal{F}_{|\mathcal{C}_\gamma(X')|}, B \in \mathcal{F}_{|\mathcal{C}_\gamma(X'')|}} \left[|A \cap B|\right]} \tag{3.35}$$

is the similarity of X' and X'' at value γ *(with respect to the optimization problem $\mathcal{P}$), and the expression*

$$S(X', X'') \coloneqq \max_\gamma S_\gamma(X', X'') \tag{3.36}$$

is the similarity of X' and X'' *with respect to the optimization problem $\mathcal{P}$.*

Thus, the similarity-based approach (in the following referred just as "similarity" approach) works as follows. First, we compute the value γ that maximizes

the similarity

$$S_\gamma(X', X'') = \frac{|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{\mathbb{E}_{A \in \mathcal{F}_{|\mathcal{C}_\gamma(X')|}, B \in \mathcal{F}_{|\mathcal{C}_\gamma(X'')|}} \left[|A \cap B|\right]}, \tag{3.35}$$

where the expectation is w.r.t. the uniform probability distribution over the elements in $\mathcal{F}_{|\mathcal{C}_\gamma(X')|}$ and $\mathcal{F}_{|\mathcal{C}_\gamma(X'')|}$, respectively. We then return a solution from $\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')$ uniformly at random.

However, there are two practical issues with the procedure shown above: a) it is not always clear how to directly optimize γ for the value of (3.35); and b) sampling from the intersection of the corresponding γ-approximation sets uniformly at random might be difficult. Despite all that, the similarity approach can be always applied in the cases, where one can provide all the steps of Algorithm 3.4.

Algorithm 3.4: Pipeline for Similarity Approach (Section 3.5)

1. Determine the domains $\mathcal{F}_k$ of feasible approximation sets of size k.
2. Provide a mathematical analysis or an algorithm $ALG_{\mathbb{E}}$ that computes the expected size of the intersection of two approximation sets of given sizes k and l.
3. Provide an algorithm $ALG_{\cap}$ that computes the size of the intersection $\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')$, given γ and two instances X' and X''.
4. Find γ^* that maximizes the similarity $S_\gamma(X', X'')$, using $ALG_{\mathbb{E}}$ and $ALG_{\cap}$.
5. Provide an algorithm ALG_{rand} that picks a uniform random solution from the intersection $\mathcal{C}_{\gamma^*}(X') \cap \mathcal{C}_{\gamma^*}(X'')$.

In order to fulfill these tasks, on can use several tools provided below. It is important to notice that these useful theorems close the gap between the ASC formulation (3.28) and the similarity approach formulation (3.35).

Theorem 3.1 (Šrámek, 2013). *Let $\mathcal{P} = (\mathcal{X}, \mathcal{C}, R)$ (see Section 3.3.1) be an optimization problem with the property that for any subset F of the set of all feasible solutions $\mathcal{C}$ there exists an instance $\tilde{X} \in \mathcal{X}$ and a value $\tilde{\gamma}$ such that $\mathcal{C}_{\tilde{\gamma}}(\tilde{X}) = F$. Then, the similarity of two instances $X', X'' \in \mathcal{X}$ at value γ is*

$$S_\gamma(X', X'') = \frac{|\mathcal{C}||\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{|\mathcal{C}_\gamma(X')||\mathcal{C}_\gamma(X'')|}. \tag{3.37}$$

However, as Šrámek (2013) notes, there exists an issue that not every subset $F \subseteq \mathcal{C}$ is a feasible approximation set, and there is still no general algorithm of

computing the expected size of the intersection. The following chain of theorems provides some approximation guarantees for the value of (3.37).

Theorem 3.2 (Šrámek, 2013). *Let $\mathcal{P} = (\mathcal{X}, \mathcal{C}, R)$ be an optimization problem. If $|\mathcal{C}_\gamma(X')| = |\mathcal{C}_\gamma(X'')|$ for a given γ, then*

$$S_\gamma(X', X'') \leq \frac{|\mathcal{C}||\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{|\mathcal{C}_\gamma(X')||\mathcal{C}_\gamma(X'')|}. \tag{3.38}$$

Theorem 3.3 (Šrámek, 2013). *Let A be a constant such that for each feasible solution c of some optimization problem $\mathcal{P} = (\mathcal{X}, \mathcal{C}, R)$ it holds that $|\{F \in \mathcal{F}_k | c \in F\}| \leq Ak|\mathcal{F}_k|/|\mathcal{C}|$. Then,*

$$S_\gamma(X', X'') \geq \frac{|\mathcal{C}||\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{A|\mathcal{C}_\gamma(X')||\mathcal{C}_\gamma(X'')|}. \tag{3.39}$$

Theorem 3.4 (Šrámek, 2013). *Let $\mathcal{P} = (\mathcal{X}, \mathcal{C}, R)$ be an optimization problem. Then,*

$$S_\gamma(X', X'') \geq \frac{|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{|\mathcal{C}_\gamma(X')||\mathcal{C}_\gamma(X'')|}. \tag{3.40}$$

Remark. This shows that the step of deriving the appropriate specific formula or algorithm to calculate the expected size of the intersection is a necessary component of the approach, unless it is possible to show that for a concrete problem the upper bound is sufficient. We will speculate more on that in the conclusion to this chapter (Section 3.9).

3.6 Proof-of-Concept Prototypic Example

Previously, in Section 3.4, we introduced a method of solution regularization by ASC, and later in Section 3.5 we gave a thorough overview of an analogical approach called instance similarity. While they stem from completely different roots, it can be easily seen that they both aim at choosing an optimal approximation set width γ in a same way. Specifically, we seek to optimize

$$\gamma^* = \arg\max_{\gamma > 0} \frac{|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{|\mathcal{C}_\gamma(X')||\mathcal{C}_\gamma(X'')|} \tag{3.41}$$

Remark. Note that this equation is *not* identical either to its ASC version (3.28) or similarity-based version (3.35), although yielding same optimization goal; we

will briefly revisit the technical differences, like presence of logarithm, later in the conclusion.

One of the contributions of this thesis is to present an abstract proof-of-concept model for prototypical combinatorial optimization problems, which would allow to experimentally the advantages of the approximation set-based approaches. We will mostly experimentally investigate how the methods of Sections 3.4 and 3.5 perform on this model.

3.6.1 The Example Setting and Terminology

We expect the approximation set-based methods to exceed the performance of other optimization methods when the set of solutions that have stable cost over all or most instances is large enough not to be completely hidden in the noise. To highlight the potential of our approach, we consider an uncertain minimization problem $(\mathcal{X}, \mathcal{C}, R)$ in which the solution space $\mathcal{C}$ is partitioned into two sets $\mathcal{C}_{\text{stable}}$ and $\mathcal{C}_{\text{unstable}}$ of sizes n_{s} and n_{u}, respectively, which contain the stable and the unstable solutions, respectively. Without loss of generality we assume that

$$\begin{aligned} \mathcal{C} &= \{c_i\}_{i=1}^{n} \\ \mathcal{C}_{\text{stable}} &= \{c_1, \ldots, c_{|n_{\text{s}}|}\} \\ \mathcal{C}_{\text{unstable}} &= \{c_{|n_{\text{s}}|+1}, \ldots, c_{|n_{\text{s}}|+|n_{\text{u}}|}\}. \end{aligned} \tag{3.42}$$

The sets $\mathcal{C}_{\text{stable}}$ and $\mathcal{C}_{\text{unstable}}$ represent solutions which are desirable and non-desirable to be chosen, which reflects the fact that the approximation set-based approaches are designed to reliably tell them apart. We further assume that $n_{\text{s}} \ll n_{\text{u}}$, which corresponds to the fact that stable solutions should be hard to identify.

Our proof-of-concept scenario abstracts from a concrete optimization problem. In other words, we do not address here the problem of specific optimization algorithms. Hence we explicitly state that instead of generating inputs $X \in \mathcal{X}$, we rather directly generate costs of solutions $c \in \mathcal{C}$. In the terminology of Section 3.3.1, an instance X can be represented as a vector of random solution costs of length n:

$$X \coloneqq \langle R_i \rangle_{i=1}^{n}, \tag{3.43}$$

and the cost function is simply

$$R(c_i, X) \coloneqq R_i, \tag{3.44}$$

i.e. the i-th entry stores the cost of the solution c_i in X.

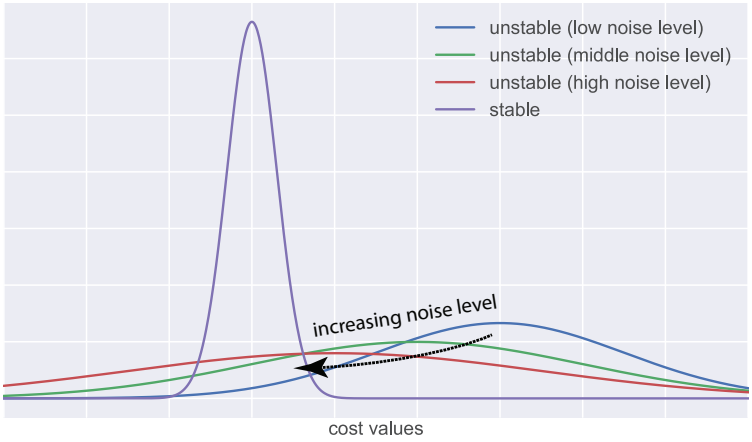


Figure 3.8 Schematic example of stable and unstable cost distributions and noise levels $N \in \mathcal{N}$ of unstable ones, as described in Section 3.6.2.

3.6.2 Problem Generation

We define the solution "desirability" by the intuition that costs of stable solutions have a small standard deviation and play the role of signal, while costs of unstable solutions have a higher mean and/or a higher standard deviation and play the role of deceiving noise.

We assume the cost vector of an instance X to be generated with the following random problem generating process $PG(\cdot)$:

1) the first n_s values are chosen at random according to some (fixed) probability distribution $\mathcal{D}_\mathrm{s}$, and

2) the remaining n_u values are chosen at random according to some (fixed) probability distribution $\mathcal{D}_\mathrm{u}$.

Naturally it is safe to assume that both $\mathcal{D}_\mathrm{s}$ and $\mathcal{D}_\mathrm{u}$ have the property that stable solutions are superior to unstable ones (Figure 3.8), e.g., because they have a smaller expected cost or a smaller variance, i.e. for any $R_\mathrm{stable} \sim \mathcal{D}_\mathrm{s}$ and $r_\mathrm{unstable} \sim \mathcal{D}_\mathrm{u}$, $\mathbb{E}[R_\mathrm{stable}] < \mathbb{E}[R_\mathrm{unstable}]$ and $\mathrm{Var}[R_\mathrm{stable}] < \mathrm{Var}[R_\mathrm{unstable}]$. We further assume $\mathcal{D}_\mathrm{s}$ and $\mathcal{D}_\mathrm{u}$ are independent of the instance and of the concrete solution (costs of stable solutions are always chosen from $\mathcal{D}_\mathrm{s}$, costs of unstable solutions are always chosen from $\mathcal{D}_\mathrm{u}$).

We model noise in a generic way by defining a set of noise levels $\mathcal{N}$ (the concrete definition depends on the type of the noise, see (Buhmann et al., 2017a)). For a fixed noise level $N \in \mathcal{N}$, we randomly generate an instance as follows. Stable

solutions are drawn from a distribution with fixed mean μ_{s} and fixed standard deviation σ_{s}. Unstable solutions are drawn from a distribution with mean $\mu_{\mathrm{u}}(N)$ and standard deviation $\sigma_{\mathrm{u}}(N)$. The distributions of the unstable solutions are chosen in a way such that for every two noise levels $N, N' \in \mathcal{N}$ with $N' > N$, we have $\mu_{\mathrm{u}}(N') < \mu_{\mathrm{u}}(N)$ or $\sigma_{\mathrm{u}}(N') > \sigma_{\mathrm{u}}(N)$.

Remark. Such assumptions on noise levels are justified by the fact that noise would naturally imply either a smaller expected cost, or a higher standard deviation, or both — resulting in a more aggressive "deceiving" of the algorithm. See Figure 3.8 for schematic illustration of this intuition.

Due to its enormous theoretical and practical relevance, we present here the results for the Gaussian noise model (for more noise settings, see (Buhmann et al., 2017a)). Stable solutions are drawn from a Gaussian distribution with mean $\mu_s = 1$ and standard deviation $\sigma_s = 1$. We define the noise levels $\mathcal{N}$ in such a way that for each noise level $N \in \mathcal{N}$, unstable solutions are drawn from a Gaussian distribution with mean $\mu_u(N) = 10$ and standard deviation $\sigma_u(N) = N$: $\mathcal{D}_{\mathrm{s}} = \mathrm{Norm}(1, 1)$ and $\mathcal{D}_{\mathrm{u}}(N) = \mathrm{Norm}(10, N^2)$.

3.6.3 The Goal and Success Metrics

Now, our goal is the following: given two instances X' and X'' generated by the random process $PG(\cdot)$ described above, our algorithm $\mathscr{A}$ has to compute a set of solutions $\widehat{\mathcal{C}}_{\mathscr{A}}$ of candidates for solutions in $\mathcal{C}_{\mathrm{stable}}$, from which it then picks a solution uniformly at random. The only knowledge of an algorithm consists of the two cost vectors of X' and X'' defined in (3.43). The algorithm cannot exploit the fact that there are two categories of solutions, and in particular it has no knowledge about $\mathcal{D}_{\mathrm{s}}$ and $\mathcal{D}_{\mathrm{u}}$.

Since we assume that a solution from $\widehat{\mathcal{C}}_{\mathscr{A}}$ is picked uniformly at random, we define the *success probability* of $\mathscr{A}$ with input X' and X'' as

$$P_{\mathscr{A}}(X', X'') = \frac{|\mathcal{C}_{\mathscr{A}} \cap \mathcal{C}_{\mathrm{stable}}|}{|\mathcal{C}_{\mathscr{A}}|}, \text{ for a solving algorithm } \mathscr{A}. \tag{3.45}$$

We want to investigate how the success probabilities of the similarity algorithm proposed in this chapter evolves with increasing noise, and benchmark it against some other algorithms. In this thesis, we present only a Joint Minimizer algorithms (see next section), but for the results produced on a more complete list of benchmarks we refer the reader to (Buhmann et al., 2017a).

3.6.4 Experimental Results

Benchmark: joint cost minimizing. When only two instances are given, the most efficient and straightforward idea to find a solution that is likely to be good for a test instance is to compute a solution c that minimizes the average cost, or equivalently, the joint cost $R(c, X') + R(c, X'')$. We refer to this method as the *Joint Minimizer* method in the plots below.

Results. For each noise level $N \in \mathcal{N}$, we perform the following experiment: we generate $r = 1000$ instance pairs $(X', X'')_{k\in\{1,\dots,r\}}$ with noise level N according to the $PG(\cdot)$ process described in Section 3.6.2, and for each of these instance pairs we compute $P_{\mathscr{A}}(X', X'')$ for all algorithms $\mathscr{A}$. After that we set

$$\widehat{P}_{\mathscr{A}}(N) \coloneqq \frac{1}{r}\sum_{k=1}^{r} P_{\mathscr{A}}(X', X'') \tag{3.46}$$

to estimate the average success probability of the proposed methods in dependency of the noise level N. Unless otherwise stated, $\mathcal{C}$ contains $n = 1000$ solutions.

In our experiments, $r = 1000$ repetitions turned out to be enough to exhibit the behaviors of the methods. Preliminary experiments with 10000 repetitions gave similar results: the rankings of the methods were the same, only the curves in the plots appeared to be smoother.

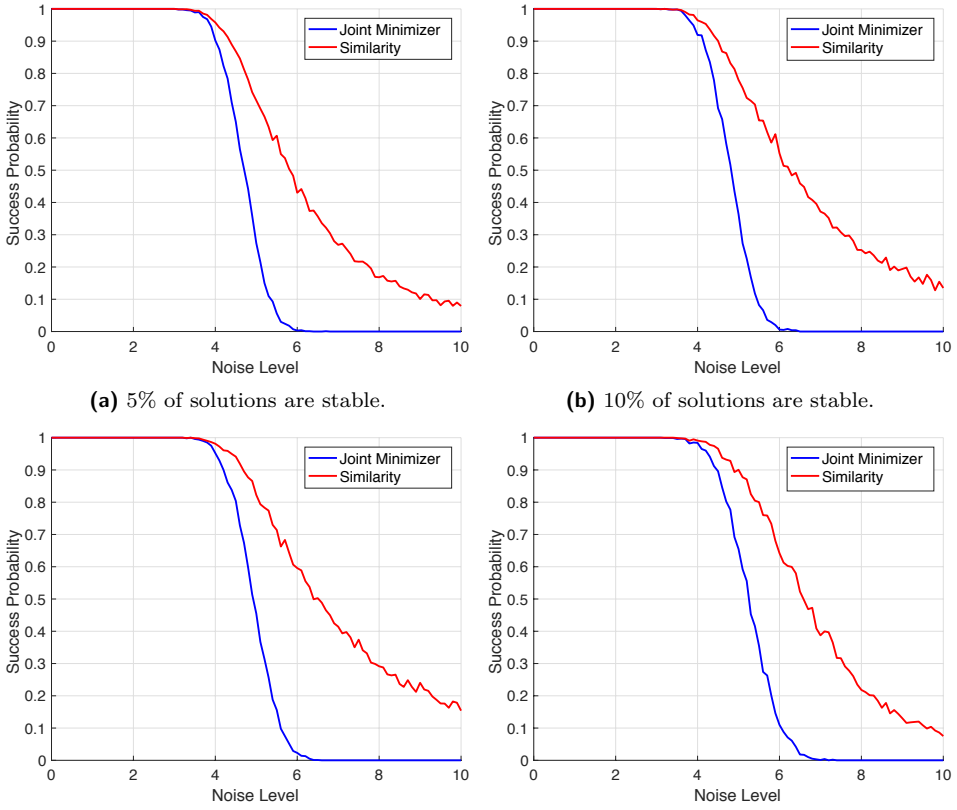
Figure 3.9 shows that the experimental results for Gaussian noise show a strong indication that the approximation set-based similarity approach is very competitive against the joint cost minimization. Note that the latter is a straightforward way to compute solutions, when only two inputs are provided.

3.7 Finding Optimal Approximations Analytically

3.7.1 Theoretical Results

One of our main assumptions was that the noise generating process is unknown to the predicting algorithm $\mathscr{A}$. It was also previously noted that the crucial step of the whole approximation set-based approach consists in deriving the appropriate specific formula or algorithm to calculate the similarity (3.37). As a first step towards a formal analysis of the model discussed in the previous section, in this section we thoroughly investigate how the similarity (3.37) behaves in expectation (where the expectation is computed over all pairs of instances generated by the random process $PG(\cdot)$), i.e., we analyze the function

$$S_\gamma^{\mathrm{EXP}} = \mathbb{E}_{X',X''\sim PG}\, S_\gamma(X', X''). \tag{3.47}$$



(a) 5% of solutions are stable.

(b) 10% of solutions are stable.

(c) 20% of solutions are stable.

(d) 50% of solutions are stable.

Figure 3.9 Experimental results where 5% **(a)**, 10% **(b)**, 20% **(c)** and 50% **(d)** of the solutions are stable. Total number of solutions equals 1000.

For simplicity we introduce the *calibrating assumption* that the minimum solutions of both instances X' and X'' have the same cost m:

$$\min_c R(c, X') \approx \min_c R(c, X'') \approx m. \tag{3.48}$$

Without this assumption our analysis would still be possible, but it would be more technical. Notice that the assumption does not imply that the minimum solutions themselves are the same: in general, it holds that

$$\arg\min_c R(c, X') \neq \arg\min_c R(c, X''), \tag{3.49}$$

i.e. minimum costs are not necessarily attained on the same solution.

Theorem 3.5. *Let $\gamma > 0$, $V = |\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|$, $W = |\mathcal{C}_\gamma(X')| \cdot |\mathcal{C}_\gamma(X'')|$, m be the minimum cost of a solution in both X' and X'' (i.e., the calibrating assumption is satisfied), and F_s and F_u denote the cumulative density functions of the stable and the unstable solutions, respectively, evaluated at $m + \gamma$. Then, the expected similarity* (3.47) *can be approximated by the estimated similarity*

$$S_\gamma^{\mathrm{EXP}} \sim \widehat{S}_\gamma \coloneqq |\mathcal{C}| \left(\frac{\mathbb{E}[V]}{\mathbb{E}[W]} - \frac{\mathrm{Cov}(V,W)}{\mathbb{E}[W]^2} + \frac{\mathrm{Var}[W] \cdot \mathbb{E}[V]}{\mathbb{E}[W]^3} \right) \tag{3.50}$$

where

$$\mathbb{E}[V] = n_s F_s^2 + n_u F_u^2, \tag{3.51}$$

$$\mathbb{E}[W] = (n_s F_s + n_u F_u)^2, \tag{3.52}$$

$$\begin{aligned} \mathrm{Cov}(V,W) = {} & n_s F_s^2 (1 - F_s^2) + 2n_s(n_s - 1) F_s^3 (1 - F_s) \\ & + 2n_s n_u F_s^2 F_u (1 - F_s) + 2n_s n_u F_s F_u^2 (1 - F_u) \\ & + n_u F_u^2 (1 - F_u^2) + 2n_u(n_u - 1) F_u^3 (1 - F_u), \ \textit{and} \end{aligned} \tag{3.53}$$

$$\begin{aligned} \mathrm{Var}[W] = {} & n_s^2 F_s^2 (1 - F_s^2) + 2n_s^2(n_s - 1) F_s^3 (1 - F_s) \\ & + 2n_s n_u (n_u - 1) F_s F_u^2 (1 - F_s) \\ & + 2n_s(n_s - 1) n_u F_s^2 F_u (1 - F_u) \\ & + 2n_s n_u F_s F_u (1 - F_s F_u) \\ & + n_u^2 F_u^2 (1 - F_u^2) + 2n_u^2(n_u - 1) F_u^3 (1 - F_u) \\ & + 4n_s^2 n_u F_s^2 F_u (1 - F_s) + 4n_s n_u^2 F_s F_u^2 (1 - F_u). \end{aligned} \tag{3.54}$$

Proof of Theorem 3.5. To make this proof more readable, we break it down into several steps.

1) *Preliminaries.* Let $m = \min_{c \in \mathcal{C}} R(c, X') = \min_{c \in \mathcal{C}} R(c, X'')$. Let c_i, $i \in \{1, \ldots, n_{\mathrm{s}}\}$ denote the solutions in $\mathcal{C}_{\mathrm{stable}}$ and $\bar{c}_i$, $i \in \{1, \ldots, n_{\mathrm{u}}\}$ denote the solutions in $\mathcal{C}_{\mathrm{unstable}}$. We define

$$\begin{aligned} A'_{i,\gamma} &= \mathbb{1}\{R(c_i, X') \le m + \gamma\}, \quad 1 \le i \le n_{\mathrm{s}} \\ A''_{i,\gamma} &= \mathbb{1}\{R(c_i, X'') \le m + \gamma\}, \quad 1 \le i \le n_{\mathrm{s}} \\ B'_{j,\gamma} &= \mathbb{1}\{R(\bar{c}_j, X') \le m + \gamma\}, \quad 1 \le j \le n_{\mathrm{u}} \\ B''_{j,\gamma} &= \mathbb{1}\{R(\bar{c}_j, X'') \le m + \gamma\}, \quad 1 \le j \le n_{\mathrm{u}}. \end{aligned}$$

Now the components of the similarity (3.34) can be expressed as

$$|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')| = \sum_{i=1}^{n_\mathrm{s}} A'_{i,\gamma} A''_{i,\gamma} + \sum_{j=1}^{n_\mathrm{u}} B'_{j,\gamma} B''_{j,\gamma},$$
$$|\mathcal{C}_\gamma(X')| = \sum_{i=1}^{n_\mathrm{s}} A'_{i,\gamma} + \sum_{j=1}^{n_\mathrm{u}} B'_{j,\gamma},$$
$$|\mathcal{C}_\gamma(X'')| = \sum_{i=1}^{n_\mathrm{s}} A''_{i,\gamma} + \sum_{j=1}^{n_\mathrm{u}} B''_{j,\gamma}.$$

For the rest of this proof we will simplify the notation as follows: 1) γ is omitted in the subscript because we can assume it to be the same throughout all considerations, 2) the limits in the sums are omitted; for stable solutions we always sum up to n_s and for unstable solutions to n_u, and 3) by F_s and F_u we denote the cumulative density functions of stable and unstable distributions, respectively, evaluated at $m + \gamma$:

$$F_\mathrm{s} \coloneqq F_\mathrm{s}(m + \gamma), \qquad F_\mathrm{u} \coloneqq F_\mathrm{u}(m + \gamma). \tag{3.55}$$

Observe that 1) $\mathbb{E}[A'_i] = \mathbb{E}[A''_i] = F_\mathrm{s}$ and $\mathbb{E}[B'_j] = \mathbb{E}[B''_j] = F_\mathrm{u}$, 2) the random variables in $\{A'_i\}_i \cup \{A''_j\}_j \cup \{B'_k\}_k \cup \{B''_\ell\}_\ell$ are jointly independent, and 3) $(A'_i)^2 = A'_i$, $(A''_i)^2 = A''_i$, $(B'_j)^2 = Y_j$ and $(B''_j)^2 = Y_j$ because these are indicators. Also, remember that for jointly independent indicator random variables Z_1, Z_2, Z_3 with $\mathbb{E}[Z_i] = z_i$ we have

$$\mathrm{Cov}(Z_1, Z_2) = z_1 z_2 (1 - z_1 z_2) \tag{3.56}$$
$$\mathrm{Cov}(Z_1 Z_2, Z_1 Z_3) = z_1 z_2 z_3 (1 - z_1) \tag{3.57}$$

2) *Taylor expansion of the expected similarity.* A second-order Taylor approximation of $\mathbb{E}[V/W]$ gives

$$\mathbb{E}\left[\frac{V}{W}\right] \approx \frac{\mathbb{E}[V]}{\mathbb{E}[W]} - \frac{\mathrm{Cov}(V, W)}{\mathbb{E}[W]^2} + \frac{\mathrm{Var}[W] \cdot \mathbb{E}[V]}{\mathbb{E}[W]^3}. \tag{3.50}$$

Remember that V denotes the size of the intersection while W is the product of the approximation set sizes. In the following, we will analyze each term of (3.50) separately.

3) *Expected values of V and W.*

$$\mathbb{E}[V] = \sum_i \mathbb{E}[A'_i] \cdot \mathbb{E}[A''_i] + \sum_j \mathbb{E}[B'_j] \cdot \mathbb{E}[B''_j] = n_\mathrm{s} F_\mathrm{s}^2 + n_\mathrm{u} F_\mathrm{u}^2. \tag{3.51}$$

Taking the independence of the random variables into account, for $\mathbb{E}[W]$ we obtain

$$\mathbb{E}[W] = \mathbb{E}\Big[\sum_i A'_i + \sum_j B'_j\Big] \cdot \mathbb{E}\Big[\sum_i A''_i + \sum_j B''_j\Big] = (n_\mathrm{s} F_\mathrm{s} + n_\mathrm{u} F_\mathrm{u})^2. \tag{3.52}$$

4) *Analyzing the covariance of V and W.* Remember that

$$V = \sum_i A'_i A''_i + \sum_i B'_i B''_i, \tag{3.58}$$

$$W = \sum_{j,k} A'_j A''_k + \sum_{i,j} A'_j B''_k + \sum_{j,k} B'_j A''_k + \sum_{j,k} B'_j B''_k, \tag{3.59}$$

hence

$$\begin{aligned}
\mathrm{Cov}(V,W) = &\sum_{i,j,k} \mathrm{Cov}(A'_i A''_i, A'_j A''_k) + \sum_{i,j,k} \mathrm{Cov}(A'_i A''_i, A'_j B''_k) \\
&+ \sum_{i,j,k} \mathrm{Cov}(A'_i A''_i, B'_j A''_k) + \sum_{i,j,k} \mathrm{Cov}(A'_i A''_i, B'_j B''_k) \\
&+ \sum_{i,j,k} \mathrm{Cov}(B'_i B''_i, A'_j A''_k) + \sum_{i,j,k} \mathrm{Cov}(B'_i B''_i, A'_j B''_k) \\
&+ \sum_{i,j,k} \mathrm{Cov}(B'_i B''_i, B'_j A''_k) + \sum_{i,j,k} \mathrm{Cov}(B'_i B''_i, B'_j B''_k)
\end{aligned} \tag{3.60}$$

We will now analyze each of the single terms.

- In the first term $\sum \mathrm{Cov}(A'_i A''_i, A'_j A''_k)$ only the summands with $j = i$ or $k = i$ are non-zero, hence we obtain

$$\begin{aligned}
\sum_{i,j,k} \mathrm{Cov}(A'_i A''_i, A'_j A''_k) &= \sum_i \mathrm{Cov}(A'_i A''_i, A'_i A''_i) \\
&\quad + \sum_{i \neq j} \Big[\mathrm{Cov}(A'_i A''_i, A'_i A''_j) + \mathrm{Cov}(A'_i A''_i, A'_j A''_i)\Big] \\
&= \sum_i \mathrm{Cov}(A'_i A''_i, A'_i A''_i) + 2 \sum_{i \neq j} \mathrm{Cov}(A'_i A''_i, A'_i A''_j),
\end{aligned}$$

$$= n_{\mathrm{s}} F_{\mathrm{s}}^2 (1 - F_{\mathrm{s}}^2) + 2n_{\mathrm{s}}(n_{\mathrm{s}} - 1) F_{\mathrm{s}}^3 (1 - F_{\mathrm{s}}), \tag{3.61}$$

where the last equality holds due to (3.56) and (3.57).

- The next two terms $\sum \mathrm{Cov}(A_i' A_i'', A_j' B_k'')$ and $\sum \mathrm{Cov}(A_i' A_i'', B_j' A_k'')$ are equal to each other (due to the symmetry of A' and A''), so their sum resolves to

$$2 \sum_{i,k} \mathrm{Cov}(A_i' A_i'', A_i' B_k'') \overset{(3.57)}{=} 2n_{\mathrm{s}} n_{\mathrm{u}} F_{\mathrm{s}}^2 F_{\mathrm{u}} (1 - F_{\mathrm{s}}). \tag{3.62}$$

- The next two terms $\sum \mathrm{Cov}(A_i' A_i'', B_j' B_k'')$ and $\sum \mathrm{Cov}(B_i' B_i'', A_j' A_k'')$ are both zero due to the independence of $A_i' A_i''$ and $B_j' B_k''$.
- The next two terms $\sum \mathrm{Cov}(B_i' B_i'', A_j' B_k'')$ and $\sum \mathrm{Cov}(B_i' B_i'', B_j' A_k'')$ can be computed in exactly the same way as (3.62) where both F_{s} and F_{u} as well as n_{s} and n_{u} are interchanged. Hence, their sum equals

$$2 \sum_{i,k} \mathrm{Cov}(B_i' B_i'', B_i' A_k'') \overset{(3.57)}{=} 2n_{\mathrm{s}} n_{\mathrm{u}} F_{\mathrm{s}} F_{\mathrm{u}}^2 (1 - F_{\mathrm{u}}).$$

- The last term $\sum \mathrm{Cov}(B_i' B_i'', B_j' B_k'')$ is computed similar as (3.61), performing the above-mentioned replacements, hence

$$\sum_{i,j,k} \mathrm{Cov}(B_i' B_i'', B_j' B_k'') = n_{\mathrm{u}} F_{\mathrm{u}}^2 (1 - F_{\mathrm{u}}^2) + 2n_{\mathrm{u}}(n_{\mathrm{u}} - 1) F_{\mathrm{u}}^3 (1 - F_{\mathrm{u}}).$$

5) *Analyzing the variance of W.* Finally we compute $\mathrm{Var}[W] = \mathrm{Cov}(W, W)$. When W is expressed as (3.59), we obtain

$$\begin{aligned}
\mathrm{Cov}(W, W) = &\sum_{i,j,k,\ell} \mathrm{Cov}(A_i' A_j'', A_k' A_\ell'') + \sum_{i,j,k,\ell} \mathrm{Cov}(A_i' B_j'', A_k' B_\ell'') \\
&+ \sum_{i,j,k,\ell} \mathrm{Cov}(B_i' A_j'', B_k' A_\ell'') + \sum_{i,j,k,\ell} \mathrm{Cov}(B_i' B_j'', B_k' B_\ell'') \\
&+ 2 \sum_{i,j,k,\ell} \mathrm{Cov}(A_i' A_j'', A_k' B_\ell'') + 2 \sum_{i,j,k,\ell} \mathrm{Cov}(A_i' A_j'', B_k' A_\ell'') \\
&+ 2 \sum_{i,j,k,\ell} \mathrm{Cov}(A_i' A_j'', B_k' B_\ell'') + 2 \sum_{i,j,k,\ell} \mathrm{Cov}(A_i' B_j'', B_k' A_\ell'') \\
&+ 2 \sum_{i,j,k,\ell} \mathrm{Cov}(A_i' B_j'', B_k' B_\ell'') + 2 \sum_{i,j,k,\ell} \mathrm{Cov}(B_i' A_j'', B_k' B_\ell'').
\end{aligned}$$

As before we analyze each of these terms separately.

- The first term $\sum \mathrm{Cov}(A'_i A''_j, A'_k A''_\ell)$ can be expressed as

$$\sum_i \mathrm{Cov}(A'_i A''_i, A'_i A''_i) + 4 \sum_{i \neq j} \mathrm{Cov}(A'_i A''_i, A'_i A''_j) + 2 \sum_{i \neq j, i \neq k, j \neq k} \mathrm{Cov}(A'_i A''_j, A'_i A''_k) + \sum_{i \neq j} \mathrm{Cov}(A'_i A''_j, A'_i A''_j)$$

where

$$\begin{aligned}
\sum_i \mathrm{Cov}(A'_i A''_i, A'_i A''_i) &= n_\mathrm{s} F_\mathrm{s}^2 (1 - F_\mathrm{s}^2), \\
4 \sum_{i \neq j} \mathrm{Cov}(A'_i A''_i, A'_i A''_j) &= 4 n_\mathrm{s} (n_\mathrm{s} - 1) F_\mathrm{s}^3 (1 - F_\mathrm{s}), \\
2 \sum_{i \neq j, i \neq k, j \neq k} \mathrm{Cov}(A'_i A''_j, A'_i A''_k) &= 2 n_\mathrm{s} (n_\mathrm{s} - 1)(n_\mathrm{s} - 2) F_\mathrm{s}^3 (1 - F_\mathrm{s}), \\
\sum_{i \neq j} \mathrm{Cov}(A'_i A''_j, A'_i A''_j) &= n_\mathrm{s} (n_\mathrm{s} - 1) F_\mathrm{s}^2 (1 - F_\mathrm{s}^2),
\end{aligned}$$

and therefore

$$\sum_{i,j,k,\ell} \mathrm{Cov}(A'_i A''_j, A'_k A''_\ell) = n_\mathrm{s} F_\mathrm{s}^2 (1 - F_\mathrm{s}^2) + 2 n_\mathrm{s}^2 (n_\mathrm{s} - 1) F_\mathrm{s}^3 (1 - F_\mathrm{s}). \quad (3.63)$$

- The next two terms $\sum \mathrm{Cov}(A'_i B''_j, A'_k B''_\ell)$ and $\sum \mathrm{Cov}(B'_i A''_j, B'_k A''_\ell)$ are equal due to the symmetry in instances, hence their sum equals

$$2 \sum_{\substack{i \\ j \neq k}} \mathrm{Cov}(A'_i B''_j, A'_i B''_k) + 2 \sum_{\substack{i \\ j \neq k}} \mathrm{Cov}(A'_j B''_i, A'_k B''_i) + 2 \sum_{i,j} \mathrm{Cov}(A'_i B''_j, A'_i B''_j),$$

where the the terms are computed as

$$\begin{aligned}
2 \sum_{\substack{i \\ j \neq k}} \mathrm{Cov}(A'_i B''_j, A'_i B''_k) &\overset{(3.57)}{=} 2 n_\mathrm{s} n_\mathrm{u} (n_\mathrm{u} - 1) F_\mathrm{s} F_\mathrm{u}^2 (1 - F_\mathrm{s}), \\
2 \sum_{\substack{i \\ j \neq k}} \mathrm{Cov}(A'_j B''_i, A'_k B''_i) &\overset{(3.57)}{=} 2 n_\mathrm{s} (n_\mathrm{s} - 1) n_\mathrm{u} F_\mathrm{s}^2 F_\mathrm{u} (1 - F_\mathrm{u}),
\end{aligned}$$

$$2\sum_{i,j}\mathrm{Cov}(A'_iB''_j, A'_iB''_j) \overset{(3.56)}{=} 2n_\mathrm{s}n_\mathrm{u}F_\mathrm{s}F_\mathrm{u}(1-F_\mathrm{s}F_\mathrm{u}).$$

- The next term $\sum \mathrm{Cov}(B'_iB''_j, B'_kB''_\ell)$ is computed analogically to (3.63) where stable and unstable solutions are interchanged, resulting in

$$\sum_{i,j,k,\ell}\mathrm{Cov}(B'_iB''_j, B'_kB''_\ell) = n_\mathrm{u}^2F_\mathrm{u}^2(1-F_\mathrm{u}^2) + 2n_\mathrm{u}^2(n_\mathrm{u}-1)F_\mathrm{u}^3(1-F_\mathrm{u}).$$

- The next terms $2\sum \mathrm{Cov}(A'_iA''_j, A'_kB''_\ell)$ and $2\sum \mathrm{Cov}(A'_iA''_j, B'_kA''_\ell)$ are equal due to the symmetry of the instances, hence their sum is

$$\begin{aligned}4\sum_{i,j,k,\ell}\mathrm{Cov}(A'_iA''_j, A'_kB''_\ell) &= 4\sum_{i,j,k}\mathrm{Cov}(A'_iA''_j, A'_iB''_k)\\ &\overset{(3.57)}{=} 4n_\mathrm{s}^2n_\mathrm{u}F_\mathrm{s}^2F_\mathrm{u}(1-F_\mathrm{s}). \qquad (3.64)\end{aligned}$$

- The next terms $2\sum \mathrm{Cov}(A'_iA''_j, B'_kB''_\ell)$ and $2\sum \mathrm{Cov}(A'_iB''_j, B'_kA''_\ell)$ are both equal to zero due to the independence of $A'_iA''_j$ and $B'_kB''_\ell$, and of $A'_iB''_j$ and $A'_kA''_\ell$.
- The last terms $2\sum \mathrm{Cov}(A'_iB''_j, B'_kB''_\ell)$ and $2\sum \mathrm{Cov}(B'_iA''_j, B'_kB''_\ell)$ are equal due to the symmetry the instances hence their sum can be computed analogically to (3.64) where stable and unstable solutions are interchanged. Hence, we obtain

$$4\sum_{i,j,k,\ell}\mathrm{Cov}(B'_iA''_j, B'_kB''_\ell) = 4n_\mathrm{s}n_\mathrm{u}^2F_\mathrm{s}F_\mathrm{u}^2(1-F_\mathrm{u}).$$

The proof is thus finished. □

3.7.2 Experimental Results

We now provide both positive and negative experimental results which highlight the scope of applicability of such similarity estimation. We performed an experimental evaluation using Gaussian noise in a setting similar to the one in Sections 3.6.2–3.6.4: parameters were set to $n_\mathrm{s} = 100$, $n_\mathrm{u} = 900$, $\mu_\mathrm{s} = 1$, $\sigma_\mathrm{s} = 1$, $\mu_\mathrm{u} = 10$, $\sigma_\mathrm{u} \in \{0, 0.1, \ldots, 10\}$. The only adjustment on had to make was a slightly changed instance generator due to the calibrating assumption (3.48): since the minima of both instances have to be sufficiently close to each other, the problem generation process disregarded each pair of instances for which the minima m' and

m'' differed by more than $\varepsilon = 10^{-4}$, and repeatedly generated a new pair until $|m' - m''| \leq \varepsilon$.

For each successful (i.e. not rejected due to calibrating assumption) instance pair (X', X''), we computed similarity (3.37) and estimated similarity (3.50), where the latter was calibrated with $m = (m'+m'')/2$. We repeated the process $r = 1000$ times and calculated the average similarity

$$\overline{S}_\gamma = \frac{1}{r} \sum_{k=1}^{r} S_\gamma(X', X''), \tag{3.65}$$

and compared it to the estimated similarity (3.50). We note that we did not compute the average estimated similarity over all instance pairs, but instead calibrated Equation (3.50) directly using the average minimum cost of the instance pairs, i.e., using $m = \frac{1}{r} \sum_{k=1}^{r} (m'^k + m''^k)/2$ where $m'^k = \min_{c \in \mathcal{C}} R(c, X'^k)$ and m'' is defined respectively.

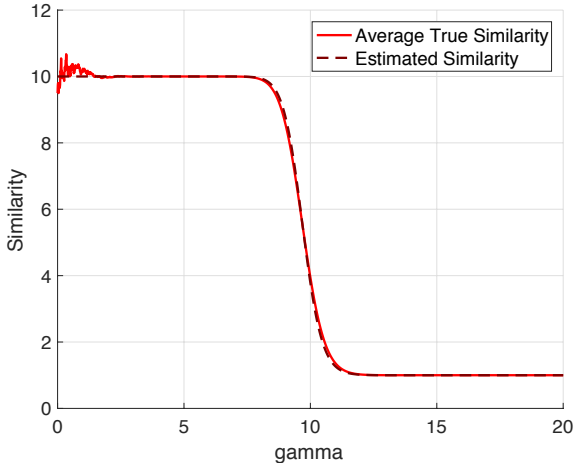
Figure 3.10 shows the plots of $\widehat{S}_\gamma$ and $\overline{S}_\gamma$ defined above for two noise levels: $\sigma_{\mathrm{u}} = 1$ and for $\sigma_{\mathrm{u}} = 5$. We see that the estimated similarity matches the average similarity relatively well, especially for larger values of γ. Although the discrepancy grows with the noise (which is natural due to the Taylor expansion used in the proof), probably the most important thing to note is that the positions of the γ^* computed based on $\widehat{S}_\gamma$ and $\overline{S}_\gamma$ remain the same.

To summarize, we considered the expected similarity of two instances from the same generator, and we derived an estimation for it that only depends on the number of stable and unstable solutions, and on the respective cumulative density functions. Our experiments showed that our estimation approximates the expected similarity well when the noise is not too low. Our experiments also showed that the γ^* that maximizes the estimated similarity does indeed help to identify stable solutions. In particular, choosing a solution from the intersection of the corresponding γ^*-approximation is a promising way of robust solving. One of the possible steps in this direction should be to analyze how many stable and how many unstable solutions this intersection contains in expectation.

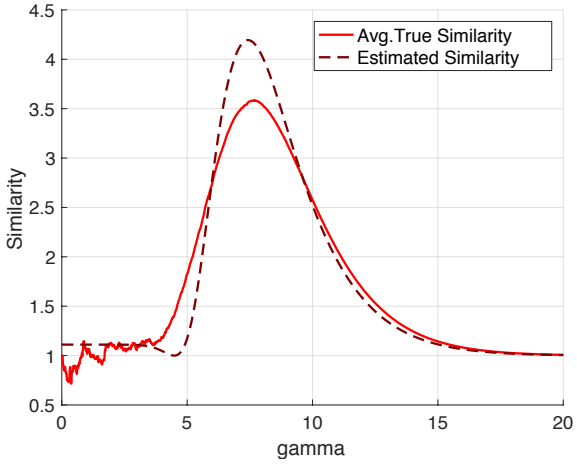
3.8 Gibbs Relaxation of the Approximation Set-Based Approach

3.8.1 Approximation Sets with Gibbs Weights

Buhmann (2010a), in addition to the approximation set-based approach, introduced its Gibbs-relaxed version which we give in this section. The idea (we adapt it for the sake of notation alignment with the material of this chapter) is as



(a)



(b)

Figure 3.10 Average vs. estimated similarity for $\sigma_{\mathrm{u}} = 1$ **(a)**, and for $\sigma_{\mathrm{u}} = 5$ **(b)**.

follows: using the maximum entropy principle (Section 2.3.4) by Jaynes (1982) from statistical physics (see also Mezard and Montanari, 2009), for a real number $\beta \geq 0$, an instance X and a solution c, the Gibbs weight of c is defined as $w_\beta^G(c, X) \coloneqq \exp(-\beta R(c, X))$. Now one computes a value β^* that maximizes

$$\beta^* = \arg\max_{\beta > 0} \log\left(|\mathcal{C}| \frac{\sum_{c \in \mathcal{C}} \left(w_\beta^G(c, X') \cdot w_\beta^G(c, X'') \right)}{\left(\sum_{c \in \mathcal{C}} w_\beta^G(c, X') \right) \cdot \left(\sum_{c \in \mathcal{C}} w_\beta^G(c, X'') \right)} \right), \tag{3.66}$$

or, since the optimization goal is the same (see remark after (3.41)), maximizes the ratio

$$\beta^* = \arg\max_{\beta>0} \frac{\sum_{c\in\mathcal{C}} \big(w_\beta^G(c,X') \cdot w_\beta^G(c,X'')\big)}{\big(\sum_{c\in\mathcal{C}} w_\beta^G(c,X')\big) \cdot \big(\sum_{c\in\mathcal{C}} w_\beta^G(c,X'')\big)}, \tag{3.67}$$

and then samples a solution c from the whole solution space $\mathcal{C}$ with probability

$$p_\beta(c) = \frac{w_{\beta^*}^G(c,X') \cdot w_{\beta^*}^G(c,X'')}{\sum_{c'\in\mathcal{C}}(w_{\beta^*}^G(c',X') \cdot w_{\beta^*}^G(c',X''))}.$$

We refer to this as the *Gibbs relaxation of approximation set-based approach.*

3.8.2 Relation of Similarity and Gibbs Similarity

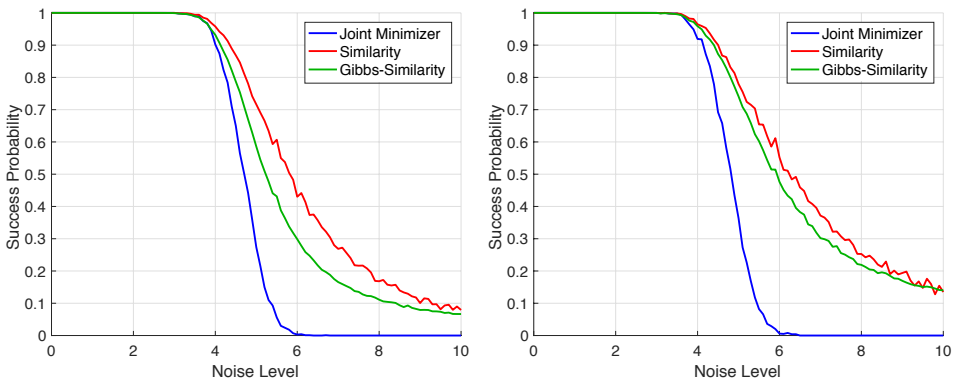
Interestingly, the classical approximation set-based approach (3.41) and its Gibbs relaxation (3.67) have a clear relation: for a number $\gamma \geq 0$, an instance X and a solution c we define a 0-1-weight $w_\gamma^{\mathbb{1}}(c,X)$ that is 1 if and only if $R(c,X) \leq R(c^\perp,X) + \gamma$, and 0 otherwise. It is easy to see that

$$\begin{aligned} |\mathcal{C}_\gamma(X')| &= \sum_{c\in\mathcal{C}} w_\gamma^{\mathbb{1}}(c,X') \\ |\mathcal{C}_\gamma(X'')| &= \sum_{c\in\mathcal{C}} w_\gamma^{\mathbb{1}}(c,X'') \\ |\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')| &= \sum_{c\in\mathcal{C}} \big(w_\gamma^{\mathbb{1}}(c,X') \cdot w_\gamma^{\mathbb{1}}(c,X'')\big). \end{aligned} \tag{3.68}$$

With these equalities it follows that the objective of maximizing $S_\gamma(X',X'')$ in (3.41) corresponds to the one of (3.67) in which the 0-1-weights $w_\gamma^{\mathbb{1}}$ are substituted for the Gibbs weights w_β^G. Moreover, notice that $w_\gamma^{\mathbb{1}}(c,X') \cdot w_\gamma^{\mathbb{1}}(c,X'') = 1$ if and only $c \in \mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')$. Hence, sampling a solution from $\mathcal{C}$ with a probability proportional to $w_{\gamma^*}^{\mathbb{1}}(c,X') \cdot w_{\gamma^*}^{\mathbb{1}}(c,X'')$ corresponds to sampling a solution from $\mathcal{C}_{\gamma^*}(X') \cap \mathcal{C}_{\gamma^*}(X'')$ uniformly at random.

Similar to the parameter γ in Equation (3.37), the parameter β (called "inverse temperature" in statistical physics[6]) controls the amount of solutions that are taken into account. For $\beta = 0$, all solutions have the same weight 1 (corresponding to the case $\gamma = \infty$ in which the intersection contains every solution in $\mathcal{C}$), while for $\beta \to \infty$ the distribution concentrates on the solutions with the minimum joint cost. Hence, the parameter β in (3.67) is by its semantics an "inverse" to the

[6] Much more on that will be given in Chapter 5.



(a) 5% of solutions are stable. **(b)** 10% of solutions are stable.

Figure 3.11 Gibbs relaxation shows almost the same performance. Experimental results where 5% **(a)** and 10% **(b)**. Model and setting are the same as in Section 3.6.

parameter γ in (3.37).

3.8.3 Experimental Results

Since the Gibbs relaxation chooses every solution $c \in \mathcal{C}$ with a probability proportional to $w^G_{\beta^*}(c, X') \cdot w^G_{\beta^*}(c, X'')$, we define its success probability as

$$P^G_{\mathscr{A}}(X', X'') := \frac{\sum_{c \in \mathcal{C}_{\text{stable}}} w^G_{\beta^*}(c, X') \cdot w^G_{\beta^*}(c, X'')}{\sum_{c \in \mathcal{C}} w^G_{\beta^*}(c, X') \cdot w^G_{\beta^*}(c, X'')} \tag{3.69}$$

where β^* is the value β that maximizes (3.67). Notice that the sums in the numerator and denominator are computed over different sets of solutions. Notice that this formula is a full analogy of (3.45).

Figure 3.11 shows, under the same setting as in Section 3.6, that the Gibbs relaxation provides yields almost the same performance and thus can be considered as a viable variant of the approximation set-based approach. This idea will be massively exploited in Chapter 5.

3.9 Discussion and Conclusion

In this chapter, we introduced an approximation set-based approach to robust optimization and justified it via a so-called Approximation Set Coding which provides an information-theoretic background. Below, we will elaborate on some points which are, in our view, highlighting its most interesting and/or controversial prop-

erties.

Role of the logarithm

Consider the comparison between the empirical ASC score (3.28)

$$\log \frac{|\mathcal{C}|\ |\Delta\mathcal{C}_\gamma(X', X'')|}{|\mathcal{C}_\gamma(X')|\ |\mathcal{C}_\gamma(X'')|}$$

and the empirical similarity score (3.37):

$$\frac{|\mathcal{C}||\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{|\mathcal{C}_\gamma(X')||\mathcal{C}_\gamma(X'')|}.$$

Note that there is a difference in putting the logarithm in front of the ASC score. Although both have their maxima at the same γ, this logarithm will be of essential importance later in Chapter 5 so it is instructive to explain this difference.

The numerator and the denominator of the score can be seen as the alternative and the null hypothesis, respectively, in statistical hypothesis testing — the likelihood ratio test. One can view the usage of logarithm of the likelihood ratio as a tool for ensuring asymptotic normality of estimators in the case of weak coupling. For the main objective of this chapter, using the logarithm had no special implications, since we were interested in the γ^* which maximized score, but not in the score itself.

We should also note that the coding argument which we brought when deriving ASC implies that logarithm allows to quantify the informativeness/capacity using *bits* (or *nats*, depending on the type of the logarithm in use). This is turns the ASC score into the one which allows interpretable value — i.e. the one answering "how many bits of information can the model extract".

Is the way of defining approximation unique?

As one can see from the material of this chapter, the whole approximation set-based approach rests on *some notion* of closeness of the given solution to the optimal one ($c^\perp$). We quantified this notion in terms of parameters γ or (in case of Gibbs approximation) β. But there exists a whole zoo of other possible parametrizations, for example, parametrizing by the step t of a stepwise algorithm. However, we advocate the point of view that such parametrization should yield a local topology around each solution according to the following informal procedure:

1. define certain measure of local closeness of solutions around the optimal;

2. make an assumption: each solution *is* the optimal solution for some input;
3. local approximation topologies induced by the above create a "cover" of the whole solution space;
4. derive conditions under which such a covering by local topology is can be turned into metric space (metrization theorems);
5. the above allows to create a uniform (i.e. non-local) closeness relation.

This high-level roadmap gives some insight into the final goal of such a journey: understand the structure of solution space in a problem-specific manner.

Are all solutions in the intersection created equal?

Our method expects all solutions in the best approximation set intersection to be equally desirable (e.g., equally good for a third, unknown instance). In some cases, it might be useful to choose the solution based on some problem specific criterion, e.g., choose the solution closest to the centroid of the intersection set.

Will more input lead to better results?

We mostly studied the two instance scenario because this is the minimum number of instances necessary to distinguish information from noise. Often, however, more than two instances are available. The extension to multiple instances is not immediately obvious.

There are several ways of addressing it: (a) first, one can break it into pairs and average. This is how the framework is intended to be used in practice; (b) second, one can derive a version of ASC for multiple agents. The latter approach sounds much more interesting from the research prospective, as it is not clear what would be the channel analogy in case of several agents (remember, in the two instance scenario, we considered one data point as a codebook benchmark, and the other as error problem generator). On can as well go in the direction of a straightforward generalization of the similarity formula (3.28). In the course of our research, some attempts have been made in that direction and they yielded promising results.

Can we find efficient algorithmic solutions?

The remark after Theorems 3.1–3.4 tells that one of the pitfalls of approximation set-based approach consists in computation of the similarity score. While we used brute-force enumeration for our proof-of-concept experiments, it would be of a

great importance to find either (a) analytical estimations for the similarity score or (b) efficient algorithms for computing it.

In this chapter, we tackled case (a) and made an attempt to derive a very simple analytical estimator, which uses the knowledge of the true distributions. This assumption, of course, renders it useless in real cases, but allows usage of plug-in estimators of the true distributions.

On a much higher level which uses less information about the true distributions, the approach (a) will be tackled in Chapter 5.

In specific cases, such as application to combinatorial algorithms, the approach (b) can be used by utilizing combinatorial structure of the solutions. This will be shown in Chapter 4.

Similarity as a computational goal-specific measure

An interesting side-result that we did not focus on in this chapter is the expressiveness of instance similarity S_{γ^*}. In fact, it utilizes a *computational goal-induced* topology on the set of solutions. We bring here a motivation which was best described in (Buhmann et al., 2017a). For example, consider the problem of computing a shortest path between two given vertices in a graph G. Having two instances X' and X'' of this problem, one may attempt to measure the similarity of these instances using certain structural information exposed to us — e.g., the correlation coefficient or the Euclidean distance between the vectors containing the edge weights.

However, if the instances differ a lot only in some weights which are usually high and thus these edges that are never used in any nearly-shortest path, then the similarity approach will correctly consider such examples as similar, whereas for example the correlation coefficient will tell the opposite. At the same time, if the computational goal was a maximum matching of edges rather than the minimizing the weight cost, the similarity would regard the two instances as significantly different. This example highlights the need for a measure of *similarity of instances with respect to a computational goal.* This is performed by inducing a local topology around each solution, and this topology depends only on the computational goal and not on anything else.

4

Minimum Spanning Tree Algorithms: Regularization by Stopping

"Besser ein Spatz in der Hand, als eine Taube auf dem Dach."
(germ. "A bird in the hand is worth two in the bush.")

— Proverbs

4.1 Introduction

4.1.1 Motivation and Examples

Noise perturbed combinatorial optimization problems arise in various real-world applications where problem instances are abstracted by weighted[1] graphs with fluctuations in the weights. In this chapter, we investigate the noisy Minimum Spanning Tree (MST) problems and their ability to infer robust spanning trees. Examples of real-world applications of minimum spanning trees come from various fields of human activity, such as *communication networks with delays* (optimizing message delivery times, cf. (Bertsekas and Gallager, 1992)) or *stock markets* (analyzing stock exchange correlations, cf. (Sandoval, 2012)), etc. All the applications require trees with a high level of robustness to fluctuations since the quality of trees is measured by expected costs. Wide real-world demand and applicability stimulated extensive research on the robust minimum spanning trees problem (Aron and Hentenryck, 2004; Kozina and Perepelitsa, 1994; Sandoval, 2012; Yaman et al., 2001), which addressed different aspects of the robust spanning tree setting, such as development of algorithms, measuring algorithmic complexity, or comparing robustness criteria.

[1] We will further utilize both the terms "costs" and "weights" in the same meaning.

In this chapter, we use an information-theoretic regularization approach introduced in Chapter 3 to analyse and validate MST algorithms from the point of view of how well they can recognize (localize) the true Minimum Spanning Tree under unknown noise in the graph instance. The validation concept is developed for "contractive" algorithms that follow a step-by-step strategy of shrinking the solution space until a single best solution is identified. This validation approach is based on the *two-instance scenario* (Vapnik, 1982) that requires statistical estimates to generalize to test instances when the estimates have been inferred from a training instance. For spanning trees, that means that they have to yield low costs on at least two problem instances without being explicitly adapted to the test instance.Such regularization remains in the spirit of the *information bottleneck method* (Tishby et al., 1999) in the sense that it tries to optimize the amount of information that the algorithm might transfer from an artificial channel (see below, Section 4.3.3).

4.1.2 Contributions and Outline of the Chapter

As main contributions of this chapter, we

- define a notion of contractive algorithm and give an general extension of the ASC regularization approach (introduced earlier in Chapter 3) for the case of stepwise contractive algorithmic problems. We call it Algorithmic ASC;
- provide a specialization of such extension to the case of three major algorithms solving the Minimum Spanning Tree (MST) problem, including the necessary tools to avoid brute-force enumeration known as a computational bottleneck of ASC;
- carry out experiments which justify usage of Algorithmic ASC score as a ranking tool related to expected localization error: higher Algorithmic ASC score yields lower error.

The chapter is outlined as follows. First, a related work overview is given in Section 4.2. Then, in Section 4.2 we revisit those ingredients of ASC approach relevant for this chapter. A comprehensive introduction into the original approximation set-based approach is then given in Section 3.4. Later, the main contribution is made in Section 4.4, where we make a generalization of the ASC to algorithmic problems, and Section 4.5 where we directly apply it to Minimum Spanning Tree (MST) problem and three most important algorithms solving it. Experimental results follow in Section 4.6. Finally, we discuss our findings in Section 4.7.

4.2 Related Work Overview

Literature entries on robust spanning trees vary by research purpose (e.g. algorithm complexity, robustness), adopted noise model (e.g. interval data), regularization strategy (regularizing by pruning, graph preprocessing). For example, (Yaman et al., 2001) investigated graphs with edge weights which are supposed to fall uniformly into a predefined interval. The approach is based on regularization *by preprocessing the graph*, which, in turn, involves eliminating selected edges and forcing other edges to be included into the spanning tree.

Regularizing an MST by *pruning* as proposed by (Sandoval, 2012) identifies the presumably robust part of a spanning tree. This goal is achieved by means of analyzing the survival matrix constructed from adjacency matrices for training minimum spanning trees.

Our approach is similar in that pruning (more precisely, optimal stopping) is used as a regularization tool, but it rather aims at *analysis and comparison* of the algorithm's robustness by means of such regularization. This approach also exhibits technical differences from the mentioned work, relying only on *two* data instances, and it follows the spirit of learning theory with its goal to analyse algorithms in a noise distribution independent way.

In Section 4.3, the general concepts of the information-theoretic framework are given, as well as the information-theoretic motivation. Then, in Section 4.4, the application to stepwise algorithms is discussed. The results and conclusion are summarized in Sections 4.6 and 4.7, respectively.

4.3 Approximation Set Coding Regularization

To make this chapter self-consistent, we first give a short overview of the approach first introduced in Chapter 3, as well as of the notation and terms used throughout this chapter.

4.3.1 Notation and Definitions

Assume, there is a data instance X given, for example measurements in a data space $\mathcal{X}$, so that $X \in \mathcal{X}$. In case of the MST problem, $\mathcal{X}$ is the set of all possible combinations of the edge costs and X is a particular realization of such costs. Let c denote a solution in a general set of solutions $\mathcal{C}$, so that $c \in \mathcal{C}$. In case of the MST, $\mathcal{C}$ is the set of feasible spanning trees.

An optimization problem is defined by a cost (objective) function $R : \mathcal{C} \times \mathcal{X} \to \mathbb{R}_+$ that assigns each solution c a real value $R(c, X)$. Furthermore, $c^{\perp}(X) \in$

$\arg\min_{c\in\mathcal{C}} R(c, X)$ denotes[2] the minimum cost solution.

For the sake of referring to it in the rest of the chapter, we give here a definition from Chapter 3:

Definition 4.1. *For a given real number $\gamma \geq 0$, an approximation set is defined as follows:*

$$\mathcal{C}_\gamma(X, R) \coloneqq \{c \in \mathcal{C} \mid R(c, X) - R^\perp(X) \leq \gamma\}, \tag{4.1}$$

In fact, it is desirable to find a *robust set* of solutions rather than a single solution, since we cannot trust the global minimizer $c^\perp$ to achieve low costs on test instance due to noise in the measurements. In this light, the sets $\mathcal{C}_\gamma(X)$ play the role of a "trade-off agent" in the process of finding the robust solution to optimization problem. This trade-off is balancing the solution sets between overfitting ($\gamma = 0$) and underfitting (large γ).

4.3.2 Robust Solving via ASC Regularization

Regularization by the Approximation Sets applies to a situation, when two data instances $X', X'' \in \mathcal{X}$ are available and when both are generated by injecting two noise realizations into the true data instance $X^0 \in \mathcal{X}$ (see Section 3.3.2). It is required for both cases that the noise follows the same noise distribution: $X', X'' \sim PG(X^0)$. In the spirit of statistical learning theory, we are interested in distribution independent results since the distribution $PG(X^0)$ might be unknown.

A robust solution to the noisy optimization problem $(\mathcal{X}, \mathcal{C}, R)$ is obtained by the two-step process described in Algorithm 4.1.

Algorithm 4.1: Robust Solving via ASC Regularization

Data:
two instances of the X', X'',
cost function $R(c, X)$
Result: Solution $c \in \mathcal{C}$.

1 Find the optimal regularization parameter γ^* that maximizes the log-ratio

$$\widehat{I}_\gamma(X', X'') \coloneqq \log\Big(\frac{|\mathcal{C}|\ |\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{|\mathcal{C}_\gamma(X')|\ |\mathcal{C}_\gamma(X'')|}\Big), \tag{4.2}$$

2 Choose a solution uniformly at random from the intersection of optimal approximation sets: $c \in C_{\gamma^*}(X') \cap C_{\gamma^*}(X'')$.

[2] In the context where it is clear what X is, we will omit the "(X)" in the notation: $c^\perp(X) \equiv c^\perp$; $R(c, X) \equiv R(c)$; $\mathcal{C}_\gamma(X) \equiv \mathcal{C}_\gamma$.

Remark. Note the difference of (4.2) with previously defined (3.28): dropping the expectation. This is perfectly legitimate in case when we have no access to the problem generation distribution and have to replace the actual value by its estimator.

4.3.3 Information-Theoretic Basis for ASC Regularization

Applying of ASC bases itself on the information-theoretic ground which uses approximation sets $\mathcal{C}_\gamma(X)$ in a fictitious communication scenario. This communication scenario involves sending and receiving a *transformation* $\tau\colon \mathcal{X} \to \mathcal{X}$ that maps the data space onto itself. We briefly recap the necessary information here: for a full introduction, refer to Chapter 3 (Section 3.4.2).

Communicating a true τ_{send} is performed by sending $\tau_{\text{send}} \circ X'$ to the receiver, while the channel perturbs this "message" by substituting X' with X''. Finally, the receiver obtains $\tau_{\text{send}} \circ X''$. The reader should note that X' is known to both sender and receiver.

As the receiver accepts $\tau_{\text{send}} \circ X''$, it has has to distinguish the fluctuations in X'' relative to X' from the applied transformation τ_{send}. Error free communication is guaranteed if the receiver can identify the correct transformation τ_{send} without being deceived by these fluctuations. The straightforward decoding rule consists in finding the "nearby" transformation $\widehat{\tau}$, which maximizes the overlap between the received approximation set $\mathcal{C}_\gamma(\tau_{\text{send}} \circ X'')$ and the "nearby" approximation set $\mathcal{C}_\gamma(\widehat{\tau} \circ X')$.

The role of the codebook vectors is played by all such transformations which enable us to cover the complete solution space $\mathcal{C}$ with approximation sets $\mathcal{C}_\gamma(\tau \circ X')$. Obviously, γ controls the number of such codebook vectors since for large γ we can only select few such sets to cover $\mathcal{C}$; otherwise we risk decoding errors. The concept is analogous to classical coding theory where every codebook vector defines the "center" of an "error-correcting" sphere, and the elements of this sphere are *indistinguishable* from the data transmission point of view.

As derived in Section 3.4.2, an asymptotically vanishing error probability is achievable for coding rates bounded by

$$\begin{aligned} I_\gamma(X', X'') &= \mathbb{E}\widehat{I}_\gamma(X', X'') \\ &= \mathbb{E}\log\left(\frac{|\mathcal{C}|\,|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|}{|\mathcal{C}_\gamma(X')|\,|\mathcal{C}_\gamma(X'')|}\right), \end{aligned} \tag{4.3}$$

The estimator $\widehat{I}_\gamma(X', X'')$ in (4.3) measures the total information content of a message.

For a fixed γ, a large overlap means that the evaluation of the first dataset generalizes to the second dataset, whereas a small or empty intersection indicates lack of generalization. The fraction of approximation set cardinalities in (4.3) measures stability of the solutions under noise fluctuations.

The value $\mathbb{E}\widehat{I}_\gamma(X', X'')$ is an estimate for the *mutual information* and, in analogy to information theory (cf. (Cover and Thomas, 2006, Ch. 7)), the *approximation capacity* is defined as $C \coloneqq \max_\gamma \mathbb{E}[\widehat{I}_\gamma(X', X'')]$ (cf. Definition 3.3). The expectation $\mathbb{E}$ is taken with respect to the random variables X', X''. If the distribution $PG(X^0)$ is unknown then we have to derive learning theoretic large deviation bounds based on the empirical quantity $\widehat{I}_\gamma(X', X'')$ and appropriate complexity penalties.

In summary, maximizing the ratio (4.2) allows us to select the optimal resolution for the solution set, and elements of this approximation set are considered indistinguishable given the present noise process.

4.4 ASC Regularization for Stepwise Algorithms

4.4.1 Application to Stepwise Algorithms: Main Idea

Direct application of ASC regularization requires optimizing (4.2) w.r.t. γ, which, in turn, amounts to compute the cardinalities

$$|\mathcal{C}_\gamma(X') \cap \mathcal{C}_\gamma(X'')|, \quad |\mathcal{C}_\gamma(X')|, \quad |\mathcal{C}_\gamma(X'')| \tag{4.4}$$

(cf. Definition 4.1). In general, computation or at least estimation of these cardinalities requires to enumerate the elements of $\mathcal{C}$ and to test if they belong to $\mathcal{C}_\gamma$. Such enumeration is computationally hard, since $\mathcal{C}$ grows sometimes very fast (exponentially in cases of some combinatorial problems).[3]

For algorithms, these enumeration problems arise in a constraint form, i.e., *the algorithm itself* might help to optimize this enumeration, eliminating all those solutions that get "out of consideration" as the algorithm progresses. In other words, some algorithms feature a step-by-step nature, and they shrink the set of feasible solutions at each next step, ending up with the optimal solution at the last step. Figure 4.1 illustrates this intuition. This contraction is very similar to the shrinkage of approximation sets, as they shrink to the optimal solution as γ decreases. The feasible sets induced by the algorithm often turn out to be efficiently computable – and we will exploit this property of contractive algorithms.

[3]This difficulty emerges as a common bottleneck in many problem settings. We elaborated on it in Section 3.9 of the previous chapter.

4.4.2 Contractive Algorithms and Their Approximation Sets

We define the algorithmic approximation set at computational step t as the set of solutions that are still considered as potential answers to be returned by the algorithm $\mathscr{A}$. This generalized notion of an approximation set for algorithm $\mathscr{A}$ measures the statistical behavior of a specific algorithm during its execution.

Remark. Even if $\mathscr{A}$ calculates the global minimum of the cost function, the algorithm may not follow a gradient flow on that costs to determine the global minimum.

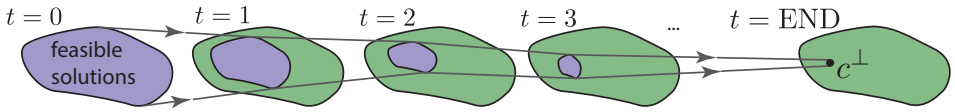


Figure 4.1 Illustration of a contractive algorithmic flow: the set of feasible (i.e. still possible) solutions contracts from step to step, shrinking to $c^{\perp}$ at the end of execution.

Definition 4.2. *Assume that for a given data instance X the stepwise execution of the algorithm $\mathscr{A}$ evaluated on this instance can be expressed as a sequence of subsets of feasible solutions*

$$\begin{aligned} &\mathscr{A}(X) = \langle A_0(X), \ldots, A_T(X)\rangle, \quad \textit{where} \\ &A_t(X) \subseteq \mathcal{C}, \quad t = 0, \ldots, T, \qquad \textit{and} \\ &A_0(X) = \mathcal{C} \\ &A_T(X) = \{c^{\perp}\}. \end{aligned} \tag{4.5}$$

We call $\mathscr{A}$ contractive, if $A_{t+1} \subseteq A_t$.

Definition 4.3. *For a contractive algorithm it is natural to define an algorithmic t-approximation set as follows: $\mathcal{C}_t^{\mathscr{A}}(X) \coloneqq A_t(X)$.*

An example of such algorithm, as well as comprehensive illustration, will be given in Section 4.5 and Figure 4.3.

Speaking informally, the algorithmic t-approximation set is an approximation set as defined in Definition 4.1, but taken w.r.t. the flow of a particular algorithm at update step t. The role of the γ parameter is now (in contrast to Definition 4.1) played by a discrete step variable t spanning from 0 to T (note that, generally, T is not a constant). All the other notions remain the same when adopting the new definition of the approximation sets $A_t(X)$. For example, the following defines the algorithmic ASC score:

Definition 4.4. *As an analogy to* (4.3)*, we will call the quantity*

$$I_t^{\mathscr{A}} = \mathbb{E}\widehat{I}_t^{\mathscr{A}}(X', X'') := \mathbb{E}\log\left(\frac{|\mathcal{C}|\,|\mathcal{C}_t^{\mathscr{A}}(X') \cap \mathcal{C}_t^{\mathscr{A}}(X'')|}{|\mathcal{C}_t^{\mathscr{A}}(X')|\,|\mathcal{C}_t^{\mathscr{A}}(X'')|}\right)$$
$$\equiv \mathbb{E}\log\left(\frac{|\mathcal{C}|\,|A_t(X') \cap A_t(X'')|}{|A_t(X')|\,|A_t(X'')|}\right), \tag{4.6}$$

where $A_t(\cdot)$ are defined above, an algorithmic ASC t-score.

As in Chapter 3, we will distinguish between expected and empirical values.

4.4.3 Algorithmic ASC Score and Optimal Stopping

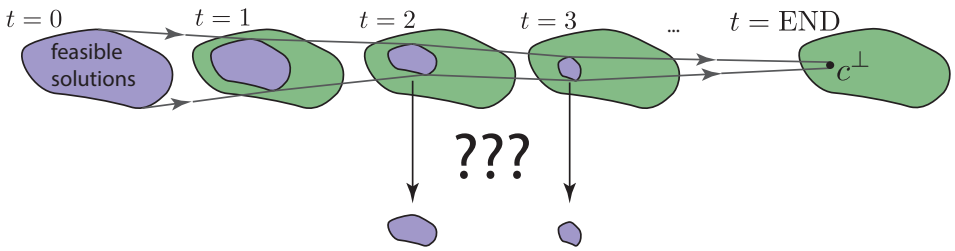


Figure 4.2 The main question addressed in this chapter: which of the steps to choose for the optimal stopping.

As an empirical extension of the notions introduced in Chapter 3, we claim that the algorithmic channel capacity

$$C^{\mathscr{A}} := \max_{t=[0,\dots,T]} I_t^{\mathscr{A}} = \max_{t=[0,\dots,T]} \mathbb{E}[\widehat{I}_t^{\mathscr{A}}(X', X'')], \tag{4.7}$$

also referred as *algorithmic approximation capacity* or *information content*, equals the maximum amount of information which could be transferred through a fictitious channel described in Section 4.3.3 (and, in more detail, earlier in Section 3.4.2), and constrained to the specific algorithm (i.e., algorithmic approximation sets are allowed). From the learning perspective, the algorithmic approximation capacity shows how well the algorithm can filter out the noise (by t-optimization), remaining robust to underfitting.

In a full analogy with the approach of Chapter 3 (cf.(3.25)), one can find the optimal stopping time t^* by maximizing the algorithmic ASC t-score:

$$t^* \in \arg\max_{t=[0,\dots,T]} \widehat{\mathbb{E}} I_t^{\mathscr{A}}(X', X''). \tag{4.8}$$

We are going to use this approach as an optimal stopping criterion for the rest of the chapter (see Figure 4.2).

Remark. We will call this approach *algorithmic ASC* opposed to the original ASC approach developed in Chapter 3.

4.5 ASC Regularization for MST Algorithms

4.5.1 Major MST Algorithms

In the rest of the chapter, we will consider the Minimum Spanning Tree (MST) problem. Its formulation is as follows: given a weighted undirected graph $G = (V, E)$, find a spanning tree of the minimum possible weight. *Spanning tree* is defined as a connected subgraph of G without cycles, whose vertex set equals V.

There are several classical solutions to this problem, of which we will focus on *Prim's*, *Kruskal's* and *reverse-delete* algorithms. We bring their definitions in a pseudo-code in Algorithms 4.2, 4.3 and 4.4 and explain a verbal intuition behind them below.

Algorithm 4.2: Prim's Algorithm for finding MST

Input: undirected graph $G(V, E)$ with non-negative weights
Output: spanning tree with minimum total weight
initialize current tree $B = (V_B, E_B)$ by choosing first vertex:
$B \leftarrow (\{v^*\}, \emptyset)$;
while $V_B \neq V$ *(not all vertices are in tree)* **do**
  find the minimal edge $e_{\min} = (v, v_{\min})$ from B to the rest $G \setminus B$;
  add $e_{\min}$ to the tree B;
return *current tree* B

Prim's algorithm ("growing tree" strategy) starts with a tree $B = (V_B, E_B)$ on an empty set of edges E_P and a starting vertex v^*. The algorithm enlarges E_B, adding one edge e_t at step t (the first step adds an edge incident to v^*), so that B remains to be a tree, until B becomes a spanning tree of G. It takes $T = n - 1$ steps.

Kruskal's algorithm ("connecting trees in a forest" strategy) of finding MSTs starts with an empty set of edges E_B. The algorithm adds a minimal possible e_t at the step t, not allowing cycles in E_B, but yet not requiring B to be connected at all times, until B becomes a spanning tree of G. It takes $T = n - 1$ steps.

Reverse-Delete ("reducing graph" strategy) algorithm starts with a graph on the full set of edges $B = G$ and shrinks it, removing one maximal edge e_t per

Algorithm 4.3: Kruskal's Algorithm for finding MST

Input: undirected graph $G(V, E)$ with non-negative weights
Output: spanning tree with minimum total weight

initialize current tree $B = (V_B, E_B)$: $B \leftarrow (\emptyset, \emptyset)$;
while $V_T \neq V$ *(not all vertices are in tree)* **do**
    find the minimal edge $e_{\min}$ such that:
        a) $e_{\min} \notin E_B$ and
        b) $B \cup e_{\min}$ has no cycles;
    add $e_{\min}$ to the tree B;
return *current tree* B

Algorithm 4.4: Reverse-Delete Algorithm for finding MST

Input: undirected graph $G(V, E)$ with non-negative weights
Output: spanning tree with minimum total weight

initialize current graph $B = (V_B, E_B)$ with the input graph: $B \leftarrow G$;
while B *is connected and yet not a tree* **do**
    find the maximal edge $e_{\max}$ in B such that:
        deleting $e_{\max}$ does not disconnect B;
    if *no such edge found* **then**
        break;
    else
        remove $e_{\max}$ from B;
return *current tree* B

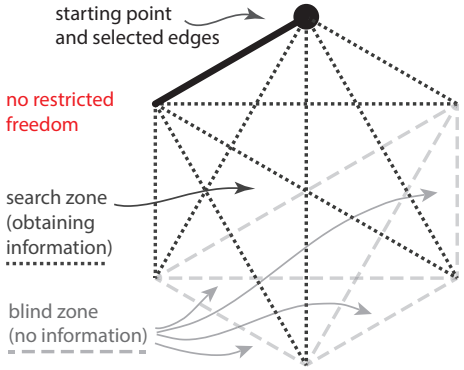
step and keeping the graph B connected, until B becomes a spanning tree. For a complete graph, it takes $T = n(n-1)/2 - n + 1$ steps.

Remark. All the three algorithms can be easily proven to reach the global minimizer solution $c^{\perp}(X) \in \arg\min_c R(c, X)$.

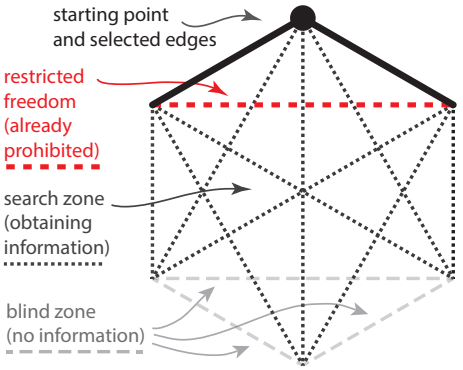
4.5.2 Counting Approximation Sets for MST

MST algorithms are contractive. From the description of three algorithms in the previous section one can see, that they all comply with the Definition 4.2 of a contractive algorithm, due to the following line of reasoning. Since each of them yields a set (yet *not* an approximation set) of *candidate edges* which still are able to make it into the final tree:

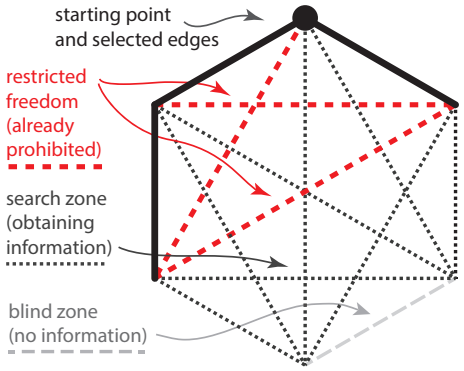
- in case of Prim's and Kruskal's: candidates are edges which are either already



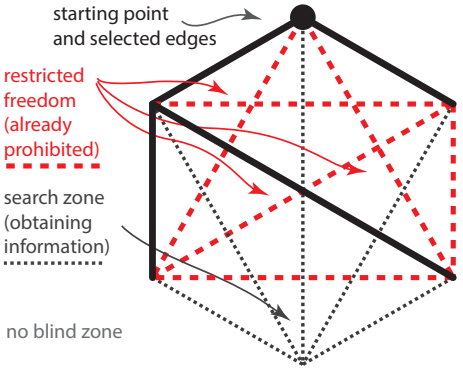
(a) Step 2: candidate edges set is large, no restricted edges yet



(b) Step 3: candidate edges set is large gets smaller, restricted edges appear



(c) Step 4: candidate edges set is large gets even smaller, restricted edges build up



(d) Step 5: almost everything is restricted

Figure 4.3 Prim's MST Algorithm: freedom reduces as information grows. Edge weights are not shown (figure from the introduction recreated here for convenience).

included or still can be included into B;

- in case of Reverse-Delete: candidates are edges which still remain (i.e. are no yet removed) in B.

To complete the reasoning, one should notice that an algorithmic approximation set is exactly the set of spanning trees built on such candidate edges.

This concept is illustrated in Figure 4.3, where several steps (2 to 5) of the Prim's algorithm are shown. One can easily see that as the algorithm flows, some edges are excluded from consideration (due to no-cycle condition; shown in red) and hence less and less spanning trees remain possible.

Computing cardinalities for MST. How can we calculate the cardinalities of algorithmic approximation sets involved in the evaluation of term (4.6)? Counting number of spanning trees for a complete graph can be performed analytically via Cayley's formula (Aigner and Ziegler, 2010):

Theorem (Cayley's Formula for Spanning Trees). *The total number of labeled trees on n vertices is equal to n^{n-2}.*

However, Cayley's formula works only for complete graphs, while in our case, one has to deal with a general case of counting trees on a non-complete subgraph (candidate edges). In this work, we utilize the Matrix-Tree Theorem (cf. Harris et al., 2008). For a connected graph $G = (V, E)$, it involves computing the adjacency matrix M^G_{adj}, the degree matrix

$$M^G_{\text{deg}} = \text{diag}(\deg v_1, \ldots, \deg v_n), \tag{4.9}$$

and uses a notion of a cofactor:

Definition 4.5. *Given an $n \times n$ matrix M, the i, j cofactor of M is defined to be $(-1)^{i+j}\det(M_{i,j})$, where $M_{i,j}$ is the submatrix of M, where i-th row and j-th column are removed.*

Theorem (Kirchhoff's Matrix-Tree Theorem). *The total number of labeled trees on a graph $G = (V, E)$ with adjacency matrix M^G_{adj}, degree matrix M^G_{deg} is equal to (any) cofactor of the matrix $L = M^G_{\text{deg}} - M^G_{\text{adj}}$.*

In our cases, applying the Matrix-Tree theorem to count cardinalities of algorithmic approximation sets $\mathcal{C}^{\mathscr{A}}_t(X')$ and $\mathcal{C}^{\mathscr{A}}_t(X'')$ is straightforward. Computing an intersection of two algorithmic approximation sets $\mathcal{C}^{\mathscr{A}}_t(X') \cap \mathcal{C}^{\mathscr{A}}_t(X'')$ is simple as well.

As the last step, it we find the optimal stopping step $t^* \in \arg\max_t \widehat{I}^{\mathscr{A}}_t(X', X'')$ which maximizes the ratio. The optimal stopping time t^* then defines a pruning operation on the solution space to make $\mathscr{A}$ robust.

4.5.3 Uniform Sampling an Optimally Stopped Spanning Tree

Once the optimal stopping time t^* is defined, we can sample from and intersection of two optimally-stopped algorithmic approximation sets $\mathcal{C}^{\mathscr{A}}_t(X') \cap \mathcal{C}^{\mathscr{A}}_t(X'')$ using the Prüfer encoding of labeled trees. This purely algorithmic task, however, goes beyond the scope of this work and we thus leave it out.

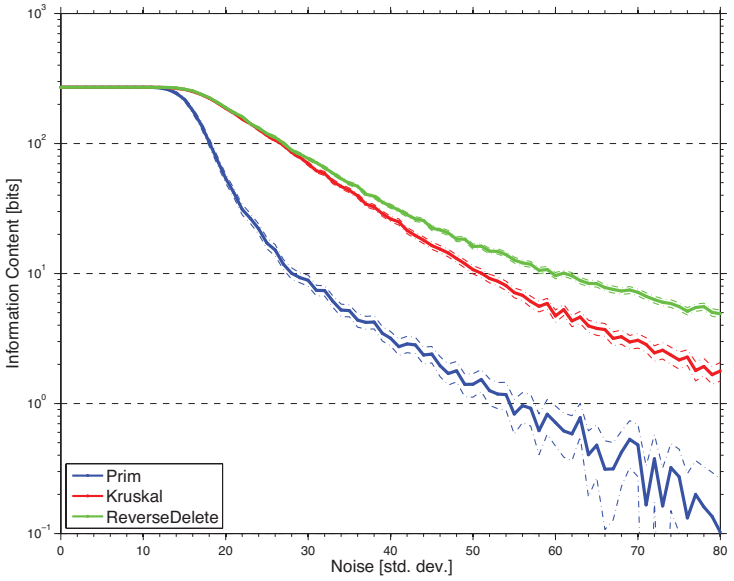


Figure 4.4 Gaussian noise model: information content

4.6 Experimental Results

We will experimentally check that algorithmic approximation capacity is a consistent measure of the information that can be extracted by an algorithm from noisy data by means of optimal stopping.

4.6.1 Experiment Setting: Gaussian Noise Model

To investigate the general information-theoretic behavior of MST algorithms in practice, we generate weighted complete graphs in a hierarchical way. First, we generate a "ground truth" graph with $n = 50$ vertices and with edges that are attributed by Gaussian weights, sampled i.i.d. from a Gaussian distribution $\mathcal{N}(\mu_0 = 100, \sigma_0^2 = 100)$. Second, perturbed versions of this ground truth graph are then obtained by adding Gaussian noise $\mathcal{N}(\mu = 0, \sigma^2)$ to the edge weights for a given noise range $\sigma \in [0, 8\sigma_0]$.

In the experiment with approximation set-regularized algorithms we repeated the experiment 400 times to ensure the statistical significance of the results. For some plots a semi-logarithmic scale was used for a better visualization of small differences. Confidence intervals were also constructed and plotted.

4.6.2 Algorithmic Approximation Capacity Ranking of Algorithms

We plot the algorithmic information content $\max_t \mathbb{E}\widehat{I}_t^{\mathscr{A}}(X', X'')$ (cf. (4.6)) for three algorithms (Figure 4.4). At low noise levels (particularly $\sigma = 0$) all the three algorithms exhibit the same information content, which is equal to $\log_2 n^{n-2} = 48 \log_2 50 \approx 270,9$ bits of information: at zero noise, all the three algorithms choose the true MST out of $|\mathcal{C}|$ possible spanning trees. For a complete graph, Cayley's tree formula calculates the number of possible solutions as $|\mathcal{C}| = n^{n-2}$.

The plot in Figure 4.4 provides a clear ranking of the three algorithms w.r.t. their information content dependent on the noise level. A qualitative explanation which accounts for such ranking and clarifies the idea behind evaluating information content, is the following:

Prim's algorithm considers for addition only the edges which are already connected to the tree built so far. Among the not yet considered edges there might be low cost edges which are more efficient to be added in the beginning of the run rather than in the end, thus making the information extraction inefficient from the point of view of the algorithm dynamics.

Kruskal's algorithm explores all the possible edges as candidates for addition at each step, thus being less inclined to add inefficient edges first and using the algorithm dynamics in a efficient way. This gain is reflected by its increased informativeness relative to Prim's algorithm.

Reverse-Delete algorithm is more informative than Kruskal's, since it efficiently discards all those edges which should not be included in any approximate spanning tree. It pursues a strategy of delayed decision making, that proved to be favourable also in other situations of decision making under uncertainty.

The above insights are proven by the plot showing the stepwise dynamics of logarithm of the cardinalities $|\mathcal{C}_t^{\mathscr{A}}(X')|$, $|\mathcal{C}_t^{\mathscr{A}}(X'')|$ (Figure 4.6). It visualizes the fact, that as t progresses, Prim's algorithm contracts for the solution faster than Kruskal's, and both contract faster that the Reverse-Delete one, which, in turn, forces earlier stopping and thus leads to the worse performance. In fact, we can formulate an informal statement:

Statement 4.1. *Assume that at the step t the algorithm exhibits the edge set B defined above in Section 4.5. Then the reduction of the amount of feasible spanning trees with edges in B obtained by adding e_{t+1} which is adjacent to B (Prim) is higher than the reduction obtained by adding non-adjacent e_{t+1} (Kruskal in general), and both are higher than the reduction obtained by deleting e_{t+1} from B (Reverse-Delete).*

The plot in the Figure 4.7 explains the discussed ranking from the stepwise

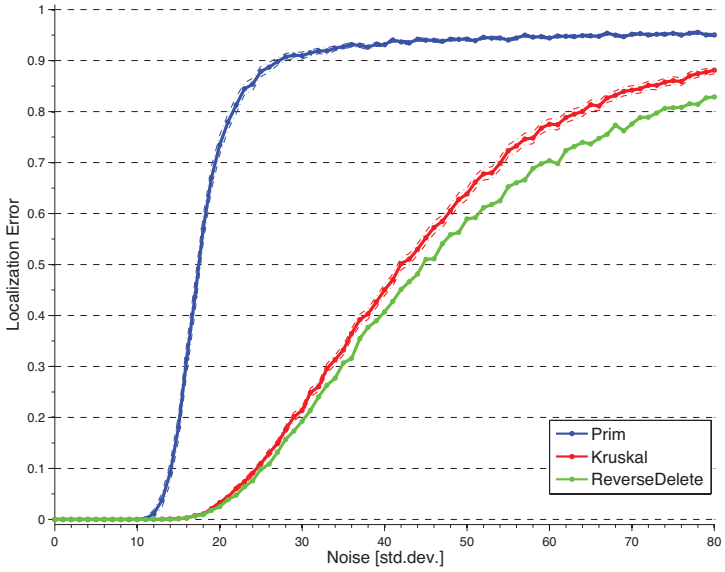


Figure 4.5 Gaussian noise model: localization error

dynamics prospective. The algorithm, which reaches maximum of mutual information earlier, is less informative in overall, and vice versa. This behavior reflects a natural trade-off between early decision and informativeness of the solution.

4.6.3 Localization Error Ranking of Algorithms

The three given algorithms "explore" the solution set with different dynamic behavior, extracting different amount of the information about the true solution. This behavior is related to the localization error.

For the algorithmically ASC-regularized (Section 4.3.2) solution $\widehat{c}$ we plot (Figure 4.5) the localization error, which is computed as $E(\widehat{c}) = 1 - |c^* \cap \widehat{c}|/|c^*|$, where straight brackets denote the cardinality of edges.

It can be seen from the figures, that the ranking of the three algorithms according to their localization capability is in connection with the information content ranking — the more informative algorithm yields less error in localizing the true solution.

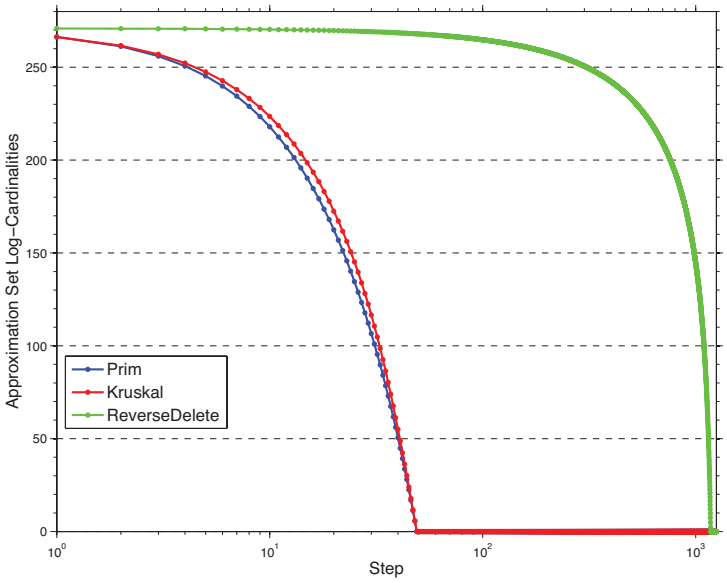


Figure 4.6 Gaussian noise model: stepwise approximation set log-cardinalities ($\sigma = 48$)

4.6.4 Algorithmic ASC vs. Original ASC

As the last experiment, we showed that the original ASC regularization (the one via γ-parameter) works still better than algorithmic ASC and even beats the Joint Minimizer (first introduced as a benchmark solution in Chapter 3), which minimizes the average cost $R(c, X') + R(c, X'')$.

Due to the computation limitations of the original ASC — it yields enumerating the whole set of spanning trees (n^{n-2}) for each step t — we could only run the experiments for graphs with few vertices ($n = 6$). Figure 4.8 shows that the original ASC remains competitive with the Joint Minimizer solution, while algorithmic version shows weak performance. This is not surprising and comes at a certain cost for which we discuss in conclusion.

4.7 Discussion and Conclusion

On the ASC-induced ranking of the algorithms

The framework of approximation set coding is generalized from the domain of models to the domain of algorithms in the chapter. This framework enables us to

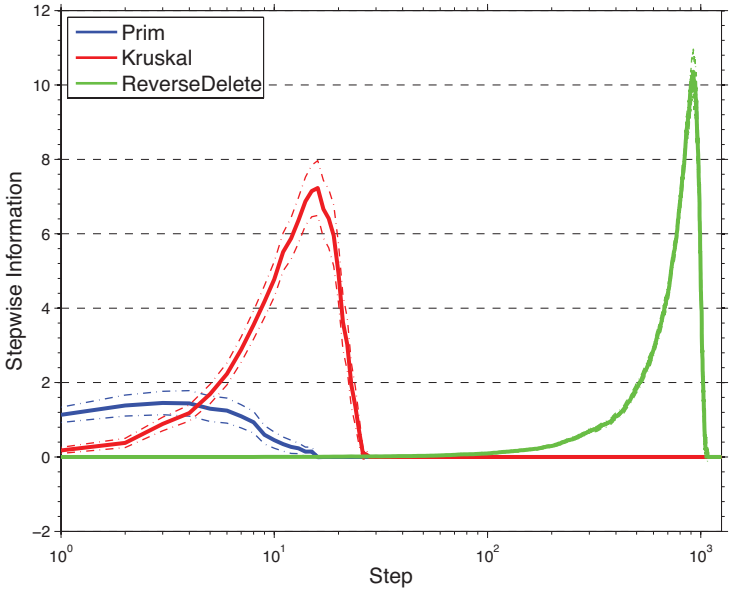


Figure 4.7 Gaussian noise model: stepwise algorithmic information defined in (4.6) ($\sigma = 48$)

apply an information-theoretic regularization and quality assessment principle to algorithms, and, in particular, to a minimum spanning tree problem with noisy graphs as input.

We interpreted the results of the quantitative analysis of the algorithms in a qualitative way, binding the strategy of the algorithm to its information content, showing consistency and agreement with the localization capability of the algorithm.

The ranking of the quantities plotted in Figure 4.4–4.7 support the following conjecture.

Conjecture 4.1. *The information content of contractive algorithms applied to the same problem establishes a ranking among them. This ranking is consistent with the average localization error of these algorithms.*

A possible way of investigating this relation is a rigorous analysis of the approximation set dynamics using the mentioned Matrix-Tree theorem. Although we utilized the simplest model of Gaussian noise, independent on separate edges, there exists evidence, which allows us to expect the same consistent results for a *structured* noise setting, when the edges of the graph are impacted by the noise in

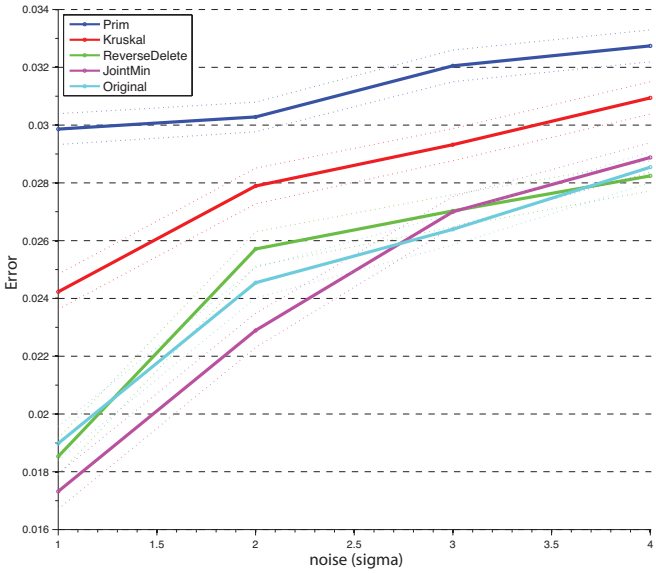


Figure 4.8 Error plotted on the solutions obtained via original ASC (Chapter 3 and algorithmic ASC (this chapter), as well as benchmarked on Joint Minimizer (Section 4.6.4).

a complex way, involving the statistical dependence of the noise ingredients and other degrees of the noise model complexity.

Algorithmic ASC vs. original ASC

Should one use algorithmic approximation approach of this chapter or the original γ-approximation approach form Chapter 3? In this section, we address a question on whether the optimal stopping rule derived by algorithmic ASC can compete with the original ASC (we initiated this discussion in Section 4.6.4).

In fact, these two approaches serve quite different purposes and feature different highlights. The original γ-parametrized ASC regularization works in conjunction with the optimization goal $R(c, X)$, while its algorithmic version makes use of algorithm-specific *flow*. Although the algorithm finds the same global optimizer as a bare $R(c, X)$-minimization procedure, the structure of approximation sets if very different. Below, we list the main points of difference:

- The original ASC relates to the continuous parameter, while algorithmic ASC works with a discrete steps. This yields different power of resolution at which we coarse-grain the solution set $\mathcal{C}$. For original ASC, this resolution

power is higher (it can allow much smaller step between approximation set sizes), while for algorithmic ASC it fully depends on the algorithm.

- Original ASC is hard to apply computationally, since it boils down to computing and enumerating approximation sets. For algorithmic ASC, it is easier to derive a problem-specific procedure which makes it *orders of magnitude* faster. For example, in the MST case, we utilized the Matrix-Tree theorem and computed the cardinalities analytically at each step!

Hence, the trade-off between the power of resolution and the computational power exists and should be addressed in each special case.

General notes on the ASC approach to algorithms

From the statistical physics perspective, a contractive algorithm can be considered as a process, where the temperature decreases step by step, and thus "freezes" the solution space, until finding the optimal solution. The survey (Merhav, 2010, Sec. 3.2) discusses the connection between such a coding framework and statistical physics. This connection (Buhmann, 2010b) inspires us to explore the dynamics of the algorithm from a generalization point of view and to relate it to the information theory (Merhav, 2010).

We will consider the statistical physics viewpoint on the ASC regularization in the next chapter.

5

Combinatorial Optimization: Regularization by Free Energy

"Tout cela, Maxwell et Boltzmann l'ont expliqué, mais celui qui l'a vu plus nettement, dans un livre trop peu lu parce qu'il est difficile à lire, c'est Gibbs dans ses principes de la Mécanique Statistique."

(fr. "All this, Maxwell and Boltzmann have explained, but the one who saw it in the cleanest way, in a book that is too little read because it is difficult to read, is Gibbs, in his 'Principles of Statistical Mechanics'.")

— Henri POINCARÉ, "La valeur de la science"

5.1 Introduction

Most real world combinatorial optimization problems are affected by noise in the input data, thus behaving like large disordered particle systems, e.g. spin glasses. And similar to large physical systems, they optimize a certain functional (energy, cost function).

In this chapter, we will work with two notions arising from this analogy: *Gibbs distribution* and *free energy*. We address two interrelated questions connected to these notions: first, we provide rigorous asymptotic computation of the matching upper and lower bounds on the free energy (for two disordered combinatorial optimization problems, the sparse Minimum Bisection Problem, sMBP, and Lawler's Quadratic Assignment Problem, LQAP). We show that the free energy exhibits phase transitions equivalent to that of Derrida's Random Energy Model (REM). Then, the obtained free energy asymptotics leads to the second contribution of this chapter: theoretic justification of the Gibbs relaxation of ASC introduced earlier

in Chapters 3 and 4. We perform it by introducing the *Gibbs posterior agreement*, which is, in simple terms, a measure of stability of the Gibbs distributions in case when the cost function fluctuates. Additionally, we carry out experiments which support phase transition findings and conjecture further extension of the theorems proven in the chapter.

5.1.1 Notation

Remark. Before reading this section, we advise the reader to briefly revisit the notation of Section 2.3.1, Definition 3.1 for notational analogies.

We consider combinatorial optimization problems that can be formulated as follows (for explanation see Fig. 5.1): let n be some integer determining the size of the problem (e.g., number of vertices in a graph, size of a matrix, etc.), and $\mathcal{S}_n$ a finite set of objects (e.g., set of edges, elements of a matrix, etc). Working in the spirit of Section 2.3.1 and under the rigorous notation of Definition 3.1, we then let X denote some random input to the problem (data).

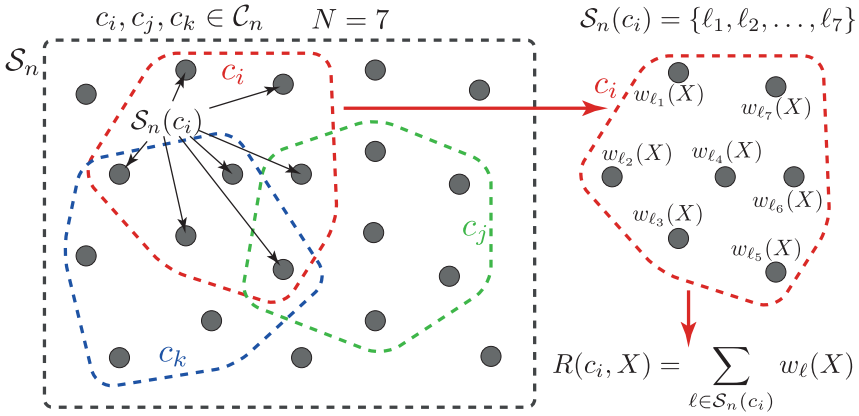


Figure 5.1 Illustration of the notation: each of solutions (examples shown in the figure are c_i, c_j, c_k) includes N (in the figure $N = 7$) objects from the underlying set $\mathcal{S}_n$. The cost function of a solution is the sum of weights assigned to the objects, which belong to that solution.

Define $\mathcal{C}_n$ as a finite set of all feasible solutions (e.g. bisections of a graph), and $\mathcal{S}_n(c) \subseteq \mathcal{S}_n$, $c \in \mathcal{C}_n$, as a finite set of objects belonging to the feasible solution c (e.g., set of edges belonging to a bisection). Let $w_i(X) = W_i$, $i \in \mathcal{S}_n$, be the weight assigned to the i-th object. In this chapter we consider optimization problems for

which the cost function and optimization task are defined as follows:

$$R(c, X) = \sum_{i \in \mathcal{S}_n(c)} w_i(X) \quad \text{and} \quad c^{\perp}(X) \in \arg\min_{c \in \mathcal{C}_n} R(c, X). \tag{5.1}$$

We also denote the cardinality of the feasible set as m (i.e., $m \coloneqq |\mathcal{C}_n|$) and the cardinality of $\mathcal{S}_n(c)$ as N for all $c \in \mathcal{C}_n$ (i.e., solution size $N \coloneqq |\mathcal{S}_n(c)|$). In this chapter, we focus on optimization problems in which $\log m = o(N)$ holds true (see Szpankowski, 1995).We call these optimization problems *parameter-rich* since the log cardinality scales sub-linearly with the number N of objects that belong to a solution c.

5.1.2 Motivation: ASC Gibbs Relaxation via Free Energy

Gibbs relaxation of ASC. As mentioned in the introduction, most real world combinatorial optimization problems are affected by noise in the input data X. Therefore, they behave like large disordered particle systems, e.g., random networks or spin glasses. Like physical systems, they optimize an application-dependent functional (cost function), and their solutions are characterized by some *posterior distribution* $p(c|X)$ given the data X. In view of this stochastic setting, one can "robustify" the solution by the maximum entropy method. In the framework of maximum entropy, it is well justified (see Vannimenus and Mézard (1984)) to use Gibbs distributions, known also as *Gibbs posteriors*, for the posterior distribution $p(c|X)$ leading to a novel pipeline presented in Fig. 5.2 (lower) as opposed to the standard one (upper).

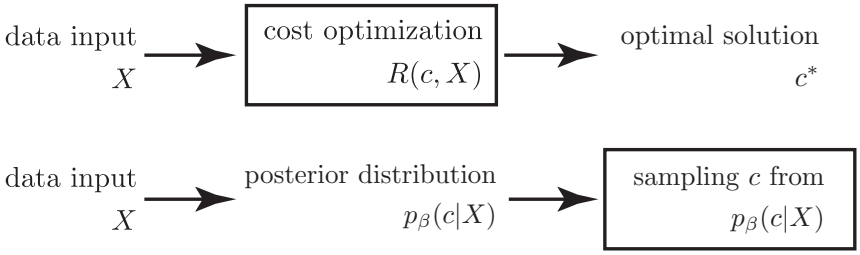


Figure 5.2 Standard risk minimization (upper) solution and a solution obtained via sampling from approximating posterior distribution (lower).

Remark. The following is the elaboration of the Gibbs ASC relaxation first introduced very briefly in Chapter 3. Although we give all the necessary definitions below, we still advise the reader to revisit the respective Section 3.8.1.

Definition 5.1. *Suppose we are given an optimization problem defined by a cost function $R(c, X) \in \mathbb{R}$, where c is a solution from the finite solution space $\mathcal{C}$ and X is a random data instance. Then the* ***Gibbs posterior distribution*** *$p_\beta(c|X)$ defined as*

$$p_\beta(c|X) = \frac{1}{Z(\beta, X)} \exp(-\beta R(c, X)) \quad \textit{with}$$
$$Z(\beta, X) = \sum_{c' \in \mathcal{C}} \exp(-\beta R(c', X)) \,. \tag{5.2}$$

The term $Z(\beta, X)$ is known as the *partition function*. The Gibbs distribution is parameterized by a parameter β which is called the *inverse temperature*. For any β the Gibbs posterior assigns the highest weights to those solutions that have the smallest costs, and β controls the level of concentration of $p_\beta(c|X)$ around minimal solutions (see Fig. 5.3(a)).

In passing we remark that we will sometimes omit X as an argument of $Z(\beta, X)$ and $R(c, X)$ for the sake of brevity. Expectation $\mathbb{E}[.]$, variance $\mathrm{Var}[.]$ and other probabilistic operations are meant to be evaluated with respect to the distribution of X, if not explicitly stated otherwise.

Obviously, β somewhat contributes to robustness of $p_\beta(c|X)$. But then the question arises: what is the right way to measure how good a particular choice of β is? To answer that, we investigate what happens to $p_\beta(c|X)$ when the input data fluctuates. Let us assume that *two* noisy instances X' and X'', come from the same source. Intuitively (see Fig. 5.3(a)), for values of β that are *very small*, the posteriors $p_\beta(c|X')$ and $p_\beta(c|X'')$ are very similar (we will informally say "stable"), but they do not carry much information due to their large variance (we will informally say "non-informative"). Conversely, using values of β that are *very high* result in very informative posteriors but they are simultaneously very sensible to noise (observe that the best solution to X'' is highly improbable under $p_\beta(\cdot|X')$). One of the ways to balance between these two limits of under- and overfitting is to introduce the posterior agreement kernel for two data instances that show how "close" $p(c|X')$ and $p(c|X'')$ are.

A natural measure of agreement between $p_\beta(c|X')$ and $p_\beta(c|X'')$ is defined by the overlap between the two posteriors in the solution space. In Section 3.8.1, we introduced (without naming however) a quantity which we call here *log-posterior agreement kernel* for two instances, and define it below.

Definition 5.2. *The* ***posterior agreement kernel*** *for two instances X', X'' is*

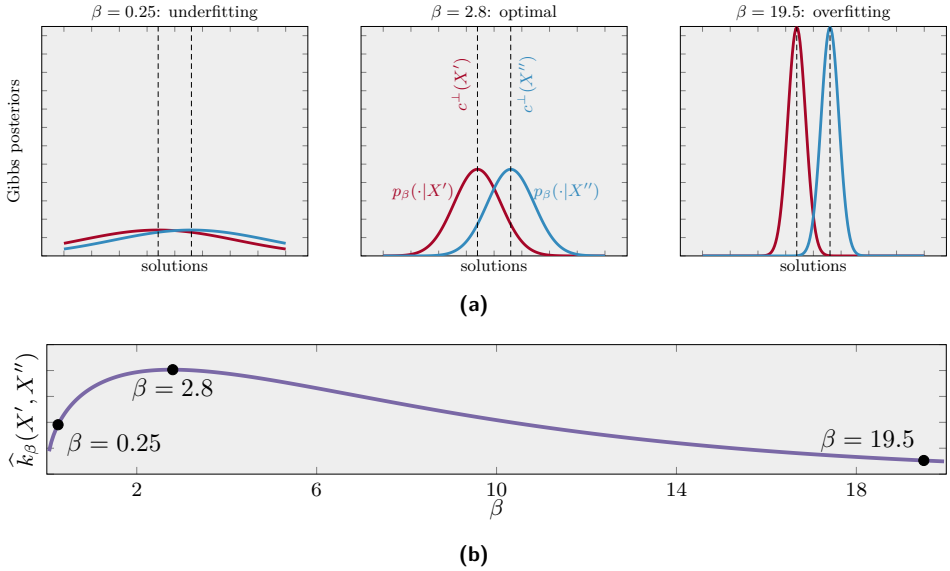


Figure 5.3 **(a)**: Schematic depiction of two Gibbs posteriors $p_\beta(\cdot|X')$ and $p_\beta(\cdot|X'')$, which may underfit (low β), be optimal (intermediate β) or overfit (high β) depending on the regularizing inverse temperature β. **(b)**: the value of the empirical agreement kernel $\widehat{k}_\beta(X', X'')$ as a function of β, computed for the toy example of the Figure (a). The value $\beta = 2.8$ maximizes this kernel, meaning that the two posteriors are possibly "stable" *and* "informative".

defined as

$$\begin{aligned}\widehat{k}_\beta(X', X'') &= \sum_{c\in\mathcal{C}} p_\beta(c|X')p_\beta(c|X'') \\ &= \frac{\sum_{c\in\mathcal{C}} \exp(-\beta(R(c, X') + R(c, X'')))}{Z(\beta, X')Z(\beta, X'')} \\ &\equiv \frac{Z(\beta, X', X'')}{Z(\beta, X')Z(\beta, X'')},\end{aligned} \tag{5.3}$$

where $Z(\beta, X', X'')$ is defined as the expression in nominator.

Definition 5.3. *Further, analogically to ASC (cf. (3.31)) we give a definition of empirical log-posterior agreement:*

$$\begin{aligned}\widehat{I}_\beta(X', X'') &:= \log \widehat{k}_\beta(X', X'') \\ &= \log \sum_{c\in\mathcal{C}} p_\beta(c|X')p_\beta(c|X'')\end{aligned} \tag{5.4}$$

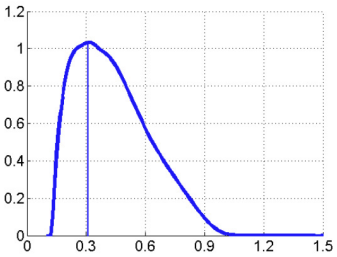


Figure 5.4 Experimental results for the averaged log-posterior agreement of a clustering problem (Chehreghani et al., 2012).

and the expected log-posterior agreement (*cf.* (3.28)) *as*

$$\begin{aligned} I_\beta \equiv \mathrm{eLPA}_\beta &\coloneqq \mathbb{E}\widehat{I}_\beta(X', X'') \\ &= \mathbb{E}\log \widehat{k}_\beta(X', X'') \\ &= \mathbb{E}\log \sum_{c\in\mathcal{C}} p_\beta(c|X')p_\beta(c|X''). \end{aligned} \tag{5.5}$$

Finally, in full analogy with the original ASC (Chapter 3) we introduce *approximation capacity* (see Definition 3.3) which quantitatively measures the maximum of expected log-posterior agreement (cf. (3.66)):

Definition 5.4. *The* ***approximation capacity*** I *of a cost function* $R(c, X)$ *is defined as*

$$\begin{aligned} I &\coloneqq \sup_\beta \mathbb{E}_{X',X''} \log |\mathcal{C}| \sum_{c\in\mathcal{C}} p_\beta(c|X')p_\beta(c|X'') \\ &\coloneqq \sup_\beta \mathbb{E}_{X',X''} \log\big(|\mathcal{C}|\ \widehat{k}_\beta(X', X'')\big)\,. \end{aligned} \tag{5.6}$$

The optimal β^* is thus, according to Chapter 3, obtained through maximizing the eLPA:

$$\beta^* \coloneqq \arg\max_\beta \mathbb{E}_{X',X''} \log\big(|\mathcal{C}|\ \widehat{k}_\beta(X', X'')\big). \tag{5.7}$$

The search for an optimal β can also be interpreted as a selection of a randomized algorithm that samples solution from a Gibbs posterior with the respective β-controlled "width". In a toy example presented in Figure 5.3(b) the posterior kernel is shown for different values of β. We see that it has a clear maximum w.r.t. the temperature, and this is always the case. In fact, this is also confirmed by re-

cent experimental results from (Chehreghani et al., 2012) shown in Figure 5.4. In this chapter, in Theorem 5.4 we provide theoretical justification for such behavior by considering in details two optimization problems, namely sparse Minimum Bisection Problem (sMBP) and Lawler's Quadratic Assignment Problem (QAP).

Computing Free Energy Density. In order to estimate the posterior kernel (5.3) we need to evaluate $\mathbb{E}\log Z(\beta, X')$, $\mathbb{E}\log Z(\beta, X'')$ as well as $\mathbb{E}\log\sum_{c\in\mathcal{C}}\exp\big(-\beta(R(c,X')+R(c,X''))\big)$, i.e. the expected log-partition functions. For a large n this task represents a computational bottleneck and is known to pose a notoriously difficult mathematical challenge (see Talagrand (2003)). We address this issue in our work and provide new solutions and novel lower bounding techniques. More precisely, we compute the *Helmholtz free energy* density defined in (5.8).

Definition 5.5. *The free energy density (or rate) of a set of solutions (configurations) $\mathcal{C}$ is defined as*

$$\mathcal{F}(\beta) = -\mathbb{E}_X[\log Z(\beta, X)]/\log|\mathcal{C}| \,. \tag{5.8}$$

Remark. Note the differences with the original definition (Definition 2.11): first, we drop the $1/\beta$ scaling for convenience here; second, we use a scaling by $\log|\mathcal{C}|$, as we refer to free energy *density*. It is important that both scalings are used for technical convenience and to ensure the existance of thermodynamic limits ($n \to \infty$), therefore, we will mostly utilize the term "free energy" without the word "density".

It is known (Bovier et al., 2002; Talagrand, 2003), that obtaining asymptotic bounds for this quantity is a difficult mathematical problem. In this chapter we tackle it for some special cases.

5.1.3 Contributions and Outline of the Chapter

As contributions of this chapter, we

- perform a mathematically rigorous asymptotic analysis of the free energy for two optimization problems (i.e., sparse MBP and Lawler QAP described below in detail) for a high-temperature regime, proving matching upper and lower bounds. For both, we introduce novel methods of proving them. We shall find phase transitions which are equivalent to the discontinuities of REM and high-temperature SK (Derrida, 1981; Aizenman et al., 1987);

- perform a rigorous asymptotic computation of *expected log-posterior agreement (eLPA)* — a Gibbs relaxation of the ASC score. Original and Gibbs ASC scores were introduced in earlier chapters, cf. (3.25), (3.28) and (3.67);
- we interpret the semantics of eLPA in a new way, supporting the ideas from Chapter 3 and (indirectly) Chapter 4;
- we carry out experiments which give a firm ground for a conjecture about a form of the free energy in a general (i.e. not constrained to sMBP or Lawler QAP) problem cases. Surprisingly, this conjecture turns into proven asymptotics for cases of sMBP and Lawler QAP.

The chapter is organized as follows. First, an overview on the related work is given in Section 5.2. Formal definitions and main result theorems are stated in Section 5.3: in Section 5.3.1 the sMBP and Lawler QAP optimization problems are described, while in Section 5.3.2 (namely, Theorems 5.2, 5.3 and 5.4) main results about them are provided. Proofs of the main results can be found in Sections 5.3.4 and 5.3.5. In Section 5.4, an important intermediary discussion is made. Further, in Section 5.5 we perform an experimental evaluation of a more general case of non-sparse MBP, and conjecture a surprisingly good ad-hoc formula for a free energy. In Section 5.6, we discuss our results.

5.2 Background and Related Work Overview

Combinatorial optimization arises in many real world settings and these problems are often notoriously difficult to solve due to data dependent noise in the parameters defining such instances. Algorithms that minimize these noisy instances or approximate their global minimum return a solution that is a random variable due to input randomness and that is most often highly unstable. Therefore, we ask the very reasonable questions: What is the distribution of the output returned by the algorithm? Can we stabilize such an output distribution by regularizing the algorithm?

Algorithm design in noise affected real world settings requires both statistical as well as computational considerations: first, we have to ensure that outputs of algorithms are typical in a statistical sense, i.e., they have to occur with high probability. Second, such typical outputs have to be computable in an efficient way with efficient resources.

Due to the statistical nature of inference, we have to efficiently compute posterior distributions of solutions given input data. Open theoretical issues emerge for this strategy, e.g., analytical computation of macroscopic properties like entropy,

expected log-partition function or expected costs (Frenk et al., 1985; Talagrand, 2003). The expected log-partition function known also as the *free energy*, appeared in the context of combinatorial optimization since the mid 80's; see e.g., the work by Vannimenus and Mézard (1984), who explored the free energy properties of the traveling salesman problem. An intriguing property of free energy is the emergence of discontinuities of certain order when changing the concentration of the posterior distribution. Such abrupt changes of macroscopic properties, also known as *phase transitions*, are characteristic features of various large systems and have been generating an uninterrupted interest in theory of discrete structures for a long time (see Cohen, 1988; Łuczak, 1994).

Free energy found also applications in theoretical computer science as discussed in the previous section. We introduced a robustness score function called the *expected log-posterior agreement* (eLPA) for measuring "goodness" of robust solutions. It is tightly connected to computing free energies, as wee noted above. Furthermore, estimating the free energy for combinatorial optimization problems allows us to justify theoretically some experimental results obtained for these problems.

For the sake of completeness we should note here that the same, if not more intensive, excitement has been generated for finding *theoretical* laws that govern the behavior of macroscopic thermodynamic properties in statistical physics of large disordered particle systems. Many interesting models of such large systems were introduced relatively early, e.g. the *Sherrington-Kirkpatrick (SK) spin glass model* (see Sherrington and Kirkpatrick, 1975). It took, however, some time and effort to develop rigorous techniques for solving them. For example, Derrida (1981) introduced a very simple, but exactly solvable model called *random energy model (REM)* as the limit of SK models family. Later, Aizenman et al. (1987) published an exact solution in the high-temperature phase for SK model. The general question about the exact free energy behavior became increasingly motivating: it triggered a new wave of latest research (see Bovier et al., 2002; Talagrand, 2003). The reader should also note that many interesting heuristic tools were developed in the context of statistical physics over the last several decades, such as the replica method (Parisi, 2009), the cavity method (Mézard and Parisi, 2003) and mean-field approximation schemes with belief propagation algorithms.

5.3 Main Results

In this chapter we focus on two optimization problems, namely the sparse Minimum Bisection Problem (sMBP) and the Lawler Quadratic Assignment Prob-

lem (LQAP). Formal definitions are given below. However, we should add that many of our results hold for a larger class of optimization problems as long as $\log m = o(N)$ (see Szpankowski, 1995). In the rest of the chapter we will utilize the temperature rescaling $\beta = \widehat{\beta}\sqrt{\log m/N}$ with $\widehat{\beta} = \mathcal{O}(1)$ which together with $\log m = o(N)$ explains $\beta \to 0$ limit. This rescaling was justified in (Buhmann et al., 2014).

For these two problems we shall provide tight asymptotics for the free energy (5.8), and compute asymptotically the log-posterior agreement as well as $\widehat{\beta}^*$ that maximizes the posterior kernel.

5.3.1 Minimum Bisection and Quadratic Assignment Optimization Problems

This section introduces the combinatorial optimization problems that will be used to describe our findings. These problems fall into the $\log m = o(N)$ class specified in Sec. 5.1.1 and cover a wide range of practical applications in signal processing and neural information processing.

Minimum bisection problem (MBP). Consider a complete undirected weighted graph $G = (V, E, X)$ of n vertices, where n is an even number. The input data instance X is represented by (random) weights $(W_i)_{i\in E}$ of the graph edges. A *bisection* is a balanced partition $c = (U_1, U_2)$ of the vertices in two disjoint sets: $U_1, U_2 \subset V$, $U_1 \sqcup U_2 = V$, $|U_1| = |U_2| = \frac{n}{2}$. Now $\mathcal{S}_n = E$ and $\mathcal{C}_n$ is the set of all bisections of graph G, while $\mathcal{S}_n(c)$ is the set of all edges cut by the bisection c. The cost of a bisection c is the sum of the weights of all cut edges

$$R(c, X) = \sum_{i\in\mathcal{S}_n(c)} W_i. \tag{5.9}$$

The minimum bisection problem finds the bisection of a graph with minimum cost. A simple calculation (we omit here $1/2$ constant for the sake of brevity) shows that $|\mathcal{C}_n| = m = \binom{n}{n/2}$ and $|\mathcal{S}_n(c)| = N = \frac{n^2}{4}$, and that

$$\log m = \log \binom{n}{n/2} \sim \log\left(2^n\sqrt{\frac{2}{\pi n}}\right) = n\log 2 - \frac{1}{2}\log n + \mathcal{O}(1), \tag{5.10}$$

which shows that the minimum bisection problem belongs to the class of stochastic optimization problems discussed in this work (i.e., $\log m = o(N)$).

Sparse minimum bisection problem (Sparse MBP, sMBP). We actually will focus on the *sparse* Minimum Bisection Problem in which the disjoint subsets are of the size $|U_1| = |U_2| \equiv d$ where d grows faster than $\log n$ and slower than n

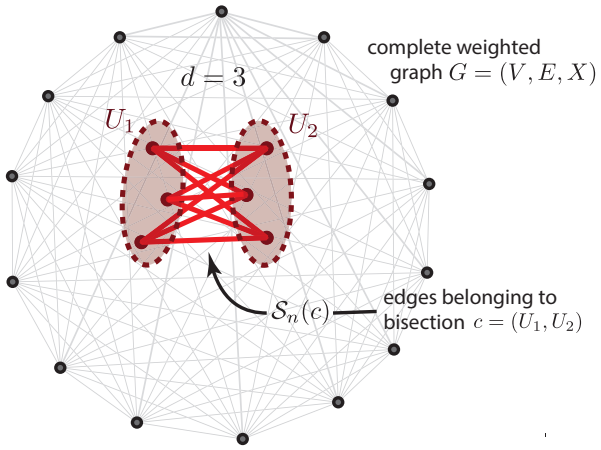


Figure 5.5 Illustration of the sparse Minimum Bisection Problem (sMBP) introduced in Section 5.3.1.

(which we write $\log n \ll d \ll n$). Thus, $N = d^2$ and the following holds

$$\log m = \log \binom{n}{d}\binom{n-d}{d} = \log \frac{n!}{d!(n-2d)!} \sim 2d \log n. \tag{5.11}$$

Thus the problem falls into the class $\log m = o(N)$ since we assume $\log n \ll d$.

Quadratic Assignment Problem (QAP). We consider two $n \times n$ real-, positive-valued matrices, namely the weight matrix V and the distance matrix H. The solution space $\mathcal{C}_n$ is the set of the n-element permutations $\mathbf{S}_n$. The cost function is then $R(\pi, V, H) = \sum_{i,j=1}^{n} V_{ij} \cdot H_{\pi(i),\pi(j)}$ for $\pi \in \mathbf{S}_n$. In our terms, the object space is the set of products of entries of V and H constrained by a relation on the indices: $\mathcal{S}_n = \{V_{ij} \cdot H_{\pi(i),\pi(j)} \mid 1 \leq i, j \leq n; \pi \in \mathbf{S}_n\}$. In our notation, $N = |\mathcal{S}_n(\pi)| = n^2$ and $m = |\mathcal{C}_n| = n!$ and thus $\log m \sim n \log n = o(N)$ is satisfied.

Lawler Quadratic Assignment Problem (Lawler QAP). Lawler (1963) introduced a generalization of the QAP where the distance and weight matrices are replaced by a 4-dimensional matrix Q with i.i.d. values: $R(\pi, Q) = \sum_{i,j=1}^{n} Q_{i,j,\pi(i),\pi(j)}$ for $\pi \in \mathbf{S}_n$. It is interesting to see that this generalization does not change the combinatorial structure of the problem: a Lawler QAP can be built from a normal QAP and thus falls into our class.

5.3.2 Free Energy and its Phase Transition

In order to give a full picture, before presenting our main results we first derive a tight upper bound on the free energy as discussed in (Buhmann et al., 2014). Interestingly, it shows that there is a phase transition in the second-order term of the upper bound of the free energy. Such a phase transition is a characteristic feature of various large-scale systems (see Łuczak, 1994; Talagrand, 2003; Mézard and Montanari, 2009).

First, let us state our main assumptions that we use throughout the chapter.

Definition 5.6 (Common Theorem Setting). *Consider a class of combinatorial optimization problems in which:*

(A) *the cardinality m of the set of feasible solutions and the size N of every feasible solution are related as $\log m = o(N)$, and we adopt the scaling $\beta = \widehat{\beta}\sqrt{\log m/N}$;*

(B) *weights W_i are identically (not necessarily independently) distributed with mean μ and variance σ^2 and that the moment-generating function of the negative centralized weights $(-\overline{W}_i)$ is finite, i.e. $\overline{G}(t) \equiv \mathbb{E}[\exp(-t\overline{W}_i)] < \infty$ exists for some $t > 0$;*

(C) *within a given solution c, the weights are mutually independent, i.e. for all $c \in \mathcal{C}_n$, the set $\{W_i \mid i \in \mathcal{S}_n(c)\}$ is a set of mutually independent variables.*

Theorem 5.1. *Under the common theorem setting of the current section the following holds:*

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z(\beta, X)] + \widehat{\beta}\mu\sqrt{N\log m}}{\log m} \leq \begin{cases} 1 + \frac{\widehat{\beta}^2\sigma^2}{2}, & \widehat{\beta} < \frac{\sqrt{2}}{\sigma}, \\ \widehat{\beta}\sigma\sqrt{2}, & \widehat{\beta} \geq \frac{\sqrt{2}}{\sigma}. \end{cases} \tag{5.12}$$

Remark. As it can be seen from the above theorem, a unit-free rescaling $\tilde{\beta} = \widehat{\beta}\sigma$ could simplify the formulation. Further, we use the initial rescaling for clarity.

Remark. The general upper bound proven above is unfortunately not tight. Consider the (non-sparse) minimum bisection problem with $d = n/2$. Under the same general assumptions for the weights, it can be shown that a tighter bound holds for $\widehat{\beta} \leq \frac{1}{\sqrt{\log 2}\sigma}$

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z(\beta, X)] + \widehat{\beta}\mu\sqrt{N\log m}}{\log m} \leq 1 + \frac{\widehat{\beta}^2\sigma^2}{4}. \tag{5.13}$$

We prove (5.13) in Section 5.5.1. □

Proof of Theorem 5.1. To get a flavor of bounding $\mathbb{E}[\log Z]$ we first observe that $\mathbb{E}[\log Z] \leq \log \mathbb{E}[Z]$ (by Jensen's inequality). We can evaluate $\mathbb{E}[Z]$ as follows:

$$\begin{aligned}\mathbb{E}[Z] &= \mathbb{E}\Big[\sum_{c\in\mathcal{C}} \exp(-\beta R(c))\Big] \\ &= \exp(-\beta N\mu)\mathbb{E}\Big[\sum_{c\in\mathcal{C}} \exp\big(-\beta(R(c)-N\mu)\big)\Big] \\ &= \exp(-\beta N\mu) m \overline{G}^N(\beta). \end{aligned} \tag{5.14}$$

Thus

$$\log \mathbb{E}[Z] = -\beta N\mu + \log m + N \log \overline{G}(\beta) \tag{5.15}$$

since the set of random variables W_i belonging to the same solution are mutually independent variables. Throughout we write $\overline{R}(c) = R(c) - \mathbb{E}[R] = R(c) - N\mu$ for the centralized cost, and $\overline{G}(\beta)$ for the moment generating function of the negative centralized weight $\overline{W} = W - \mu$:

$$\overline{G}(\beta) \coloneqq \mathbb{E}\exp(-\beta \overline{W}). \tag{5.16}$$

We can expand $\overline{G}(\beta)$ into the Taylor series around zero and obtain

$$\overline{G}(\beta) = 1 + \frac{1}{2}\beta^2\sigma^2 + \mathcal{O}(\beta^3). \tag{5.17}$$

We find as long as $\beta \to 0$

$$\begin{aligned}\log \mathbb{E}[Z] &= -\beta N\mu + \log m + N \log \overline{G}(\beta) \\ &= -\beta N\mu + \log m + N \log\Big(1 + \frac{1}{2}\beta^2\sigma^2 + \mathcal{O}(\beta^3)\Big) \\ &= -\beta N\mu + \log m + \frac{1}{2}N\beta^2\sigma^2(1 + \mathcal{O}(\beta)). \end{aligned} \tag{5.18}$$

Now we apply the rescaling from the Common Theorem Setting

$$\beta = \widehat{\beta}\sqrt{\frac{\log m}{N}} \tag{5.19}$$

for some constant $\widehat{\beta}$ leading to

$$\frac{\log \mathbb{E}[Z] + \beta N\mu}{\log m} = 1 + \frac{1}{2}\widehat{\beta}^2\sigma^2(1 + \mathcal{O}(\beta)). \tag{5.20}$$

In terms of $\mathbb{E}[\log Z]$ we find

$$\frac{\mathbb{E}[\log Z] + \widehat{\beta}\mu\sqrt{N\log m}}{\log m} \leq 1 + \frac{1}{2}\widehat{\beta}^2\sigma^2\left(1 + \mathcal{O}\left(\sqrt{\frac{\log m}{N}}\right)\right). \tag{5.21}$$

But there is a surprise! Let us denote

$$\phi(\beta) = \mathbb{E}[\log Z] + \beta N\mu =: \mathbb{E}[\log \widehat{Z}(\beta)] \tag{5.22}$$

where $\widehat{Z}(\beta) = \sum_{c\in\mathcal{C}} \exp(\beta\overline{R}(c))$ with $\overline{R}(c) = -\sum_{i\in\mathcal{S}(c)} \overline{W}_i$. It is easy to observe that

$$\beta \max_{c\in\mathcal{C}} \overline{R}(c) \leq \log \widehat{Z}(\beta). \tag{5.23}$$

Using the upper bound obtained in (5.21) we find

$$\frac{\mathbb{E}[\max_{c\in\mathcal{C}} \overline{R}(c)]}{\log m} \leq \sqrt{\frac{N}{\log m}}\left(\widehat{\beta}^{-1} + \frac{1}{2}\widehat{\beta}\sigma^2\right). \tag{5.24}$$

Choosing a critical inverse temperature $\widehat{\beta}_c = \sqrt{2}/\sigma$ that minimizes the right-hand side of (5.24) we arrive at

$$\mathbb{E}[\max_{c\in\mathcal{C}} \overline{R}(c)] \leq \sqrt{2\sigma^2 N\log m} \tag{5.25}$$

Now proceeding as in Talagrand (2003, Proposition 1.1.3) we obtain

$$\phi'(\beta) \leq \mathbb{E}[\max_{c\in\mathcal{C}} \overline{R}(c)]. \tag{5.26}$$

But for $\beta > \beta_c \coloneqq \widehat{\beta}_c\sqrt{\log m/N}$,

$$\phi(\beta) \leq \phi(\beta_c) + \phi'(\beta_c)(\beta - \beta_c), \tag{5.27}$$

since $\phi(\beta)$ is known to be convex. Applying the upper bound for $\phi'(\beta)$ yields

$$\mathbb{E}[\log \widehat{Z}] \leq \widehat{\beta}\sigma\sqrt{2}\log m \tag{5.28}$$

and the upper bound for the second $\widehat{\beta}$ region is obtained. Observe that in this region the growth is linear with respect to $\widehat{\beta}$.

In summary, Theorem 5.1 is proven. □

For some combinatorial optimization problems, the asymptotic upper bound of Theorem 5.1 turns out to be tight. Below we present our two main results which give the asymptotically matching lower bounds for the Sparse MBP and Lawler QAP. For the Sparse MBP we develop a novel approach of proving it since the techniques proposed by Talagrand (2003, Chapter 1) seem not to work.

Theorem 5.2. *Consider the Sparse MBP complying with the Common Theorem Setting whose edge weights have mean μ and variance σ^2. Then the following holds:*

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z(\beta, X)] + \widehat{\beta}\mu\sqrt{N \log m}}{\log m} = \begin{cases} 1 + \frac{\widehat{\beta}^2\sigma^2}{2}, & \widehat{\beta} < \frac{\sqrt{2}}{\sigma}, \\ \widehat{\beta}\sigma\sqrt{2}, & \widehat{\beta} \geq \frac{\sqrt{2}}{\sigma} \end{cases} \tag{5.29}$$

provided $\log \ll d \ll n^{2/7}/\sqrt{\log n}$.

Let us now consider the Lawer QAP. In this case, we apply a slightly modified approach developed in Talagrand. However, we should point out that Lawer's QAP has some dependency that were not present in Derrida's model for which Talagrand proposed his method.

Theorem 5.3. *Consider Lawler QAP complying with Common Theorem Setting, whose matrix entries have mean μ and variance σ^2. Then the following holds:*

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z(\beta, X)] + \widehat{\beta}\mu\sqrt{N \log m}}{\log m} = \begin{cases} 1 + \frac{\widehat{\beta}^2\sigma^2}{2}, & \widehat{\beta} < \frac{\sqrt{2}}{\sigma}, \\ \widehat{\beta}\sigma\sqrt{2}, & \widehat{\beta} \geq \frac{\sqrt{2}}{\sigma}. \end{cases} \tag{5.30}$$

Remark. Proofs of both theorems will be given in separate Sections 5.3.4 and 5.3.5 due to their length.

5.3.3 Expected Log-Posterior Agreement Asymptotics

The above two theorems allow to operate (in a theoretically justified way) with the free energy, which, as we stated in the introduction, is a well-known *mathematical* (or *statistical mechanical*) problem. However, we would like to bring the connection to a more applied field as well: namely, to robust sampling. By that we refer once more to the definitions of log-posterior agreement (eLPA) and generalization capacity (see (5.3) and (5.6)). We recall the intuition behind these

two notions: they serve the purpose of selecting the "best" temperature (β) for a Gibbs distribution over the solutions to a combinatorial problem.

The matching lower bounds for sMBP and Lawler QAP given above in Theorems 5.2 and 5.3 allow us to present theoretical justification for the behavior of the posterior agreement kernel as shown in Figure 5.4. To see that, we observe that

$$\mathbb{E}_{X',X''} \log \sum_{c \in \mathcal{C}} p_\beta(c|X') p_\beta(c|X'') \tag{5.31}$$

$$= \mathbb{E}_{X',X''} \log \sum_{c \in C} \frac{\exp(-\beta(R(c,X') + R(c,X')))}{Z(\beta,X')Z(\beta,X'')}$$

$$= \mathbb{E}_{X',X''} \log Z(\beta,X',X'') - \mathbb{E}_{X'} \log Z(\beta,X') - \mathbb{E}_{X''} \log Z(\beta,X''), \tag{5.32}$$

where $Z(\beta,X',X'')$ can naturally be defined as a "partition function".

$$Z(\beta,X',X'') \coloneqq \sum_{c \in C} \exp(-\beta(R(c,X') + R(c,X''))). \tag{5.33}$$

Eventually this allows us to use the above theorems to compute all the three terms of (5.31).

To make the final step, we first need to formalize how exactly X' and X'' are obtained: let us assume that the two instances X' and X'' are both represented by two sets of weights $X' = \{W_i'\}$ and $X'' = \{W_i''\}$ through adding two "noise" instances $\delta X' = \{\delta W_i'\}$ and $\delta X'' = \{\delta W_i''\}$ to the same "signal" instance $X = \{W_i\}$ all the mentioned sets being of the same size $|\mathcal{S}|$:

$$W_i' = W_i + \delta W_i', \quad W_i'' = W_i + \delta W_i'' \quad \text{for} \quad i \in \mathcal{S}. \tag{5.34}$$

We also require that the signal and noise weights have certain means and variances:

$$\mathbb{E}[W_i] = \mu \qquad \mathrm{Var}[W_i] = \sigma^2 \tag{5.35}$$

$$\mathbb{E}[\delta W_i'] = \mathbb{E}[\delta W_i''] = 0 \qquad \mathrm{Var}[\delta W_i'] = \mathrm{Var}[\delta W_i''] = \tilde{\sigma}^2. \tag{5.36}$$

We define the *noise-to-signal ratio* as $\gamma = \tilde{\sigma}/\sigma$. Applying Theorems 5.2 and 5.3 we are led to the following result for the posterior agreement kernel (the consequences of this result will be discussed later in Section 5.4).

Theorem 5.4. *Consider Sparse MBP or Lawler QAP complying with the Common Theorem Setting. Let set X be "signal" weights with mean μ and variance σ^2 and*

two sets $\delta X'$, $\delta X''$ be "noise" with mean 0, and variance $\tilde{\sigma}^2$, all the sets of the same size. Let $X' = X + \delta X'$ and $X'' = X + \delta X''$ (elementwise sum) be the two problem instances. Let $\gamma := \tilde{\sigma}/\sigma$ be noise-to-signal ratio. Then the expectation of the log-posterior agreement (5.3) *satisfies*

$$\lim_{n\to\infty} \frac{\mathbb{E}_{X,\delta X',\delta X''} \log\big(|\mathcal{C}|\ \widehat{k}_\beta(X', X'')\big)}{\log m} = \eta(\widehat{\beta}), \tag{5.37}$$

where

$$\eta(\widehat{\beta}) = \begin{cases} (\widehat{\beta}\sigma)^2, & \widehat{\beta}\sigma < \frac{\sqrt{2}}{\sqrt{4+2\gamma^2}} \\ \widehat{\beta}\sigma\sqrt{2}\sqrt{4+2\gamma^2} - (\widehat{\beta}\sigma)^2(1+\gamma^2) - 1, & \frac{\sqrt{2}}{\sqrt{4+2\gamma^2}} \le \widehat{\beta}\sigma < \frac{\sqrt{2}}{\sqrt{1+\gamma^2}} \\ \widehat{\beta}\sigma\sqrt{2}\big(\sqrt{4+2\gamma^2} - 2\sqrt{1+\gamma^2}\big) + 1, & \frac{\sqrt{2}}{\sqrt{1+\gamma^2}} \le \widehat{\beta}\sigma \end{cases} \tag{5.38}$$

In particular, the expected log-posterior agreement is maximized at the eLPA-optimal inverse temperature:

$$\widehat{\beta}^* \equiv \widehat{\beta}^*_{\mathrm{eLPA}} = \frac{\sqrt{2+\gamma^2}}{\sigma(1+\gamma^2)}. \tag{5.39}$$

5.3.4 Proof of Theorem 5.2: Matching Lower Bound for sMBP

In this section we present a proof of the matching lower bound for Sparse MBP. The proof technique that we propose here is novel to the best of our knowledge and was also used in (see Magner et al., 2015, 2016).

The proof is broken into several lemmas. Let us start with defining D as elementwise overlap between two solutions (i.e. number of shared *edges*) sampled *uniformly at random*. We will refer to this uniform distribution as $\mathcal{D}$.

Lemma 5.5. *The following holds*

$$\frac{\#\{\textit{vertex-non-overlapping}\}}{m^2} = 1 - \Theta(d^2/n). \tag{5.40}$$

Proof. Observe that

$$\frac{\#\{\text{vertex-non-overlapping}\}}{m^2} = \frac{\binom{n}{d}\binom{n-d}{d}\binom{n-2d}{d}\binom{n-3d}{d}}{\binom{n}{d}^2\binom{n-d}{d}^2}$$

$$= \frac{\binom{n-2d}{d}\binom{n-3d}{d}}{\binom{n}{d}\binom{n-d}{d}}. \tag{5.41}$$

We now use Stirling's approximation, for any integer c to find

$$\begin{aligned}\binom{n-cd}{d} &\leq \frac{(n-cd)^d}{d!} \\ &= \frac{n^d(1-cd/n)^d}{d!} \\ &\sim \frac{n^d(1-cd^2/n)}{d!}.\end{aligned} \tag{5.42}$$

Similarly,

$$\begin{aligned}\binom{n-cd}{d} &\geq \frac{(n-(c+1)d)^d}{d!} \\ &= \frac{n^d(1-(c+1)d/n)^d}{d!} \\ &\sim \frac{n^d(1-(c+1)d^2/n)}{d!}.\end{aligned} \tag{5.43}$$

Applying these bounds we find

$$\frac{\#\{\text{vertex-non-overlapping}\}}{m^2} \leq \frac{(1-2d^2/n)(1-3d^2/n)}{(1-d^2/n)(1-2d^2/n)} \sim 1-2d^2/n \tag{5.44}$$

and

$$\frac{\#\{\text{vertex-non-overlapping}\}}{m^2} \geq \frac{(1-3d^2/n)(1-4d^2/n)}{(1-d^2/n)} \sim 1-6d^2/n. \tag{5.45}$$

This completes the proof. □

Lemma 5.6. *The following holds:*

$$\mathbb{P}_{\mathcal{D}}(D=0) \sim \frac{\#\{\textit{vertex-non-overlapping}\}}{m^2} \tag{5.46}$$

Proof.

$$\begin{aligned}\mathbb{P}_{\mathcal{D}}(D=0) = &\frac{\#\{\text{vertex-non-overlapping}\}}{m^2} \\ &+ \frac{\#\{\text{edge-non-overlapping} \mid \text{vertex-overlapping}\}}{m^2}.\end{aligned} \tag{5.47}$$

Since the following inclusion holds:

$$\{\text{edge-non-overlapping}|\text{vertex-overlapping}\} \subseteq \{\text{vertex-overlapping}\}, \tag{5.48}$$

we can conclude that

$$\begin{aligned}\frac{\#\{\text{edge-non-overlapping} \mid \text{vertex-overlapping}\}}{m^2} &\leq \frac{\{\text{vertex-overlapping}\}}{m^2} \\ = \frac{m^2 - \{\text{vertex-non-overlapping}\}}{m^2} &= 1 - 1 + \Theta(d^2/n) \\ &= o(1), \end{aligned} \tag{5.49}$$

where the last equation comes from Lemma 5.5 and $d = o(n)$. Hence, the the following holds:

$$\begin{aligned}\frac{\mathbb{P}_{\mathcal{D}}(D=0)}{\#\{\text{vertex-non-overlapping}\}/m^2} &= 1 \\ + \frac{\#\{\text{edge-non-overlapping} \mid \text{vertex-overlapping}\}/m^2}{\#\{\text{vertex-non-overlapping}\}/m^2} & \\ = 1 + \frac{o(1)}{1+o(1)} & \\ = 1 + o(1), & \end{aligned} \tag{5.50, 5.51}$$

which proves the lemma. □

These lemmas allow us to estimate the expected value of D.

Lemma 5.7. *The following holds:*

$$\mathbb{E}_{\mathcal{D}} D = \mathcal{O}(d^4/n). \tag{5.52}$$

Proof. To compute $\mathbb{E}_{\mathcal{D}} D$, observe

$$\begin{aligned}\mathbb{E}_D D &= 0 \cdot \mathbb{P}_{\mathcal{D}}(D=0) + \sum_{k=1}^{N} k \cdot \mathbb{P}_{\mathcal{D}}(D=k) \\ &\leq N \sum_{k=1}^{N} \mathbb{P}_{\mathcal{D}}(D=k) \\ &= d^2 \cdot \mathbb{P}_{\mathcal{D}}(D \neq 0) = d^2\big(1 - \mathbb{P}_{\mathcal{D}}(D=0)\big) \sim \Theta(d^4/n), \end{aligned} \tag{5.53}$$

where the last asymptotic equivalence follows from Lemmas 5.5 and 5.6. The less-than-equal sign turns Θ into $\mathcal{O}$. The lemma is proven. □

We also state the lemma which establishes the behavior of $\mathrm{Var}Z$.

Lemma 5.8. *For any $\beta > 0$ we have*

$$\mathrm{Var}Z = (\mathbb{E}Z)^2\Big(\mathbb{E}_{\mathcal{D}}\Big(\frac{G(2\beta)}{G^2(\beta)}\Big)^D - 1\Big) \tag{5.54}$$

where D is a random variable denoting the size of the elementwise overlap for two solutions $c, c' \in \mathcal{C}$, chosen uniformly at random (this uniformness is referred by $\mathcal{D}$). Here, $G(\beta)$ is the moment generating function of the negative weights $(-W_i)$.

Proof. Let

$$Z(\beta) = \exp(-\beta N\mu)\widehat{Z}(\beta), \tag{5.55}$$

where, as previously, $\widehat{Z}(\beta) = \sum_{c\in\mathcal{C}} \exp(\beta\overline{R}(c))$ with $\overline{R}(c) = -\sum_{i\in\mathcal{S}(c)} \overline{W}_i$. To compute $\mathrm{Var}\widehat{Z}$, we proceed as follows

$$\begin{aligned}\mathbb{E}\widehat{Z}^2 &= \mathbb{E}\Big[\sum_{c\in\mathcal{C}} \exp(\beta\overline{R}(c)) \cdot \sum_{c'\in C} \exp(\beta\overline{R}(c'))\Big] \\ &= \sum_{c,c'\in\mathcal{C}} \mathbb{E}\exp\Big(-\beta\Big(\sum_{i\in\mathcal{S}(c)} \overline{W}_i + \sum_{j\in\mathcal{S}(c')} \overline{W}_j\Big)\Big).\end{aligned} \tag{5.56}$$

Now define the elementwise overlap between the solutions c and c' as $\mathcal{S}_{\mathrm{ovr}}(c,c') \coloneqq \mathcal{S}(c) \cap \mathcal{S}(c')$, and its cardinality $d = d(c,c') \coloneqq |\mathcal{S}(c,c')|$. We also define the symmetric difference $\bar{\mathcal{S}}_{\mathrm{ovr}}(c,c') \coloneqq \mathcal{S}(c)\triangle\mathcal{S}(c')$ and continue the chain of equalities:

$$\mathbb{E}\widehat{Z}^2 = \sum_{c,c'\in\mathcal{C}} \mathbb{E}\exp\Big(-\beta\Big(2\sum_{i\in\mathcal{S}_{\mathrm{ovr}}(c,c')} \overline{W}_i + \sum_{j\in\bar{\mathcal{S}}_{\mathrm{ovr}}(c,c')} \overline{W}_j\Big)\Big). \tag{5.57}$$

Here the sets of weights $\mathcal{S}_{\mathrm{ovr}}(c,c')$ and $\bar{\mathcal{S}}_{\mathrm{ovr}}(c,c')$ are independent, allowing us to decompose the expectation into the product:

$$\begin{aligned}\mathbb{E}\widehat{Z}^2 &= \sum_{c,c'\in\mathcal{C}} \mathbb{E}\exp\Big(-\beta\Big(2\sum_{i\in\mathcal{S}_{\mathrm{ovr}}(c,c')} \overline{W}_i\Big)\Big) \cdot \mathbb{E}\exp\Big(-\beta\Big(\sum_{j\in\bar{\mathcal{S}}_{\mathrm{ovr}}(c,c')} \overline{W}_j\Big)\Big) \\ &= \sum_{c,c'\in\mathcal{C}} \big(\widehat{G}(2\beta)\big)^d\big(\widehat{G}(\beta)\big)^{2(N-d)} = \big(\widehat{G}(\beta)\big)^{2N} \sum_{c,c'\in\mathcal{C}} \Big(\frac{\widehat{G}(2\beta)}{(\widehat{G}(\beta))^2}\Big)^d.\end{aligned} \tag{5.58}$$

Now assume that the probability of the two solutions c and c', chosen uniformly at random, to have a d-element overlap is $\mathbb{P}_{\mathcal{D}}(d)$ and rewrite the above as follows:

$$\begin{aligned}\mathbb{E}\widehat{Z}^2 &= \big(\widehat{G}(\beta)\big)^{2N}\sum_{d=0}^{N} m^2\mathbb{P}_{\mathcal{D}}(d)\bigg(\frac{\widehat{G}(2\beta)}{(\widehat{G}(\beta))^2}\bigg)^d \\ &= m^2\big(\widehat{G}(\beta)\big)^{2N}\sum_{d=0}^{N} \mathbb{P}_{\mathcal{D}}(d)\bigg(\frac{\widehat{G}(2\beta)}{(\widehat{G}(\beta))^2}\bigg)^d \\ &= (\mathbb{E}\widehat{Z})^2\sum_{d=0}^{N} \mathbb{P}_{\mathcal{D}}(d)\bigg(\frac{\widehat{G}(2\beta)}{(\widehat{G}(\beta))^2}\bigg)^d. \end{aligned} \tag{5.59}$$

We conclude that

$$\mathrm{Var}\widehat{Z} = \mathbb{E}\widehat{Z}^2 - (\mathbb{E}\widehat{Z})^2 = (\mathbb{E}\widehat{Z})^2\bigg(\mathbb{E}_{\mathcal{D}}\bigg(\frac{\widehat{G}(2\beta)}{(\widehat{G}(\beta))^2}\bigg)^D - 1\bigg). \tag{5.60}$$

Recalling that $Z(\beta) = \exp(-\beta N\mu)\widehat{Z}(\beta)$ and, as well, $G(\beta) = \exp(-\beta\mu)\widehat{G}(\beta)$, we obtain the version without hats:

$$\mathrm{Var}Z = \mathbb{E}Z^2 - (\mathbb{E}Z)^2 = (\mathbb{E}Z)^2\bigg(\mathbb{E}_{\mathcal{D}}\bigg(\frac{G(2\beta)}{(G(\beta))^2}\bigg)^D - 1\bigg). \tag{5.61}$$

This proves Lemma 5.8. □

Now we are in the position to prove Theorem 5.2.

Proof of Theorem 5.2. Let us introduce an event A for some ϵ we choose later:

$$A \coloneqq \{Z \geq \epsilon\mathbb{E}Z\}. \tag{5.62}$$

This implies, by Chebychev inequality,

$$1 - \mathbb{P}(A) \leq \mathbb{P}\big(|Z - \mathbb{E}Z| \geq (1-\epsilon)\mathbb{E}Z\big) \leq \frac{\mathrm{Var}Z}{(1-\epsilon)^2(\mathbb{E}Z)^2}. \tag{5.63}$$

In Lemma 5.8 later we will prove the following:

$$\mathrm{Var}Z = (\mathbb{E}Z)^2\Big(\mathbb{E}_{\mathcal{D}}\Big(\frac{G(2\beta)}{G^2(\beta)}\Big)^D - 1\Big). \tag{5.64}$$

Expanding $G(2\beta)$ and $G^2(\beta)$ in Taylor's series, we find

$$\mathrm{Var}Z \sim (\mathbb{E}Z)^2\big(\sigma^2\beta^2\mathbb{E}_{\mathcal{D}}D\big). \tag{5.65}$$

Thus (5.63) can be further rewritten as

$$\begin{aligned} 1-\mathbb{P}(A) &\leq \frac{\mathrm{Var}Z}{(1-\epsilon)^2(\mathbb{E}Z)^2} \sim \frac{\sigma^2\beta^2\mathbb{E}_{\mathcal{D}}D}{(1-\epsilon)^2} \\ &= \mathcal{O}\bigg(\frac{\beta^2\mathbb{E}_{\mathcal{D}}D}{(1-\epsilon)^2}\bigg) \\ &= \mathcal{O}\bigg(\frac{d^4\log m}{n(1-\epsilon)^2N}\bigg) \\ &= \mathcal{O}\bigg(\frac{d^3\log n}{n(1-\epsilon)^2}\bigg), \end{aligned} \tag{5.66}$$

where we used Lemma 5.7 for $\mathbb{E}_{\mathcal{D}}D$ asymptotics.

We now proceed to compute $\mathbb{E}\log Z$ along the way of (Magner et al., 2015):

$$\mathbb{E}\log Z = \mathbb{E}[\log Z \mid A]\cdot\mathbb{P}(A) + \mathbb{E}[\log Z\mathbb{1}(\bar{A})] \tag{5.67}$$

$$\geq (\log\mathbb{E}Z + \log\epsilon)\mathbb{P}(A) + \mathbb{E}[\log Z\mathbb{1}(\bar{A})]. \tag{5.68}$$

But by (5.18) we find

$$\log\mathbb{E}Z = -\beta N\mu + \log m + \frac{1}{2}N\beta^2\sigma^2 + o(\beta^2). \tag{5.69}$$

Let the above expression be denoted as $L(\beta, N, m, \sigma)$ for the sake of brevity. So, using (5.66), we rewrite (5.68):

$$\mathbb{E}\log Z \geq \big(L(\beta,N,m,\sigma)+\log\epsilon\big)\cdot\bigg(1-\mathcal{O}\bigg(\frac{d^3\log n}{n(1-\epsilon)^2}\bigg)\bigg) + \mathbb{E}[\log Z\mathbb{1}(\bar{A})] \tag{5.70}$$

$$= L(\beta,N,m,\sigma) + \log\epsilon - \big(L(\beta,N,m,\sigma)+\log\epsilon\big)\cdot\mathcal{O}\bigg(\frac{d^3\log n}{n(1-\epsilon)^2}\bigg) \tag{5.71}$$

$$+ \mathbb{E}[\log Z\mathbb{1}(\bar{A})].$$

Thus,

$$\frac{\mathbb{E}\log Z + \beta N\mu}{\log m} \geq 1 + \frac{\widehat{\beta}^2\sigma^2}{2} + \frac{\log\epsilon}{\log m} - \big(L(\beta,N,m,\sigma)+\log\epsilon\big)\cdot\mathcal{O}\bigg(\frac{d^3\log n}{n(1-\epsilon)^2}\bigg) \tag{5.72}$$

$$+ \mathbb{E}[\log Z\mathbb{1}(\bar{A})].$$

Now we introduce below assumption (5.73) proved later to be true: assume

that

$$\frac{d^3 \log n}{n(1-\epsilon)^2} \to 0 \quad (n \to \infty). \tag{5.73}$$

Having that we notice

$$\big(L(\beta, N, m, \sigma) + \log \epsilon\big) \cdot \mathcal{O}\bigg(\frac{d^3 \log n}{n(1-\epsilon)^2}\bigg) = o(1), \tag{5.74}$$

i.e. it is small and thus is further neglected. So we can rewrite

$$\frac{\mathbb{E} \log Z + \beta N \mu}{\log m} \gtrsim 1 + \frac{\widehat{\beta}^2 \sigma^2}{2} + \frac{\log \epsilon}{\log m} + \frac{\mathbb{E}[\log Z \mathbb{1}(\overline{A})]}{\log m}. \tag{5.75}$$

We now estimate the term $\mathbb{E}[\log Z \mathbb{1}(\overline{A})]$. For some solution c,

$$\begin{aligned}
\mathbb{E}[\log Z \mathbb{1}(\overline{A})] &\geq \mathbb{E}[\log e^{-\beta R(c)} \mathbb{1}(\overline{A})] \\
&= \mathbb{E}[-\beta R(c) \cdot \mathbb{1}(\overline{A})] \\
&= \mathbb{E}[-\beta(\overline{R}(c) + \mathbb{E}R) \cdot \mathbb{1}(\overline{A})] \\
&= \mathbb{E}[-\beta \overline{R}(c) \mathbb{1}(\overline{A})] - \beta \mathbb{E}R \cdot \mathbb{P}(\overline{A}) \\
&\geq -\beta \mathbb{E}[|\overline{R}(c)|] - \beta \mathcal{O}(N)(1 - \mathbb{P}(A)) \\
&\geq -\beta \mathcal{O}(\sqrt{N}) - \beta \mathcal{O}\bigg(N \frac{d^3 \log n}{n(1-\epsilon)^2}\bigg).
\end{aligned} \tag{5.76}$$

Thus, essentially,

$$\begin{aligned}
\frac{\mathbb{E}[\log Z \mathbb{1}(\overline{A})]}{\log m} &\geq -\frac{\beta}{\log m} \mathcal{O}\bigg(N \frac{d^3 \log n}{n(1-\epsilon)^2}\bigg) \\
&\sim -\mathcal{O}\bigg(\frac{d^{7/2} \sqrt{\log n}}{n(1-\epsilon)^2}\bigg) \\
&= o(1)
\end{aligned} \tag{5.77}$$

provided $d = o(n^{2/7}/\sqrt{\log n})$ which we also write as $d \ll n^{2/7}/\sqrt{\log n}$. Consequently, (5.75) becomes

$$\frac{\mathbb{E} \log Z + \beta N \mu}{\log m} \gtrsim 1 + \frac{\widehat{\beta}^2 \sigma^2}{2} + \frac{\log \epsilon}{\log m} - \mathcal{O}\bigg(\frac{d^{7/2} \sqrt{\log n}}{n(1-\epsilon)^2}\bigg). \tag{5.78}$$

We will now choose ϵ in order to produce the lower bounds, and then check

that assumption (5.73) is satisfied. For $\widehat{\beta} > \widehat{\beta}_c \coloneqq \sqrt{2}/\sigma$ we choose

$$\epsilon = m^{-(1-\widehat{\beta}\sigma\sqrt{2}+\frac{\widehat{\beta}^2\sigma^2}{2})}.$$

This gives

$$\frac{\mathbb{E}\log Z + \beta N\mu}{\log m} \gtrsim \widehat{\beta}\sigma\sqrt{2} - o(1), \tag{5.79}$$

since for this choice $\epsilon = o(1)$, yielding

$$\mathcal{O}\left(\frac{d^{7/2}\sqrt{\log n}}{n(1-\epsilon)^2}\right) = o(1) \tag{5.80}$$

by our choice of d. For this choice of ϵ the assumption (5.73) holds.

For $\widehat{\beta} \leq \widehat{\beta}_c \coloneqq \sqrt{2}/\sigma$ we choose $\epsilon = 1/2$, yielding

$$\frac{\mathbb{E}\log Z + \beta N\mu}{\log m} \gtrsim 1 + \frac{\widehat{\beta}^2\sigma^2}{2} + o(1), \tag{5.81}$$

since

$$\frac{\log \epsilon}{\log m} = o(1), \quad \mathcal{O}\left(\frac{d^{7/2}\sqrt{\log n}}{n(1-\epsilon)^2}\right) = o(1) \tag{5.82}$$

and the assumption (5.73) holds. This completes the proof of Theorem 5.2. □

5.3.5 Proof of Theorem 5.3: Matching Lower Bound for Lawler QAP

In this section we present the proof of Theorem 5.3 for the matching lower bound for the Lawler QAP.

Theorem 5.1 gives us a general upper bound. To find the matching lower bound we follow the strategy of Talagrand (2003) that we briefly review. We should point out up front that Talagrand's technique was designed for proving the matching lower bound for the Random Energy Model (REM) without any dependency. In our case, there are clear dependency between solutions, however, not strong enough to destroy the the essence of our argument, thought the details are much more involved as discussed below.

To start, as in Talagrand, we define Y to be the cardinality of the solution subset for which the centered negative cost function $\overline{R}(c)$ is large enough, that is,

$$Y \coloneqq \operatorname{card}\{c\colon \overline{R}(c) \geq u_n(\widehat{\beta})\}, \tag{5.83}$$

where we set in our case

$$u_n(\widehat{\beta}) = \begin{cases} \widehat{\beta}\sigma^2\sqrt{N\log m}, & \widehat{\beta} < \widehat{\beta}_c \\ \widehat{\beta}_c\sigma^2\sqrt{N\log m}, & \widehat{\beta} \geq \widehat{\beta}_c. \end{cases} \tag{5.84}$$

We also define

$$a_n = \mathbb{P}(\overline{R}(c) \geq u_n(\widehat{\beta})), \tag{5.85}$$

and the event A as

$$A = \{Y \leq ma_n/2\}.$$

Observe that

$$\mathbb{E}[Y] = ma_n \tag{5.86}$$

and by Markov inequality

$$\mathbb{P}(A) \leq \mathbb{P}\big((Y - \mathbb{E}[Y])^2 \geq m^2a_n^2/4\big) \leq \frac{4\mathrm{Var}[Y]}{m^2a_n^2} \leq \frac{4\mathbb{E}[Y^2]}{m^2a_n^2} - 1. \tag{5.87}$$

To follow Talagrand's approach, we need to show that $\mathbb{E}[Y^2]/(ma_n)^2 \to 1$ so that $\mathbb{P}(A) \to 0$ which we formulate as the following lemma proved at the end of this section.

Lemma 5.9. *There exits $\varepsilon > 0$ such that*

$$\mathbb{P}(A) = \mathcal{O}(n^{-\varepsilon}) \tag{5.88}$$

for large n.

In order to establish it and present a concise proof of our matching bound, we need one more technical result regarding large deviations of $\mathbb{P}(\overline{R}(c) \geq u_n(\widehat{\beta}))$ formulated next.

Lemma 5.10. *Assume $\widehat{\beta}_c < \sqrt{2}/\sigma$. The following holds*

$$\log \mathbb{P}(\overline{R}(c) \geq u_n(\widehat{\beta})) = -\frac{u_n(\widehat{\beta})^2}{2N\sigma^2} + o(\log m) \tag{5.89}$$

where $u_n(\widehat{\beta})$ is defined in (5.84) above.

Proof. We will demonstrate the result only for $\widehat{\beta} < \widehat{\beta}_c$. The proof is identical for the right region of $\widehat{\beta}$. The main tool of this demonstration is the Gärtner-Ellis large-deviation theorem, as best described in (Dembo and Zeitouni, 2009). The following developments introduce the quantities at play in this theorem.

First observe that

$$\mathbb{P}(\overline{R}(c) \geq u_n(\widehat{\beta})) = \mathbb{P}\Big(\frac{1}{\sqrt{N \log m}} \overline{R}(c) \geq \widehat{\beta}\sigma^2\Big). \tag{5.90}$$

Define the logarithmic generating function of $\overline{R}(c)/\sqrt{N \log m}$:

$$\Lambda_n(\lambda) \coloneqq \log \mathbb{E}\Big[\exp\Big(\lambda \frac{1}{\sqrt{N \log m}} \overline{R}(c)\Big)\Big] = N \log \overline{G}\left(\frac{\lambda}{\sqrt{N \log m}}\right), \tag{5.91}$$

where $\overline{G}(\lambda)$ is the moment generating function of a negative centered weight $(-\overline{W})$. The last transition is valid because we assume that the weights within a solution are independent as expressed by assumption (C). Define the limiting moment generating function as

$$\Lambda(\lambda) \coloneqq \lim_{n\to\infty} \frac{1}{\log m} \Lambda_n(\lambda \log m) \tag{5.92}$$

$$= \lim_{n\to\infty} \frac{N}{\log m} \log \overline{G}\left(\lambda \sqrt{\frac{\log m}{N}}\right) \tag{5.93}$$

$$= \frac{\lambda^2 \sigma^2}{2} \tag{5.94}$$

because $\overline{G}(\lambda) = 1 + \frac{\lambda^2\sigma^2}{2} + o(\lambda^2)$ for $\lambda \to 0$.

The Gärtner-Ellis theorem (Dembo and Zeitouni, 2009, Theorem 2.3.6) yields

$$\lim_{n\to\infty} \frac{1}{\log m} \log \mathbb{P}(\overline{R}(c) \geq u_n(\widehat{\beta})) = -\Lambda^*(\widehat{\beta}\sigma^2) \tag{5.95}$$

where $\Lambda^*(x) = \sup_{\lambda>0} \lambda \cdot x - \Lambda(\lambda) = x^2/\left(2\sigma^2\right)$ is the Fenchel-Legendre transform of $\Lambda(\lambda)$. Hence,

$$\log \mathbb{P}(\overline{R}(c) \geq u_n(\widehat{\beta})) = -\frac{\widehat{\beta}^2\sigma^2}{2} \log m + o(\log m) \tag{5.96}$$

$$= -\frac{u_n(\widehat{\beta})^2}{N\sigma^2} + o(\log m) \tag{5.97}$$

The proof for $\widehat{\beta} \geq \widehat{\beta}_c$ is similar in all aspects. □

Now, we are in the position to prove Theorem 5.3 granted Lemma 5.9 that we prove at the end of this section. To accomplish it, we need a lower bound for $\mathbb{E}[\log \widehat{Z}(\beta)]$. We consider two cases conditioning on event A defined above and its

complementary event $\Omega\backslash A$.

We have on event $\Omega\backslash A$

$$\widehat{Z}(\beta) = \sum_{c\in\mathcal{C}_n} \exp(\beta\overline{R}(c)) \geq \sum_{c:\overline{R}(c)\geq u_n(\widehat{\beta})} \exp(\beta u_n(\widehat{\beta})) \tag{5.98}$$

$$\geq Y \exp(\beta u_n(\widehat{\beta})) \tag{5.99}$$

$$\geq \frac{ma_n}{2} \exp(u_n(\widehat{\beta})). \tag{5.100}$$

Therefore,

$$\mathbb{E}[\mathbb{1}_{\Omega\backslash A} \log \widehat{Z}(\beta)] \geq (1-\mathbb{P}(A)) \left(\log m + \log a_n - \log 2 + \beta u_n(\widehat{\beta})\right). \tag{5.101}$$

On event A, we derive the lower bound in the following way. Choosing an arbitrary solution c_0, we notice that $\widehat{Z}(\beta) \geq \exp(\beta\overline{R}(c_0))$ and thus

$$\mathbb{E}[\mathbb{1}_A \log \widehat{Z}(\beta)] \geq -\beta\mathbb{E}[-\mathbb{1}_A \overline{R}(c_0)] \geq -\beta\mathbb{E}[|\overline{R}(c_0)|] \geq -L\sigma\beta\sqrt{N} + o(1), \tag{5.102}$$

where L is some constant coming from expectation of half-normal distribution, which is the limiting distribution for $|\overline{R}(c_0)|$. Here we use the fact that $|\overline{R}(c_0)|$ converges in distribution to a half-normal (due to Central Limit Theorem, CLT), and then we determine that, due to the dominated convergence theorem and uniform integrability of $|\overline{R}(c_0)|$ (Feller, 1971, Ch. XVI.7), the expectation value of $|\overline{R}(c_0)|$ also converges to the one of half-normal.

Combining (5.101) and (5.102), we obtain

$$\mathbb{E}[\log \widehat{Z}(\beta)] \geq (1-\mathbb{P}(A)) (\log m + \log a_n - \log 2 + \beta u_n(\widehat{\beta})) - L\sigma\beta\sqrt{N} + o(1). \tag{5.103}$$

In summary, by Lemmas 5.9 and 5.10 we arrive at

$$\frac{\mathbb{E}[\log \widehat{Z}(\beta)]}{\log m} \geq 1 - \frac{u_n(\widehat{\beta})^2}{2\sigma^2 N \log m} + \frac{\beta u_n(\widehat{\beta})}{\log m} - \frac{L\sigma\beta\sqrt{N}}{\log m} + o(1) \tag{5.104}$$

$$= 1 - \frac{u_n(\widehat{\beta})^2}{2\sigma^2 N \log m} + \frac{\beta u_n(\widehat{\beta})}{\log m} + o(1). \tag{5.105}$$

Now for the regime $\widehat{\beta} < \widehat{\beta}_c$, recall that $u_n(\widehat{\beta}) = \widehat{\beta}\sigma^2\sqrt{N\log m}$, which yields

$$\frac{\mathbb{E}[\log \widehat{Z}(\beta)]}{\log m} \geq 1 + \frac{\widehat{\beta}^2\sigma^2}{2} + o(1). \tag{5.106}$$

For regime $\widehat{\beta} \geq \widehat{\beta}_c$, $u_n(\widehat{\beta}) = \widehat{\beta}_c \sigma^2 \sqrt{N \log m}$, hence

$$\frac{\mathbb{E}[\log \widehat{Z}(\beta)]}{\log m} \geq 1 - \frac{\widehat{\beta}_c{}^2 \sigma^2}{2} + \widehat{\beta}\widehat{\beta}_c \sigma^2 + o(1). \tag{5.107}$$

All in all, we have

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z(\beta, X)] + \widehat{\beta}\mu\sqrt{N \log m}}{\log m} \geq \begin{cases} 1 + \dfrac{\widehat{\beta}^2 \sigma^2}{2}, & \widehat{\beta} < \widehat{\beta}_c, \\ 1 - \dfrac{\widehat{\beta}_c{}^2 \sigma^2}{2} + \widehat{\beta}\widehat{\beta}_c \sigma^2, & \widehat{\beta} \geq \widehat{\beta}_c. \end{cases} \tag{5.108}$$

Theorem 5.3 is proven if we establish Lemma 5.9, which we do next. □

Proof of Lemma 5.9. First, note that if solutions (permutations) π and π' have k common points, then they share k^2 entries of the 4-dimensional matrix Q (out of the $N = n^2$ entries appearing in $R(\pi, X)$). Besides, since the solution space $\mathcal{C}_n$ (the set of permutations of size n) is a group, there exists a permutation π'' such that $\pi' = \pi \circ \pi''$. Thus, counting the common points between π and π' is equivalent to counting the fixed points of π'', which is a well-studied problem.

The number of permutations with k fixed points is the rencontre number (see e.g., Szpankowski (2001))

$$D_{n,k} = \frac{n!}{k!} \sum_{j=0}^{n-k} \frac{(-1)^j}{j!}. \tag{5.109}$$

Therefore, the number of ordered pairs of permutations sharing k fixed points is

$$B_{n,k} = n! D_{n,k} = \frac{n!^2}{k!} \sum_{j=0}^{n-k} \frac{(-1)^j}{j!}. \tag{5.110}$$

Similarly to the case of the sMBP, we can break down $\mathbb{E}[Y^2]$ using those elements

$$\mathbb{E}[Y^2] = \sum_{\pi,\pi' \in C_n} \mathbb{P}\left(\overline{R}(\pi, X) \geq u_n(\widehat{\beta}) \text{ and } \overline{R}(\pi', X) \geq u_n(\widehat{\beta})\right) \tag{5.111}$$

$$= \sum_{k=0}^{N} B_{n,k} \mathbb{P}\left(O_{n,k} + I_{n,k} \geq u_n(\widehat{\beta}) \text{ and } O'_{n,k} + I_{n,k} \geq u_n(\widehat{\beta})\right) \tag{5.112}$$

where $O_{n,k}, O'_{n,k} \sim \mathcal{N}(0, N - k^2)$, $I_{n,k} \sim \mathcal{N}(0, k^2)$ are independent of each other,

$I_{n,k}$ represents the sum of the entries shared by the two solutions, and $O_{n,k}, O'_{n,k}$ the entries exclusive to one of the two.

Let us now bound the probability

$$p_{n,k}(\widehat{\beta}) = \mathbb{P}\left(O_{n,k} + I_{n,k} \geq u_n(\widehat{\beta}) \text{ and } O'_{n,k} + I_{n,k} \geq u_n(\widehat{\beta})\right) \tag{5.113}$$

that two solutions with k^2 shared entries exceed the threshold $u_n(\widehat{\beta})$. This is exactly the probability that the two coordinates of a multivariate centered normal vector with covariance matrix $\begin{pmatrix} n^2 & k^2 \\ k^2 & n^2 \end{pmatrix}\sigma^2$ exceed $u_n(\widehat{\beta})$. Applying the results of (Savage, 1962) about multivariate Gaussian bounds, we have

$$p_{n,k} \leq \frac{\sigma^2}{2\pi u_n(\widehat{\beta})^2}\sqrt{\frac{(n^2+k^2)^3}{n^2-k^2}}\exp\left(-\frac{u_n(\widehat{\beta})^2}{(n^2+k^2)\sigma^2}\right) \tag{5.114}$$

$$= \frac{1}{2\pi\widehat{\beta}^2\sigma^2}\frac{1}{n^2\log n!}\sqrt{\frac{(n^2+k^2)^3}{n^2-k^2}}\exp\left(-\widehat{\beta}^2\sigma^2\frac{n^2\log n!}{(n^2+k^2)\sigma^2}\right) \tag{5.115}$$

for $k < n$.

For $k = n$, $p_{n,n} = a_n$ and we know that

$$a_n \sim \frac{n\sigma}{\sqrt{2\pi}u_n(\widehat{\beta})}\exp\left(-\frac{u_n(\widehat{\beta})^2}{2n^2\sigma^2}\right) \tag{5.116}$$

$$= \frac{1}{\sqrt{2\pi\log n!}\widehat{\beta}\sigma}\exp\left(-\frac{\widehat{\beta}^2\sigma^2}{2}\log n!\right). \tag{5.117}$$

Combining equations (5.110), (5.112), (5.115) and (5.117) yield

$$\frac{\mathbb{E}[Y^2]}{m^2a_n^2} \lesssim S_n = \sum_{k=0}^{n-1}\frac{1}{k!}\left(\sum_{j=0}^{n-k}\frac{(-1)^j}{j!}\right)\frac{1}{n^2}\sqrt{\frac{(n^2+k^2)^3}{n^2-k^2}}\mathrm{e}^{\left(1-\frac{n^2}{n^2+k^2}\right)\widehat{\beta}^2\sigma^2\log n!}$$
$$+ \frac{\sqrt{2\pi\log n!}\widehat{\beta}\sigma}{n!}\exp(\frac{\widehat{\beta}^2\sigma^2}{2}\log n!). \tag{5.118}$$

It is obvious that the term outside of the sum will tend to 0 as long as $\widehat{\beta} < \sqrt{2}/\sigma$. Let us now address the asymptotics of

$$S_n = \sum_{k=0}^{n-1}\frac{1}{k!}\left(\sum_{j=0}^{n-k}\frac{(-1)^j}{j!}\right)\frac{1}{n^2}\sqrt{\frac{(n^2+k^2)^3}{n^2-k^2}}\mathrm{e}^{\left(1-\frac{n^2}{n^2+k^2}\right)\widehat{\beta}^2\sigma^2\log n!}. \tag{5.119}$$

For that, set $k = o\Big(\frac{n}{\log n}\Big)$ and consider the following approximations of various terms of S_n.

First,

$$\sum_{j=0}^{n-k} \frac{(-1)^j}{j!} = \frac{1}{\mathrm{e}} + \mathcal{O}\Big(\frac{1}{(n-k)!}\Big). \tag{5.120}$$

Second,

$$\frac{1}{n^2}\sqrt{\frac{(n^2+k^2)^3}{n^2-k^2}} = 1 + \mathcal{O}\Big(\frac{k^2}{n^2}\Big). \tag{5.121}$$

Eventually,

$$\mathrm{e}^{\left(1-\frac{n^2}{n^2+k^2}\right)\widehat{\beta}^2\sigma^2\log n!} \sim \mathrm{e}^{\frac{k^2}{n^2}n\log n} = 1 + \mathcal{O}\Big(\frac{k\log n}{n}\Big). \tag{5.122}$$

Plugging back into S_n, we find

$$S_n = \sum_{k=0}^{\infty} \frac{1}{k!}\cdot\frac{1}{\mathrm{e}}\cdot\left(1+\mathcal{O}\Big(\frac{k^2}{n^2}\Big)\right)\cdot\left(\mathcal{O}\Big(\frac{k\log n}{n}\Big)\right) = 1 + \mathcal{O}\Big(\frac{1}{n^{\epsilon}}\Big), \tag{5.123}$$

provided that $k = \frac{n^{1-\epsilon}}{\log n}$. Thus,

$$\frac{\mathbb{E}[Y^2]}{m^2a_n^2} \to 1 + \mathcal{O}(n^{-\varepsilon}) \;\text{ and }\; \mathbb{P}(A) \leq \frac{4\mathrm{Var}[Y]}{m^2a_n^2} = \mathcal{O}(n^{-\varepsilon}). \tag{5.124}$$

This proves the lemma. □

5.4 Intermediary Discussion

Due to complexity of results obtained in this chapter, we would like to break a common rule and bring some intermediary discussion here, before proceeding to the next section.

First, the reader should notice that the free energy of Sparse MBP (5.29) and of Lawler QAP (5.30) exhibit a phase transition similar to that of Derrida's *Random Energy Model (REM)* (Derrida, 1981, Section V). We like to emphasize that Sparse MBP and Lawler QAP *introduce some correlation* between costs of pairs of solutions, while REM defines a technically much simpler setting without any correlations between cost values. Much more on that will be discussed in Chapter 6.

Second, regarding the behavior of expected log-posterior agreement (5.6) shown in Theorem 5.4, we can notice that it shows two phase transitions with quadratic,

mixed and linear phases, which corresponds to three phases of the Generalized REM, well explained in (Derrida and Gardner, 1986, Section 3). Its behavior is visualized in Fig. 5.6. We make the following brief observations, explaining the combinatorial meaning for an approximate learning process:

— The normalized eLPA in (5.38) depends on the temperature by the product $\widehat{\beta}\sigma$, which pronounces the fact that the reference scale for the temperature of Gibbs posteriors is adjusted by the amount of signal in data.

— The noise-to-signal ratio γ plays the crucial role. For a fixed signal σ, the optimal temperature $\widehat{\beta}^*$ grows to $\sqrt{2}/\sigma$, as the noise-to-signal vanishes ($\gamma \to 0$, i.e. $\tilde{\sigma} \ll \sigma$). This behavior supports our intuition that a posterior adapted to the signal variance is better to choose in the absence of noise. Optimal $\widehat{\beta}^*$ is located in the so-called *retrieval phase*.

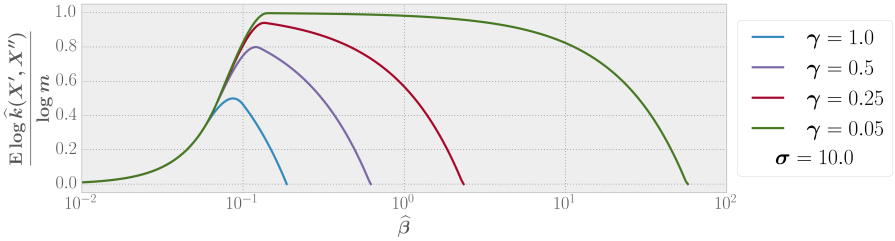


Figure 5.6 Behavior of the expected log-posterior agreement (5.6).

— The *high-temperature phase* $\widehat{\beta} \to 0$ results in low eLPA meaning that informative solutions cannot be found by sampling from Gibbs posteriors (which are too "broad") in this phase.

— *Freezing phase* $\widehat{\beta} \to \infty$: the decreasing expected low-posterior agreement reflects the instability of local minima under perturbations $\delta X'$, $\delta X''$. Solutions do not generalize and a learning algorithm cannot extract information from dataset X' and test it on X''.

Third, we compare the log-posterior agreement behavior to the previous experimental evidence of Chehreghani et al. (2012, Fig. 2), which shows the same shape of log-posterior agreement. Although the referred paper has an experimental nature and considers another optimization problem, it proves the concept in a nutshell, thereby showing that the theorems presented in this chapter support to the approach pioneered in (Buhmann, 2010b; Busse et al., 2013).

Forth, Theorems 5.2 and 5.3 allow to directly optimize the expected Gibbs risk

$$\mathbb{E}_{p_\beta(c|X),X}[R(c,X)] = -\frac{\partial}{\partial\beta}\mathbb{E}_X \log Z(\beta, X), \tag{5.125}$$

by means of applying differentiation to the results of these theorems on the right-

hand side. As a simple corollary, we thus obtain the following theorem:

Theorem 5.11 (Minimizing expected Gibbs risk)**.** *The expected Gibbs risk* (5.125) *is minimized at the GR-optimal inverse temperature:*

$$\widehat{\beta}^*_{\mathrm{GR}} := \arg\min_{\widehat{\beta}} \mathbb{E}_{p_\beta(c|X),X}[R(c,X)] = \frac{\sqrt{2+2\gamma^2}}{\sigma(1+\gamma^2)}. \tag{5.126}$$

It is interesting to compare GR-optimal (5.126) and eLPA-optimal (5.39) inverse temperatures:

$$\widehat{\beta}^*_{\mathrm{eLPA}} = \frac{\sqrt{2+\gamma^2}}{\sigma(1+\gamma^2)} \quad \text{and} \quad \widehat{\beta}^*_{\mathrm{GR}} = \frac{\sqrt{2+2\gamma^2}}{\sigma(1+\gamma^2)} \tag{5.127}$$

and not that they have a slight difference: eLPA selects slightly less (by a factor of $\sqrt{1+\frac{\gamma^2}{2+\gamma^2}}$) inverse temperature. This can be interpreted as follows: eLPA approach tends to be a bit more conservative, selecting slightly "broader" Gibbs posterior as opposed to expected Gibbs risk minimization.

5.5 Main Results Extension: a More General Case

5.5.1 A Tighter Upper Bound for Ordinary MBP

The general upper bound proven in Theorem 5.1 above is unfortunately not always tight. To show it we consider the general (non-sparse) minimum bisection problem with n vertices, that is, $d = n$. In this case, $N = |\mathcal{S}_n(c)| = n^2/4$ is the number of edges cut in a bisection, and $m = |\mathcal{C}_n| = \binom{n}{n/2}$ is the number of possible bisections.Thus we are still in our framework of $\log m = o(N)$, and therefore we define a scaling $\beta = \widehat{\beta}\sqrt{\log m/N}$.

Theorem 5.12. *For the general minimum bisection problem the following holds for* $\widehat{\beta} \leq \frac{1}{\sqrt{\log 2\sigma}}$

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z(\beta,X)] + \widehat{\beta}\mu\sqrt{N\log m}}{\log m} \leq 1 + \frac{\widehat{\beta}^2\sigma^2}{4}. \tag{5.128}$$

Remark. The idea of the proof is that the minimum bisection problem is a constrained version of the Sherrington-Kirkpatrick model, which is a spin model where all the spins are independent (cf. Sherrington and Kirkpatrick, 1975). In the minimum bisection problem, it is required that the partition of the graph be

balanced, or equivalently rephrased in spin model terms, it is required that there is the same number of up-spins as down-spins.

Therefore, the only difference between the two problems is the solution space. More precisely, we have $\mathcal{C}_n^{\mathrm{MBP}} \subset \mathcal{C}_n^{\mathrm{SK}}$. Hence

$$Z^{\mathrm{MBP}}(\beta) = \sum_{c \in \mathcal{C}_n^{\mathrm{MBP}}} \mathrm{e}^{-\beta R(c,X)} \leq \sum_{c \in \mathcal{C}_n^{\mathrm{SK}}} \mathrm{e}^{-\beta R(c,X)} = Z^{\mathrm{SK}}(\beta), \tag{5.129}$$

which allows us to extend any upper bound on Z^{SK} to Z^{MBP}. In particular, Talagrand provides such an upper bound in (Talagrand, 2003).

Proof of Theorem 5.12. First, we introduce some alternate notations for the minimum bisection problem in order to ease the transition to the Sherrington-Kirkpatrick formalism. Denote by G an undirected weighted complete graph with n vertices. The problem consists in finding a bisection of the graph (a partition in two subsets of equal size) of minimum cost. More formally, define by g_{ij} the weight assigned to the edge between vertices i and j ($g_{ij} = g_{ji}$). Denote by $c_i \in \{-1, 1\}$ an indicator of the subset containing vertex i.

We need to find $c \in \{-1, 1\}^n$ such that $\sum_i c_i = 0$ (balance condition) and the sum of the weights of cut edges

$$R(c, X) = \sum_{\substack{c_i = -c_j \\ i<j}} g_{ij} \tag{5.130}$$

is minimal. Here, X denotes a problem instance of size n, i.e. the particular values $(g_{ij})_{ij}$ of the edge weights.

Define the partition function as

$$Z(\beta, X) = \sum_{c \in \mathcal{C}_n} \mathrm{e}^{-\beta R(c,X)} \tag{5.131}$$

where $\mathcal{C}_n = \{c \in \{-1, 1\}^n | \sum_i c_i = 0\}$ is the solution space. Let us now prove Theorem 5.12. Observe that

$$\begin{aligned} \log Z(\beta, X) + \widehat{\beta}\mu\sqrt{N \log m} &= \log Z(\beta, X) + \beta\mu N \\ &= \log \sum_{c \in \mathcal{C}_n} \mathrm{e}^{-\beta(R(c,X) - N\mu)} = \log Z(\beta, \overline{X}), \end{aligned} \tag{5.132}$$

where the edge weights of $\overline{X}$ are defined by $\bar{g}_{ij} = g_{ij} - \mu$. Hence without loss of generality, we will only consider centered problem instances in the rest of the proof. For clarity, the explicit mention of the dependence to X is dropped in the

partition function, i.e. $Z(\beta, X) \coloneqq Z(\beta)$.

Then, let us relax our problem by allowing the partitions to be unbalanced:

$$Z^*(\beta) = \sum_{c \in \mathcal{C}_n^*} \mathrm{e}^{-\beta R(c)} \tag{5.133}$$

where $\mathcal{C}_n^* = \{-1, 1\}^n$ is the relaxed set of solutions. Since $\mathcal{C}_n \subset \mathcal{C}_n^*$, it follows that

$$Z(\beta) \leq Z^*(\beta). \tag{5.134}$$

Now rewrite the cost function as

$$R(c) = \sum_{\substack{c_i = -c_j \\ i<j}} g_{ij} = \frac{1}{2}\Big(\sum_{i<j} g_{ij} - \sum_{i<j} c_i c_j g_{ij}\Big) = \frac{1}{2}\Big(\sum_{i<j} g_{ij} + \sqrt{n} R^{\mathrm{SK}}(c)\Big) \tag{5.135}$$

where

$$R^{\mathrm{SK}}(c) = -\frac{1}{\sqrt{n}} \sum_{i<j} c_i c_j g_{ij} \tag{5.136}$$

is the cost function of the Sherrington-Kirkpatrick model. This entails

$$Z^*(\beta) = \mathrm{e}^{-\frac{\beta}{2}\sum_{i<j} g_{ij}} \sum_{c \in \mathcal{C}_n^*} \mathrm{e}^{-\frac{\sqrt{n}\beta}{2} R^{\mathrm{SK}}(c)} = \mathrm{e}^{-\frac{\beta}{2}\sum_{i<j} g_{ij}} Z^{\mathrm{SK}}\Big(\frac{\sqrt{n}\beta}{2}\Big) \tag{5.137}$$

where

$$Z^{\mathrm{SK}}(\beta) = \sum_{c \in \mathcal{C}_n^*} \mathrm{e}^{-\beta R^{\mathrm{SK}}(c)} \tag{5.138}$$

is the partition function associated with the Sherrington-Kirkpatrick model. Since the g_{ij} are centered, it follows that

$$\mathbb{E}\big[\log Z^*(\beta)\big] = \mathbb{E}\Big[\log Z^{\mathrm{SK}}\Big(\frac{\sqrt{n}\beta}{2}\Big)\Big]. \tag{5.139}$$

We need now the following statement from (Talagrand, 2003), that we present next.

Theorem 5.13 (Talagrand, 2003, Theorem 2.2.1)**.** *If $\beta < \frac{1}{\sigma}$, we have*

$$\lim_{n\to\infty} \frac{1}{n} \mathbb{E}\big[\log Z^{SK}(\beta)\big] = \frac{\beta^2\sigma^2}{4} + \log 2. \tag{5.140}$$

Having stated it, we continue by determining the limit of $\sqrt{n}\beta$:

$$\sqrt{n}\beta = \sqrt{n}\,\widehat{\beta}\sqrt{\frac{\log m}{N}} = \sqrt{n}\,\widehat{\beta}\sqrt{\frac{\log\binom{n}{n/2}}{n^2/4}} \sim \sqrt{n}\,\widehat{\beta}\sqrt{\frac{n\log 2}{n^2/4}} = 2\sqrt{\log 2}\,\widehat{\beta} \quad (5.141)$$

Thus, we can use Theorem 5.13 to obtain, for $\widehat{\beta} < \frac{1}{\sqrt{\log 2}\sigma}$

$$\lim_{n\to\infty}\frac{1}{n}\mathbb{E}\Big[\log Z^{\mathrm{SK}}\Big(\frac{\sqrt{n}\beta}{2}\Big)\Big] = \Big(\frac{\widehat{\beta}^2\sigma^2}{4}+1\Big)\log 2. \quad (5.142)$$

The equivalence $\frac{\log m}{n} \sim \log 2$ (in $n\to\infty$) and (5.139) both allow to write

$$\lim_{n\to\infty}\frac{\mathbb{E}[\log Z^*(\beta)]}{\log m} = \frac{\widehat{\beta}^2\sigma^2}{4}+1 \quad (5.143)$$

for $\widehat{\beta} < \frac{1}{\sqrt{\log 2}\sigma}$. Now (5.134) implies that

$$\lim_{n\to\infty}\frac{\mathbb{E}[\log Z(\beta)]}{\log m} \le \frac{\widehat{\beta}^2\sigma^2}{4}+1 \quad (5.144)$$

for $\widehat{\beta} < \frac{1}{\sqrt{\log 2}\sigma}$. □

5.5.2 Sampling Procedure for Simulating the Free Energy

To produce simulations of the partition function for any given optimization problem, we use a Metropolis-Hastings procedure to sample solutions at a given temperature $1/\beta$, coupled with an *importance sampling* cooling schedule scheme to efficiently sample solutions at low temperature levels. Below, we provide a brief review of importance sampling.

Importance sampling. Let us assume that samples from a distribution $\mathbb{Q}$ over a random variable X are given. Then, the expectation $\mathbb{E}_{\mathbb{P}}\phi(X)$ of a function $\phi(X)$ under a distribution $\mathbb{P}$ can be estimated by sampling X under $\mathbb{Q}$ with

$$\widehat{E}_N = \frac{1}{N}\sum_{i=1}^{N}\phi(X_i)\frac{\mathbb{P}(X_i)}{\mathbb{Q}(X_i)}, \quad \text{since} \quad \mathbb{E}_{\mathbb{Q}}\widehat{E}_N = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathbb{Q}}\phi(X_i)\frac{\mathbb{P}(X_i)}{\mathbb{Q}(X_i)} = \mathbb{E}_{\mathbb{P}}\phi(X). \quad (5.145)$$

This method is called the importance sampling since each sample is "re-weighted" using the target distribution.

Sampling from Gibbs distribution. We adapt importance sampling for the computation of Gibbs distribution partition functions. Suppose we have Gibbs distribution at temperature $1/\beta$ over a space $\mathcal{C}$ defined by a cost function $R : \mathcal{C} \to \mathbb{R}$. The probability of $c \in \mathcal{C}$ is $\mathbb{P}(c|\beta) = \mathrm{e}^{-\beta R(c)}/Z(\beta)$ where $Z(\beta) = \sum_{c\in\mathcal{C}} \mathrm{e}^{-\beta R(c)}$ is the partition function. Let us assume the partition function $Z(\beta)$ is given and we can sample from the Gibbs distribution at temperature $1/\beta'$. Then

$$Z_N^*(\beta, \beta') = \frac{1}{N} \sum_{i=1}^{N} Z(\beta) \mathrm{e}^{-(\beta'-\beta)R(c_i)} \tag{5.146}$$

is an unbiased estimator of $Z(\beta')$ when sampled under $\mathbb{P}(\cdot|\beta)$. Its precision is controlled by the relative variance:

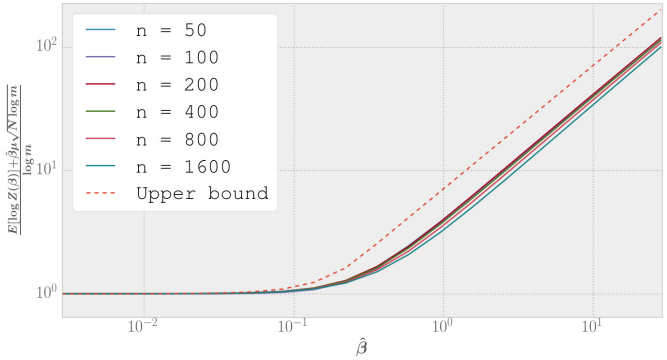
$$\mathrm{Var}^{\mathrm{rel}}_{\mathbb{P}(\cdot|\beta)} Z_N^*(\beta, \beta') = \frac{\mathrm{Var}_{\mathbb{P}(\cdot|\beta)} Z_N^*(\beta, \beta')}{\mathbb{E}^2_{\mathbb{P}(\cdot|\beta)} Z_N^*(\beta, \beta')} = \frac{1}{N} \left(\frac{Z(2\beta' - \beta) Z(\beta)}{Z(\beta')^2} - 1 \right). \tag{5.147}$$

Observe that when β differs significantly from β', the variance may be large, leading to poor simulations results. Furthermore, when β is close to β', the variance is small, thus simulations are more accurate.

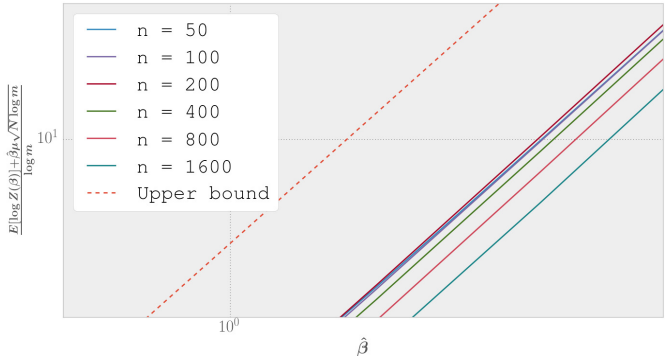
Cooling schedule. Our goal is to estimate the partition function for a wide range of β. The difficulties arise mostly for large values of β, since the partition function is then very concentrated. To overcome this, we apply our importance sampling philosophy and simulate first the partition function for small values of β (this makes the partition function more uniform and easier to estimate). Once we have computed the partition function for small β, we use equation (5.146) to evaluate it for the targeted value β'. But that is not the end of the story since we need to proceed in small steps using a cooling schedule $\beta_0 = 0 < \beta_1 < \cdots < \beta_k$ in order to reach the regions of the solution space contributing the most to the partition function. This is called a cooling schedule (Huber, 2012). In practice, we use a Metropolis-Hastings procedure to sample from the Gibbs distribution at a given temperature.

5.5.3 Simulation Results

Figure 5.7 shows the simulation of the free energy in the case of the minimum bisection problem ($d = n/2$) for different graph sizes n. The dashed line corresponds to the upper bound defined in Theorem 5.1. It appears that the general behavior



(a) Full view

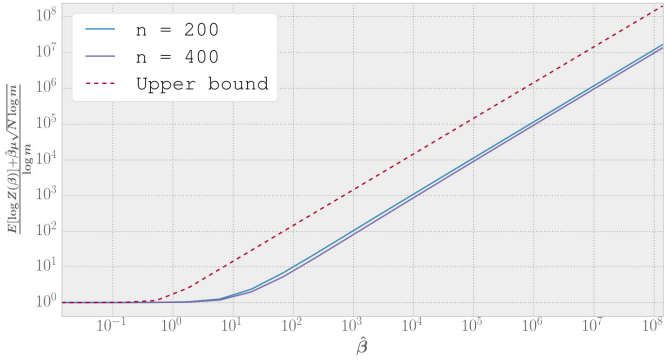


(b) A zoom on the right part of the plot

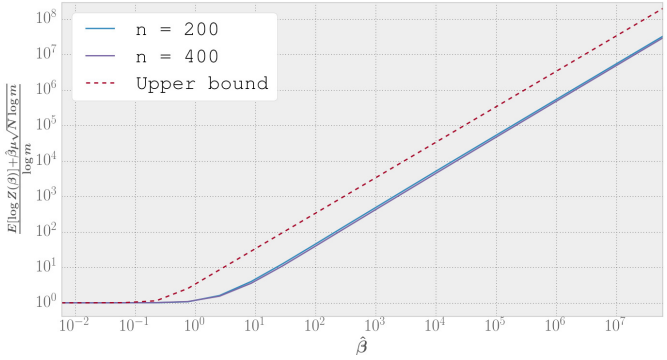
Figure 5.7 Second-order terms of the free energy rate in the case of the minimum bisection problem. The edge weights are i.i.d. and generated from a Gaussian distribution $\mathcal{N}(\mu = 20, \sigma = 5)$. Every curve is the average of 10 different problem instances. The curve labeled "upper bound" corresponds to the prediction of Theorem 5.1.

is quite good for the quadratic part of the free energy while for the linear part there is some discrepancy (a multiplicative factor correction is needed).

Figure 5.8 shows the simulation of the free energy in the case of the quadratic assignment problem for different graph sizes n. The dashed line corresponds to the upper bound defined in Theorem 5.1. The two plots correspond to different variances. Interestingly, in this problem the correction coefficient depends on the variance, which was not the case for the minimum bisection problem. Indeed, the correction coefficient is around $1/12$ for $\sigma = 1.0$ and near $1/8$ for $\sigma = 2.4$.



(a) Standard deviation of $\sigma = 1$



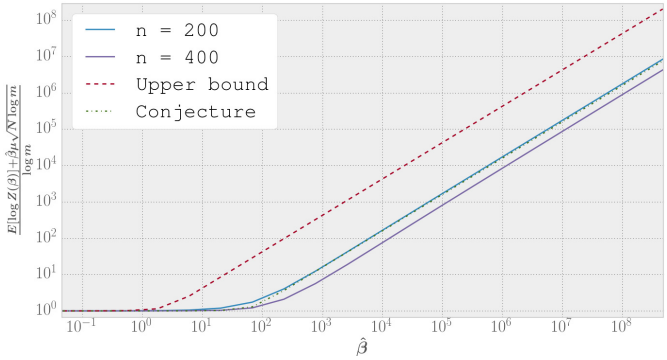
(b) Standard deviation of $\sigma = 2.4$

Figure 5.8 Second-order terms of the free energy rate in the case of the quadratic assignment problem. The distance and weight matrix entries are i.i.d. and generated from equal Gaussian distributions so that the product of two entries has mean $\mu = 4$ and varying standard deviation. Every curve is the average of 10 different problem instances. The curve labeled "upper bound" corresponds to the prediction of Theorem 5.1.

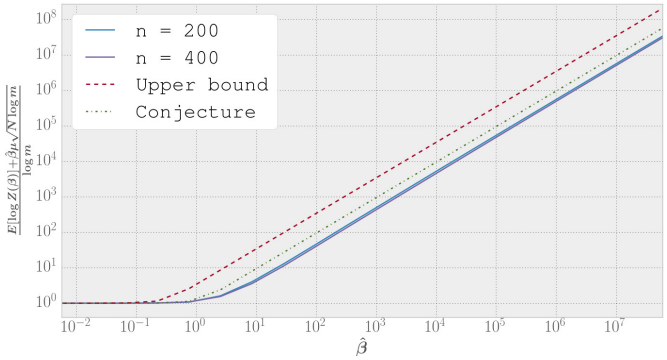
5.5.4 A Conjecture about Asymptotic Free Energy Behavior

Based on our empirical results presented in Figures 5.7 and 5.8, we are able to conjecture a more precise behavior of the free energy for the two optimization problems discussed in this chapter. We shall introduce a correction coefficient α whose value is determined experimentally in the sequel.

Conjecture 5.1. *Consider a class of combinatorial optimization problems complying with Common Theorem Setting, weights W_i having mean μ and variance σ^2.*



(a) $\sigma = 0.3$



(b) $\sigma = 2.4$

Figure 5.9 Influence of the correction in the case of the quadratic assignment problem for different standard deviation. The mean is $\mu = 4$ and $\mu_V = \mu_H$ and $\sigma_V^2 = \sigma_H^2$.

Then the free energy satisfies

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z(\beta, X)] + \widehat{\beta}\mu\sqrt{N\log m}}{\log m} = \begin{cases} 1 + \alpha^2 \frac{\widehat{\beta}^2\sigma^2}{2}, & \widehat{\beta} < \frac{\sqrt{2}}{\alpha\sigma} \\ \alpha\widehat{\beta}\sigma\sqrt{2}, & \widehat{\beta} \geq \frac{\sqrt{2}}{\alpha\sigma} \end{cases} \tag{5.148}$$

for some $\alpha \geq 1$.

The correction coefficient α is related to the variance of the partition function which involves strong correlations between feasible solutions (that was largely ignored in (Buhmann et al., 2014)). Based on our experimental results, we conclude

that α is well approximated by the following formula

$$\alpha = \sqrt{\frac{\mathbb{E}_X \mathrm{Var}_{\mathcal{D}} R(c,X)}{\mathbb{E}_{\mathcal{D}} \mathrm{Var}_X R(c,X)}} = \sqrt{\frac{\mathbb{E}_X \mathrm{Var}_c R(c,X)}{N\sigma^2}} \tag{5.149}$$

where the expectation $\mathbb{E}_{\mathcal{D}}[\cdot]$ and variance $\mathrm{Var}_{\mathcal{D}}[\cdot]$ are taken w.r.t. to all feasible solutions selected uniformly.

Important and surprising is the following (proof is trivial)

Statement 5.1. *For sMBP and Lawler QAP, Conjecture 5.1 turns into the proven asymptotics, since* $\alpha = 1$.

We will revisit the importance of this conjecture in the next chapter, in Section 6.4.

5.6 Discussion and Conclusion

This chapter investigates the asymptotics of the information score called the *expected log-posterior agreement* to validate cost functions and algorithms for "parameter rich" combinatorial optimization problems. We advise the reader to revisit Section 5.4, where we already started discussing the consequences of the main results.

As subtasks, first we provided rigorous derivations for free energy of Sparse MBP and Lawler QAP, results that had been wanting for some time. However, for general MBP and QAP we do not expect the lower bound to match the upper bound found in Theorem 5.1. In fact, based on extensive simulation we concluded that there is an additional scaling in the part of linear growth. To establish it, we realize that we need some new techniques to prove lower bounds that we developed.

Second, we showed that two second order phase transitions occur for the expected log-posterior agreement. Our analysis and experimental results show three regions of the the expected log-posterior agreement: a high temperature phase with low information, a retrieval phase and a disordered frozen phase. Only the retrieval phase can be used for efficient sampling solutions. While investigating the asymptotics of the log-posterior agreement and free energy we faced a challenging mathematical problem leading to some new research on the interplay between statistical physics and computation. We hope that techniques presented here can be successfully used for a large class of different combinatorial structures and problems.

We also have empirically-inspired conjectures for approximating free energy for general problems (i.e. for MBP, QAP, and potentially other problems), which

are well supported by our experiments and rigorous analysis for special cases (i.e. conjecture turns into proven asymptotics for Sparse MBP, Lawler QAP).

6

Does the Free Energy Define the Model Behavior?

"When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck."

— James Whitcomb RILEY (attr.)

6.1 Introduction

6.1.1 Motivation

Random combinatorial optimization problems exhibit a highly complex structure with a spin glass behavior (Mezard et al., 1987; Kirkpatrick et al., 1983). Optimization algorithms for these problems are slowed down by fluctuations in the problem instances when they search for solutions with low costs. Conceptually, we consider an optimization algorithm as a mapping from an input space of random instances to an output space of solutions and such algorithms should sample "typical" solutions from appropriate posterior distributions. In this chapter, we concentrate on maximum entropy sampling principles guided by Gibbs distributions to study information theoretic properties of random combinatorial optimization problems and their search landscape. Analytical computation of free energy, entropy and other macroscopic thermodynamical properties enables us to understand the solution structure of a large system, but — as already clarified in Chapter 5, — this goal has been known to be notoriously difficult and challenging from a mathematical standpoint (Talagrand, 2003).

While we solved this problem for specific cases (Chapter 5) with the purpose of applying in to robust optimization (Chapters 3 and 4), here we will be interested in a more general consequence of such results: a relation between the Random Energy Model (REM; see Derrida, 1981) and the Sparse Minimum Bisection Problem (sMBP; see Section 5.3.1).

Why are the relations between REM and sMBP of interest? The REM does not introduce any statistical dependencies between solutions. Therefore, optimization algorithms have to exhaustively inspect all exponentially many solutions of REM to find the one with minimal costs. Sparse Minimum Bisection introduces correlations between solutions but they are asymptotically so weak that they do not change the free energy. Since the free energy is the moment generating function of the Gibbs distribution we hypothesize that the Gibbs distributions of both problems are equivalent in terms of Kullback-Leibler divergences. If this claim would hold then we would not be able to efficiently search for low cost solutions of sMBP.

6.1.2 Contributions and Outline of the Chapter

In this chapter, we revisit the idea of characterizing structural information in solutions for combinatorial problems by information theoretic properties.

More specifically:

- we revisit results on the asymptotic behavior of the free energy (Chapter 5) on obtaining bounds on the free energy of solutions for the sMBP with random edge weights;
- these results reveal a remarkable phenomenon that the free energy of sMBP behaves very similarly to that of REM. Specifically, we show that the free energy of sMBP with random edge weights exhibits phase transitions equivalent to Derrida's REM;
- in order to deeper understand this observation and solution structure for dependent and independent solutions, we then make and prove statements about various ways sMBP and REM can be quantitatively related to each other: we show that the Kullback-Leibler divergence between Gibbs distributions induced by sMBP and REM are bounded, but not zero, which allows to make a conjecture about their complexity relations.

The chapter is organized as follows. As usual, we start with describing some of related work in Section 6.2. We then present and discuss our results about the similar behavior of REM and sMBP in Section 6.3.2. We speculate on the ways to interpret these results in Sections 6.4 and 6.5.

6.2 Background and Related Work Overview

Information theory, statistical mechanics and combinatorial optimization in large disordered systems have been disciplines enjoying several waves of intensive research. The first wave, associated exclusively with the statistical mechanics, was

marked by the works of Sherrington and Kirkpatrick (1975) or Derrida (1981) on mean-field models of spin glasses. For the reasons stated in the introduction, we concentrate on Derrida's solvable REM. In short, REM is the simplest example of a disordered system, whose configurations have i.i.d. energies and, therefore, are not efficiently "searchable". It will become important in the rest of the chapter that REM reflects the situation with no stochastic dependencies between solutions. This work inspired several continuations, of which we can mention, e.g., (Derrida and Gardner, 1986) as a generalization of REM, or (Aizenman et al., 1987) as exact solution of Sherrington-Kirkpatrick model (Sherrington and Kirkpatrick, 1975).

The second wave of interest was associated not exclusively with statistical mechanics, but also researched its connection to combinatorial optimization. Inspired by the work of Derrida, Vannimenus and Mézard (1984) considered the Traveling Salesman Problem (TSP) as a large disordered system which seeks to optimize its energy defined by respective *Hamiltonians* (see Section 2.3.1). This approach to view a combinatorial optimization problem from the statistical mechanics prospective turned out to be extremely fruitful: we recommend the book (Łuczak, 1994) as a good overview of the results. We should also mention here the work (Auffinger and Chen, 2014) who studied algorithmic complexity from the statistical mechanics viewpoint.

Finally, in the last two decades, many attempts have been made to systematize approaches traditionally used in statistical mechanics and render them rigorous in a mathematical sense. Here, we point out the work by Bovier et al. (2002), as well as extensive reviews by Talagrand (2003); Bovier (2012).

6.3 Comparison of REM and sMBP

6.3.1 Random Energy Model (REM)

The REM introduced by Derrida (1981) is a model $\mathcal{P}^{\text{rem}} = (\mathcal{X}, \mathcal{C}^{\text{rem}}, R^{\text{rem}})$ where the following conditions apply:

1. Number of solutions (in the original terminology, *configurations*) equals

$$|\mathcal{C}^{\text{rem}}| = 2^K. \tag{6.1}$$

2. Here, the data source $X \in \mathcal{X}$ is a vector of 2^K Gaussian random variables (for the notation, see Definition 3.1), and all solutions $c_i \in \mathcal{C}^{\text{rem}}$ carry costs (in the original terminology, *Hamiltonians* or *energy levels*)

$$R^{\text{rem}}(c_i, X) = X_i, \quad \text{where} \quad X_i \sim \mathcal{N}(0, \sigma^2) \tag{6.2}$$

3. The costs X_i are i.i.d.

As Derrida noted in his paper, "the third property is specific to this model. It simplifies the model enough to allow us to solve it exactly". While it is true for REM, such independence is not characteristic for the most of the models. The next section which discusses dependencies in sMBP.

6.3.2 Similar Behavior of REM and sMBP

Earlier, we introduced the sparsity constraint on d in Theorem 5.2 because without it, the stochastic dependency between two random solutions for original MBP (Garey and Johnson, 1979) is very high. Indeed, in original MBP, any two bisections *always* share edges. By introducing d we: a) substantially reduce such dependency, but on the other hand b) do not eliminate it at all, like in REM (Derrida, 1981).

But by introducing sparsity, didn't we essentially *transform* MBP into REM? E.g. one can observe that for classical dependency-free REM, the free energy asymptotics looks just the same, in particular exhibits the same phase transition and same phase shapes:

Theorem 6.1 (adapted formulation from (Talagrand, 2003)). *Assume $m = 2^K$ is the number of configurations for the REM model with Gaussian cost values, with parameters $\mathcal{N}(0, \tau^2)$. Then the free energy rate is (asymptotically in $n \to \infty$) equal to*

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z]}{\log m} = \begin{cases} \frac{\beta^2\tau^2}{2\log m} + 1 & \beta < \sqrt{2\log m}/\tau, \\ \frac{\beta\tau\sqrt{2}}{\sqrt{\log m}} & \beta \geq \sqrt{2\log m}/\tau. \end{cases} \tag{6.3}$$

We note that Talagrand formulated it in a more general setting (cf. Talagrand, 2003, Prop. 1.1.3), which is adapted here for clarity. Choosing $\tau = \sigma\sqrt{N}$ and applying β rescaling from 5.6, we arrive at an equivalent formulation: for Gaussian cost values with parameters $(0, \sigma^2 N)$, we derive

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z]}{\log m} = \begin{cases} 1 + \frac{\widehat{\beta}^2\sigma^2}{2}, & \widehat{\beta} < \frac{\sqrt{2}}{\sigma}, \\ \widehat{\beta}\sigma\sqrt{2}, & \widehat{\beta} \geq \frac{\sqrt{2}}{\sigma} \end{cases} \tag{6.4}$$

Remark. For non-centered cost values, the necessary correction similar to the one of (5.29) should be made on the left-hand side which is trivial.

We are now going to sketch an answer to the following question: how much does sMBP look like REM? We give the following result and then discuss it. First, let's make some definitions.

Definition 6.1. *For an sMBP setting stated in Theorem 5.2, we will call equivalent such a REM, for which the number of configurations is equal to m and the cost values are Gaussian with parameters $(\mu N, \sigma^2 N)$.*

For convenience of the following explanation, let us denote the random source behind such a REM as Y (analogically to X in case of sMBP). Hence for REM, the cost values $R(c, Y) \sim \mathcal{N}(\mu N, \sigma^2 N)$ and all are independent. We denote the respective Gibbs distributions $p_\beta^{\mathrm{smbp}}(c|X)$ and $p_\beta^{\mathrm{rem}}(c|Y)$.

Theorem 6.2. *The rate of the KL-divergence between configurations' Gibbs distributions for sMBP and equivalent REM is non-zero and exhibits a phase transition.*

$$\frac{\mathbb{E}_{X,Y}[\mathcal{D}^{KL}(p_\beta^{\mathrm{rem}} \| p_\beta^{\mathrm{smbp}})]}{\log m} = \begin{cases} \widehat{\beta}^2\sigma^2 & \widehat{\beta} < \frac{\sqrt{2}}{\sigma}, \\ \widehat{\beta}\sigma\sqrt{2}, & \widehat{\beta} \geq \frac{\sqrt{2}}{\sigma} \end{cases} \tag{6.5}$$

Proof. In the first part of the proof, we will omit the expectation for the sake of brevity.

$$\begin{aligned}
\mathcal{D}^{\mathrm{KL}}(p_\beta^{\mathrm{rem}} \| p_\beta^{\mathrm{smbp}}) &= \sum_c p_\beta^{\mathrm{rem}}(c|Y) \log \frac{p_\beta^{\mathrm{rem}}(c|Y)}{p_\beta^{\mathrm{smbp}}(c|X)} = \\
&= \sum_c p_\beta^{\mathrm{rem}}(c|Y) \left(\log \frac{e^{-\beta R^{\mathrm{rem}}(c,Y)}}{Z^{\mathrm{rem}}(Y)} \right. \\
&\qquad \left. - \log \frac{e^{-\beta R^{\mathrm{smbp}}(c,X)}}{Z^{\mathrm{smbp}}(X)} \right) \\
&= \sum_c p_\beta^{\mathrm{rem}}(c|Y) \Big(\log e^{-\beta R^{\mathrm{rem}}(c,Y)} \\
&\qquad - \log e^{-\beta R^{\mathrm{smbp}}(c,X)} \\
&\qquad - \log Z^{\mathrm{rem}}(Y) + \log Z^{\mathrm{smbp}}(X) \Big) \\
&= -\beta \sum_c p_\beta^{\mathrm{rem}}(c|Y) \Big(R^{\mathrm{rem}}(c,Y) - R^{\mathrm{smbp}}(c,X) \Big) \\
&\qquad - \log Z^{\mathrm{rem}}(Y) + \log Z^{\mathrm{smbp}}(X)
\end{aligned} \tag{6.6}$$

Returning to the expectation $\mathbb{E}_{X,Y}$ and recalling that all the sMBP-related terms depend on X and all the REM-related terms depend on Y, we can continue:

$$-\beta \mathbb{E}_{X,Y} \left[\sum_c p_\beta^{\mathrm{rem}}(c|Y) \Big(R^{\mathrm{rem}}(c,Y) - R^{\mathrm{smbp}}(c,X) \Big) \right]$$

$$
\begin{aligned}
&\underbrace{-\mathbb{E}_Y[\log Z^{\mathrm{rem}}(Y)] + \mathbb{E}_X[\log Z^{\mathrm{smbp}}(X)]}_{\text{cancel out due to Thm. 5.2 and 6.1}} \\
&= -\beta\Big(\mathbb{E}_Y\Big[\sum_c p_\beta^{\mathrm{rem}}(c|Y) R^{\mathrm{rem}}(c,Y)\Big] \\
&\qquad - \mathbb{E}_Y \underbrace{\sum_c p_\beta^{\mathrm{rem}}(c|Y)}_{1} \cdot \underbrace{\mathbb{E}_X R^{\mathrm{smbp}}(c,X)}_{\mu N}\Big) \\
&= \beta\mu N + \beta\mathbb{E}_Y\Big[\frac{d}{d\beta}\log Z^{\mathrm{rem}}(Y)\Big]. \qquad (6.7)
\end{aligned}
$$

By the argument of dominated convergence theorem, we can under mild conditions interchange expectation and differentiation, which together with (6.3) and (6.4) leads to

$$
\mathbb{E}_{X,Y}[\mathcal{D}^{\mathrm{KL}}(p_\beta^{\mathrm{rem}} \| p_\beta^{\mathrm{smbp}})] = \begin{cases} \widehat{\beta}^2\sigma^2 \log m & \widehat{\beta} < \frac{\sqrt{2}}{\sigma}, \\ \widehat{\beta}\sigma \log m\sqrt{2}, & \widehat{\beta} \geq \frac{\sqrt{2}}{\sigma}, \end{cases} \qquad (6.8)
$$

which completes the proof of theorem. □

6.3.3 Consequences of Similar Behavior of REM and sMBP

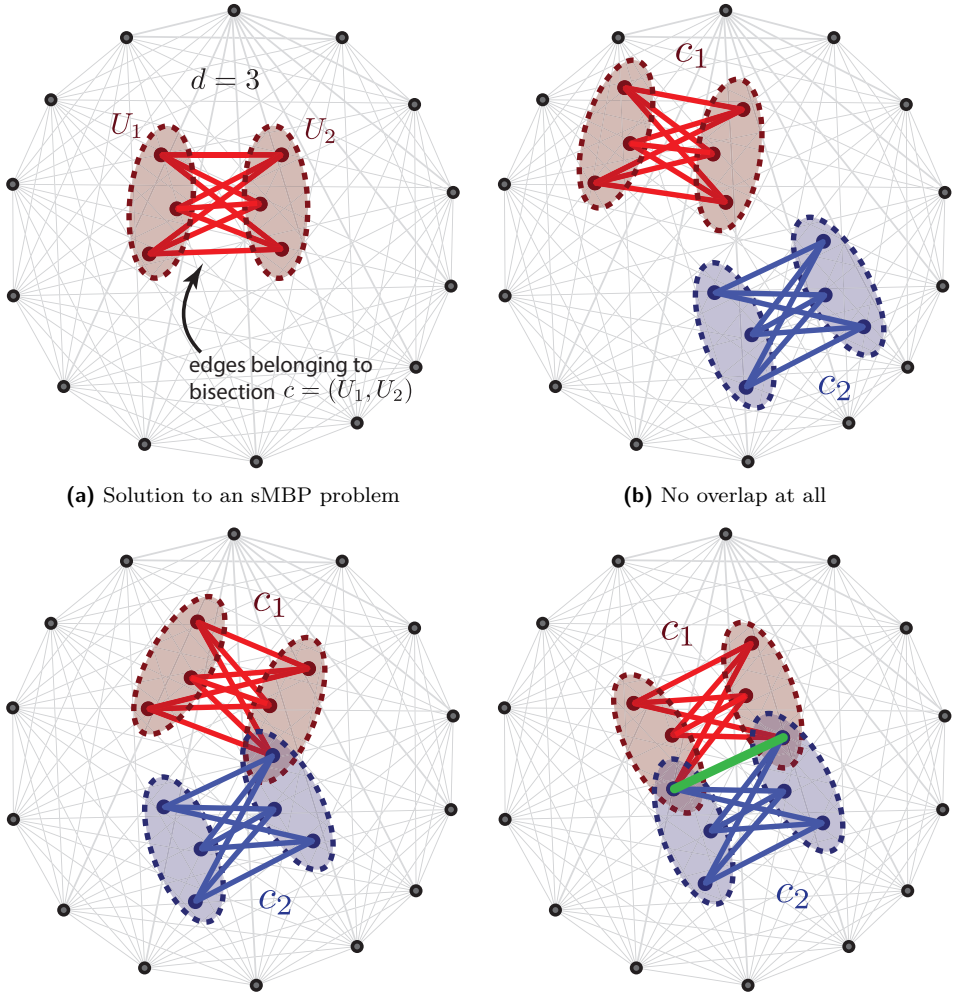
We now discuss this result. There are several observations to be made about the whole line of research reflected in Theorems 5.2, 6.1 and 6.2.

First, Theorems 5.2, 6.1 and 6.2 can be interpreted as follows: they yield the similarity of both problems in terms of macroscopic thermodynamical properties like free energy rate, but at the same time they convey their difference in terms of KL-divergence rate: by definition of $\mathcal{D}^{\mathrm{KL}}$, it measures the amount of information one might gain (or lose) by assuming p^{rem} instead of p^{smbp} or vice versa. The fact that the expected KL-divergence rate is non-zero allows us to say that sMBP and REM are still different in terms of their distributions.

Second, and probably most importantly, due to the same line of reasoning as in Theorem 6.2, we can conclude that a KL-divergence between every pair $(p^{\mathrm{rem}\prime}, p^{\mathrm{rem}\prime\prime})$ of independent (that is, with independent Y' and Y'') equivalent REMs is identical to that of $(p^{\mathrm{rem}}, p^{\mathrm{smbp}})$. This highlights the fact that for full understanding of relations (similarities and differences) between them one needs to properly quantify this difference in terms of higher moments and not only *expectation* of KL-divergence rate.

It is important to realize that the higher moments of the KL-divergence can be most likely controlled by the higher moments of $\log Z$, i.e. $\mathrm{Var}[\log Z]$ and so on.

The necessary understanding of the behavior of $\mathrm{Var}[\log Z]$, in turn, can be reached via careful computing of higher moments of edge overlap D of the two



(a) Solution to an sMBP problem

(b) No overlap at all

(c) Vertex overlap, but no edge overlap

(d) Only edge overlap counts

Figure 6.1 Illustration of various types of overlaps. Only case **(d)** contributes to statistical dependence between costs of solutions. Carefully computing edge overlap can make a huge step forward in understanding higher moments of $\log Z$ (in Lemma 5.7 we computed only expected value).

solutions c_1 and c_2 (illustrated in Figure 6.1): one can see this, for example, from the proof of Theorem 5.2, where we have

$$\mathrm{Var} Z \sim (\mathbb{E}Z)^2 \left(\sigma^2 \beta^2 \mathbb{E}_{\mathcal{D}} D\right)$$

as an intermediate result. There exists evidence of a possibility to use Taylor expansions to express higher moments of $\log Z$ via those of Z.

6.4 Comparison of REM and non-sparse MBP

Interesting question arises: can we say anything about non-sparse (i.e. case when $d \not\ll n$) MBP? Note that in Section 5.5 of Chapter 5 we made a conjecture which is backed up by extensive simulations, which we restate here:

Conjecture 5.1 (see Section 5.5). *Consider a class of combinatorial optimization problems complying with Common Theorem Setting, weights W_i having mean μ and variance σ^2. Then the free energy satisfies*

$$\lim_{n\to\infty} \frac{\mathbb{E}[\log Z(\beta, X)] + \widehat{\beta}\mu\sqrt{N \log m}}{\log m} = \begin{cases} 1 + \alpha^2 \frac{\widehat{\beta}^2\sigma^2}{2}, & \widehat{\beta} < \frac{\sqrt{2}}{\alpha\sigma} \\ \alpha\widehat{\beta}\sigma\sqrt{2}, & \widehat{\beta} \geq \frac{\sqrt{2}}{\alpha\sigma} \end{cases}$$

for some $\alpha \geq 1$, where α is well approximated by

$$\alpha = \sqrt{\frac{\mathbb{E}_X \mathrm{Var}_{\mathcal{D}} R(c, X)}{\mathbb{E}_{\mathcal{D}} \mathrm{Var}_X R(c, X)}} = \sqrt{\frac{\mathbb{E}_X \mathrm{Var}_{\mathcal{D}} R(c, X)}{N\sigma^2}}$$

It is instructive to note here that, loosely speaking, parameter α represents the ratio variability across solutions (nominator) and the variability inside each solution (denominator). The higher the dependence is, the less variability across solutions exists.

Apparently $\alpha = 1$ corresponds to a case of no dependencies (sMBP), while lower alpha corresponds to higher levels of dependencies. And again, to further back up this conjecture, one has to develop the understanding of $\mathrm{Var}_{\mathcal{D}} R(c, X)$, i.e. variance of costs for uniformly chosen solutions, which boils down to taking overlaps under control (computing $\mathrm{Var}_{\mathcal{D}}[D]$ analogically to $\mathbb{E}_{\mathcal{D}}[D]$ as we did in Lemma 5.7.

6.5 Discussion and Conclusion

The geometry of random combinatorial optimization problems is often characterized by exponentially many local minima which prevent an efficient search for low cost solutions. In this chapter, we have studied random instances of sparse Minimum Bisection Problem which exhibit a statistical behavior very similar to that of the Random Energy Model. This similarity between the Gibbs distributions for

REM instances and for sMBP instances is documented by identical values for the free energies and for the Kullback-Leibler divergences between pairs of distributions.

Furthermore, this equivalence also suggests that the computational complexity of REM and sMBP might be the same and, consequently, random instances of sMBP might not be efficiently optimized due to the lack of search information as in REM. As a future research direction we plan to analyze the finite $\log m$ corrections to estimate the convergence rate of the free energy toward its asymptotic limit. Another open question remains if there exist REM instances that are arbitrarily close (w.r.t. KL-divergence) to sMBP instances in the asymptotic limit, explaining why we cannot find efficient optimization schemes for sparse Minimum Bisection.

For answering the last question, we have highlighted the importance of developing a better understanding of the higher moments of the solutions overlap (Figure 6.1), since the proofs we gave earlier yield that controlling the dependency between the solutions gives us the essential information about the problem's (dis)similarity from REM, where such dependencies don't exist at all.

7
Concluding Remarks

"The measure of greatness in a scientific idea is the extent to which it stimulates thought and opens up new lines of research."

— Paul DIRAC

In this thesis, we addressed the problem of robust approximate optimization in different settings: general, algorithmic and thermodynamic, and also looked into the thermodynamic behavior of optimization problem in a more general (i.e. not related to approximation) sense. Because detailed remarks are given at the end of each chapter, here we give some very general thoughts about possible directions of the further research.

7.1 Approximate Optimization in General

It would be interesting to extend the approaches presented in Chapter 3 to the case of more than two instances. While some straightforward generalizations may exist — e.g. mechanistically extend the formula for ASC score, adding more terms in nominator and denominator, — it becomes unclear how to justify it from the point of view of coding theory. It can require modifying the definition of the channel which we presented in that chapter.

Further, while we partially addressed the question of computing intersection cardinalities, this result relies on knowing distributions, which renders it unusable in practice or at least requires the use of plug-in estimators. It would be beneficial to design a class of problems for which this issue is eliminated. We partially did this for algorithmic problems, but at the cost of worse performance, as noted in the discussion of that chapter.

Last but not least, there is hope that one can integrate our approach into a toolbox of stability-related approaches (see related work), because our approach

essentially attacks the problem of identifying stability conditions (by imposing approximations). It would be interesting to see a connection of our approach with more conventional techniques.

7.2 Robust Algorithmic Optimization

As we have seen, the ASC score establishes a ranking of algorithms, which also yields a corresponding ranking of their localization errors. Is this by chance or can it be proven rigorously? Another question here arises. The Reverse-Delete algorithm requires many more steps than Prim's and Kruskal's (see experiments in Chapter 4), but gains much better robustness. Obviously, this happens because Reverse-Delete is much more elaborate in exploring the graph: Prim's and Kruskal's algorithms eliminate unexplored edges much more aggressively, and thus "skip" a lot of opportunities without actually seeing them. We have a clear runtime vs. robustness trade-off, which raises a question: can one construct an intermediate algorithm using the above three as building blocks?

Another massive task would be to study more algorithmic problems from this perspective. It must be recalled that the extension of ASC for algorithms was performed in some sense "blindly", since proving the communication error bounds (in analogy to that of Chapter 3) is hard in this case. Consequently, it might turn out that with other algorithmic problems this approach works much better/worse. Additional research would be beneficial here.

7.3 Thermodynamic Behavior of Optimization Problems

Although we have proven (Chapter 5) the asymptotics of free energy in two specific cases, we still did not devise a general methodology. Although we are currently under an impression that there is no such general methodology, more attempts should be made.

Next, we made and experimentally backed up an attractive conjecture about the behavior of free energy in more general cases than those where theoretic results were obtained. We brought up an intuitive explanation for it, and it is extremely interesting to continue a line of research on that.

Further, it is still not clear how far one can go with the sparsity constraint (which, should be recalled, was introduced to reduce the influence of interactions between solutions without fully eliminating it). Current condition $\log n \ll d \ll n^{2/7}$ might be extended, if one uses more advanced bounding techniques.

Finally, but no less importantly, the free energy behavior of Gibbs-regularized combinatorial optimization problems like sMBP looks very similar to that of REM

(Chapter 6). Is this coincidental or does this tell us anything about other aspects of their analogy? In particular, we see that they are obviously different from the algorithmic point of view: while REM represents total "chaos" with independent costs, the sMBP by definition has a certain amount of cost dependence and thus intuitively should allow a more efficient optimum searching than REM. Is this so, and if yes — can this line of reasoning be further developed?

To understand that, one needs to properly estimate higher moments of the log-partition function, which is highly non-trivial, but for sure a noble goal.

Index

N

O

P

Q

R

S

T

U

Bibliography

Aigner, M., Ziegler, G.M., 2010. Cayley's formula for the number of trees. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 201–206.

Aizenman, M., Lebowitz, J.L., Ruelle, D., 1987. Some rigorous results on the Sherrington-Kirkpatrick spin glass model. Communications in Mathematical Physics 112, 3–20.

Alabdulmohsin, I., 2015. Algorithmic stability and uniform generalization, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, pp. 19–27.

Aron, I.D., Hentenryck, P.V., 2004. On the complexity of the robust spanning tree problem with interval data. Oper. Res. Lett. 32, 36–40. URL: https://doi.org/10.1016/S0167-6377(03)00058-0, doi:10.1016/S0167-6377(03)00058-0.

Auffinger, A., Chen, W.K., 2014. Free energy and complexity of spherical bipartite models. Journal of Statistical Physics 157, 40–59.

Bertsekas, D.P., Gallager, R.G., 1992. Data Networks, Second Edition. Prentice Hall.

Bian, Y., Gronskiy, A., Buhmann, J.M., 2016. Information-theoretic analysis of maxcut algorithms, in: 2016 Information Theory and Applications Workshop, (ITW), 2016, pp. 1–5.

Bilò, D., Gatto, M., Gualà, L., Proietti, G., Widmayer, P., 2009. Stability of networks in stretchable graphs, in: SIROCCO, pp. 100–112.

Bilu, Y., Linial, N., 2012. Are stable instances easy? Combinatorics, Probability & Computing 21, 643–660.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York.

Bousquet, O., Elisseeff, A., 2002. Stability and generalization. J. Mach. Learn. Res. 2, 499–526.

Bovier, A., 2012. Statistical Mechanics of Disordered Systems: A Mathematical Perspective. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press.

Bovier, A., Kurkova, I., Löwe, M., 2002. Fluctuations of the free energy in the rem and the p-spin sk models. The Annals of Probability 30, 605–651.

Buhmann, J., Gronskiy, A., Mihalák, M., Pröger, T., Šrámek, R., Widmayer, P., 2017a. Robust optimization in the presence of uncertainty: A generic approach. Journal of Computer and System Sciences .

Buhmann, J.M., 2010a. Information theoretic model validation for clustering, in: IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings, pp. 1398–1402.

Buhmann, J.M., 2010b. Information theoretic model validation for clustering, in: International Symposium on Information Theory (ISIT), Austin, TX, USA. pp. 1398–1402.

Buhmann, J.M., 2011. Context sensitive information: Model validation by information theory, in: Pattern Recognition - Third Mexican Conference, MCPR 2011, Cancun, Mexico, June 29 - July 2, 2011. Proceedings, pp. 12–21.

Buhmann, J.M., Dumazert, J., Gronskiy, A., Szpankowski, W., 2017b. Phase transitions in parameter rich optimization problems, in: Proceedings of the Fourteenth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2017, Barcelona, Spain, 2017., pp. 148–155.

Buhmann, J.M., Gronskiy, A., Szpankowski, W., 2014. Free energy rates for a class of very noisy optimization problems, in: Analysis of Algorithms (AofA), France. pp. 67–78.

Busetto, A.G., 2012. Information Theoretic Modeling of Dynamical Systems: Estimation and Experimental Design. Ph.D. thesis. ETH Zürich. (Diss. No. 20918).

Busse, L., Chehreghani, M., Buhmann, J.M., 2013. German Conference on Pattern Recognition. Springer Berlin Heidelberg, Berlin, Heidelberg. chapter Approximate Sorting. pp. 142–152.

Chehreghani, M.H., Busetto, A.G., Buhmann, J.M., 2012. Information theoretic model validation for spectral clustering, in: International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 495–503.

Cohen, J.E., 1988. Threshold phenomena in random structures. Discrete Applied Mathematics 19, 113–128.

Cover, T., Thomas, J., 2006. Elements of Information Theory. John Wiley and Sons.

Dembo, A., Zeitouni, O., 2009. Large deviations techniques and applications. volume 38. Springer Science & Business Media.

Derrida, B., 1981. Random-energy model: An exactly solvable model of disordered systems. Phys. Rev. B 24, 2613–2626.

Derrida, B., Gardner, E., 1986. Solution of the generalised random energy model. Journal of Physics C: Solid State Physics 19, 2253–2274.

Devroye, L., Wagner, T., 1979. Distribution-free performance bounds for potential function rules. IEEE Transactions on Information Theory 25, 601–604.

Feller, W., 1971. An Introduction to Probability Theory and Its Applications. volume 2. 2nd ed., Wiley, NY.

Frenk, J.B.G., van Houweninge, M., Kan, A.H.G.R., 1985. Asymptotic properties of the quadratic assignment problem. Mathematics of Operations Research 10, 100–116.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA.

Gatto, M., Widmayer, P., 2011. On robust online scheduling algorithms. J. Scheduling 14, 141–156.

Goerigk, M., Schöbel, A., 2016. Algorithm engineering in robust optimization, in: Algorithm Engineering - Selected Results and Surveys, pp. 245–279.

Gronskiy, A., Buhmann, J.M., 2014. How informative are minimum spanning tree algorithms?, in: 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, June 29 - July 4, 2014, pp. 2277–2281.

Harris, J., Hirst, J., Mossinghoff, M., 2008. Combinatorics and Graph Theory. Undergraduate Texts in Mathematics, Springer New York. URL: `https://books.google.ch/books?id=CxSoZcNymacC`.

Huber, M.L., 2012. Approximation algorithms for the normalizing constant of gibbs distributions. arXiv preprint arXiv:1206.2689 .

Jaynes, E.T., 1957a. Information theory and statistical mechanics. Physical Review 106, 620–630.

Jaynes, E.T., 1957b. Information theory and statistical mechanics II. Physical Review 108, 171–190.

Jaynes, E.T., 1982. On the rationale of maximum-entropy methods. Proceedings of the IEEE 70, 939–952.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.

Konishi, S., Kitagawa, G., 2007. Information Criteria and Statistical Modeling. Springer Publishing Company, Incorporated.

Kozina, G.L., Perepelitsa, V.A., 1994. Interval spanning trees problem: Solvability and computational complexity. Interval Computations , 42–50.

Lawler, E.L., 1963. The quadratic assignment problem. Management science 9, 586–599.

Łuczak, T., 1994. Phase transition phenomena in random discrete structures. Discrete Mathematics 136, 225 – 242.

Magner, A., Kihara, D., Szpankowski, W., 2015. A study of the Boltzmann sequence-structure channel. Technical report URL: `https://www.cs.purdue.edu/homes/spa/papers/boltzmann15.pdf`.

Magner, A., Kihara, D., Szpankowski, W., 2016. The boltzmann sequence-structure channel, in: IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016, pp. 255–259.

Merhav, N., 2010. Statistical physics and information theory. Foundations and Trends in Communications and Information Theory 6, 1–212.

Mezard, M., Montanari, A., 2009. Information, Physics, and Computation. Oxford University Press, Inc., New York, NY, USA.

Mézard, M., Montanari, A., 2009. Information, Physics, and Computation. Oxford University Press.

Mézard, M., Parisi, G., 2003. The cavity method at zero temperature. Journal of Statistical Physics 111, 1–34.

Mezard, M., Parisi, G., Virasoro, M.A., 1987. Spin Glass Theory and Beyond. World Scientific, Singapore.

Mihalák, M., Schöngens, M., Šrámek, R., Widmayer, P., 2011. On the complexity of the metric TSP under stability considerations, in: SOFSEM, pp. 382–393.

Parisi, G., 2009. The mean field theory of spin glasses: The heuristic replica approach and recent rigorous results. Letters in Mathematical Physics 88, 255–269.

Prim, R.C., 1957. Shortest connection networks and some generalizations. Bell System Technology Journal 36, 1389–1401.

Pröger, T., 2016. Robust Routing in Urban Public Transportation Networks. Ph.D. thesis. ETH Zürich. (Diss. No. 23513).

Raginsky, M., Sason, I., 2015. Concentration of measure inequalities and their communication and information-theoretic applications. CoRR URL: `http://arxiv.org/abs/1510.02947`.

Rissanen, J., 1978. Paper: Modeling by shortest data description. Automatica 14, 465–471.

Russo, D., Zou, J., 2015. How much does your data exploration overfit? Controlling bias via information usage. ArXiv e-prints URL: `https://arxiv.org/abs/1511.05219`, `arXiv:1511.05219`.

Sandoval, L., 2012. Pruning a minimum spanning tree. Physica A Statistical Mechanics and its Applications 391, 2678–2711.

Savage, I.R., 1962. Mills' ratio for multivariate normal distributions. J. Res. Nat. Bur. Standards Sect. B 66, 93–96.

Shalev-Shwartz, S., Ben-David, S., 2014. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, New York, NY, USA.

Shannon, C.E., 1948. A mathematical theory of communication. Bell system technical journal 27.

Shannon, C.E., Weaver, W., 1963. A Mathematical Theory of Communication. University of Illinois Press, Champaign, IL, USA.

Sherrington, D., Kirkpatrick, S., 1975. Solvable Model of a Spin-Glass. Physical Review Letters 35, 1792–1796.

Sherrington, D., Kirkpatrick, S., 1975. Solvable model of a spin glass. J. Physique Lett. 45, 1145–1153.

Shiryaev, A.N., 1995. Probability (2Nd Ed.). Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Szpankowski, W., 1995. Combinatorial optimization problems for which almost every algorithm is asymptotically optimal. Optimization 33, 359–368.

Szpankowski, W., 2001. Average Case Analysis of Algorithms on Sequences. John Wiley & Sons, Inc., New York, NY, USA.

Talagrand, M., 2003. Spin Glasses: A Challenge for Mathematicians: Cavity and Mean Field Models. Springer Verlag.

Tishby, N., Pereira, F.C., Bialek, W., 1999. The information bottleneck method, in: Proc. 37^{th} Allerton Conf. on Communication and Computation, pp. 368–377.

Vannimenus, J., Mézard, M., 1984. On the statistical mechanics of optimization problems of the travelling salesman type. Phys. Rev. Lett. 35, 1792–1796.

Vapnik, V., 1982. Estimation of Dependences Based on Empirical Data. Springer.

Vapnik, V.N., Chervonenkis, A.Y., 1971. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications 16, 264–280.

Šrámek, R., 2013. Uncertain Optimization Using Approximation Sets. An Algorithmic Prospective. Ph.D. thesis. ETH Zürich. (Diss. No. 21432).

Xu, A., Raginsky, M., 2017. Information-theoretic analysis of generalization capability of learning algorithms, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pp. 2521–2530.

Yaman, E., Karasan, O.E., Nar, M.C.P., 2001. The robust spanning tree problem with interval data. Operations Research Letters 29, 2001.

Curriculum Vitae

Alexey Gronskiy

General

Date and place of birth	16 November 1989 Moscow, Russia
Citizenship	Russian Federation
Civil status	Married

Education

2012–2018	*Doctoral studies* Information Science and Engineering Group, D-INFK ETH Zürich Switzerland
2006–2011	*Specialist degree in Mathematics* (with honours) Department of Mechanics and Mathematics Lomonosov Moscow State University (MSU) Russia
2004–2006	*High School* (honours gold medal) City School No. 179 Moscow Russia