

Optimization under Uncertainty with Applications to Multi-Agent Coordination

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Daniel Lyons

aus Bielefeld

Tag der mündlichen Prüfung: 31.01.2013

Erster Gutachter: Prof. Dr.-Ing. Uwe D. Hanebeck

Zweiter Gutachter: Prof. Dr. Giuseppe C. Calafiore

Optimization under Uncertainty with Applications to Multi-Agent Coordination

by
Daniel Lyons

Contents

| | |
|--|-------------|
| Zusammenfassung | v |
| Abstract | ix |
| Acknowledgment | xiii |
| Notation | xv |
| 1 Motion Planning with Chance Constraints | 1 |
| 1.1 Introduction | 1 |
| 1.2 Related Work | 6 |
| 1.3 Notation and Conventions | 8 |
| 1.4 General Problem Formulation | 9 |
| 1.5 Preliminaries | 12 |
| 1.5.1 Inter-Agent Collision Probability | 12 |
| 1.5.2 LTI Systems | 17 |
| 1.6 Sample Average Approximation | 23 |
| 1.6.1 Approximation of Collision Probabilities | 24 |
| 1.6.2 Convergence Properties | 25 |
| 1.6.3 Constraint Formulation | 29 |
| 1.7 Efficient Collision Avoidance in 2D | 33 |
| 1.7.1 Confidence Regions | 35 |
| 1.7.2 Collision Avoidance | 39 |
| 1.7.3 Constraint Formulation | 45 |
| 1.7.4 Higher Dimensional State Spaces | 47 |
| 1.8 Efficient Empirical Collision Avoidance | 57 |

| | | |
|----------|---|------------|
| 1.8.1 | Determination of eRPPs | 58 |
| 1.8.2 | Convergence Properties of eRPPs | 61 |
| 1.8.3 | Collision Avoidance | 63 |
| 1.9 | Results | 69 |
| 1.9.1 | Model Parameters | 69 |
| 1.9.2 | Example Scenarios | 72 |
| 1.9.3 | Quantitative Results | 80 |
| 1.10 | Conclusions | 88 |
| 2 | Mixed-Integer Random Convex Programs | 93 |
| 2.1 | Introduction to MI-RCPs | 93 |
| 2.2 | Preliminary Results on Mixed-Integer Problems | 96 |
| 2.2.1 | Support Constraints Integer Lattice | 97 |
| 2.2.2 | Support Constraints Mixed-Integer Space | 99 |
| 2.2.3 | Examples | 103 |
| 2.2.4 | Additional Definitions and Facts | 105 |
| 2.3 | Mixed-Integer Random Convex Programs | 109 |
| 2.3.1 | Main Result on MI-RCPs | 111 |
| 2.3.2 | Discussion of the Results | 116 |
| 2.4 | Mixed-Integer Random Linear Programs | 116 |
| 2.4.1 | Vertices of RLPs | 118 |
| 2.4.2 | Inner and Integer Points | 128 |
| 2.5 | Linear Programs with Random Right Hand Sides | 134 |
| 2.5.1 | Comparison to Robust Optimization | 136 |
| 2.6 | Results | 138 |
| 2.6.1 | Perturbed Integer Knapsack | 138 |
| 2.6.2 | Data-Driven Inventory Control | 142 |
| 2.6.3 | Multi-Agent Coordination | 145 |
| 2.7 | Conclusions | 150 |
| 3 | Distributed Optimization under Uncertainty | 153 |
| 3.1 | Introduction | 153 |

| | | |
|----------|--|------------|
| 3.2 | Problem Formulation | 157 |
| 3.2.1 | Alternating Optimization for ND-POMDPs | 164 |
| 3.3 | Efficient Alternating Optimization | 165 |
| 3.3.1 | Maximum Likelihood Observation (MLO) | 166 |
| 3.3.2 | The Local POMDP | 169 |
| 3.3.3 | Convergence | 171 |
| 3.4 | Algorithms | 174 |
| 3.5 | Extension to Continuous State Spaces | 177 |
| 3.5.1 | MLO in Kalman Filters | 181 |
| 3.5.2 | MLO in Particle Filters | 182 |
| 3.6 | Discussion and Related Work | 184 |
| 3.7 | Results | 187 |
| 3.7.1 | Sensor Network Target Tracking | 187 |
| 3.7.2 | Multi-Robot Coordination | 194 |
| 3.8 | Conclusions | 204 |
| 4 | Conclusions | 207 |
| | Appendices | 212 |
| A | Appendix | 213 |
| A.1 | General Convergence of the SAA | 214 |

Zusammenfassung

Motivation

Immer wenn Optimierungsverfahren eingesetzt werden um Lösungen für reale Probleme zu finden, ist es sehr wahrscheinlich, dass die vorliegenden Daten und Modelle unsicher oder gestört sind. Um die gefundenen Lösungen dennoch *ex ante* robust gegen Unsicherheiten zu machen, ist es notwendig, die Unsicherheiten bei der Optimierung bereits zu berücksichtigen. In dieser Arbeit werden daher verschiedene Optimierungsverfahren mit Unsicherheiten betrachtet und auf die Koordination in Multiagentensystemen angewendet. Diese Anwendung hat die Eigenschaft, dass sie hochgradig nicht-konvex sein kann. Beispielsweise ist der Fall, wenn zwei Agenten gleichzeitig eine Ressource nutzen wollen, auf die aber jeweils nur einer der Agenten zugreifen kann, eine „entweder-oder“ Entscheidung. So eine Entscheidung kann durch eine diskrete Entscheidungsvariable modelliert werden, die den Wert „0“ annimmt, wenn der eine Agent auf die Ressource zugreifen darf und den Wert „1“, wenn der andere Agent Zugriff bekommt. Diese Eigenschaft ist der Grund dafür, dass in großen Teilen der vorliegenden Arbeit Optimierungsprobleme betrachtet werden, die sowohl kontinuierliche als auch diskrete Entscheidungsvariablen haben.

Optimale Bewegungsplanung unter Unsicherheit

Als erstes wird die *optimale Bewegungsplanung* in einem Multiagentensystem mit unsicheren Zuständen der Agenten und Nebenbedingungen an die Wahrscheinlichkeit einer Kollision von Agenten mit Hindernissen oder anderen Agenten betrachtet. Die stochastischen Unsicherheiten können eine

unsichere Lokalisierung der Agenten oder exogene Störungen modellieren, die auf die Bewegung der Agenten wirken. Die Nebenbedingungen an Kollisionswahrscheinlichkeiten sind in der Regel nicht geschlossen lösbar, da für ihre Berechnung in allen außer den einfachsten Fällen multivariate Integrale über nicht-konvexe Gebiete bestimmt werden müssen. Um das Bewegungsplanungsproblem (BPP) dennoch numerisch handhabbar zu machen, wird zunächst eine Approximation der Zufallsgrößen durch Stichproben und die daraus resultierende empirische Approximation der stochastischen Nebenbedingungen betrachtet. Der signifikante Nachteil dieser Approximation ist der aus ihr resultierende sehr große Anteil an diskreten Entscheidungsvariablen. Dieser kann zu hohen Laufzeiten führen. Es wird deshalb eine effizientere Approximation basierend auf der Idee von Regionen mit erhöhter Aufenthaltswahrscheinlichkeit betrachtet: dies sind Regionen, für die sichergestellt ist, dass die Wahrscheinlichkeit, dass ein Agent sich außerhalb der Region aufhält, kleiner ist als eine vorgegebene Schranke. Das Optimierungsproblem wird dann so formuliert, dass die Regionen disjunkt von den jeweiligen Regionen der anderen Agenten sind.

Konvexe Programme mit Zufälligen Nebenbedingungen

Im zweiten Teil der Arbeit werden sogenannte *Random Convex Programs* (RCPs) betrachtet. Dies sind konvexe Optimierungsprobleme mit diskreten und kontinuierlichen Entscheidungsvariablen und einer festen Anzahl an zufällig gezogenen Nebenbedingungen. Die Realisierungen der Nebenbedingungen können zum Beispiel wie im ersten Teil der Arbeit aus Stichproben oder aus gegebenen historischen Daten stammen. Da die Nebenbedingungen zufällige Realisierungen sind, ist die optimale Lösung eines RCPs eine Zufallsvariable, die von den gezogenen Realisierungen abhängt. Wir beweisen für RCPs mit diskreten und kontinuierlichen Entscheidungsvariablen, dass ihre optimale Lösung mit hoher Wahrscheinlichkeit immer noch optimal sein wird für die nächste zufällige Realisierung der Neben-

bedingungen. Dies ist eine entscheidende Eigenschaft von RCPs, da sie garantiert, dass die optimale Lösung mit hoher Wahrscheinlichkeit auf zukünftige, zum Zeitpunkt der Optimierung noch unbekannte, Realisierungen der Nebenbedingungen *generalisiert*. Neben anderen klassischen Problemen der Entscheidungstheorie, wenden wir die in diesem Kapitel gewonnenen Erkenntnisse auf das BPP in einem Multiagentensystem an. Im ersten Teil der Arbeit wurden unter anderem Aussagen über die Zulässigkeit von optimalen Lösungen im Grenzwert gemacht, also wenn die Anzahl der Realisierungen der Zufallsvariablen gegen unendlich geht. Im Unterschied dazu liegt der Schwerpunkt der theoretischen Untersuchungen in diesem Teil der Arbeit auf den Eigenschaften von optimalen Lösungen, die unter einer endlichen Anzahl von Realisierungen gefunden wurden.

Verteilte Koordination unter Unsicherheit

Vorteilhafter als zentralisierte Architekturen sind in Multiagentensystemen oft Algorithmen zur *verteilten Entscheidungsfindung*. Deshalb wird im dritten Teil der Arbeit ein verteilter Algorithmus zur Koordination in Multiagentensystemen vorgeschlagen. Der Algorithmus nutzt das Modell der dezentralen partiell beobachtbaren Markov Entscheidungsprozesse (Dec-POMDPs). Das grundlegende Merkmal in diesem Modell ist, dass bereits während der Optimierung berücksichtigt wird, dass in der Zukunft neue unsicherheitsbehaftete Informationen (z.B. in der Form von Sensormessungen) über die Zustände der Agenten vorliegen werden. Gesucht werden dann optimale Strategien, die abhängig von zukünftigen, zum Optimierungszeitpunkt noch unbekanntem Informationen, die jeweils beste zugehörige Handlung bestimmen. Da es nachweislich NEXP komplex ist, optimale Lösungen für Dec-POMDPs zu finden selbst wenn nur zwei Agenten betrachtet werden, wird ein suboptimaler Algorithmus vorgeschlagen. Hier tauschen die Agenten alternierend vereinfachte Strategievorschläge aus und bestimmen, basierend auf den gegenwärtigen Vorschlägen der an-

deren Agenten, eine beste Antwort auf diese. Es werden ein Beweis über die endliche Konvergenz und die Komplexitätsreduktion gegenüber dem gegenwärtigen Stand der Technik erbracht und die empirische Auswertung in Simulationen in einem BPP und einem Sensor-Netzwerk Szenario.

Abstract

Motivation

Whenever optimization techniques are employed to find solutions to real world problems it is very likely that the data and models at hand are uncertain or disturbed. In order to make the solutions robust against these uncertainties, and to assure they do not violate unanticipated constraints, the uncertainties have to be accounted for in the optimization process. In this thesis different methods for optimization under uncertainty are considered and applied to multi-agent coordination. This application has the particular property that coordination problems can be highly non-convex. Consider for example the case when two agents plan to simultaneously utilize an indivisible resource. The decision on which agent may use the resource becomes an “either-or” decision which is usually modeled through a discrete decision variable that is equal to zero if one agent is granted access to the resource and equals one if the other agents gets access. This special property is the motivation why in this thesis there is a strong emphasis on optimization problems that have continuous as well as discrete decision variables.

Optimal Motion Planning Under Uncertainty

First, we consider *optimal motion planning* in multi-agent systems in which the states of the agents are uncertain and there are constraints on the probability of a collision of agents with obstacles or with other agents. The uncertainty in this scenario can be used to model uncertain localization or exogenous disturbances that interfere with the motion of the agents. In

general for the constraints on the collision probabilities there will be no closed form solution since in all but the simplest cases the probabilities are given by integrals of multivariate state distributions over complex non-convex regions. In order to make the motion planning problem numerically tractable we first consider an empirical approximation of the probabilistic collision avoidance constraints obtained from samples drawn from the random variables modelling the uncertain states of the agents. The fundamental drawback of this constraint approximation is that it leads to a high ratio of discrete decision variables and, hence, possibly high run times. Therefore a more efficient reformulation of the collision avoidance constraints based on regions of increased probability of presence is considered: these are regions for each agent for which it can be guaranteed that the probability that the agent is outside of its region is below a given threshold. The multi-agent coordination problem is then formulated in such a way that the regions are disjoint from the respective regions of other agents. The theoretical results in this chapter show that the resulting plans are feasible for the motion planning problem with constraints on the collision probabilities or will converge for increasing numbers of samples in the approximation.

Mixed-Integer Random Convex Programs

In the second part of this thesis *Mixed-Integer Random Convex Programs* (MI-RCPs) are considered. MI-RCPs are convex optimization problems with discrete and continuous decision variables and a fixed number of randomly drawn constraints. The realizations of the constraints could for instance stem from samples of a random variable like in the first part of the thesis, or from given historical data. Since the constraints are random, the optimal solution of an MI-RCP is a random variable depending on the realizations of the constraints. We prove for MI-RCPs that their optimal solution will remain optimal for the next random constraint realization with high probability. This is an essential property of MI-RCPs since it guar-

antees that their optimal solution will with high probability *generalize* to further yet unknown random constraint realizations. Besides classical decision theoretic problems we apply the results of this chapter to the motion planning problem in a multi-agent system. In the first part of this thesis, results on the feasibility and optimality of solutions with approximations of probabilistic constraints, when the number of samples used in these approximations goes to infinity are obtained. In contrast, the focus of the theoretical considerations in this section lies on the feasibility and optimality of solutions found under a *finite* number of samples.

Distributed Optimization Under Uncertainty

In general, algorithms for *distributed decision making* are more beneficial in multi-agent systems than centralized architectures. Therefore, in the third part of this thesis a distributed algorithm for coordination in a multi-agent system is proposed. The algorithm is based on the model of decentralized partially observable Markov decision processes (Dec-POMDPs). The signature property of this model is that the fact that updated information (e.g. in the form of sensor measurements) will be available at future time instances is explicitly accounted for during optimization. Since the search for optimal solutions of Dec-POMDPs is provably NEXP complex even if only two agents are considered, a suboptimal algorithm is proposed. In it the agents exchange simplified solution proposals in an alternating fashion and each determines its own best response to the current proposals of the other agents. A proof of the finite convergence of this algorithm as well as a proof of the complexity reduction are provided together with empirical evaluations of the algorithm in a multi-robot motion planning and a sensor network problem.

Acknowledgment

The research contained in this thesis was conducted over a period of four years at the ISAS lab in Karlsruhe and partly at the Politecnico in Turin. During the course of these four years many people supported me directly or indirectly and I want to express my gratitude to some of them in the following.

My research was conducted at the Intelligent Sensor-Actuator-Systems Lab at the Karlsruhe Institute of Technology within the Research Training Group 1194 “Self-Organizing Sensor-Actuator-Networks” funded by the German Research Foundation. I would like to thank my advisor Prof. Dr.-Ing. Uwe D. Hanebeck for the supervision of my thesis, his continued support, and for giving me the opportunity to work in his lab and do my PhD within the Research Training Group.

I want to express my gratitude to all my colleagues and all the staff at the ISAS lab. My special thanks go to colleagues and collaborators Marcus Baum, Frederik Beutler, Jan Calliess, Achim Hekler, Peter Krauthausen, Benjamin Noack, and Antonia Pérez Arias mostly for their kind support and fruitful discussions. I also would like to thank my colleagues from the research training group Jens Horneber, Chih-Ming Hsieh, Alexander Kettler, Christoph Roth, Dennis Schieferdecker, Johannes Schmid, and Markus Völker for our always interesting exchanges over lunch or at the group retreats.

At the ISAS lab I was also involved in teaching and very warm thanks go to my students Markus Kuderer, Matthias Nagel, Boris Groß-Hardt, Benedikt Schneyer, Erik Prause, and Sebastian Bittel whose theses I su-

pervised or who directly worked for me. I feel that from advising them I learned a lot.

As mentioned above, part of my research was conducted at the Politecnico di Torino in Turin and I am very grateful to prof. Giuseppe C. Calafiore for hosting me at Politecnico, for constantly throwing challenging problems at me, and for co-supervising my thesis. Furthermore, I would like to express my gratitude to fellow PhD students Luca Carlone, Nima Pour, and Stefano Rosa at Politecnico for their warm and friendly welcome.

Finally, I want to express my deep gratitude to my family and especially to my mother, my grandmother Ruth, and my sister Naomi for always truly believing in me, even when I was not. My deepest gratitude and warmest thanks go to my fiancée Kezban for calming me down when I was going crazy over my work, criticizing me when I needed frank words, motivating me when I needed motivation, and patiently enduring endless monologues on robots, math, the university system or science in general.

Karlsruhe, Fall 2013

Daniel Lyons

Notation

| | |
|---|---|
| x, y, u, a, b, \dots | Scalar or vector variable. |
| $x(i)$ | The i -th component of vector x . |
| \top | Transpose of a vector or a matrix. |
| $x, \delta, v, \mathbf{v}, \mathbf{w}, \omega$ | Random variable or random vector. |
| $\mathbf{A}, \mathbf{B}, \mathbf{V}, \Sigma, \Lambda \dots$ | Real-valued matrix. |
| N | Number of realizations of a random variable of random vector. |
| M | Number of agents. |
| $\gamma, \gamma^{i,j}, \gamma_i^{j,j}$ | Chance constraint bounds. |
| T | Length of planning horizon for model predictive control. |
| t | Index referencing considered time step. |
| $1 : T$ | Short for enumeration $1, 2, \dots, T$. |
| x^i | Superscript identifies the agent this variable refers to. |
| \vee | Logical “or”. |
| $\mathbb{E}_X\{g\}$ | Expectation of function g w.r.t. random vector X . |
| $\mathbb{1}_E$ | Indicator function of a set E . |
| \mathbb{P} | Probability measure on a space Ω with σ -algebra \mathcal{F} . |
| \mathbb{P}_N | Empirical probability measure on Ω . |
| \mathbb{P}^N | Product probability measure on the product space Ω^N with product σ -algebra. |

1. Motion Planning in Multi-Agent Systems with Chance Constraints

1.1. Introduction

In this chapter, we propose several approaches to formulate stochastic model predictive control for multi-agent systems (MA-MPC) with chance constraints on the probability of inter-agent collisions as a mixed-binary linear program (MBLP). Chance constraints are user-defined upper bounds on the probability that the uncertain states of the agents violate state constraints and their application in model predictive control has received great interest in the recent past. MBLPs are well-understood optimization problems that can be solved efficiently for moderate problem sizes and find many applications in robot planning, flight control, and receding horizon control (see for example [115, 132]).

Example 1.1: Air Traffic Control

Consider a controlled airspace at an airport in which an air traffic control (ATC) center is responsible for ensuring that the aircraft in the airspace are “separated”, i.e., have sufficient spatial and temporal clearance, and that the aircraft can land on an airstrip. See Figure 1.1 for a simple illustrative example.

In this example, both aircraft want to land on the airstrip and the task of the ATC is to find fuel optimal landing trajectories for both aircraft, depicted as dashed lines in Figure 1.1. In addition to constraints given by the motion model of the aircraft, further constraints are given through a no-fly zone that the aircraft have to avoid and the fact that a collision of

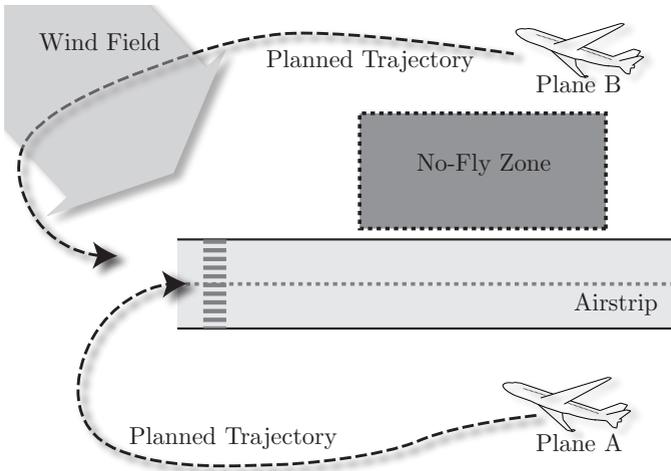


Figure 1.1.: Path Planning for Air Traffic Control.

both aircraft has to be avoided. Furthermore, wind disturbance, as depicted by the gray arrow at the top left in Figure 1.1, acts on the aircraft and, hence, the trajectories will be *disturbed* and it is not guaranteed that the aircraft will be able to exactly follow the planned trajectories.

An important question in this scenario is to find the right tradeoff between *short* and, hence, fuel optimal and *safe* trajectories. When the trajectories are in such a way that the aircraft pass each other with very little clearance, under strong wind gusts it can happen that a collision occurs. If, on the other hand, the trajectories are such that the aircraft have to make detours to keep a large separation, excessive fuel will be spent on the detour and delays in the schedule of the airport will occur. The algorithms presented in this chapter allow to solve this problem efficiently and to precisely control how cautious or risk-taking the trajectories will become. ■

There are many applications, in which the deployment of multiple aircraft, robots or UAVs is prevalent or can be advantageous for the comple-

tion of a mission as in the air traffic control example from above. Also, the search for a target with multiple UAVs [77] or localization of an odor source with multiple robots [73] have been studied in the past. Further applications that have been studied are cooperative load transportation with multiple UAVs [85], the cooperative inspection of airplane turbines with miniature robots [56] and simultaneous localization and mapping in a multi-robot system [50]. Another topic that received attention are driver assistance systems for cooperative cognitive vehicle systems [67]. For example, braking and evasion assistants in a vehicle autonomously make decisions, possibly after communicating with other vehicles, in order to avoid collisions in traffic situations. In all of these applications, path planning, obstacle avoidance, and collision avoidance with other members of the system also play an important role, since without path planning and collision avoidance, the successful completion of the task at hand and the physical intactness of the aircraft, robots or UAVs cannot be warranted. We will from now on not further distinguish between an aircraft, a robot, a UAV, or a ground vehicle and will therefore call the entity making path planning decisions an “agent”.

Noisy sensor measurements for localization and imprecise models of the motion dynamics can lead to uncertain estimates of the pose of the agents. Additionally the motion of the agents can be affected by exogenous disturbances like wind. When these uncertainties and disturbances are ignored by the path planning algorithm, a failure of the mission through collisions with obstacles or other agents is likely to occur and, hence, it is crucial to account for them while planning.

In order to account for uncertainties and disturbances in this chapter all system states of the agents will be modeled as random variables and the constraints on the states will be formulated as probabilistic chance constraints. Probabilistic *chance constraints* are constraints on the probability that the agents are in states that could cause a failure of the mission, i.e., a collision with obstacles or with each other [36, 107, 117]. Constraints on the probability that system states violate state bounds are in all but the simplest

cases given by integrals of multivariate state distributions over complex non-convex regions [107]. Particularly chance constraints on the probability of a collision of two agents, whose study is the focus of this chapter, lead to complex and non-convex constraints. For example two agents can not pass through the same location in space at the same time, so the decision which agent will be allowed to pass is a non-convex “either-or” decision. The main challenge in chance constrained control is therefore to make the chance constraints computationally tractable while keeping the assumptions on state distributions and state constraints as general as possible.

Another factor that increases safety and optimality of planning is planning model predictively. In *model predictive control* (MPC) the controller uses a model of the agents dynamics to extrapolate the agents’ states under candidate control sequences several time steps into the future. The controller then chooses the control sequence that is not only optimal for the current states but also for the extrapolated states. In general, the agent then applies only the first control in this sequence, possibly updates the estimate of its uncertain state and then replans an optimal control sequence based on the updated information. MPC enables the agents to plan pro-actively to avoid obstacles or other agents, since conflicts or possible collisions are detected earlier and the agents can react more quickly and more efficiently to avoid them. However, stochastic MPC has the drawback that the planning problem can become very complex and consequently it is vital to pay attention to real-time capabilities of planning.

In order to formulate an efficient algorithm for the MA-MPC problem with chance constraints, we first study a sample average approximation of the collision probabilities and use this approximation to formulate constraints for the stochastic control problem. We show that optimal controls found with the sample average approximation will converge against controls that are optimal for the MA-MPC problem without approximation of the chance constraints. However, the computational complexity of the resulting optimization problem is so high that this approach proves unsuitable

for the control of large multi-agent systems under real-time requirements.

To alleviate the computational burden we propose several approaches that all determine regions of increased probability of presence (RPP) for each agent, i.e., regions in state space in which the agent's true state will lie with a certain, prespecified probability. Then, we formulate constraints for the control problem that guarantee that these regions do not overlap for different agents. The main distinctive feature in these approaches is how the RPP is obtained: First we use probabilistic Chebychev inequalities to obtain these confidence regions and show for the resulting constraints that they are *conservative* for the original problem with probabilistic constraints. This means that every control feasible under these constraints will also satisfy the original chance constraints. This is a remarkable property of the construction, since it guarantees feasibility for the original problem while at the same time we do not have to evaluate the complex inter-agent collision chance constraints. The second approach is to find empirical RPPs for a sample approximation of the agents uncertain states, i.e., these are regions in state space in which the agent's state lies with a certain, prespecified empirical probability. We show that controls found with these empirical RPPs will converge against controls that are conservative for the original MA-MPC problem in the same sense as above, as the number of samples goes to infinity.

Since we employ a sample approximation of the agents' uncertain states and the probabilistic bounds we use for the RPP regions hold for arbitrary random variables, we do not have to make any assumptions (such as being Gaussian) on the nature of the occurring noise or disturbances. Also in our approach we do not have to assume that the chance constraints are given by linear inequalities only (as is frequently done in the literature) and thus, are able to model the more complex and inherently non-convex coupling constraints on the states of the agents.

To the best of our knowledge this is the first time a practical approach for the control of a multi-agent system with chance constraints on the prob-

ability of a collision of the agents is proposed. Existing work on chance constrained control either deals with planning for single agent systems or for systems without coupling chance constraints on joint states of different agents. The only other work that also considers coupling state constraints in a slightly different setting is our own work [8]. The sample average approximation of collision probabilities and the results on RPPs in two-dimensional state space were published in [11, 12]. Novel contributions are that we establish convergence of the sample average approximation, the extension of the RPP approach to collision avoidance in state spaces with arbitrary dimension and the results on empirical RPPs.

The chapter is structured as follows. In Section 1.4, we formally define the general problem of model predictive control for a multi-agent system with chance constraints. In Section 1.5 we introduce some preliminaries on the agents' linear and time invariant dynamics and sample-based chance constrained control for single-agent systems. In Section 1.6, we approximate the collision avoidance constraints directly with samples. In Section 1.7 and Section 1.8, we introduce the RPP approaches and prove their feasibility for the original problem. In Section 1.9, we study UAV path planning scenarios with non-Gaussian wind turbulence models. Section 1.10 concludes the chapter.

1.2. Related Work

The body of work on interactions of agents in multi-agent systems is immense for an introduction on how they are treated in artificial intelligence see [120] and the references therein. We focus here on recent results on chance constrained control and robust control of single and multi-agent systems.

Chance Constrained Optimization and Control

Chance constrained problems were introduced in [53] and have received interest in several different fields of research such as chemical process control [74, 90, 117], portfolio optimization [42, 103], call center staffing under uncertain demands [71], water management [63], control [55], and blending problems [103]. For more recent work on the theory of chance constrained programming see the monographs [36, 107, 119]. Convergence properties of sample average approximations in chance constrained programming are studied under varying assumptions in [20, 93, 103, 109, 112, 116].

Most recent work on chance-constrained MPC for linear systems can roughly be classified into three parts: conservative approximations [126], conservative control of systems with Gaussian disturbance [39, 41, 61, 101, 128], and sample-based techniques [38, 40].

The authors of [126] approximate chance constraints by *conservative constraints* stating that certain confidence ellipsoids are completely contained in the feasible region. Computational comparisons in [101] and [39] indicate that, albeit being fast, this approach introduces a high degree of sub-optimality through its conservativeness.

For systems where the disturbance distributions are assumed to be *Gaussian*, [39, 41, 101] propose that instead of enforcing that all state constraints are satisfied with a certain probability at the same time, each linear state constraint should be considered separately and the probability of violating this constraint be enforced separately. Utilizing a probabilistic inequality it can then be guaranteed that the controls found will be feasible for the chance constrained problem with the joint state constraints.

The authors of [102] extend these approaches to the control of multi-agent systems. However, without considering coupling constraints on the states of the agents that are necessary for modeling collision avoidance. Instead, the goal of that work is to propose an algorithm for the decentralized distribution of the overall amount of the chance constraint risk among the

agents.

In [40], the authors propose to approximate the probabilities for collisions with obstacles for the single agent case by *sample-based methods*. They transform the search for an optimal solution to the stochastic single agent control problem under chance constraints to finding a solution of a mixed-binary linear program (MBLP). This approximation has the advantage that almost arbitrary state and noise distributions can be treated. However, the considerations in [40] are restricted to chance-constrained control of a single agent.

Robust Optimization and Control

In robust approaches the aim is to find a solution that is optimal for “all” uncertainty realizations bound to lie in a compact and convex set. For references on general robust optimization and applications please see the introduction of Chapter 2 in Section 2.1 and Section 2.5.1.

Robust control has also been applied to single- and multi-agent path planning problems in several works [86–88, 110, 111]

1.3. Notation and Conventions

The probability that a multivariate random variable x with probability density function (pdf) $f(x)$ takes values in a set E is denoted by $\mathbb{P}\{x \in E\}$ and is the multivariate expectation

$$\mathbb{P}\{x \in E\} = \int_E f(x)dx = \int \mathbb{1}_E(x)f(x)dx, \quad (1.1)$$

where $\mathbb{1}_E$ is the indicator function of E , i.e., $\mathbb{1}_E(x)$ equals one if $x \in E$ and zero otherwise. $\text{Cov}(x)$ denotes the covariance $\mathbb{E}\{(x - \mathbb{E}\{x\})(x - \mathbb{E}\{x\})^\top\}$

of the random vector x , which is a matrix for multivariate random variables. Analogously

$$\text{Cov}(x, y) := \mathbb{E} \left\{ (x - \mathbb{E}\{x\})(y - \mathbb{E}\{y\})^\top \right\} \quad (1.2)$$

is the covariance between the random vectors x and y [104]. We define the Manhattan-norm as $\|x\|_1 := \sum_k |x_k|$, the Euclidean norm through $\|x\|_2 := \sqrt{\sum_k x_k^2}$ and the maximum norm as $\|x\| := \max_k |x_k|$ for real vectors $x = [x_1, x_2, \dots, x_n]^\top$.

1.4. General Problem Formulation

The general multi-agent model predictive control (MA-MPC) problem we want to solve is as follows: For M agents $i = 1, \dots, M$ with discrete-time stochastic system models in state space form, we plan over a horizon of length T in order to minimize the sum of the agents' expected cumulative cost functions. This minimization is subject to the probabilistic constraints that the probabilities of inter-agent collisions and the probability of agents leaving the feasible region are kept below certain user-defined thresholds. The formal formulation of this problem is

$$\text{MA-MPC : } \quad \underset{u_{1:T}^i, i=1, \dots, M}{\text{minimize}} \quad \sum_{i=1}^M \mathbb{E}_{x_{0:T}^i} \left\{ h^i(x_{0:T}^i, u_{1:T}^i) \right\} \quad (1.3)$$

$$\text{s.t. for all agents } i = 1, \dots, M : \quad (1.4)$$

$$u_{1:T}^i \in F_u^i \quad (1.5)$$

$$\forall_{t=1, \dots, T} x_t^i = f_t^i(x_0^i, u_{1:t}^i, v_{1:t}^i) \quad (1.6)$$

$$\mathbb{P} \left\{ x_{1:T}^i \notin F^i \right\} \leq \gamma^i \quad (1.7)$$

$$\forall_{j=1, \dots, M, j \neq i} \mathbb{P} \left\{ (x_{1:T}^i, x_{1:T}^j) \notin F^{i,j} \right\} \leq \gamma^{i,j} \quad (1.8)$$

$$\mathbb{E} \left\{ x_{1:T}^i \in G \right\} . \quad (1.9)$$

The decision variables $u_{1:T}^i = [(u_1^i)^\top, (u_2^i)^\top, \dots, (u_T^i)^\top]^\top$ are the control inputs to agent i each confined to lie in a compact, and convex polytope $F_u^i \subset \mathbb{R}^{d_{u,i}}$. The function h^i is the control objective and rates how desirable certain states of the agents are. In this chapter, we assume that each function h^i is piecewise linear and convex both in the controls $u_{1:T}^i$ as well as in the state trajectories $x_{1:T}^i$. It depends on the control inputs and the system states of the agents. The system state of agent i over the planning horizon is modeled as a random vector and is denoted by $x_{0:T}^i = [(x_0^i)^\top, (x_1^i)^\top, \dots, (x_T^i)^\top]^\top$. The notation x_t^i is an abbreviation in the sense that

$$x_t^i = f_t^i(x_0^i, u_{1:t}^i, v_{1:t}^i), \quad (1.10)$$

i.e., the random state of agent i at time step t depends on the control inputs $u_{1:t}^i$ up to t , the initial random state x_0^i , and the disturbances $v_{1:t}^i$ up to time step t . The functional relation between these quantities is given by the model of the dynamics f_t^i . We assume throughout the chapter that the dynamics are linear and time-invariant, see Section 1.5.2 for more details. The x_t^i are multivariate random vectors, hence they lie in some \mathbb{R}^{d_i} .

The stochastic disturbances v_t^i are used to account for possible errors in the dynamic model or exogenous disturbances that act upon the systems, such as wind turbulence on UAVs. They can also account for uncertainties in the initial state estimate x_0^i and how those are carried forward and possibly increased over time through state prediction with the model f_t^i .

We assume here that the second central moments of the disturbances v_t^i , i.e., the covariances, and the prior state distributions x_0^i are known. Even if there is no analytic knowledge about the second central moments, since we will assume in later sections that we can draw samples from these distributions, it should be possible to obtain good estimates of these quantities through the sample covariance [37].

We assume all random vectors to be stochastically independent for different agents $i_0 \neq i_1$ and we also assume that the system disturbance is

independent of the control inputs. Since the states are modeled as random vectors, we take the expectation of the objective $h^i(x_{0:T}^i, u_{1:T}^i)$ with respect to the agents' state distributions in (1.3).

F^i is the feasible region for the states of an agent i and $\mathbb{P}\{x_{1:T}^i \notin F^i\}$ is the probability that the states of agent i leave the feasible region. The feasible region can for example model an area the agents are not supposed to leave or obstacles the agents have to avoid.

$\mathbb{P}\{(x_{1:T}^i, x_{1:T}^j) \notin F^{i,j}\}$ specifies the probability that agent i and agent j do not meet the coupling state constraints defined by the feasible region $F^{i,j}$ that controls the interaction among agents. We understand these constraints as collision avoidance constraints, i.e., constraints on the distance of the states of agents i and j . For notational simplicity we assume that the coupling constraints are consistent for all agents, i.e., there is a feasible set F^c such that $F^c = F^{i,j}$ for all agent pairings. We will formally define F^c in the next section.

We assume that all probabilities involving the state random variables are understood as actually conditioned on a concrete control sequence, i.e. we write for example $\mathbb{P}\{x_{1:T}^i \in F^i\}$ as shorthand term for

$$\mathbb{P}\{x_{1:T}^i(x_0^i, u_{1:T}^i, v_{1:T}^i) \in F^i \mid u_{1:T}^i\} \quad (1.11)$$

the probability that the state trajectory

$$x_{1:T}^i(x_0^i, u_{1:T}^i, v_{1:T}^i) := [(x_0^i)^\top, (x_1^i)^\top, \dots, (x_T^i)^\top]^\top \quad (1.12)$$

$$(1.13)$$

is inside the feasible region *given* control inputs $u_{1:T}^i$. We will use the abbreviated notation throughout this chapter.

The upper bounds $0 \leq \gamma^i, \gamma^{i,j} \leq 1$ on the probability that the agents' states leave their own feasible region F^i or the joint feasible regions F^c characterize the chance constraints on the failure of planning [36, 107, 117]. These

chance constraint bounds are specified by the user and can be used to adjust how cautious or risk taking the agents' controls will become: If the chance constraint bounds are chosen "large", i.e., close to one, the agents will be allowed to carry out motion controls that have a high probability of them violating state constraints. If the bounds are small (e.g. $\gamma^i = \gamma^{i,j} = 10^{-4}$) then the probabilities of constraint violation will be enforced to be almost zero and the agents will plan very cautiously and conservatively.

The constraint $\mathbb{E} \{x_{1:T}^i \in G\}$ can be used to model a terminal constraint on the mean state trajectory for example for ensuring stability of the system.

1.5. Preliminaries

In this section we will define the probability of a collision of states of two agents, study linear time-invariant system dynamics and their impact on the random variables x_t^i .

1.5.1. Definition of Inter-Agent Collision Probability

In this section we define our notion of a collision of the states of two agents, the resulting collision avoidance coupling constraints on the states, and the probability of a collision.

Let $\varepsilon > 0$ be the prespecified minimum distance between the states of two agents for collision avoidance, for instance we could set ε to be twice the diameter of a robot or twice the wingspan of a fixed-wing UAV. Let x_t^1 and x_t^2 denote the d_1 -dimensional and d_2 -dimensional states of agents 1 and 2 in \mathbb{R}^{d_1} and \mathbb{R}^{d_2} respectively at a time step t . Since we formulated the coupling constraints as pairwise constraints on the states of two agents, in the following we will only consider two agents numbered by one and two.

Definition 1.1 (Collision of Two Agents' States) Given an $\varepsilon > 0$ and $t \in \{1, \dots, T\}$, we define the event of a *collision* as

$$\|\Pi^1(x_t^1) - \Pi^2(x_t^2)\| < \varepsilon, \quad (1.14)$$

i.e., the event that two agents' states are closer than the minimum clearance ε . Here,

$$\Pi^i: \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{\hat{d}} \quad (1.15)$$

projects the state variable $x_t^i \in \mathbb{R}^{d_i}$ of agent i onto a subspace $\mathbb{R}^{\hat{d}}$ in which the collision avoidance constraints are formulated.

For example in the simulations in Section 1.9 we will consider path planning for UAVs where the state of an UAV is modeled as six-dimensional variable but collision avoidance is formulated in three-dimensional space. For notational simplicity we will omit the mapping Π_i in the other sections and assume that it is inserted where necessary.

Definition 1.2 (Feasible Region for Joint States) The feasible region

$$F^c \subset \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \quad (1.16)$$

for joint states of two agents is the set of joint states that have a distance greater or equal than ε :

$$F^c := \{(x^1, x^2) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} : \|\Pi^1(x_t^1) - \Pi^2(x_t^2)\|_2 \geq \varepsilon\}. \quad (1.17)$$

Definition 1.3 (Probability of Collision) The probability of a collision of two agents' states is the probability that the uncertain states of the agents

are not in F^c

$$\begin{aligned}
 \mathbb{P} \{ (x_t^1, x_t^2) \notin F^c \} &= \mathbb{P} \{ \|\Pi^1(x_t^1) - \Pi^2(x_t^2)\|_2 < \varepsilon \} \\
 &= \mathbb{E} \{ \mathbb{1}_{CF^c}(\Pi^1(x_t^1), \Pi^2(x_t^2)) \} \\
 &\quad \left(= \int \int \mathbb{1}_{CF^c}(\Pi^1(x_t^1), \Pi^2(x_t^2)) g_{x_t^1}(x_t^1) g_{x_t^2}(x_t^2) dx_t^1 dx_t^2 \right),
 \end{aligned} \tag{1.18}$$

where

$$\mathbb{1}_{CF^c}(\Pi^1(x_t^1), \Pi^2(x_t^2)) = \begin{cases} 1, & \text{if } \|\Pi^1(x_t^1) - \Pi^2(x_t^2)\| < \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{1.19}$$

is the indicator function of the complement of F^c and $g_{x_t^1}(x_t^1)$ and $g_{x_t^2}(x_t^2)$ are the probability density functions of the state estimates of the agents if they exists, i.e. if the random variables x_t^1 and x_t^2 have a probability density function.

Since in practical applications, the uncertain initial states and the system disturbances will be given by some stochastic model, it is safe to assume that all random variables have a probability density function. However, the sample-based approaches for approximating collision probabilities and the conservative approximations presented in this chapter will not need this assumption so we will require it only when necessary. If the random variables have density functions, the integral (1.18) that determines the probability of a collision is the integral of multivariate density functions multiplied by the indicator function $\mathbb{1}_{CF^c}$. For general density functions this integral will have no closed-form solution because of the possibly complex structure of the densities. Even for multivariate Gaussian distributions it will become difficult to evaluate the integral since the indicator function is a non-convex and nonlinear function and there are no known closed form expressions to evaluate this integral. Even if the integral (1.18) was given by a closed-form representation, it is not guaranteed that the resulting constraints on

the probability of an inter-agent collision would be tractable for an optimization algorithm.

This renders the chance constraints numerically intractable if one wants to directly pose them as constraints to the MA-MPC problem: Given a set of joint controls for the agents, checking if these controls are feasible for the chance constraints would require the application of numerical quadrature methods in order to evaluate the multivariate expectation in 1.18. When trying to find optimal controls for the multi-agent systems a plethora of possible controls will have to be checked for feasibility in order to find an optimal solution. Due to these numerical difficulties we will in this chapter present approximations and conservative reformulations of the constraints on the probability of a collision of agents.

We conclude this section with two remarks.

Remark 1.1 Recall that the inter-agent collision avoidance constraints in Section 1.4 were formulated for state trajectories over the complete planning horizon T in the form

$$\forall_{i,j=1,\dots,M, j \neq i} \mathbb{P} \left\{ (x_{1:T}^i, x_{1:T}^j) \notin F^{i,j} \right\} \leq \gamma^{i,j} \quad (1.20)$$

and the definition we gave in this section is for one time step t only. However the generalization from the constraints on the states in one time step to the states over the whole planning horizon is straight forward: Define

$$F_{1:T}^c := \left\{ (x_{1:T}^1, x_{1:T}^2) \in \mathbb{R}^{Td_1} \times \mathbb{R}^{Td_2} : \|\Pi^1(x_t^1) - \Pi^2(x_t^2)\| \geq \varepsilon, t = 1, \dots, T \right\} \quad (1.21)$$

and the probability of a collision as

$$\begin{aligned}
 \mathbb{P}\{(x_{1:T}^1, x_{1:T}^2) \notin F_{1:T}^c\} &= \mathbb{P}\{\exists t_0 \in \{1, \dots, T\} : (x_{t_0}^1, x_{t_0}^2) \notin F^c\} \\
 &= \mathbb{P}\{(x_1^1, x_1^2) \notin F^c \vee (x_2^1, x_2^2) \notin F^c \vee \dots \vee (x_T^1, x_T^2) \notin F^c\} \\
 &= \mathbb{P}\left\{\bigcup_{t=1}^T \{(x_t^1, x_t^2) \notin F^c\}\right\}.
 \end{aligned} \tag{1.22}$$

So from Definition 1.2 the probability of a collision over the complete planning horizon can be derived. In this chapter we will adopt the more convenient notation that we write $(x_{1:T}^1, x_{1:T}^2) \in F^c$ but actually mean $(x_{1:T}^1, x_{1:T}^2) \in F_{1:T}^c$.

In the next remark we will demonstrate how the probability of a collision over the complete planning horizon, i.e., for all time steps $t = 1, \dots, T$, is related to the probabilities of collision in a single time step.

Remark 1.2 From Boole's inequality for finite unions [108]

$$\mathbb{P}\left\{\bigcup_i A_i\right\} \leq \sum_i \mathbb{P}\{A_i\} \tag{1.23}$$

we can deduce for (1.22)

$$\begin{aligned}
 \mathbb{P}\{(x_{1:T}^1, x_{1:T}^2) \notin F_{1:T}^c\} &= \mathbb{P}\left\{\bigcup_{t=1}^T \{(x_t^1, x_t^2) \notin F^c\}\right\} \\
 &\leq \sum_{t=1}^T \mathbb{P}\{(x_t^1, x_t^2) \notin F^c\}
 \end{aligned} \tag{1.24}$$

and, hence, if

$$\sum_{t=1}^T \mathbb{P}\{(x_t^1, x_t^2) \notin F^c\} \leq \gamma^{1,2} \tag{1.25}$$

then also $\mathbb{P}\{(x_{1:T}^1, x_{1:T}^2) \notin F_{1:T}^c\} \leq \gamma^{1.2}$. We will use this fact in later sections when we consider efficient conservative reformulations of the inter-agent collision avoidance chance constraint.

1.5.2. Systems with Linear Time-Invariant Dynamics

In this section we will first briefly introduce linear time-invariant system dynamics, describe how with these dynamics samples from the random variables x_t^i can be generated and reference prior work on MPC with chance constraints in linear time-invariant systems for a single agent.

For arbitrary system dynamics f_t^i the MA-MPC problem will be a non-linear and non-convex optimization problem and computationally demanding global optimization algorithms would have to be employed to solve it with local minima being prevalent [79]. Additionally the collision avoidance constraints are nonlinear and will be formulated as mixed-binary linear (MBL) constraints in the next sections so that MA-MPC problem would in fact become a nonlinear non-convex optimization problem with discrete and continuous decision variables.

Instead, we consider here the more simple case of linear system dynamics and piecewise linear and convex objective functions and, hence, the optimization problem that has to be solved to find an optimal solution to the MA-MPC problem is a mixed-integer linear problem (MILP). Still the theoretical complexity of MILPs is high, but there are powerful commercial solvers available for these problems that are very fast for moderate problem sizes.

We assume in this chapter that the stochastic discrete-time dynamic state space model of each agent $i = 1, \dots, M$ is given by the linear time-invariant system equation

$$x_{t+1}^i = \mathbf{A}^i x_t^i + \mathbf{B}^i u_t^i + v_t^i, \quad t = 1, \dots, T-1. \quad (1.26)$$

Here x_t^i is the uncertain state of agent i modeled as a random variable, u_t^i is the deterministic control input and v_t^i is the stochastic system disturbance and with time-invariant matrices $\mathbf{A}^i \in \mathbb{R}^{d_i \times d_i}$ and $\mathbf{B}^i \in \mathbb{R}^{d_i \times d_{u,i}}$. For the linear system dynamics (1.26) the mapping $x_t^i = f_t^i(x_0^i, u_{1:t}^i, v_{1:t}^i)$ that determines how the agent's state depends on the initial state x_0^i and behaves under control inputs $u_{1:t}^i$, is linear in the control inputs and can be stated as

$$x_{t+1}^i = (\mathbf{A}^i)^t x_0^i + \sum_{s=1}^t (\mathbf{A}^i)^{t-s-1} (\mathbf{B}^i u_s^i + v_s^i), \quad (1.27)$$

where

$$(\mathbf{A}^i)^t = \underbrace{\mathbf{A}^i \mathbf{A}^i \cdots \mathbf{A}^i}_{t \text{ times}}. \quad (1.28)$$

This can be checked by iteratively applying the dynamics (1.26) t times. In this case the mapping $x_t^i = f_t^i(x_0, u_{1:t}, v_{1:t})$ is linear in x_0 and $v_{1:t}$ and because a linear transformation of random variables is again a random variable, x_t^i is a random variable.

Sample-Based Techniques for Systems with Linear Time-Invariant Dynamics

We will use sample approximations throughout the rest of this chapter to approximate probabilities involving random variables describing the agents' state and therefore we will sketch how samples from the random variables x_t^i can be generated. For arbitrarily distributed initial states x_0^i and for arbitrarily distributed system disturbance v_t^i there is in general no closed-form representation (or one with a finite number of parameters) of the densities of the random vectors x_t^i . We will therefore make use of sample-based techniques for representing the uncertain states x_t^i .

We assume that for each agent $i = 1, \dots, M$ we can draw N independent and identically distributed (i.i.d.) samples of the random vector x_0^i at time

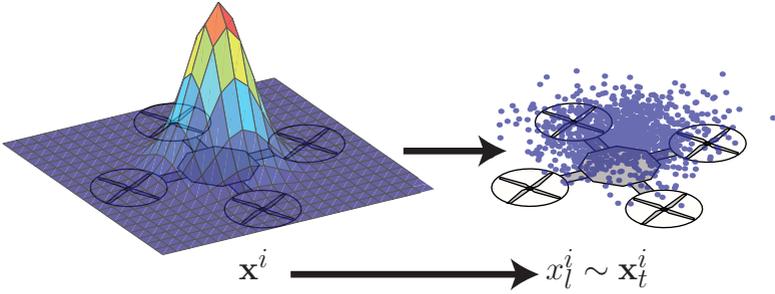


Figure 1.2.: Samples are drawn from the random variable representing the state of agent j , e.g., the position estimate of an UAV.

step $t = 0$ (see Figure 1.2 for an illustrative depiction of the samples). We will denote these samples by $\{x_{0j}^i\}_{j=1}^N$ and assume, for notational convenience only, that we draw an equal number N of realizations from each agent's initial random vector x_0^i . Furthermore for each agent i we draw N i.i.d. disturbance samples $(v_{1j}^i, v_{2j}^i, \dots, v_{Tj}^i)$, $j = 1, \dots, N$ from the random system disturbance $v_{1:T}^i$ that affects agent i over the planning horizon of length T .

The model of the dynamics (1.26) then allows us to generate N “sample trajectories” over time. These trajectories are obtained by propagating each initial sample in combination with a joint disturbance sample through the system equation (1.27). Each sample trajectory is of length $T + 1$ and is of the form

$$x_{0:T,j}^i := [(x_{0j}^i)^\top, (x_{1j}^i)^\top, \dots, (x_{Tj}^i)^\top]^\top, \quad (1.29)$$

where for x_{tj}^i it holds that

$$x_{t+1,j}^i = (\mathbf{A}^i)^t x_{0j}^i + \sum_{s=1}^t (\mathbf{A}^i)^{t-s-1} (\mathbf{B}u_s^i + v_{sj}^i). \quad (1.30)$$

The most important property of this construction is that the sample tra-

jectories depend deterministically and linearly on the control inputs. This will enable us to formulate a sample-based approximation of the MA-MPC problem as a deterministic optimization problem.

What else have we achieved by this construction? For one we have obtained a representation of agent i 's uncertain states for instance at time step t the samples $\{x_{tj}^i\}_{j=1}^N$ are samples from the random variable x_t^i . These samples can be used to evaluate probabilistic quantities depending on the agents' random states that otherwise we would have had to employ computationally expensive methods to calculate. One example of such a probabilistic quantity is the expectation $\mathbb{E}_{x_t^i}\{g\}$ of a function g with respect to x_t^i . It can be approximated through

$$\mathbb{E}_{x_t^i}\{g\} \approx \frac{1}{N} \sum_{j=1}^N g(x_{tj}^i) \quad (1.31)$$

and the expectation over an event E through

$$\mathbb{E}_{x_t^i}\{E\} \approx \frac{1}{N} \sum_{j=1}^N \mathbb{1}_E(x_{tj}^i) \quad (1.32)$$

both with convergence to the true expectation as the number of samples goes to infinity [104].

With this construction in mind, we will from now on assume that we can always draw samples from the random variables x_t^i .

Sample-Based Control for a Single Agent

For the single agent chance constrained model predictive control problem (SA-MPC)

$$\text{SA-MPC : } \underset{u_{1:T}}{\text{minimize}} \mathbb{E}_{x_{0:T}} \{h(x_{0:T}, u_{1:T})\} \quad (1.33)$$

$$\text{s.t. } u_{1:T} \in F_u \quad (1.34)$$

$$\forall_{t=1, \dots, T} x_t = f_t(x_0, u_{1:t}, v_{1:t}) \quad (1.35)$$

$$\mathbb{P} \{x_{1:T} \notin F\} \leq \gamma \quad (1.36)$$

$$\mathbb{E} \{x_{1:T} \in G\} . \quad (1.37)$$

for one agent and *without* coupling constraints on joint states of agents, the work [39] proposes an approach to formulate the problem as MBL optimization problem. The assumptions made in that work are that

- the objective function is piecewise linear and convex,
- the feasible region F^u for the controls is a convex polytope,
- the system model f_t is linear,
- the feasible region F for the system states is either a convex polytope or a non-convex polytope with convex polytopal obstacles.

In that work realizations of the state trajectory $x_{1:T}$ are drawn as demonstrated in the previous section and the probability $\mathbb{P} \{x_{1:T} \notin F\}$ is approximated by the empirical probability computed from these samples. The resulting sample average approximation constraints that ensure that the empirical probability $\mathbb{P}_N \{x_{1:T} \notin F\} \leq \gamma$ are then formulated as MBL constraints.

The focus of this chapter lies on efficient modeling of constraints on inter-agent collision probabilities so we take the formulation of the single agent planning problem with the above assumptions for granted and assume

that it is modeled as in [39] as a MBL problem. This is the reason why it is often convenient to consider a reduced version of the MA-MPC problem of the form

$$\begin{aligned}
 \text{MA-MPC : } \quad & \underset{u_{1:T}^i, i=1, \dots, M}{\text{minimize}} \quad \sum_{i=1}^M h^i(u_{1:T}^i) \\
 & \text{s.t. for all agents } i = 1, \dots, M : \\
 & \quad u_{1:T}^i \in F_u^i \\
 & \quad \forall_{t=1, \dots, T} x_t^i = f_t^i(x_0^i, u_{1:t}^i, v_{1:t}^i) \\
 & \quad \forall_{j=1, \dots, M, j \neq i} \mathbb{P} \left\{ (x_{1:T}^i, x_{1:T}^j) \notin F^{i,j} \right\} \leq \gamma^{i,j},
 \end{aligned} \tag{1.38}$$

with only the inter-agent collision avoidance constraints. When considering the reduced MA-MPC problem, we assume that the chance constraints $\mathbb{P} \{x_{1:T}^i \notin F\} \leq \gamma$ and the constraints $\mathbb{E} \{x_{1:T}^i \in G\}$ are already handled by methods as those introduced in [39] and, hence, are included in the deterministic constraint $u_{1:T}^i \in F_u^i$.

The Mean and Covariance in Linear Time-Invariant Systems

We assume without loss of generality that the system disturbance v_t^i for all agents and for all time steps is zero-mean¹. For system disturbance with vanishing mean the mean μ_t^i of agent i 's random state follows the recursive rule

$$\mu_t^i = \mathbf{A}^i \mu_{t-1}^i + \mathbf{B} u_{t-1}^i, \tag{1.39}$$

¹If the disturbance had a non-zero mean, we could subtract this mean from the system equations as deterministic disturbance and would have reduced this disturbance to zero-mean disturbance again.

as one can easily check from (1.26) by taking the expectation. It follows that the mean is determined through

$$\mu_t^i = (\mathbf{A}^i)^t \mu_0^i + \sum_{s=1}^t (\mathbf{A}^i)^{t-s-1} \mathbf{B}^i u_s^i \quad (1.40)$$

for each agent i analogously to Equation (1.27) and always depends *linearly* on the controls.

The covariances of the random vectors x_t^i do not depend on any control inputs, but only on the covariances of the prior distribution x_0^i and the disturbances $v_{1:t}^i$. The recursive formula for the evolution of the covariances is

$$\begin{aligned} \text{Cov}(x_t^i) &= \mathbf{A}^i \text{Cov}(x_{t-1}^i) (\mathbf{A}^i)^\top + \text{Cov}(v_{t-1}^i) \\ &+ \mathbf{A}^i \text{Cov}(x_{t-1}^i, v_{t-1}^i) + \text{Cov}(v_{t-1}^i, x_{t-1}^i) (\mathbf{A}^i)^\top. \end{aligned} \quad (1.41)$$

This property can be derived from basic matrix manipulations and covariance matrix properties. In the general problem formulation of MA-MPC, we assumed that covariances of the prior distributions x_0^i and the disturbances v_t^i are known in advance, so the agents can recursively compute the covariance of their random state at time step t .

1.6. Sample Average Approximation of Inter-Agent Collision Probability

This section is concerned with the approximation of the probability of an inter-agent collision and the derivation of constraints that keep these probabilities below user-defined thresholds. We use the sample approximations of the agents' uncertain states to derive a sample approximation of the probability of an inter-agent collision. This direct approximation converges to the true probability of a collision as the number of samples goes to infinity. However, the complexity of the resulting optimization problem can be

so high that this approach proves problematic for control under real-time requirements.

1.6.1. Approximation of Collision Probabilities

In order to render the probability of an inter-agent collision computationally tractable, we replace it by the sample average approximation (SAA) [75, 93, 103]. The general idea in SAA of chance constraints is to replace the probability measure \mathbb{P} by the empirical measure constructed from i.i.d. samples from the random vectors describing the states of the agents. More precisely, let $\{x_j^1\}_{j=1}^N$ and $\{x_l^2\}_{l=1}^N$ be N i.i.d. samples from the independent random vectors $x_{1:T}^1$ and $x_{1:T}^2$ that represent the state trajectories of agents 1 and 2, derived as highlighted in Section 1.5.2. Then a joint sample from the random vector $(x_{1:T}^1, x_{1:T}^2)$ is given by (x_j^1, x_l^2) resulting in N^2 joint samples. We replace the measure \mathbb{P} by the empirical measure

$$\mathbb{P}_N := \frac{1}{N^2} \sum_{j=1}^N \sum_{l=1}^N \delta_{x_j^1, x_l^2}, \quad (1.42)$$

where $\delta_{x_j^1, x_l^2}$ denotes the discrete measure with mass one at (x_j^1, x_l^2) . This is a discrete measure assigning the mass $\frac{1}{N^2}$ to each point (x_j^1, x_l^2) .

Definition 1.4 (SAA of the Inter-agent Collision Probability) The sample average approximation of the chance constraint on the probability of a collision is

$$\mathbb{P}_N \{(x_{1:T}^1, x_{1:T}^2) \notin F^c\} = \mathbb{E}_{\mathbb{P}_N} \{\mathbb{1}_{CF^c}\} = \frac{1}{N^2} \sum_{j=1}^N \sum_{l=1}^N \mathbb{1}_{CF^c}(x_j^1, x_l^2) \leq \gamma^{1,2}, \quad (1.43)$$

where $\mathbb{1}_{CF^c}$ is the indicator function of the complement of F^c as defined in Section 1.5.1.

In problem MA-MPC constraint $\mathbb{P} \left\{ (x_{1:T}^i, x_{1:T}^j) \notin F^{i,j} \right\} \leq \gamma^{i,j}$ will be replaced by constraint (1.43).

Remark 1.3 In Section 1.6.3 we will demonstrate how the SAA constraints can be formulated as mixed-binary linear constraints. In Definition 1.4 the indicator function of the complement of F^c has to be evaluated N^2 times and each of these evaluations requires a fixed number of binary constraints in the MBL formulation as we will see in Section 1.6.3. Since one usually desires to employ as many samples as possible to guarantee good approximation properties and since binary variables have a crucial impact on the complexity and thus the run-time of the program, a quadratic dependence of the number of binaries on the number of samples renders this formulation problematic under real-time requirements.

1.6.2. Convergence Properties

In this section we will consider convergence properties of plans found with the approximated chance constraint. Since in this section we are only concerned with inter-agent collision probabilities, we assume that we consider only two agents and we assume that the other chance constraints on the agents' states are given through deterministic constraints, i.e. the inter-agent collision constraints are the only probabilistic constraints. We will show convergence in the following simplified program

$$\begin{aligned}
 \text{2MA-MPC : } \quad & \min_{u_{1:T}^i, u_{1:T}^j} h^i(u_{1:T}^i) + h^j(u_{1:T}^j) \\
 \text{s.t. } \quad & u_{1:T}^i \in F_u^i \text{ and } u_{1:T}^j \in F_u^j \\
 & x_t^i = f_t^i(x_0^i, u_{1:t}^i, v_{1:t}^i), x_t^j = f_t^j(x_0^j, u_{1:t}^j, v_{1:t}^j) \\
 & \mathbb{P} \left\{ (x_{1:T}^i, x_{1:T}^j) \notin F_{1:T}^c \right\} \leq \gamma^{i,j}.
 \end{aligned} \tag{1.44}$$

Denote by 2MA-MPC_N the counterpart of 2MA-MPC with the approximated inter-agent collision chance constraint in Definition 1.4.

In order to prove convergence, we will employ a result from [103] on the convergence of the SAA of general chance constrained optimization problems to our problem of chance constrained MPC (for a brief introduction, please see Appendix A.1).

Definition 1.5 (Optimal Values) Define $\varphi^{i,j}$ to be the optimal objective function value of 2MA-MPC with the “true” chance constraint on inter-agent collision probabilities. Define $\varphi_N^{i,j}$ to be the optimal objective function value for 2MA-MPC_N with the “approximated” chance constraint from Definition 1.4.

Definition 1.6 (Optimal Controls) Define $\Phi^{i,j}$ to be the set of joint controls $(u_{1:T}^i, u_{1:T}^j)$ at which the optimal value $\varphi^{i,j}$ is achieved. Analogously, define $\Phi_N^{i,j}$ to be the set of joint controls $(u_{1:T}^i, u_{1:T}^j)$ at which the optimal value $\varphi_N^{i,j}$ is achieved for the approximated constraints.

We need the following assumptions to hold to prove the convergence.

Assumption 1 For the problem 2MA-MPC

1. $\Phi^{i,j} \neq \emptyset$ and $\Phi_N^{i,j} \neq \emptyset$ holds, i.e., both 2MA-MPC and 2MA-MPC_N are feasible.
2. For an optimal joint plan $(*u_{1:T}^i, *u_{1:T}^j) \in \Phi^{i,j}$ with collision probability

$$\mathbb{P}\left\{(x_{1:T}^i, x_{1:T}^j) \notin F_{1:T}^c \mid (*u_{1:T}^i, *u_{1:T}^j)\right\} = \gamma^{i,j} \quad (1.45)$$

and arbitrary $\rho > 0$ there exists a joint plan $(u_{1:T}^i, u_{1:T}^j)$ with

$$d_U\left((*u_{1:T}^i, *u_{1:T}^j), (u_{1:T}^i, u_{1:T}^j)\right) < \rho \quad (1.46)$$

such that

$$\mathbb{P}\left\{(x_{1:T}^i, x_{1:T}^j) \notin F_{1:T}^c \mid (u_{1:T}^i, u_{1:T}^j)\right\} < \gamma^{i,j}, \quad (1.47)$$

where d_U is the normal Euclidean distance in $\mathbb{R}^{d_U^i} \times \mathbb{R}^{d_U^j}$ of two joint plans.

3. The objective functions h^i and h^j are continuous.

Assumption 1.2 requires that if optimal joint controls have violation probability equal to $\gamma^{i,j}$, then there is a “backoff” joint plan that is arbitrarily close to the optimal plan, for that the inter-agent collision probability will be smaller than $\gamma^{i,j}$. In most real world scenarios it is safe to assume this, since e.g. for motion planning agents will have the chance to wait a short while or make a very small detour to decrease the collision probabilities. Assumption 1.3 is also no severe restriction, since in applications mostly linear, piece-wise linear and convex, or quadratic objective functions are considered.

Now we are in a position to establish the convergence of plans found with the SAA.

Proposition 1.1 *If Assumption 1 holds for problem 2MA-MPC, then we have that $\phi_N^{i,j} \rightarrow \phi^{i,j}$ and $d_H(\Phi_N^{i,j}, \Phi^{i,j}) \rightarrow 0$ with probability one for $N \rightarrow \infty$, i.e., optimal controls found with the SAA of the chance constraints converge against controls that are optimal with the original constraints on inter-agent collision probabilities. Here, d_H denotes the Hausdorff distance between two sets*

$$d_H(\mathcal{A}, \mathcal{B}) := \max \left\{ \sup_{a \in \mathcal{A}} d(a, \mathcal{B}), \sup_{b \in \mathcal{B}} d(b, \mathcal{A}) \right\}. \quad (1.48)$$

Remark 1.4 Here convergence is meant in the following sense: Since we draw N realizations from the random variables $x_{1:T}^i$ and $x_{1:T}^j$ and use these realizations to compute an optimal solution of 2MA-MPC_N the optimal

solution set and optimal value will also be random, depending on the N realizations. The statement of Proposition 1.1 for the optimal solution sets is then

$$\mathbb{P} \left\{ \lim_{N \rightarrow \infty} d_H(\Phi_N^{i,j}, \Phi^{i,j}) = 0 \right\} = 1 \quad (1.49)$$

and equivalently for $\varphi_N^{i,j} \rightarrow \varphi^{i,j}$.

PROOF. As we already mentioned the proof of the Proposition uses a result on general SAA of chance constraints introduced in [103]. In order to apply the result, we have to check that the assumptions of [103, Proposition 2.2] are given (see Appendix A.1 for details).

First, we will show that the constraint function in the chance constraints in 2MA-MPC is a Carathéodory function, i.e., continuous in the decision variables and measurable in the random parameters. Define

$$G_t(u_{1:t}^i, u_{1:t}^j, x_0^i, x_0^j, v_{1:t}^i, v_{1:t}^j) := \|x_t^i - x_t^j\| - \varepsilon \quad (1.50)$$

$$= \|x(u_{1:t}^i, x_0^i, v_{1:t}^i) - x(u_{1:t}^j, x_0^j, v_{1:t}^j)\| - \varepsilon. \quad (1.51)$$

Since the model of the agents' dynamics are linear and $x \rightarrow \|x\|$ is a continuous mapping, the mapping

$$(u_{1:t}^i, u_{1:t}^j) \rightarrow G_t(u_{1:t}^i, u_{1:t}^j, x_0^i, x_0^j, v_{1:t}^i, v_{1:t}^j) \quad (1.52)$$

is continuous for fixed random parameters. With the same argument also the mapping

$$(x_0^i, x_0^j, v_{1:t}^i, v_{1:t}^j) \rightarrow G_t(u_{1:t}^i, u_{1:t}^j, x_0^i, x_0^j, v_{1:t}^i, v_{1:t}^j) \quad (1.53)$$

is continuous for fixed control inputs and, hence, G_t is measurable in the random parameters. So we have shown that G_t is a Carathéodory function.

The inter-agent collision avoidance chance constraint is equivalent to

$$\mathbb{P} \left\{ \max_{t=1, \dots, T} \left\{ G_t(u_{1:t}^i, u_{1:t}^j, x_0^i, x_0^j, v_{1:t}^i, v_{1:t}^j) \right\} > 0 \right\} \leq \gamma^{i,j} \quad (1.54)$$

since the requirement that $G_t \leq 0$ for all $t = 1, \dots, T$ is equivalent to the requirement that $\max_{t=1, \dots, T} \{G_t\} \leq 0$. The pointwise maximum of a finite number of continuous functions is again continuous since for two function $f(x)$ and $g(x)$ it holds that

$$h(x) := \max\{f(x), g(x)\} = \frac{f(x) + g(x) + |f(x) - g(x)|}{2}, \quad (1.55)$$

which is continuous if f and g are continuous. The same holds for measurability and, hence, the overall constraint function is a Carathéodory function.

Our Assumption 1.2 is equivalent to [103, Assumption (A)] and, hence, Proposition 1.1 follows from [103, Proposition 2.2]. \square

1.6.3. MBL Constraint Formulation for Sample-Based Approximations

In this section, we will outline how the SAA from Definition 1.4 of the probability of a collision between two agents can be transformed into MBL constraints. For notational simplicity we assume that the dimensions of the agents' states coincide, i.e., $d_1 = d = d_2$ and collision avoidance is formulated for the complete agent state.

According to Definition 1.4, the approximation of the constraints

$$\mathbb{P} \left\{ (x_{1:T}^1, x_{1:T}^2) \notin F^c \right\} \leq \gamma^{1,2} \quad (1.56)$$

is

$$\sum_{j=1}^N \sum_{l=1}^N \mathbb{1}_{CF^c}(x_j^1, x_l^2) \leq N^2 \gamma^{1,2}, \quad (1.57)$$

where

$$\mathbb{1}_{CF^c}(x_j^1, x_l^2) := \begin{cases} 1, & \text{if } \|x_j^1 - x_l^2\| < \varepsilon \\ 0, & \text{otherwise,} \end{cases} \quad (1.58)$$

is the indicator function of the complement of F^c . Define the auxiliary indicator

$$\mathbb{1}_{CF^c,t}(x_{tj}^1, x_{tl}^2) := \begin{cases} 1, & \text{if } \|x_{tj}^1 - x_{tl}^2\| < \varepsilon \\ 0, & \text{otherwise,} \end{cases} \quad (1.59)$$

where x_{tj}^1 and x_{tl}^2 are the part of the sample trajectories x_j^1 and x_l^2 at time step t according to the construction in Section 1.5.2, i.e., this function will be one if the samples at time step t come to close. It follows that

$$\mathbb{1}_{CF^c} = \min_{t=1,\dots,T} \{\mathbb{1}_{CF^c,t}\} \quad (1.60)$$

from the definition of the event of a collision and, hence,

$$\sum_{j=1}^N \sum_{l=1}^N \mathbb{1}_{CF^c}(x_j^1, x_l^2) \quad (1.61)$$

$$= \sum_{j=1}^N \sum_{l=1}^N \min_{t=1,\dots,T} \{\mathbb{1}_{CF^c,t}(x_{tj}^1, x_{tl}^2)\} . \quad (1.62)$$

We first formulate the indicator functions $\mathbb{1}_{CF^c,t}$ through MBL constraints: We have to iterate through all samples $j = 1 \dots N$ of the first agent and all samples $l = 1 \dots N$ of the second agent and check if $\mathbb{1}_{CF^c,t}(x_{tj}^1, x_{tl}^2)$ equals one or zero. In order to formulate this “check” we introduce a binary variable $e_{jlt} \in \{0, 1\}$ for that holds if $e_{jlt} = 1$ then samples x_{tj}^1 and x_{tl}^2 are within ε proximity of each other and if $e_{jlt} = 0$ they are not.

By definition, the sample $x_{tj}^1 = [x_{tj}^1(1), \dots, x_{tj}^1(d)]^\top$ is more than ε away

from sample $x_{it}^2 = [x_{it}^2(1), \dots, x_{it}^2(d)]^\top$ in the maximum norm if

$$\|x_{ij}^1 - x_{it}^2\| = \max_{i=1, \dots, d} \{|x_{ij}^1(i) - x_{it}^2(i)|\} > \varepsilon. \quad (1.63)$$

This is equivalent to the condition that one of the following inequalities holds

$$x_{ij}^1(1) - x_{it}^2(1) > \varepsilon \text{ or} \quad (1.64)$$

$$x_{it}^2(1) - x_{ij}^1(1) > \varepsilon \text{ or} \quad (1.65)$$

$$\vdots$$

$$x_{ij}^1(d) - x_{it}^2(d) > \varepsilon \text{ or} \quad (1.66)$$

$$x_{it}^2(d) - x_{ij}^1(d) > \varepsilon, \quad (1.67)$$

where we just formulated the absolute value and the maximum from Equation 1.63 differently. Since the logical “or”-constraints above are not directly applicable as linear constraints, we use the “Big M”-method to transform them into logical “and”-constraints.

So we check whether sample x_{ij}^1 is in ε proximity of sample x_{it}^2 at time step t through the constraints

$$\begin{aligned} x_{ij}^1(1) - x_{it}^2(1) &> \varepsilon - b_{jit}^1 M_o \text{ and} \\ x_{it}^2(1) - x_{ij}^1(1) &> \varepsilon - b_{jit}^2 M_o \text{ and} \\ &\vdots \end{aligned} \quad (1.68)$$

$$x_{ij}^1(d) - x_{it}^2(d) > \varepsilon - b_{jit}^{2d-1} M_o \text{ and}$$

$$x_{it}^2(d) - x_{ij}^1(d) > \varepsilon - b_{jit}^{2d} M_o,$$

with binary slack variables $b_{jit}^i \in \{0, 1\}$ for $i = 1, \dots, 2d$ and arbitrary large positive number M_o . If at least one of the b_{jit}^i above is zero, then the samples have sufficient distance because then the corresponding inequality $x_{it}^2(i) -$

$x_{tj}^1(i) > \varepsilon$ or $x_{tj}^1(i) - x_{tl}^2(i) > \varepsilon$ holds. If all b_{jtl}^i equal one, the samples can be within ε distance of each other.

Now we define the binary variable e_{jtl} in such a way that it determines if the samples at time step t have sufficient clearance. This is done by the constraint

$$\sum_{i=1}^{2d} b_{jtl}^i \leq 2d - 1 + M_o e_{jtl} . \quad (1.69)$$

It follows that if $e_{jtl} = 1$, then all b_{jtl}^i , $i = 1, \dots, 2d$ can be equal to one and the samples x_{tj}^1 and x_{tl}^2 can be within ε distance of each other and if $e_{jtl} = 0$, at least one of the b_{jtl}^i , $i = 1, \dots, 2d$ has to be zero and the samples have sufficient clearance. So the binary variables e_{jtl} correspond to the indicator function $1 - \mathbb{1}_{CF^c,t}$ in the sense that if $e_{jtl} = 0$ it follows that $\mathbb{1}_{CF^c,t}(x_{tj}^1, x_{tl}^2) = 1$.

Now we want to construct a mixed binary formulation for the minimum of the indicators over the complete time horizon. This is done with the constraint

$$\sum_{t=1}^T e_{jtl} \leq M_o o_{jl} \quad (1.70)$$

with $o_{jl} \in \{0, 1\}$. If for one $t \in \{1, \dots, T\}$ the binary variable $e_{jtl} = 1$, then the sum will be greater than zero and o_{jl} has to be set to one. If all of the $e_{jtl} = 0$, i.e., the samples x_j^1 and x_l^2 have sufficient clearance for all time steps, then o_{jl} can be set to zero. The final constraint is now that at most $\gamma^{i,j} N^2$ of the o_{jl} are allowed to be equal to one

$$\sum_{j=1}^N \sum_{l=1}^N o_{jl} \leq \gamma^{i,j} N^2 , \quad (1.71)$$

i.e., at most $\gamma^{i,j} N^2$ many samples can have a clearance lower than ε at some time step. The constraints (1.68), (1.69), (1.70), and (1.71) are the

MBL formulation of the SAA of the chance constraint on the probability of a collision between two agents.

From this construction it can be seen that the number of binary decision variables in the MBL formulation depends quadratically on the number of samples N drawn from the random variables x_t^j . Since one usually aims to employ as many samples as possible in order to warrant a good approximation quality and since the number of binary decision variables in an MBL program has a crucial impact on the runtime of the program a high number of binaries is not desirable. Especially when MA-MPC is to be applied in scenarios in which decisions on controls have to be made under real-time requirements, the high runtime of the SAA makes it almost impossible to apply as we will see in the simulations in Section 1.9. Therefore in the next sections we will investigate more efficient methods for formulating the collision avoidance constraints.

1.7. Efficient Conservative Collision Avoidance in 2D

In the previous section we proposed a SAA of the constraints on inter-agent collision probabilities that will converge when the number of samples goes to infinity. When we transformed this approximation into MBL constraints, it turned out that we have to introduce binary variables for each pairing of samples of different agents. Hence, the number of binary variables that results from the sample-based approximation can be prohibitively high.

In this section, we will therefore propose a computationally more efficient formulation of inter-agent collision avoidance constraints based on *regions of increased probability of presence (RPP)* of agents. Not only does the RPP formulation allow us to generate controls under real-time requirements but we can also prove that controls found with the RPP algorithm are feasible to the problem with chance constraints on the true probabilities of agent collisions. In contrast to the SAA where we showed feasibility

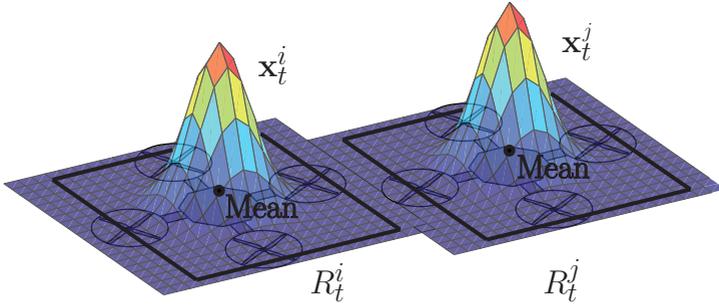


Figure 1.3.: Schematic depiction of regions of increased probability of presence for the example of the positions of two UAVs.

of controls when the number of samples goes to infinity, here feasibility always holds when the RPP constraints are satisfied.

The derivation of the RPP algorithm proceeds in two steps: First we will define a region around the mean of the random position x^i of an agent i and study the probability that the agent is outside this region. The larger this region is, the less probable it is that the position of the agent lies outside the region. This can be quantified by a multivariate generalization of the Chebychev inequality that gives an upper bound on the probability that the position of the agent lies outside of the RPP region.

In the second step we introduce constraints to the MA-MPC problem that ensure that for different agents their respective RPPs do not overlap. We will prove that if these RPPs have the adequate size and they do not overlap, then we can control the collision probabilities in such a way that they do not exceed the chance constraint bounds. In order to define the RPPs and ensure their non-overlap, we have to do the following two things.

1. We have to determine the size of the confidence region of increased probability of presence in such a way that just the right amount of probability mass lies outside of the region. We will do this in Section 1.7.1.

2. We have to formulate constraints that ensure that these regions will not overlap for different agents in order to make sure that the probability of a collision does not exceed the chance constraint bound. We will do this in Section 1.7.2

As before in Section 1.6 we will only consider two agents 1 and 2. Also, in this section we will construct the RPPs for collision avoidance of two-dimensional states of the agents. The reason for this restriction will become apparent in Section 1.7.1 and the case of collision avoidance for arbitrary state dimensions will be considered in Section 1.7.4.

Remark 1.5 We demonstrated in Remark 1.2 that with the help of Boole's inequality the probability of a collision $\mathbb{P}\{(x_{1:T}^1, x_{1:T}^2) \notin F^c\}$ for two complete joint plans can be bounded from above by sum of the probabilities $\mathbb{P}\{(x_t^1, x_t^2) \notin F^c\}$ for time steps $t = 1, \dots, T$. We will use the bound in this section and construct the RPPs in such a way that we can guarantee that

$$\mathbb{P}\{(x_t^1, x_t^2) \notin F^c\} \leq \gamma_t^{i,j} \quad (1.72)$$

for each time step and for some $\gamma_t^{i,j}$. If the $\gamma_t^{i,j}$ are chosen in such a way that

$$\sum_{t=1}^T \gamma_t^{i,j} \leq \gamma^{i,j} \quad (1.73)$$

we can guarantee that the original chance constraint of MA-MPC is satisfied. Since we will only consider the collision probability in one time step t , we will omit the subscript t in this section for notational convenience.

1.7.1. Regions of Increased Probability of Presence (RPP)

Let $\mu^i := [\mu^i(1), \mu^i(2)]^\top \in \mathbb{R}^2$ be the mean of the random position x^i of agent i .

Definition 1.7 (Region of Increased Probability of Presence) We define a rectangular region of increased probability of presence (RPP) around the mean through

$$R^i := \{[x^i(1), x^i(2)]^\top : |x^i(1) - \mu^i(1)| \leq \alpha_1^i, |x^i(2) - \mu^i(2)| \leq \alpha_2^i\} \subset \mathbb{R}^2. \quad (1.74)$$

The RPP R^i describes the set of points in \mathbb{R}^2 for which both coordinates deviate at most some distance from the mean value of the random state of the agent. Its position depends on the mean of the random state of the agent and its size depends on the two parameters α_1^i and α_2^i .

Definition 1.8 (Probability of Presence) For each agent i , we define the probability that the random position of agent i lies outside of the RPP R^i as

$$P^i := \mathbb{P}\{x^i \notin R^i\} = 1 - \mathbb{P}\{x^i \in R^i\}. \quad (1.75)$$

For larger RPPs R^i , the probability P^i becomes smaller as is quantified by the following theorem.

Theorem 1.1 (P. Whittle [130]) Let $x = [x(1), \dots, x(d)]^\top$ be a multivariate random vector with zero mean and covariance matrix $\mathbf{V} \in \mathbb{R}^{d \times d}$, then it holds for

$$P := 1 - \mathbb{P}\{|x(i)| \leq \alpha_i, i = 1, \dots, d\} \quad (1.76)$$

that

$$P \leq \text{trace}\{\mathbf{V}\mathbf{B}^{-1}\}, \quad (1.77)$$

where $\mathbf{B} \in \mathbb{R}^{d \times d}$ is a positive definite matrix with diagonal entries α_i^2 .

If $x^i = [x^i(1), x^i(2)]^\top$ is a zero mean bivariate random vector with covariance matrix $(\mathbf{C}_{kl})_{k,l=1,2}$ and probability P^i for x^i defined as in Definition 1.8, then the probability P^i can be bounded by

$$P^i \leq \frac{\mathbf{C}_{11}(\alpha_2^i)^2 + \mathbf{C}_{22}(\alpha_1^i)^2}{2(\alpha_1^i)^2(\alpha_2^i)^2} + \frac{\sqrt{[\mathbf{C}_{11}(\alpha_2^i)^2 + \mathbf{C}_{22}(\alpha_1^i)^2]^2 - 4\mathbf{C}_{12}^2(\alpha_1^i)^2(\alpha_2^i)^2}}{2(\alpha_1^i)^2(\alpha_2^i)^2}. \quad (1.78)$$

Here $\text{trace}\{\mathbf{A}\}$ denotes the trace of a square matrix \mathbf{A} and is given by the sum of the diagonal elements of the matrix.

Remark 1.6 According to [130, Theorem 2] for a general multivariate random variable bound (1.77) is at its tightest when the matrix \mathbf{B} is the unique solution to $\mathbf{V} = \mathbf{B}\mathbf{\Lambda}\mathbf{B}$ where $\mathbf{\Lambda}$ is some positive definite diagonal matrix and \mathbf{B} has diagonal entries α_i^2 . For general dimension greater than two there appears to be no efficient way to determine such a splitting of \mathbf{V} . This is the reason why we consider collision avoidance in two-dimensional space first and will have to work a bit more to generalize to higher state dimensions in Section 1.7.4.

We will use the upper bound (1.78) on the rate how P^i decreases as α_1^i and α_2^i increase to derive a method on how to determine a size of the region R^i such that just the right amount of probability mass lies outside the region. Since the bound will play an important role, we will denote it by $C(x^i, R^i)$, i.e.,

$$C(x^i, R^i) := \frac{\mathbf{C}_{11}(\alpha_2^i)^2 + \mathbf{C}_{22}(\alpha_1^i)^2}{2(\alpha_1^i)^2(\alpha_2^i)^2} + \frac{\sqrt{[\mathbf{C}_{11}(\alpha_2^i)^2 + \mathbf{C}_{22}(\alpha_1^i)^2]^2 - 4\mathbf{C}_{12}^2(\alpha_1^i)^2(\alpha_2^i)^2}}{2(\alpha_1^i)^2(\alpha_2^i)^2}. \quad (1.79)$$

RPPs for Fixed Probability of Presence

In this section we want to investigate the following question: Given an amount γ^i of probability mass, how do we determine the size of the region such that at most γ^i of the probability mass of the random state x^i lies outside of the region? Or more formally: *Given $0 \leq \gamma^i \leq 1$ and an random vector $x^i \in \mathbb{R}^2$, how do we determine α_1^i and α_2^i such that*

$$\mathbb{P} \{x^i \notin R^i\} \leq \gamma^i ? \quad (1.80)$$

If we find α_1^i and α_2^i so that

$$C(x^i, R^i) = \gamma^i \quad (1.81)$$

holds, then by Whittle's Chebychev inequality (1.78) in Theorem 1.1 we can guarantee that

$$\mathbb{P} \{x^i \notin R^i\} \leq C(x^i, R^i) = \gamma^i \quad (1.82)$$

the probability mass of the random position x^i outside of the region R^i is at most γ^i .

Since the equation $C(x^i, R^i) = \gamma^i$ is only one equation for the two unknowns α_1^i and α_2^i it is under determined. We propose to choose the parameters α_1^i and α_2^i such that additionally

$$\frac{\alpha_1^i}{\alpha_2^i} = \sqrt{\frac{\mathbf{C}_{11}^i}{\mathbf{C}_{22}^i}} \quad (1.83)$$

holds. This choice is motivated by the intuition that for a random state with axis-aligned Gaussian distribution (i.e., $\mathbf{C}_{12}^i = 0$ in the covariance), the diagonal of the covariance matrix given through \mathbf{C}_{11}^i and \mathbf{C}_{22}^i , quantifies the extent of the covariance ellipsoid in x -direction and y -direction. If the ratio

of α_1^i and α_2^i equals the ratio of $\sqrt{\mathbf{C}_{11}^i}$ and $\sqrt{\mathbf{C}_{22}^i}$, the shape and extent of the region R^i follows the shape and extent of the covariance ellipsoid. So if there is considerable uncertainty in one of the coordinate directions, indicated by a covariance ellipsoid with strong extent in this direction, the region will also have a stronger spread in this direction to account for this increased uncertainty.

When we insert the equation for α_2^i that results from (1.83)

$$\alpha_2^i = \sqrt{\frac{\mathbf{C}_{22}^i}{\mathbf{C}_{11}^i}} \alpha_1^i \quad (1.84)$$

into the equation for the bound $C(x^i, R^i) = \gamma^i$ the latter is an equation with only one remaining unknown. The equation for the remaining unknown is a polynomial of degree four with two real solutions that can be determined analytically. The positive real solution of the equation is

$$\alpha_1^i = \sqrt{\frac{\mathbf{C}_{11}^i}{\gamma^i} + \frac{\sqrt{\mathbf{C}_{11}^i \mathbf{C}_{22}^i (\mathbf{C}_{11}^i \mathbf{C}_{22}^i - (\mathbf{C}_{12}^i)^2) (\gamma^i)^2}}{\mathbf{C}_{22}^i (\gamma^i)^2}}. \quad (1.85)$$

Given an uncertain state x^i together with the covariance matrix of this state and given a level γ^i , Equations (1.85) and (1.83) allow us to determine the RPP region R^i such that $C(x^i, R^i) = \gamma^i$ holds. Together with Whittle's Chebychev inequality we can then guarantee that the probability mass of x^i outside the RPP region R^i is at most γ^i and, hence, $\mathbb{P}\{x^i \notin R^i\} \leq \gamma^i$.

1.7.2. Collision Avoidance Based on Non-Overlapping RPPs

In this section we will make use of the results of the previous section to derive the RPP formulation of collision avoidance constraints. We consider two agents 1 and 2 with states x_t^1 and x_t^2 together with an upper bound $0 \leq \gamma_t^{1,2} \leq 1$ on the probability of a collision of the states of these agents at time t . We will derive the RPP formulation of collision avoidance constraints

and then prove that controls found with the RPP constraints are feasible for the original problem with bound on the probability of a collision.

The RPP constraints are constructed in two steps: In the first step we determine the RPPs R_t^1 and R_t^2 for agents 1 and 2 such that the probability mass outside of the RPPs is at most some γ_t^i for $i = 1, 2$ with $\gamma_t^{1,2} = \gamma_t^1 + \gamma_t^2$. In the second step a constraint is added to the optimization problem that warrants that the two RPPs will not overlap at time step t .

Remark 1.7 Since we will introduce constraints on the states that ensure that RPPs for different agents do not overlap, the RPPs of the other agents will in a way pose obstacles for an agent. In order to have a feasible state space that is as unrestricted as possible and since more constraints on the states of an agent will have detrimental effects on their objective, it is natural to desire that the RPPs are as small as possible. In all of our considerations about RPPs, we will therefore strive to minimize their size.

We propose that agents 1 and 2 split the chance constraint bound $\gamma_t^{1,2}$ into parts according to

$$\gamma_t^1 = \frac{1}{s} \gamma_t^{1,2} \text{ and } \gamma_t^2 = \frac{s-1}{s} \gamma_t^{1,2}, \quad (1.86)$$

with free parameter $s > 1$. In our simulations in Section 1.9, we used an even split at $s = 2$. Equations $C(x_t^1, R_t^1) = \gamma_t^1$ and $C(x_t^2, R_t^2) = \gamma_t^2$ then uniquely determine the size parameters for the RPP regions.

Now we have everything we need to construct the RPP regions for agents 1 and 2: the means and covariances of their random states and a size for the RPPs. Denote by R_t^1 and R_t^2 the RPPs constructed from these parameters. Next, we define the constraint that ensures that these RPPs do not overlap:

Definition 1.9 (Constraint \mathcal{C}_t) The means μ_t^1 and μ_t^2 have a distance of

more than $(\alpha_{t_1}^1 + \alpha_{t_1}^2) + \varepsilon$ in the x -direction, i.e.

$$|\mu_t^1(1) - \mu_t^2(1)| \geq (\alpha_{t_1}^1 + \alpha_{t_1}^2) + \varepsilon \quad (1.87)$$

or a distance of more than $(\alpha_{t_2}^1 + \alpha_{t_2}^2) + \varepsilon$ in the y -direction, i.e.

$$|\mu_t^1(2) - \mu_t^2(2)| \geq (\alpha_{t_2}^1 + \alpha_{t_2}^2) + \varepsilon . \quad (1.88)$$

The following proposition proves that controls for agents 1 and 2 for that constraint \mathcal{C}_t holds, satisfy that the probability of a collision between agent 1 and 2 at time step t is less or equal than the bound $\gamma_t^{1,2}$, i.e., they are feasible for the problem with bounds on inter-agent collision probabilities.

Proposition 1.2 *Let γ_t^1 and γ_t^2 be such that $\gamma_t^1 + \gamma_t^2 = \gamma_t^{1,2}$ and let R_t^1 and R_t^2 be RPPs so that $C(x_t^i, R_t^i) = \gamma_t^i$ for $i = 1, 2$. Then for any control sequences $u_{1:T}^1$ and $u_{1:T}^2$ for that constraint \mathcal{C}_t is satisfied at time step $1 \leq t \leq T$, the probability of a collision of agents 1 and 2 at time step t is below $\gamma_t^{1,2}$*

$$\mathbb{P} \{ (x_t^1, x_t^2) \notin F^c \} \leq \gamma_t^{1,2} . \quad (1.89)$$

In order to prove Proposition 1.2, we will first establish the following Lemma.

Lemma 1.1 *Let F^c , x_t^1 , and x_t^2 be as above. If constraint \mathcal{C}_t is satisfied, the inequality*

$$\mathbb{P} \{ (x_t^1, x_t^2) \notin F^c \} \leq P_t^1 + P_t^2 \quad (1.90)$$

holds, where as in the previous section $P_t^i = \mathbb{P} \{ x_t^i \notin R_t^i \}$.

PROOF. By marginalization we have

$$\begin{aligned}
 & \mathbb{P}((x_t^1, x_t^2) \notin F^c) \\
 &= \mathbb{P}\{(x_t^1, x_t^2) \notin F^c, x_t^1 \in R_t^1, x_t^2 \in R_t^2\} \\
 &+ \mathbb{P}\{(x_t^1, x_t^2) \notin F^c, x_t^1 \in R_t^1, x_t^2 \notin R_t^2\} \\
 &+ \mathbb{P}\{(x_t^1, x_t^2) \notin F^c, x_t^1 \notin R_t^1, x_t^2 \in R_t^2\} \\
 &+ \mathbb{P}\{(x_t^1, x_t^2) \notin F^c, x_t^1 \notin R_t^1, x_t^2 \notin R_t^2\}.
 \end{aligned} \tag{1.91}$$

We will first show that the probability

$$\mathbb{P}((x_t^1, x_t^2) \notin F^c, x_t^1 \in R_t^1, x_t^2 \in R_t^2) \tag{1.92}$$

is zero by showing that $(R_t^1 \times R_t^2) \cap CF^c = \emptyset$ from which the claim follows because then the event $x_t^1 \in R_t^1 \wedge x_t^2 \in R_t^2 \wedge (x_t^1, x_t^2) \in CF^c$ has zero probability mass. We have $CF^c \subset CF_\infty^c := \{(x, y) \mid \|x - y\|_\infty < \varepsilon\}$ because of the norm inequality $\|x\|_\infty \leq \|x\|$. Hence, $(R_t^1 \times R_t^2) \cap CF^c \subset (R_t^1 \times R_t^2) \cap CF_\infty^c$ and we will show $(R_t^1 \times R_t^2) \cap CF_\infty^c = \emptyset$, then $(R_t^1 \times R_t^2) \cap CF^c = \emptyset$. Let $x_t^1 \in R_t^1$ and $x_t^2 \in R_t^2$, and for contradiction assume that $(x_t^1, x_t^2) \in CF_\infty^c$. For the x -coordinates of x_t^1 and x_t^2 we have that $|x_t^1(1) - x_t^2(1)| < \varepsilon$ and also for the y -coordinates $|x_t^1(2) - x_t^2(2)| < \varepsilon$ since by the definition of CF_∞^c it holds that $\|x_t^1 - x_t^2\|_\infty = \max\{|x_t^1(1) - x_t^2(1)|, |x_t^1(2) - x_t^2(2)|\} < \varepsilon$. For the means we have $|\mu_t^1(1) - \mu_t^2(1)| \leq |\mu_t^1(1) - x_t^1(1)| + |x_t^1(1) - x_t^2(1)| + |x_t^2(1) - \mu_t^2(1)| < \varepsilon + \alpha_{t1}^1 + \alpha_{t1}^2$ because $|x_t^1(1) - \mu_t^1(1)| \leq \alpha_{t1}^1$ and $|x_t^2(1) - \mu_t^2(1)| \leq \alpha_{t1}^2$ per definition of the RPPs R_t^1 and R_t^2 . The same is true for the y -coordinate and, thus, we have constructed a contradiction to the assumption that constraint \mathcal{C}_t holds. Thus, $\mathbb{P}\{(x_t^1, x_t^2) \notin F^c, x_t^1 \in R_t^1, x_t^2 \in R_t^2\} = 0$ holds if constraint \mathcal{C}_t is satisfied.

For the second summand in (1.91), we have

$$\mathbb{P}\{(x_t^1, x_t^2) \notin F^c, x_t^1 \in R_t^1, x_t^2 \notin R_t^2\} \quad (1.93)$$

$$\stackrel{(a)}{\leq} \mathbb{P}\{x_t^1 \in R_t^1, x_t^2 \notin R_t^2\} \quad (1.94)$$

$$\stackrel{(b)}{=} \mathbb{P}\{x_t^1 \in R_t^1\} \mathbb{P}\{x_t^2 \notin R_t^2\} = (1 - P_t^1)P_t^2,$$

where inequality (a) follows from the fact that intersecting with an additional event can only decrease their probability mass. Equality (b) holds because we assumed in the formulation of the MA-MPC problem that the initial random vectors x_0^1 and x_0^2 as well as all the disturbances $v_{1:t}^1$ and $v_{1:t}^2$ are stochastically independent and, hence, so are the random vectors x_t^1 and x_t^2 .

The third and fourth summand in (1.91) can be bounded with the same arguments as the second summand, with the fourth summand being bounded by $P_t^1 P_t^2$. Through summation of all the bounds, we obtain

$$\mathbb{P}((x_t^1, x_t^2) \notin F^c) \quad (1.95)$$

$$\leq (1 - P_t^1)P_t^2 + (1 - P_t^2)P_t^1 + P_t^1 P_t^2 \quad (1.96)$$

$$= P_t^1 + P_t^2 - P_t^1 P_t^2. \quad (1.97)$$

Since both $P_t^1 \geq 0$ and $P_t^2 \geq 0$ hold and then also $P_t^1 P_t^2 \geq 0$, we have the slightly more coarse inequality

$$\mathbb{P}\{(x_t^1, x_t^2) \notin F^c\} \leq P_t^1 + P_t^2. \quad (1.98)$$

□

We have thus shown that the probability of a collision $\mathbb{P}\{(x_t^1, x_t^2) \notin F^c\}$ can be bounded from above by the sum of the probabilities that the random states x_t^1 and x_t^2 are outside the RPP regions R_t^1 and R_t^2 .

PROOF.(Proof of Proposition 1.2)

Summarizing the results of Lemma 1.1 and the construction of the RPPs together with Theorem 1.1 we can deduce Proposition 1.2:

$$\mathbb{P} \{ (x_t^1, x_t^2) \notin F^c \} \leq P_t^1 + P_t^2 \quad (1.99)$$

$$\leq C(x_t^1, \alpha_t^1) + C(x_t^2, \alpha_t^2) \quad (1.100)$$

$$= \gamma_t^1 + \gamma_t^2 \quad (1.101)$$

$$= \gamma_t^{1,2} \quad (1.102)$$

if constraint \mathcal{C}_t holds, where $C(x_t^1, \alpha_t^1)$ and $C(x_t^2, \alpha_t^2)$ are the Chebychev bounds as defined in (1.78). \square

Corollary 1.1 *If constraint \mathcal{C}_t holds for joint controls of agents 1 and 2 for every $t = 1, \dots, T$ and if*

$$\sum_{t=1}^T \gamma_t^{1,2} \leq \gamma^{1,2}, \quad (1.103)$$

it follows that

$$\mathbb{P} \{ (x_{1:T}^1, x_{1:T}^2) \notin F^c \} \leq \gamma^{1,2}, \quad (1.104)$$

i.e., the controls are feasible for the inter-agent collision avoidance constraint over the complete planning horizon.

PROOF. The proof follows from Proposition 1.2 and Remark 1.2. \square

The proposition and the corollary warrant that if we solve the chance constrained MA-MPC problem with RPP constraints of the form \mathcal{C}_t , then the obtained controls are automatically feasible for the chance constrained MA-MPC problem with full constraints on inter-agent collision probabilities. Hence, the RPP constraints allow us to find solutions to the MA-MPC problem with the complicated probabilistic coupling constraints without any knowledge about the agents' random states besides the covariance and without evaluating the probability of a collision of two agents.

It suggests itself that controls found for the MA-MPC problem with RPP constraints are more suboptimal than controls for the MA-MPC problem with full probabilistic constraints, since the RPP constraints are a conservative tightening. We will therefore in Section 1.8 propose a less conservative reformulations and study in simulations the effects of the conservativeness on the controls.

1.7.3. Constraint Formulation for RPP Method

In this section we will outline how the RPP constraints can be formulated as mixed-binary linear constraints. In summary, the RPP constraints for a time instance t can be constructed as follows:

- For all agents $i = 1, \dots, M$ we determine $\text{Cov}(x_t^i)$ according to (1.41). Please note, that this is not done at run time of the MBL optimization routine but before.
- Also before the mixed-binary solver starts, for all agent combinations i and j and upper bounds on the collision probability $\gamma_t^{i,j}$ we determine the RPP regions such that $C(x_t^i, R_t^i)$ and $C(x_t^j, R_t^j)$ equal $\frac{1}{s}\gamma_t^{i,j}$ and $\frac{s-1}{s}\gamma_t^{i,j}$ respectively.
- We replace the probabilistic collision avoidance constraint (1.8) by the MBL formulation of constraint \mathcal{C}_t .
- Then the MBL optimization routine solves the MBL program possibly with further constraints resulting from obstacle avoidance (see Section 1.5.2).

Constraint \mathcal{C}_t for agents 1 and 2 is that at least one of the coordinates in the 2D-plane of their expected values have a distance of at least $(\alpha_{t1}^1 + \alpha_{t1}^2) + \varepsilon$ or $(\alpha_{t2}^1 + \alpha_{t2}^2) + \varepsilon$ respectively.

Now we model the RPP constraint \mathcal{C}_t for a time step t in the planning horizon and a pair of agents denoted by 1 and 2 as constraints for the MBL

optimization routine. Therefore, we introduce constraints on the distance of the means to model the RPP constraint \mathcal{C}_i

$$\mu_i^1(1) - \mu_i^2(1) \geq (\alpha_{r1}^1 + \alpha_{r1}^2) + \varepsilon \text{ or} \quad (1.105)$$

$$\mu_i^2(1) - \mu_i^1(1) \geq (\alpha_{r1}^1 + \alpha_{r1}^2) + \varepsilon \text{ or} \quad (1.106)$$

$$\mu_i^1(2) - \mu_i^2(2) \geq (\alpha_{r2}^1 + \alpha_{r2}^2) + \varepsilon \text{ or} \quad (1.107)$$

$$\mu_i^2(2) - \mu_i^1(2) \geq (\alpha_{r2}^1 + \alpha_{r2}^2) + \varepsilon . \quad (1.108)$$

Since a MBL solver can not directly understand logical “or“-constraints, we formulate them as logical “and“-constraints with the ”Big M“-method as in Sec. 1.6.3

$$\mu_i^1(1) - \mu_i^2(1) \geq (\alpha_{r1}^1 + \alpha_{r1}^2) + \varepsilon - M_o b_i^1 \text{ and} \quad (1.109)$$

$$\mu_i^2(1) - \mu_i^1(1) \geq (\alpha_{r1}^1 + \alpha_{r1}^2) + \varepsilon - M_o b_i^2 \text{ and} \quad (1.110)$$

$$\mu_i^1(2) - \mu_i^2(2) \geq (\alpha_{r2}^1 + \alpha_{r2}^2) + \varepsilon - M_o b_i^3 \text{ and} \quad (1.111)$$

$$\mu_i^2(2) - \mu_i^1(2) \geq (\alpha_{r2}^1 + \alpha_{r2}^2) + \varepsilon - M_o b_i^4 \text{ and} \quad (1.112)$$

$$b_i^i \in \{0, 1\} \text{ and} \quad (1.113)$$

$$\sum_{i=1}^4 b_i^i \leq 3 , \quad (1.114)$$

with large positive number M_o . Please note, that the proceeding is exactly the same as in Section 1.6.3: if one of the binary variables b_i^i equals one it is possible that the corresponding constraint on the distance of the means is not satisfied. Since at least one of the four constraints in the “or“-formulation above has to be satisfied in order to guarantee that the means are far enough apart, we limit the number of binary variables b_i^i that are allowed to be equal to one by three. The equality constraint (1.40) for the mean and the inequality constraints (1.109)- (1.114) form the MBL constraints equivalent to the RPP constraint \mathcal{C}_i .

1.7.4. Extension to Higher Dimensional State Spaces

Unfortunately the explicit form of Whittle's Chebychev inequality in Theorem 1.1 holds only for two-dimensional random vectors. However, it may be desirable to formulate collision avoidance chance constraints in state spaces with dimension higher than two. This may be the case for example for path planning with UAVs in three-dimensional space where also the height of flight can be controlled. In this section we therefore propose an efficient and exact approach based on Chebychev inequalities for chance constraints for collision avoidance in state spaces with arbitrary dimension.

The general idea of the approach is based on the following observation: If \mathbf{V} denotes the covariance matrix of the random vector $x = [x(1), \dots, x(d)]^\top$ and if P is defined as in Theorem 1.1 by

$$P := 1 - \mathbb{P}\{|x(i)| \leq \alpha_i, i = 1, \dots, d\} \quad (1.115)$$

then the inequality

$$P \leq \text{trace}\{\mathbf{V}\mathbf{B}^{-1}\} \quad (1.116)$$

with

$$\mathbf{B} = \begin{bmatrix} \alpha_1^2 & & 0 \\ & \ddots & \\ 0 & & \alpha_d^2 \end{bmatrix} \quad (1.117)$$

simplifies to

$$P \leq \sum_{i=1}^d \frac{v_i}{\alpha_i^2} \quad (1.118)$$

if \mathbf{V} is also a diagonal matrix, i.e.,

$$\mathbf{V} = \begin{bmatrix} v_1 & & 0 \\ & \ddots & \\ 0 & & v_d \end{bmatrix}. \quad (1.119)$$

The equality

$$\sum_{i=1}^d \frac{v_i}{\alpha_i^2} = \gamma \quad (1.120)$$

that is solved in Section 1.7.1 to determine the optimal RPP is then unambiguously solvable when we consider additional $d - 1$ equalities for the α_i . The procedure in this section is therefore the following

1. The covariance matrices of the states of the agents will in general not be diagonal matrices. However, through a simple transformation of the state random vectors their respective covariance matrices are converted to diagonal matrices.
2. We will define RPPs transformed with the transformation found in the first step and the probability of presence in state dimensions higher than two and will propose additional constraints on the α_i in order to find optimal RPPs.
3. We will show that controls that are feasible for this constraint will also be feasible for the original MA-MPC problem in Section 1.4.
4. We will formulate MBL constraints that ensure that the transformed RPPs of this section will not overlap.

Since the following constructions will be carried out for a fixed time step t , we will omit the subscript t denoting the time step for notational convenience.

Transformation of Agents' States

Since the covariance matrix \mathbf{C}^i for an agent i is symmetric and positive semi-definite there always exists an *eigendecomposition*

$$\mathbf{C}^i = (\mathbf{U}^i)^\top \Lambda^i \mathbf{U}^i \quad (1.121)$$

with orthonormal matrix \mathbf{U}^i and *diagonal* matrix Λ^i . The fact that such an eigendecomposition exists is a well known fact from multivariate analysis [37, 47, 70]. If we now consider the transformed random vector

$$y^i := \mathbf{U}^i x^i \quad (1.122)$$

it follows from covariance matrix algebra that

$$\text{Cov}(y^i) = \mathbf{U}^i \text{Cov}(x^i) (\mathbf{U}^i)^\top = \mathbf{U}^i \mathbf{C}^i (\mathbf{U}^i)^\top = \Lambda^i, \quad (1.123)$$

i.e., the linearly transformed random vector y^i has a covariance matrix that is a diagonal matrix. Given a random state x^i of an agent i with covariance matrix \mathbf{C}^i there always exists an orthonormal linear transformation \mathbf{U}^i such that the transformed random vector $y^i = \mathbf{U}^i x^i$ has diagonal covariance matrix.

Remark 1.8 In multivariate statistics the above procedure is well-known in *principle component analysis* in order to transform strongly correlated data, i.e., data with a sample covariance that is not in diagonal form, into uncorrelated data with sample covariance in diagonal form. Then data compression, feature extraction or a projection of the data onto lower dimensional subspaces can be accomplished [37].

Transformed RPPs

In the previous section we transformed the random state x^i into another random vector y^i with an orthonormal matrix \mathbf{U}^i such that y^i has covariance matrix in diagonal form. For y^i a RPP analogously to the one in Section 1.7.1 can be constructed. Then with the inverse transformation $(\mathbf{U}^i)^\top$ we can transform the RPP of y^i back into a RPP for x^i and consider the probability the x^i takes values outside of it.

Definition 1.10 (Transformed RPP) For the transformed random vector y^i as above let

$$\tilde{R}^i := \{y^i \in \mathbb{R}^d : |y^i(k) - (\mathbf{U}^i \mu^i)(k)| \leq \alpha_k^i, k = 1, \dots, d\} \subset \mathbb{R}^d \quad (1.124)$$

where μ^i is the mean of x^i and, hence, $\mathbf{U}^i \mu^i$ is the mean of y^i . Define the transformed RPP R^i as

$$R^i := (\mathbf{U}^i)^\top \tilde{R}^i, \quad (1.125)$$

i.e., while \tilde{R}^i is the straight forward extension of the concept of an RPP for the transformed random variable y^i , the transformed RPP is the inverse transformation of (\mathbf{U}^i) applied to \tilde{R}^i .

Definition 1.11 (Transformed Probability of Presence) Define for agent i

$$P^i := 1 - \mathbb{P}\{x^i \in R^i\} \quad (1.126)$$

for the transformed RPP R^i , i.e., P^i is the probability that the random vector x^i will take values outside of R^i .

By construction the following lemma holds.

Lemma 1.2 For a random vector x^i , the transformed random vector y^i as above, and the transformed RPPs R^i and \tilde{R}^i as in Definition 1.10 it holds that

$$\mathbb{P}\{x^i \in R^i\} = \mathbb{P}\{y^i \in \tilde{R}^i\}, \quad (1.127)$$

i.e., the probability that x^i takes values inside R^i equals the probability that y^i takes values inside \tilde{R}^i .

PROOF. The proof follows from the construction in this and the previous section

$$\mathbb{P}\{y^i \in \tilde{R}^i\} = \mathbb{P}\{\mathbf{U}^i x^i \in \tilde{R}^i\} \stackrel{(a)}{=} \mathbb{P}\{x^i \in (\mathbf{U}^i)^{-1} \tilde{R}^i\} \quad (1.128)$$

$$= \mathbb{P}\{x^i \in (\mathbf{U}^i)^\top \tilde{R}^i\} = \mathbb{P}\{x^i \in R^i\}. \quad (1.129)$$

Equality (a) holds since if $y^i \in \tilde{R}^i$ there exists a $z \in \tilde{R}^i$ such that $y = z$ which is equivalent to $\mathbf{U}^i x^i = z$ and this is equivalent to $x^i = (\mathbf{U}^i)^\top z \in R^i$ by definition of R^i . \square

From Lemma 1.2 we can conclude that for the transformed probability of presence it holds that

$$P^i = 1 - \mathbb{P}\{x^i \in R^i\} = 1 - \mathbb{P}\{y^i \in \tilde{R}^i\} \stackrel{(a)}{\leq} \sum_{j=1}^d \frac{\lambda_j}{\alpha_j^2}, \quad (1.130)$$

where inequality (a) follows from the observation in (1.118) at the beginning of this Section for

$$\text{Cov}(y^i) = \Lambda^i = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{bmatrix}. \quad (1.131)$$

Determination of RPPs

The transformed RPP is defined through the linear transformation \mathbf{U}^i , that is computed from the covariance matrix of the random vector x^i , and the size parameters α_j . In order to find the parameter α_j for a transformed RPP of optimal size, i.e., minimal in the sense of Remark 1.7, for a minimum probability of presence $1 - \gamma^i$ we have to add to the under-determined equality

$$\sum_{j=1}^d \frac{\lambda_j}{\alpha_j^2} = \gamma^i \quad (1.132)$$

further equality constraints on the α_j . Possible further constraints are for $j_1 \neq j_2$

$$\frac{\alpha_{j_1}}{\alpha_{j_2}} = \frac{\lambda_{j_1}}{\lambda_{j_2}} \quad \text{or} \quad \frac{\alpha_{j_1}^2}{\alpha_{j_2}^2} = \frac{\lambda_{j_1}}{\lambda_{j_2}} \quad \text{or} \quad \alpha_{j_1} = \alpha_{j_2}, \quad (1.133)$$

where it strongly depends on the random vectors x^i which of these additional equality constraints yields the best results in terms of minimal size of the RPPs. Together with one of these additional equality constraints it is then possible to find unique size parameters α_j for the transformed RPP.

Here, we will explicitly showcase the determination of the α_j only for the case that all the α_j are equal. Then Equation (1.132) simplifies to

$$\frac{1}{\alpha^2} \sum_{j=1}^d \lambda_j = \gamma^i \quad (1.134)$$

which is equivalent to

$$\alpha = \sqrt{\frac{1}{\gamma^i} \sum_{j=1}^d \lambda_j}. \quad (1.135)$$

Non-overlapping RPPs

For two agents with states x_t^1 and x_t^2 and an upper bound $\gamma_t^{i,j}$ on the probability of a collision of these agents, we propose a split as mentioned in Section 1.7 of the chance constraint bound. Then we define the collision avoidance constraint in arbitrary state dimensions as follows:

Definition 1.12 (Transformed Constraint \mathcal{C}_t) The transformed RPPs of the states x_t^1 and x_t^2 have a distance greater than or equal to ε .

In the two-dimensional case in Section 1.7.1 the constraint \mathcal{C}_t was easier to define since it was a constraint on the distance of the means of the random vectors x_t^j . In the case of transformed RPPs it is not as easy to formulate, however, in the next section we will see that the transformed constraint \mathcal{C}_t for transformed RPPs can be still formulated as MBL constraints.

Given that constraint \mathcal{C}_t from Definition 1.12 holds, we can now formulate an equivalent to Proposition 1.2

Proposition 1.3 *Let $1 \leq t \leq T$, γ_t^1 and γ_t^2 be such that $\gamma_t^{1,2} = \gamma_t^1 + \gamma_t^2$ for two agents 1 and 2 and let R_t^1 and R_t^2 be the transformed RPPs of these agents with bounds on the probability of presence given by γ_t^1 and γ_t^2 . Then for any two control sequences $u_{1:T}^1$ and $u_{1:T}^2$ such that constraint \mathcal{C}_t from Definition 1.12 is satisfied by the states resulting from the controls it holds that*

$$\mathbb{P}\{(x_t^1, x_t^2) \notin F^c\} \leq \gamma_t^{i,j}. \quad (1.136)$$

PROOF. With a similar argumentation as in the proof of Proposition 1.2 it follows that

$$\mathbb{P}\{(x_t^1, x_t^2) \notin F^c\} \leq P_t^1 + P_t^2 \quad (1.137)$$

with P_t^j as in Definition 1.11 when constraint \mathcal{C}_t from Definition 1.12 holds. With Lemma 1.2 and by construction the claim then follows. \square

The analog of Corollary 1.1 also holds for the transformed RPPs.

Corollary 1.2 *If constraint \mathcal{C}_t from Definition 1.12 holds for joint controls of agents 1 and 2 for every $t = 1, \dots, T$ and if*

$$\sum_{t=1}^T \gamma_t^{1,2} \leq \gamma^{1,2}, \quad (1.138)$$

it follows that

$$\mathbb{P}\{(x_{1:T}^1, x_{1:T}^2) \notin F^c\} \leq \gamma^{1,2}, \quad (1.139)$$

i.e., the controls are feasible for the inter-agent collision avoidance constraint over the complete planning horizon.

Proposition 1.3 and Corollary 1.2 guarantee that controls for two agents that satisfy the constraint from Definition 1.12 are also feasible for the true chance constraints on collision probabilities. Please note, that we do not make any approximations of the random variables and, hence, all statements hold exactly for the true probability of a collision and not in convergence as for the SAA. Since the constraints on collision probabilities are always defined as pairwise constraints the statement of Corollaries 1.1 and 1.2 generalizes to the case with arbitrary agent numbers.

Constraint Formulation

In this section we will outline how constraint \mathcal{C}_t from Definition 1.12 can be formulated through MBL constraints.

In summary, the transformed RPP constraints at time step t can be constructed as follows (analogously to the procedure in Section 1.7.2):

- For all agents $i = 1, \dots, M$ determine $\text{Cov}(x_t^i)$ according to (1.41).

- For all agent combinations i and j and upper bounds on the collision probability $\gamma_i^{i,j}$ we determine transformed RPP regions as introduced in the previous sections.
- Replace the probabilistic collision avoidance constraint (1.8) by the MBL formulation of constraint \mathcal{C}_i specified in this section.
- Solve the path planning problem with these constraints, again with possibly further constraints resulting from obstacle avoidance.

The RPP \tilde{R}^i is a hyperrectangle around the mean of y^j and, hence, the transformed RPPs can be visualized as rotated hyperrectangles in \mathbb{R}^d , since the linear transformation \mathbf{U}^j is orthonormal and, hence, does not change angles or lengths. Each of the $2d$ faces F_k of such a rotated hyperrectangle defines a hyperplane in \mathbb{R}^d given by a normal vector n_k pointing outward of the hyperrectangle and a support vector b_k . The corresponding “outward pointing” halfspace can be defined through

$$\mathcal{H}_k := \left\{ x \in \mathbb{R}^d : n_k^\top x \leq n_k^\top b_k \right\}. \quad (1.140)$$

In the following we will sketch the formulation of constraint \mathcal{C}_i for collision avoidance between two agents 1 and 2. It can be seen for two hyperrectangles the following fact holds:

Fact 1.1 *Two rotated hyperrectangles R^1 and R^2 in \mathbb{R}^d have a distance greater than or equal to ε if there is a face in either R^1 or R^2 such that all vertices of the other rotated hyperrectangle are in the corresponding “outward pointing” halfspace and have a distance greater than or equal to ε from that face.*

The 2^d respective vertices of the transformed RPPs R_t^1 and R_t^2 can be determined through

$$v_m^i = (\mathbf{U}^i)^\top (\mathbf{U}\mu^i + \tilde{\sigma}_m^i), \quad (1.141)$$

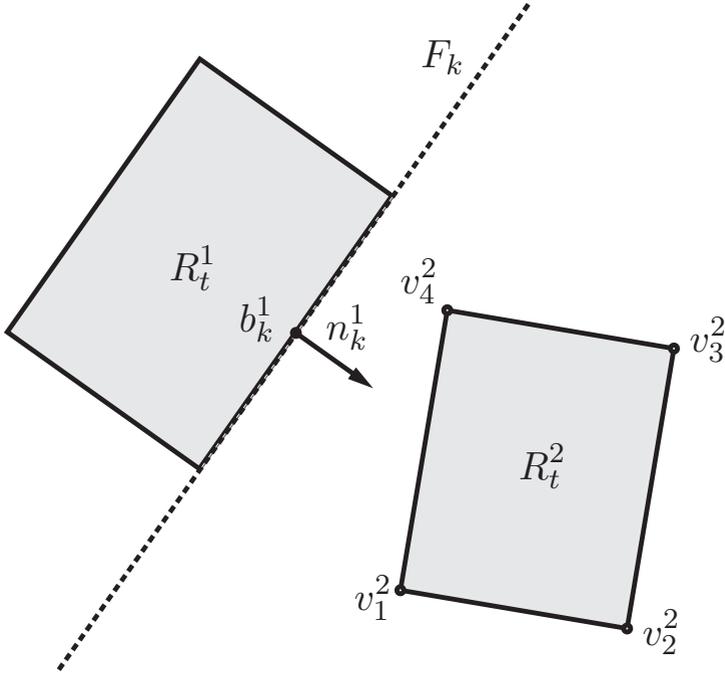


Figure 1.4.: This is an illustration of non-overlapping transformed RPPs: There has to be a face F_k of an RPP R_t^1 such that all vertices v_l^2 of the other RPP R_t^2 lie in the “outwardpointing” halfspace defined by F_k .

where

$$\bar{\alpha}_m^i = \begin{bmatrix} s_1 \alpha_1^i \\ s_2 \alpha_2^i \\ \vdots \\ s_d \alpha_d^i \end{bmatrix} \quad (1.142)$$

with $s_l \in \{-1, 1\}$ for $l = 1, \dots, d$. In Equation 1.141 we go through all vertices of the untransformed RPP hyperrectangle \tilde{R}_t^i from Definition 1.10 and map those vertices onto the vertices of R_t^i under the transformation U_t^i .

Then, for the “outward pointing” normals n_k^i of the faces of the RPPs and the support vectors b_k^i , the MBL constraints are

$$(n_k^1)^\top v_m^2 \leq (n_k^1)^\top b_k^1 - \varepsilon + M_o e_k^1 \quad (1.143)$$

$$(n_k^2)^\top v_m^1 \leq (n_k^2)^\top b_k^2 - \varepsilon + M_o e_k^2 \quad (1.144)$$

for all respective faces $k = 1, \dots, 2d$ in R^1 and R^2 and for all respective vertices v_m^i , $m = 1, \dots, 2^d$ with binary decision variables $e_k^i \in \{0, 1\}$. Of the above inequalities $(n_k^i)^\top v_m^j < (n_k^i)^\top b_k^i - \varepsilon$, one has to be satisfied, i.e., the corresponding binary variable e_k^j set to zero such that Fact 1.1 is fulfilled. For an illustration please see Figure 1.4. So we introduce the additional constraint on the binary variables

$$\sum_{k=1}^{2d} e_k^1 + \sum_{k=1}^{2d} e_k^2 \leq 4d - 1 \quad (1.145)$$

that insures that at least one of the e_k^i is equal to zero and, hence, one of the above inequalities is satisfied.

1.8. Efficient Empirical Conservative Collision Avoidance

RPP regions determined with Whittle’s Chebychev inequality can become conservative for very small chance constraint bounds $\gamma_i^{1,2}$ and respective γ_i^i , since the determination of their size depends inversely on γ_i^i as can be seen from Equations (1.85) and (1.135). Thus, for very small chance constraint bounds on inter-agent collision probabilities the RPPs can become very big and the resulting collision avoidance constraints can become very conservative. In these cases it is desirable to have a formulation that is less conservative than the RPP formulation but can still be solved efficiently under real time requirements. Therefore in this section we will develop a method for obtaining RPP regions whose size is more adequate for small chance constraint bounds.

The main idea is to approximate the true probability $\mathbb{P}\{x \notin R\}$ that the uncertain state of an agent lies outside of the RPP region by the empirical probability derived from the sample approximation of the random vector x as constructed in Section 1.5.2. We will first show how the true probability $\mathbb{P}\{x \notin R\}$ can be approximated and then how an optimal empirical RPP (eRPP) region can be constructed through mixed-binary linear programming. Lastly we will show that eRPPs will converge to true optimal RPP regions for a given probability of presence γ and that controls found with the eRPPs will converge against controls that are feasible for the chance constraint on inter-agent collision probabilities.

1.8.1. Determination of eRPPs

Definition 1.13 (Optimal RPP) We will say that a RPP R is optimal for the true probability $\mathbb{P}\{x \notin R\}$ for a given random vector $x \in \mathbb{R}^d$ with mean μ if its size parameters $\alpha_i, i = 1, \dots, d$ are such that they are a solution to the program

$$\begin{aligned}
 \text{P[oRPP]} : \quad & \min_{\alpha_i, i=1, \dots, d} \sum_{i=1}^d \alpha_i \\
 \text{s.t. } & R = \{x : |x(i) - \mu(i)| \leq \alpha_i, i = 1, \dots, d\} \quad (1.146) \\
 & \mathbb{P}\{x \notin R\} \leq \gamma \\
 & \alpha_i \geq 0, i = 1, \dots, d.
 \end{aligned}$$

Please note that here optimality of the RPP is again understood in the sense of Remark 1.7, i.e., we want the RPP to be as small as possible in order to limit the movement space of the other agents as little as possible.

The determination of an optimal RPP in Definition 1.13 is an optimization problem with probabilistic constraints that for general multivariate distribution of the uncertain state x can not be solved efficiently. This is the reason, why we will again resort to sample approximation methods and

construct what we will call an empirical RPP (eRPP) region based on a sample approximation of the distribution of the state x .

The first step for constructing optimal empirical RPP regions is to approximate the true probability $\mathbb{P}\{x \notin R\}$ through the empirical probability:

$$\mathbb{P}\{x \notin R\} \approx \frac{1}{N} \sum_{j=1}^N \mathbb{1}_{CR}(x_j), \quad (1.147)$$

where the samples $\{x_j\}_{j=1}^N$ are drawn from the distribution of the random vector x and

$$\mathbb{1}_{CR}(x) := \begin{cases} 1 & , \text{ if } x \notin R \\ 0 & , \text{ if } x \in R. \end{cases} \quad (1.148)$$

Then we bound the empirical probability in (1.147) that the uncertain position lies out side of the region R by γ .

Definition 1.14 (SAA of Optimal RPPs) The SAA of the determination of the optimal RPPs is to find the size of the rectangular region R such that the empirical probability is less or equal than γ :

$$\frac{1}{N} \sum_{j=1}^N \mathbb{1}_{CR}(x_j) \leq \gamma. \quad (1.149)$$

It corresponds to the following program

$$\begin{aligned} \text{P[eRPP]} : \quad & \min_{\alpha_i, i=1, \dots, d} \sum_{i=1}^d \alpha_i \\ & \text{s.t. } R = \{x : |x(i) - \mu(i)| \leq \alpha_i, i = 1, \dots, d\} \\ & \frac{1}{N} \sum_{j=1}^N \mathbb{1}_{CR}(x_j) \leq \gamma \\ & \alpha_i \geq 0, i = 1, \dots, d, \end{aligned} \quad (1.150)$$

where as in the definition of the optimal RPP μ is the mean of the random vector x .

Problem P[eRPP] can be visualized as finding the smallest hyperrectangle around the mean μ such that at most γN samples lie outside this hyperrectangle. From now on we will refer to problem (1.146) as the true problem and to problem (1.150) as the approximated problem.

Remark 1.9 The approximated problem is an MBL program. To see this, note that the optimization program of the approximated problem is equivalent to the following linear program with mixed-binary and continuous decision variables

$$\min_{\alpha_i, i=1, \dots, d} \sum_{i=1}^d \alpha_i \quad (1.151)$$

$$\text{s.t. } \forall_{i=1, \dots, d} \forall_{j=1, \dots, N} x_j(i) - \mu(i) \leq \alpha_i + M_o e_{j1}^i \quad (1.152)$$

$$\forall_{i=1, \dots, d} \forall_{j=1, \dots, N} \mu(i) - x_j(i) \leq \alpha_i + M_o e_{j2}^i \quad (1.153)$$

$$\sum_{i=1}^d e_{j1}^i + \sum_{i=1}^d e_{j2}^i \leq M_o o_j \quad (1.154)$$

$$\sum_{j=1}^N o_j \leq \gamma N \quad (1.155)$$

$$e_{j1}^i, e_{j2}^i, o_j \in \{0, 1\} \quad (1.156)$$

$$\alpha_i \geq 0, i = 1, \dots, d. \quad (1.157)$$

If the binary variables e_{j1}^i, e_{j2}^i both are equal to zero, then the sample x_j lies inside the eRPP R around the mean μ given by the side lengths α_i and the corresponding o_j can also be set to zero. If the sample x_j lies outside of the eRPP, one of the inequalities (1.152) or (1.153) is not satisfied and the corresponding e_{j1}^i or e_{j2}^i has to be set to one and, hence, o_j has to be set to one. The bound on the sum of the o_j guarantees that at most γN of the samples lie outside of the region.

1.8.2. Convergence Properties of eRPPs

Now assume we found size parameters α_i such that the empirical probability mass outside of the eRPP region is at most γ . The question that remains is: Do these approximated eRPP regions in some way converge to RPP regions that are optimal to the problem where the true probability mass outside of this region is at most γ ? The answer is yes.

Assumption 2 *We will in the following assume that the probability distribution of x has a positive probability density function, i.e., for the probability density function f of x it holds that $f > 0$.*

There are many examples of positive density functions like Gaussians, sums of Gaussians or general exponential densities. For example if Kalman filtering techniques are used for state estimation of the uncertain states of the agents, the random vectors x all will be Gaussians [121].

We will show that we can guarantee that as the number of samples used to build the eRPP region goes to infinity, the eRPP region will converge against a true RPP region.

Lemma 1.3 *Let Assumption 2 hold for x and let α^* denote the optimal solution of $P[\text{oRPP}]$ and α_N^* the optimal solution of the empirical program $P[\text{eRPP}]$. Then for $N \rightarrow \infty$, i.e., as the number of samples used to construct the eRPP in program $P[\text{eRPP}]$ goes to infinity, we have that $\alpha_N^* \rightarrow \alpha^*$ with probability one. It follows that also the eRPP R_N constructed from α_N^* converges to the true optimal RPP R constructed from α^* with probability one.*

PROOF. We will show that the true problem (1.146) of finding the optimal region and the approximated problem (1.150) satisfy the assumptions of [103, Proposition 2.2]. When we have established this, then it follows that the size parameters α_N^* that are optimal for the approximated problem will converge against size parameters that are optimal for the true problem.

The objective function of the true problem is linear and as such continuous. Define for the random vector $x = [x(1), \dots, x(d)]^\top$ and $\alpha = [\alpha_1, \dots, \alpha_d]^\top$ the functions

$$G_{2i-1}(\alpha, x) := x(i) - \alpha_i, \quad G_{2i}(\alpha, x) := -x(i) + \alpha_i. \quad (1.158)$$

The constraints $G_i(\alpha, x) \leq 0, \forall i = 1, \dots, 2d$ are equivalent to the constraint $x \in R$ where R is constructed with parameter α . In order to summarize these $2d$ constraints into one equivalent constraint we require that

$$G(\alpha, x) := \max_{i=1, \dots, 2d} \{G_i(\alpha, x)\} \leq 0. \quad (1.159)$$

Each of the G_i is an affine function in both α and x and, hence, measurable in x for all α and continuous in α for all x . The maximum of a finite number of measurable functions is again measurable, hence, G is measurable and also G is continuous as can easily be checked.

The last condition we have to check is, whether Assumption (A) from [103] holds for P[oRPP], i.e., if for any $\varepsilon > 0$ there is an optimal solution α^* of the true problem such that there exists an $\bar{\alpha}$ with $\|\bar{\alpha} - \alpha^*\| \leq \varepsilon$ and

$$\mathbb{P} \{G(\bar{\alpha}, x) > 0\} < \gamma. \quad (1.160)$$

Let α^* be an optimal solution to the true problem and let $\varepsilon_0 > 0$. If

$$\mathbb{P} \{G(\alpha^*, x) > 0\} < \gamma \quad (1.161)$$

then we are done, since then Assumption (A) from [103] holds. So assume that

$$\mathbb{P} \{G(\alpha^*, x) > 0\} = \gamma. \quad (1.162)$$

From the definition of the function G we have that

$$\mathbb{P}\{G(\alpha^*, x) > 0\} = \mathbb{P}\{x \notin R(\alpha^*)\}, \quad (1.163)$$

where $R(\alpha^*)$ denotes the RPP constructed from α^* . Let $\bar{\alpha}$ be such that $\bar{\alpha}_i > \alpha_i^*$ in every dimension and further $\|\bar{\alpha} - \alpha^*\| \leq \varepsilon_0$. Then it follows that

$$R(\alpha^*) \subsetneq R(\bar{\alpha}), \quad (1.164)$$

i.e., $R(\alpha^*)$ is strictly included in $R(\bar{\alpha})$ where $R(\bar{\alpha})$ is the RPP constructed from $\bar{\alpha}$. It holds that

$$\begin{aligned} \gamma &= \mathbb{P}\{x \notin R(\alpha^*)\} \\ &= \int_{\mathbb{R}^d \setminus R(\alpha^*)} f(x) dx \\ &\stackrel{(a)}{>} \int_{\mathbb{R}^d \setminus R(\bar{\alpha})} f(x) dx \\ &= \mathbb{P}\{x \notin R(\bar{\alpha})\}, \end{aligned} \quad (1.165)$$

where f is the pdf of the random vector x . Inequality (a) holds because we assumed that $f > 0$ and, hence, the integral strictly increases for increasing region of integration. So we have found an $\bar{\alpha}$ that satisfies Assumption (A) in [103] and the Lemma is proved. \square

1.8.3. Collision Avoidance Based on Non-Overlapping eRPPs

In this section we will introduce a coupling constraint \mathcal{C}_{IN} similar to the constraints for the exact approaches based on Whittle's Chebychev inequality. The constraint \mathcal{C}_{IN} will guarantee that the empirical RPPs R_{IN}^1 and R_{IN}^2 of two agents do not overlap. We already showed in the previous section that the optimal empirical RPPs will converge against optimal RPPs determined by the stochastic program $\mathbb{P}[\text{oRPP}]$ when the number of samples goes to infinity. It will follow that controls for two agents that satisfy constraint

\mathcal{C}_{iN} on the empirical RPPs will converge against optimal controls that satisfy the constraint that the optimal RPPs do not overlap. When the optimal RPPs do not overlap, we can use an argumentation as in Proposition 1.3 to conclude a bound on the probability of a collision of two agents.

First, we will formulate the constraint \mathcal{C}_{iN} . As before in this section we will consider constraints on the agents at a time step $t \in \{1, \dots, T\}$ within the planning horizon.

Definition 1.15 (Constraint \mathcal{C}_{iN}) The empirical RPPs R_{iN}^1 and R_{iN}^2 for a sample number $N > 0$ have a distance that is greater than or equal to ε . More precisely, if

$$\alpha_N^i = [\alpha_{iN}^i(1), \dots, \alpha_{iN}^i(d)]^\top \quad (1.166)$$

denote the optimal size parameters of the empirical RPPs for agents 1, 2, constraint \mathcal{C}_{iN} requires that for the means μ_t^1 and μ_t^2 of the agents it holds that

$$\|\mu_t^1(i) - \mu_t^2(i)\| \geq \varepsilon + \alpha_{iN}^1(i) + \alpha_{iN}^2(i) \quad (1.167)$$

for at least one $i \in \{1, \dots, d\}$.

Remark 1.10 This constraint is transformed into MBL constraints on the means analogously to the constraint formulation in Section 1.7.3 in arbitrary dimension d . We therefore omit the explicit formulation here for the sake of brevity.

Convergence Properties

Now we will show that controls that satisfy constraint \mathcal{C}_{iN} will converge against controls that are feasible for the constraint on the probability of inter-agent collisions.

Consider the auxiliary programs

$$\text{PR} : \min_{u_{1:T}^1, u_{1:T}^2} h(u_{1:T}^1) + h(u_{1:T}^2) \quad (1.168)$$

$$\text{s.t. } u_{1:T}^i \in F_t^i, \quad i = 1, 2 \quad (1.169)$$

$$\exists j \in \{1, \dots, d\} : |\mu_t^1(j) - \mu_t^2(j)| \geq \varepsilon + \alpha_t^1(j) + \alpha_t^2(j) \quad (1.170)$$

$$\mu_{t+1}^i = \mathbf{A}\mu_t^i + \mathbf{B}u_t^i, \quad i = 1, 2, t = 1, \dots, T-1 \quad (1.171)$$

where $1 \leq t \leq T$ is an arbitrary time step and R_t^1 and R_t^2 are optimal RPPs for γ_t^1 and γ_t^2 with $\gamma_t^1 + \gamma_t^2 = \gamma_t^{1,2}$ and the empirical counterpart

$$\text{PR}_N : \min_{u_{1:T}^1, u_{1:T}^2} h(u_{1:T}^1) + h(u_{1:T}^2) \quad (1.172)$$

$$\text{s.t. } u_{1:T}^i \in F_t^i, \quad i = 1, 2 \quad (1.173)$$

$$\exists j \in \{1, \dots, d\} : \|\mu_t^1(j) - \mu_t^2(j)\| \geq \varepsilon + \alpha_{tN}^1(j) + \alpha_{tN}^2(j) \quad (1.174)$$

$$\mu_{t+1}^i = \mu_t^i + \mathbf{B}u_t^i, \quad i = 1, 2, t = 1, \dots, T-1 \quad (1.175)$$

where R_{tN}^1 and R_{tN}^2 are empirical RPPs.

These programs can be understood as simplification of the MA-MPC problem when there are only two agents. The constraint is that the RPPs have a distance greater than or equal to ε at time step t . For simplicity reasons we consider in PR only the coupling constraint at one time step $1 \leq t \leq T$, however, an extension to coupling constraints in all time steps is straightforward.

We will show that controls that are optimal for problem PR_N will converge against controls that are optimal for PR. Controls that are feasible for

PR are feasible for the program

$$\text{SP}_t : \quad \min_{u_{1:T}^1, u_{1:T}^2} h(u_{1:T}^1) + h(u_{1:T}^2) \quad (1.176)$$

$$\text{s.t. } u_{1:T}^i \in F_t^i, \quad i = 1, 2 \quad (1.177)$$

$$\mathbb{P}\{(x_t^2, x_t^1) \notin F^c \mid (u_{1:T}^1, u_{1:T}^2)\} \leq \gamma_t^{1,2} \quad (1.178)$$

$$(1.179)$$

which is a similar simplification of MA-MPC as we already made in Section 1.6.2 with the only difference that we only consider the collision avoidance constraint at one time step t .

Lemma 1.4 *Let φ_N denote the optimal value and Φ_N the set of optimal solutions of PR_N and let φ denote the optimal value and Φ the set of optimal solutions for PR. Let Assumption 2 hold. Then it holds that $\varphi_N \rightarrow \varphi$ and $d_H(\Phi_N, \Phi) \rightarrow 0$ with probability one as $N \rightarrow \infty$. Furthermore it holds that a feasible solution of PR is also feasible for SP_t , i.e., if controls for two agents satisfy the constraint that $d(R^1, R^2) \geq \varepsilon$ then they also satisfy $\mathbb{P}\{(x_t^2, x_t^1) \notin F^c \mid (u_{1:T}^1, u_{1:T}^2)\} \leq \gamma_t^{1,2}$.*

PROOF. The proof again follows a similar argumentation as the proof of [103, Proposition 2.2]. For notational convenience define $\alpha := (\alpha^1, \alpha^2)$, $X := F_u^1 \times F_u^2$, $x = (u_{1:T}^1, u_{1:T}^2)$, $h(x) := h(u_{1:T}^1) + h(u_{1:T}^2)$. Let h^* be the optimal objective value of PR and let h_N^* denote the optimal objective value of PR_N . Define the constraint function

$$f_i(x, \alpha) := |\mu_t^1[u_{1:t}^1](i) - \mu_t^2[u_{1:t}^2](i)| - \alpha^1 - \alpha^2 - \varepsilon, \quad (1.180)$$

where the $\mu_t^1[u_{1:t}^1](i)$ denotes the i -th component of the mean of agent 1 at time step t depending on controls $u_{1:t}^1$. The functions $f_i(x, \alpha)$ are continuous in both x and α as can be easily checked since $\mu_t^1[u_{1:t}^1](i)$ the mean at time

t depending on the controls $u_{1:t}^*$ depends linearly on x . Define

$$f(x, \alpha) := \max_{1, \dots, d} \{f_1(x, \alpha), \dots, f_d(x, \alpha)\} \quad (1.181)$$

then $f(x, \alpha) \leq 0$ is equivalent to the constraint \mathcal{C}_t .

Let x_N be the optimal solution of PR_N . Since $x_N \in X$ and since X is compact, it follows (possibly by passing to a subsequence) that there exists a $x \in X$ such that $x_N \rightarrow x$ for $N \rightarrow \infty$. In Lemma 1.3 we already showed that $\alpha_N \rightarrow \alpha$ for $N \rightarrow \infty$ with probability one. It follows that

$$\lim_{N \rightarrow \infty} f(x_N, \alpha_N) = f(x, \alpha) \quad (1.182)$$

with probability one and in particular

$$f(x, \alpha) = \limsup_{N \rightarrow \infty} f(x_N, \alpha_N) \leq 0. \quad (1.183)$$

So x is feasible for PR and, hence, $h(x) \geq h^*$. It also holds by continuity of h that

$$\lim_{N \rightarrow \infty} h(x_N) = h(x) \quad (1.184)$$

and, hence,

$$\liminf_{N \rightarrow \infty} h(x_N) \geq h^* \quad (1.185)$$

with probability one.

Let x^* be optimal for PR. Since we consider a path planning scenario it is reasonable to assume that there will always be joint controls x that differ from x^* only by an arbitrary small amount and for that the distance of the optimal RPPs has a distance $> \varepsilon$. On an optimization level one can also argue that for a MBL program an optimal solution x^* will “fix” the values of all binary variables. For fixed binary variables the problem is a linear

problem and then it is always possible to find a feasible point arbitrarily close to x^* such that no inequality constraints are active in this point.

So for all $\varepsilon > 0$ there exists a point x with $f(x, \alpha) < 0$ and $\|x^* - x\| < \varepsilon$. Let $\varepsilon := \frac{1}{m}$ and let x_m be a sequence of points that satisfy the above, i.e., $x_m \rightarrow x$ and $f(x_m, \alpha) < 0$. For x_m with $f(x_m, \alpha) < 0$ there exists an $N(m)$ such that for all $N \geq N(m)$ it holds that

$$f(x_m, \alpha_{N(m)}) < 0 \quad (1.186)$$

and, hence, x_m is feasible for $\text{PR}_{N(m)}$. Without loss of generality assume that $N(m) \rightarrow \infty$ for $m \rightarrow \infty$. Since x_m is feasible for $\text{PR}_{N(m)}$ it holds that $h(x_m) \geq h_{N(m)}^*$ and also $h(x_m) \geq h^*$. It follows that

$$\limsup_{m \rightarrow \infty} h_{N(m)}^* \leq \limsup_{m \rightarrow \infty} h(x_m) = h^* \quad (1.187)$$

with probability one.

From inequalities (1.185) and (1.187) it follows that

$$\lim_{N \rightarrow \infty} h_N^* \rightarrow h^* \quad (1.188)$$

with probability one and the first part of the lemma is proved.

The second part of the lemma follows with the same argumentation as in the proof of Proposition 1.2 \square

From the previous lemma the equivalent to Corollaries 1.1 and 1.2 follows: When the chance constraint bounds for each time step are chosen adequately, controls that are optimal for the eRPP constraints will converge against controls that are feasible for the MA-MPC problem with the original chance constraints.

Corollary 1.3 *Let $u_{1:T,N}^1$ and $u_{1:T,N}^2$ be controls for agents 1 and 2 that are*

optimal for the MPC problem with constraints \mathcal{C}_{tN} for $t = 1, \dots, T$ and let

$$\sum_{t=1}^T \gamma_t^{1,2} \leq \gamma^{1,2}. \quad (1.189)$$

Then for $N \rightarrow \infty$ with probability one $u_{1:T,N}^1$ and $u_{1:T,N}^2$ will converge against controls $u_{1:T}^1$ and $u_{1:T}^2$ for that

$$\mathbb{P} \{ (x_{1:T}^1, x_{1:T}^2) \notin F^c \} \leq \gamma^{1,2} \quad (1.190)$$

holds.

1.9. Results

In this section, we consider path planning for multiple UAVs whose movements are affected by stochastic wind disturbances according to the Dryden wind model as specified in [19]. Path planning will be conducted first in two-dimensional space with the assumption that all UAVs fly at the same altitude and then in three-dimensional space where the UAVs can also control their altitude. The task of the UAVs in all considered scenarios is to reach a certain goal point as quickly as possible on a path that is as fuel efficient as possible. Bounds on the control inputs are given by bounds on the maximum acceleration and bounds on the maximum speed of the UAVs.

1.9.1. Model Parameters

We assumed that the UAVs all have the same linear time-invariant motion model given by the double integrator model

$$x_t^i = \mathbf{A}x_{t-1}^i + \mathbf{B}u_{t-1}^i + v_{t-1}^i. \quad (1.191)$$

For planning in two-dimensional state space the system state $x_t^i \in \mathbb{R}^4$ of each UAV was

$$x_t^i = \begin{bmatrix} x_t^i(1) \\ x_t^i(2) \\ \dot{x}_t^i(1) \\ \dot{x}_t^i(2) \end{bmatrix} \quad (1.192)$$

with bounds on the velocity $|\dot{x}_t^i(j)| \leq 45 \text{ ft/s}$, $j = 1, 2$ and control inputs $u_t^i \in \mathbb{R}^2$ being the lateral and longitudinal accelerations

$$u_t^i = [\ddot{x}_t^i(1), \ddot{x}_t^i(2)]^\top, \|u_t^i\|_\infty \leq 12 \text{ ft/s}^2. \quad (1.193)$$

The system matrices were given through

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_2 & s \cdot \mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{0}_2 \\ s \cdot \mathbf{I}_2 \end{bmatrix}, \quad (1.194)$$

where \mathbf{I}_2 is the two-dimensional identity matrix and $\mathbf{0}_2$ the two-dimensional zero matrix and s is the sampling time of the system. For planning in three-dimensional state space the system state $x_t^i \in \mathbb{R}^6$ was

$$x_t^i = \begin{bmatrix} x_t^i(1) \\ x_t^i(2) \\ x_t^i(3) \\ \dot{x}_t^i(1) \\ \dot{x}_t^i(2) \\ \dot{x}_t^i(3) \end{bmatrix} \quad (1.195)$$

with bounds on the velocity $|\dot{x}_t^i(j)| \leq 45 \text{ ft/s}$, $j = 1, 2, 3$. The control inputs $u_t^i \in \mathbb{R}^3$ were the lateral, longitudinal, and vertical acceleration

$$u_t^i = [\ddot{x}_t^i(1), \ddot{x}_t^i(2), \ddot{x}_t^i(3)]^\top, \|u_t^i\|_\infty \leq 12 \text{ ft/s}^2. \quad (1.196)$$

The system matrices were given similarly to the two-dimensional case by

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_3 & s \cdot \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{0}_3 \\ s \cdot \mathbf{I}_3 \end{bmatrix}, \quad (1.197)$$

where again s is the sampling time of the system.

We assumed that the initial positions of the UAVs had Gaussian distribution

$$[x_0^i(1), x_0^i(2)]^\top \sim \mathcal{N}(\boldsymbol{\mu}^i, \mathbf{C}_0^i), \quad (1.198)$$

where $\boldsymbol{\mu}^i$ is the initial starting position of the UAV and the covariance is

$$\mathbf{C}_0^i = \mathbf{I}_2. \quad (1.199)$$

For planning in three-dimensional space the initial positions had Gaussian distribution

$$[x_0^i(1), x_0^i(2), x_0^i(3)]^\top \sim \mathcal{N}(\boldsymbol{\mu}^i, \mathbf{C}_0^i) \quad (1.200)$$

with mean $\boldsymbol{\mu}^i$ the initial starting position and covariance

$$\mathbf{C}_0^i = \mathbf{I}_3. \quad (1.201)$$

We assumed no uncertainty in the initial velocity of the UAVs and the initial velocity was set to zero.

We assumed that the target waypoints are given as two-dimensional or three-dimensional positions Z^i for each UAV. In applications these could for example be given by GPS waypoints. For the objective instead of taking the quadratic deviation of the d -dimensional position part of the state to the goal waypoint, we took the ℓ^1 deviation plus a weighted penalty term for

the control inputs

$$h^i(x_t^i, u_t^i) := \sum_{j=1}^d |x_t^i(j) - Z^i(j)| + \sum_{j=1}^d r |u_t^i(j)|, \quad (1.202)$$

with weight $r = \frac{1}{T}$. The overall control objective over the complete planning horizon is

$$h^i(x_{1:T}^i, u_{1:T}^i) := \sum_{t=1}^T h^i(x_t^i, u_t^i). \quad (1.203)$$

This is a piecewise linear and convex function and, hence, can be formulated as the objective in a MBL program (see [40] for details).

The disturbance samples $v_{t,j}^i$ affecting the UAVs were drawn from the discrete Dryden model to simulate wind turbulence acting on the UAVs [19]. The UAVs were assumed to fly with a maximum speed of 45 ft/s at an altitude below 1000 ft through a field with light turbulence with wind speed of 15 knots at 20 feet height, i.e., we applied the discrete time low-altitude Dryden wind model for light turbulence. The covariances needed for the exact RPP approaches were determined numerically through the sample covariances with a suitably large number of samples. This was done offline before planning. The minimum clearance between the UAVs was set to $\varepsilon = 2$ ft. The MBL solver we used is IBM ILOG CPLEX [57].

1.9.2. Example Scenarios

2D Example With the models detailed above we first evaluated an example for path planning in two-dimensional space with two UAVs. In the scenario considered, the UAVs started at $\mu^1 = [50, 50]$ and $\mu^2 = [25, 50]$ and their respective goal waypoints were $Z^1 = [250, 250]$ and $Z^2 = [50, 250]$. We computed optimal controls with the RPP, transformed RPP, eRPP, and SAA collision avoidance constraints. The chance constraint bound was set to $\gamma^{1,2} = 0.05$ and the horizon length was set to $T = 12$. For the determina-

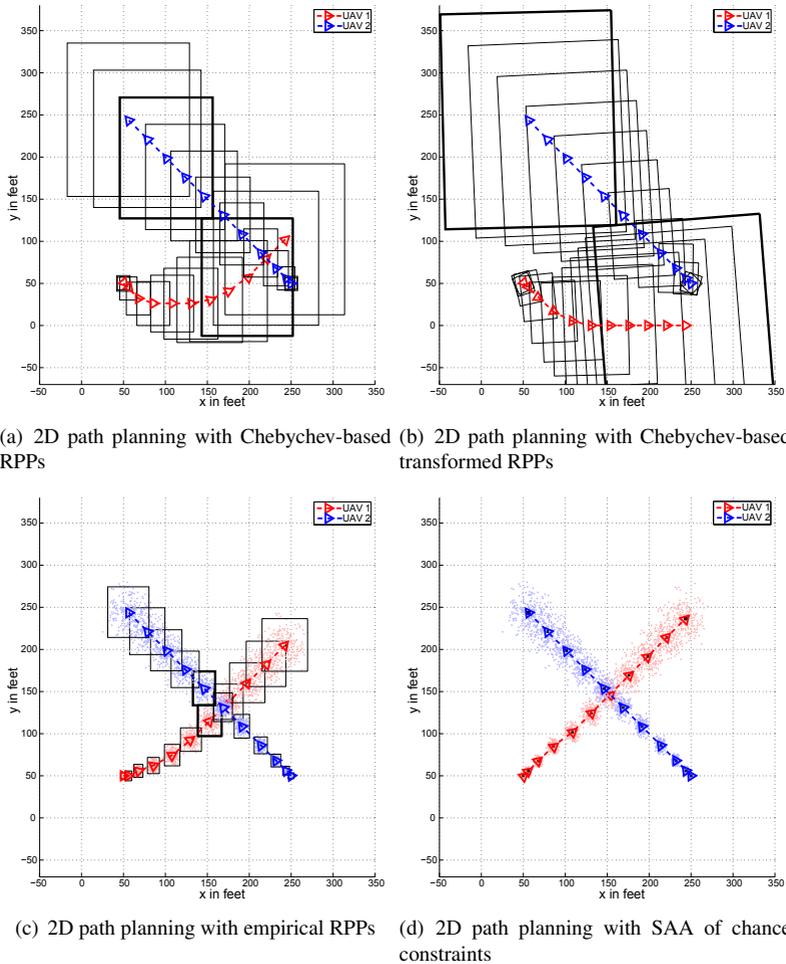


Figure 1.5.: Comparison of path planning in 2D for two UAVs for the Chebychev-based RPPs, transformed RPPs, empirical RPPs and the SAA of the chance constraints of inter-agent collision probabilities. The stronger accentuated regions in (a), (b), and (c) are the ones that lead to constraints that are active in the optimal controls.

1. Motion Planning with Chance Constraints

| | |
|---|----------------------|
| eRPP chance constraint usage | 9.7×10^{-5} |
| RPP chance constraint usage | 0 |
| transformed RPP chance constraint usage | 0 |
| SAA chance constraint usage | 0.0155 |

Table 1.1.: Chance constraint usage of the eRPP, transformed RPP, and SAA controls in the 2D path planning example determined a-posteriori with 10^8 Monte-Carlo samples.

tion of the eRPPs we used $N = 200$ samples for each UAV and for the SAA we used $N = 50$ samples for each UAV. The trajectories resulting from the optimal controls together with the respective rectangular regions for RPP, transformed RPP, and eRPP constraints and the samples for the eRPP and the SAA constraints are depicted in Figure 1.5.

In Figure 1.5(a) and Figure 1.5(b) we depict the trajectories that were determined with the RPP and transformed RPP constraints. The regions accentuated in black are the ones whose constraints on the positions of the UAVs are active, i.e., tight. It becomes apparent that both approaches that are based on the Chebychev inequality lead to constraints that are rather conservative in comparison to the eRPP and SAA constraints: UAV 1 is forced to keep a large clearance to UAV 2 and, hence, needs to make a large detour in order to satisfy the collision avoidance constraints. This detour is especially large for the transformed RPP constraints in Figure 1.5(b). The even higher conservatism of the transformed RPPs in comparison to the normal RPPs can be explained through the fact that in the right hand side of the Chebychev inequality in Equation (1.78) the off-diagonal entry of the covariance \mathbf{C}_{12} appears with a negative sign and therefore tightens the bounds. When the covariance is transformed such that there is no correlation, i.e., $\mathbf{C}_{12} = 0$ this tightening summand disappears and, hence, the inequality becomes more conservative. The trajectory resulting from the optimal controls with the eRPP constraints (depicted in Figure 1.5(c)) is much less conservative and strongly resembles the one determined with the SAA constraints regarding the evasive maneuver UAV 1 has to make in or-

der to avoid UAV 2. In Figure 1.5(d) the trajectories of the UAVs resulting from the SAA constraints is plotted. Here, UAV 1 just makes a very slight evasive maneuver in the third time step in order to let UAV 2 pass first in the seventh time step. Since Figure 1.5(d) is relatively small, a zoomed-in view on the trajectories with time step three for UAV 1 and the seventh time step highlighted is depicted in Figure 1.6. In all scenarios UAV 2 can reach its target waypoint while for the Chebychev-based constraints UAV 1 does not even come close to its target. For the eRPP and SAA constraints also UAV 1 is able to approach its target.

For all controls we computed *a-posteriori* the empirical chance constraint usage by drawing 10^8 Monte-Carlo disturbance samples from the Dryden wind model and evaluating the empirical collision probability of the trajectories. The results of this analysis are given in Table 1.1 and it can be seen that the controls computed with the Chebychev-based RPP constraints have such a high degree of conservatism that the controls have an actual collision probability of zero. The eRPPs constraints also have some degree of conservatism compared to the SAA constraint but this is by far not as strong as for the other two constraints. Also from the *a-posteriori* analysis it turns out that the complete collision “risk” of the UAVs in the SAA is in the seventh time step, where they come closest. Still, even the SAA does not make use of all of allowed “risk” given through the chance constraint bound. We expect that this stems from the fact that we only used $N = 50$ samples to approximate the agents’ states and for more samples in the approximation the usage will be higher. However, computation of the depicted solution for the SAA took around 24 minutes while the eRPP solution was computed in 0.028 seconds on a standard office PC and using a higher sample number will increase the runtime of the SAA even further.

3D Example Next, we considered an example scenario for motion planning in three-dimensional space with two UAVs in which the UAVs additionally were allowed to adjust their altitude. The starting positions for the

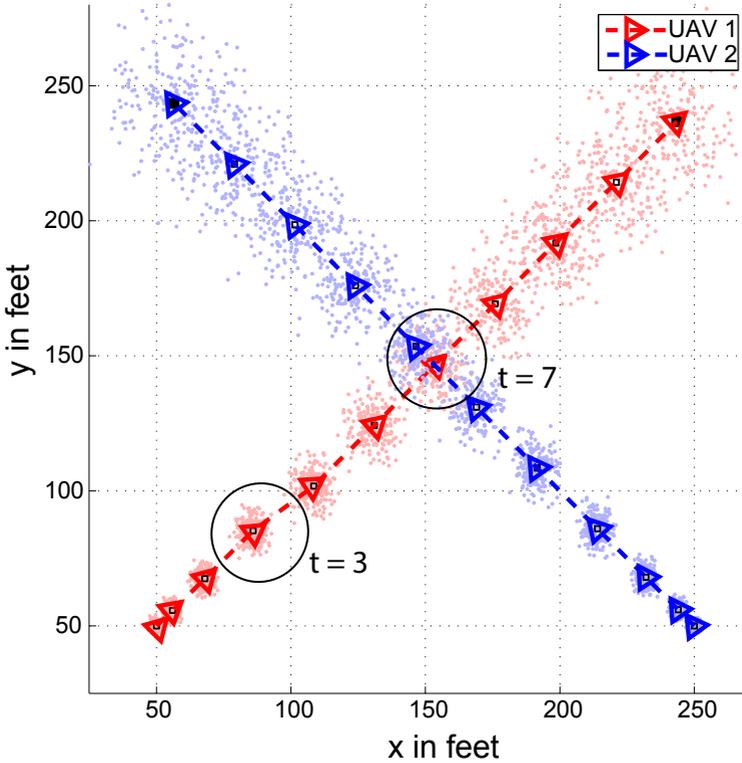
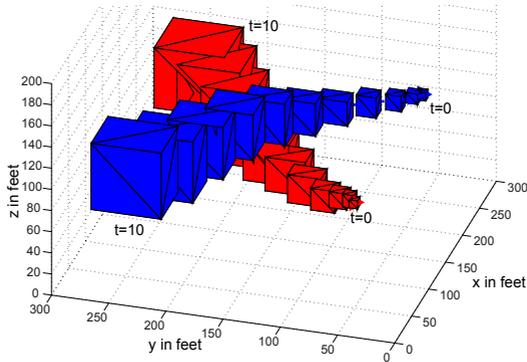


Figure 1.6.: Zoomed-in view of 2D UAV trajectories for the SAA. Time step $t = 3$ is highlighted for UAV 1, since here it makes a very small evasive maneuver in order to avoid a collision with UAV 2 in time step $t = 7$. The a-posteriori collision probability is zero in all time steps except for $t = 7$, where the value given in Table 1.1 is reached.

UAVs were $\mu^1 = [50, 50, 100]$ and $\mu^2 = [250, 50, 100]$ and the goal waypoints were $Z^1 = [250, 250, 100]$ and $Z^2 = [50, 250, 100]$. The time horizon was again set to $T = 10$ and the chance constraint bound was set to $\gamma_t^{1,2} = 0.05$. Due to the high computational complexity and high memory requirements of the SAA formulation, we only considered planning with the transformed RPP and eRPP constraints. In Figure 1.7 we depict two



(a) 3D path planning with eRPP constraints.

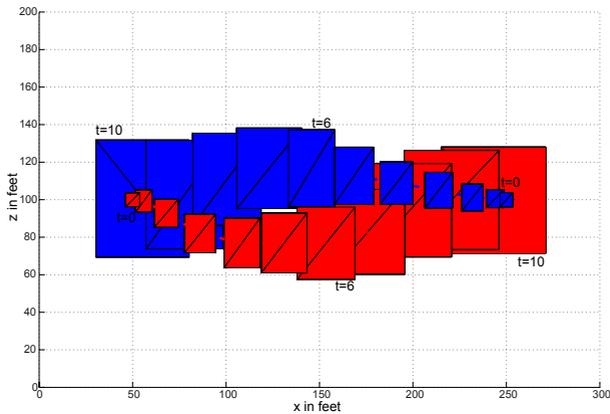
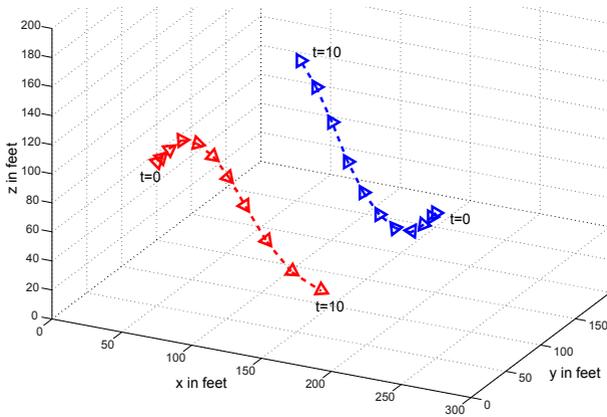
(b) 3D path planning with eRPPs side view. The eRPP constraints are active at time step $t = 6$. From this view one can see that the UAVs not only avoid each other in the horizontal plane but also adjust their altitude in order to avoid a collision.

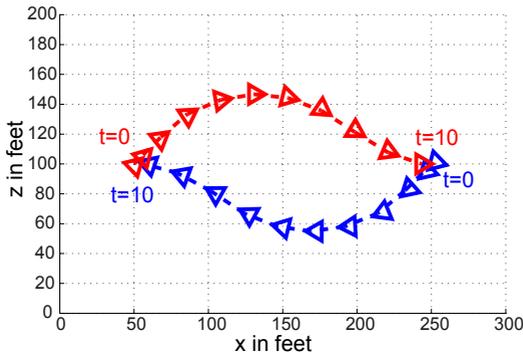
Figure 1.7.: 3D path planning example with two UAVs with eRPP constraints. Depicted are the eRPPs and the eRPPs of UAV 1 are in red and the ones of UAV 2 in blue. The time instance the respective eRPP relates to is written next to the box for times $t = 0$ and $t = 10$ in order to improve the comprehensibility of the plots.

different views of the trajectories of the UAVs resulting from controls that were optimal with the eRPP constraints. Here, the collision avoidance con-

1. Motion Planning with Chance Constraints



(a) 3D path planning with transformed RPP constraints.



(b) 3D path planning with transformed RPP constraints (side view). The respective time step related to the mean position is depicted in the corresponding color.

Figure 1.8.: 3D path planning example with two UAVs with the transformed RPP constraints. Here, we only depict the planned mean trajectories of the UAVs since the transformed RPPs became very big and, hence, their inclusion would have made the image too cluttered. From the mean trajectories it can be seen however, that the UAVs need to make large evasive maneuvers that result in large detours and the UAVs ending far from their target waypoints.

| | Chance constraint usage | Runtime in seconds |
|-----------------|-------------------------|--------------------|
| eRPP | 0.00011562 | 0.221 |
| transformed RPP | 0 | 5.186 |

Table 1.2.: Chance constraint usage of the eRPP and transformed RPP controls in the 3D path planning example determined a-posteriori with 10^8 Monte-Carlo samples and runtime of the optimization routine.

straints were active at time $t = 7$ and for better comprehensibility we also attached the initial and final time step to the respective regions. Because the size of the regions became large, the constraints resulting from the transformed RPP became so conservative that it was difficult to tell anything from a plot of the trajectories of the UAVs together with the transformed RPPs. Therefore we only depict the mean trajectories resulting from controls optimal with the transformed RPP constraints in Figure 1.8.

We again conducted an a-posteriori analysis of the actual chance constraint usage of the controls by drawing 10^8 Monte-Carlo samples from the Dryden wind model and computing the collision probability of the controls. The results are depicted in Table 1.2 and strongly resemble the results for the two-dimensional example in the sense that the transformed RPPs are so conservative that the collision probability is zero and the conservatism of the eRPP is much lower. In general, when the user specifies a chance constraint bound greater than zero, it is desirable that the UAVs use as much of this granted “risk” as possible in order to find controls that are more optimal (i.e. result in shorter, more fuel efficient paths). Therefore an actual chance constraint usage of zero is undesirable and in this regard the eRPP constraints perform well since they actually make use of the positive chance constraint bound. Also in Table 1.2 we depict the runtimes of the CLPEX optimization routine with the corresponding constraints. It can be seen that the runtime with the eRPP constraints is of order of magnitude faster than the transformed RPP approach.

| | |
|--------------------------------|--|
| Time horizon | $T = 10$ |
| Chance constraint bounds | $\gamma_i^{i,j} = 1\%, 2.5\%, 5\%$ |
| Number of samples | $N = 25, 50, 100$ |
| Time limit for CPLEX for SAA | <code>cplex.tilim = 600 s</code> |
| Integrality gap for CPLEX | <code>cplex.epgap = 0.01</code> |
| Random starting positions | $\mu^i \sim \mathcal{U}([0 \ 50] \times [0 \ 170])$ |
| Random target waypoint | $Z^i \sim \mathcal{U}([450 \ 500] \times [0 \ 170])$ |
| PC and OS used for simulations | Intel Pentium Q9300 2.5 GHz, 8 GB RAM, Windows Server 2008 R2 |

Table 1.3.: Parameters used for quantitative simulations.

1.9.3. Quantitative Results

Additionally to the illustrative examples we conducted extensive numerical simulations with randomly generated two-dimensional scenarios. In these scenarios the starting positions and goal waypoints for UAV path planning in two-dimensional space were randomly drawn from certain areas in order to encourage potential conflicts or the possibility of a collision. The starting position were drawn in such a way that there are no collisions at time $t = 0$, i.e., such that the MA-MPC problem is feasible. Since the starting and goal positions were randomly drawn and, hence, sometimes many conflicts occurred and sometimes only very little, an a-posteriori evaluation of the chance constraint usage of the controls as in the previous examples turned out to be not very informative. We therefore evaluated and compared the conservatism of the different collision avoidance methods through the overall cumulative objective of the controls, i.e., the higher the objective, the longer detours the UAVs had to make, the more conservative the controls. The time horizon was set to $T = 10$ in all simulations.

The solver CPLEX provides the possibility to manually set the relative mixed-integer programming (mip) gap tolerance via the parameter `cplex.epgap`. This parameter is defined as follows: When CPLEX solves a general MILP with branch and cut methods (for algorithmic details please see [100]) it builds a tree with the linear relaxation of the original problem

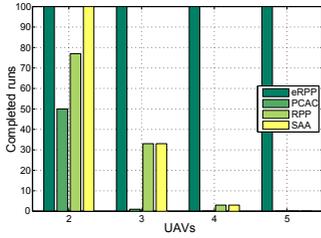
at the root and subproblems resulting from the problem at the parent node with further linear inequalities (so-called cuts) added as nodes. Let J_{node}^* be the best objective function achieved in a node and let J_{mip}^* be the best objective function with MBL variables, i.e., without any relaxations. Then the relative mip gap tolerance is defined as

$$\text{epgap} := \frac{|J_{node}^* - J_{mip}^*|}{10^{-10} + |J_{mip}^*|}. \quad (1.204)$$

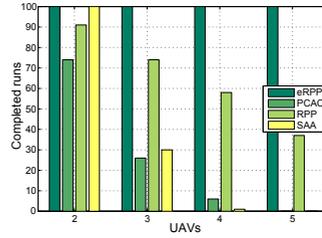
The default value for the gap is 10^{-4} and we set it in our simulations to 0.01. This has the advantage that the optimization routine will stop prematurely since often CPLEX is able to find a good integer solution very quickly but does not have a “proof” of its optimality, i.e., there does not yet exist a node that has the same objective and, hence, would have to investigate further nodes. By setting a slightly higher gap parameter, CPLEX is allowed to abort optimization earlier when the objective of the best integer solution lies within the tolerance of the objective of the best node. For all the parameters of the simulations, please see Table 1.3.

Runtime We compared the SAA, RPPs, transformed RPPs and eRPPs regarding runtime of the MBL optimization routine and the objective of the solutions. Due to time constraints, we set an upper limit of 600 s for the optimization for the SAA by setting the CPLEX `timlim` parameter to 600. After this time CPLEX will return the best feasible solution it has found so far and nothing if it has not found any feasible solutions. In Figure 1.9 we depicted for how many random runs out of 100 CPLEX was able to find a feasible solution for increasing numbers of UAVs and $N = 25$ samples. The low completion rates for UAV numbers higher than two for the SAA follows because of the time limit and it can be seen that already for low numbers of samples it became difficult to find a feasible solution within the time limit. The low completion rate of planning with the RPP-based approaches stems from the fact that their constraints can become so conservative, that it is

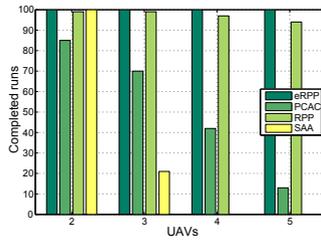
1. Motion Planning with Chance Constraints



(a) Completions for $\gamma_t^{i,j} = 0.01$ and $N = 25$ samples



(b) Completions for $\gamma_t^{i,j} = 0.025$ and $N = 25$ samples



(c) Completions for $\gamma_t^{i,j} = 0.05$ and $N = 25$ samples

Figure 1.9.: Depicted is in how many runs out of 100 CPLEX found an optimal solution for the eRPPs, transformed RPPs (PCAC), and RPPs, or was able to return a feasible solution within 600 s for the SAA for increasing numbers of UAVs and $N = 25$ samples. Lower completion rates of the approaches based on the multivariate Chebychev inequality (RPP and PCAC) stem from the fact that the conservatism causes the problem to become infeasible, while the low completion rates for the SAA stem from long runtimes.

not possible to find any feasible control given the motion constraints of the UAVs. This means that the collision avoidance constraints demand such a high clearance between the UAVs that given the motion constraints of the UAVs they were not able to find feasible trajectories. Especially for low chance constraint bounds like $\gamma_t^{i,j} = 0.01$ this causes problems, while for higher bounds ($\gamma_t^{i,j} = 0.05$) less problems occur for lower UAV numbers. For the eRPP-based problems a solution was always found since by con-

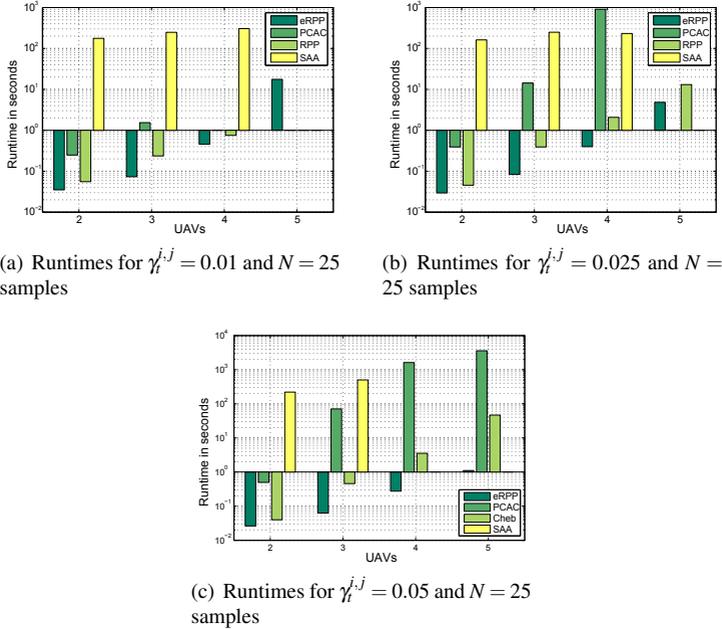
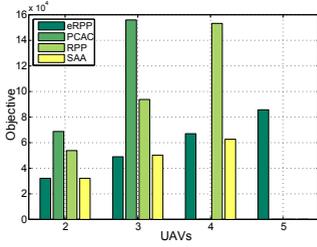


Figure 1.10.: Depicted are the runtimes of the eRPP, transformed RPP (PCAC), RPPs, and the SAA until an optimal solution was found. The runtimes depicted here are only the ones for problem instances that were feasible (RPP, PCAC) or for that an optimal solution was found within the set time limit (SAA). The times are averaged over these runs. Please note that in all figures the vertical axis is in log-scale.

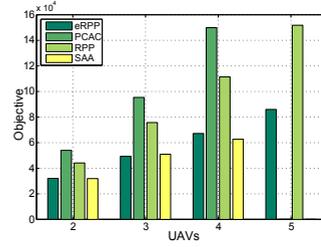
struction they alleviate the computational complexity of the SAA and the conservatism of the other Chebychev-based RPPs.

In Figure 1.10 we depict the average runtime of CPLEX for the different chance constraint bounds. Here, we only show the runtime of the runs for that either a feasible solution was found within the time limit (SAA) or there exists a feasible solution (Chebychev-based RPPs). Even for the runs in which the SAA was able to find an feasible solution within 600 seconds the runtime was still relatively high with average runtime around

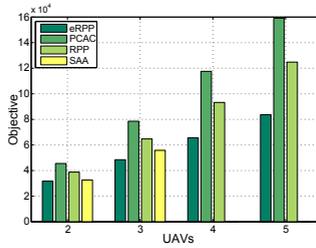
1. Motion Planning with Chance Constraints



(a) Optimal objective for $\gamma_i^{j,j} = 0.01$ and $N = 25$ samples



(b) Optimal objective for $\gamma_i^{j,j} = 0.025$ and $N = 25$ samples



(c) Optimal objective for $\gamma_i^{j,j} = 0.05$ and $N = 25$ samples

Figure 1.11.: Depicted are the averaged values of the objective of the optimal controls or the best feasible controls returned after 600 s (SAA).

100 seconds. The Chebychev-based RPP approaches showed low runtime for low UAV numbers ($M = 2, 3$) in the range of one second. However, for the transformed RPPs for more than three UAVs either no feasible solutions could be found at all or it took very long to find one.

For the eRPPs, before the actual MPC problem could be solved, for each agent the MBL problem corresponding to $P[\text{eRPP}]$ had to be solved for each agent in order to determine the optimal size of the eRPPs. The runtime of these MBLs were added to the runtime of the MPC problem with eRPPs and in Figure 1.10 the accumulated times are depicted. The average runtime of the eRPPs for less than five UAVs are significantly below one second and only increase for five UAVs.

Optimality In Figure 1.11 we depict for the runs that found feasible controls within the time limit the objective of the best controls found averaged over the completed runs. Since the SAA is the most direct approximation and optimal controls for the SAA will converge against controls that are optimal with the true chance constraints, one would expect that its objective should be the best. However for example in Figure 1.11(c) it can be seen that for three UAVs the eRPP actually achieved a better objective. This can be explained since the objective depicted does not need to be the objective of the optimal controls but is the objective of the best controls found within the time limit. This means that within 600 seconds CPLEX was not able to find a solution with better objective. From Figure 1.11 one can see that for the approaches based on the Chebychev inequality (RPP and PCAC) the objective is worst and, hence, the degree of conservatism is high. Also note that for smaller chance constraint bounds $\gamma_i^{i,j}$ the Chebychev inequality becomes more conservative and the objective even worsens. The eRPP constraints achieved good objectives that are comparable and not much worse than those achieved by the SAA and we conclude that their degree of conservatism is comparatively low.

Further Results So far the simulations indicated that the performance of the eRPPs in comparison with the SAA with regard to runtime is way better with only slight losses in optimality and is also better in comparison with the Chebychev-based RPP approaches. As we mentioned earlier, one disadvantage the eRPPs can have, is that in order to determine the optimal size of the eRPP for each agent the additional MBL problem $P[\text{eRPP}]$ has to be solved. The complexity of this additional MBL depends on the number of samples drawn from the uncertain states of the UAVs. In the quantitative simulations due to the slow runtimes of the SAA we have so far only considered $N = 25$ samples. This is a very low number and can hardly produce representative results for the construction of the eRPPs (e.g. in $P[\text{eRPP}]$ for $N = 25$ and the chance constraint bounds used in these simulations all

1. Motion Planning with Chance Constraints

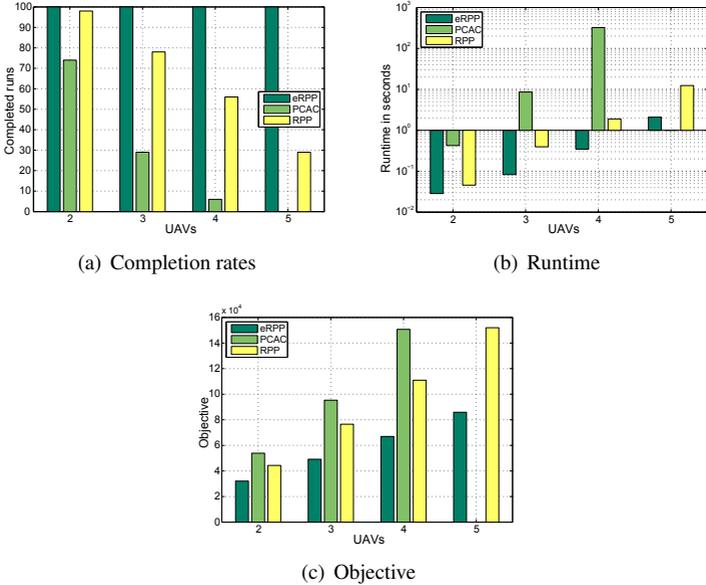


Figure 1.12.: Completion rate, runtime, and objective with $N = 50$ samples and chance constraint bound $\gamma_i^{j,j} = 0.025$ for the eRPP, transformed RPPs (PCAC) and RPP approaches. Due to the high runtime of the SAA it was omitted here.

samples need to lie within the region and none are allowed to lie outside). Therefore we additionally compared the performance of the eRPPs with the RPPs and the transformed RPPs for higher sample numbers.

In Figure 1.12 we depict the completion rates (Figure 1.12(a)), the runtimes (Figure 1.12(b)) and the optimal objective achieved on the runs that were feasible (Figure 1.12(c)) for the eRPPs, the transformed RPPs (PCAC), and the RPPs for 100 randomly generated scenarios and $N = 50$ samples for each agent. Further, in Figure 1.13 we depict the completion rates (Figure 1.13(a)), the runtimes (Figure 1.13(b)) and the optimal objective achieved on the runs (Figure 1.13(c)) also for the eRPPs, the transformed RPPs (PCAC), and the RPPs for 100 randomly generated scenarios

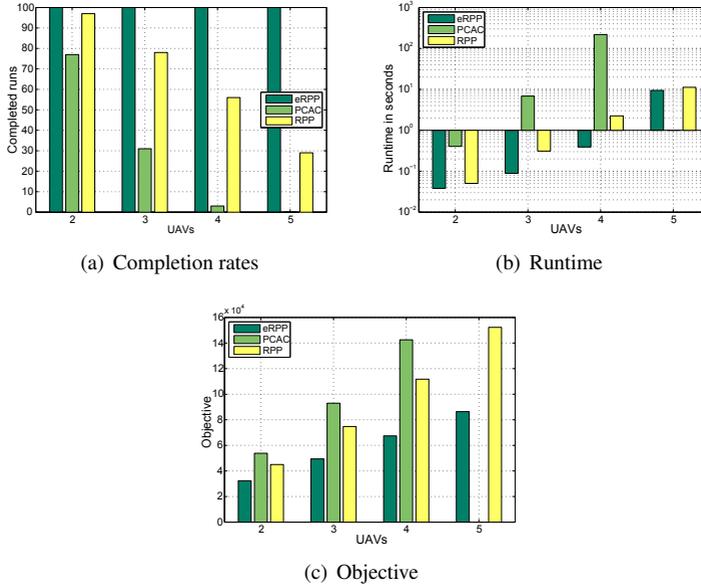


Figure 1.13.: Completion rate, runtime, and objective with $N = 100$ samples and chance constraint bound $\gamma_i^{i,j} = 0.025$ for the eRPP, transformed RPPs (PCAC) and RPP approaches. Due to the high runtime of the SAA it was omitted here.

and $N = 100$ samples for each agent.

In Figure 1.12(b) and Figure 1.13(b) it becomes apparent that doubling the sample numbers from $N = 50$ to $N = 100$ for one to four UAVs has only very little impact on the runtime of the eRPPs. Only for $M = 5$ UAVs an increase in runtime from around 20 seconds to 90 seconds can be observed. Please recall here that in all runtime plots the time of P[eRPP] to find the optimal eRPPs was added to the runtime of the actual MPC problem. The results regarding the achieved optimal objective are as in the previous simulations, i.e., with the eRPPs a much better objective can be achieved from which we conclude that their degree of conservatism for collision avoidance is much lower.

1.10. Conclusions

This chapter was concerned with model predictive control (MPC) for motion planning in a multi-agent system with chance constraints on the probability of collisions of agents. In order to make the non-convex constraints on multivariate integrals that result from the chance constraints on the probability of a collision of agents numerically tractable we first proposed a sample-based approximation of the collision probabilities. We proved convergence of controls that are optimal with the sample-based approximation of the constraints against controls that are optimal for the MPC problem with the original chance constraints.

Since the numerical complexity of the sample-based approximation is still relatively high, we proposed an efficient and conservative reformulation of the chance constraints on collision probabilities. It is based on the concept of confidence regions of increased probability of presence (RPP). These are regions for each agent for that the probability that the agent is outside of the region is below a given threshold. The chance constraints on collision probabilities were then replaced by the constraint that the regions for different agents do not overlap. We presented two different methods for determining the regions: The first is based on a multivariate generalization of the well-known Chebychev inequality and the second is based on a sample-based empirical determination of the RPPs. We proved that controls found with the Chebychev based RPP constraints are feasible for the original MPC problem with chance constraints on collision probabilities and that controls found with the empirical RPPs converge against feasible controls for the original problem. In extensive simulations we evaluated and compared the different approaches proposed in this chapter.

Strengths The two major strengths of the approximations and reformulations of the chance constraints presented in this chapter are that only very little assumptions and prior knowledge about the random variables describ-

ing the agents uncertain states is required and that the resulting constraints can be formulated as mixed-binary linear constraints.

For the sample-based approximation the only requirement on the occurring random variables is that it is possible to draw samples from them. For the RPP based reformulations either knowledge of the second central moment, the covariance, is required which is a very general attribute of random variables and for example can be estimated through sample-based methods, or again one needs to be able to draw samples from them for the empirical RPPs. So even if the disturbance models are highly complex and the occurring random variables have probability distributions that cannot be given in closed form, only samples need to be generated from the disturbances. Computer-assisted random number generation and simulation of stochastic processes is a well-studied problem in applied mathematics and there are several methods in the literature for generating samples from random variables, for details see [84] and [68] and the references therein.

The second strength is the formulation of the constraints resulting from the approximations of the chance constraints as mixed-binary linear constraints. This enables the user to employ powerful and dedicated commercial solver software for finding optimal solutions of the MPC problem². There is no need for the user to care about actually solving the underlying optimization problem but only has to provide the models and the constraints to the solver software. Also these solvers come with high-level modelling languages like IBM OPL or AMPL with which the mixed-binary linear constraints can be implemented in almost the same format as in the mathematical formulation without any expert knowledge about linear optimization. Hence, the proposed approximations stand out due to their high degree of general applicability and ease of use and implementation.

²Examples for such solver software are IBM ILOG CPLEX [57], Gurobi (<http://www.gurobi.com/>) or MOSEK (<http://www.mosek.com/>) just to name a few.

Current Limitations The RPP based collision avoidance constraints are very general, distribution independent and numerically efficient. However their generality comes at the price of a certain degree of conservatism of the resulting controls, i.e., the resulting collision avoidance constraints can become too conservative and force the agents to keep a safety clearance that is too cautious and, hence, controls become unnecessarily suboptimal. The empirical RPPs remedy most of the conservatism but there is still some conservatism remaining when using them. The user is faced with the trade-off between generating possibly suboptimal controls very fast and efficiently with the RPP approach or determining controls with the SAA that are guaranteed to converge against optimal controls but at the cost of increased runtime.

In general, mixed-integer linear programs with input data of moderate size can be solved very fast and efficiently. However since theoretically solving them is NP-hard, certain problem instances and starting from a certain size of the problem the runtime until an optimal solution is available can become very long as we saw in the numerical experiments. This means that for large multi-agent systems suboptimal and distributed solution techniques need to be employed in order to alleviate the high runtimes. For example [87] considers a robust and distributed coordination approach to find a suboptimal joint solution for all agents and these techniques could also be applied to MPC with the RPP constraints. In [78] distributed techniques for solving mixed-integer linear programs in a multi-agent system based on Gomory cuts were proposed and these approaches could also be applied.

Applications The algorithms for chance constrained collision avoidance proposed in this chapter rely on only very general assumptions regarding the disturbances and, hence, they are applicable in numerous different scenarios. Possible applications are motion planning in multi-robot, multi-UAV, multi-vehicle and multi underwater vehicle scenarios. Further ap-

plications could be the generation of trajectories for motion planning for multiple robotic end effectors with collision avoidance constraints, for example for automatic line operation or motion planning for robot arms of a humanoid robot. Also the scenario from Example 1.1 for air traffic control (ATC) at an airport in which an ATC center has to plan collision free trajectories for multiple aircraft under wind disturbance is a potential application for the algorithms provided in this chapter.

2. Mixed-Integer Random Convex Programs

2.1. Introduction to MI-RCPs

A random convex program (RCP) is a continuous convex optimization problem with linear cost objective, subject to N randomly drawn convex constraints $f(x, \delta^{(j)}) \leq 0$ for $j = 1, \dots, N$. Here the $\delta^{(j)}$ are independent and identically distributed realizations of a random vector δ . Since the constraints of the convex optimization problem are randomly drawn, the optimal objective J^* and the optimal solution if it exists are random variables. The most notable property of an RCP is that the optimal objective will remain optimal with a high probability for the next “unseen” constraint realization. More precisely, if an additional constraint realization is drawn from δ after the optimization, with high probability J^* will *generalize* and still be the optimal objective value for the new constraint.

For RCPs with continuous decision vectors, [45, 46] initially provided bounds on the tails of the distribution of the probability that an optimal solution found under N random constraints will become infeasible for the next randomly drawn constraint. These bounds were refined in [49] under the restriction that the RCP is feasible with probability one for all random constraints. The work [44] lifted this restriction and proved bounds without any assumptions on feasibility. Further it extends previous results to the case that some of the random constraints may be violated with the goal of further improving the optimal objective.

In this chapter, we extend existing work on continuous RCPs to the case of random programs with mixed-integer decision vectors lying in the intersection of $\mathbb{R}^n \times \mathbb{Z}^d$ and random convex constraints. We prove bounds on

the probability that the optimal mixed-integer objective J^* does not generalize. The class of mixed-integer optimization problems is clearly not convex in the usual sense, since the domain has discrete components, however it seems appropriate to name them *mixed-integer random convex programs* (MI-RCP) to highlight the fact that the feasible set is the intersection of $\mathbb{R}^n \times \mathbb{Z}^d$ with a convex set. Unfortunately, unlike in the continuous case where the equivalent of the bounds on the tails of the violation probability depends linearly on the dimension of the decision variable, in the mixed-integer case the bounds depend geometrically on the number of discrete decision variables. A consequence of the geometrical dependence is that more samples are needed in order to immunize an MI-RCP against uncertainty.

Therefore, in the second part of this chapter we aim to improve and tighten on the probabilistic bounds. Here, we consider the special case of mixed-integer random linear programs (MI-RLPs) i.e., MI-RCPs in which the constraint function is either affine in the decision variable x or piecewise affine and convex. In this case the feasible region of the continuous relaxation of the MI-RLP is a polytope with finitely many vertices. We derive bounds on the probability that an arbitrary vertex of the feasible region of the linear relaxation or even a convex combination of vertices becomes infeasible under the next constraint realization. The optimal integer solution will in general *not* coincide with the optimal solution of the relaxation and can be located anywhere in the feasible region of the relaxation. However, any point in the feasible region of the relaxation is the convex combination of its vertices and, hence, we can utilize the former results to bound the probability that the integer solution becomes suboptimal under the next constraint. Parts of the results on constraint sets of mixed-integer convex problems in Section 2.2 and on general nonlinear MI-RCPs in Section 2.3 are to be published in [7]. All results on mixed-integer random linear programs in Section 2.4, Section 2.5, and all numerical examples in Section 2.6 are novel to this thesis.

There are several other approaches to make optimization robust to uncertainty. The *robust convex optimization* approach [24,65] finds a solution to a convex problem that is robust to all uncertainty realizations bound to lie in a given bounded uncertainty set (for more details see Section 2.5.1). *Chance-constrained* approaches (as we also considered in the previous chapter of this thesis) assume that there is a probability measure over the uncertainty set and try to find an optimal solution that satisfies uncertain constraints with a guaranteed high probability, see [119] and the references therein.

In many real-world applications integer decision variables occur, for example when they are used to model decisions on indivisible resources like in the well-known Knapsack or Traveling Salesman problems [100] or multi-agent problems. Uncertainty in integer problems has been studied in the literature before: The works [30,31,69] are concerned with complexity and efficient solvability of robust linear integer problems with uncertainty in both the cost vector and the constraint matrices. [32] deals with the application of the theory developed in [31] to the optimal control of a supply chain subject to random demand, which results in a robust mixed-integer linear problem. In [42] the authors consider the optimal portfolio selection problem with integer decisions variables to model stock market restrictions and chance constraints on the expected returns.

This chapter is structured as follows. In Section 2.2 we first study properties of constraint sets of mixed-integer deterministic convex problems. In Section 2.3 we give a formal definition of mixed-integer RCPs and of their violation probability and we derive the first bounds on the violation probability. In Section 2.4 mixed-integer random linear programs are studied and a different bound on the violation probability is derived. In Section 2.5 these bounds are applied to random linear programs with uncertain right hand side, i.e., random linear programs in which the uncertainty only is in the right hand side and not in the constraint coefficients. In Section 2.6 we apply the results of this chapter to a Knapsack problem with disturbance acting on the weights of the items, an inventory problem with uncertain

demands, and a multi-agent problem.

2.2. Preliminary Results on Mixed-Integer Problems

In this section we consider mixed-integer “convex” optimization problems of the form

$$\begin{aligned} \text{P}[K] : \min_{x \in \Omega} c^\top x \\ \text{s.t. } f_j(x) \leq 0, \quad j \in K \end{aligned} \tag{2.1}$$

with a linear objective $c \neq 0$ and decision variable x confined to lie within a compact domain Ω that will either be a subset of \mathbb{Z}^d , $\mathbb{R}^n \times \mathbb{Z}^d$ or \mathbb{R}^n . The constraints are given by convex and lower-semicontinuous functions $f_j: \mathbb{R}^m \rightarrow \mathbb{R}$ (where either $m = d$, $n + d$, or n) that are indexed over the finite set K . We denote the optimal solution of $\text{P}[K]$ by $x^*(K)$ if it exists and the optimal objective by $J^*(K)$. If $\text{P}[K]$ is infeasible, we set $J^*(K) = \infty$. Denote by $\text{Sat}(K)$ the feasible set of $\text{P}[K]$, i.e., $\text{Sat}(K) := \{x \in \Omega: f_j(x) \leq 0, j \in K\}$.

Definition 2.1 (Support Constraints) A constraint $j \in K$ is a *support constraint* of $\text{P}[K]$ if $J^*(K \setminus j) < J^*(K)$, i.e., if the optimal objective strictly improves when constraint j is removed from $\text{P}[K]$. Denote the set of support constraints of $\text{P}[K]$ by $\text{Sc}(K) \subset K$.

The following definition of convex sets in mixed-integer spaces will conclude this section.

Definition 2.2 (M -convexity) Let $M \subset \mathbb{R}^m$ be a closed set. We say that a subset $C \subset M$ is *M -convex* if there is a convex subset \bar{C} of \mathbb{R}^m such that $C = \bar{C} \cap M$.

For example, on the integer lattice \mathbb{Z}^d , M -convex sets are the intersection of standard convex sets in \mathbb{R}^d with the integer lattice.

2.2.1. Support Constraints on the Integer Lattice

In this section we will prove the crucial result that the number of support constraints of a problem $P[K]$ is less than or equal to 2^d if $\Omega \subset \mathbb{Z}^d$, that is, if the decision variables are confined to lie on the *integer* lattice. The proof uses the following fact from [59, Proof of Proposition 4.2, p.83]:

Fact 2.1 *Let $S \subset \mathbb{Z}^d$ be a set with $2^d + 1$ points and let $\text{ci}(S)$ denote the smallest convex set in \mathbb{Z}^d that includes S . Then it holds that*

$$\bigcap_{x \in S} \text{ci}(S \setminus \{x\}) \neq \emptyset. \quad (2.2)$$

This means that, if the number of points in a set $S \subset \mathbb{Z}^d$ is large enough, then the intersection of \mathbb{Z}^d and the convex hulls of all subsets $S \setminus \{x\}$, consisting of S with the point x removed, is nonempty.

Lemma 2.1 *For convex optimization problem $P[K]$ that is feasible with $\Omega \subset \mathbb{Z}^d$, the number of support constraints is no larger than $2^d - 1$.*

PROOF. The proof is by contradiction. Assume that there are more than $2^d - 1$ support constraints. Without loss of generality we consider the case with 2^d support constraints and also assume that the support constraints are the first 2^d constraints $k = 1, \dots, 2^d$.

Let $x_0^* := x^*(K) \in \mathbb{Z}^d$ denote the optimal solution of the optimization problem and let $x_k^* := x^*(K \setminus k) \in \mathbb{Z}^d$ for $k = 1, \dots, 2^d$ be the optimal solution when support constraint k is removed. Let $J^* := c^\top x_0^*$ and $J_k^* := c^\top x_k^*$, for $k = 1, \dots, 2^d$, denote the respective optimal objective values. Define the set \mathcal{X} to be

$$\mathcal{X} := \{x_0^*, x_1^*, \dots, x_q^*\} \subset \mathbb{Z}^d, \quad q = 2^d. \quad (2.3)$$

Notice that all points in \mathcal{X} are distinct. To prove this, assume for contradiction that two of them coincide e.g. $x_{k_1}^* = x_{k_2}^*$. The point $x_{k_1}^*$ satisfies

all constraints except for k_1 and the same holds for the point $x_{k_2}^*$ with constraint k_2 . Since they are equal, they satisfy all constraints. Since k_1 and k_2 are support constraints, the points have a better objective value than x_0^* : $J_{k_1}^* = J_{k_2}^* < J_0^*$ by definition. So the point $x_{k_1}^* = x_{k_2}^*$ satisfies all constraints in K and has a better objective value than x_0^* , which is a contradiction to x_0^* being optimal for $P[K]$.

Define η_{min} as the smallest objective improvement when discarding a support constraint

$$\eta_{min} := \min_{k=1, \dots, 2^d} (J^* - J_k^*) \quad (2.4)$$

and let η be such that $0 < \eta < \eta_{min}$. Consider the halfspace

$$\mathcal{H} := \left\{ x : c^\top x < J^* - \eta \right\}. \quad (2.5)$$

By construction, all points x_k^* , $k = 1, \dots, 2^d$, lie in \mathcal{H} while x_0^* does not.

Since there are 2^d support constraints and also $x_0^* \in \mathcal{X}$ we have that $|\mathcal{X}| = 2^d + 1$. We now apply Fact 2.1 and obtain that

$$\bigcap_{x \in \mathcal{X}} \text{ci}(\mathcal{X} \setminus \{x\}) \neq \emptyset \quad (2.6)$$

and hence there exists a $z \in \mathbb{Z}^d$ that is in the intersection.

Since z is in the intersection (2.6) we have $z \in \text{ci}(\mathcal{X} \setminus \{x_0^*\})$ and hence it is in the convex hull of all the points x_k^* , $k = 1, \dots, 2^d$. It follows that $z \in \mathcal{H}$ since all the x_k^* for $k \neq 0$ are in \mathcal{H} .

For all support constraints $k \in \text{Sc}(K)$ we know that $z \in \text{ci}(\mathcal{X} \setminus \{x_k^*\})$, and since all other points x_j^* , $j \neq k$, satisfy constraint k (and all other non-support constraints), z does too. So z is feasible for $P[K]$. Hence, z is an integer point that is feasible for $P[K]$ and also $z \in \mathcal{H}$, which means that $c^\top z < J^* - \eta < J^* = c^\top x_0^*$. We have thus found a feasible integer point with objective smaller than x_0^* , which is a contradiction to x_0^* being optimal.

□

2.2.2. Support Constraints for Mixed-Integer Problems

In this section, we prove an upper bound on the number of support constraints of a *mixed-integer* convex optimization problem, i.e., a problem $P[K]$ where the decision variables are confined to lie in $\Omega \subset \mathbb{R}^n \times \mathbb{Z}^d$. In fact, we will prove an upper bound on the number of support constraints for any $\Omega \subset M$ where M is a general closed subset of \mathbb{R}^k . This can be an integer or a mixed-integer or a some other space.

Definition 2.3 (Helly's Dimension) We define *Helly's dimension* $h(M)$ of M to be the smallest integer h such that for every finite collection of M -convex sets C_1, \dots, C_m with $m \geq h$ for which every subcollection of h sets has nonempty intersection

$$\bigcap_{i \in I} C_i \neq \emptyset, \quad (2.7)$$

with $|I| = h$, it follows that the intersection $C_1 \cap C_2 \cap \dots \cap C_m$ of all sets is also nonempty.

Lemma 2.2 *For a feasible convex optimization problem $P[K]$ with $\Omega \subset M$, the number of support constraints is less than or equal to $h(M) - 1$.*

PROOF. The proof is by contradiction. We assume that $P[K]$ has q support constraints and that $q \geq h$. Without loss of generality we assume that the first q constraints $\{1, \dots, q\}$ are the support constraints. Let x_0^* be the optimal solution of $P[K]$ and denote by x_k^* , $k = 1, \dots, q$ the optimal solutions of $P[K \setminus k]$, i.e., for $P[K]$ with support constraint k removed. All the points x_k^* for $k = 0, \dots, q$, are distinct with the same argumentation as in the proof of Lemma 2.1.

2. Mixed-Integer Random Convex Programs

Let $J_k^* := J^*(K \setminus k)$ be the optimal objective value of $P[K \setminus k]$. Define η_{min} as the smallest objective improvement when discarding a support constraint

$$\eta_{min} := \min_{k=1, \dots, q} (J^* - J_k^*) \quad (2.8)$$

and let η be such that $0 < \eta < \eta_{min}$. Consider the halfspace

$$\mathcal{H} := \left\{ x : c^\top x < J^* - \eta \right\}. \quad (2.9)$$

By construction, all points x_k^* , $k = 1, \dots, q$, lie in \mathcal{H} while x_0^* does not.

Define the polytopes

$$P_k := \text{conv}_M \{x_i^* : i \in \{0, \dots, q\} \setminus k\}, \quad (2.10)$$

for $k = 0, \dots, q$ to be the convex hull in M of the points $\{x_0^*, x_1^*, \dots, x_q^*\}$ except for the point x_k^* . We have $q + 1$ polytopes P_k , since there are $q + 1$ points x_i^* . Since the points in $\{x_0^*, x_1^*, \dots, x_q^*\} \setminus x_k^*$ satisfy constraint k , by convexity of the constraints it follows that all points in the convex hull P_k satisfy constraint k .

Let $I \subset \{0, \dots, q\}$ be an arbitrary index set of cardinality $|I| = h$. Since by assumption $h < q + 1$ there is an index $j \in \{0, \dots, q\}$ that is not in the index set I : $\exists j \notin I$. So there is a point $x_j^* \in M$ that lies in all the sets P_i , $i \in I$, by construction of the polytopes P_i , and hence x_j^* lies in the intersection

$$x_j^* \in \bigcap_{i \in I} P_i. \quad (2.11)$$

It follows that for arbitrary index sets $I \subset \{0, \dots, q\}$ with cardinality $|I| = h$ the intersection $\bigcap_{i \in I} P_i \neq \emptyset$ is nonempty. Since h is the Helly dimension of M , we can conclude from Definition 2.3 that the intersection of *all* P_k is not

empty

$$\bigcap_{k=0,\dots,q} P_k \neq \emptyset \tag{2.12}$$

and there exists a $z \in M$ that lies in the intersection of all the P_k . This point z is in $P_0 = \text{conv}_M \{x_1^*, \dots, x_q^*\}$, which does not include x_0^* , hence z is in \mathcal{H} by construction. The point z also satisfies all q support constraints (and all other constraints) because it is in each of the P_k . So, $c^\top z < J^* - \eta < J^*$ because $z \in \mathcal{H}$, and z is feasible for problem (2.1) and has better objective than x_0^* which is a contradiction to x_0^* being optimal. \square

Corollary 2.1 *For a feasible mixed-integer convex optimization problem $P[K]$ with $\Omega \subset \mathbb{R}^n \times \mathbb{Z}^d$, the number of support constraints is less than or equal to $(n + 1)2^d - 1$.*

PROOF. Averkov and Weismantel show in [22] that Helly’s dimension of $\mathbb{R}^n \times \mathbb{Z}^d$ is $h(\mathbb{R}^n \times \mathbb{Z}^d) = (n + 1)2^d$. The claim then follows from Theorem 2.2. \square

We have thus established that a feasible mixed-integer convex problem $P[K]$ has at most $h - 1$ support constraints.

The final result of this section is a direct generalization of [44, Lemma 2.3] for infeasible programs $P[K]$:

Lemma 2.3 *For an infeasible convex optimization problem $P[K]$ with $\Omega \subset M$ the number of support constraints is less than or equal to h .*

PROOF. The proof is by contradiction and proceeds along similar lines as the proof of Lemma 2.2 in [44]. Let $P[K]$ be infeasible and without loss of generality let the first q constraints be support constraints. Assume for contradiction that $q > h$.

For each support constraint j we have that $J^*(K \setminus j) < J^*(K) = \infty$, i.e., the optimal objective when support constraint j is removed is smaller than

∞ . It follows that if a support constraint j is removed from the constraint set K the problem becomes feasible. Define

$$\mathcal{X}_i := \text{Sat}(i) := \{x: f_i(x) \leq 0\} \quad (2.13)$$

and define for each $i \in K$

$$\mathcal{S}_i := \text{Sat}(K \setminus i) := \{x: f_j(x) \leq 0, j \in K, j \neq i\} \quad (2.14)$$

For a support constraint i the sets \mathcal{S}_i are nonempty by the argumentation above. Define

$$\Psi := \bigcap_{i=h+2}^{|K|} \mathcal{X}_i. \quad (2.15)$$

It holds that

$$\left(\bigcap_{i=1}^{h+1} \mathcal{X}_i \right) \cap \Psi = \{x: f_j(x) \leq 0, j = 1, \dots, |K|\} \quad (2.16)$$

$$= \text{Sat}(K) = \emptyset. \quad (2.17)$$

Consider the collection of M -convex sets

$$\mathcal{C} := \{\mathcal{X}_1, \dots, \mathcal{X}_{h+1}, \Psi\} \quad (2.18)$$

and consider an arbitrary subcollection of \mathcal{C} consisting of h sets. Denote the intersection over this subcollection by \mathcal{S} . Since \mathcal{C} consists of $h+2$ sets, there will be at least one \mathcal{X}_i missing in the subcollection and, hence, $\mathcal{S}_i = \text{Sat}(K \setminus i)$ will be contained in the subcollection. It follows that the intersection \mathcal{S} is not empty, since $\mathcal{S}_i \subset \mathcal{S}$ and \mathcal{S}_i is not empty since i is a support constraint.

We have thus showed that for an arbitrary subcollection of \mathcal{C} consisting of h sets, the intersection over this subcollection is always nonempty. Since

h is the Helly dimension of M , it follows that the intersection over *all* sets in \mathcal{C} is nonempty. The intersection over all sets in \mathcal{C} is $\text{Sat}(K)$ which is a contradiction to $P[K]$ infeasible, i.e., $\text{Sat}(K) = \emptyset$. It follows that the number of support constraints must be less than or equal to h . \square

2.2.3. Examples

Previous results in [44, 46, 49] show that for continuous spaces the number of support constraints depends linearly on the dimension of the decision variable. In the case of integer or mixed-integer decision variables our results show that there is an exponential dependence of the number of support constraints on the dimension of the integer decision variables. In this section, we show that these bounds on the number of support constraints can actually be attained, i.e., we construct two examples of integer linear problems (ILP) with exactly $2^d - 1$ support constraints.

Example 2.1: A two-dimensional Example

Consider the ILP

$$\begin{aligned} \min \quad & -x - y \\ \text{s.t.} \quad & x + y \leq 1.5 \end{aligned} \tag{2.19}$$

$$x - 0.3y \leq 0.75 \tag{2.20}$$

$$-0.3x + y \leq 0.75 \tag{2.21}$$

$$(x, y) \in \mathbb{Z}^2 \cap [-10, 10]^2 .$$

The optimal solution to this ILP is $(x^*, y^*) = (0, 0)$, as it can be easily checked (see Figure 2.1). If we remove constraint (2.19), the integer point $(x_1, y_1) = (1, 1)$ has a better objective than (x^*, y^*) and, hence, constraint (2.19) is a support constraint. If we remove constraint (2.20), the integer point $(x_2, y_2) = (1, 0)$ has a better objective than (x^*, y^*) and, hence, constraint (2.20) is also a support constraint. If we remove

2. Mixed-Integer Random Convex Programs

constraint (2.21), the integer point $(x_3, y_3) = (0, 1)$ has a better objective than (x^*, y^*) and, hence, constraint (2.21) is also a support constraint. This feasible ILP has therefore $2^d - 1 = 3$ support constraints. ■

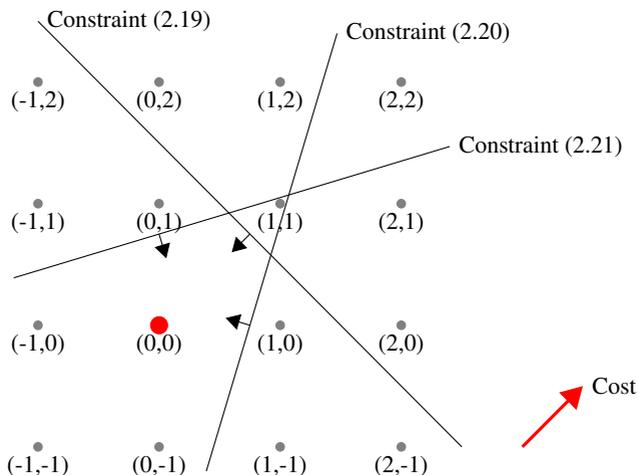


Figure 2.1.: Illustration of the feasible region of a two-dimensional ILP. The gray dots depict the integer lattice and the black lines the linear constraints. The bold arrows are the normal directions pointing into the feasible region. The red arrow on the bottom right depicts the direction of cost improvement and the red point is the optimal integer solution.

Example 2.2: A three-dimensional Example

Consider the ILP

$$\begin{aligned} \min \quad & -x - y - z \\ \text{s.t.} \quad & x + y + z \leq 2.5 \end{aligned} \tag{2.22}$$

$$-0.35x + y + z \leq 1.7 \tag{2.23}$$

$$x - 0.35y + z \leq 1.7 \tag{2.24}$$

$$x + y - 0.35z \leq 1.7 \tag{2.25}$$

$$x - 0.35y - 0.35z \leq 0.85 \tag{2.26}$$

$$-0.35x + y - 0.35z \leq 0.85 \tag{2.27}$$

$$-0.35x - 0.35y + z \leq 0.85 \tag{2.28}$$

$$(x, y, z) \in \mathbb{Z}^3 \cap [-10, 10]^3 .$$

It can be checked that the optimal integer solution of this problem is $(x^*, y^*, z^*) = (0, 0, 0)$. If constraint (2.22) is removed, the integer point $(x_1, y_1, z_1) = (1, 1, 1)$ has a better objective than (x^*, y^*, z^*) and, hence, constraint (2.22) is a support constraint. The same can be checked for the removal of each of the constraints (2.23)-(2.28). So we obtain an ILP that has $2^3 - 1 = 7$ support constraints. ■

2.2.4. Additional Definitions and Facts

In this section we will introduce the essential constraint set, a minimal set of constraints on which the objective of the optimization problem is the same as on the whole constraint set K .

Definition 2.4 (Essential Constraint Set) Define the essential constraint set of a problem $P[K]$ to be

$$\text{Es}(K) := \arg \min \{ |S| : J^*(S) = J^*(K), S \subset K \} , \tag{2.29}$$

i.e., the minimal set of constraints that uniquely defines the objective value of $P[K]$.

Definition 2.5 (Nondegenerate Problem) A problem $P[K]$ is called nondegenerate if for the optimal objective holds $J^*(K) = J^*(\text{Sc}(K))$, i.e., when the optimal objective under all constraints equals the optimal objective when only the support constraints are considered.

The following Lemmas will be stated mostly without proof, since their proof is verbatim the same as their corresponding counterparts in [44]. The argumentation in [44] does not rely on the continuity of the space Ω but instead argues on the constraint sets only. This is the reason, why the proofs carry over from the continuous to the mixed-integer case without change. We will state the corresponding number of the Lemma in [44] at the top.

Lemma 2.4 (Lemma 2.10 in [44]) For an optimization problem $P[K]$ we have

1. $\text{Sc}(K) \subset \text{Es}(K)$
2. $|\text{Sc}(K)| \leq |\text{Es}(K)| \leq h$, where h is the Helly dimension.

This Lemma is analog to Lemma 2.10 in [44] and both the statement and the arguments for the proof remain unchanged except for the inequality $|\text{Es}(K)| \leq h$. We give a proof of this inequality now.

PROOF. $|\text{Es}(K)| \leq h$:

The proof follows the same line of argumentation as the proof of Lemma 2.10 in [44]. We give it here for the sake of completeness. Let a problem $P[K]$ with constraint set K be given. We will now construct a descending sequence $K_0 = K \supset K_1 \supset K_2 \dots$ of constraint sets that are all invariant for $P[K]$, i.e., $\text{Obj}[K] = \text{Obj}[K_i]$ for all i . We will show that there is an index i and a corresponding constraint set K_i with $|K_i| = h$ and from this the desired statement will follow since $\text{Es}(K)$ is the *smallest* invariant constraint set and, hence, $h = |K_i| \geq |\text{Es}(K)|$.

Define $K_0 := K$, $S_0 := \text{Sc}(K)$, $\bar{K}_0 := K_0 \setminus S_0$ and $i = 0$. Consider the following procedure:

1. Is $|S_i| = h$ or $\bar{K}_i = \emptyset$ halt the procedure.
2. Let $k \in \bar{K}_i$ and define $K_{i+1} := K_i \setminus \{k\}$, set $i = i + 1$, $S_{i+1} = \text{Sc}(K_{i+1})$, and go to step 1.

If $P[K]$ is feasible all the other problems with smaller constraint sets K_i will also remain feasible. From the definition of support constraints it follows that the optimal objective remains the same on all the constraint sets K_i and, hence, they are all invariant sets. With the same argumentation as in [44, Proof Lemma 2.10] it follows that support constraints “inherit” $S_i \subset S_{i+1}$. The procedure terminates when either $|S_i| = h$, then the problem $P[K_i]$ is fully-supported and by Lemma 2.6 nondegenerate. Then S_i is an invariant set for $P[K]$ and since the constraints in S_i are support constraints for problem $P[K_i]$, there at most h of them by Theorems 2.2 and 2.3. The other termination condition is that $\bar{K}_i = \emptyset$. Then all constraints of $P[K_i]$ are support constraints $S_i = K_i$ and, hence, the cardinality of K_i is bounded by h . \square

Lemma 2.5 (Lemma 2.11 in [44]) *If $P[K]$ is nondegenerate its essential constraint set is unique and coincides with the support constraint set.*

We conclude the section by introducing the concept of fully supported problems, i.e., problems in which the number of support constraints is exactly $h - 1$ if they are feasible and h if they are infeasible.

Definition 2.6 (Fully Supported Problem) A problem $P[K]$ is called *fully supported* if the number of its support constraints is exactly $h - 1$ if $P[K]$ is feasible and h if it is not feasible.

Lemma 2.6 (Lemma 2.8 in [44]) *If problem $P[K]$ is fully-supported, then it is nondegenerate.*

Lemma 2.7 (Lemma 2.12 in [44]) For a subset of the constraint set $Y \subset K$ and additional constraints $h_1, \dots, h_n \in K$ it holds that

$$J^*(Y, h_j) = J^*(Y), j = 1, \dots, n \Leftrightarrow J^*(Y, h_1, \dots, h_n) = J^*(Y). \quad (2.30)$$

The case that a problem $P[K]$ is nondegenerate but *not fully-supported* can occur. There is a technique called regularized refinement that allows to “refine” a nondegenerate problem $P[K]$ to another problem $\bar{P}[K]$ that will always be fully supported. We will here briefly sketch this procedure, for details see [44]. Assume that the number of constraints is greater than or equal to $h - 1$ (or h in the infeasible case) and let \bar{K} be the set of constraints in K that are not essential constraints. Define the essential cardinality gap $\nu := (h - 1) - |\text{Es}(K)|$ (or $\nu := h - |\text{Es}(K)|$ in the infeasible case). Further, define an arbitrary total order on the constraints in K , i.e., attach a number to each constraint such that we can compare constraints. Let $Z(K)$ be the ν constraints in K with the highest ranking and define the augmented objective

$$\bar{J}^*(K) := [J^*(K), Z(K)]. \quad (2.31)$$

This means the following: we say that $\bar{J}^*(K) > \bar{J}^*(K')$ for two constraint sets K and K' if and only if $J^*(K) > J^*(K')$ or, if they are equal $J^*(K) = J^*(K')$ then the order on the constraints decides which constraint set has the higher objective $Z(K) > Z(K')$. Hence, the regularized refinement is a way of discriminating between constraint sets that lead to the same optimal objective function J^* . The most important property of the regularized refinement is that

$$J^*(K) > J^*(K') \Rightarrow \bar{J}^*(K) > \bar{J}^*(K') \quad (2.32)$$

which means that $\bar{P}(K)$ is a refinement of $P[K]$ in the sense of LP-type problems [94]. It can be shown that the refinement has exactly $h - 1$ many

essential constraints (h in the infeasible case) and if $P[K]$ is nondegenerate, so will its refinement. Also, it can be shown that all the preceding Lemmas in the section hold for the regularized refinement [44].

2.3. Mixed-Integer Random Convex Programs

In this section, we consider random convex programs in which the decision variable can be integer or continuous or a mixture of both. Consider a function $f(x, \delta) : (\mathbb{R}^n \times \mathbb{R}^d) \times \mathbb{R}^l \rightarrow \mathbb{R}$ that is convex and lower-semicontinuous in x for any fixed δ . Let $\delta \in \Delta \subset \mathbb{R}^l$ denote a random vector and let \mathbb{P} be a probability measure over Δ . Denote by $\omega := (\delta^{(1)}, \dots, \delta^{(N)})$ N independent realizations drawn from δ .

Definition 2.7 (Mixed-Integer Random Convex Program) We will call an optimization problem of the form

$$\begin{aligned} P[\omega] : \min c^\top x \\ \text{s.t. } f(x, \delta^{(j)}) \leq 0, \quad j = 1, \dots, N \\ x \in \Omega \cap (\mathbb{R}^n \times \mathbb{Z}^d), \end{aligned} \quad (2.33)$$

where the decision variable x lies in the convex compact domain $\Omega \cap (\mathbb{R}^n \times \mathbb{Z}^d)$ (take $n = 0$ for purely integer random convex programs and $d = 0$ for continuous RCPs) a *mixed-integer random convex program* (MI-RCP).

Definition 2.8 (Violation Probability) For an MI-RCP $P[\omega]$ the violation probability is defined as

$$V_{mi}^*(\omega) := \mathbb{P}\{\delta \in \Delta : J^*(\omega, \delta) > J^*(\omega)\}, \quad (2.34)$$

that is, the probability that the optimal objective of $P[\omega]$ computed under the realization ω will not remain optimal under the next realization $\delta \in \Delta$.

$V_{mi}^*(\omega)$ is itself a random variable with values in $[0, 1]$ depending on the random extraction ω .

Definition 2.9 (Helly's number) Define Helly's number ζ to be the smallest integer such that

$$\zeta \geq \text{ess sup}_{\omega \in \Delta^N} |\text{Sc}(P[\omega])| \quad (2.35)$$

for any number of realizations N .

So Helly's number of an MI-RCP is the smallest integer such that with probability one, there are less or equal than ζ many support constraints for every problem $P[\omega]$. Note that from Lemmas 2.2 and 2.3 it follows that $\zeta \leq h$, where h is the Helly dimension of the underlying decision space.

We make the following assumptions regarding $P[\omega]$:

Assumption 3

1. *The optimal solution of $P[\omega]$ is unique if it exists.*
2. *$P[\omega]$ is fully supported with probability one.*

The first assumption is no severe restriction, since the uniqueness of the optimal solution can always be achieved, for instance by introducing tie-breaking rules (see e.g. [46]). Also, the assumption that $P[\omega]$ is fully supported can be relaxed to the assumption that $P[\omega]$ is nondegenerate with probability one if refinement techniques as in [44] are applied. We require it here only to streamline the proof of the main result. See Corollary 2.2 for results on the more general case of not fully supported problems.

Please note that we do not require here the problems $P[\omega]$ to be feasible with probability one (in contrast to e.g. [7, 49]). In the case that $P[\omega]$ is infeasible, we have that $J^*(\omega) = \infty = J^*(\omega, \delta)$ and $V_{mi}^*(\omega) = 0$.

2.3.1. Main Result on MI-RCPs

We are now in a position to state our main result on the tails of the random variable $V_{mi}^*(\omega)$ for random convex programs with mixed-integer decision variables.

Theorem 2.1 *Consider an MI-RCP $P[\omega]$ as in Definition 2.7, let $N \geq h$, where $h = (n+1)2^d$ is the Helly dimension of $\mathbb{R}^n \times \mathbb{Z}^d$ and let Assumption 3 hold. Then for $\varepsilon \in (0, 1]$*

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon \} \leq \Phi(\varepsilon; \zeta - 1, N) \leq \Phi(\varepsilon; h - 1, N), \quad (2.36)$$

where $\Phi(\varepsilon; q, N)$ denotes the cumulative distribution of the binomial random variable

$$\Phi(\varepsilon; q, N) := \sum_{j=0}^q \binom{N}{j} \varepsilon^j (1 - \varepsilon)^{N-j}. \quad (2.37)$$

PROOF. The proof follows the lines of Theorem 3.3 in [44]. Since $P[\omega]$ is fully supported with probability one, the support constraint set has cardinality exactly ζ . $P[\omega]$ is nondegenerate with probability one by Lemma 2.6 and the support constraint set equals the essential constraint set by Lemma 2.5. Let $I_{\zeta}^i(\omega), i = 1, \dots, C_{N, \zeta}$ with $C_{N, \zeta} = \binom{N}{\zeta}$ denote the subsets of ζ elements of $\omega = (\delta^{(1)}, \dots, \delta^{(N)})$. Without loss of generality let $I_{\zeta}^1(\omega)$ be the first ζ elements of ω . Define the events S_i as

$$S_i := \left\{ \omega \in \Delta^N : \text{the essential constraint set of } P[\omega] \text{ is } I_{\zeta}^i(\omega) \right\}$$

for all $i = 1, \dots, C_{N, \zeta}$. S_i is the event that constraints $I_{\zeta}^i(\omega)$ are the essential constraints of problem $P[\omega]$, i.e., the event that the optimal solution with constraints $I_{\zeta}^i(\omega)$ equals the optimal solution found with all constraints ω .

Since there is exactly one essential constraint set of cardinality ζ , the events S_i are mutually exclusive $S_i \cap S_j = \emptyset$ for $i \neq j$ and exhaustive

$$\Delta^N = \bigcup_{i=1}^{C_{N,\zeta}} S_i. \quad (2.38)$$

It follows that

$$1 = \mathbb{P}^N \{\Delta^N\} = \sum_{i=1}^{C_{N,\zeta}} \mathbb{P}^N \{S_i\} = C_{N,\zeta} \mathbb{P}^N \{S_1\} \quad (2.39)$$

because all constraint sets have the same probability of being the essential constraint set and, hence, $\mathbb{P}^N \{S_1\} = \frac{1}{C_{N,\zeta}}$. Define now

$$V_i(\omega) := \mathbb{P} \left\{ \delta \in \Delta : J^*(I_\zeta^i(\omega), \delta) > J^*(I_\zeta^i(\omega)) \right\}. \quad (2.40)$$

the violation probability of an optimal solution found on the constraint set $I_\zeta^i(\omega)$. We will now, without loss of generality, consider V_1 , as all V_i have the same distribution, because no constraint set $I_\zeta^i(\omega)$ has a higher probability of occurring than the others. Define for $\alpha \in [0, 1]$ the probability distribution of V_1 as

$$F_1(\alpha) := \mathbb{P}^\zeta \left\{ \omega^\zeta : V_1(\omega^\zeta) \leq \alpha \right\}, \quad (2.41)$$

where $\omega^\zeta = (\delta^{(1)}, \dots, \delta^{(\zeta)})$. Assume for now that $V_1 = v$ is given.

The event S_1 equals the event

$$S_1 = \left\{ J^*(I_\zeta^1(\omega)) = J^*(I_\zeta^1(\omega), \delta^{(\zeta+1)}, \dots, \delta^{(N)}) \right\} \quad (2.42)$$

by definition of S_1 . Then by Lemma 2.7 it follows that

$$S_1 = \left\{ J^*(I_\zeta^1(\omega)) = J^*(I_\zeta^1(\omega), \delta^{(\zeta+j)}), j = 1, \dots, N - \zeta \right\}. \quad (2.43)$$

Since all extractions of δ are drawn independently the probability mass of S_1 equals the probability mass of the event that the optimal objective found on $I_\zeta^1(\omega)$ remains the same on $N - \zeta$ independent realizations drawn from δ . The event that the optimal objective obtained on $I_\zeta^1(\omega)$ remains equal when a further randomly drawn constraint is added, is exactly the probability $1 - V_1(\omega) = 1 - v$.

We obtain that

$$\mathbb{P}^N\{S_1|V_1 = v\} = (1 - v)^{N-\zeta}. \quad (2.44)$$

If we now decondition V_1 and recall (2.39), we obtain

$$\frac{1}{C_{N,\zeta}} = \mathbb{P}^N\{S_1\} = \int_0^1 (1 - v)^{N-\zeta} dF_1(v). \quad (2.45)$$

This is a Hausdorff moment problem [66] and with the same argumentation as in [44, 49] it follows that $F_1(\alpha) = \alpha^\zeta$.

It follows for the set $B := \{\omega : V_{mi}^*(\omega) > \varepsilon\}$, for which we want to prove the theorem, that

$$B = \bigcup_{i=1}^{C_{N,\zeta}} B \cap S_i, \quad (2.46)$$

where $B \cap S_i = \{\omega : V_i(\omega) > \varepsilon\}$. All intersections $B \cap S_i$ are disjoint and have the same probability mass because all constraint sets have the same probability of being an essential constraint set. From

$$B \cap S_i = \bigcup_{\alpha \in (\varepsilon, 1]} \{\omega : V_i(\omega) = \alpha\} \quad (2.47)$$

we conclude that

$$\mathbb{P}^N\{B \cap S_1\} = \int_{\varepsilon}^1 \mathbb{P}^N\{S_1, V_1 = \alpha\} d\alpha \quad (2.48)$$

$$= \int_{\varepsilon}^1 \mathbb{P}^N\{S_1 | V_1 = \alpha\} dF_1(\alpha) \quad (2.49)$$

$$\stackrel{(a)}{=} \zeta \int_{\varepsilon}^1 (1-v)^{N-\zeta} \alpha^{\zeta-1} d\alpha, \quad (2.50)$$

where equality (a) follows from Equation (2.44) and $F_1(\alpha) = \alpha^\zeta$. The integral (2.50) is the incomplete beta function $\zeta B(1 - \varepsilon; N - \zeta + 1, \zeta)$ and it follows that

$$\mathbb{P}^N\{B \cap S_1\} = \zeta B(1 - \varepsilon; N - \zeta + 1, \zeta) \quad (2.51)$$

$$= \binom{N}{\zeta}^{-1} \Phi(\varepsilon; \zeta - 1, N) \quad (2.52)$$

(for more details on this derivation please see [44, Line 3.16]). Finally, observe that

$$\mathbb{P}^N\{B\} = C_{N,\zeta} \mathbb{P}^N\{B \cap S_1\} = \Phi(\varepsilon; \zeta - 1, N) \stackrel{(b)}{\leq} \Phi(\varepsilon; h - 1, N), \quad (2.53)$$

where (b) follows since $\zeta \leq h$ and the binomial distribution is monotonically increasing in this second argument. This concludes the proof. \square

Corollary 2.2 *Let $P[\omega]$, N , and h be as in Theorem 2.1 and let the optimal solution of $P[\omega]$ be unique if it is feasible and let $P[\omega]$ be nondegenerate with probability one (i.e., we relax the assumption that $P[\omega]$ is fully-supported with probability one), then the inequality*

$$\mathbb{P}^N\{\omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon\} \leq \Phi(\varepsilon; h - 1, N) \quad (2.54)$$

holds.

PROOF. As mentioned above, the use of a regularized refinement analogously as in [44] in combination with Theorem 2.1 can be used to prove this corollary. The regularized refinement of problem $P[\omega]$ is constructed by attaching to each realization of δ a real number drawn uniformly from the unit interval $[0, 1]$. With probability one this induces a total order on all the constraint realizations. Then Theorem 2.1 can be applied to the regularized refinement $\bar{P}[\omega]$ to obtain the violation bound for it. From the property (2.32) of the regularized refinement it then follows that

$$J^*(\omega, \delta) > J^*(\omega) \Rightarrow \bar{J}^*(\omega, \delta) > \bar{J}^*(\omega) \quad (2.55)$$

and, hence,

$$V_{mi}^* := \mathbb{P}\{J^*(\omega, \delta) > J^*(\omega)\} \leq \mathbb{P}\{\bar{J}^*(\omega, \delta) > \bar{J}^*(\omega)\} := \bar{V}_{mi}^* \quad (2.56)$$

Then it follows that

$$\mathbb{P}^N\{V_{mi}^* > \varepsilon\} \leq \mathbb{P}^N\{\bar{V}_{mi}^* > \varepsilon\} \stackrel{\text{Thm. 2.1}}{\leq} \Phi(\varepsilon; h-1, N). \quad (2.57)$$

□

Remark 2.1 For an MI-RCP with $n = 0$, i.e., for a pure integer random convex program, the resulting bound on the tails of the violation probability is

$$\mathbb{P}^N\{\omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon\} \leq \Phi(\varepsilon; 2^d - 1, N)$$

and for a true mixed-integer problem, the bound is

$$\mathbb{P}^N\{\omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon\} \leq \Phi(\varepsilon; (n+1)2^d - 1, N).$$

2.3.2. Discussion of the Results

The bounds on the tails of the violation probability in MI-RCPs derived in this section hold for the most general case of optimization problems with random nonlinear convex constraints and mixed-integer decision variables. These bounds can be tractable when the dimension of the space of integer decision variables is comparatively low. However, differently from the continuous case, where the upper bound on the number of support constraints increases linearly with the number of variables, in the mixed-integer setting we obtained a geometric dependence of this upper bound with respect to the number of discrete variables (also, we showed examples in dimension two and three where the upper bound is attained). It follows that the parameter in the right hand side of the bounds on the tails of the violation probability also depends geometrically on the dimension of the integer decision variable. This can lead to the bound being conservative and as a consequence high sample numbers could be needed in order to ensure a desired confidence. This suggests that mixed-integer convex problems, besides being much harder computationally with respect to their continuous counterparts, may also be more difficult to immunize against uncertainty via the RCP approach.

The rest of this chapter is concerned with the study of the special case of random linear programs and the derivation of bounds that are an improvement to the bounds derived in this section.

2.4. Mixed-Integer Random Linear Programs

In this section we will consider mixed-integer random linear programs of the form

$$\begin{aligned}
 \text{ILP}[\omega] : \quad & \min c^\top x \\
 \text{s.t.} \quad & f(x, \delta^{(j)}) \leq 0, \quad j = 1, \dots, N \\
 & x \in \Omega \cap \mathbb{R}^n \times \mathbb{Z}^d
 \end{aligned} \tag{2.58}$$

where Ω is a convex polytope and the constraint function $f(x, \delta)$ is of the form

$$f(x, \delta) = \max_{i=1, \dots, m} \left\{ a_i(\delta)^\top x - b_i(\delta) \right\}, \quad (2.59)$$

i.e., piecewise affine and convex in x . Note that piecewise affine and convex constraint functions correspond to random constraints of the form $A(\delta)x \leq b(\delta)$ where $A(\delta)$ and $b(\delta)$ have rows $a_i(\delta)^\top$, $b_i(\delta)$ respectively. We denote the linear relaxation of ILP $[\omega]$ by LP $[\omega]$, i.e., LP $[\omega]$ is the random linear program resulting from ILP $[\omega]$ when the integer variables are relaxed to also be continuous. The feasible region of LP $[\omega]$ is always a polytope. Denote by l the dimension of the decision variable of LP $[\omega]$, i.e., $l = n + d$.

First, our focus will lie on vertices of the feasible region of LP $[\omega]$. More precisely, we will study the tails of the probability that a vertex of the feasible region becomes infeasible under the next random constraint extraction. Then we will derive bounds on the tails of the probability that a convex combination of vertices of the feasible region of LP $[\omega]$ becomes infeasible under the next random constraint. Since it follows from Minkowski's Theorem that every point in a polytope can be expressed as the convex combination of the polytope's vertices, these bounds then effectively allow us to bound the violation probability of *any* point in the feasible region of LP $[\omega]$.

In general, the optimal mixed-integer solution of ILP $[\omega]$ will not coincide with the optimal solution of LP $[\omega]$ but will lie in the interior of its feasible region. So we will utilize the results on the violation probability of arbitrary points in the feasible region to bound the tails of the violation probability of the optimal solution of a mixed-integer random linear program. This bound is different from the bound in Theorem 2.1 in that it does not depend geometrically on number of integer decision variables. Since it depends on the number of vertices of the relaxed feasible region, it can also become conservative in the case when the feasible region is given by

many inequalities. However, we will see in numerical examples that the new bound needs way less samples than the one from Theorem 2.1.

2.4.1. Vertices of RLPs

In this subsection we will consider vertices of the feasible region of RLPs and deduce bounds on the tails of the probability that a vertex becomes infeasible under the next realization of the random constraints.

Definition 2.10 (Vertex of a Polytope) An extreme point or vertex of a polytope P is a point $x \in P$ such that there are no points $y, z \in P$, $x \neq y$ and no $\lambda \in (0, 1)$ with $x = \lambda y + (1 - \lambda)z$.

Definition 2.11 (Simple Polytope) A l -polytope $P \subset \mathbb{R}^l$ is called simple, if at each vertex of P exactly l hyperplanes intersect. Simple polytopes correspond to nondegenerate linear programs in linear programming [133].

If P is represented through a finite set of inequalities

$$P = \{x: Ax \leq b\} \tag{2.60}$$

for a matrix A and vector b and is simple, then an extreme point of P corresponds to a set of exactly l linear independent rows a_i^\top of A whose corresponding constraints $a_i^\top x \leq b_i$ are active in x , i.e., $a_i^\top x = b_i$. The other constraints will not be active, i.e., $a_j^\top x < b_j$ for j not one of the active rows. These rows correspond to the hyperplanes $\{x \in \mathbb{R}^n: a_i x = b_i\}$ that intersect in the vertex. For details see e.g. [34, Chapter 2].

Assumption 4 We assume that with probability one the feasible region of LP $[\omega]$

$$\text{Sat}(\omega) := \{x \in \Omega: f(x, \delta^{(j)}) \leq 0, j = 1, \dots, N\} \tag{2.61}$$

is a nonempty simple polytope. We also assume that the optimal solution of $\text{LP}[\omega]$ is always unique.

Definition 2.12 (Active Constraints of $\text{LP}[\omega]$) For a point $v \in \text{Sat}(\omega)$ we say that a constraint $\delta^{(j)}$ is active if $f(v, \delta^{(j)}) = 0$. Define the active constraint set of v as

$$\text{Ac}(v) := \left\{ \delta^{(j)} : \delta^{(j)} \text{ active in } v \right\}. \quad (2.62)$$

If $f(x, \delta)$ is affine in x then $\delta^{(j)}$ being active means that the linear inequality corresponding to $\delta^{(j)}$ is active in x . If $f(x, \delta)$ is piecewise affine and convex for $\delta^{(j)}$ to be active means that there exists at least one $i \in \{1, \dots, m\}$ such that the linear inequality $a_i(\delta^{(j)})^\top x - b_i(\delta) \leq 0$ is active in v .

Lemma 2.8 *If the feasible region $\text{Sat}(\omega)$ of $\text{LP}[\omega]$ is simple with probability one, then $\text{LP}[\omega]$ is nondegenerate with probability one and if the constraint function $f(x, \delta)$ is affine in x then $\text{LP}[\omega]$ is also fully-supported.*

PROOF. If the constraint function $f(x, \delta)$ is affine in x and $\text{Sat}(\omega)$ is simple with probability one, then $\text{LP}[\omega]$ will also be nondegenerate and fully-supported in the sense of [44] with probability one. Constraints that are active (i.e., satisfied with equality) in the optimal solution of $\text{LP}[\omega]$ correspond to support constraints and also essential constraints, since every constraint realization $\delta^{(j)}$ corresponds to one linear inequality. So it follows that $\text{LP}[\omega]$ has exactly l support constraints.

If $f(x, \delta)$ is not affine in x but piecewise affine and convex, $\text{LP}[\omega]$ will be *nondegenerate* with probability one in the sense of Definition 2.5. To see this, let x be the optimal solution of $\text{LP}[\omega]$ with piecewise affine constraints. Since the feasible region is assumed to be a simple polytope, there are exactly l linear inequalities $a_i(\delta^{(j_i)})^\top x - b_i(\delta^{(j_i)})$, $i = 1, \dots, l$ that are active in x and all the other inequalities will not be active. These active inequalities stem from at most l realizations $\delta^{(j_i)}$ and hence there are at

most l constraints $f(x, \delta^{(j_i)})$ that are active in x and all the others will not. If we remove one of the constraints

$$f(x, \delta^{(j_i)}) = \max_{i=1, \dots, m} \left\{ a_i \left(\delta^{(j_i)} \right)^\top x - b_i(\delta^{(j_i)}) \right\} \quad (2.63)$$

we will also remove one of the l active linear constraints $a_i(\delta^{(j_i)})x - b_i^{(j_i)}$. By definition, x is then no longer a basic feasible solution, i.e., a vertex of the feasible region, since less than l hyperplanes intersect in x and hence it cannot be the optimum under c since the optimum of a linear program is always assumed at a vertex. This means that there exists a better optimal solution and the constraint $\delta^{(j_i)}$ was a support constraint. Hence, we showed that active constraints are support constraints. In [51,52] it is shown that for a general convex program the support constraint set is a subset of the active constraint set. With the argumentation above it follows that the support constraint set and the active constraint set coincide. The claim then follows. \square

Lemma 2.9 *Let $x(\omega)$ be a vertex of the feasible region $\text{Sat}(\omega)$ and assume that Assumption 4 holds for $\text{Sat}(\omega)$. Then it follows that*

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{\mathcal{Y}}^*(\omega) > \varepsilon \} \leq \Phi(\varepsilon; l-1, N), \quad (2.64)$$

where

$$V_{\mathcal{Y}}^*(\omega) := \mathbb{P} \{ \delta \in \Delta : f(x(\omega), \delta) > 0 \} \quad (2.65)$$

is the probability that the vertex $x(\omega)$ will become infeasible under the next constraint realization.

This means, we can bound the violation probability of a vertex of the feasible region of an LP $[\omega]$ with simple feasible region.

The proof follows the same line of argumentation as the proof of Theorem 3.3 in [44] for a random convex program, except that here we reason on active constraint sets and not support constraints. Since the proof has been elaborated in detail in [44] we will omit most explanations from it.

PROOF. Since $\text{Sat}(\omega)$ is simple with probability one and $x(\omega)$ a vertex of $\text{Sat}(\omega)$ there are exactly l linear constraints that are active in $x(\omega)$. These l linear constraints can stem from *at most* l realizations $\delta^{(j)}$ but it could happen that less than l random constraints $\delta^{(j)}$ are active in $x(\omega)$ (in the sense of Definition 2.12). For now assume that the active linear constraints in $x(\omega)$ stem from *exactly* l realizations $\delta^{(j)}$. We will first prove the lemma under this assumption and then explain how the proof can be extended to the more general case.

Define

$$B := \{ \omega \in \Delta^N : V_{\mathcal{V}}(\omega) > \varepsilon \} . \quad (2.66)$$

Let $I_l^i(\omega)$ for $i = 1, \dots, C_{N,l}$ with $C_{N,l} = \binom{N}{l}$ be subsets of $(\delta^{(1)}, \dots, \delta^{(N)})$ consisting of l elements and without loss of generality let

$$I_l^1(\omega) = (\delta^{(1)}, \dots, \delta^{(l)}) . \quad (2.67)$$

Define the event

$$S_i := \{ \omega \in \Delta^N : I_l^i(\omega) \text{ active in } x(\omega) \} \quad (2.68)$$

that the i -th subset $I_l^i(\omega)$ is comprised of the realizations that are active in $x(\omega)$. Since exactly l constraint realizations are active in $x(\omega)$ it follows that

$$S_i \cap S_j = \emptyset \text{ for } i \neq j \text{ and } \bigcup_{i=1}^{C_{N,l}} S_i = \Delta^N . \quad (2.69)$$

Then it holds that

$$B = B \cap \Delta^N = B \cap \left(\bigcup_{i=1}^{C_{N,l}} S_i \right) = \bigcup_{i=1}^{C_{N,l}} B \cap S_i, \quad (2.70)$$

where

$$B \cap S_i = \{ \omega \in \Delta^N : V_{\mathcal{V}}(\omega) > \varepsilon \text{ and } I_i^l(\omega) \text{ is active in } x(\omega) \}. \quad (2.71)$$

With the same argumentation as in the proof of [44, Theorem 3.3] all of the events $B \cap S_i$ have the same probability mass.

Define

$$V_i(\omega) := \mathbb{P} \{ \delta \in \Delta : I_i^l(\omega) \text{ active in } x(\omega) \text{ and } f(x(\omega), \delta) > 0 \}, \quad (2.72)$$

then

$$B \cap S_i = \{ \omega \in \Delta^N : V_i(\omega) > \varepsilon \}. \quad (2.73)$$

Consider now only $i = 1$ and define for $\omega_l := (\delta^{(1)}, \dots, \delta^{(l)})$

$$F_1(\alpha) := \mathbb{P}^l \{ \omega_l \in \Delta^l : V_1(\omega_l) \leq \alpha \} \quad (2.74)$$

the probability distribution of the random variable V_1 . The conditional probability

$$\mathbb{P}^N \{ S_1 \mid V_1(\omega) = \alpha \} \quad (2.75)$$

is the conditional probability that event S_1 occurs, i.e., that the first l constraint realizations $I_i^l(\omega)$ are active in $x(\omega)$, given that the violation probability is α . It holds that $x(\omega)$ satisfies all constraints arising from realizations $I_i^l(\omega)$ with equality and the constraints $\delta^{(l+1)}, \dots, \delta^{(N)}$ with strict inequality, i.e., $f(x(\omega), \delta^{(j)}) < 0$ for $j = l+1, \dots, N$. Since the realizations

$\delta^{(j)}$ are drawn independently it holds that

$$\mathbb{P}^N\{\mathcal{S}_1 \mid V_1(\omega) = \alpha\} \quad (2.76)$$

$$= \mathbb{P}\left\{I_l^1(\omega) \text{ active in } x(\omega), f(x(\omega), \delta^{(j)}) < 0, \quad (2.77)$$

$$j = l + 1, \dots, N \mid V_1(\omega) = \alpha\} \quad (2.78)$$

$$= (1 - \alpha)^{N-l}. \quad (2.79)$$

Deconditioning with respect to $V_1(\omega)$ it follows

$$\mathbb{P}^N\{\mathcal{S}_1\} = \int_0^1 (1 - \alpha)^{N-l} dF_1(\alpha) \quad (2.80)$$

and with the same argumentation as in the proof of [44, Theorem 3.3] we obtain that $F_1(\alpha) = \alpha^l$. The proof concludes as the proof of Theorem 3.3 and the rest is omitted here because the argumentation carries over verbatim from the proof of Theorem 3.3. This proves the lemma for the case that exactly l constraint realizations are active in $x(\omega)$.

If this is not the case, i.e., if with a nonnegative probability less than l constraint realizations can be active in $x(\omega)$, the concept of “active” constraint realizations has to be extended. We propose to employ similar techniques as in Part 2 of the proof of [49, Theorem 2.4]. There the concept of active constraints is extended as follows: A “*ball-vertex*” is a ball centered at the vertex $x(\omega)$ with radius such that the ball is fully contained in the feasible region for all constraint realizations except for $l - 1$ constraint realizations. The concept of active constraints is then extended to constraints that are active for the ball, i.e., constraints that are “touched” by the ball in the sense that the ball has nonempty intersection with a hyperplane arising from the constraint. With this construction it is guaranteed that the ball-vertex has active constraints arising from at most l constraint realizations and if the radius of the ball is zero, then we are in the case as above. The

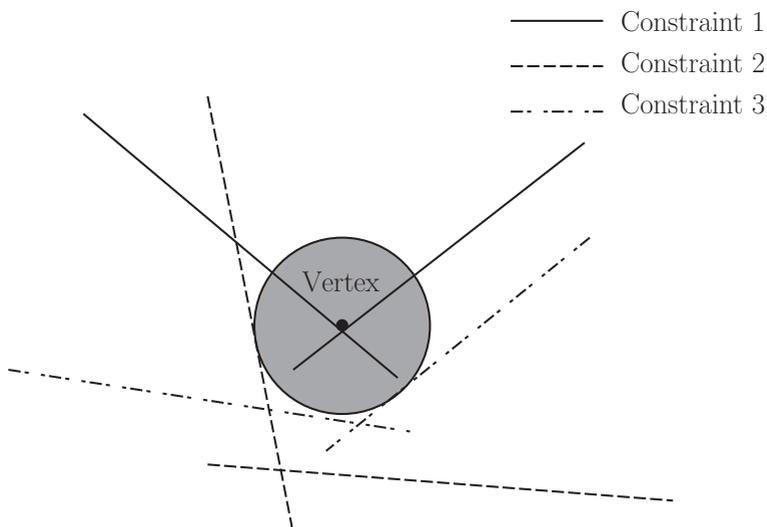


Figure 2.2.: In this figure a vertex and the corresponding ball vertex with three active constraints arising from three different constraint realizations is depicted. In this case *heating* and *cooling* techniques from [49] can be applied to randomly perturb the constraints so that exactly l constraint realizations are active for the ball.

above proof carries over more or less verbatim for the ball-vertex when it touches exactly l constraint realizations and, hence, is omitted here.

However, it can happen that this is not the case either, i.e., the ball touches more than l active constraint realizations as depicted in Figure 2.2 (cf. [49, Figure 3.4] for the same problem but with general nonlinear constraints). In this case so-called *heating* and *cooling* techniques as proposed in Part 2 of the proof of Theorem 2.4 in [49] can be employed. Here the constraints are first randomly perturbed in a process that is called heating so that the case in Figure 2.2 will only occur on constraint realizations that have a probability mass zero. Then one can show that with a similar argumentation as in [49] a process called cooling can be employed to recover the original vertex from the heated problem. The proof of this statement

has to be slightly modified, since in [49] optimal solutions of RCPs and not general vertices are considered. However, for general linear problems it holds that for any vertex there exists a cost direction such that the vertex is the optimal solution under this cost direction [100, Theorem 4.6, page 95]. Hence, for fixed realization ω it is possible to construct a cost direction for $x(\omega)$ such that it is the optimal solution under this cost. In the proof of the cooling property in Equation (3.12) in [49] this cost has to be considered. Once the heating and cooling properties are established, with the same argumentation as in [49] the violation probability of the vertex of the original problem can be recovered from the violation probability of the heated problem where the constraints cannot be as in Figure 2.2. All in all the construction and the proofs in [49] are rather technical and since they carry over almost verbatim, we omit the complete proof here. \square

We conclude this section with a result on the violation probability of a convex combination of n vertices of the feasible region of $\text{LP}[\omega]$.

Lemma 2.10 *Let $x_1(\omega), \dots, x_m(\omega)$ be vertices of the feasible region $\text{Sat}(\omega)$ of $\text{LP}[\omega]$ that satisfies Assumption 4. Let $N > ml$ and let $x_\lambda(\omega)$ be a convex combination of the vertices $x_1(\omega), \dots, x_n(\omega)$, i.e.,*

$$x_\lambda(\omega) = \sum_{i=1}^m \lambda_i x_i(\omega), 0 \leq \lambda_i \leq 1 \text{ and } \sum_{i=1}^m \lambda_i = 1. \quad (2.81)$$

Then it holds for $x_\lambda(\omega)$

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{\text{conv}}^*(\omega) > \varepsilon \} \leq \Phi(\varepsilon; ml - 1, N), \quad (2.82)$$

where

$$V_{\text{conv}}^*(\omega) = \mathbb{P} \{ \delta \in \Delta : f(x_\lambda(\omega), \delta) > 0 \} \quad (2.83)$$

is the probability that the convex combination $x_\lambda(\omega)$ becomes infeasible under the next constraint realization.

PROOF. Define $x_i^* := x_i(\omega)$. The proof proceeds in two steps. We first show that

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{conv}^*(\omega) > \varepsilon \} \leq \mathbb{P}^N \{ \omega \in \Delta^N : V_c^*(\omega) > \varepsilon \} \quad (2.84)$$

where

$$V_c^*(\omega) := \mathbb{P} \{ \delta \in \Delta : f(x_1^*, \delta) > 0 \text{ or } \dots \text{ or } f(x_m^*, \delta) > 0 \} \quad (2.85)$$

$$= \mathbb{P} \left\{ \bigcup_{j=1}^m \{ \delta \in \Delta : f(x_j^*, \delta) > 0 \} \right\}. \quad (2.86)$$

and then we show that

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_c^*(\omega) > \varepsilon \} \leq \Phi(\varepsilon; nl - 1, N). \quad (2.87)$$

Proof of (2.84): Because of the convexity of f we have that

$$f(x_\lambda, \delta) = f \left(\sum_{i=1}^m \lambda_i x_i^*, \delta \right) \leq \sum_{i=1}^m \lambda_i f(x_i^*, \delta). \quad (2.88)$$

It also holds that

$$\{ \delta : f(x_\lambda, \delta) > 0 \} = \Delta \setminus \{ \delta : f(x_\lambda, \delta) \leq 0 \}. \quad (2.89)$$

From (2.88) it follows that if $\sum_{i=1}^m \lambda_i f(x_i^*, \delta) \leq 0$ also $f(x_\lambda, \delta) \leq 0$ and, hence,

$$\{ \delta : \sum \lambda_i f(x_i^*, \delta) \leq 0 \} \subset \{ \delta : f(x_\lambda, \delta) \leq 0 \}. \quad (2.90)$$

Then, it follows that

$$\begin{aligned}
 \{\delta: f(x_\lambda, \delta) > 0\} &\stackrel{(a)}{\subset} \{\delta: \sum \lambda_i f(x_i^*, \delta) > 0\} \\
 &\stackrel{(b)}{\subset} \{\delta: f(x_1^*, \delta) > 0 \text{ or } \dots \text{ or } f(x_m^*, \delta) > 0\} \quad (2.91) \\
 &= \bigcup_{i=1}^n \{\delta: f(x_i^*, \delta) > 0\},
 \end{aligned}$$

where (a) follows from equation (2.89) and inclusion (2.90) and (b) follows from the fact that all $\lambda_i \geq 0$ and, hence, for the sum to be strictly positive at least one of the summands has to be strictly positive. From the inclusions (2.91) it follows that

$$V_{conv}^*(\omega) = \mathbb{P}\{\delta: f(x_\lambda, \delta) > 0\} \quad (2.92)$$

$$\leq \mathbb{P}\{\delta: f(x_1^*, \delta) > 0 \text{ or } \dots \text{ or } f(x_m^*, \delta) > 0\} \quad (2.93)$$

$$= V_c^*(\omega) \quad (2.94)$$

and if $V_{conv}^*(\omega) > \varepsilon$ it follows that also $V_c^*(\omega) > \varepsilon$ and so we obtain (2.84)

$$\mathbb{P}^N\{\omega \in \Delta^N: V_{conv}^*(\omega) > \varepsilon\} \leq \mathbb{P}^N\{\omega \in \Delta^N: V_c^*(\omega) > \varepsilon\}. \quad (2.95)$$

Proof of (2.87): Consider the n product of the feasible region

$$\mathcal{S} := \underbrace{\text{Sat}(\omega) \times \dots \times \text{Sat}(\omega)}_{n \text{ times}}. \quad (2.96)$$

It can be easily checked that the direct product of simple polytopes is again a simple polytope and, hence, \mathcal{S} is simple. The point $x := (x_1^*, \dots, x_m^*)$ is a vertex of \mathcal{S} , since a vertex of \mathcal{S} is given as an m -vector of vertices of

Sat(ω). The product feasible region \mathcal{S} can also be written as

$$\mathcal{S} = \left\{ (x_1, \dots, x_m) \in \mathbb{R}^{ml} : \right. \quad (2.97)$$

$$\left. g(x_1, \dots, x_m) := \max_{i=1, \dots, m} \{f(x_i, \delta^{(j)})\} \leq 0, j = 1, \dots, N \right\} \quad (2.98)$$

with piecewise affine and convex function g . We can apply Lemma 2.9 to the region \mathcal{S} and its vertex x to obtain

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{\mathcal{S}}^*(\omega) > \varepsilon \} \leq \Phi(\varepsilon; ml - 1, N), \quad (2.99)$$

where

$$V_{\mathcal{S}}^*(\omega) = \mathbb{P} \left\{ \delta \in \Delta : \max_{i=1, \dots, m} \{f(x_i^*, \delta)\} > 0 \right\} \quad (2.100)$$

$$= \mathbb{P} \{ \delta \in \Delta : f(x_1^*, \delta) > 0 \text{ or } \dots \text{ or } f(x_m^*, \delta) > 0 \} \quad (2.101)$$

□

2.4.2. Violation Probabilities of Inner and Integer Points

In this subsection we will combine the results on the violation probability of vertices of the feasible region of an RLP from the previous subsection with the well-known fact that each point in a polytope is the convex combination of the vertices of the polytope. This will allow us to provide bounds on the tails of the violation probability of arbitrary points in the random feasible region of an RLP.

Lemma 2.11 *If F is an l -polytope (i.e. a compact polyhedron in \mathbb{R}^l), then F is the convex hull of its extreme points. Furthermore, each point of F is the convex combination of atmost $l + 1$ extreme points.*

PROOF. Because of Minkowski's theorem for polytopes [100] it holds that each point is the convex combination of the vertices of F . From Caratheodory's theorem [58] it follows that each point is actually the convex combination of at most $l + 1$ vertices. \square

Definition 2.13 (Upper Bound on Vertices of $\text{Sat}(\omega)$) For $\text{LP}[\omega]$ denote by $\tilde{V}(\omega)$ the number of vertices of its feasible region $\text{Sat}(\omega)$. Define V to be the smallest integer such that

$$V \geq \text{ess sup}_{\omega \in \Delta^N} \tilde{V}(\omega) \quad (2.102)$$

i.e., the smallest integer number such that the number of vertices of $\text{Sat}(\omega)$ is less than or equal to V with probability one.

Since the number of constraint realizations N is finite, V will always be finite. See Remark 2.2 below for a general bound on the number of vertices of a simple d -polytope.

Theorem 2.2 Consider $\text{LP}[\omega]$ with feasible region $\text{Sat}(\omega)$ that satisfies Assumption 4 and let V be as in Definition 2.13. For each point $x_0 \in \text{Sat}(\omega)$ that is feasible for $\text{LP}[\omega]$ it holds that

$$\mathbb{P}^N \{ \omega \in \Delta^N : V^*(\omega, x_0) > \varepsilon \} \leq \binom{V}{l+1} \Phi(\varepsilon; s, N), \quad (2.103)$$

for $s := l(l+1) - 1$ and $N \geq s$. Here

$$V^*(\omega, x_0) := \mathbb{P} \{ \delta \in \Delta : f(x_0, \delta) > 0 \} \quad (2.104)$$

is the probability that the point x_0 becomes infeasible under the next constraint realization.

PROOF. Let $x_0 \in \text{Sat}(\omega)$ be a point in the feasible region $\text{Sat}(\omega)$. We know by Lemma 2.11 that x_0 can be expressed as the convex combination of $l + 1$

extreme points of $\text{Sat}(\omega)$. This will hold with probability one for every realization ω . It can happen, however, that the vertices in the convex combination change from realization to realization. So we cannot a priori determine which vertices will be needed in the convex combination. Therefore, we have to consider all possible combinations of $l + 1$ vertices: Number all vertices from $1, \dots, V$ and consider subsets of vertex indices $E^i(\omega)$ of cardinality $l + 1$ of $\{1, \dots, V\}$ and index them by $i = 1, \dots, C := \binom{V}{l+1}$. If for a realization ω $\text{Sat}(\omega)$ has less than V vertices, we recount the last vertex in the enumeration until we get to exactly V vertices. Define the event

$$A_i := \{ \omega \in \Delta^N : x_0 \text{ convex combination of vertices } E^i(\omega) \} . \quad (2.105)$$

Since x_0 will be the convex combination of some $l + 1$ extreme points by Lemma 2.11, we have that

$$1 = \mathbb{P}^N \{ \Delta^N \} = \mathbb{P}^N \left\{ \bigcup_{i=1}^C A_i \right\} . \quad (2.106)$$

From the proof of Lemma 2.10 it becomes apparent that the violation bound for the convex combination does not depend on the actual weights in the combination (i.e. the λ_j 's) but only on the extreme points involved. That is the reason why we restrict our studies here to only the subsets of extreme points and completely neglect the actual convex combination.

It follows that

$$\mathbb{P}^N\{V^*(\omega, x_0) > \varepsilon\} \stackrel{(a)}{=} \mathbb{P}^N\left\{(V^*(\omega, x_0) > \varepsilon) \cap \left(\bigcup_{i=1}^C A_i\right)\right\} \quad (2.107)$$

$$= \mathbb{P}^N\left\{\bigcup_{i=1}^C (V^*(\omega, x_0) > \varepsilon \cap A_i)\right\} \quad (2.108)$$

$$\leq \bigcup_{i=1}^C \mathbb{P}^N\{V^*(\omega, x_0) > \varepsilon \cap A_i\} \quad (2.109)$$

$$\stackrel{(b)}{\leq} \binom{V}{l+1} \Phi(\varepsilon; l(l+1) - 1, N). \quad (2.110)$$

In the above (a) follows from (2.106) and (b) from Lemma 2.10 and the fact that $C = \binom{V}{l+1}$. \square

Corollary 2.3 (Mixed-Integer RLPs) Consider a MI-RCP

$$\min c^\top x \quad (2.111)$$

$$\text{s.t. } f(x, \delta^{(j)}) \leq 0, \quad j = 1, \dots, N \quad (2.112)$$

$$x \in \Omega \subset \mathbb{R}^n \times \mathbb{Z}^d \quad (2.113)$$

with constraint function $f(x, \delta)$ that is either affine in x or piecewise affine and convex. Denote the maximum number of vertices according to Definition 2.13 of the feasible region by V and let $l = n + d$. Assume the feasible region $\text{Sat}(\omega)$ of the relaxed continuous LP $[\omega]$ satisfies Assumption 4 and that the optimal mixed-integer solution is unique. Then it holds for $s := l(l+1) - 1$ and $N \geq s$ that

$$\mathbb{P}^N\{\omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon\} \leq \binom{V}{l+1} \Phi(\varepsilon; s, N), \quad (2.114)$$

where $V_{mi}^*(\omega)$ is the violation probability of the optimal mixed-integer solution as in Definition 2.8.

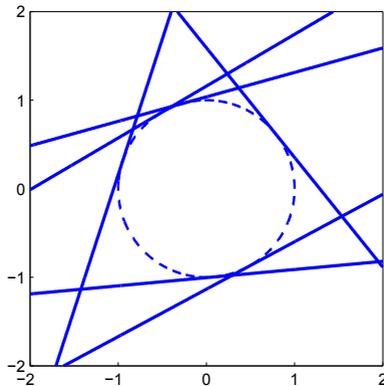


Figure 2.3.: Example of the feasible region with $N = 6$ in which also $V = 6$. The feasible region includes the sphere S^1 which was drawn for illustration purposes only, the linear constraints (2.117) are the solid lines.

PROOF. Let x_i^* be the optimal mixed-integer solution. Since $\text{Sat}(\omega) \neq \emptyset$ with probability one and the optimal solution is unique by assumption, it holds that

$$J^*(\omega) > J^*(\omega, \delta) \Leftrightarrow f(x_i^*, \delta) > 0, \tag{2.115}$$

i.e., the optimal mixed-integer objective gets worse under the next realization δ if and only if the optimal integer solution becomes infeasible under the new constraint. Now we apply the previous theorem to obtain the bound on the violation probability of the optimal mixed-integer solution. \square

Unfortunately in general $\text{LP}[\omega]$'s the number of vertices V of the feasible region will depend on the number N of constraint realization as can be seen from the following example.

Example 2.3: Vertices of the Feasible Region

Consider the unit sphere in \mathbb{R}^2 :

$$S^1 := \{(\sin(\theta), \cos(\theta)), \theta \in [0, 2\pi]\} . \tag{2.116}$$

Let $\delta \sim \mathcal{U}[0, 2\pi]$ be uniformly distributed on $[0, 2\pi]$ and let the constraint realizations for a $\delta^{(j)}$ be the half space including S^1 defined by the line that goes through $(\sin(\delta^{(j)}), \cos(\delta^{(j)}))$ and is tangential to S^1 . That is, constraint j is given through

$$\sin(\delta^{(j)})x + \cos(\delta^{(j)})y \leq 1 \tag{2.117}$$

and also consider the constraints $(x, y) \in [-2, 2]^2$. Then the feasible region is with probability one a simple polytope (compact and exactly 2 linear constraints intersect at each vertex with probability one). If we draw more than four constraints, each constraint can have an intersection with his left and right “neighbors” on the circle, so we can have exactly N vertices for the feasible region. See Figure 2.3 for an example with $N = 6$ and six vertices. ■

Remark 2.2 A general bound on the number of vertices in a simple polytope in dependence on the n faces is given by the upper bound theorem [81, 95, 133]. It states that the maximum number of vertices in a simple polytope in \mathbb{R}^l with n faces is bounded from above by

$$C(l, n) = \binom{n + \lfloor \frac{l+1}{2} \rfloor}{n-l} + \binom{n + \lfloor \frac{l+2}{2} \rfloor}{n-l} . \tag{2.118}$$

We will now consider a special case of random linear programs, in which the number of vertices of the feasible region *will not* depend on the number of constraint extractions.

2.5. Linear Programs with Random Right Hand Sides

In the last section we derived a bound on the tails of the violation probability of an optimal mixed-integer solution of an RLP that depends on the number of vertices of the feasible region of the corresponding relaxed continuous LP $[\omega]$. We saw in an example that the number of vertices can grow with the number of constraint realizations drawn, which can be undesirable since it worsens the tightness of the bounds.

In this section we consider RLPs with constraints of the form

$$f(x, \delta) = \max_{i=1, \dots, m} \left\{ a_i^\top x - g_i(\delta) \right\}, \quad (2.119)$$

where $g_i: \Delta \rightarrow \mathbb{R}$ is a lower semi-continuous function. The constraint $f(x, \delta) \leq 0$ is then equivalent to $Ax \leq g(\delta)$, where A has rows a_i^\top and $g(\delta)$ has rows $g_i(\delta)$ and the inequality is a vector inequality. Constraints of this form with uncertain right hand sides occur in many applications, for example in inventory control with uncertain demands or in the control of discrete time linear systems affected by additive disturbance [28].

Theorem 2.3 *Let all assumptions of Theorem 2.2 hold. Consider the random linear program*

$$\text{LP}[\omega]: \quad \min c^\top x \quad (2.120)$$

$$\text{s.t. } f(x, \delta^{(j)}) \leq 0, \quad j = 1, \dots, N \quad (2.121)$$

with $f(x, \delta) = \max_{i=1, \dots, m} \{ a_i^\top x - g_i(\delta) \}$. For any point x_0 that is feasible for LP $[\omega]$ it holds for $s := l(l+1) - 1$ and $N \geq s$ that

$$\mathbb{P}^N \{ \omega \in \Delta^N : V^*(\omega, x_0) > \varepsilon \} \leq \binom{V}{l+1} \Phi(\varepsilon; s, N), \quad (2.122)$$

where in this case V is independent of the number of constraint realizations N and only depends the constraint matrix A with rows a_i^\top .

PROOF. For constraints of the form as in (2.119) the number of vertices of the feasible region does not depend on the number of constraint realizations extracted. This can be seen as follows: Consider a single constraint $a_i^\top x - g_i(\delta) \leq 0$ and let

$$F_{ij} := \{x: a_i^\top x - g_i(\delta^{(j)}) = 0, j = 1, \dots, N\}, \quad (2.123)$$

be the faces of the realizations of this constraint. Then it follows that

$$\bigcap_{j=1}^N F_{ij} = \emptyset \quad (2.124)$$

with probability one. So all faces given by the same constraint index i for different random right hand side realizations have empty intersection. Let $x(\omega)$ be a vertex of the feasible region for a realization ω . Then there are exactly d constraints $a_i^\top x - g_i(\delta^{(j_i)})$ that are active in $x(\omega)$. We showed that for same constraint index i the constraints for different realizations $\delta^{(j)}$ cannot be active at the same time. It follows that for each active constraint i there is a j_i such that the constraint $a_i^\top x - g_i(\delta^{(j_i)})$ is active and all other constraints with index i are satisfied with strict inequality. In fact, this unique j_i is given through

$$j_i = \arg \max_{j=1, \dots, N} g_i(\delta^{(j)}). \quad (2.125)$$

The proof is then direct consequence of Theorem 2.2 and the fact that the number of vertices depends only on the constraint rows a_i . \square

Corollary 2.4 (MI-RLPs with Random Right Hand Side) *Consider a MI-RCP as in Corollary 2.3 with constraint function*

$$f(x, \delta) = \max_{i=1, \dots, m} \{a_i^\top x - g_i(\delta)\} \quad (2.126)$$

and all assumptions of Corollary 2.3. Then it holds for $s := l(l+1) - 1$ and $N \geq s$ that

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon \} \leq \binom{V}{l+1} \Phi(\varepsilon; s, N), \quad (2.127)$$

where V only depends on the constraint matrix A with rows a_i^\top .

2.5.1. Comparison to Robust Optimization

In this section we compare how linear problems with uncertain right hand side are treated in the robust optimization literature.

In [124] the case of *column-wise uncertainty* in a linear program is considered. The problem considered in that work is

$$\min c^\top x \quad (2.128)$$

$$\text{s.t. } Ax \leq b \quad \forall A \quad (2.129)$$

$$x \geq 0 \quad (2.130)$$

and the constraint matrix A has uncertain columns a^k confined to lie in closed convex sets K^k . Hence, the constraint $Ax \leq b \quad \forall A$ is equivalent to $\sum_{i=1}^d a^k x_k \leq b \quad \forall a^k \in K^k$. Then, in [124] it is shown that these constraints are satisfied if the constraint $A^* x \leq b$ with $a_{ik}^* = (\sup_{a^k \in K^k} a^k)_i$ is satisfied.

In the robust setting, uncertain right hand sides are a special case of the model above. This can easily be seen by transforming the constraint $Ax \leq g(\delta) \quad \forall \delta \in \Delta$ into $\tilde{A}\tilde{x} \leq 0$ with $\tilde{x} = [x, 1]$ and $\tilde{A} = [A, -g(\delta)]$. The robust constraint

$$Ax \leq g(\delta) \quad \forall \delta \in \Delta \quad (2.131)$$

is satisfied if

$$Ax \leq g^*, \quad (2.132)$$

where $g_i^* = \inf_{\delta \in \Delta} g_i(\delta)$. With the deterministic constraint g^* , robust constraint satisfaction can be guaranteed, but the found solution can be way too conservative. For example if $g(\Delta)$ is an ellipsoid, the point g^* will correspond to the lower left corner of the smallest box including the ellipsoid.

The *finite adaptability* approach [29, 33] can be applied to the case of uncertain right hand sides, although its main focus lies in robust multi-stage optimization. In this approach the uncertainty set $\mathcal{B} := g(\Delta)$ is “covered” by K partitions, represented by K vectors b_k such that for every $b \in \mathcal{B}$ there exists at least one b_k such that $b \leq b_k$. Given such a partition the linear problem with uncertain right-hand side is equivalent to

$$\min \max_{k=1, \dots, K} c^\top x_k \quad (2.133)$$

$$\text{s.t. } Ax_k \leq b_k, \quad k = 1, \dots, K. \quad (2.134)$$

For $K = 1$ this would correspond to (2.132). The degree of conservatism of finite adaptability can be lower than that of (2.132) but strongly depends on the partition. In [29] the authors show that finding an optimal partition with $K = 2$ is already NP-hard.

In [25] uncertain linear programs with *row-wise uncertainty* are considered, i.e., the rows a_i of the constraint matrix A are uncertain. In the case of right-hand side uncertainty, that means that all entries of a row of \tilde{A} are certain except for the last one. One assumption made in [25] is that the uncertainty is constraint-wise, i.e., the uncertainty set \mathcal{B} has the form

$$\mathcal{B} = \mathcal{B}_1 \times \dots \times \mathcal{B}_m \quad (2.135)$$

where $\mathcal{B}_i = \Pi_i(\mathcal{B})$ is the projection of the uncertainty set onto the components corresponding to the i -th row of \tilde{A} , and that these \mathcal{B}_i are convex and closed. For the case of uncertain right-hand side this would mean that

$$g(\Delta) = g_1(\Delta_1) \times \dots \times g_m(\Delta_m), \quad (2.136)$$

i.e., that $g_i(\delta) = g_i(\delta_i)$ only depends on the i -th component of the uncertainty realization δ .

If constraint-wise uncertainty is not given, [25] propose to instead consider the enlarged uncertainty set

$$\mathcal{B} = \text{conv}(\Pi_1(\mathcal{B})) \times \cdots \times \text{conv}(\Pi_m(\mathcal{B})). \quad (2.137)$$

It can be checked that solving the LP with uncertain right-hand side corresponds again to the conservative constraints (2.132).

This means that all the robust approaches will treat the case of uncertainty in the right hand side with a higher degree of conservatism.

2.6. Results

In order to demonstrate the applications of RLPs with uncertainty in the right hand side and the potential advantages the bounds in Theorem 2.3 and Corollary 2.4 have over the bounds in Theorem 2.1, we will now consider three examples.

2.6.1. Perturbed Integer Knapsack

We assume that we have d goods each with fixed and known weights $w_i > 0$. Instead of packing amount x_i of good i into the knapsack, we assume that a perturbed quantity $x_i + \delta_i$ with random variable δ_i gets into the knapsack. We do not assume any knowledge about the perturbation vector $\delta = (\delta_1, \dots, \delta_d)$ but assume that we are able to draw realizations from it. The objective is to maximize the worth of the goods in the knapsack given through a cost vector c subject to the constraints that the capacity must not be exceeded. Since we can only take integer multiples of the goods, the

resulting problem is an integer linear problem. The LP $[\omega]$ integer formulation is

$$\text{KP}[\omega] : \quad \max c^\top x \quad (2.138)$$

$$\text{s.t.} \quad \sum_{i=1}^d w_i \left(x_i + \delta_i^{(j)} \right) \leq \tilde{W}, \quad j = 1, \dots, N \quad (2.139)$$

$$x_i \geq 0, i = 1, \dots, d \quad (2.140)$$

$$x_i \in \mathbb{Z}. \quad (2.141)$$

This is an integer RLP with uncertain right-hand side. We assume that $x_i = 0, i = 1, \dots, d$ is feasible with probability one, hence,

$$\sum_{i=1}^d w_i \delta_i \leq \tilde{W} \quad (2.142)$$

with probability one.

For this example the feasible region of the continuous relaxation of KP $[\omega]$ will always be a $d + 1$ -simplex: There is one vertex $0 \in \mathbb{R}^d$ and d vertices

$$(0, \dots, 0, d_i, 0, \dots, 0), \quad i = 1, \dots, d \quad (2.143)$$

where

$$d_i = \max \left\{ x_i : w_i x_i \leq \tilde{W} - \max_{j=1, \dots, N} \sum_{i=1}^d w_i \delta_i^{(j)} \right\} \quad (2.144)$$

is the point in which the hyperplane $\{x \in \mathbb{R}^d : w^\top x = \tilde{W} - \sum_{i=1}^d w_i \delta_i^{(j)}\}$ intersects the $d - 1$ coordinate hyperplanes

$$F_k := \left\{ x \in \mathbb{R}^d : x_k = 0, x_j \geq 0 \quad j \neq k \right\} \quad k \neq i. \quad (2.145)$$

Thus the feasible region has $V = d + 1$ vertices. Let x_{int}^* be the optimal integer solution. We obtain the bound for the violation probability of x_{int}^*

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon \} \leq \binom{V}{d+1} \Phi(\varepsilon; d(d+1) - 1, N) \quad (2.146)$$

$$= \Phi(\varepsilon; d(d+1) - 1, N). \quad (2.147)$$

Note that for $d \geq 6$ this bound is much better than the bound $\Phi(\varepsilon; 2^d, N)$ we would get from the general MI-RCP result because then $d(d+1) - 1 < 2^d$.

Now, additionally to the weight constraint, we also consider a volume constraint

$$\sum_{i=1}^d v_i (x_i + \tilde{\delta}_i^{(j)}) \leq \tilde{V}, \quad j = 1, \dots, N. \quad (2.148)$$

We use the bound in Remark 2.2 with $n = d + 2$ to obtain an upper bound on the number of vertices of the relaxed feasible region.

For the perturbed knapsack problem with weight and volume constraints, we compared for different numbers of goods d how many samples were necessary to obtain

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon \} \leq \beta \quad (2.149)$$

for fixed ε and β . In order to guarantee (2.149), we numerically determined the smallest numbers N_1 and N_2 such that

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon \} \leq \Phi(\varepsilon; 2^d - 1, N_1) \leq \beta \quad (2.150)$$

and

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon \} \leq \binom{V}{d+1} \Phi(\varepsilon; d(d+1) - 1, N_2) \leq \beta. \quad (2.151)$$

The results for $\varepsilon = 0.1$, $\beta = 0.01$, and dimensions $d = 6, \dots, 10$ are plotted in Figure 2.4 and for $d = 10, \dots, 15$ are given in Table 2.1 .

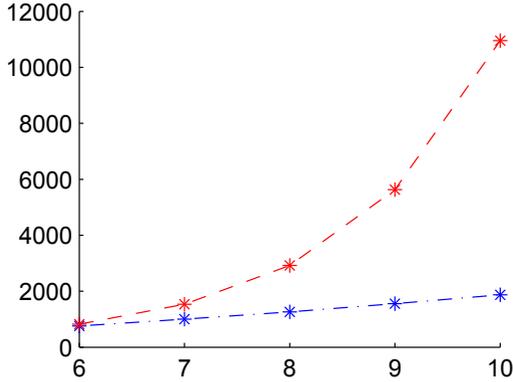


Figure 2.4.: Number of samples needed in the Knapsack example with weight and volume constraints to guarantee $\mathbb{P}^N\{V_{mi}^*(\omega) > \varepsilon\} \leq \beta$ using bound (2.150) depicted as red dashed line and number of samples needed to guarantee $\mathbb{P}^N\{V_{mi}^*(\omega) > \varepsilon\} \leq \beta$ using bound (2.151) depicted as dashed-dotted blue line, for $d = 6, \dots, 10$, $\varepsilon = 0.1$, and $\beta = 0.01$.

| | $d = 10$ | $d = 11$ | $d = 12$ | $d = 13$ | $d = 14$ | $d = 15$ |
|-------|----------|----------|----------|----------|----------|----------|
| N_1 | 10952 | 21486 | 42384 | 83930 | 166672 | 331686 |
| N_2 | 1875 | 2226 | 2600 | 3005 | 3432 | 3890 |

Table 2.1.: Number of samples N_1 needed to guarantee $\mathbb{P}^N\{V_{mi}^*(\omega) > \varepsilon\} \leq \beta$ using bound (2.150) and number of samples N_2 needed to guarantee $\mathbb{P}^N\{V_{mi}^*(\omega) > \varepsilon\} \leq \beta$ using bound (2.151) for $d = 10, \dots, 16$, $\varepsilon = 0.1$, and $\beta = 0.01$.

It can be seen that the number N_2 of samples needed to ensure (2.149) is of orders of magnitude lower than the number of samples N_1 given by the general MI-RCP bound. Furthermore, bound (2.151) holds for the violation probability $V_{mi}^*(\omega, x_0)$ of *any point* x_0 in the feasible region and not only

for the theoretical integer optimum. This fact is especially useful if for example numerical methods like cutting-plane techniques [100] are used to obtain the optimal solution of the integer problem. A numerical solver will in general return a solution x_0 that is not the optimal integer solution but within a certain prespecified tolerance of the integer solution. In such a case the bounds derived in the previous section can also be applied to x_0 and, hence, are useful to bound the violation probability of approximately optimal integer points.

2.6.2. Data-Driven Inventory Control

In this example we consider the single-station inventory control problem with fixed ordering costs and uncertain demands [28]. The variables are x_k , the stock available at the beginning of period k , u_k , the stock ordered at the beginning of period k , and w_k , the uncertain demand during period k . The available stock follows the recursive equation

$$x_{k+1} = x_k + u_k - w_k . \quad (2.152)$$

The cost in period k is

$$C(u_k) = \begin{cases} K + cu_k, & u_k > 0 \\ 0, & u_k = 0, \end{cases} \quad (2.153)$$

i.e., if we decide to order in period k , we will have to pay fixed costs $K > 0$ regardless of the quantity ordered. The shortage/holding costs are

$$r(x) = p \max\{0, -x\} + h \max\{0, x\} \quad (2.154)$$

and as in [28] we assume that $p > c$ since otherwise ordering in the last period would never be optimal.

We assume that the demand is random with unknown distribution, but we can either draw N realizations from the demand or are given N realizations e.g. from historic data. Similar to the robust inventory control problem introduced in [32] we formulate the inventory control problem with random demands as random mixed-integer linear problem:

$$\min_{u_k, v_k, y_k} \sum_{k=0}^{T-1} (cu_k + Kv_k + y_k) \quad (2.155)$$

$$\text{s.t. } y_k \geq h \left(x_0 + \sum_{i=0}^k (u_i - w_i^{(j)}) \right), \quad k = 0, \dots, T-1 \quad (2.156)$$

$$y_k \geq -p \left(x_0 + \sum_{i=0}^k (u_i - w_i^{(j)}) \right), \quad k = 0, \dots, T-1 \quad (2.157)$$

$$\text{for all } j = 1, \dots, N \quad (2.158)$$

$$0 \leq u_k \leq Mv_k, \quad v_k \in \{0, 1\}, \quad k = 0, \dots, T-1. \quad (2.159)$$

Here the $w_i^{(j)}$ are realizations of the random demands. It can be checked through simple transformations of the random constraints (2.156) and (2.157) that this problem is a mixed-integer random linear problem with uncertain right-hand sides.

The continuous decision variables are u_k and y_k for $k = 0, \dots, T-1$ and the binary variables are v_k for $k = 0, \dots, T-1$. Therefore the decision variables lie in the space $\mathbb{R}^{2T} \times \mathbb{Z}^T$ and the Helly dimension h of this space is $h = (2T+1)2^T - 1$. From Theorem 2.1 it follows for the optimal mixed-integer solution $(\mathbf{u}^*, \mathbf{y}^*, \mathbf{v}^*)$ of the inventory problem for $N_1 \geq h$ that

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon \} \leq \Phi(\varepsilon; (2T+1)2^T - 1, N_1) \quad (2.160)$$

where Δ is the space of all random demands over T time steps.

The relaxed linear programming version of the inventory problem has $6T$ constraints and $3T$ continuous (relaxed) decision variables. With Theorem

2.3 and Corollary 2.4 it follows that for $N_2 \geq 3T(3T + 1)$

$$\mathbb{P}^N \{ \omega \in \Delta^N : V_{mi}^*(\omega) > \varepsilon \} \leq \binom{V}{3T+1} \Phi(\varepsilon; 3T(3T + 1) - 1, N_2), \tag{2.161}$$

with V determined from Remark 2.2 with $d = 3T$ and $n = 6T$.

As in the previous section we numerically determined minimal N_1 and N_2 for both bounds in order to guarantee

$$\mathbb{P}^N \{ V_{mi}^* > \varepsilon \} \leq \beta \tag{2.162}$$

for $\varepsilon = 0.1$ and $\beta = 0.01$. The results are depicted in Table 2.2 and it can be seen that for $T > 6$ the bound (2.161) requires much less samples to guarantee $\mathbb{P}^N \{ V_{mi}^* > \varepsilon \} \leq \beta$. As in the previous example the bound derived in this section proves to be better than the general MI-RCP bound from Section 2.3, when the decision of the integer space is greater than six.

| | | | | |
|-------|---------|---------|---------|---------|
| | $T = 2$ | $T = 3$ | $T = 4$ | $T = 5$ |
| N_1 | 319 | 745 | 1723 | 3951 |
| N_2 | 993 | 2241 | 3967 | 6271 |
| | $T = 6$ | $T = 7$ | $T = 8$ | $T = 9$ |
| N_1 | 8971 | 20189 | 44991 | 99477 |
| N_2 | 9023 | 12383 | 16181 | 20617 |

Table 2.2.: Number of samples N_1 needed to guarantee $\mathbb{P}^N \{ V_{mi}^*(\omega) > \varepsilon \} \leq \beta$ using bound (2.160) and number of samples N_2 needed to guarantee $\mathbb{P}^N \{ V_{mi}^*(\omega) > \varepsilon \} \leq \beta$ using bound (2.161) for the inventory problem with $T = 1, \dots, 9$, $\varepsilon = 0.1$, and $\beta = 0.01$.

We conclude this example with a remark:

Remark 2.3 Even if the number of samples in Table 2.2 required to guarantee certain generalization probabilities seem to be high, it is not necessary to solve a mixed-integer problem with all of the constraints: The

general constraint $Ax \leq b^{(j)}$ $j = 1, \dots, N$ is equivalent to the constraint that $Ax \leq b^*(b^{(1)}, \dots, b^{(N)})$ where the i -th component of $b^*(b^{(1)}, \dots, b^{(N)})$ is

$$b_i^*(b^{(1)}, \dots, b^{(N)}) = \min_{j=1, \dots, N} b_i^{(j)} \quad (2.163)$$

is the minimum of the i -th components of the constraint realizations. So the complexity of the actual optimization problem does not increase.

2.6.3. Multi-Agent Coordination

In this section we will consider a two-dimensional model predictive multi-agent coordination problem similar to the model in [102]. We assume that there are two agents that plan over a horizon of length T in two-dimensional state space subject to the constraint that their states do not violate a minimum clearance. The model parameters are

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \quad (2.164)$$

with control inputs $u_t^i \in [u_{\min}^i, u_{\max}^i]$ and given initial states. There are $2T$ linear inequalities needed to model the input constraints for each agent.

The recursive state equation for each agent i is

$$\hat{x}_{t+1}^i = \mathbf{A}\hat{x}_t^i + \mathbf{B}u_t^i, t = 0, \dots, T-1. \quad (2.165)$$

The state recursion is a two-dimensional equality constraint so we need $4T$ inequality constraints to model the state recursion for each agent and additional 8 linear inequalities for each agent to describe the initial state equality constraint.

The collision avoidance constraints are that the states x_t^1 and x_t^2 of agents

1 and 2 do not come too close, i.e.,

$$\|x_t^1 - x_t^2\|_1 > \varepsilon_0, t = 0, \dots, T-1 \quad (2.166)$$

for the minimum clearance parameter ε_0 . We assume that the agents' states are perturbed by a disturbance w_t^i , i.e., $x_t^i = \hat{x}_t^i + w_t^i$. So we treat the collision avoidance constraints as random constraints. As in the previous chapter these collision avoidance constraints are modeled as mixed-binary linear constraints with the big- M method, resulting in $5T$ inequalities with $4T$ binary variables. The constraints are

$$\forall t = 1, \dots, T \quad \hat{x}_t^1(1) + w_t^1(1) - \hat{x}_t^2(1) - w_t^2(1) \geq \varepsilon_0 - M_o b_t^1 \quad (2.167)$$

$$\hat{x}_t^2(1) + w_t^2(1) - \hat{x}_t^1(1) - w_t^1(1) \geq \varepsilon_0 - M_o b_t^2 \quad (2.168)$$

$$\hat{x}_t^1(2) + w_t^1(2) - \hat{x}_t^2(2) - w_t^2(2) \geq \varepsilon_0 - M_o b_t^3 \quad (2.169)$$

$$\hat{x}_t^2(2) + w_t^2(2) - \hat{x}_t^1(2) - w_t^1(2) \geq \varepsilon_0 - M_o b_t^4 \quad (2.170)$$

$$\sum_{i=1}^h b_t^i \leq 3, \quad (2.171)$$

where M_o is as before some big number and $x_t^i(j)$ denotes the j -th coordinate of x_t^i and the same for w_t^i .

For the relaxation we need additional $2T$ linear inequalities to model that the relaxation of the binaries lies in the interval $[0, 1]$. The goal of the agents is to reach a certain goal position $Z^i \in \mathbb{R}^2$.

The decision variables lie in $\mathbb{R}^{2T} \times \mathbb{Z}^{4T}$ and, hence, the Helly dimension of the decision space is $h = (2T + 1)2^{4T}$ and the dimension of the relaxed decision space is $l = 6T$. The number of linear inequalities in the relaxation of the model predictive control problem with two agents is

$$\underbrace{4T}_{\text{input constraints}} + \underbrace{8T}_{\text{state constraints}} + \underbrace{16}_{\text{start value}} + \underbrace{7T}_{\text{collisions}} = 19T + 16. \quad (2.172)$$

| | $T = 1$ | $T = 2$ | $T = 3$ | $T = 4$ | $T = 5$ |
|-------|---------|---------|---------|---------|-----------|
| N_1 | 658 | 13680 | 290672 | 5919824 | 115424336 |
| N_2 | 1559 | 6031 | 13543 | 24123 | 37785 |

Table 2.3.: Number of samples N_1 needed to guarantee $\mathbb{P}^N\{V_{mi}^*(\omega) > \varepsilon\} \leq \beta$ using bound (2.173) and number of samples N_2 needed to guarantee $\mathbb{P}^N\{V_{mi}^*(\omega) > \varepsilon\} \leq \beta$ using bound (2.174) for the two-dimensional model predictive control problem for two agents with planning horizon lengths $T = 1, \dots, 5$, and $\varepsilon = 0.1$, and $\beta = 0.01$.

For the MI-RCP bound we have for $N_1 \geq (2T + 1)2^{4T} - 1$

$$\mathbb{P}^N\{V_{mi}^* > \varepsilon\} \leq \Phi(\varepsilon; (2T + 1)2^{4T} - 1, N_1) \quad (2.173)$$

and for the MI-RLP bound for $N_2 \geq 6T(6T + 1) - 1$

$$\mathbb{P}^N\{V_{mi}^* > \varepsilon\} \leq \binom{V}{6T + 1} \Phi(\varepsilon; 6T(6T + 1) - 1, N_2) \quad (2.174)$$

with V obtained from the upper bound theorem (Remark 2.2). In this example the violation probability is

$$V_{mi}^* = \mathbb{P}\{\delta \in \Delta: \|x_1^1 - x_1^2\|_1 \leq \varepsilon_0 \text{ or } \dots \text{ or } \|x_T^1 - x_T^2\|_1 \leq \varepsilon_0\}, \quad (2.175)$$

the probability that there will be a collision in time steps $1, \dots, T$. The random parameter $\delta = (w_1^1, w_1^2, \dots, w_T^1, w_T^2)$ is the joint disturbance acting on both agents over the planning horizon.

As in the previous two examples we numerically determined the N_1 and N_2 that ensure

$$\mathbb{P}^N\{V_{mi}^* > \varepsilon\} \leq \beta. \quad (2.176)$$

Remark 2.4 Since in this example there are many linear inequalities, V can become very big. We encountered numerical difficulties with the de-

termination of N_2 since $\binom{V}{6T+1}$ can become bigger than the largest floating point number although the right-hand side of (2.174) is actually much smaller. In order to circumvent this problem, we employed a more conservative bound along the lines of [44, Section 5]. We will briefly sketch it here.

For the binomial coefficient $\binom{V}{6T+1}$ the well-known inequality

$$\binom{V}{6T+1} \leq \left(\frac{eV}{6T+1} \right)^{6T+1} \quad (2.177)$$

holds. For the cumulative binomial distribution, Chernoff's inequality [54]

$$\Phi(\varepsilon; s, N) \leq \exp\left(-\frac{(N\varepsilon - s)^2}{2N\varepsilon}\right) \quad (2.178)$$

holds for $N\varepsilon \geq s$. Putting these two together, we obtain

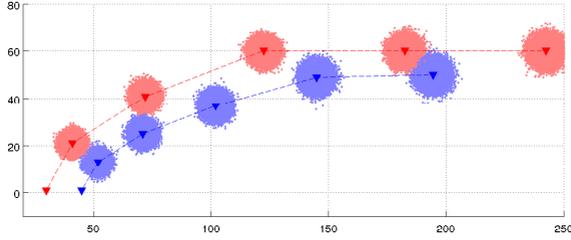
$$\begin{aligned} & \binom{V}{6T+1} \Phi(\varepsilon; 6T(6T+1) - 1, N_2) \\ & \leq \left(\frac{eV}{6T+1} \right)^{6T+1} \exp\left(-\frac{(N_2\varepsilon - 6T(6T+1) - 1)^2}{2N_2\varepsilon}\right) \end{aligned} \quad (2.179)$$

for $N_2\varepsilon \geq 6T(6T+1) - 1$. The right-hand side of (2.179) is less than or equal to β if and only if

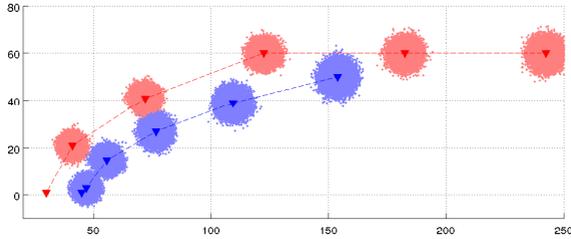
$$(6T+1) \ln\left(\frac{eV}{6T+1}\right) - \ln\beta \leq \left(\frac{(N_2\varepsilon - 6T(6T+1) - 1)^2}{2N_2\varepsilon}\right) \quad (2.180)$$

which follows from (2.179) by taking the logarithm on both sides of the inequality and rearranging the terms. Inequality (2.180) was used in the numerical experiments in this section to obtain N_2 such that (2.176) holds.

In Table 2.3 we depict the numbers of samples N_1 and N_2 needed to ensure that (2.176) holds, i.e., that a collision probability greater than ε occurs only with probability less than or equal to β . Here, we chose $\beta =$



(a) Example state trajectory without coordination constraints.



(b) Example state trajectory with coordination constraints.

Figure 2.5.: Example state trajectory for two agents without and with random constraints.

0.01 and $\varepsilon = 0.1$. It can be seen that for $T \geq 2$ bound (2.174) requires much less samples than bound (2.173) to ensure the same level of confidence.

Further, we considered a concrete example with $T = 5$, $\varepsilon_0 = 1$, $u_t^1 \in [-20, 20]$, $u_t^2 \in [-12, 12]$, $x_0^1 = [30, 1]$, $x_0^2 = [45, 1]$, $Z^1 = 60$, $Z^2 = 50$ and objective function $\sum_{t=0}^{T-1} |Z^i - x_t^i(2)|$, where $x_t^i(2)$ denotes the y -coordinate of agent i . The system disturbance for each agent in each direction was modeled to be correlated over time with covariance matrix $\Sigma \in \mathbb{R}^{H \times H}$, with entries $\Sigma_{st} = 0.5$ for $s \neq t$ and $\Sigma_{tt} = 2 + t$ to model the increase in uncertainty over time. The disturbance acting on agent i in direction j was drawn according to $(w_0^i(j), \dots, w_{T-1}^i(j)) \sim \mathcal{N}(0, \Sigma)$. Since we considered a planning horizon of length $T = 5$, we drew 37785 joint system distur-

bance samples. In Figure 2.5(a) we plotted the state trajectories of the two agents without any collision avoidance constraints. It can be seen that at time steps $t = 2$ and $t = 3$ there would be potential violations of the random constraint. The optimal objective value function without constraints was $J^* = 242$. In Figure 2.5(b) the state trajectories were determined with collision avoidance constraints. The optimal objective value in this case was $J^* = 282,6172$. For this state trajectory it holds that $\mathbb{P}^N\{V_{mi}^* > 0.1\} \leq 0.01$, i.e., the probability that we drew a realization of the joint disturbance such that the probability that the next disturbance sample will lead to a collision is higher than 10%, is less than 99%.

2.7. Conclusions

In this chapter we considered so-called mixed-integer random convex programs (MI-RCPs), convex optimization problems with a fixed number of random constraints with discrete and continuous decision variables. We proved explicit bounds on the tails of the probability that the optimal solution of an MI-RCP remains optimal for further, at the time of optimization yet unknown, realizations of the random constraints (the so-called violation probability). These bounds hold for the most general case of MI-RCPs with nonlinear convex random constraints. In order to sharpen the bounds we considered MI-RCPs in which the constraints are linear and the uncertainty only affects the right hand side of the linear constraints. For problems with this structure we used a different argumentation to prove significantly sharper bounds. The effect of this sharpening on the number of samples needed to guarantee a prespecified confidence of an optimal solution was studied in several different numerical examples.

Strengths The MI-RCP approach is a design approach to ex-ante immunize the solutions of convex programs with random constraints against the effects of uncertainty. It is not concerned with concrete numerical so-

lution techniques of an optimization problem but with designing a convex optimization problem in such a way that it can be guaranteed with high confidence that an optimal solution of the problem will remain optimal for further realizations of the random constraints. Hence, the focus lies on giving theoretical guarantees no matter how the optimal solution was found. The remarkable result of this chapter is that it is indeed possible to immunize convex programs with random constraints against the effect of uncertainty with very little further assumptions on the actual nature of the uncertainty.

The theoretical results on MI-RCPs are highly general and do not make any assumptions on the structure of the optimization problem besides convexity of the constraints and on the structure of the random parameters besides that one can draw independent and identically distributed realizations from it.

Current Limitations When the dimension of the integer decision space is high, due to the nature of the bounds on the tails of the violation probability, also a high number of samples will be needed in order to ensure that an optimal solution will be optimal for further realizations with given confidence. In those cases it is very likely that the bounds derived in this chapter are too conservative and it is therefore an open question how to sharpen and improve the bounds. Our results on MI-RCPs with linear constraints are a first step in this direction and future work will be concerned with finding tighter bounds.

Applications Mixed-integer problems have a wide range of applications in operations research, engineering and finance. MI-RCPs can be applied whenever an optimization problem with randomness or uncertainty in the constraints with mixed-integer decision variables has to be solved and the user is interested in strong theoretical guarantees about the generalization properties of an optimal solution. For example, MI-RCPs were applied to the optimal design of truss structures with uncertainty about external loads

and material stiffness properties with the goal of using as little material as possible in the structure [7]. In this thesis we considered a Knapsack problem with perturbed capacity constraints, an inventory control problem with random demands and a multi-agent MPC problem with collision avoidance constraints under disturbance.

As an application in finance, random convex programs with continuous decision variables have already been applied to data-driven portfolio optimization in [48]. When fixed transaction costs for buying assets in a portfolio are assumed the portfolio optimization problem becomes an mixed-integer problem [91] and the approaches on MI-RCPs presented in this thesis can be applied.

3. Distributed Optimization under Uncertainty

3.1. Introduction

In this chapter we tackle the problem of multiple agents coordinating under uncertainty in a *distributed manner*. For example consider a team of mobile robots where each robot's task is to reach a target way point and the robots need to coordinate in order to avoid collisions among each other and safely reach their targets. Or consider a distributed sensor network whose task is to track several targets as efficiently and accurately as possible. Both these applications have several key characteristics and challenges in common and in the following we will briefly highlight these.

Firstly, in both scenarios the state transitions are uncertain: For example for a differential drive robotic platform if the robot applies the action of turning 45° to the left, it will only do so with a certain probability and with some other nonnegative probability it will turn, say, 44° or 46° etc. In the sensor network the sensor will in general not be able to exactly predict the movements of the targets and will therefore have to resort to some probabilistic model for the target motion.

Secondly, the internal states of the systems are not directly accessible but have to be estimated from noisy observations: The robots in the multi-robot system will in general not have perfect knowledge about their absolute positions but will have to use noisy measurements, e.g. from laser scanners, to infer their own and the other robots' positions. In the sensor network a sensor will in general not be able to perfectly locate the tracked targets and there will always be the possibility that a target is within the sensing radius

of a sensor and due to sensor noise the sensor will not be able to correctly detect the target.

The *decentralized partially observable Markov decision process* (Dec-POMDP) model allows to model distributed planning under uncertainty in multi-agent systems taking the challenges above into account [27]. In the Dec-POMDP model each agent can carry out actions that have certain probabilistic effects on the states of the agents and their environment. A probabilistic modeling of the effects means that when an agent carries out an action, with a probability specified by the model, some effect occurs, with some probability a different effect occurs etc.

Furthermore, in this model the agents do not have exact, but uncertain knowledge about their current internal states and their environment. After execution of an action this knowledge can be updated by noisy, i.e., inexact observations. The goals or tasks of the agents (e.g. in the multi-robot scenario the robots want to reach their target locations as fast and fuel-efficiently as possible) are modeled in the Dec-POMDP model through a reward function. This reward function depends on the actions of the agents and the actions' effects on the states and the environment. The goal of the agents in the Dec-POMDP model is then to find actions that optimize the expected reward from the actions and their effects.

When optimizing the reward, the agents do not only take into account their current knowledge and the immediate reward resulting from the next action. They also incorporate future, yet unknown, information gathered through future observations and the possibility of executing future actions based on the future knowledge. The result of the optimization is then not only the next action that optimizes the short-term reward but a so-called *policy* that for all possible future observations within some time limit determines the optimal action for these observations. In other words, a policy maps possible future observations to a respective action that is optimal with respect to the knowledge resulting from the observations.

Unfortunately, studies of complexity have shown that the problem of

finding optimal policies in Dec-POMDPs is very hard to solve. In the special case of Dec-POMDPs, where all agents have the same goal, the complexity of finding an optimal policy is complete in NEXP, even if only two agents are considered [27]. Intuitively the complexity is related to the combinatorial examination of all possible future observations, actions and their effect on the states of the agents. It is therefore apparent that the size of the state space, i.e., the number of possible states, the size of the action space, i.e., the number of possible actions and the size of the observation space have a crucial impact on the complexity of finding optimal policies.

In order to reduce the complexity of finding policies in Dec-POMDPs *alternating optimization* approaches to find locally optimal policies were proposed in the past [98, 99]. In alternating optimization the policies of all agents except one are held fixed and the one agent computes its best response policy to the fixed policies of the others. After an agent has adapted its policy to the fixed policies of the other agents another agent is picked to do the same and so forth. A joint policy for all agents is then called locally optimal, if no agent can improve the overall reward by changing its own policy. The proceeding is similar to hill-climbing approaches in general combinatorial optimization and artificial intelligence in which all but one decision variable are held fixed at some value and the objective is optimized with respect to the remaining free variable [114].

The challenge when applying alternating optimization approaches from [98, 99] is that when an agent wants to find its best response to the fixed policies of the other agents, it is not clear which action the others will eventually execute, since, as we explained above, a policy maps future observation to actions. Hence, in order to coordinate its actions with the actions of the other agents and to find a best response policy, an agent has to account for *all possible future observations* the other agents could make within some time limit and for each find its best response.

This necessity of also considering all future observations for the other agents leads to an explosion of the space of possible “states” the agent

needs to consider and, hence, of the complexity of the problem each agent has to solve locally in order to compute a best response. To alleviate this increase in complexity, we propose in this chapter that each agent does not account for all possible future information the other agents might gather when it searches for its best response to the policies of the other agents but only the information they will *most likely* gather in the future. Each agent can easily compute these most likely information from the models specified in the Dec-POMDP and the policies of the other agents. It is intuitively apparent that when an agent considers only the most likely future information for the other agents, the complexity of the problem it has to solve locally is reduced significantly. In fact, we prove in this chapter that each agent when solving the local best response problem only has to account for the effect of its actions on the state space directly linked to itself and the future observations of the other agents only play a marginal role. Furthermore, we prove in this chapter that if the agents employ a coordination scheme based on alternating optimization as described above, for which it is guaranteed that the expected reward is monotonically increasing, and under the assumption that always the most likely information will be gathered when solving the local best response problems by the agents, then convergence to a locally optimal joint policy for all agents can always be guaranteed. We provide empirical evidence in experiments and simulations that the most likely information approach leads to a significant improvement of runtime compared to other approaches in which all information is considered.

This chapter is structured as follows: In Section 3.2 we formally define the Dec-POMDP model for planning under uncertainty, introduce its simplification, the network distributed POMDP model, and describe how existing approaches find locally optimal policies in these models. In Section 3.3 we formalize the concept of most likely information in POMDPs and describe how this concept can be applied in multi-agent planning. Furthermore, we prove the resulting complexity reduction and finite convergence of alternating optimization planning with the most likely information con-

cept. In Section 3.4 we provide concrete algorithms on how to incorporate the most likely information approach in planning. In Section 3.5 we describe how the concepts of the previous sections can be extended to continuous planning and information spaces. In Section 3.6 we critically discuss the most likely information paradigm and address related work. In Section 3.7 we present empirical results obtained from experiments in a sensor network and a multi-robot system and conduct comparisons with other state-of-the-art planning techniques.

3.2. Problem Formulation

The problem addressed in this chapter is the search for joint policies that maximize a joint cumulative expected reward in a multi-agent setting. Such systems can be modeled as *Decentralized Partially Observable Markov Decision Processes* (Dec-POMDP).

Definition 3.1 (Dec-POMDP) A Dec-POMDP is a tuple

$$\langle \mathcal{X}, \mathcal{U}, P, \Omega, O, R, b_0 \rangle, \quad (3.1)$$

where

- $\mathcal{X} := \mathcal{X}^1 \times \dots \times \mathcal{X}^M \times \mathcal{X}^u$ denotes the state space of the system consisting of local state spaces \mathcal{X}^i for each agent $i \in \{1, \dots, M\}$ and the space of unaffected world states \mathcal{X}^u . All spaces are assumed to be discrete and finite in this model. A joint state $x \in \mathcal{X}$ subsumes the current state of the whole system, comprised of the states of all agents and the world state.
- $\mathcal{U} := \mathcal{U}^1 \times \dots \times \mathcal{U}^M$ denotes the space of possible actions available to the system consisting of the discrete action spaces \mathcal{U}^i of each agent i . An action $u \in \mathcal{U}$ is called joint action.

- $p(x_{t+1}, x_t, u_t)$ is the transition function for the probabilistic transition from system state $x_t \in \mathcal{X}$ to system state $x_{t+1} \in \mathcal{X}$ under joint action u_t . It is defined as

$$p(x_{t+1}, x_t, u_t) := \mathbb{P}\{x_{t+1} = x_{t+1} | x_t = x_t, u_t\} , \quad (3.2)$$

i.e., the probability that the system transitions from state $x_t = x_t$ to state $x_{t+1} = x_{t+1}$ given action u_t .

- $\Omega := \Omega^1 \times \dots \times \Omega^M$ is the discrete joint observation space. Each agent i only observes $\omega_t^i \in \Omega^i$ from its own observation space, i.e., the observations of all other agents are hidden to agent i .
- $o(\omega_t, x_t, u_t)$ is the observation function defined as

$$o(\omega_t, x_t, u_t) := \mathbb{P}\{\omega_t = \omega_t | x_t = x_t, u_t\} , \quad (3.3)$$

i.e., the probability of making observation $\omega_t = \omega_t$ given that the system is in joint state $x_t = x_t$ and the agents applied joint action u_t .

- $R: \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$, $(x_t, u_t) \rightarrow R(x_t, u_t) \in \mathbb{R}$ is the reward function that maps state-action combinations to real values.
- b_0 is an initial discrete probability distribution over the state space \mathcal{X} . It is assumed to be independent for all agents, i.e., b_0^i is the probability distribution over the local state space \mathcal{X}^i of agent i and the probability of being in joint state $x = (x^1, \dots, x^M, x^M)$ is $b_0(x) = b_0^M(x^M) \cdot \prod_{i=1}^M b_0^i(x^i)$. The initial distribution is known to all agents.

Remark 3.1 In the case of $M = 1$, i.e., if there is only one agent in the system, the model is called (single-agent) POMDP. For a general introduction to Markov Decision Processes in which the states are directly observable see [106], for an introduction to POMDPs [80] and to Dec-POMDPs [27, 72, 98].

In the Dec-POMDP model the agents cannot directly observe the joint state of the system but can only take noisy measurements and from these make inference about the state. Hence, all the knowledge the agents have are the actions they have executed so far and the observations they have made. The goal of the agents is to optimize the reward over some planning horizon of length T . Since when the agents plan, future measurements are not known yet, they need to optimize the expected reward and the result of the optimization will not be action strategy consisting of a sequence of actions but it will be a so-called *policy* π^i for each agent i . This policy gives an optimal action for each possible sequence of observations. For example for a sequence of observations some action may be optimal because the agent believes to be in some state with high probability and for some other sequence of observations another action might be optimal.

Since agent i does not know observations or actions of the other agents, the decision at each time step t can only be based on its local observation sequence $(\omega_0^i, \dots, \omega_t^i)$. Therefore, the behavior of an agent i is fully defined by a policy $\pi^i(\omega_0^i, \dots, \omega_t^i) = u_t^i \in \mathcal{U}^i$ that maps observation sequences to possible actions. For a given joint policy $\pi = (\pi^1, \dots, \pi^M)$ and an initial state distribution b_0 , the expected cumulative reward over T steps is

$$J_\pi(b_0) = \mathbb{E}_{x_0, T} \left\{ R_T(x_T) + \sum_{t=0}^T R(x_t, u_t) \mid b_0, \pi \right\}, \quad (3.4)$$

where $R(x_t, u_t)$ denote one-step rewards and $R_T(x_T)$ is some terminal reward. It can be shown that the probability distributions over joint states x_t and actions u_t , which are needed to compute $J_\pi(b_0)$, are fully determined by the initial state distribution and the joint policy [28]. Hence, the control objective is to find the joint policy $\pi^* = \arg \max_\pi (J_\pi(b_0))$ that maximizes the expected reward $J_\pi(b_0)$.

Unfortunately, the general problem to find the optimal policy in Dec-POMDPs is NEXP-complete and therefore not feasible for most systems

[27]. In this chapter we will therefore consider a special case of Dec-POMDPs called Network Distributed POMDPs (ND-POMDPs) that are motivated by making decisions under uncertainty in domains like sensor networks [99, 127].

Example 3.1: Sensor Network Target Tracking Example

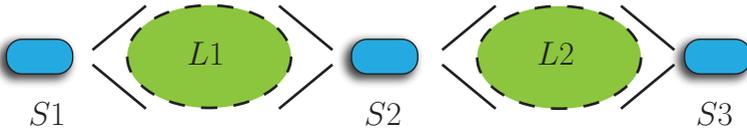


Figure 3.1.: Distributed sensor network for tracking targets in locations $L1$ and $L2$.

The following example is taken from [99, 127] and originates from the area of tracking a target in a sensor network [89]. See Figure 3.1 for the setup of the example. We assume that there are three sensors $S1, S2, S3$ and that sensors $S1$ and $S2$ can scan location $L1$ and $S2, S3$ can scan location $L2$, i.e., $S2$ can scan two locations but only one at a time. In order to locate a target in one of the locations both sensors that can scan the location need to *coordinate* and both scan the location at the same time.

There is a target that can be either in $L1$ or in $L2$ or absent and that has an uncertain motion, i.e., if it is present in one location with a certain probability it will stay in that location or leave it with a certain probability.

If the sensors choose to scan a location it is not guaranteed that they will make accurate observations. With a certain probability they can make a false-negative observation, i.e., the target is present in that location but the sensor does not see it, or a false-positive observation, i.e., the target is not present but the observation indicates to the sensor that it is. These observation probabilities only depend on the sensor

and, hence, are stochastically independent for different sensors. When a sensor scans a location it incurs a certain cost, if it does not scan at all it receives no cost. When two sensors scan a location with the target present at the same time, they receive a reward. The goal of the sensors is to track the target under as little costs as possible gaining a reward as high as possible.

The special characteristics of this scenario are that the transition probabilities of the target are independent of the actions of the sensors, the transitions and observation probabilities of the sensors are independent of the actions of the other sensors and the rewards are "local", i.e., when sensors $S1$ and $S2$ track a target in $L1$, $S3$ could be turned off and would not receive any reward or costs. ■

Motivated by the main characteristics of this scenario the networked distributed POMDP (ND-POMDP) model was introduced in [99]. In the following we will formally introduce the model.

Definition 3.2 (Transition and Observation Independence) We call a Dec-POMDP *transition independent* if the joint transition function from Definition 3.1 can be written as the product

$$p(x_{t+1}, x_t, u_t) = p^u(x_{t+1}^u, x_t^u) \cdot \prod_{i=1}^M p^i(x_{t+1}^i, x_t^i, u_t^i), \quad (3.5)$$

where p^u is the transition function for the unaffectable world states and p^i are the transition functions of the individual agents and $x_t = (x_t^1, \dots, x_t^M, x_t^u)$, $x_{t+1} = (x_{t+1}^1, \dots, x_{t+1}^M, x_{t+1}^u)$, and $u_t = (u_t^1, \dots, u_t^M)$.

A Dec-POMDP is *observation independent* when the joint observation function can be written as the product

$$o(\omega_t, x_t) = \prod_{i=1}^N o^i(\omega_t^i, x_t^i, x_t^u), \quad (3.6)$$

where o^i are the observation functions of the individual agents and $\omega_t = (\omega_t^1, \dots, \omega_t^M)$.

In a transition and observation independent model the state transition probabilities and observation probabilities are stochastically decoupled. We will further simplify this model by assuming that the reward function also has a special structure.

Definition 3.3 (Local Reward) The fact that in many multi-agent settings interaction only occurs locally, i.e., an agent will in general not interact with all other agents but only with a smaller subset, can be exploited by considering *local reward functions* in which each term models local interaction between a group of agents

$$R(x_t, u_t) = \sum_{I \subset \mathcal{I}} R_I(x_t^{i_1}, \dots, x_t^{i_k}, u_t^{i_1}, \dots, u_t^{i_k}), \quad (3.7)$$

where $\mathcal{I} = \{\{i_1, \dots, i_k\} \subset \{1, \dots, M\}\}$ refers to all subsets of agents. Each part of the sum only depends on the states and actions of some of the agents and the unaffected world state. When there is not interaction in a certain subset I of agents set $R_I \equiv 0$.

Based on the local reward function we can now study which agent interacts with which others agents by means of an interaction graph.

Definition 3.4 (Interaction Hypergraph) Define the *interaction hypergraph* as a graph $G = (A, E)$ in which the vertices A are the agents and there is an edge between two agents i_1 and i_2 if there exists an $I = \{i_1, \dots, i_k\} \subset \{1, \dots, M\}$ such that $\{i_1, i_2\} \subset I$ and $R_I \neq 0$, i.e., if agent i_1 and i_2 have a reward function in common.

Based on the interaction hypergraph we define the neighbor of an agent to be the other agents with which it interacts. This could be all other agents

in the network as for example for sensor $S2$ in Example 3.1 or just a single agent as for sensors $S1$ and $S3$.

Definition 3.5 (Neighborhood) The *neighborhood* N^i of an agent i are the agents

$$N^i := \{j \in A: j \neq i \text{ and } (i, j) \in E\}, \quad (3.8)$$

i.e., the neighborhood of an agent i are all the other agents it interacts with via the reward function. Define the neighborhood state

$$x^{N^i} := (x^{i_1}, \dots, x^{i_k}) \quad (3.9)$$

as the states of all the agents $i_j \in N^i$ in the neighborhood of agent i .

Together with transition- and observation-independence, this definition allows to represent the expected reward (Equation (3.4)) as

$$J(b_0) = \sum J_I(b_0^I), \quad (3.10)$$

i.e., as the sum of cumulative rewards for agents that interact with each other. It was shown that each part of this sum only depends on the policies π^I of agents in the subset I [99]. Therefore, any change in policy of agent i only affects the reward in its local neighborhood J_I for $i \in I$.

Definition 3.6 (ND-POMDP) Define a *Network Distributed POMDP* as a Dec-POMDP with transition and observation independent model as in Definition 3.2 and local reward as in Definition 3.3.

Definition 3.7 (Belief State) We will call the discrete probability distributions over (joint) states *belief states* or *beliefs*.

3.2.1. Alternating Optimization for ND-POMDPs

Several approaches were proposed in the literature to find locally optimal policies of Dec-POMDPs by alternating optimization. An alternating optimization algorithm iterates through the agents according to some polling scheme and finds an optimal policy for each agent assuming that the policies of the other agents are held fixed. This iteration goes on until no agent can improve its policy with respect to the others anymore and a locally optimal joint policy is found. The "*Joint Equilibrium-based search for policies*" (JESP) algorithm [98] is an alternating optimization algorithm for general Dec-POMDPs whereas the "*Locally interacting distributed joint Equilibrium-based search for policies*" (LIDJESP) algorithm [99] extends JESP to ND-POMDPs.

The key idea of our approach can be adapted to any algorithm using alternating optimization in transition and observation independent models. However, since LIDJESP can be applied in conjunction with the approach introduced in this chapter, we will briefly outline its functionality.

Given an ND-POMDP the local cumulative reward J_I only depends on the behavior of the agents $i \in I$ in the neighborhood. This allows to split the planning problem into subproblems that represent the interaction in local neighborhoods. At the beginning all agents in a neighborhood exchange (random) initial local policies. To improve the joint policy, each agent holds the policies of the other agents in its neighborhood fixed and computes its best response to the fixed policies of the other agents in their neighborhood. Then, the expected gains for potential improvements are exchanged among the agents in a neighborhood and the agent achieving the largest gain is allowed to change its policy. This continues until no agent can further improve its policy and thus, the joint policy is in a local optimum.

In order to find a best response to fixed policies of other agents at every optimization step the agents need to solve a single-agent POMDP with an extended state space [98]. A state in this POMDP is a combination of the

local state x_t^i of agent i , the states x^{N^i} of all other agents in the neighborhood N^i of the agent, the unaffected world state x_t^u and the possible observation sequences $\omega_{1:t}^{N^i}$ the neighbors could make until time t . An element $e_t^i = (x_t^i, x_t^{N^i}, x_t^u, \omega_{1:t}^{N^i})$ of this space is called *episode*. The possible future observations of its neighbors have to be considered by the agent since it does not know which future observations its neighbors will make and, hence, which actions the neighbors will eventually carry out. So it has to account for all possible future observations of its neighbors. Thus, to obtain the best response in each optimization step, an optimal policy for a POMDP over episodes has to be found. The complexity of this computation depends on the dimension of the state and observation spaces of all other agents in N^i . Therefore, the complexity of the search for joint policies increases considerably with the number of agents within a local neighborhood.

In the following we will introduce a heuristic that will drastically reduce the dimension of the state space of the local POMDP an agent has to solve. More precisely, with the heuristic each agent i does not have to solve a POMDP over episodes but only a POMDP over its own local state space \mathcal{X}^i and the space of unaffected world states \mathcal{X}^u . Especially for algorithms that are based on alternating improvement of joint policies in which each agent could possibly be required to compute best responses to the policies of other agents several times, this represents a considerable speedup.

3.3. Alternating Optimization with Maximum Likelihood Observations (MLOs)

To reduce the complexity of alternating optimization in multi-agent systems, the key idea of the approach presented in this chapter is as follows: Instead of accounting for all possible future observations its neighbors could make and computing its best response for each observation sequence, the agent assumes *for planning purposes* that its neighbors will always make the observation that is most likely to occur in a probabilistic sense based

on the observation probabilities as explained below. Under this assumption the behavior of an agent's neighbors is deterministic and an optimal policy simplifies to a sequence of actions. We will show that under the most likely observations assumption and the resulting sequences of actions for the neighbors, the local optimization problem for each agent is a POMDP over only its own local state space \mathcal{X}^i .

3.3.1. Maximum Likelihood Observation (MLO)

In this section we will formally define the most likely observation in the context of an ND-POMDP model and demonstrate how it can be computed from a given belief.

Definition 3.8 (Maximum Likelihood Observation) Given a belief $b_t^{u_{t-1}^i}$ and an action u_{t-1}^i , the *maximum likelihood observation (MLO)* is defined as as follows:

$$\hat{\omega}_t^i = \arg \max_{\omega_t^i \in \Omega^i} \left\{ \mathbb{P} \left\{ \omega_t^i = \omega_t^i \mid b_t^{u_{t-1}^i}, u_{t-1}^i \right\} \right\}, \quad (3.11)$$

i.e., the maximum likelihood observation $\hat{\omega}_t^i$ is the observation that has the highest probability of being made by agent i at time step t given the belief $b_t^{u_{t-1}^i}$ over the states of agent i and the unaffected world states and action u_{t-1}^i .

Remark 3.2 Please note here that we implicitly assume that there is a lexicographic ordering on the observation space. So when there is a tie in the likelihood among two or more observations, the lexicographic criterion can be used as a tie-breaking rule.

The conditional probability in Equation (3.11) can be computed from an agent's action u_{t-1}^i , belief b_t^u , and the observation o^i of the agent

$$\begin{aligned}
 & \mathbb{P}\left\{\omega_t^i = \omega_t^i | b_t^{u_{t-1}^i}, u_t^i\right\} \\
 &= \sum_{x_{t+1}^i \in \mathcal{X}^i, x_{t+1}^u \in \mathcal{X}^u} \mathbb{P}\left\{\omega_t^i = \omega_t^i | x_{t+1}^i = x_{t+1}^i, x_{t+1}^u = x_{t+1}^u, u_{t-1}^i\right\} \\
 & \quad \cdot \mathbb{P}\left\{x_{t+1}^i = x_{t+1}^i, x_{t+1}^u = x_{t+1}^u | u_{t-1}^i\right\} \\
 &= \sum_{x_{t+1}^i \in \mathcal{X}^i, x_{t+1}^u \in \mathcal{X}^u} o^i(\omega_t^i, x_{t+1}^i, x_{t+1}^u, u_{t-1}^i) \cdot b_t^{u_{t-1}^i}(x_{t+1}^i, x_{t+1}^u).
 \end{aligned} \tag{3.12}$$

$b_t^{u_{t-1}^i}$ is the belief of agent i at time t and subsumes all the information agent i has at that time and depends only on the initial belief, the agent's actions up to time t and the observations the agent made. The belief can be recursively computed through

$$b_t^{u_{t-1}^i}(x_{t+1}^i, x_{t+1}^u) := \sum_{x_t^i \in \mathcal{X}^i, x_t^u \in \mathcal{X}^u} p^i(x_{t+1}^i, x_t^i, u_{t-1}^i) \cdot p^u(x_{t+1}^u, x_t^u) \cdot b_t^e(x_t^i, x_t^u) \tag{3.13}$$

where

$$b_t^e(x_t^i, x_t^u) = \frac{1}{c} o^i(\tilde{\omega}_{t-1}^i, x_t^i, x_t^u) \cdot b_{t-1}^{u_{t-1}^i}(x_t^i, x_t^u) \tag{3.14}$$

and c is a normalization constant

$$c := \sum_{x_t^i \in \mathcal{X}^i, x_t^u \in \mathcal{X}^u} b_t^e(x_t^i, x_t^u) \tag{3.15}$$

and $\tilde{\omega}_t^i$ the observation made in time instance $t - 1$ [80]. It is important to note here that since we consider transition and observation independent models, all beliefs of an agent will be independent of the states and actions of all other agents. Since further all observation probabilities for different

agents are independent, it follows that the determination of the MLO for agent i only depends on the observation model and the beliefs of agent i .

Remark 3.3 (MLO Actions) A policy π^i determines for a sequence of observations $\tilde{\omega}_{1:t}^i$ that were made by the agent up to time t the *next* action $u_t^i := \pi^i(\tilde{\omega}_{1:t}^i)$ that is optimal for this observation history. For time $t = 0$ based on the probability distribution b_0 the first optimal action $u_0^i := \pi^i(\langle \rangle)$ is given by the policy before any observations are made. Based on the resulting predicted belief $b_0^{u_0^i}$ the MLO $\hat{\omega}_1^i$ can be computed as described above. From the MLO the corresponding optimal action $u_1^i := \pi^i(\hat{\omega}_1^i)$ can be computed and then the next MLO $\hat{\omega}_2^i$ can be computed according to Equation (3.11). By iterating these computations, it is possible to obtain an MLO action sequence $u_{0:T}^i$ and a corresponding sequence of MLOs $\hat{\omega}_{1:T}^i$. See Algorithm 3 for details.

The following definition considers the effect the MLO assumption has on the resulting beliefs.

Definition 3.9 (Nominal Belief Trajectory) Consider an ND-POMDP, an initial discrete probability distribution b_0^i over a discrete state space \mathcal{X}^i of an agent and an initial probability distribution b_0^u over the space of unaffordable world states \mathcal{X}^u . For an MLO sequence $\hat{\omega}_{1:T}^i$ and a corresponding action sequence $u_{0:T}^i$, define the *nominal belief trajectory* to be the sequence of probability distributions

$$b_{0:T}^i := (b_0^i \cdot b_0^u, b_1^i, \dots, b_T^i), \quad (3.16)$$

where for $1 \leq t \leq T$

$$b_t^i(x_{t+1}^i, x_{t+1}^u) := \sum_{x_t^i \in \mathcal{X}^i, x_t^u \in \mathcal{X}^u} p^i(x_{t+1}^i, x_t^i, u_{t-1}^i) \cdot p^u(x_{t+1}^u, x_t^u) \cdot b_t^e(x_t^i, x_t^u) \quad (3.17)$$

where

$$b_t^e(x_t^i, x_t^u) = \frac{1}{c} o^i(\hat{\omega}_{t-1}^i, x_t^i, x_t^u, u_{t-1}^i) \cdot b_{t-1}^{u_{t-1}^i}(x_t^i, x_t^u) \quad (3.18)$$

and c is the normalization constant as in (3.15).

Remark 3.4 The nominal belief trajectory is the sequence of probability distributions obtained by iterated prediction with actions and filtering with maximum likelihood observations of the initial probability distribution over the state space of agent i and the unaffected world states [114, Chapter 15]. Please note that given actions and MLOs the belief trajectories are deterministic in belief space, i.e., in the space of all discrete probability distributions over the state space.

3.3.2. The Local POMDP

In this section, we show that the local problem for an agent of computing its best response to the fixed policies of its neighbors and the sequences of actions resulting from the MLO assumption is a POMDP over the local state space.

Proposition 3.1 *In an ND-POMDP the problem for agent i of finding a best response to the policies of its neighbors N^i when it is assumed that the neighbors make the most likely observation is a single-agent POMDP over the local state space \mathcal{X}^i and the space of unaffected world states \mathcal{X}^u .*

PROOF. We will show that the reward function for agent i only depends on the states of the agents, the world states and the actions of the agent. Since by assumption the model is transition and observation independent, the proposition is then proved. Given a stochastically independent initial joint belief b_0 , policies π^{N^i} for all neighbors, under the MLO assumption fixed sequences of actions result from the policies and, hence, the nominal belief trajectories $b_{0:T}^{N^i}$ are fixed. In the following we define $x_t^j :=$

3. Distributed Optimization under Uncertainty

$(x_t^{i_1}, \dots, x_t^i, \dots, x_t^k, x_t^\mu)$ as the joint state of agents in subset I , the unaffordable world states and joint actions $u_t^I := (u_t^{i_1}, \dots, u_t^i, \dots, u_t^k)$, respectively. The goal of agent i is to maximize the cumulative expected reward

$$\sum_{t=0}^T \left(\sum_{x_t^{N^i}, x_t^\mu} b_t(x_t^I, x_t^\mu) \cdot \sum_{I \in \mathcal{I}} R_I(x_t^I, x_t^\mu, u_t^I) \right), \quad (3.19)$$

where the actions u_t^j are fixed for all $j \in N^i$ because of the MLO assumption, the decision variables are the actions $u_{0:T}^i$, and the probability distributions b_t^i are computed according to the prediction (3.13) and update process (3.14). For each of the summands in the large round brackets it holds that

$$\begin{aligned} & \sum_{x_t^{N^i}, x_t^\mu} b_t(x_t^I, x_t^\mu) \cdot \sum_{I \in \mathcal{I}} R_I(x_t^I, x_t^\mu, u_t^I) \\ &= \sum_{x_t^i, x_t^\mu} b_t^i(x_t^i, x_t^\mu) \cdot \underbrace{\left(\sum_{x_t^{N^i}} \prod_{j \in N^i, j \neq i} b_t^j(x_t^j, x_t^\mu) \cdot \sum_{I \in \mathcal{I}} R_I(x_t^I, x_t^\mu, u_t^I) \right)}_{=: \tilde{R}(x_t^i, x_t^\mu, u_t^i)} \\ &= b_t^i(x_t^i, x_t^\mu) \cdot \tilde{R}(x_t^i, x_t^\mu, u_t^i), \end{aligned} \quad (3.20)$$

where $\tilde{R}_I(x_t^i, x_t^\mu, u_t^i)$ only depends on the actions of agent i since the actions of the other agents are fixed and, hence, their belief trajectories $b_t^j(x_t^j, x_t^\mu)$ deterministic. It follows that

$$\sum_{t=0}^T \sum_{x_t^{N^i}, x_t^\mu} b_t(x_t^I, x_t^\mu) \cdot \sum_{I \in \mathcal{I}} R_I(x_t^I, x_t^\mu, u_t^I) = \sum_{t=0}^T \sum_{x_t^i, x_t^\mu} b_t^i(x_t^i, x_t^\mu) \cdot \tilde{R}(x_t^i, x_t^\mu, u_t^i) \quad (3.21)$$

and the state space of the decision problem for agent i is only $\mathcal{X}^i \times \mathcal{X}^u$ with modified reward function \tilde{R} . \square

We have thus shown in Proposition 3.1 that the MLO assumption significantly reduces the complexity of the local problem agents have to solve in order to compute a best response.

3.3.3. Convergence

In this section we will show that if each agent makes the MLO assumption for its neighbors and a scheme is applied such that the agents are allowed to change their local policies such that the expected reward resulting from the joint policies is non-decreasing, we can always guarantee finite convergence of the coordination.

Definition 3.10 (Locally Increasing Coordination) A coordination algorithm using MLOs is called *locally increasing*, when alternately one of the agents in a neighborhood $N^i \cup \{i\}$ changes its policy π^k to a different policy $\tilde{\pi}^k$ such that for the corresponding respective sequences of actions derived from the MLO assumption $u_{0:T}^k$ and $\tilde{u}_{0:T}^k$ it holds that

$$\sum_{I \in \mathcal{I}} J_I(\tilde{u}_{0:T}^k, u_{0:T}^{(N^i \cup \{i\}) \setminus \{k\}}, b_0) \geq \sum_{I \in \mathcal{I}} J_I(u_{0:T}^k, u_{0:T}^{(N^i \cup \{i\}) \setminus \{k\}}, b_0), \quad (3.22)$$

where

$$J_I(u_{0:T}^k, u_{0:T}^{(N^i \cup \{i\}) \setminus \{k\}}, b_0) = \sum_{t=0}^T \sum_{x_t^{N^i}, x_t^i, x_t^u} b_t(x_t^I, x_t^u) \cdot R_I(x_t^I, x_t^u, u_t^I) \quad (3.23)$$

is the expected cumulative reward resulting from R_I if agent i executes the sequence of actions $u_{0:T}^i$ and the other agents execute the sequences $u_{0:T}^{N^i}$, respectively. This means that in a locally increasing coordination algorithm the agents change their policies such that the cumulative expected reward evaluated on the actions resulting from the MLO assumption is monotonically increasing.

Remark 3.5 The polling structure according to which agents are selected to improve the cumulative reward could for example be a fixed order as we will demonstrate in MLOFIXEDORDER in Algorithm 1 or a more smart “auctioning” method as MLOLIDJESP as in Algorithm 4.

The following proposition shows convergence for any locally increasing algorithm.

Proposition 3.2 *For an ND-POMDP, a locally increasing algorithm based on the MLO assumption converges after finitely many improvement rounds where an improvement round is when one agent changes its policy to a better one.*

PROOF. Since we modeled the system to be transition and observation independent, changes in the actions of agent i only cause changes in the belief b_i^j and not in the beliefs of the other agents. Hence, a change in actions of agent i only has an impact on the expected rewards J_I for $i \in I$. Therefore, even if changes to the local policy have been made, we have for the J_I with $i \notin I$

$$J_I(u_{0:T}^i, u_{0:T}^{N^i}, b_0) = J_I(\tilde{u}_{0:T}^i, u_{0:T}^{N^i}, b_0), \quad (3.24)$$

i.e., changes of the cumulative reward in a neighborhood do not have a negative impact on the cumulative reward of agents that do not belong to the neighborhood (see also [99]).

The overall cumulative reward consisting of the sum of all reward functions is thus monotonically increasing:

$$\sum_I J_I(\tilde{u}_{0:T}^i, u_{0:T}^{N^i}, b_0) \quad (3.25)$$

$$= \sum_{I,i \in I} J_I(\tilde{u}_{0:T}^i, u_{0:T}^{N^i}, b_0) + \sum_{I,i \notin I} J_I(\tilde{u}_{0:T}^i, u_{0:T}^{N^i}, b_0) \quad (3.26)$$

$$\geq \sum_{I,i \in I} J_I(u_{0:T}^i, u_{0:T}^{N^i}, b_0) + \sum_{I,i \notin I} J_I(\tilde{u}_{0:T}^i, u_{0:T}^{N^i}, b_0) \quad (3.27)$$

$$= \sum_{I,i \in I} J_I(u_{0:T}^i, u_{0:T}^{N^i}, b_0) + \sum_{I,i \notin I} J_I(u_{0:T}^i, u_{0:T}^{N^i}, b_0) \quad (3.28)$$

$$= \sum_I J_I(u_{0:T}^i, u_{0:T}^{N^i}, b_0). \quad (3.29)$$

Since the overall expected cumulative reward is bounded from above by the global optimum that can be found by a centralized optimizer, a locally increasing algorithm leads to a bounded, monotonically increasing sequence of cumulative rewards. Since the space of all joint actions is finite and discrete, there are only finitely many sequences of actions and corresponding cumulative rewards. Thus, we have a monotonically increasing finite sequence that is bounded from above and, hence, any locally increasing algorithm converges in finitely many steps. \square

Remark 3.6 Convergence is achieved when no agent can improve its policy with respect to the policies of its neighbors. However, when a distributed scheme for improvement is applied (like in [99] or MLOLIDJESP) in which the agent that can establish the best improvement in local neighborhood objective is allowed to change its policy there has to be some way for the agents to check for convergence. This means there has to be some way for the agents to decide when the algorithm has globally converged in the whole system. In [99] this is accomplished by the introduction of a termination counter that is locally stored in each agent. We refer the reader to [99] and Algorithm 4 for details.

Algorithm 1 MLOFIXEDORDER(i)

- 1: $\pi^i \leftarrow$ random policy,
 - 2: $(u_{0:T}^i, \hat{\omega}_{1:T}^i) \leftarrow$ COMPUTEMLO(π^i, b_0, T)
 - 3: Transmit $u_{0:T}^i$ and corresponding MLOs $\hat{\omega}_{1:T}^i$ to other agents
 - 4: Receive best responses $u_{0:T}^j$, MLOs $\hat{\omega}_{0:T}^j$ from agents $j \in \{1, \dots, i-1\}$
 - 5: Receive random responses $u_{0:T}^j$ and MLOs $\hat{\omega}_{0:T}^j$ from agents $j \in \{i+1, \dots, m\}$
 - 6: $(\pi^i, u_{0:T}^i) \leftarrow$ BESTRESPONSE($i, b_0, (u_{0:T}^{1:i-1}, u_{0:T}^{i+1:M}), \hat{\omega}_{0:T}^{\{1, \dots, M\} \setminus i}, T$)
 - 7: Transmit $u_{0:T}^i$ and MLOs $\hat{\omega}_{1:T}^i$ to agents $\{i+1, \dots, M\}$
 - 8: **return** $(\pi^i, u_{0:T}^i)$
-

3.4. Algorithms

Fixed Ordering In this section, we will first present the case that there is a fixed ordering on *all* agents given, i.e., the local neighborhoods are neglected and there is no “smart” improvement of the plans. We present this simpler version first in order to better emphasize the characteristics of planning with MLOs. In this algorithm each agent computes a best-response to the policies of the other agents only once.

Every agent starts first with a randomly generated policy and determines for this policy the MLOs and corresponding actions as described in previous sections and Remark 3.3 (Line 2 in Algorithm 1). These are then transmitted to all other agents in Line 3. Agent i receives actions and MLOs from the other agents (Lines 4 and 5) where the actions of agents $\{1, \dots, i-1\}$ are already optimized. This means that agents $\{1, \dots, i-1\}$ already computed their *best responses* while agents $\{i+1, \dots, M\}$ will optimize their plans based on the best responses of the agents with lower index. This is a hierarchical concept in which agent i has to adapt to the already optimized policies of agents $\{1, \dots, i-1\}$ and all agents j with $j > i$ have to adapt to the already optimized policy of agent i .

In line 6 the agent computes its own best response to the plans of the other agents. In order to do so it first determines the nominal belief tra-

Algorithm 2 BESTRESPONSE($i, b_0, b_0^u, b_0^{N^i}, u_{0:T}^{N^i}, \hat{\omega}_{1:T}^{N^i}, T$)

```

1: // compute nominal belief trajectories for neighbors
2: for all  $j \in N^i$  do
3:   for all  $t = 1, \dots, T$  do
4:      $b_t^j \leftarrow \text{MLBELIEFUPDATE}(b_{t-1}^j, u_{t-1}^j, \hat{\omega}_{t-1}^j)$ 
5:   end for
6: end for
7: COMPUTEVALUEFUNCTION( $i, b_0^i, b_0^u, b_{0:T}^{N^i}, u_{0:T}^{N^i}, 0, T$ )
8:  $\pi^i \leftarrow \text{COMPUTEPOLICY}(i, b_0^i, b_0^u, \langle \rangle)$ 
9:  $(u_{0:T}^i, \hat{\omega}_{1:T}^i) \leftarrow \text{COMPUTEMLO}(\pi^i, b_0, T)$ 
10: return  $(\pi^i, u_{0:T}^i)$ 

```

jectories for the others agents and then determines its best response by calling BESTRESPONSE. Here each agent first determines the nominal belief trajectories for its neighbors resulting from the MLO assumption and then calls COMPUTEVALUEFUNCTION and COMPUTEPOLICY to solve the POMDP over its own local state space and find its best response policy. The optimal policy is computed through dynamic programming (see Algorithms 6 and 7) similarly as in [98, 99]. The MLOs and actions corresponding to the resulting policy are then transmitted to agents $\{i + 1, \dots, M\}$ and the optimal policy is returned.

Smart Improvement Now we will briefly highlight in Algorithm 4 how planning with the most likely observation assumption can be combined with the LIDJESP algorithm from [99]. After exchanging action sequences and MLOs based on a randomly generated policy with the other agents in its neighborhood, agent i first determines the utility of the current joint plan in line 9. The joint plan consists of its own policy combined with the action sequences of the other agents and the utility is computed as described in the proof of Proposition 3.1. Afterwards, in line 10 the agent determines the utility of its best response policy and the improvement compared to the previously determined utility. The agent that can achieve the best improve-

Algorithm 3 COMPUTEMLO(π^i, b_0, T)

```

1:  $u_0^i \leftarrow \pi^i(\langle \rangle)$ 
2:  $b_1^{u_0^i} \leftarrow \text{PREDICTBELIEF}(b_0, u_0^i)$  // belief prediction according to Eq.
   3.13
3: for all  $t = 1, \dots, T$  do
4:    $best\_like \leftarrow 0$ 
5:   for all  $\omega_t^i \in \Omega^i$  do
6:      $prob \leftarrow 0$ 
7:     for all  $x_{t+1}^i \in \mathcal{X}^i, x_{t+1}^u \in \mathcal{X}^u$  do
8:        $prob \leftarrow^+ o^i(\omega_t^i, x_{t+1}^i, x_{t+1}^u) \cdot b_{t+1}^{u_0^i}(x_{t+1}^i, x_{t+1}^u)$ 
9:     end for
10:    if  $prob > best\_like$  then
11:       $best\_like \leftarrow prob$ 
12:       $\hat{\omega}_t^i \leftarrow \omega_t^i$ 
13:    end if
14:  end for
15:   $u_t^i \leftarrow \pi^i(\hat{\omega}_{1:t}^i)$ 
16:   $b_t^e \leftarrow \text{FILTERBELIEF}(\hat{\omega}_t^i, b_t^{u_t^i}, u_t^i)$  // belief update according to Eq.
   3.14
17:   $b_{t+1}^{u_t^i} \leftarrow \text{PREDICTBELIEF}(b_t^e, u_t^i)$  // belief prediction according to Eq.
   3.13
18: end for
19: return  $(u_{0:T}^i, \hat{\omega}_{1:T}^i)$ 

```

ment is determined and is allowed to change its policy. The convergence of MLOLIDJESP follows from Proposition 3.2 and the convergence of LIDJESP in [99].

Please note that after MLOLIDJESP has converged, each agent locally not only has a fixed action sequence but a complete policy that leads to the MLO action sequence. So although coordination is based on fixed action sequences and the local POMDP each agent solves has a reduced complexity than in other approaches it still results in full policies for all agents.

Algorithm 4 MLOLIDJESP(i)

```

1: Compute interaction hypergraph and  $N^i$ 
2:  $\pi^i \leftarrow$  random policy,  $u_{0:T}^i \leftarrow \pi^i(\hat{\omega}_{1:T}^i)$ 
3:  $prevVal \leftarrow 0$ 
4:  $d \leftarrow$  diameter of hypergraph,  $termCtr_i \leftarrow 0$ 
5: Exchange  $u_{0:T}^i$  with  $N^i$ 
6: while  $termCtr_i < d$  do
7:   for all  $x_0^i, x_0^{N^i}, x_0^u$  do
8:      $b_0^i(x_0^i, x_0^{N^i}, x_0^u) \leftarrow b_0^u(x_0^u) \cdot b_0^i(x_0^i) \cdot b_0^{N^i}(x_0^{N^i})$ 
9:      $prevVal \leftarrow^+ b_0^i(x_0^i, x_0^{N^i}, x_0^u)$ 
10:     $gain_i \leftarrow$  COMPUTEVALUEFUNCTION( $i, b_0^i, u_{0:T}^{N^i}, 0, T$ )  $- prevVal$ 
11:  end for
12:  if  $gain_i > 0$  then
13:     $termCtr_i \leftarrow 0$ 
14:  else
15:     $termCtr_i \leftarrow^+ 1$ 
16:  end if
17:  Exchange  $gain_i, termCtr_i$  with  $N^i$ 
18:   $termCtr_i \leftarrow \min_{j \in N^i \cup \{i\}} \{termCtr_j\}$ 
19:   $maxgain \leftarrow \max_{j \in N^i \cup \{i\}} \{gain_j\}$ 
20:   $winner \leftarrow \arg \max_{j \in N^i \cup \{i\}} \{gain_j\}$ 
21:  if  $maxgain > 0$  and  $i == winner$  then
22:     $(\pi^i, u_{0:T}^i) \leftarrow$  BESTRESPONSE( $i, b_0, \langle \rangle, u_{0:T}^{N^i}, 0, T$ )
23:  else if  $maxgain > 0$  then
24:    Receive  $u_{0:T}^{winner}$  from  $winner$  and update  $u_{0:T}^{N^i}$ 
25:  end if
26: end while
27: return  $\pi^i$ 

```

3.5. Extension to Continuous State Spaces

If the algorithms presented in this section are to be applied to POMDP models in which the agents have continuous state spaces, i.e., the spaces \mathcal{X}^i and \mathcal{X}^u are no longer discrete but continuous, and also the observation

Algorithm 5 MLBELIEFUPDATE($j, b_{t-1}^j, u^j, \hat{\omega}^j$)

```

1:  $\tilde{b}_t^{u^j} \equiv 0$ 
2: // belief prediction according to (3.13)
3: for all  $x_t^j, x_t^u$  do
4:   for all  $x_{t-1}^j, x_{t-1}^u$  do
5:      $b_t^{uj}(x_t^j, x_t^u) \leftarrow^+ p^u(x_t^u, x_{t-1}^u) \cdot p^j(x_t^j, x_{t-1}^j, u_{t-1}^j) \cdot b_{t-1}^j(x_{t-1}^j, x_{t-1}^u)$ 
6:   end for
7: end for
8: // filtering with MLO according to (3.14)
9: for all  $x_t^j, x_t^u$  do
10:   $b_t^e(x_t^j, x_t^u) \leftarrow o^j(\hat{\omega}_t^j, x_t^j, x_t^u) \cdot b_t^{uj}(x_t^j, x_t^u)$ 
11: end for
12: normalize  $b_t^e$ 
13: return  $b_t^e$ 

```

Algorithm 6 COMPUTEVALUEFUNCTION($i, b_i^i, b_{0:T}^{N^i}, \hat{\omega}_{1:T}^{N^i}, u_{0:T}^{N^i}, t, T$)

```

1: if  $V_t^i(b_t^i)$  already exists then
2:   return  $V_t^i(b_t^i)$ 
3: end if
4:  $best \leftarrow -\infty$ 
5: for all  $u_{t+1}^i \in \mathcal{U}^i$  do
6:    $val \leftarrow \text{COMPUTEVALUEACTION}(i, b_t^i, u_{t+1}^i, b_{0:T}^{N^i}, u_{0:T}^{N^i}, t, T)$ 
7:    $V_{t+1}^{u_{t+1}^i}(b_t^i) \leftarrow val$ 
8:   if  $val > best$  then
9:      $best \leftarrow val$ 
10:  end if
11: end for
12:  $V_t^i(b_t^i) \leftarrow best$ 
13: return  $best$ 

```

space is continuous, two challenges occur.

The first challenge is to find a suitable representation of the beliefs, i.e., the probability distributions over the state space. In the discrete case one just has to store a probability for each possible state. In the continuous case

Algorithm 7 COMPUTEVALUEACTION($i, b_t^i, u_t^i, b_{0:T}^{Ni}, \hat{\omega}_{1:T}^{Ni}, u_{0:T}^{Ni}, t, T$)

```

1:  $val \leftarrow 0$ 
2: for all  $x_t^i, x_t^u$  do
3:    $reward \leftarrow \tilde{R}(x_t^i, x_t^u, u_t^i)$ 
4:    $val \leftarrow^+ b_t^i(x_t^u, x_t^i) \cdot reward$ 
5: end for
6: if  $t < T$  then
7:   for all  $\omega_{t+1}^i \in \Omega^i$  do
8:      $b_{t+1}^i \leftarrow \text{UPDATE}(i, b_t^i, u_t^i, \omega_{t+1}^i)$  // belief update
9:     // probability of making observation  $\omega_{t+1}^i$  according to (3.12):
10:     $prob \leftarrow \mathbb{P}\{\omega_{t+1}^i = \omega_{t+1}^i | b_t^i, u_{t+1}^i\}$ 
11:     $prob\_others \leftarrow \mathbb{P}\{\omega_{t+1}^{Ni} = \hat{\omega}_{t+1}^{Ni} | b_t^{Ni}, u_{t+1}^{Ni}\}$ 
12:     $val \leftarrow^+ prob \cdot prob\_others \cdot$ 
                                COMPUTEVALUEFUNCTION( $i, b_{t+1}^i, b_{0:T}^{Ni}, \hat{\omega}_{1:T}^{Ni}, u_{0:T}^{Ni}, \dots$ )
13:   end for
14: end if
15: return  $val$ 

```

Algorithm 8 COMPUTEPOLICY($i, b_t^i, \omega_{1:t}^i$), t, T

```

1:  $u_{t+1}^i \leftarrow \arg \max_{u_{t+1}^i} \left\{ V_i^{u_{t+1}^i}(b_t^i) \right\}$ 
2:  $\pi^i(\omega_{1:t}^i) \leftarrow u_{t+1}^i$ 
3: if  $t < T$  then
4:   for all  $\omega_{t+1}^i$  do
5:      $b_{t+1}^i \leftarrow \text{UPDATE}(i, b_t^i, u_{t+1}^i, \omega_{t+1}^i)$  // belief update
6:     COMPUTEPOLICY( $i, b_{t+1}^i, [\omega_{1:t}^i, \omega_{t+1}^i], t+1, T$ )
7:   end for
8: end if
9: return

```

we propose a projection of the belief space \mathcal{B} , i.e., the infinite-dimensional space of continuous probability distributions, to a finite low-dimensional representation (e.g. Gaussians characterized by their mean and covariance) as described in [113]. Over the belief states, which in the case of

3. Distributed Optimization under Uncertainty

Algorithm 9 EVALUATEPOLICY($i, x_t^i, x_t^u, x_t^{N^i}, \pi^i, \omega_{0:t}^i, \hat{\omega}_{0:T}^{N^i}, u_{0:T}^{N^i}, t, T$)

```

1:  $u_t^i \leftarrow \pi^i(\omega_{0:t}^i)$ 
2:  $value \leftarrow \sum_{d,i \in d} R_I(x_t^i, x_t^{d \setminus \{i\}}, x_t^u, u_t^i, u_t^{d \setminus \{i\}})$ 
3: if  $t < T$  then
4:   for all  $x_{t+1}^i \in \mathcal{X}^i, x_{t+1}^u \in \mathcal{X}^u, x_{t+1}^{N^i} \in \mathcal{X}^{N^i}$  do
5:     for all  $\omega_{t+1}^i$  do
6:        $\omega_{0:t+1}^i \leftarrow [\omega_{0:t}^i, \omega_{t+1}^i]$ 
7:        $value \leftarrow^+ p^u(x_{t+1}^u, x_t^u) \cdot p^i(x_{t+1}^i, x_{t+1}^i, u_t^i) \cdot p^{N^i}(x_{t+1}^{N^i}, x_{t+1}^{N^i}, u_{t+1}^{N^i}) \cdot$ 
          $o^i(\omega_{t+1}^i, x_{t+1}^i, x_{t+1}^u, u_t^i) \cdot o^{N^i}(\hat{\omega}_{t+1}^{N^i}, x_{t+1}^{N^i}, x_{t+1}^u, u_t^{N^i}) \cdot$ 
         EVALUATEPOLICY
         ( $i, x_{t+1}^i, x_{t+1}^u, x_{t+1}^{N^i}, \pi^i, \omega_{0:t+1}^i, \hat{\omega}_{1:T}^{N^i}, u_{0:T}^{N^i}, t+1, T$ )
8:     end for
9:   end for
10: end if
11: return  $value$ 

```

a finite dimensional representation are given by parameters of the probability distributions, the system is fully observable and called belief state MDP [80, 114]. Therefore, value iteration for finite horizon problems can be used to recursively compute the optimal value function [106]. *In theory*, the value function $J_T^*(b)$ is initialized with the expected final reward for every $b \in \mathcal{B}$. The Bellman equation

$$J_{t-1}^*(b_{t-1}) = \max_u \mathbb{E} \{ J_t^*(b_t^u) \} \quad (3.30)$$

is then applied to expand the value function recursively, where the belief state b_t^u is computed from the preceding belief according to the prediction step in Bayesian estimation [118]. At each step J_t^* is the maximal expected reward that can be gained in $T - t$ steps. The result of the final step J_0^* allows to compute the optimal action for arbitrary belief state in constant time [28].

However, since the belief state MDP has infinitely many states, it is not

possible to evaluate the value function at every point $b \in \mathcal{B}$. To overcome this problem, Fitted Value Iteration as proposed in [43, 92] can be employed to efficiently approximate the optimal value function. The idea is to evaluate the value function only over a finite subset $\tilde{\mathcal{B}} \subset \mathcal{B}$. Since in the value iteration process points b_t^u may not be included in $\tilde{\mathcal{B}}$, an interpolation method is used to estimate their values by means of neighboring points. This calculation repeats over the planning horizon and computations such as coefficients of interpolations can be performed off-line if the transition and observation model is time-invariant. Furthermore, if the model of the agents resembles each other, the same precalculations can be used for all agents. From the resulting approximation of the optimal value function, optimal actions from any initial belief can be derived in constant time.

A second challenge for applying the approach presented in this section to multi-agent systems with continuous state spaces is the determination of the MLO. For continuous observation spaces Ω^i the nonlinear and possibly non-convex optimization problem

$$\hat{\omega}_t^i = \arg \max_{\omega_t^i \in \Omega^i} \left\{ \mathbb{P} \left\{ \omega_t^i = \omega_t^i \middle| b_t^{u_t^i}, u_t^i \right\} \right\} \quad (3.31)$$

with continuous variable ω_t^i has to be solved. However, for two important and widely applied filtering techniques, i.e., for the Kalman filter for linear systems and the particle filter for general nonlinear systems, we will in the following show how to obtain the MLO exactly (in the linear case) or approximately (in the general nonlinear case).

3.5.1. MLO in Kalman Filters

In linear systems with additive Gaussian white noise characterized by the linear state equation

$$x_{t+1}^i = \mathbf{A}^i x_t^i + \mathbf{B}^i u_t^i + \mathbf{v}_t^i \quad (3.32)$$

with Gaussian system disturbance $\mathbf{v}_t^i \sim \mathcal{N}(0, \mathbf{Q})$ and measurement equation

$$\omega_t^i = \mathbf{H}^i \mathbf{x}_t^i + \mathbf{w}_t^i \quad (3.33)$$

with Gaussian measurement noise $\mathbf{w}_t^i \sim \mathcal{N}(0, \Sigma)$, the belief prediction and update are given by the Kalman filter equations [121]. These consist of state prediction conducted on the mean $\hat{\mathbf{x}}_t^i$ and the covariance \mathbf{C}_t^i of the Gaussian belief

$$\hat{\mathbf{x}}_t^{p^i} = \mathbf{A} \hat{\mathbf{x}}_{t-1}^{e^i} + \mathbf{B} u_t^i \quad (3.34)$$

$$\mathbf{C}_t^{p^i} = \mathbf{A} \mathbf{C}_{t-1}^{e^i} \mathbf{A}^\top + \mathbf{Q} \quad (3.35)$$

$$(3.36)$$

and an update when a measurement ω_t^i was taken

$$\hat{\mathbf{x}}_t^{e^i} = \hat{\mathbf{x}}_t^{p^i} + \mathbf{K}_t \left(\omega_t^i - \mathbf{H} \hat{\mathbf{x}}_t^{p^i} \right) \quad (3.37)$$

$$\mathbf{K}_t = \mathbf{C}_t^{p^i} \mathbf{H}^\top \left(\mathbf{H} \mathbf{C}_t^{p^i} \mathbf{H}^\top + \Sigma \right)^{-1} \quad (3.38)$$

$$\mathbf{C}_t^{e^i} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{C}_t^{p^i} . \quad (3.39)$$

A special characteristic in this setting is that all occurring beliefs are Gaussian and, hence, unambiguously parametrized by their mean vector and covariance matrix. It then follows directly that ML observations are given by $\hat{\omega}_t^i = \mathbf{H}^i \hat{\mathbf{x}}_t^{p^i}$, where \mathbf{H}^i is the observation matrix and $\hat{\mathbf{x}}_t^{p^i}$ the mean of the predicted state estimate [23].

3.5.2. MLO in Particle Filters

For state estimators in general nonlinear continuous systems, such as particle filters, the ML observation could be computed by sampling the observation distribution and subsequent maximization over the samples. Particle

filters are applied in many fields of engineering ranging from e.g. robotic localization [125], general estimation and tracking problems [21, 60], and sensor management problems [77, 122]. Here we will briefly sketch how maximum likelihood observations could be obtained in nonlinear models when particle filters are applied. We here consider the case where the system models are given by the general nonlinear time variant equation

$$x_{t+1}^i = f_t(x_t^i, u_t^i, \mathbf{v}_t^i) \quad (3.40)$$

and the measurement model is given by

$$\omega_t^i = h_t(x_t^i, u_t^i) + \mathbf{w}_t^i \quad (3.41)$$

with additive white Gaussian measurement noise $\mathbf{w}_t^i \sim \mathcal{N}(0, \Sigma)$ with covariance Σ . For this measurement model the likelihood is given by

$$p(\omega_t^i | x_t^i, u_t^i) = \mathcal{N}(\omega_t^i - h_t(x_t^i, u_t^i), \Sigma) \quad (3.42)$$

as can be easily checked. Assume that the predicted belief b^{u^i} is approximated by the empirical distribution

$$b^{u^i}(x_t^i, x_t^u) = \sum_{j=1}^N w_{j,t} \delta_{\{x_{j,t}^i\}} \quad (3.43)$$

for samples $x_{j,t}^i$ and importance weights $w_{j,t}$. For the distribution of the measurement process under the particle approximation it holds that

$$p(\omega_t^i) \approx \sum_{j=1}^N w_{j,t} p(\omega_t^i | x_t = x_{j,t}^i) \quad (3.44)$$

which is a mixture of Gaussians (GM) with N components with means located at

$$\{h(x_{j,t}^i, u_t^i)\}_{j=1}^N \quad (3.45)$$

and covariances Σ [37, 77]. From this GM we can now draw N samples $\omega_{j,t}^i \sim p(\omega_t^i)$ for $j = 1, \dots, N$ and the determination of the MLO can be approximated by determining

$$\hat{\omega}_t^i := \arg \max_{j=1, \dots, N} \{p(\omega_{j,t}^i)\} , \quad (3.46)$$

the maximum evaluated over the samples.

3.6. Discussion and Related Work

The algorithms presented in this chapter are based on some simplifying assumptions (transition and observation independence) and use some heuristics (ML observations and actions from the neighbors). In this section we therefore want to discuss in which scenarios one can assume that these simplified models will be feasible and when they potentially are too simple to model the real world.

If transition- and observation-independence is assumed, the stochastic models of the agents are completely independent and their interaction is solely modeled by a joint reward function. Several interesting problems can be sufficiently described under these assumptions.

An important example in which transition and observation independence can safely be assumed (and actually is in the literature, see for example [23, 83, 96, 99]) is tracking of multiple targets with one or several stationary or mobile sensors. In these applications it is assumed that the movement of the targets is independent and the measurements in the sensors are too. When applicable, coordination of the sensors in order to track the targets is then achieved by a joint reward function.

In multi-robot path planning scenarios it is generally beneficial to have a continuous model of the state space of the robots since a discretization will either be very simple and, hence, imprecise or it will be detailed resulting in a large state space. In path planning in continuous space, we are not interested in modeling the physical impact of actions between the agents, but their impact on the objective. For example, we do not want to model the physical implications of a collision of two robots in the transition model, but to avoid collisions while planning through the right choice of the objective function. In other words, if we avoid collisions between the robots by introducing a carefully modeled objective function, transition independence can be assumed. In Section 3.7.2 we will consider a problem as described above with continuous state space for the robots. If, on the other hand, the area in which the robots move is modeled by grid-based techniques and, hence, there is a discrete state space of locations for the agents, it can be too optimistic to assume transition independence. For example if a robot occupies a certain grid cell no other robot will be able to move into that grid cell and this fact will have to be modeled by the transition function. In such a case the assumption of transition independence will be violated.

The search for solutions of Dec-POMDPs is highly complex and requires approximations in all but the simplest of problems. We avoid the full probabilistic dynamics for the sake of efficiency, but nevertheless incorporate the observation model into planning, which is a strong advantage in comparison to pure open loop approaches (also called unobservable Markov decision processes [97]). The MLO assumption is not suitable, when the future reward highly depends on the observations as for example in the tiger scenario (see [98]). In this scenario there are only two observations (hear tiger or not) with equal probability and the successful action depends on this binary observation. With our approach, one instance of a joint trajectory in belief space is computed. This can be seen as a plan under best conditions. If used as online algorithm, i.e., with replanning after each ex-

executed action, unforeseen changes to this over-confident assumptions can be incorporated after each executed step, when the planning is performed anew with the updated information. We see our approach suitable in applications, where it is reasonable to pursue a 'most likely trajectory' in belief space and adapt the plan if required. This is for example the case for sensor management and robot navigation as we will show in our empirical results in the next section.

The idea of approximating POMDPs by assuming nominal belief state dynamics was first proposed in [96] for the problem of coordinated guidance of autonomous UAVs for tracking a target. To achieve fixed trajectories in belief space for a given sequence of actions, stochastic disturbance in the system and observation model is set to zero. When several UAVs coordinate to track targets, the paper does not consider a distributed coordination algorithm but a centralized one. In [105], Maximum Likelihood (ML) observations are introduced that lead to deterministic dynamics in belief space given a sequence of actions. Furthermore, theoretical aspects of the impact on Linear Quadratic Regulation are addressed. However, this paper considers only *deterministic system dynamics* and the derivation of its theoretical results heavily relies on this assumption. Both [96, 105] would not be applicable in the sensor network scenario in Example 3.1 since they assume deterministic dynamics for the system state and either do not consider coordination among several planning agents at all or utilize centralized coordination techniques. Hence, the important difference to our approach is that [96, 105] only solve single-agent POMDPs with deterministic system dynamics and not much harder distributed problems where the goal is to find joint policies for multiple agents.

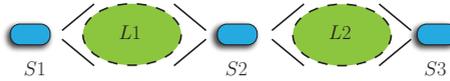


Figure 3.2.: Depicted is the '3-chain' sensor network in which two targets are to be tracked by three sensors.

3.7. Results

3.7.1. Sensor Network Target Tracking

Our goal in this section was to compare MLOLIDJESP using the MLO assumption with the original algorithm LIDJESP from [99] regarding runtime and cumulative reward of the resulting policies. We considered the sensor network example from Example 3.1 with increasing numbers of sensors and increasing complexity of the network. First, we considered the exact scenario from Example 3.1 (called '3-chain'), then considered a chain of four sensors (called '4-chain'), then a more complex scenario consisting of five sensors (called '5-p'), and finally considered a scenario with six sensors (called '6-h'). In order to ensure a high degree of comparability, the scenarios '3-chain', '4-chain' and '5-p' were exactly as in the experiments section of [99] from which we have also taken the details of the LIDJESP implementation. Planning with the MLO assumption was conducted according to the MLOLIDJESP algorithm. Since we conducted the simulations on a single office PC (Intel Pentium E5300 2.6 GHz, 8 GB RAM, Windows Server 2008 R2) the computations for the agents were conducted sequentially. We did not need the termination counter (*termCtr*) stopping criterion from line 6 in MLOLIDJESP since we immediately noticed when the agents' policies have converged. In the following we will present the details of each scenario and results obtained.

Scenarios

3. Distributed Optimization under Uncertainty

| | | | | | |
|-----------------|--------|------|-----------------|--------|------|
| Target 1 | absent | $L1$ | Target 2 | absent | $L2$ |
| absent | 0.5 | 0.5 | absent | 0.6 | 0.4 |
| $L1$ | 0.2 | 0.8 | $L2$ | 0.25 | 0.75 |

Table 3.1.: Transition probabilities of the targets for the '3-chain' scenario.

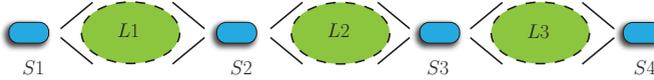


Figure 3.3.: Depicted is the '4-chain' sensor network in which two targets are to be tracked by four sensors.

| | | | |
|-----------------|--------|------|------|
| Target 2 | absent | $L2$ | $L3$ |
| absent | 0.4 | 0.35 | 0.25 |
| $L2$ | 0.2 | 0.5 | 0.3 |
| $L3$ | 0.3 | 0.25 | 0.45 |

Table 3.2.: Transition probabilities of Target 2 for the '4-chain' scenario.

'3-chain' In this scenario there were three sensors arranged in a chain and two targets (see Figure 3.2). The first target could either be absent or appear in location $L1$, the second target could be either absent or appear in location $L2$. It follows that there were four different possible joint state x^u . The transition probabilities for the targets are given in Table 3.1

'4-chain' In this scenario there were four sensors also arranged in a chain (see Figure 3.3). There were two targets, the first one could be either absent or appear in location $L1$ and has the same transition function as the first target in the previous scenario, i.e., its transition function is specified in Table 3.1. The second target could either be absent, appear in location $L2$ or appear in location $L3$. Target 2's transition probabilities are given in Table 3.2. In total there are six different possible joint states x^u .

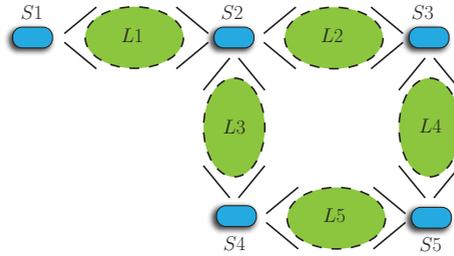


Figure 3.4.: Depicted is the '5-p' sensor network in which two targets were to be tracked by five sensors.

| Target 1 | absent | $L1$ | $L3$ | $L5$ |
|-----------------|--------|------|------|------|
| absent | 0.15 | 0.5 | 0.2 | 0.15 |
| $L1$ | 0.1 | 0.5 | 0.3 | 0.1 |
| $L3$ | 0.2 | 0.1 | 0.45 | 0.25 |
| $L5$ | 0.35 | 0.05 | 0.1 | 0.5 |

Table 3.3.: Transition probabilities of Target 1 for the '5-p' scenario.

| Target 2 | absent | $L2$ | $L4$ |
|-----------------|--------|------|------|
| absent | 0.4 | 0.35 | 0.25 |
| $L2$ | 0.2 | 0.5 | 0.3 |
| $L4$ | 0.3 | 0.25 | 0.45 |

Table 3.4.: Transition probabilities of Target 2 for the '5-p' scenario.

'5-p' In this scenario there were five sensors arranged in a 'p' shape depicted in Figure 3.4. Again, there were two targets that the sensors want to track. The first target could be either absent, in location $L1$ or in $L3$ or in location $L5$. The second target could be either absent, in location $L2$ or in location $L4$. The transition probabilities for both targets are given in Tables 3.3 and 3.4. In total there were 12 different possible joint states x^u .

'6-h' In this scenario six sensors were arranged in the 's' shape as depicted in Figure 3.5. Again there were two targets the sensors tried to track

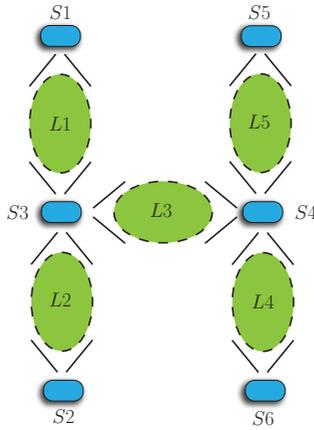
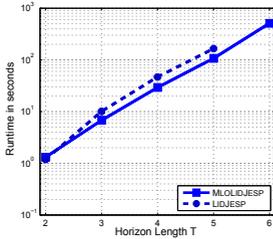


Figure 3.5.: Depicted is the '6-h' sensor network in which two targets were to be tracked by six sensors.

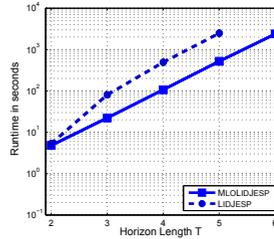
where the first target could either be absent or appear in locations $L1$, $L3$, and $L5$. Its transition function is the same as in the '5-p' scenario and therefore also given by Table 3.3. The second target could be either absent or appear in locations $L2$ or $L4$ and its transition function is given by Table 3.4. In total there were 12 different joint states for the positions of the targets.

Evaluation

In Figures 3.6 and 3.7 we depict the runtime of MLOLIDJESP and LIDJESP in the '3-chain' scenario (Figure 3.6(a)) and in the '4-chain' scenario (Figure 3.6(b)) for different lengths of planning horizon T and averaged over ten runs with different random initial policies each. In these Figures the vertical axis is always in log-scale. It can be seen that especially for shorter planning horizons like $T = 2$ or for very simple scenarios like for the '3-chain' scenario there is not a huge gain in runtime from planning with the most likely observation assumption. In MLOLIDJESP there is some overhead for determining the most likely observation and the nominal belief trajectories.

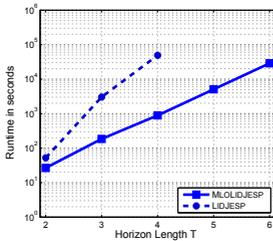


(a) Runtime comparison of LIDJESP and MLOLIDJESP for the '3-chain' sensor network scenario.

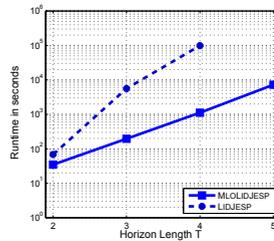


(b) Runtime comparison of LIDJESP and MLOLIDJESP for the '4-chain' sensor network scenario.

Figure 3.6.: Runtimes in seconds for the '3-chain' and for the '4-chain' sensor network scenarios. Due to numerical complexity and high memory requirements for LIDJESP only policies for a time horizon up to length $T = 5$ were evaluated.



(a) Runtime comparison of LIDJESP and MLOLIDJESP for the '5-p' sensor network scenario.

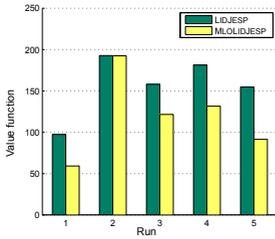


(b) Runtime comparison of LIDJESP and MLOLIDJESP for the '6-h' sensor network scenario.

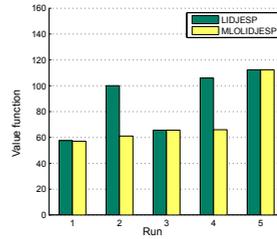
Figure 3.7.: Runtimes in seconds for the '5-p' and for the '6-h' sensor network scenarios. MLOLIDJESP's runtime is orders of magnitude faster than the time it takes LIDJESP to find a solution. Due to numerical complexity and memory requirements for LIDJESP only policies up to $T = 4$ were evaluated.

Since in these simple scenarios each sensor did not have many neighbors and each of these neighbors did not have many potential observation, the overhead in MLOLIDJESP balanced with the theoretical complexity reduc-

3. Distributed Optimization under Uncertainty



(a) Cumulative reward function values for the '4-chain' sensor network scenario.



(b) Cumulative reward function values for the '5-p' sensor network scenario.

Figure 3.8.: Value of cumulative reward obtained by joint policies for the '4-chain' and for the '5-p' sensor network scenarios for 5 different initial policies. It can be seen that the value achieved by MLOLIDJESP is less than that of LIDJESP. However, since the runtime needed with MLOLIDJESP to find a joint policy is so much lower, MLOLIDJESP can employ random restarts and find several locally optimal joint policies in the time LIDJESP would need to find only one solution. This is especially beneficial since in both algorithms the quality of the locally optimal solutions heavily depends on the initial starting policies.

tion.

In more complex scenarios like the '5-p' and '6-h' scenarios where the local neighborhoods were bigger and, hence, the local planning problem for a sensor had a higher complexity we can see in Figure 3.7 that the complexity reduction that comes with planning with the most likely observation assumption in MLOLIDJESP had an impact of orders of magnitude on the runtime compared to LIDJESP. This shows that the most likely observation assumption constitutes a significant improvement in terms of runtime in all but the most simple scenarios, especially in those with more neighbors and observation spaces with higher cardinality.

Since the most likely observation assumption is a suboptimal heuristic not only the runtime performance of MLOLIDJESP is of importance but also the quality of the solutions found when employing this heuristic.

| | LIDJESP | MLOLIDJESP |
|--------------------|---------|------------|
| Scenario '4-chain' | 156.822 | 119.269 |
| Scenario '5-p' | 95.3162 | 73.6413 |

Table 3.5.: Average cumulative reward obtained by LIDJESP and MLOLIDJESP.

Therefore, we were also interested in evaluating the overall cumulative reward obtained from joint policies in the sensor network scenario and in comparing this with the reward gained from joint policies obtained with LIDJESP. In Figure 3.8(a) we depict the cumulative reward of policies obtained with LIDJESP and with MLOLIDJESP for the '4-chain' sensor network scenario for $T = 3$ and in Figure 3.8(b) we depict the cumulative reward gained in the '5-p' scenario. Each bar corresponds to a single run and for each run both algorithms were initialized with the same random initial policy in order to ensure comparability. The cumulative rewards achieved averaged over ten runs are depicted in Table 3.5

It shows that on average the quality of solutions found with MLOLIDJESP were slightly more suboptimal than for the ones found with LIDJESP. In [99] it was also shown that the quality of solutions found with LIDJESP were likely to be suboptimal compared to the globally optimal solution, hence, MLOLIDJESP's solutions were even more suboptimal. However, both algorithms only find local optima whose value strongly depends on the starting value, the initial joint policy. Since MLOLIDJESP showed a runtime performance that is of orders of magnitude better than LIDJESP's especially in complex scenarios, MLOLIDJESP can be used in combination with random restarts in order to improve the value of the solution found. The much faster runtime allows MLOLIDJESP to explore the solutions space more thoroughly in much faster time than LIDJESP. For example, in the '5-p' scenario in Figure 3.8(b) MLOLIDJESP could evaluate five different random initial policies and choose the joint policy that gives the best cumulative reward in half the time LIDJESP would need to find only one locally optimal joint policy.

3.7.2. Multi-Robot Coordination

In this section, we consider a path planning problem in a multi-robot system where the task of the robots was to reach a target area in an environment with obstacles with motion affected by disturbances and noisy sensor measurements for localization. In this scenario we applied the MLO coordination algorithm in continuous robot state space in a receding horizon fashion, i.e., the robots coordinated and agreed on a set of policies that maximize a cumulative reward over a time horizon of length T and each robot i executed the first action $u_0^i = \pi^i(\langle \rangle)$. Then the robots took measurements in order to improve their localization and replanned.

System Model

The state of the robot was modeled as a three-dimensional continuous vector $[x_t, y_t, \psi_t]$, where x_t and y_t denoted the two-dimensional position of the robot in the plane and ψ_t the orientation of the robot. The discrete-time state space motion model of the robots we employed both in the simulation and in the experiments was

$$x_{t+1} = \begin{bmatrix} x_t \\ y_t \\ \psi_t \end{bmatrix} + \begin{bmatrix} (u_t^s + \mathbf{v}_t^s) \cos(\psi_t) \\ (u_t^s + \mathbf{v}_t^s) \sin(\psi_t) \\ u_t^\psi + \mathbf{v}_t^\psi \end{bmatrix}, \quad (3.47)$$

where the input to the system u_t^s denoted the speed and the input u_t^ψ the applied turning angle. The disturbances \mathbf{v}_t^s and \mathbf{v}_t^ψ were modeled as zero-mean and white with Gaussian distribution with variances $\sigma_s^2 = 1\text{cm}^2$ and $\sigma_\psi^2 = 0.001\text{rad}^2$. The disturbances were added to the control inputs to model disturbances on the actuators of the robots or model inaccuracies.

In order to improve their position estimates each robot took noisy distance measurements to two landmarks after each movement. State estimation consisting of prediction and filtering with these measurements was conducted with linearized system and observation models by means of the

extended Kalman filter (EKF) [121]. We did not discretize the state space of the robots and applied the extension to continuous state spaces as described in Section 3.5. Since we employed the EKF for state estimation, all occurring belief states were Gaussians and could be parametrized by their first and second moments. Each local POMDP over the robot's state space was solved by fitted value iteration as proposed in [43]. The MLO was obtained as $\mathbf{H}\hat{x}_t^p$, where \mathbf{H} was the first order linearization of the measurement model by its partial derivative matrix and \hat{x}_t^p the mean of the predicted state estimate [23, 121].

The possible control inputs to the robots were inputs of speed $u_t^s = 1$ or $u_t^s = 0$. Furthermore, the turning angles $u_t^\psi \in \{-\frac{\pi}{4}\text{rad}, 0\text{rad}, \frac{\pi}{4}\text{rad}\}$ could be applied resulting in six possible control inputs. We restricted the possible turning angles to going straight and making a turn to the left or to the right since it was shown that for a Dubins vehicle (which has essentially the same dynamics as in (3.47) except for the omission of noise and with a fixed speed) paths of minimal length from a starting configuration to a goal configuration only consist of straights and curves with maximal turning angle [62, 123].

The environment in which the robots acted and planned was static with obstacles and a target region at known positions. The reward function was split into a local reward R_L^i that depended only on the state of robot i and a reward R^{ij} evaluating pairwise locations of robots. The local reward R_L^i

$$R_L^i(x^i, y^i, \psi^i) = \begin{cases} 0, & \text{if } [x^i, y^i]^T \text{ is in the target region} \\ -3, & \text{if } [x^i, y^i]^T \text{ is not in the target region} \\ -20, & \text{if } [x^i, y^i]^T \text{ is in an obstacle.} \end{cases} \quad (3.48)$$

penalized all system states outside the target area as well as collisions with obstacles. In order to penalize potential collisions between two robots in the reward function R^{ij} we used the pairwise Mahalanobis distance D_M of

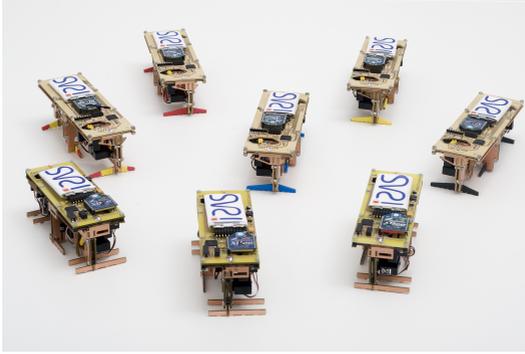


Figure 3.9.: Walking robots used for experiments.

their respective Gaussian belief states b^i and b^j

$$D_M(b^i, b^j) = \sqrt{(\hat{x}^i - \hat{x}^j)^T (\Sigma^i + \Sigma^j)^{-1} (\hat{x}^i - \hat{x}^j)} \quad (3.49)$$

where \hat{x}^i and \hat{x}^j were the means of the Gaussian belief states and Σ^i and Σ^j the covariances. The joint probabilistic reward function that modeled the interaction of two robots i and j was

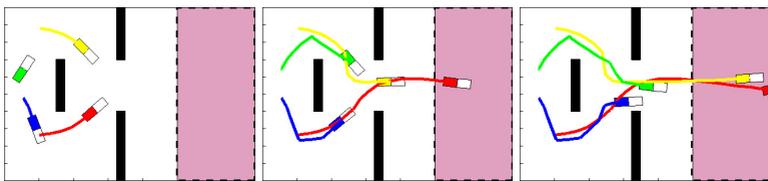
$$R_{ij}(b^i, b^j) = -20 \cdot \max\left(0, 1 - \frac{D_M(b^i, b^j)}{10}\right). \quad (3.50)$$

Mahalanobis distances larger than 10 were not considered and hence those robots did not interact with each other, as we assumed a collision as unlikely. The reward of each robot i was

$$R_L^i(x^i, y^i, \psi^i) + \sum_{j \neq i} R_{ij}(b^i, b^j). \quad (3.51)$$

Experiments with Real Robots

We conducted several experiments with robots introduced in [129] depicted in Figure 3.9. The size of the robots was $5\text{cm} \times 10\text{cm}$ and they were deployed in an $0.7\text{m} \times 1\text{m}$ test environment. The test environment was overlooked by a high resolution camera, whose images were used to create the GUI plots of a run in a environment with obstacles depicted in Figure 3.10. In Figures. 3.10(a) and 3.10(b) conflicts on which robot will pass the gap first occurred and both conflicts were resolved without a collision by the algorithm. In Figure 3.10(a) both the upper yellow as well as the lower red robot wanted to pass a gap between two obstacles and they coordinated in such a way that they did not collide. In these experiments localization was carried out with noisy measurements and *only* this data and the output of the EKF was used in our algorithm to determine coordinated joint motion plans for the robots. The precise readings from the overhead camera were not used for coordination in any of the experiments.



(a) First conflict before gap (b) Second conflict before gap (c) All conflicts were resolved.
 The two robots closest to the gap were in a position to pass the gap.
 The two backmost robots were in a position to pass the gap.

Figure 3.10.: Images taken from the GUI output of a run conducted on real robots. Here the robots task was to reach the target region on the right hand side of the environment. The obstacles are depicted in solid black. The complete trajectories of the robots up to the current depicted position are drawn as solid line in the color of the robot. The images show two conflict configurations (a) and (b) and a configuration after all conflicts have been resolved (d).

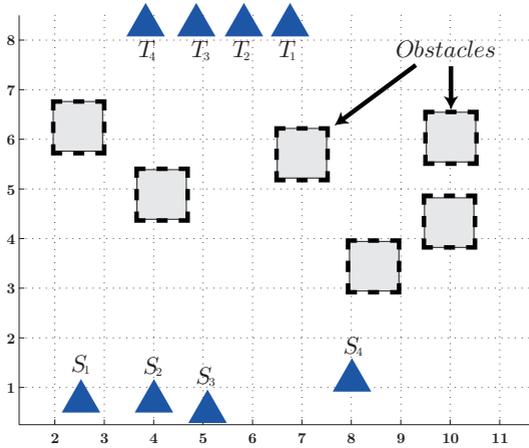


Figure 3.11.: Example environment with obstacles. The starting positions for four robots are denoted by S_1, \dots, S_4 and the target positions by T_1, \dots, T_4 . The dashed boxes were randomly placed obstacles of fixed size. The solid triangles on the bottom of the environment are the randomly placed starting positions and the top triangles are the goals for the robots. It can be seen that in this example there can be a lot of interaction among the robots, since their paths will most likely intersect at some point.

Quantitative Results

We conducted several quantitative tests in multi-robot path planning simulations with the models introduced above to evaluate the scalability and optimality of our algorithm for increasing numbers of robots in the system. The task of the robots was to reach a certain goal in randomly generated test environments with obstacles. For the simulations the obstacles were randomly placed and the starting and target positions for the robots were randomly chosen in such a way that their paths will intersect and, hence, collisions were likely to occur. A typical realization of the positions of the obstacles together with the starting and goal positions are depicted in Figure 3.11.

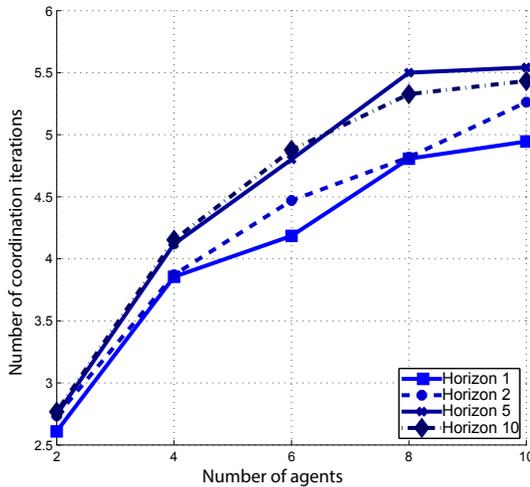


Figure 3.12.: Depicted are the average number of coordination iterations needed until a converged joint plan was found as a function of the number of robots in the system and for different lengths of the planning horizon. The number of coordination iterations necessary only moderately grows with increasing numbers of robots.

Scalability First it was our goal to evaluate how many best response coordination iterations were necessary until a converged locally optimal joint plan was found. Proposition 3.2 theoretically guarantees convergence but does not make any statements on *when* a locally optimal plan was found. However, the number of coordination iterations is of crucial importance for real-world applications, since it determines how frequently the robots have to communicate and how often each robot has to compute a best response according to Algorithm 2. Here MLOLIDJESP was used for deciding which robot can change its plan in its local neighborhood.

In Figure 3.12 the average number of coordination iterations needed until a converged joint plan was found are plotted as a function of the number of robots in the system and for different lengths of the planning horizon. For example, for six robots planning with a horizon of length five, in av-

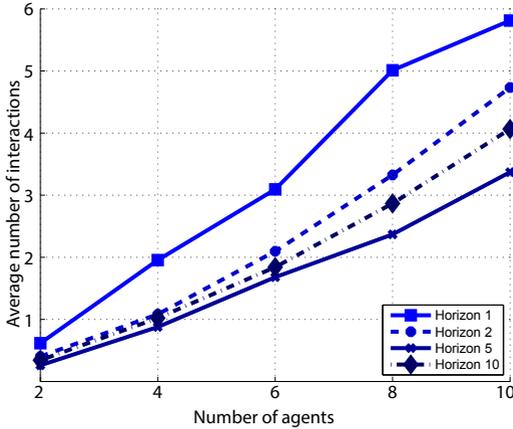


Figure 3.13.: Average number of interactions of each robot per coordination iteration and timestep in the planning horizon.

erage each robot had to compute its best response to the plans of the other robots according to Algorithm 2 a little less than five times until a locally optimal joint plan was found. It can be seen that the number of coordination iterations necessary was almost independent of the length of the planning horizon and only moderately increased with increasing numbers of robots. This is a particularly pleasant property since usually the complexity of multi-robot coordination problems grows exponentially with the number of robots.

In Figure 3.13 we depict the average number of other robots a robot interacted with, i.e. the average number of times the joint reward function $R_{i_0j} \neq 0$ for robot i_0 . This is an important statistic to look at, since even if there were hundreds of robots in an environment, if their paths were such that there was only pairwise interaction among a few robots, then the average number of coordination iterations would be low. It can be seen from the figure that in the randomly generated scenarios this was not the case as the

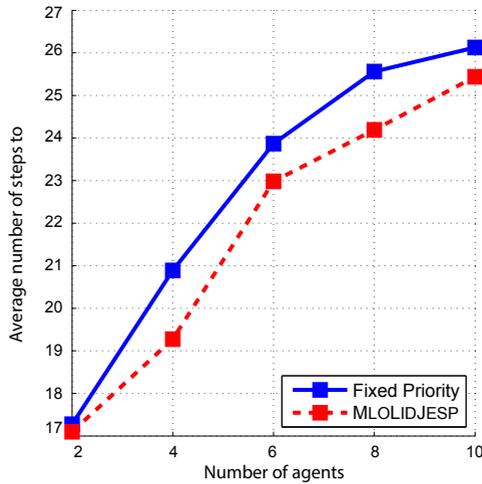


Figure 3.14.: Average number of steps the robots need to reach their goal as a function of the number of robots for a fixed priority polling scheme from Algorithm 1 and for a dynamic auctioning-like polling scheme from Algorithm 4. Averaged over 20 Monte-Carlo (MC) runs.

average number of interactions increased with increasing number of robots in the system.

Optimality Next, we wanted to evaluate the effect of the polling scheme, with which the robots were allowed to compute their best response to the plans of the other robots, on the optimality of the resulting locally optimal joint plan. We compared fixed order polling schemes from Algorithm 1 with the dynamic auctioning scheme from Algorithm 4.

In Figure 3.14 we plot the average number of steps each robot needed to reach its goal as a function of the number of robots in the system for a fixed priority polling scheme and for the dynamic scheme. Here a “step” has to be understood as the execution of one action, i.e., when a robot needed 26 steps to reach its goal, it applied 26 actions. Note that an action can also be to not move for one time step. Hence, the higher number of steps needed

3. Distributed Optimization under Uncertainty

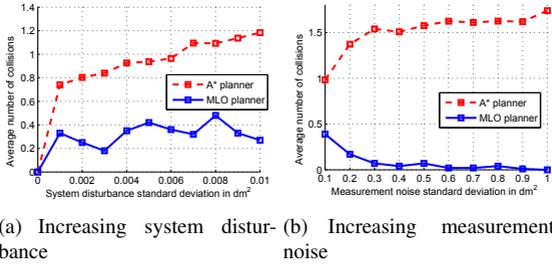


Figure 3.15.: Average number of robot collisions with A^* planning and the MLO planner as a function of increasing system disturbance (a) and measurement noise in localization (b). The results were averaged over 100 MC runs with random obstacles. Since the MLO planner accounts for uncertain system dynamics, noisy measurements and uncertain localization while planning, it was very robust and allowed the robots to safely reach their goals.

by the fixed priority scheme means that either the robots had to take larger detours or were forced more often to wait, e.g. in order to avoid a collision. It can be seen that on average a dynamic polling scheme lead to shorter and, hence, better paths than a fixed priority scheme, since the suboptimality of locally optimal joint plans found with dynamic polling schemes was lower than the suboptimality of plans found with fixed priority schemes.

Comparison to Deterministic Planning

Finally, we were interested how our approach of planning with nominal belief dynamics would compare to the A^* algorithm that is widely used in path planning in multi-robot systems (see e.g. [26, 35]). We implemented the multi-robot A^* planning algorithm as proposed in [26] in conjunction with an extended Kalman filter by letting the A^* planner plan on the means of the filter's estimates. The simulations in this subsection were conducted with a system consisting of four robots.

In Figure 3.15(a) we depict the average number of robot collisions as a function of increasing disturbance in the system dynamics (3.47). The

values on the x -axis refer to the standard deviation σ_s^2 . The measurement noise standard deviation was $\sigma_\omega^2 = 1\text{cm}^2$. When the system disturbance increases the movements of the robots become more imprecise and it becomes more difficult to precisely control them. Especially when robots get close to each other this causes problems since it becomes more difficult to avoid collisions. It can be seen that the average number of collisions was higher for the A^* planner and this number even increases for higher system disturbances. The number of collisions of our planner with nominal belief dynamics (MLO planner) was generally much lower and unaffected by the increase of system disturbance.

In Figure 3.15(b) we depict the average number of robot collisions as a function of increasing measurement noise of the distance sensors that were used for localization. The system disturbance was set to $\sigma_s^2 = 0.5\text{cm}^2$. When the standard deviation of the measurement noise increased, the sensors were more imprecise and hence the localization of the robots became more imprecise. For increasing measurement noise the number of robot collisions, that occurred when the robots executed plans computed with the A^* algorithm, were much higher than when planning with the MLO planner. Notice that as the measurement noise increased, the MLO planner accounted for the increase and became more cautious, leading to even less collisions. Collision-free paths determined with the A^* method had an average length of 79cm and MLO paths starting at 88cm at low measurement noise to 90cm for the highest measurement noise.

Our planner using nominal belief dynamics explicitly accounted for the uncertain system dynamics and the measurement model through the incorporation of ML observations when planning. Further, noisy system dynamics and uncertain localization corresponded to Gaussian belief states with high variance. Since our algorithm used these belief states to determine the reward of different state-action combinations, it also explicitly incorporated these uncertainties while planning. So the MLO planner was very robust and allowed the robots to safely reach their goals with just a slight increase

in overall path length.

3.8. Conclusions

In this chapter we proposed an heuristic for the efficient and distributed search for locally optimal joint policies in a multi-agent system in Dec-POMDPs. The most likely observation assumption in transition and observation independent Dec-POMDPs allows an agent, when alternating optimization is employed, to find its best response not to full observation dependent policies of its neighbors but to fixed sequences of actions. The advantage is that the local POMDP an agent has to solve does not have an extended state space that is comprised of the space of all observation sequences of the neighbors in addition to the state space of the agent but it is only a POMDP over the state space of the agent. Furthermore, we extended the most likely observation approach to POMDPs with continuous state and observation spaces and for the most common cases of linear agent dynamics together with Kalman filters, and nonlinear agent dynamics with particle filters we demonstrated ways to compute the most likely observation. In real and numerical experiments in a sensor network and a multi-robot motion planning scenario we evaluated the efficiency and robustness of planning with the most likely observation assumption.

Strengths Planning with the most likely observation assumption is an efficient heuristic to quickly find locally optimal joint policies in complex multi-agent systems with a lot of interaction among the agents. Under the most likely observation assumption each agent finds its best response to fixed sequences of actions of the other agents and the result is a full observation dependent joint policy for all agents. Although this is a suboptimal heuristic the orders of magnitudes faster runtime compared to other known suboptimal algorithms together with random restarts can be used to quickly and efficiently explore the joint policy space.

Limitations In Section 3.6 we already discussed that there are systems in which transition and observation independence might be too simple to model the behavior of a system. In these cases it is advisable to use the full dynamics and employ other solution strategies e.g. dynamic programming based techniques for Dec-POMDPs [98, 131].

We also already mentioned that in scenarios in which there are several different observations with equal likelihood of being made (e.g. in the distributed Tiger problem [98]) the most likely observation assumption can be too optimistic and even misleading. In these scenarios also the full Dec-POMDP model will have to be employed.

Applications In this thesis we studied two different applications for planning in Dec-POMDPs with the most likely observation assumption: a sensor network example and motion planning in a multi-robot system. General sensor management is a broad field of research and many approaches use the POMDP model to find optimal sensor management strategies [76]. Sensor management applications range from information theoretic sensor selection for maximizing the expected information gain from the sensors' measurements to the optimal placement of mobile sensing devices such as UAVs or sensor equipped robots [96]. Our approach will be applicable and we expect it to perform well in these scenarios.

4. Conclusions

In this thesis several approaches for optimization and decision-making under uncertainty with a strong focus on applications in multi-agent systems were considered.

A special characteristic of optimization in multi-agent systems is that coordination regarding the utilization of indivisible resources can be modeled through discrete decision variables. This is the reason why optimization problems with discrete and continuous decision variables occurred at several different occasions in this thesis.

One fundamental question when considering optimization under uncertainty is how the uncertainty actually influences the optimization problem. We first examined optimization problems in which the uncertainty affects the constraints through some random disturbance parameter. Our goal here was to develop algorithms that are able to efficiently deal with the arising uncertainties, i.e., algorithms in which the incorporation of uncertainties does not lead to a major increase in complexity. Furthermore, we were interested in giving strong theoretical guarantees on the solutions found with random constraints regarding the behavior of the solutions under realizations of the constraints that we not yet know or have not explicitly accounted for. In addition, we considered models in which the uncertainty affects the impact of the decision variables on the objective function. For these models we derived distributed algorithms that are characterized through their efficiency though at the cost that they do not find a globally optimal solution.

More precisely, we first studied model predictive control for motion planning in a multi-agent system with chance constraints on the probability of a

collision of agents. We highlighted that these chance constraints cannot be evaluated in closed form and lead to complex and non-convex constraints for the MPC problem. In order to make the MPC problem numerically tractable, we proposed several approximations and conservative reformulations of the chance constraints. In each case we proved that optimal controls found with these approximations converge against either optimal or feasible controls for the MPC problem with the true chance constraints. The approximations and reformulations proposed in this chapter are to the best of our knowledge the first that deal with the efficient application of chance constraints that couple the states of agents in a multi-agent system.

The *main challenge* tackled in this chapter was reducing the computational complexity of the constraints on collision probabilities while still ensuring theoretical properties such as feasibility and convergence.

Summarizing, the main advances and contributions of this chapter from a users perspective are the following:

- The techniques derived in this chapter facilitate efficient, safe, and reliable control of multi-agent systems even if the states of the agents are uncertain.
- To the best of our knowledge we are the first to consider coupling chance constraints on joint states of agents in a multi-agent system.
- The major applications are the safe and efficient operation and coordination of mobile robots, UAVs, vehicles, or aircraft.

We also studied mixed-integer convex programs with a fixed number of random constraints. We proved explicit bounds on the tails of the probability that the optimal solution of such a program will remain optimal under further, yet unseen, random constraints. Through a slightly different argumentation we proved, for the case that the random constraints are linear, different bounds that are sharper when the uncertainty only affects the right

hand side of the linear constraints. The explicit results on the generalization properties derived in this thesis are to the best of our knowledge the first for random convex programs with discrete and continuous decision variables.

The *main challenge* in this chapter was the theoretical derivation of bounds on the tails of the violation probability for optimal solutions of mixed-integer programs with random constraints.

Summarizing, the main advances and contributions of this chapter from a users perspective are the following:

- The theory derived in this chapter allows to immunize solutions of mixed-integer convex optimization problems with random constraints against uncertainty.
- The explicit results on the generalization properties derived in this thesis are to the best of our knowledge the first for random convex programs with discrete and continuous decision variables.
- The applications are whenever one has to deal with randomness and uncertainty in optimization and wants to immunize the solutions found against this uncertainty, e.g. in the design of truss structures or MPC in a multi-agent system with disturbance.

Finally, we studied distributed algorithms for efficiently finding suboptimal sequential solutions in the Dec-POMDP model. We proposed the most likely observation assumption for alternating optimization in multi-agent systems. We proved that the size of the local problem each agent has to solve is drastically reduced and we provided empirical evidence that this theoretical reduction also has an huge impact on runtime that can be up to orders of magnitude compared to similar suboptimal algorithms.

The *main challenge* in this chapter was to derive a computationally efficient coordination scheme that finds locally optimal joint policies even for high numbers of agents and long planning horizons.

Summarizing, the main advances and contributions of this chapter from a users perspective are the following:

- The algorithm derived in this chapter allows agents to agree on a locally optimal joint strategy in a distributed manner. This saves computation time and communication bandwidth, since all computations can be executed distributedly and communication is only used to exchange changes in the joint policy.
- The coordination strategy proposed in this chapter is an efficient heuristic to quickly find locally optimal joint solutions in complex multi-agent systems when there is a lot of interaction among the agents.
- The main applications are decentralized probabilistic planning in large and spatially distributed sensor networks with stationary or mobile sensors.

Future Work An interesting direction for future research is the exploration of distributed solution techniques for chance constrained MPC and MI-RCP problems. The optimization problems that result from the approximations and reformulations we presented for the chance constrained MPC problem are mixed-integer linear programs (MILPs). For solving these MILPs one could apply the approach [78] that proposes a combination of the Dantzig-Wolfe decomposition for linear programs together with Gomory cuts, a technique for finding optimal solutions to MILPs [34]. A further technique for solving MILPs are branch and cut algorithms in which branching is conducted over a subset of the integer variables. The application of algorithms from the field of parallel computing for message-based distributed solving of branch and cut problems [64, 82] to the multi-agent MILP could be worth exploring.

The works [51,52] already proposed a method with which random convex programs with only continuous decision variables can be solved in a distributed fashion in a multi-agent system. They assume that each agent only has knowledge about a small subset of the random constraints and through a certain communication protocol agents agree on a subset of relevant constraints such that then every agent can compute the globally optimal solution. When applying these algorithms to mixed-integer random convex programs the challenge is that the communication protocol is designed in such a way that the agents repeatedly exchange constraints that are active for their local optimization problems and the determination of active constraints in mixed-integer problems is highly nontrivial.

A third interesting direction for future work is the study of bounds on the tails of the violation probability for the SAA of chance constraints. In this thesis we were able to prove statements on the convergence of plans that are optimal under the SAA constraints, however, no statements about the rate of convergence or quantitative results on the error for fixed sample numbers were made. There is research on the topics for example [20,75] but the works largely depend on the knowledge or computation of the Vapnik-Chervonenkis (VC) dimension of the feasible set or use very conservative bounds on the dimension. The bounds derived in these works therefore can be too conservative as was shown in [44]. A first step towards sharpening the bounds with the help of statistical estimation of the VC dimension was made in [4] but needs to be extended further.

A. Appendix

A.1. General Convergence of the SAA

In this appendix we will very briefly state the results and assumptions from [103] on the convergence of optimal solutions of the sample average approximation (SAA) of a chance constrained program against the optimal solutions of the “true” problem.

Consider a general chance constrained problem

$$\min_{x \in X} f(x) \tag{A.1}$$

$$\text{s.t. } \mathbb{P}\{G(x, \mathbf{v}) \leq 0\} \geq 1 - \alpha \tag{A.2}$$

for compact set X , continuous function $f: \mathbb{R}^d \rightarrow \mathbb{R}$, random vector $\mathbf{v} \in \mathfrak{E}$, and a function $G: \mathbb{R}^d \times \mathfrak{E} \rightarrow \mathbb{R}$. The function G is assumed to be a Carathéodory function, i.e., $G(x, \cdot)$ is measurable for all $x \in \mathbb{R}^d$ and $G(\cdot, \mathbf{v})$ is continuous in x for almost every \mathbf{v} . As in [103] define

$$p(x) := \mathbb{P}\{G(x, \mathbf{v}) > 0\}, \tag{A.3}$$

then the general chance constrained problem is equivalent to

$$\min_{x \in X} f(x) \text{ s.t. } p(x) \leq \alpha \tag{A.4}$$

Let $\mathbf{v}^1, \dots, \mathbf{v}^N$ be N i.i.d. realizations of the random vector \mathbf{v} and denote by \mathbb{P}_N the empirical measure constructed from these samples. The SAA of the chance constrained problem is

$$\min_{x \in X} f(x) \text{ s.t. } p_N(x) \leq \alpha, \tag{A.5}$$

where

$$p_N(x) = \sum_{j=1}^N \mathbb{1}_{(0, \infty)}(G(x, \mathbf{v}^j)). \tag{A.6}$$

Another assumption has to be made in order to show convergence:

Assumption 5 (Assumption (A) in [103]) *There is an optimal solution x to the chance constrained problem such that for every $\varepsilon_0 > 0$ there exists a $\hat{x} \in X$ with $\|x - \hat{x}\| \leq \varepsilon_0$ and $p(\hat{x}) < \alpha$.*

This condition states that there is a sequence that converges to the optimal solution x and for each member of the sequence $p(x_k) < \alpha$. It guarantees that the optimal solution can be perturbed by a small margin and the probability of a failure will be smaller α . Then the following proposition holds

Proposition A.1 (Proposition 2.2 in [103]) *Let φ_N be the optimal value of the SAA and φ the optimal value of the general chance constrained problem. Further denote by S_N the set of all optimal solutions where φ_N is attained and by S the set where φ is attained. Then it holds that $\varphi_N \rightarrow \varphi$ with probability one for $N \rightarrow \infty$ and $d_H(S_N, S) \rightarrow 0$ with probability one for $N \rightarrow \infty$.*

List of Figures

| | | |
|------|--|-----|
| 1.1 | Path Planning for Air Traffic Control | 2 |
| 1.2 | Sample Approximation of Agent's State | 19 |
| 1.3 | Schematic Depiction of Regions of Increased Probability of Presence | 34 |
| 1.4 | Illustration of Non-overlapping Transformed RPPs | 56 |
| 1.5 | 2D Path Planning Comparison | 73 |
| 1.6 | Zoomed in view of 2D UAV trajectories for the SAA | 76 |
| 1.7 | 3D Path Planning with eRPP constraints | 77 |
| 1.8 | 3D Path Planning with transformed RPP constraints | 78 |
| 1.9 | Completion Rates for Quantitative Simulations | 82 |
| 1.10 | Runtimes for Quantitative Simulations | 83 |
| 1.11 | Objectives for Quantitative Simulations | 84 |
| 1.12 | Completion rate, runtime, and objective $N = 50$ | 86 |
| 1.13 | Completion rate, runtime, and objective $N = 100$ | 87 |
| | | |
| 2.1 | Illustration of the feasible region of a two-dimensional ILP | 104 |
| 2.2 | Three active constraints at the ball constraint | 124 |
| 2.3 | Feasible Region of Example 2.3 | 132 |
| 2.4 | Comparison of sample numbers for Knapsack problem | 141 |
| 2.5 | State Trajectories for Multi Agent Coordination | 149 |
| | | |
| 3.1 | Distributed Sensor Net Example | 160 |
| 3.2 | '3-chain' Sensor Network | 187 |
| 3.3 | '4-chain' Sensor Network | 188 |
| 3.4 | '5-p' Sensor Network | 189 |

| | | |
|------|---|-----|
| 3.5 | '6-h' Sensor Network | 190 |
| 3.6 | Runtimes for the '3-chain' and '4-chain' sensor networks . | 191 |
| 3.7 | Runtimes for the '5-p' and '6-h' sensor networks | 191 |
| 3.8 | Value of cumulative reward obtained by joint policies . . . | 192 |
| 3.9 | Walking robots used for experiments | 196 |
| 3.10 | GUI Output of Real Run | 197 |
| 3.11 | Example Environment with Obstacles | 198 |
| 3.12 | Average Number of Coordination Iterations | 199 |
| 3.13 | Average Number of Interactions | 200 |
| 3.14 | Average Number of Steps for a Robot | 201 |
| 3.15 | Average Number of Collisions | 202 |

List of Tables

| | | |
|-----|---|-----|
| 1.1 | Chance constraint usage of eRPP, transformed RPP, and SAA in 2D | 74 |
| 1.2 | Chance constraint usage and runtime eRPP and transformed RPP in 3D | 79 |
| 1.3 | Parameters used for quantitative simulations | 80 |
| 2.1 | Comparison of sample numbers for Knapsack problem for $d = 10, \dots, d = 15$ | 141 |
| 2.2 | Comparison of sample numbers for inventory problem for $T = 1, \dots, T = 9$ | 144 |
| 2.3 | Comparison of sample numbers for $T = 1, \dots, T = 5$ | 147 |
| 3.1 | Transition probabilities of the targets for the '3-chain' scenario | 188 |
| 3.2 | Transition probabilities of Target 2 for the '4-chain' scenario | 188 |
| 3.3 | Transition probabilities of Target 1 for the '5-p' scenario | 189 |
| 3.4 | Transition probabilities of Target 2 for the '5-p' scenario | 189 |
| 3.5 | Average cumulative reward obtained by LIDJESP and MLOLID-JESP | 193 |

Supervised Student Theses

- [1] B. Gross-Hardt. Bilineare Programmierung mit stochastischen Nebenbedingungen für modell-prädiktive Multi-Agenten-Planung (Bilinear Programming for Model Predictive Control with Stochastic Constraints). Diploma Thesis, Lehrstuhl für Intelligente Sensor-Aktor-Systeme, KIT Karlsruhe, 2011.
- [2] M. Kuderer. Dezentrale Regelung eines Multi-Agenten-Systems auf Basis nichtkooperativer Spiele (Decentralized Control of Multi-Agent Systems based on Non-cooperative Games). Diploma Thesis, Lehrstuhl für Intelligente Sensor-Aktor-Systeme, KIT Karlsruhe, 2010.
- [3] M. Nagel. Approximation von Wahrscheinlichkeitsdichten mithilfe einer Verallgemeinerung der Aitchison Geometrie (Approximation of Probability Density Functions with a Generalization of Aitchison Geometry). Student Research Project, Lehrstuhl für Intelligente Sensor-Aktor-Systeme, KIT Karlsruhe, 2010.
- [4] M. Nagel. Konfidenzabschätzungen für Erfolgswahrscheinlichkeiten stochastischer Regelung (Confidence Bounds for the Probability of Success of Stochastic Control). Diploma Thesis, Lehrstuhl für Intelligente Sensor-Aktor-Systeme und Institut für Stochastik, KIT Karlsruhe, 2012. Cited on page 211.
- [5] E. Prause. Chance-constrained Model Predictive Control For Multiple UAVs. Bachelor Thesis, Lehrstuhl für Intelligente Sensor-Aktor-Systeme, KIT Karlsruhe, 2011.

- [6] B. Schneyer. Betrügerresistente auktionenbasierte nicht-kooperative Sensoreinsatzplanung (Cheat-proof Auction-based Non-cooperative Sensor Management). Diploma Thesis, Lehrstuhl für Intelligente Sensor-Aktor-Systeme, KIT Karlsruhe, 2011.

Own Publications

- [7] G. C. Calafiore, D. Lyons, and L. Fagiano. On Mixed-Integer Random Convex Programs. In *Proceedings of the 51st IEEE Conference on Decision and Control (CDC 2012)*, Maui, Hawaii, USA, Dec. 2012. Cited on pages 94, 110, and 152.
- [8] J.-P. Calliess, D. Lyons, and U. D. Hanebeck. Lazy auctions for multi-robot collision avoidance and motion control under uncertainty. In *Autonomous Robots and Multirobot Systems (ARMS) Workshop at AA-MAS 2011*, Taipei, Taiwan, May 2011. Cited on page 6.
- [9] J.-P. Calliess, D. Lyons, and U. D. Hanebeck. Lazy Auctions for Multi-robot Collision Avoidance and Motion Control under Uncertainty. *Technical Report. No: PARG-11-01. University of Oxford*, 2011.
- [10] A. Hekler, D. Lyons, B. Noack, and U. D. Hanebeck. Nonlinear Model Predictive Control Considering Stochastic and Systematic Uncertainties with Sets of Densities. In *Proceedings of the IEEE Multi-Conference on Systems and Control (MSC 2010)*, Yokohama, Japan, Sept. 2010.
- [11] D. Lyons, J.-P. Calliess, and U. D. Hanebeck. Chance-constrained Model Predictive Control for Multi-Agent Systems. *arXiv preprint: Systems and Control (cs.SY)*, 2011. Cited on page 6.
- [12] D. Lyons, J.-P. Calliess, and U. D. Hanebeck. Chance Constrained Model Predictive Control for Multi-Agent Systems with Coupling

- Constraints. In *Proceedings of the 2012 American Control Conference (ACC 2012)*, Montréal, Canada, June 2012. Cited on page 6.
- [13] D. Lyons, A. Hekler, M. Kuderer, and U. D. Hanebeck. Robust Model Predictive Control Incorporating Least Favorable Measurements. In *Proceedings of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2010)*, Salt Lake City, Utah, USA, Sept. 2010.
- [14] D. Lyons, A. Hekler, B. Noack, and U. D. Hanebeck. Maße für Wahrscheinlichkeitsdichten in der informationstheoretischen Sensoreinsatzplanung. In F. Puente León, K.-D. Sommer, and M. Heizmann, editors, *Verteilte Messsysteme*, pages 121–132. KIT Scientific Publishing, Mar. 2010.
- [15] D. Lyons, B. Noack, and U. D. Hanebeck. A Log-Ratio Information Measure for Stochastic Sensor Management. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2010)*, Newport Beach, California, USA, June 2010.
- [16] B. Noack, V. Klumpp, D. Lyons, and U. D. Hanebeck. Modellierung von Unsicherheiten und Zustandsschätzung mit Mengen von Wahrscheinlichkeitsdichten. *tm - Technisches Messen, Oldenbourg Verlag*, 77(10):544–550, Oct. 2010.
- [17] B. Noack, V. Klumpp, D. Lyons, and U. D. Hanebeck. Systematische Beschreibung von Unsicherheiten in der Informationsfusion mit Mengen von Wahrscheinlichkeitsdichten. In F. Puente León, K.-D. Sommer, and M. Heizmann, editors, *Verteilte Messsysteme*, pages 167–178. KIT Scientific Publishing, Mar. 2010.
- [18] B. Noack, D. Lyons, M. Nagel, and U. D. Hanebeck. Nonlinear Information Filtering for Distributed Multisensor Data Fusion. In *Pro-*

ceedings of the 2011 American Control Conference (ACC 2011), San Francisco, California, USA, June 2011.

Bibliography

- [19] MIL-F-8785C, Military Specification. Technical report, 1980. Cited on pages 69 and 72.
- [20] T. Alamo, R. Tempo, and E. Camacho. Randomized Strategies for Probabilistic Solutions of Uncertain Feasibility and Optimization Problems. *IEEE Transactions on Automatic Control*, 54(11):2545–2559, 2009. Cited on pages 7 and 211.
- [21] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. Cited on page 183.
- [22] G. Averkov and R. Weismantel. Transversal Numbers over Subsets of Linear Spaces. *Advances in Geometry*, 12(1):19–28, 2012. Cited on page 101.
- [23] Y. Bar-Shalom, X. Li, T. Kirubarajan, and J. Wiley. *Estimation with Applications to Tracking and Navigation*. Wiley Online Library, 2001. Cited on pages 182, 184, and 195.
- [24] A. Ben-Tal and A. Nemirovski. Robust Convex Optimization. *Mathematics of Operations Research*, pages 769–805, 1998. Cited on page 95.
- [25] A. Ben-Tal and A. Nemirovski. Robust Solutions of Uncertain Linear Programs. *Operations Research Letters*, 25(1):1–14, 1999. Cited on pages 137 and 138.

- [26] M. Bennewitz, W. Burgard, and S. Thrun. Finding and Optimizing Solvable Priority Schemes for Decoupled Path Planning Techniques for Teams of Mobile Robots. *Robotics and Autonomous Systems*, 41(2-3), 2002. Cited on page 202.
- [27] D. S. Bernstein, S. Zilberstein, and N. Immerman. The Complexity of Decentralized Control of Markov Decision Processes. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, Stanford, California, 2000. Cited on pages 154, 155, 158, and 160.
- [28] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts, USA, 2007. Cited on pages 134, 142, 159, and 180.
- [29] D. Bertsimas and C. Caramanis. Finite Adaptability in Multistage Linear Optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, 2010. Cited on page 137.
- [30] D. Bertsimas and M. Sim. Robust Discrete Optimization and Network Flows. *Mathematical Programming*, 98:49–71, 2003. Cited on page 95.
- [31] D. Bertsimas and M. Sim. The Price of Robustness. *Operations research*, pages 35–53, 2004. Cited on page 95.
- [32] D. Bertsimas and A. Thiele. A Robust Optimization Approach to Inventory Theory. *Operations Research*, 54(1):150, 2006. Cited on pages 95 and 143.
- [33] D. Bertsimas and A. Thiele. Robust and Data-driven Optimization: Modern Decision-making Under Uncertainty. *INFORMS Tutorials in Operations Research: Models, Methods, and Applications for Innovative Decision Making*, 2006. Cited on page 137.

- [34] D. Bertsimas and J. Tsitsiklis. Introduction to linear optimization. 1997. Cited on pages 118 and 210.
- [35] S. Bhattacharya, V. Kumar, and M. Likachev. Distributed Optimization with Pairwise Constraints and its Application to Multi-robot Path Planning. In *Robotics: Science and Systems*, Zaragoza, Spain, June 2010. Cited on page 202.
- [36] J. R. Birge and F. V. Louveaux. *Introduction to Stochastic Programming*. Springer series in operations research. Springer, New York, 1997. Cited on pages 3, 7, and 11.
- [37] C. Bishop. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006. Cited on pages 10, 49, 184, and 48.
- [38] L. Blackmore, A. Bektassov, M. Ono, and B. Williams. Robust, Optimal Predictive Control of Jump Markov Linear Systems Using Particles. *Hybrid Systems: Computation and Control*, pages 104–117, 2007. Cited on page 7.
- [39] L. Blackmore and M. Ono. Convex Chance Constrained Predictive Control Without Sampling. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2009. Cited on pages 7, 21, 22, and 20.
- [40] L. Blackmore, M. Ono, A. Bektassov, and B. Williams. A Probabilistic Particle Approach to Optimal, Robust Predictive Control. *IEEE Transactions on Robotics*, 26(5), 2010. Cited on pages 7, 8, and 72.
- [41] L. Blackmore, M. Ono, and B. Williams. Chance-Constrained Optimal Path Planning With Obstacles. *IEEE Transactions on Robotics*, (99):1–15, 2011. Cited on page 7.

- [42] P. Bonami and M. Lejeune. An Exact Solution Approach for Portfolio Optimization Problems Under Stochastic and Integer Constraints. *Operations research*, 57(3):650–670, 2009. Cited on pages 7 and 95.
- [43] A. Brooks, A. Makarenko, W. S., and H. Durrant-Whyte. Parametric POMDPs for Planning in Continuous State Spaces. *Robotics and Autonomous Systems*, 54(11), 2006. Cited on pages 181 and 195.
- [44] G. Calafiore. Random Convex Programs. *SIAM Journal on Optimization*, 20(6):3427–3464, 2010. Cited on pages 93, 101, 103, 106, 107, 108, 109, 110, 111, 113, 114, 115, 119, 121, 122, 123, 148, and 211.
- [45] G. Calafiore and M. Campi. Uncertain Convex Programs: Randomized Solutions and Confidence Levels. *Mathematical Programming*, 102(1):25–46, 2005. Cited on page 93.
- [46] G. Calafiore and M. Campi. The Scenario Approach to Robust Control Design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006. Cited on pages 93, 103, and 110.
- [47] G. Calafiore and L. El Ghaoui. *Optimization Models*. Cambridge Univ Pr, 2014. Cited on pages 49 and 48.
- [48] G. Calafiore and B. Monastero. Data-Driven Asset Allocation with Guaranteed Short-Fall Probability. In *Proceedings of the American Control Conference (ACC)*, 2012. Cited on page 152.
- [49] M. Campi and S. Garatti. The Exact Feasibility of Randomized Solutions of Uncertain Convex Programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008. Cited on pages 93, 103, 110, 113, 123, 124, and 125.
- [50] L. Carlone, M. Kaouk Ng, J. Du, B. Bona, and M. Indri. Simultaneous Localization and Mapping Using Rao-Blackwellized Particle

- Filters in Multi Robot Systems. *Journal of Intelligent and Robotic Systems*, 63:283–307, 2011. Cited on page 3.
- [51] L. Carlone, V. Srivastava, F. Bullo, and G. Calafiore. A Distributed Algorithm for Random Convex Programming. In *Proceedings of the 5th IEEE International Conference on Network Games, Control and Optimization (NetGCooP)*, pages 1–7, 2011. Cited on pages 120 and 211.
- [52] L. Carlone, V. Srivastava, F. Bullo, and G. Calafiore. Distributed Random Convex Programming via Constraints Consensus. *Arxiv preprint arXiv:1207.6226*, 2012. Cited on pages 120 and 211.
- [53] A. Charnes, W. Cooper, and G. Symonds. Cost Horizons and Certainty Equivalents: An Approach to Stochastic Programming of Heating Oil. *Management Science*, 4(3):235–263, 1958. Cited on page 7.
- [54] H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952. Cited on page 148.
- [55] E. Cinquemani, M. Agarwal, D. Chatterjee, and J. Lygeros. On Convex Problems in Chance-constrained Stochastic Model Predictive Control. *Arxiv preprint arXiv:0905.3447*, 2009. Cited on page 7.
- [56] N. Correll and A. Martinoli. Multirobot Inspection of Industrial Machinery. *IEEE Robotics Automation Magazine*, 16(1):103–112, 2009. Cited on page 3.
- [57] CPLEX. ILOG CPLEX User’s Manual. Cited on pages 72 and 89.
- [58] L. Danzer, B. Grünbaum, and V. Klee. Helly’s Theorem and its Relatives. In *Proceedings of the Symposium on Pure Math*, volume 7, pages 101–180, 1963. Cited on page 129.

- [59] J. Doignon. Convexity in Cristallographical lattices. *Journal of Geometry*, 3(1):71–85, 1973. Cited on page 97.
- [60] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001. Cited on page 183.
- [61] N. Du Toit and J. Burdick. Robot Motion Planning in Dynamic, Uncertain Environments. *IEEE Transactions on Robotics*, (99):1–15, 2012. Cited on page 7.
- [62] L. Dubins. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics*, 79(3), 1957. Cited on page 195.
- [63] J. Dupacova, A. Gaivoronski, Z. Kos, and T. Szantai. Stochastic Programming in Water Management: A Case Study and a Comparison of Solution Techniques. *European Journal of Operational Research*, 52(1):28–44, 1991. Cited on page 7.
- [64] J. Eckstein. Parallel Branch-and-Bound Algorithms for General Mixed Integer Programming on the CM-5. *SIAM Journal on Optimization*, 4:794–814, 1994. Cited on page 210.
- [65] L. El Ghaoui, F. Oustry, and H. Lebret. Robust Solutions to Uncertain Semidefinite Programs. *SIAM Journal on Optimization*, 9:33–52, 1998. Cited on page 95.
- [66] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1,2. Wiley, 1968. Cited on page 113.
- [67] C. Frese. *Planung kooperativer Fahrmanöver für kognitive Automobile*. PhD thesis, Karlsruhe Institute of Technology, Karlsruhe, 2011. Cited on page 3.

- [68] P. Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53. Springer, 2004. Cited on page 89.
- [69] K.-S. Goetzmann, S. Stiller, and C. Telha. Optimization over Integers with Robustness in Cost and Few Constraints. In *WAOA '11*, Lecture Notes in Computer Science. Springer, 2011. Cited on page 95.
- [70] G. Golub and C. Van Loan. *Matrix Computations*, volume 3. Johns Hopkins University Press, 1996. Cited on pages 49 and 48.
- [71] I. Gurvich, J. Luedtke, and T. Tezcan. Staffing Call Centers with Uncertain Demand Forecasts: A Chance-constrained Optimization Approach. *Management Science*, 56(7):1093–1115, 2010. Cited on page 7.
- [72] E. Hansen, D. Bernstein, and S. Zilberstein. Dynamic Programming for Partially Observable Stochastic Games. In *Proceedings of the National Conference on Artificial Intelligence*, pages 709–715, 2004. Cited on page 158.
- [73] A. Hayes, A. Martinoli, and R. Goodman. Distributed Odor Source Localization. *IEEE Sensors Journal*, 2(3):260–271, 2002. Cited on page 3.
- [74] R. Henrion, P. Li, A. Möller, M. Steinbach, M. Wendt, and G. Wozny. Stochastic Optimization for Operating Chemical Processes Under Uncertainty. *Online optimization of large scale systems*, pages 457–478, 2001. Cited on page 7.
- [75] R. Henrion and W. Römisch. Metric Regularity and Quantitative Stability in Stochastic Programs with Probabilistic Constraints. *Mathematical Programming*, 84(1):55–88, 1999. Cited on pages 24, 211, and 23.

- [76] A. Hero. *Foundations and Applications of Sensor Management*. Springer, 2006. Cited on page 205.
- [77] G. Hoffmann and C. Tomlin. Mobile Sensor Network Control Using Mutual Information Methods and Particle Filters. *IEEE Transactions on Automatic Control*, 55(1):32–47, 2010. Cited on pages 3, 183, and 184.
- [78] S. A. Hong and G. Gordon. Optimal Distributed Market-Based Planning for Multi-Agent Systems with Shared Resources. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011. Cited on pages 90 and 210.
- [79] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, 1996. Cited on page 17.
- [80] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*, 101, 1998. Cited on pages 158, 167, and 180.
- [81] G. Kalai. Linear Programming, the Simplex Algorithm and Simple Polytopes. *Mathematical Programming*, 79(1):217–233, 1997. Cited on page 133.
- [82] R. M. Karp and Y. Zhang. Parallel Algorithms for Backtrack Search and Branch-and-Bound. *Journal of the ACM*, 40(3):765–789, 1993. Cited on page 210.
- [83] T. Kirubarajan and Y. Bar-Shalom. Probabilistic Data Association Techniques for Target Tracking in Clutter. *Proceedings of the IEEE*, 92(3):536–557, 2004. Cited on page 184.
- [84] D. Knuth. *The Art of Computer Programming Volume 2 Seminumerical Algorithms*, 1981. Cited on page 89.

- [85] K. Kondak, M. Bernard, F. Caballero, I. Maza, and A. Ollero. Cooperative Autonomous Helicopters for Load Transportation and Environment Perception. *Advances in Robotics Research*, pages 299–310, 2009. Cited on page 3.
- [86] Y. Kuwata and J. How. Robust Cooperative Decentralized Trajectory Optimization Using Receding Horizon MILP. In *Proceedings of the American Control Conference (ACC)*, pages 522–527, 2007. Cited on page 8.
- [87] Y. Kuwata and J. How. Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP. *IEEE Transactions on Control Systems Technology*, (99):1–9, 2010. Cited on pages 8 and 90.
- [88] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How. Distributed Robust Receding Horizon Control for Multivehicle Guidance. *IEEE Transactions on Control Systems Technology*, 15(4):627–641, 2007. Cited on page 8.
- [89] V. Lesser, C. Ortiz, and M. Tambe. *Distributed Sensor Networks: A Multiagent Perspective*, volume 9. Springer, 2003. Cited on page 160.
- [90] P. Li, M. Wendt, and G. Wozny. A Probabilistically Constrained Model Predictive Controller. *Automatica*, 38(7):1171–1176, 2002. Cited on page 7.
- [91] M. Lobo, M. Fazel, and S. Boyd. Portfolio Optimization with Linear and Fixed Transaction Costs. *Annals of Operations Research*, 152(1):341–365, 2007. Cited on page 152.
- [92] W. S. Lovejoy. Computationally Feasible Bounds for Partially Observed Markov Decision Processes. *Operations Research*, 39(1), 1991. Cited on page 181.

- [93] J. Luedtke and S. Ahmed. A Sample Approximation Approach for Optimization with Probabilistic Constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008. Cited on pages 7, 24, and 23.
- [94] J. Matoušek. On Geometric Optimization with Few Violated Constraints. *Discrete and Computational Geometry*, 14(1):365–384, 1995. Cited on page 108.
- [95] P. McMullen. The Maximum Numbers of Faces of a Convex Polytope. *Mathematika*, 17(2):179–184, 1970. Cited on page 133.
- [96] S. A. Miller, Z. A. Harris, and E. K. P. Chong. Coordinated Guidance of Autonomous UAVs via Nominal Belief-state Optimization. In *Proceedings of the American Control Conference (ACC)*, 2009. Cited on pages 184, 186, and 205.
- [97] M. Mundhenk, J. Goldsmith, C. Lusena, and E. Allender. Complexity of Finite-horizon Markov Decision Process Problems. *Journal of the ACM*, 47, July 2000. Cited on page 185.
- [98] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming Decentralized POMDPs: Towards Efficient Policy Computation for Multiagent Settings. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 2003. Cited on pages 155, 158, 164, 175, 185, and 205.
- [99] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Network Distributed POMDPs: A Synthesis of Distributed Constraint Optimization and POMDPs. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 2005. Cited on pages 155, 160, 161, 163, 164, 172, 173, 175, 176, 184, 187, and 193.
- [100] G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*, volume 18. Wiley New York, 1988. Cited on pages 80, 95, 125, 129, and 142.

- [101] M. Ono and B. C. Williams. Iterative Risk Allocation: A New Approach to Robust Model Predictive Control with a Joint Chance Constraint. In *Proceedings of 47th IEEE Conference on Decision and Control*, 2008. Cited on page 7.
- [102] M. Ono and B. C. Williams. Decentralized Chance-Constrained Finite-Horizon Optimal Control for Multi-Agent Systems. In *Proceedings of 49th IEEE Conference on Decision and Control*, 2010. Cited on pages 7 and 145.
- [103] B. Pagnoncelli, S. Ahmed, and A. Shapiro. Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications. *Journal of Optimization Theory and Applications*, 142(2):399–416, 2009. Cited on pages 7, 24, 26, 28, 29, 61, 62, 63, 66, 214, 215, 23, 25, and 27.
- [104] A. Papoulis, S. Pillai, and S. Unnikrishna. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill New York, 2002. Cited on pages 9 and 20.
- [105] R. Platt, R. Tedrake, L.-P. Tomas, and L. P. Kaelbling. Belief Space Planning Assuming Maximum Likelihood Observations. In *Robotics: Science and Systems (RSS)*, 2010. Cited on page 186.
- [106] W. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, volume 703. Wiley-Blackwell, 2007. Cited on pages 158 and 180.
- [107] A. Prékopa. *Stochastic Programming*. Springer, 1995. Cited on pages 3, 4, 7, and 11.
- [108] A. Prékopa. The Use of Discrete Moment Bounds in Probabilistic Constrained Stochastic Programming Models. *Annals of Operations Research*, 85:21–38, 1999. Cited on page 16.

- [109] S. Rachev and W. Römisch. Quantitative Stability in Stochastic Programming: The Method of Probability Metrics. *Mathematics of Operations Research*, pages 792–818, 2002. Cited on page 7.
- [110] A. Richards and J. How. Robust Stable Model Predictive Control with Constraint Tightening. In *Proceedings of the American Control Conference (ACC)*, 2006. Cited on page 8.
- [111] A. G. Richards. *Robust Constrained Model Predictive Control*. PhD thesis, Massachusetts Institute of Technology, Cambridge MA, February 2005. Cited on page 8.
- [112] W. Römisch and R. Schultz. Stability Analysis for Stochastic Programs. *Annals of Operations Research*, 30(1):241–266, 1991. Cited on page 7.
- [113] N. Roy, G. Gordon, and S. Thrun. Finding Approximate POMDP Solutions Through Belief Compression. *Journal of Artificial Intelligence Research*, 23(1):1–40, 2005. Cited on page 179.
- [114] S. Russell, P. Norvig, and A. Artificial Intelligence. Artificial Intelligence: A Modern Approach. *Artificial Intelligence*, 1995. Cited on pages 155, 169, and 180.
- [115] T. Schouwenaars, J. How, and E. Feron. Receding Horizon Path Planning with Implicit Safety Guarantees. In *Proceedings of the American Control Conference (ACC)*, volume 6, pages 5576–5581, 2004. Cited on page 1.
- [116] R. Schultz. Some Aspects of Stability in Stochastic Programming. *Annals of Operations Research*, 100(1):55–84, 2000. Cited on page 7.
- [117] A. Schwarm and M. Nikolaou. Chance-constrained Model Predictive Control. *AIChE Journal*, 45(8):1743–1752, 1999. Cited on pages 3, 7, and 11.

- [118] F. Schweppe. *Uncertain Dynamic Systems*, volume 160. Prentice-Hall Englewood Cliffs, New Jersey., 1973. Cited on page 180.
- [119] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*, volume 9. SIAM, 2009. Cited on pages 7 and 95.
- [120] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009. Cited on page 6.
- [121] D. Simon. *Optimal State Estimation: Kalman, H [infinity] and Non-linear Approaches*. John Wiley and Sons, 2006. Cited on pages 61, 182, and 195.
- [122] S. Singh, N. Kantas, B. Vo, A. Doucet, and R. Evans. Simulation-based Optimal Sensor Scheduling with Application to Observer Trajectory Planning. *Automatica*, 43(5):817–830, 2007. Cited on page 183.
- [123] P. Soueres and J. Boissonnat. Optimal Trajectories for Nonholonomic Mobile Robots. *Robot motion planning and control*, 1998. Cited on page 195.
- [124] A. Soyster. Convex Programming with Set-inclusive Constraints and Applications to Inexact Linear Programming. *Operations research*, pages 1154–1157, 1973. Cited on page 136.
- [125] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. 2001. Cited on page 183.
- [126] D. Van Hessem, C. Scherer, and O. Bosgra. LMI-based Closed-loop Economic Optimization of Stochastic Process Operation under State and Input Constraints. In *Proceedings of the 40th IEEE Conference on Decision and Control*, 2002. Cited on page 7.

- [127] P. Varakantham, J. Marecki, Y. Yabu, M. Tambe, and M. Yokoo. Letting Loose a SPIDER on a Network of POMDPs: Generating Quality Quaranteed Policies. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, page 218. ACM, 2007. Cited on page 160.
- [128] M. P. Vitus and C. J. Tomlin. Closed-Loop Belief Space Planning for Linear, Gaussian Systems. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2011. Cited on page 7.
- [129] F. Weissel, M. F. Huber, and U. D. Hanebeck. Test-Environment based on a Team of Miniature Walking Robots for Evaluation of Collaborative Control Methods. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, California, 2007. Cited on page 197.
- [130] P. Whittle. A Multivariate Generalization of Tchebichev’s Inequality. *The Quarterly Journal of Mathematics*, 9(1):232, 1958. Cited on pages 36 and 37.
- [131] F. Wu, S. Zilberstein, and X. Chen. Trial-based Dynamic programming for Multi-agent Planning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 908–914, 2010. Cited on page 205.
- [132] N. Yilmaz, C. Evangelinos, P. Lermusiaux, and N. Patrikalakis. Path Planning of Autonomous Underwater Vehicles for Adaptive Sampling Using Mixed Integer Linear Programming. *IEEE Journal of Oceanic Engineering*, 33(4):522–537, 2008. Cited on page 1.
- [133] G. Ziegler. *Lectures on Polytopes*, volume 152. Springer, 1995. Cited on pages 118 and 133.