Titre: Title:	Contributions méthodologiques en optimisation pour le calcul de mesures d'ensembles, réduction d'arbres décisionnels et programmation linéaire
Auteur: Author:	Youssouf Emine
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Emine, Y. (2025). Contributions méthodologiques en optimisation pour le calcul de mesures d'ensembles, réduction d'arbres décisionnels et programmation linéaire [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/66528/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/66528/
Directeurs de recherche: Advisors:	Issmaïl El Hallaoui, & Andrea Lodi
Programme: Program:	Doctorat en mathématiques de l'ingénieur

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Contributions méthodologiques en optimisation pour le calcul de mesures d'ensembles, réduction d'arbres décisionnels et programmation linéaire

Youssouf EMINE

Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*

Mathématiques

Mai 2025

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée

Contributions méthodologiques en optimisation pour le calcul de mesures d'ensembles, réduction d'arbres décisionnels et programmation linéaire

présentée par **Youssouf EMINE**

en vue de l'obtention du diplôme de *Philosophiæ Doctor*

a été dûment acceptée par le jury d'examen constitué de :

Michel GENDREAU, président
Issmail EL HALLAOUI, membre et directeur de recherche
Andrea LODI, membre et codirecteur de recherche
Youssef DIOUANE, membre
Bissan GHADDAR, membre externe

DÉDICACE

À François Soumis, mon directeur, dont la passion et la vision continuent d'éclairer mon chemin malgré son départ prématuré, à ma conjointe Asmaa, à mes parents, et à tous ceux dont l'amour, le soutien et l'amitié ont rendu possible ce parcours.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de cette thèse.

Tout d'abord, je souhaite rendre un hommage particulier à mon directeur de recherche, François Soumis, dont la passion et la vision scientifique m'ont guidé tout au long de ce parcours. Sa disparition pendant la thèse laisse un vide immense, et je lui dédie ce travail en signe de reconnaissance éternelle.

Je remercie également mes codirecteurs. À Issmail El Hallaoui, pour sa disponibilité et ses conseils, et surtout à Andrea Lodi, dont le soutien constant et l'engagement sans faille ont été déterminants dans l'avancement de mes recherches.

Je tiens à remercier chaleureusement Jean Bernard Lasserre, coauteur d'un de mes articles, dont l'impact sur ce travail a été particulièrement significatif. À Thibaut Vidal et Alexandre Forel, mes co-auteurs du dernier et amis, je témoigne toute ma reconnaissance pour leur aide précieuse, avec une mention spéciale pour Alexandre, dont le soutien a été exceptionnel.

Je souhaite également remercier Asmaa Jerroumi, mon épouse, pour son soutien inestimable et sa présence réconfortante tout au long de ces quatre années de mariage. Mes sincères remerciements vont aussi à mes parents, Mohamed, Bint Kheir et ma grand-mère Marieme, pour leur amour et leur soutien indéfectible, ainsi qu'aux parents de mon épouse, Abdelmajid et Habiba, pour leur appui continu et chaleureux.

Un remerciement tout particulier est adressé à Khalid Laaziri, responsable informatique, qui a tout mis en œuvre pour m'aider à bien mener les recherches dans un environnement idéal.

Enfin, je n'oublie pas tous les autres membres de ma famille ainsi que mes collègues du laboratoire, dont la présence et l'amitié ont enrichi ce parcours.

À toutes et à tous, je dédie ce travail avec toute ma reconnaissance et ma sincère gratitude.

RÉSUMÉ

Cette thèse explore divers aspects de l'optimisation et ses applications à la résolution de problèmes complexes en recherche opérationnelle et en apprentissage automatique. En combinant des méthodes basées sur les moments avec des techniques d'optimisation combinatoire issues de l'apprentissage automatique, nous mettons en lumière de nouveaux résultats théoriques et développons des algorithmes améliorés. Le travail est structuré autour de trois contributions principales, chacune abordant un défi important du domaine et proposant des solutions concrètes, rigoureusement fondées sur des principes mathématiques.

La première contribution étudie des formulations polynomiales alternatives du problème du plus grand sous-ensemble stable d'un graphe comme un problème d'optimisation polynomiale. Ce problème classique de la théorie des graphes, connu pour sa complexité NP-difficile, a des applications importantes en conception de réseaux, en ordonnancement et en allocation de ressources. Le défi consiste à identifier le plus grand sous-ensemble de sommets non adjacents dans un graphe, tâche qui devient de plus en plus difficile à mesure que la taille du graphe augmente. Étant donné que le problème peut être exprimé sous différentes formes polynomiales équivalentes, chaque formulation peut être traitée à l'aide de la hiérarchie de Lasserre des relaxations semi-définies. Notre étude examine systématiquement comment le choix de la formulation influence la qualité des bornes obtenues. Des expériences approfondies révèlent que certaines formulations produisent des bornes nettement plus serrées que d'autres. Nous apportons également un éclairage théorique sur la manière dont la structure d'une formulation polynomiale donnée affecte l'efficacité du processus de relaxation, offrant ainsi des indications précieuses pour choisir les meilleures formulations dans les problèmes combinatoires. **Cette étude a donné lieu à la publication de l'article « A Note on the Lasserre Hierarchy for Different Formulations of the Maximum Independent Set Problem » dans *Optimization Letters*, 2021.**

La deuxième contribution se concentre sur l'amélioration de l'interprétabilité des modèles d'ensemble d'arbres boostés, largement utilisés en apprentissage automatique sur des données structurées dans des domaines tels que la finance, la santé et l'évaluation des risques. Ces modèles, qui agrègent plusieurs apprenants faibles (généralement des arbres de décision), sont réputés pour leur grande précision et leur robustesse. Toutefois, leur complexité rend leur interprétation difficile, ce qui constitue un inconvénient majeur dans les contextes où la compréhension du processus décisionnel est essentielle. Pour remédier à cela, nous introduisons une technique de réduction de modèle (pruning) fondée sur l'optimisation,

visant à réduire la taille des modèles tout en maintenant leurs performances prédictives. En formulant cette tâche comme un problème d'optimisation exacte et en appliquant une stratégie itérative et adverse, nous parvenons à réduire considérablement la complexité des modèles sans perte de précision. Nos expériences démontrent que cette méthode permet de simplifier fortement le modèle, facilitant ainsi son interprétation sans compromettre sa performance. **Ce travail a été accepté pour publication à la *Conférence AAAI sur l'intelligence artificielle*, 2025, sous le titre « Free Lunch in the Forest : Functionally Identical Pruning of Tree Ensembles ».**

La dernière contribution présente une version améliorée de l'algorithme du simplexe primal, un outil fondamental en programmation linéaire pour résoudre efficacement des problèmes d'optimisation. Bien que cette version atténue certains problèmes de dégénérescence inhérents à l'algorithme classique, elle peut encore sous-performer dans les cas non dégénérés, ce qui engendre des inefficacités sur des problèmes de grande taille. Pour y remédier, nous proposons une formulation affinée du sous-problème auxiliaire qui garantit des améliorations substantielles de la fonction objectif à chaque itération, jusqu'à l'obtention d'une solution ϵ-optimale. Notre analyse théorique confirme que le nombre de directions de descente reste polynomial, ce qui rend notre approche bien plus évolutive que l'algorithme classique du simplexe primal amélioré. Cette stratégie plus robuste et efficace a le potentiel d'accroître la performance des solveurs de programmation linéaire, aussi bien dans la recherche que dans les applications industrielles. **Ce travail est actuellement en révision pour publication dans le *Journal of Optimization Theory and Applications*, 2025.**

ABSTRACT

This thesis explores various aspects of optimization and its applications in solving complex problems in operations research and machine learning. By blending moment-based methods with combinatorial optimization techniques from machine learning, we uncover fresh theoretical insights and develop improved algorithms. The work is organized around three key contributions, each addressing a significant challenge in the field and offering practical, mathematically grounded solutions.

The first contribution examines alternative formulations of the maximum independent set problem as a polynomial optimization problem. This well-known NP-hard problem in graph theory has important applications in network design, scheduling, and resource allocation. The central challenge is to identify the largest subset of non-adjacent vertices in a graph; a task that becomes increasingly difficult as the graph grows. Since the problem can be expressed in several equivalent polynomial forms, each formulation can be tackled using the Lasserre hierarchy of semidefinite relaxations. Our study systematically investigates how the choice of formulation influences the quality of the resulting bounds. Extensive experiments reveal that certain formulations yield much stronger bounds than others. We also provide theoretical insights into how the structure of a given polynomial formulation affects the efficiency of the relaxation process, offering valuable guidance for selecting better formulations for combinatorial problems. **This study resulted in the publication of the article "A Note on the Lasserre Hierarchy for Different Formulations of the Maximum Independent Set Problem" in *Optimization Letters*, 2021.**

The second contribution is focused on enhancing the interpretability of boosted tree ensemble models, which are widely used in machine learning for structured data in domains such as finance, healthcare, and risk assessment. These models, which aggregate multiple weak learners (typically decision trees), are celebrated for their high accuracy and robustness. However, their complexity can make them hard to interpret, which is a serious drawback in situations where understanding the decision-making process is crucial. To tackle this issue, we introduce an optimization-based pruning technique designed to reduce model size while preserving predictive performance. By formulating pruning as an exact optimization problem and employing an iterative adversarial strategy, we are able to significantly reduce the complexity of these models without sacrificing accuracy. Our experiments demonstrate that this method can greatly simplify the model, making it far easier for practitioners to interpret the results without losing any predictive power. **This work has been accepted**

for publication in the *AAAI Conference on Artificial Intelligence*, 2025, under the title "Free Lunch in the Forest: Functionally Identical Pruning of Tree Ensembles".

The final contribution presents an improved version of the primal simplex algorithm, a fundamental tool in linear programming used to solve optimization problems efficiently. Although the improved primal simplex method mitigates some degeneracy issues inherent in the classical algorithm, it can still underperform in non-degenerate cases, resulting in inefficiencies for large-scale problems. To address this, we propose a refined formulation of the auxiliary subproblem that guarantees substantial improvements in the objective function at every iteration until an ϵ-approximate optimal solution is reached. Our theoretical analysis confirms that the number of descent directions remains polynomial, making our approach significantly more scalable than the classical improved primal simplex method. This more robust and efficient strategy has the potential to boost the performance of linear programming solvers in both research and industrial applications. **This work is currently under review for publication in the *Journal of Optimization Theory and Applications*, 2025.**

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES LOGICIELS

GloptiPoly3	A software package for manipulating and solving generalized problems of moments [1].
MATLAB	A high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment. type.
SeDuMi	A software package for solving convex optimization problems over symmetric cones. type.

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

L'optimisation mathématique et l'apprentissage automatique jouent un rôle fondamental dans de nombreux domaines scientifiques et industriels, allant de la modélisation de systèmes complexes à la prise de décision automatisée. Cette thèse s'inscrit dans ce cadre en apportant des contributions méthodologiques autour des axes principaux suivants : l'approximation des mesures de Lebesgue et gaussiennes de l'union d'ensembles semi-algébriques via des relaxations semidéfinies[1], l'étude des hiérarchies de Lasserre appliquées au problème du plus grand sous-ensemble stable d'un graphe, la réduction fonctionnelle des ensembles d'arbres de décision, et enfin l'introduction d'une nouvelle formulation de la méthode du simplexe amélioré. Bien que ces problématiques soient distinctes, elles partagent une approche commune consistant à exploiter des techniques avancées d'optimisation afin d'améliorer l'efficacité et la précision des algorithmes existants.

Le calcul de la mesure d'un ensemble $\mathbf{\Omega} \subset \mathbb{R}^n$ relativement à une mesure borélienne μ demeure un problème difficile, même lorsque $\mathbf{\Omega}$ est un ensemble simple et convexe, tel qu'un polytope [2]. Pour ces ensembles, plusieurs algorithmes non déterministes ont été développés [3,4]. Parmi ces approches, la méthode Monte Carlo se distingue par sa simplicité : elle consiste à générer un grand nombre de points suivant une distribution donnée et à estimer $\mu(\mathbf{\Omega})$ par le rapport des points se trouvant dans $\mathbf{\Omega}$ [5]. Bien que cette technique soit efficace en faible dimension, elle ne fournit pas de bornes monotones convergentes vers $\mu(\mathbf{\Omega})$, ce qui limite son utilisation pour des problèmes nécessitant des garanties de borne supérieure et inférieure sur $\mu(\mathbf{\Omega})$. Dans le cas d'ensembles non convexes, notamment ceux définis par des inégalités polynomiales (ensembles semi-algébriques), des méthodes déterministes reposant sur des hiérarchies de programmes semidéfinis ont été proposées afin d'obtenir des suites monotones de bornes convergentes vers $\mu(\mathbf{\Omega})$ [6]. Cependant, cette approche souffre souvent d'une convergence lente, et les tentatives d'accélération aboutissent fréquemment à la perte de la monotonie des bornes, rendant ainsi leur utilisation moins fiable pour des estimations où la monotonie est cruciale. De plus, ces méthodes sont généralement limitées aux ensembles semi-algébriques individuels et ne peuvent être directement étendues à l'union de plusieurs ensembles de ce type. La première contribution de cette thèse s'articule autour de deux axes complémentaires. D'une part, nous proposons une méthode déterministe innovante permettant d'approximer, avec une précision prédéfinie, la mesure de l'union finie d'ensembles semi-algébriques. Cette approche est applicable aussi bien aux ensembles com-

1. Des programmes qui cherchent à minimiser une fonction linéaire sous des contraintes d'inégalités linéaires et de positivité semidéfinie de matrices.

pacts, en utilisant la mesure de Lebesgue, qu'aux ensembles non compacts, en utilisant par exemple la mesure gaussienne définie par $d\mu = \exp(-\|\mathbf{x}\|^2)\, d\mathbf{x}$ pour $\mathbf{x} \in \mathbb{R}^n$. L'approximation est obtenue en résolvant une hiérarchie de programmes semidéfinis, garantissant des bornes monotones convergentes vers $\mu(\mathbf{\Omega})$. D'autre part, nous introduisons une technique visant à accélérer la convergence de la hiérarchie de programmes semidéfinis tout en préservant la monotonie des bornes obtenues. Cette accélération repose sur l'intégration de contraintes de moments issues d'une application judicieuse du théorème de Stokes [7], garantissant ainsi des approximations avec des bornes monotones et une convergence plus rapide.

L'optimisation polynomiale et ses liens étroits avec l'optimisation semidéfinie et conique ont suscité un vif intérêt ces dernières années [8]. Il est bien établi que l'optimisation semidéfinie a eu un impact considérable sur l'optimisation combinatoire, en particulier grâce aux résultats révolutionnaires de Lovász et Schrijver [9] ainsi que de Goemans et Williamson [10]. Cela motive l'étude de l'application de l'optimisation polynomiale aux problèmes combinatoires. De plus, les problèmes combinatoires peuvent généralement être formulés de différentes manières. Il est en effet reconnu que diverses formulations polynomiales d'un même problème peuvent conduire à des relaxations semidéfinies différentes et, par conséquent, à des bornes globales distinctes. L'exemple du problème de *la coupe maximale dans un graphe* est exploré sous cet angle dans [11]. Enfin, plusieurs approches ont été proposées pour construire des hiérarchies de relaxations semidéfinies pour les problèmes combinatoires (binaires), notamment appliquées au polytope de l'ensemble stable. Lovász et Schrijver [9] utilisent une séquence d'opérations de lift-and-project pour construire leur hiérarchie, tandis que les travaux de Lasserre [12] partent d'une formulation polynomiale qu'ils raffinent progressivement à travers une hiérarchie. De Klerk et Pasechnik [13] font appel à la programmation copositive pour établir une autre hiérarchie, et une approche supplémentaire découle de la technique de Reformulation-Linéarisation proposée par Sherali et Adams [14]. Toutes ces hiérarchies ont en commun la propriété de converger vers la solution optimale en un nombre fini d'étapes, comme l'a comparé Laurent dans [15]. Dans la deuxième contribution de cette thèse, nous nous concentrons sur l'approche de Lasserre et sur l'influence des différentes formulations polynomiales d'un même problème. Plus précisément, nous considérons plusieurs formulations d'optimisation polynomiale pour le problème du *sous-ensemble stable maximal d'un graphe* et l'application de la hiérarchie de Lasserre à ces différentes formulations. Nos expérimentations computationnelles démontrent que le choix de la formulation peut avoir un impact significatif sur les bornes obtenues. Nous fournissons également des justifications théoriques pour expliquer ce comportement. L'objectif de notre travail est d'analyser ces différentes formulations en termes de performance numérique et

théorique. À travers des expériences computationnelles et une analyse théorique détaillée, nous démontrons que le choix de la formulation influence significativement la qualité des bornes obtenues. Certaines formulations conduisent à des relaxations plus serrées et permettent une convergence plus rapide vers la solution optimale. Ces résultats apportent un éclairage crucial sur l'utilisation pratique de la hiérarchie de Lasserre et permettent d'orienter le choix des formulations afin d'améliorer la qualité des approximations.

Dans un troisième temps, nous nous intéressons à l'utilisation des techniques d'optimisation combinatoire pour la réduction des méthodes d'ensembles, qui constituent l'un des modèles d'apprentissage automatique les plus efficaces et les plus répandus. Parmi celles-ci, les ensembles d'arbres, tels que les forêts aléatoires et les ensembles d'arbres *boostés*, sont reconnus pour atteindre des performances de pointe sur des données tabulaires [16,17] tout en offrant une meilleure interprétabilité que de grands réseaux de neurones. Par ailleurs, la théorie et la pratique montrent que des ensembles de grande taille, c'est-à-dire des forêts comportant de nombreux arbres, améliorent significativement les performances globales [18–20]. Toutefois, ces grands ensembles imposent d'importantes exigences en termes de mémoire et de temps d'inférence, ce qui peut constituer un véritable frein lorsqu'il s'agit de déployer ces modèles sur des dispositifs embarqués, tels que microcontrôleurs ou appareils mobiles. De plus, la complexité issue des interactions entre leurs composantes limite leur interprétabilité. La réduction d'ensemble désigne le processus de diminution de la taille d'un modèle d'ensemble entraîné, que ce soit en supprimant certains nœuds au sein des arbres ou en retirant complètement certains arbres de la forêt. Les premières approches se sont intéressées à la réduction des forêts aléatoires constituées d'arbres peu profonds, ces derniers ayant déjà fait l'objet d'une réduction préalable [21, 22]. Ces travaux ont démontré que la réduction pouvait améliorer l'erreur de test et la généralisation hors échantillon. Cependant, pour les ensembles *boostés*, dont les arbres sont généralement déjà peu profonds, cette stratégie s'avère inappropriée. Ainsi, les recherches récentes cherchent à établir un compromis entre l'ampleur de la réduction et son impact sur l'erreur de test [23,24], sans toutefois garantir de bonnes performances sur de nouvelles données. Dans le cadre de la troisième contribution de cette thèse, nous contournons entièrement ce compromis en procédant à une réduction des modèles d'ensemble sans modifier leur comportement prédictif. Nous qualifions cette approche de réduction *fonctionnellement identique*, car les fonctions de prédiction du modèle réduit et de l'original restent identiques. Cette stratégie présente deux avantages majeurs. D'une part, les problèmes d'optimisation qui en résultent, comme nous le démontrons ici, peuvent être résolus de manière très efficace. D'autre part, ces modèles réduits offrent un véritable "free lunch" : ils atteignent exactement les mêmes performances que le modèle original, tout en étant considérablement réduits en taille. Notre approche, appliquée à des

ensembles d'arbres entraînés sur des données tabulaires réelles, montre que la réduction fonctionnellement identique permet de réduire la taille des modèles d'ensemble de 30% à 60% tout en préservant les performances de prédiction.

Enfin, nous nous intéressons à la résolution des problèmes d'optimisation linéaire, qui sont au cœur de nombreux domaines d'application. En effet, la programmation linéaire est un outil puissant pour modéliser et résoudre des problèmes d'optimisation dans des domaines variés tels que la logistique, la finance, l'ingénierie et bien d'autres. Plusieurs problèmes d'optimisation linéaire proviennent de la relaxation de problèmes d'optimisation combinatoire, tels que le problème du *sous-ensemble stable maximal* ou le problème du *voyageur de commerce*. Ces problèmes sont souvent formulés sous forme de programmes linéaires, où l'objectif est de maximiser ou minimiser une fonction linéaire soumise à des contraintes linéaires et des contraintes d'intégralité. Pour résoudre ces problèmes, on utilise généralement un algorithme de *branchement* qui explore un arbre de recherche en divisant le problème initial en sous-problèmes plus simples. Dans chaque nœud de l'arbre, on relâche les contraintes d'intégralité et on résout le problème linéaire associé. La méthode du simplexe, introduite par Dantzig en 1947 [25], reste l'un des algorithmes les plus efficaces pour résoudre ces problèmes. Le simplexe parcourt itérativement les sommets du polytope défini par les contraintes du problème, se déplaçant de sommet en sommet jusqu'à atteindre la solution optimale. Cependant, dans les cas où le problème est dégénéré, c'est-à-dire lorsque plusieurs bases correspondent à un même point extrême, le simplexe peut rencontrer des difficultés pour progresser vers la solution optimale. Plusieurs approches ont été proposées pour remédier à ce problème, notamment la méthode du simplexe amélioré [26]. Cette méthode consiste à décomposer le problème en deux sous-problèmes plus simples : l'un, le problème réduit, non dégénéré, est résolu par le simplexe primal classique, tandis que l'autre, le problème complémentaire, faiblement dégénéré du point de vue dual, est résolu en utilisant un simplexe dual. Le problème complémentaire permet de trouver une direction d'amélioration pour la solution courante. Toutefois, bien que la méthode du simplexe amélioré ait montré des performances prometteuses, elle ne garantit pas une amélioration significative[2] de la solution à chaque itération et risque d'effectuer un nombre important d'itérations entre le problème réduit et le problème complémentaire. La dernière contribution de cette thèse consiste à présenter une nouvelle formulation de la méthode du simplexe amélioré, basée sur une nouvelle décomposition du problème en deux sous-problèmes. Cette nouvelle formulation permet de garantir une amélioration significative de la solution à chaque itération tout en préservant la convergence polynomiale globale de la méthode. Nous démontrons que cette formulation est plus efficace que la méthode du

2. Supérieure à un seuil fixé.

simplexe amélioré classique, en particulier dans les cas de dégénérescence sévère.

Bien que ces contributions puissent, à première vue, sembler indépendantes, elles sont en réalité profondément interconnectées par des concepts et des outils méthodologiques communs. L'un des fils conducteurs majeurs réside dans l'utilisation des hiérarchies de Lasserre, qui constituent une méthode puissante pour aborder le problème généralisé des moments. Ce dernier offre un cadre unificateur englobant de nombreux problèmes d'optimisation, tels que l'optimisation polynomiale, le calcul de la mesure d'ensembles semi-algébriques, ainsi que diverses relaxations de problèmes combinatoires. Les hiérarchies de Lasserre permettent ainsi de traiter de manière systématique des problèmes qui, bien que différents dans leur formulation, partagent une structure mathématique sous-jacente commune.

De plus, une analogie intéressante peut être établie entre les méthodes d'optimisation abordées dans cette thèse. Par exemple, la méthode proposée pour la réduction fonctionnelle des ensembles d'arbres de décision adopte une approche itérative qui rappelle, par son esprit, la démarche de la méthode du simplexe amélioré en programmation linéaire. Dans les deux cas, il s'agit de partir d'une solution initiale et de l'améliorer progressivement à travers une succession d'étapes en résolvant un problème auxiliaire pour obtenir une direction de descente, jusqu'à atteindre une solution optimale ou à certifier qu'elle l'est. Cette similitude de démarche souligne l'universalité de certains principes d'optimisation, indépendamment du domaine d'application spécifique.

Un autre point commun entre les contributions réside dans l'importance de trouver des bornes supérieures et inférieures pour les problèmes d'optimisation. Dans le cas de l'approximation des mesures de Lebesgue et gaussiennes, nous avons développé une méthode garantissant des bornes monotones convergentes, ce qui est essentiel pour évaluer la qualité des approximations. De même, dans le contexte de la hiérarchie de Lasserre appliquée au problème du sous-ensemble stable maximal, le choix de la formulation polynomiale influence directement la qualité des bornes supérieures obtenues. Enfin, la nouvelle formulation du simplexe amélioré vise à garantir une amélioration significative de la solution à chaque itération, ce qui est crucial pour assurer la qualité des bornes inférieures dans le cadre de la résolution de programmes linéaires.

En résumé, les différentes contributions de cette thèse, bien qu'abordant des problématiques variées, sont unies par une approche commune fondée sur l'exploitation de structures mathématiques partagées et sur l'utilisation de techniques d'optimisation avancées. Cette transversalité méthodologique permet non seulement d'apporter des solutions innovantes à des problèmes spécifiques, mais aussi de renforcer la compréhension des liens profonds qui existent entre des domaines a priori distincts de l'optimisation et de l'apprentissage

automatique.

Cette thèse est structurée en plusieurs chapitres. Après cette introduction qui a exposé le contexte et les contributions principales, le Chapitre 2 présente une revue de la littérature relative aux thématiques abordées. Par la suite, le Chapitre 3 présente la structure de la thèse et les contributions principales. Ensuite, toute contribution est incluse dans un chapitre dédié sous forme d'article publié ou soumis, avec

- Le Chapitre 4 examine l'impact du choix de la formulation polynomiale sur l'efficacité de la hiérarchie de Lasserre appliquée au problème du sous-ensemble stable maximal d'un graphe.
- Le Chapitre 5 décrit l'approche utilisée pour la réduction fonctionnelle des ensembles d'arbres de décision.
- Le Chapitre 6 présente la nouvelle formulation de la méthode du simplexe amélioré.

En outre, l'annexe A présente les méthodes développées pour l'approximation des mesures de Lebesgue et gaussiennes de l'union d'ensembles semi-algébriques en utilisant les hiérarchies de Lasserre. L'annexe B fournit du matériel supplémentaire pour le chapitre sur la réduction fonctionnelle des ensembles d'arbres de décision, incluant des détails techniques et des résultats expérimentaux supplémentaires.

Le Chapitre 8 conclut la thèse en résumant les contributions majeures et en discutant des perspectives de recherche futures. Enfin, une bibliographie complète est fournie à la fin de ce document pour référencer les travaux cités dans les différents chapitres.

CHAPITRE 2 REVUE DE LITTÉRATURE

Cette thèse aborde plusieurs problématiques distinctes. Nous allons donc consacrer ce chapitre à une revue de la littérature relative à chacune d'entre elles. Nous commencerons par présenter une revue du problème généralisé des moments, qui est au cœur des deux premières contributions. Ensuite, nous aborderons les travaux liés à la hiérarchie de Lasserre et son application au problème du sous-ensemble stable maximal d'un graphe. Nous poursuivrons avec une revue sur les ensembles d'arbres de décision et les techniques de réduction de leur taille. Enfin, nous terminerons par une revue des méthodes de résolution des problèmes d'optimisation linéaire, et plus particulièrement dans un contexte de dégénérescence, en utilisant la méthode du simplexe amélioré. Dans la suite de ce chapitre, et pour chaque section, nous présenterons certains résultats sous forme de théorèmes, de propositions ou de lemmes. Afin de mieux saisir l'importance de ces résultats, nous détaillerons les notations employées, les hypothèses requises ainsi que les démonstrations des résultats les plus significatifs. Pour chaque section, les notations seront réinitialisées.

2.1 Problème de moments généralisé

Dans cette section, nous considérons $n \in \mathbb{N}$ et $\mathcal{K}$ un sous-ensemble de $\mathbb{R}^n$. L'anneau des polynômes à n variables réelles $\mathbf{x} = (x_1, \ldots, x_n)$ est noté $\mathbb{R}[\mathbf{x}]$. Le problème de moments généralisé, introduit dans [27], s'exprime comme suit :

$$\sup_{\mu} \int_{\mathbb{R}^n} c(\mathbf{x})\, d\mu(\mathbf{x}) \tag{2.1a}$$

$$\int_{\mathbb{R}^n} h_i(\mathbf{x})\, d\mu(\mathbf{x}) \leq b_i, \quad \forall i \in \mathcal{I}, \tag{2.1b}$$

$$\mu \in \mathcal{M}_+(\mathcal{K}), \tag{2.1c}$$

où $\mathcal{I}$ est un ensemble dénombrable d'indices, c et h_i (pour $i \in \mathcal{I}$) sont des polynômes, b_i des scalaires, et $\mathcal{M}_+(\mathcal{K})$ désigne le cône des mesures boréliennes positives sur $\mathcal{K}$. Ce problème consiste à déterminer une mesure borélienne positive μ supportée sur $\mathcal{K}$ qui maximise l'intégrale de c tout en satisfaisant les contraintes linéaires (2.1b) imposées par les fonctions h_i et les scalaires b_i.

Le problème de moments généralisé (2.1) est un problème d'optimisation convexe défini

sur l'espace des mesures boréliennes positives, lequel est de dimension infinie, ce qui le rend souvent intractable. Pour le résoudre, on peut reformuler le problème en termes de *moments* de la mesure μ, ce qui permet de le transformer en un problème d'optimisation semi-définie. Nous commençons par présenter les notions de moments d'une mesure et de matrice de localisation, qui constituent des outils essentiels pour reformuler le problème de moments généralisé et ainsi permettre sa résolution.

Définition 2.1 (Moments d'une mesure)**.** Soit μ une mesure borélienne sur $\mathbb{R}^n$ et $\alpha \in \mathbb{N}^n$. Le moment d'ordre α de μ est donné par :

$$\boldsymbol{\mu}_\alpha = \int_{\mathbb{R}^n} \mathbf{x}^\alpha \, d\mu(\mathbf{x}), \tag{2.2}$$

avec $\mathbf{x}^\alpha = \prod_{i=1}^n x_i^{\alpha_i}$.

Autrement dit, le moment d'ordre α représente ainsi l'intégrale du produit des coordonnées de $\mathbf{x}$ élevées à la puissance α par rapport à μ.

2.1.1 Reformulation du problème de moments généralisé

L'objectif de cette section est de reformuler le problème de moments généralisé (2.1) en termes de moments de la mesure μ. Nous exprimerons la fonction objectif (2.1a) et les contraintes (2.1b) en fonction de ces moments, en introduisant les notions de matrice des moments et de matrice de localisation, outils essentiels pour traiter la contrainte de support (2.1c). Dans le reste de cette section, nous supposons que $\mathcal{K}$ est un ensemble semi-algébrique, défini par des inégalités polynomiales $\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^n \colon g_j(\mathbf{x}) \geq 0,\ j \in \mathcal{J}\}$, où $\mathcal{J}$ est un ensemble fini d'indices et g_j des polynômes pour $j \in \mathcal{J}$. Pour simplifier, nous supposons que $\mathcal{J}$ contient l'indice 0 avec $g_0(\mathbf{x}) = 1$.

Fonction objectif

Pour exprimer la fonction objectif (2.1a) en termes de moments, nous écrivons le polynôme c sous la forme $c(\mathbf{x}) = \sum_{\alpha \in \mathbb{N}^n} c_\alpha \mathbf{x}^\alpha$. Ainsi, la fonction objectif devient :

$$\int_{\mathbb{R}^n} c(\mathbf{x}) d\mu(\mathbf{x}) = \sum_{\alpha \in \mathbb{N}^n} c_\alpha \boldsymbol{\mu}_\alpha. \tag{2.3}$$

Contraintes linéaires

De manière similaire, si on écrit les polynômes h_i sous la forme $h_i(\mathbf{x}) = \sum_{\alpha\in\mathbb{N}^n} h_{i,\alpha}\mathbf{x}^\alpha$, les contraintes (2.1b) deviennent :

$$\sum_{\alpha\in\mathbb{N}^n} h_{i,\alpha}\boldsymbol{\mu}_\alpha \leq b_i, \quad \forall i \in \mathcal{I}. \tag{2.4}$$

Il est important de noter que les sommes dans les expressions (2.3) et (2.4) sont finies, car les polynômes c, h_i sont de degré fini.

Matrice de localisation

La contrainte de support de μ est gérée à l'aide des matrices de localisation, qui permettent de reformuler cette condition sous forme d'une contrainte de semi-définition positive. Nous noterons :

$$\mathbb{N}^n_p = \left\{\gamma \in \mathbb{N}^n : \sum_{i=1}^n \gamma_i \leq p\right\}.$$

Définition 2.2 (Matrice de localisation)**.** Soit $g(\mathbf{x}) = \sum_{\gamma\in\mathbb{N}^n} g_\gamma\, \mathbf{x}^\gamma$ un polynôme de degré $d \in \mathbb{N}$, et $\boldsymbol{\mu}$ une suite de moments. La matrice de localisation est définie par :

$$\mathbf{M}_p(\boldsymbol{\mu}\star g) = \left(\sum_{\gamma\in\mathbb{N}^n} g_\gamma\, \boldsymbol{\mu}_{\alpha+\beta+\gamma}\right)_{\alpha,\,\beta\in\mathbb{N}^n_p}.$$

Il faut noter que la somme dans la définition (2.2) est finie, car le polynôme g est de degré fini. La matrice de localisation $\mathbf{M}_p(\boldsymbol{\mu}\star g)$ est une matrice symétrique dans $\mathbb{R}^{\mathbb{N}^n_p\times\mathbb{N}^n_p}$.

Remarque 2.3. Quand g est le polynôme constant 1, la matrice de localisation $\mathbf{M}_p(\boldsymbol{\mu}\star 1)$ est appelée la matrice des *moments* de la suite $(\boldsymbol{\mu}_\alpha)_{\alpha\in\mathbb{N}^n}$. On la note simplement $\mathbf{M}_p(\boldsymbol{\mu})$.

Lemme 2.1. *Soit μ une mesure borélienne positive sur $\mathbb{R}^n$ et $g(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ un polynôme de degré $d \in \mathbb{N}$. Alors, pour tout entier $p \in \mathbb{N}$ et tout polynôme $f(\mathbf{x}) = \sum_{\alpha\in\mathbb{N}^n_p} f_\alpha\, \mathbf{x}^\alpha$ de degré au plus p, on a*

$$\mathbf{f}^\intercal \mathbf{M}_p(\boldsymbol{\mu}\star g)\,\mathbf{f} = \int_{\mathbb{R}^n} g(\mathbf{x})\, f(\mathbf{x})^2\, d\mu(\mathbf{x}). \tag{2.5}$$

où $\mathbf{f} = (f_\alpha)_{\alpha\in\mathbb{N}^n_p} \in \mathbb{R}^{\mathbb{N}^n_p}$.

Démonstration. Pour tout $\mathbf{f} = (f_\alpha)_{\alpha\in\mathbb{N}^n_p} \in \mathbb{R}^{\mathbb{N}^n_p}$, on écrit le polynôme $g(\mathbf{x}) = \sum_{\gamma\in\mathbb{N}^n} g_\gamma\, \mathbf{x}^\gamma$

comme une série de puissances. Il s'ensuit que

$$\begin{aligned}
\mathbf{f}^\intercal \mathbf{M}_p(\boldsymbol{\mu} \star g)\,\mathbf{f} &= \sum_{\alpha,\beta\in\mathbb{N}^n_p} f_\alpha \left(\sum_{\gamma\in\mathbb{N}^n} g_\gamma\, \boldsymbol{\mu}_{\alpha+\beta+\gamma} \right) f_\beta \\
&= \sum_{\alpha,\beta\in\mathbb{N}^n_p} f_\alpha \left(\sum_{\gamma\in\mathbb{N}^n} g_\gamma \int_{\mathbb{R}^n} \mathbf{x}^{\alpha+\beta+\gamma}\, d\mu(\mathbf{x}) \right) f_\beta \\
&= \int_{\mathbb{R}^n} \left(\sum_{\alpha,\beta\in\mathbb{N}^n_p} \sum_{\gamma\in\mathbb{N}^n} f_\alpha\, \mathbf{x}^\alpha\, g_\gamma\, \mathbf{x}^\gamma\, f_\beta\, \mathbf{x}^\beta \right) d\mu(\mathbf{x}) \\
&= \int_{\mathbb{R}^n} \left(\sum_{\alpha\in\mathbb{N}^n_p} f_\alpha\, \mathbf{x}^\alpha \right) \left(\sum_{\beta\in\mathbb{N}^n_p} f_\beta\, \mathbf{x}^\beta \right) \left(\sum_{\gamma\in\mathbb{N}^n} g_\gamma\, \mathbf{x}^\gamma \right) d\mu(\mathbf{x}) \\
&= \int_{\mathbb{R}^n} g(\mathbf{x})\, f(\mathbf{x})^2\, d\mu(\mathbf{x}).
\end{aligned}$$

□

Corollaire 2.2. *Soit μ une mesure borélienne positive sur $\mathbb{R}^n$, dont le support est inclus dans $\{\mathbf{x} \in \mathbb{R}^n \colon g(\mathbf{x}) \geq 0\}$. Alors, pour tout entier $p \in \mathbb{N}$, $\mathbf{M}_p(\boldsymbol{\mu}\star g)$ est positive semi-définie.*

Démonstration. Le corollaire découle immédiatement du Théorème 2.1 en observant que si μ est une mesure borélienne positive, dont le support est inclus dans $\{\mathbf{x} \in \mathbb{R}^n \colon g(\mathbf{x}) \geq 0\}$, alors

$$\int_{\mathbb{R}^n} g(\mathbf{x}) f(\mathbf{x})^2 d\mu(\mathbf{x}) = \int_{\mathbf{x}\colon\, g(\mathbf{x})\geq 0} g(\mathbf{x}) f(\mathbf{x})^2 d\mu(\mathbf{x}) \geq 0,$$

pour tout polynôme f. □

Remarque 2.4. Si on applique le Théorème 2.2 à une mesure borélienne positive μ et au polynôme constant 1, on obtient que la matrice des moments $\mathbf{M}_p(\boldsymbol{\mu})$ est positive semi-définie pour tout $p \in \mathbb{N}$.

Les résultats précédents montrent que si $\mu \in \mathcal{M}_+(\mathcal{K})$, alors les matrices de localisation $\mathbf{M}_p(\boldsymbol{\mu} \star g_j)$ pour $j \in \mathcal{J}$ sont positives semi-définies pour tout $p \in \mathbb{N}$. On établit ainsi l'inclusion suivante.

Résultat 2.1.

$$\mathcal{M}_+(\mathcal{K}) \subset \{\mu \in \mathcal{M}(\mathbb{R}^n) : \mathbf{M}_p(\boldsymbol{\mu} \star g_j) \succeq 0,\ \forall j \in \mathcal{J},\ \forall p \in \mathbb{N}\}\,. \tag{2.6}$$

Pour montrer que l'inclusion est une égalité, il suffit de montrer que si $(\boldsymbol{\mu})_{\alpha\in\mathbb{N}^n}$ est une suite telle que les matrices $\mathbf{M}_p(\boldsymbol{\mu} \star g_j)$ pour $j \in \mathcal{J}$ sont positives semi-définies pour tout $p \in \mathbb{N}$,

alors il existe une mesure borélienne positive μ supportée sur $\mathcal{K}$ telle que $(\boldsymbol{\mu}_\alpha)_{\alpha\in\mathbb{N}^n}$ est la suite des moments de μ. La suite de cette section est consacrée à l'étude des conditions sous lesquelles cette égalité est vérifiée.

Somme de carrés

Définition 2.5 (Somme de carrés)**.** On dit qu'un polynôme $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ est une somme de carrés si $f(\mathbf{x}) = \sum_{i=1}^m q_i(\mathbf{x})^2$ pour des polynômes q_i pour $i \in \{1,\dots,m\}$.

Lemme 2.3. *Soit $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ un polynôme de degré $2d$. Alors $f(\mathbf{x})$ est une somme de carrés si et seulement s'il existe une matrice $\mathbf{A} \in \mathbb{R}^{\mathbb{N}^n_d\times\mathbb{N}^n_d}$ semi-définie positive telle que*

$$f(\mathbf{x}) = \sum_{\alpha,\,\beta\in\mathbb{N}^n_d} \mathbf{A}_{\alpha,\beta}\,\mathbf{x}^\alpha\,\mathbf{x}^\beta.$$

Démonstration. Si $f(\mathbf{x}) = \sum_{i=1}^m q_i(\mathbf{x})^2$, alors pour tout $\alpha \in \mathbb{N}^n_d$, on a

$$f(\mathbf{x}) = \sum_{i=1}^m q_i(\mathbf{x})^2 = \sum_{i=1}^m \left(\sum_{\alpha\in\mathbb{N}^n_d} q_{i,\alpha}\,\mathbf{x}^\alpha\right)^2 = \sum_{\alpha,\,\beta\in\mathbb{N}^n_d}\sum_{i=1}^m q_{i,\alpha}q_{i,\beta}\mathbf{x}^\alpha\mathbf{x}^\beta.$$

On pose $\mathbf{q}_i \coloneqq (q_{i,\alpha})_{\alpha\in\mathbb{N}^n_d}$ et $\mathbf{A} \coloneqq \sum_{i=1}^m \mathbf{q}_i\,\mathbf{q}_i^\intercal \succeq 0$. On a alors $f(\mathbf{x}) = \sum_{\alpha,\,\beta\in\mathbb{N}^n_d} \mathbf{A}_{\alpha,\beta}\mathbf{x}^\alpha\mathbf{x}^\beta$. La réciproque est immédiate en décomposant $\mathbf{A} = \sum_{i=1}^m \mathbf{q}_i\,\mathbf{q}_i^\intercal$ en m matrices de rang 1. □

On note $\boldsymbol{\Sigma}^2[\mathbf{x}]$ l'ensemble des polynômes qui sont des sommes de carrés et $\boldsymbol{\Sigma}^2_d[\mathbf{x}]$ l'ensemble des polynômes de degré au plus d qui sont des sommes de carrés. De manière similaire, on peut définir le module quadratique associé à un ensemble de polynômes $g_j, j \in \mathcal{J}$ comme

$$\boldsymbol{\Sigma}^2(g_j, j\in\mathcal{J})[\mathbf{x}] = \left\{ f \in \mathbb{R}[\mathbf{x}] \colon f(\mathbf{x}) = \sum_{j\in\mathcal{J}} \sigma_j(\mathbf{x})\,g_j(\mathbf{x}),\ \sigma_j \in \boldsymbol{\Sigma}^2[\mathbf{x}],\ j\in\mathcal{J} \right\}.$$

Définition 2.6 (Condition d'Archimède)**.** On dit que l'ensemble semi-algébrique $\mathcal{K} = \{\mathbf{x}\in\mathbb{R}^n \colon g_j(\mathbf{x}) \geq 0,\ j\in\mathcal{J}\}$ satisfait la condition d'Archimède s'il existe $R \geq 0$ tel que le polynôme $R - \|\mathbf{x}\|_2^2$ est un élément de $\boldsymbol{\Sigma}^2(g_j, j\in\mathcal{J})[\mathbf{x}]$.

Théorème 2.4 (Théorème de Positivstellensatz [28, Lemme 3.2])**.** *Soit $\lambda \in \mathbb{R}[\mathbf{x}]^*$ une forme linéaire de $\mathbb{R}[\mathbf{x}]$ et $\mathcal{K} = \{\mathbf{x}\in\mathbb{R}^n \colon g_j(\mathbf{x}) \geq 0,\ j\in\mathcal{J}\}$ un ensemble semi-algébrique compact qui satisfait la condition d'Archimède. Si $\lambda(f)$ est positif pour tout $f \in \boldsymbol{\Sigma}^2(g_j, j\in\mathcal{J})[\mathbf{x}]$,*

alors il existe une mesure borélienne positive $\mu \in \mathcal{M}_+(\mathcal{K})$ *telle que*

$$\lambda(f) = \int_{\mathbb{R}^n} f(\mathbf{x})\, d\mu(\mathbf{x})$$

pour tout $f \in \mathbb{R}[\mathbf{x}]$.

Afin de montrer que l'inclusion inverse du Résultat 2.1 est vérifiée, on fixe une suite $(\boldsymbol{\mu}_\alpha)_{\alpha\in\mathbb{N}^n}$ telle que les matrices $\mathbf{M}_p(\boldsymbol{\mu})$ et $\mathbf{M}_p(\boldsymbol{\mu} \star g_j)$ pour $j \in \mathcal{J}$ sont positives semi-définies pour tout $p \in \mathbb{N}$. On définit alors la forme linéaire λ par $\lambda(f) = \sum_{\alpha\in\mathbb{N}^n} f_\alpha\, \boldsymbol{\mu}_\alpha$ pour tout $f \in \mathbb{R}[\mathbf{x}]$ tel que $f(\mathbf{x}) = \sum_{\alpha\in\mathbb{N}^n} f_\alpha\, \mathbf{x}^\alpha$. On suppose également que l'ensemble $\mathcal{K}$ est un ensemble semi-algébrique compact qui satisfait la condition d'Archimède. Pour tout polynôme $f \in \boldsymbol{\Sigma}^2(g_j, j \in \mathcal{J})[\mathbf{x}]$ de degré $2p$, on a

$$f(\mathbf{x}) = \sum_{j\in\mathcal{J}} \sigma_j(\mathbf{x})\, g_j(\mathbf{x}) = \sum_{\alpha,\,\beta\in\mathbb{N}^n_p} \sum_{j\in\mathcal{J}} \mathbf{A}_{j,\alpha,\beta} \left(\sum_{\gamma\in\mathbb{N}^n} g_{j,\gamma}\, \mathbf{x}^{\alpha+\beta+\gamma} \right),$$

où $\mathbf{A}_j$ sont des matrices symétriques semi-définies positives. On a alors

$$\lambda(f) = \sum_{\alpha\in\mathbb{N}^n} f_\alpha\, \boldsymbol{\mu}_\alpha = \sum_{\alpha,\,\beta\in\mathbb{N}^n_p} \sum_{j\in\mathcal{J}} \mathbf{A}_{j,\alpha,\beta} \left(\sum_{\gamma\in\mathbb{N}^n} g_{j,\gamma}\, \boldsymbol{\mu}_{\alpha+\beta+\gamma} \right) = \sum_{j\in\mathcal{J}} \langle \mathbf{A}_j, \mathbf{M}_p(\boldsymbol{\mu} \star g_j) \rangle \geq 0.$$

En appliquant le Théorème 2.4, on obtient qu'il existe une mesure borélienne positive μ telle que $\lambda(f) = \int_{\mathbb{R}^n} f(\mathbf{x})\, d\mu(\mathbf{x})$ pour tout $f \in \mathbb{R}[\mathbf{x}]$. On a donc montré que l'inclusion inverse du Résultat 2.1 est vérifiée. On peut alors résumer les résultats précédents dans le résultat suivant.

Résultat 2.2. Soit $\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^n \colon g_j(\mathbf{x}) \geq 0,\ j \in \mathcal{J}\}$ un ensemble semi-algébrique compact qui satisfait la condition d'Archimède. Alors, pour tout polynôme c, les polynômes h_i pour $i \in \mathcal{I}$ et les scalaires b_i, le problème de moments généralisé (2.1) est équivalent au problème d'optimisation suivant :

$$\sup_{\boldsymbol{\mu}\in\mathbb{R}^{\mathbb{N}^n}} \sum_{\alpha\in\mathbb{N}^n} c_\alpha\, \boldsymbol{\mu}_\alpha \tag{2.7a}$$

$$\sum_{\alpha\in\mathbb{N}^n} h_{i,\alpha}\, \boldsymbol{\mu}_\alpha \leq b_i,\ \forall i \in \mathcal{I}, \tag{2.7b}$$

$$\mathbf{M}_p(\boldsymbol{\mu} \star g_j) \succeq 0,\ \forall j \in \mathcal{J},\ \forall p \in \mathbb{N}. \tag{2.7c}$$

Il est important de noter que le problème (2.7) est un problème d'optimisation semi-définie sur les moments de la mesure μ.

2.1.2 Hiérarchie de Lasserre

Pour résoudre le problème obtenu au Résultat 2.2, on peut utiliser la hiérarchie de Lasserre introduite dans [27]. Celle-ci repose sur une relaxation du problème de moments en une suite de programmes semi-définis (SDP) de degré croissant. Plus précisément, pour un degré p donné, la hiérarchie de Lasserre consiste à résoudre le programme semi-défini suivant :

$$\begin{aligned} \sup_{\boldsymbol{\mu}\in\mathbb{R}^{\mathbb{N}^n_{2p}}} \quad & \sum_{|\alpha|\leq 2p} c_\alpha \boldsymbol{\mu}_\alpha \\ & \sum_{|\alpha|\leq 2p} h_{i,\alpha}\boldsymbol{\mu}_\alpha \leq b_i,\ \forall i \in \mathcal{I} \\ & \mathbf{M}_{p-d_j}\left(\boldsymbol{\mu} \star g_j\right) \succeq 0,\ \forall j \in \mathcal{J} \end{aligned} \tag{2.8}$$

où $d_j = \left\lceil \frac{\deg(g_j)}{2} \right\rceil$. La solution de ce programme, un vecteur des moments $\boldsymbol{\mu}$, constitue une approximation de la solution optimale du problème initial. En augmentant le degré p, on obtient une suite de solutions $\boldsymbol{\mu}^{(p)}$ qui, sous certaines conditions, converge vers la solution optimale du problème initial.

2.1.3 Convergence de la hiérarchie de Lasserre

La convergence de cette hiérarchie a été étudiée dans [27]. Plus précisément, si le problème (2.1) est faisable et que l'ensemble semi-algébrique $\mathcal{K}$ satisfait la condition d'Archimède, alors la suite de solutions $\boldsymbol{\mu}^{(p)}$ converge faiblement vers la solution optimale du problème initial.

Théorème 2.5 (Convergence des valeurs de la hiérarchie de Lasserre [27, Théorème 1])**.** *Si $\mathcal{K} = \{x \in \mathbb{R}^n \mid g_i(x) \geq 0,\ i \in \mathcal{I}\}$ est un ensemble semi-algébrique compact satisfaisant la condition d'Archimède et que le problème (2.1) est faisable, alors la valeur optimale de ce dernier correspond à la limite des valeurs optimales des programmes (2.8) lorsque p tend vers l'infini.*

Théorème 2.6 (Convergence des moments de la hiérarchie de Lasserre [27, Théorème 2])**.** *Sous les mêmes hypothèses, la suite de solutions $\boldsymbol{\mu}^{(p)}$ converge faiblement vers une solution optimale du problème (2.1) lorsque p tend vers l'infini.*

Le problème des moments généralisé possède de nombreuses applications en optimisation et en analyse numérique. Nous présentons ici quelques-unes de ces applications qui sont particulièrement pertinentes dans le cadre de cette thèse. La première application concerne la résolution de problèmes d'optimisation polynomiale. Celle-ci revêt un intérêt particulier dans le cadre de cette thèse, puisque, dans le Chapitre 4, nous procéderons à une analyse de la hiérarchie de Lasserre pour résoudre le problème du plus grand sous-ensemble indépendant d'un graphe selon différentes formulations polynomiales. La seconde application porte sur le calcul de la mesure d'un ensemble semi-algébrique. Celle-ci est également pertinente dans le cadre de cette thèse, car dans le Appendix A nous utiliserons la hiérarchie de Lasserre afin de développer une méthode d'approximation de la mesure d'une union finie d'ensembles semi-algébriques.

2.1.4 Optimisation polynomiale

Cette sous-section étudie l'utilisation du problème de moments généralisé pour résoudre des problèmes d'optimisation polynomiale. Nous considérons le problème suivant :

$$\inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) : \; g_j(\mathbf{x}) \geq 0,\, j = 1,\ldots,m, \tag{2.9}$$

où $f(\mathbf{x})$ et $g_j(\mathbf{x})$ sont des polynômes en $\mathbf{x}$. Ce problème se reformule en un problème de moments généralisés, comme montré dans [12] :

$$\inf_{\mu} \int_{\mathbb{R}^n} f(\mathbf{x})\, d\mu\,(\mathbf{x}) \; : \; \int_{\mathbb{R}^n} d\mu\,(\mathbf{x}) = 1, \quad \mu \in \mathcal{M}_+\,(\mathcal{K})\,, \tag{2.10}$$

avec $\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^n : \; g_j(\mathbf{x}) \geq 0,\, j = 1,\ldots,m\}$. Dans [12], il est démontré que les problèmes (2.9) et (2.10) sont équivalents. Ainsi, ce dernier appartient à la catégorie des problèmes de moments généralisé. La contrainte $\mu \in \mathcal{M}_+\,(\mathcal{K})$ garantit que la mesure μ est supportée sur l'ensemble $\mathcal{K}$, restreignant ainsi la recherche du minimiseur global de $f(\mathbf{x})$ aux points satisfaisant $g_j(\mathbf{x}) \geq 0$ pour $j = 1,\ldots,m$. Le problème (2.10) peut être résolu à l'aide de la hiérarchie de Lasserre, comme décrit dans [12]. Toutefois, sa résolution fournit une vecteur de moments de la mesure μ, et l'extraction d'un minimiseur global $\mathbf{x}^*$ pour le problème (2.9) à partir de ces moments n'est pas immédiate. Le lecteur est invité à consulter [12, 29] pour plus de détails sur le processus d'extraction, qui n'est pas un objet principal de cette thèse.

2.1.5 Mesure d'un semialgébrique

Dans cette sous-section, nous examinons l'utilisation du problème de moments généralisé pour calculer la mesure d'un ensemble semialgébrique. Ce problème, bien connu en géométrie algorithmique, consiste à déterminer le volume d'un ensemble semialgébrique par rapport à une mesure donnée ν. Il est établi que ce problème demeure complexe, même dans le cas d'un polytope simple [2]. Pour y remédier, des méthodes probabilistes ont été proposées [2,3]. Ces approches reposent sur la génération de points aléatoires dans l'ensemble et l'estimation de sa mesure en fonction du nombre de points tombant dans l'ensemble, sans toutefois garantir que les bornes obtenues soient monotones en fonction du nombre de points générés. Afin de préserver cette monotonie, [6] suggère de reformuler le problème de calcul de la mesure d'un ensemble semialgébrique en un problème de moments généralisés. Le problème consiste alors à calculer $\nu(\mathcal{K})$, où $\mathcal{K}$ est un ensemble semialgébrique défini par les inégalités polynomiales $g_i(\mathbf{x}) \geq 0,\ i = 1, \ldots, m$ et ν est une mesure borélienne fixée sur $\mathbb{R}^n$. On peut ainsi reformuler le problème de la manière suivante :

$$\sup_{\mu} \int_{\mathbb{R}^n} d\mu\left(\mathbf{x}\right) \ : \ \nu - \mu \in \mathcal{M}_+\left(\mathbb{R}^n\right),\ \mu \in \mathcal{M}_+\left(\mathcal{K}\right), \tag{2.11}$$

Pour résoudre ce problème, [6] propose la méthode suivante : on suppose que les moments de la mesure ν peuvent être calculés sur un ensemble semialgébrique $\mathcal{K}_0$ contenant $\mathcal{K}$. On cherche alors une mesure μ qui maximise le volume de $\mathcal{K}$ tout en veillant à ce que $\nu - \mu$ demeure une mesure positive sur $\mathcal{K}_0$.

Comme nous l'avons vu, le problème de moments généralisé est un outil puissant pour résoudre des problèmes d'optimisation polynomiale et pour calculer la mesure d'un ensemble semi-algébrique. Il est également possible d'utiliser ce problème pour résoudre des problèmes de contrôle optimal [30,31], des problèmes de statistiques [32] et des problèmes de géométrie algorithmique [33]. Nous avons choisi de présenter que ces deux applications car elles sont particulièrement pertinentes dans le cadre de cette thèse.

2.1.6 Complexité de la hiérarchie de Lasserre

La complexité de la hiérarchie de Lasserre dépend du degré p de la relaxation. En effet, le nombre de variables du programme (2.8) est de l'ordre de $\binom{n+2p}{n}$, rendant la résolution de ce programme coûteuse en temps lorsque p augmente. Pour les problèmes d'optimisation polynomiale, la complexité a été récemment étudiée dans [34–37], ainsi que dans plusieurs autres travaux.

2.1.7 Le problème du sous-ensemble stable maximal

Le problème du sous-ensemble stable maximal est un problème fondamental en théorie des graphes. Il consiste à identifier un sous-ensemble de sommets d'un graphe tel qu'aucune paire de sommets de cet ensemble ne soit connectée par une arête. Formellement, étant donné un graphe $G = (V, E)$, un sous-ensemble $I \subseteq V$ est dit stable si, pour tout $i, j \in I$, $(i, j) \notin E$. Le problème du sous-ensemble stable maximal vise à trouver un sous-ensemble stable de cardinalité maximale. Ce problème est généralement NP-difficile [38]. Il peut être reformulé comme un problème d'optimisation polynomiale et résolu via la méthode des moments [12]. Nous allons présenter plusieurs formulations polynomiales de ce problème. Pour cela, introduisons les variables x_i pour tout $i \in V$, où $x_i = 1$ si le sommet i appartient au sous-ensemble stable, et $x_i = 0$ sinon. Afin de garantir que les variables x_i sont binaires, on impose les contraintes $x_i^2 - x_i = 0$ pour tout $i \in V$. Concernant la contrainte d'indépendance, plusieurs formulations sont possibles. Pour chaque arête $(i, j) \in E$, on peut ajouter l'une des trois contraintes suivantes :

— $x_i + x_j \leq 1$,
— $x_i x_j = 0$,
— $x_i x_j \leq 0$.

Une autre approche consiste à imposer la contrainte $x_i x_j = 0$ pour tout $(i, j) \in E$ et à relâcher les contraintes $x_i^2 - x_i = 0$ en $0 \leq x_i \leq 1$ pour tout $i \in V$. Ces formulations sont équivalentes, et chacune d'entre elles peut être résolue à l'aide de la méthode des moments. Cependant, la qualité des bornes obtenues dépend de la formulation choisie.

2.2 Réduction des ensembles d'arbres de décision

Dans cette section, nous présentons une revue des méthodes d'ensembles d'arbres de décision, en mettant l'accent sur les arbres de décision, les forêts aléatoires et les méthodes de boosting. Nous abordons également les techniques de réduction des ensembles d'arbres de décision, en distinguant les méthodes de réduction optimale et les méthodes de réduction fidèles.

2.2.1 Arbres de décision

Les arbres de décision sont des méthodes d'apprentissage supervisé non paramétriques utilisées en classification et en régression [39]. Leur objectif est de créer un modèle prédisant la valeur d'une variable cible en apprenant des règles de décision simples, déduites des

caractéristiques des données. Un arbre de décision se présente sous la forme d'une structure arborescente : chaque nœud interne contient une condition sur une caractéristique et chaque branche en indique l'issue, tandis que les feuilles affichent les valeurs de la variable cible. La Figure 2.1 illustre un exemple où la racine teste x_1 en le comparant à 5 pour déterminer la suite du parcours, et les nœuds suivants appliquent la même logique, les feuilles indiquant les probabilités associées à chaque classe.

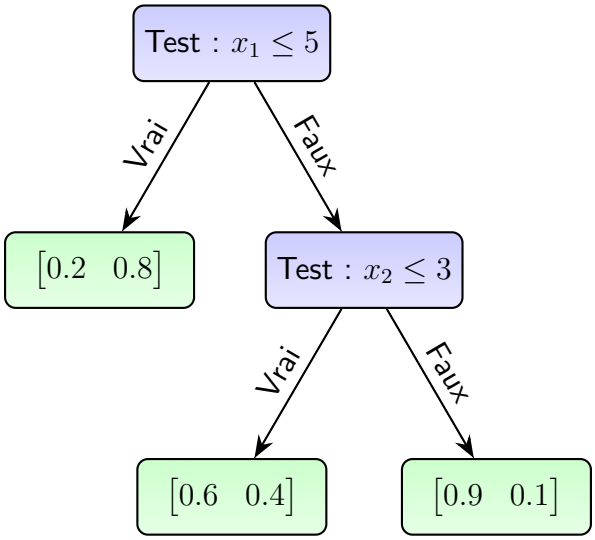


FIGURE 2.1 Exemple d'arbre de décision utilisé pour une classification binaire.

2.2.2 Entraînement d'un arbre de décision

L'algorithme CART (*Classification and Regression Trees*) [40] est fréquemment employé pour construire des arbres de décision. Ce procédé récursif consiste, à chaque étape, à choisir la caractéristique et la valeur qui divisent le mieux les données en sous-ensembles homogènes, selon un critère de division (les plus courants étant l'indice de Gini et l'entropie). L'indice de Gini mesure l'impureté d'un ensemble par la probabilité qu'un élément choisi au hasard soit mal classé, tandis que l'entropie quantifie l'incertitude en évaluant l'information nécessaire pour déterminer la classe d'un élément. La division se poursuit jusqu'à l'atteinte d'une condition d'arrêt, par exemple une profondeur maximale ou un nombre minimal d'observations dans une feuille, puis un élagage intervient pour supprimer les branches superflues et limiter le surajustement. En classification, les feuilles fournissent les probabilités des classes, tandis qu'en régression elles affichent la moyenne des observations.

2.2.3 Ensembles d'arbres de décision

Bien que simples et interprétables, les arbres de décision présentent souvent des performances limitées. Pour y remédier, il est courant de les combiner en ensembles, c'est-à-dire de construire plusieurs arbres de manière itérative et d'agréger leurs prédictions pour obtenir un résultat plus robuste. Les deux approches d'ensemble les plus répandues sont le *bagging* et le *boosting*.

Bagging

Le *bagging* (abréviation de *bootstrap aggregating*) consiste à entraîner plusieurs arbres sur des sous-ensembles aléatoires des données d'entraînement, obtenus par échantillonnage avec remplacement. Les prédictions ainsi obtenues sont agrégées (par vote majoritaire en classification ou par moyenne en régression) pour former la prédiction finale. Cette technique réduit la variance du modèle et atténue le surajustement. La forêt aléatoire (*Random Forest*) [41] en est un exemple typique, où chaque arbre est construit à partir d'un échantillon bootstrap et, à chaque nœud, un sous-ensemble aléatoire de caractéristiques est sélectionné pour la division. L'Algorithme 2.1 résume ce processus.

Algorithme 2.1 : Bagging

Entrée : Données d'entraînement $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, nombre d'arbres M, pourcentage de caractéristiques à sélectionner $f \in [0,1]$, pourcentage d'échantillons bootstrap $b \in [0,1]$ et critère de division.

Sortie : Modèle de Bagging F_M.

1 **Pour** $m = 1$ **to** M **faire**
2 Échantillonner avec remplacement $\lfloor b \times N \rfloor$ observations : $\{(\mathbf{x}_{j_i}, y_{j_i})\}_{i=1}^{\lfloor bN \rfloor}$;
3 Sélectionner $\lfloor f \times D \rfloor$ caractéristiques aléatoires ;
4 Entraîner un arbre de décision h_m sur les données échantillonnées et les caractéristiques sélectionnées, en appliquant le critère de division ;
5 **fin**
6 Le modèle de Bagging est $F_M(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} h_m(\mathbf{x})$;

Boosting

La méthode de *boosting* consiste à entraîner des arbres de décision de manière séquentielle, en se concentrant sur les observations mal prédites par les modèles précédents, afin de construire un modèle performant à partir de modèles faibles. L'algorithme de *Gradient Boosting* [42] est le plus utilisé. À chaque itération, un arbre est construit pour prédire le

résidu du modèle précédent, et la prédiction finale résulte de la somme des contributions de tous les arbres. Le processus est résumé dans l'Algorithme 2.2.

Algorithme 2.2 : Gradient Boosting

Entrée : Données d'entraînement $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, nombre d'arbres M, taux d'apprentissage η, modèle initial F_0 et fonction de perte $\mathcal{L}$.

Sortie : Modèle de Gradient Boosting F_M.

Pour $m = 1$ **to** M **faire**

  Calculer les résidus : $\widehat{r}_i = -\frac{\partial \mathcal{L}}{\partial \widehat{y}}\Big|_{\widehat{y}=F_{m-1}(\mathbf{x}_i)}$;

  Entraîner un arbre de décision h_m sur les résidus $\{(\mathbf{x}_i, \widehat{r}_i)\}_{i=1}^N$;

  Mettre à jour le modèle : $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \eta\, h_m(\mathbf{x})$;

fin

Des variantes du *Gradient Boosting* ont été développées pour améliorer la performance et la robustesse, telles que le *Gradient Boosting* stochastique, qui intègre une régularisation par l'ajout de bruit aux observations. XGBoost [43] et LightGBM [44] en sont deux exemples, utilisant respectivement une approximation de la fonction de perte pour accélérer l'entraînement et un algorithme de division en feuille pour réduire les temps d'entraînement et d'inférence.

Les ensembles d'arbres de décision capturent efficacement les relations dans les données tabulaires [16]. Toutefois, leur interprétabilité diminue lorsque leur taille (nombre d'arbres dans l'ensemble) devient trop important. Pour pallier ce problème, diverses méthodes de réduction de la taille de l'ensemble ont été proposées. La section suivante passe en revue ces approches.

2.2.4 Réduction des ensembles d'arbres de décision

La réduction de la taille des ensembles d'arbres a fait l'objet de nombreuses études, comme le montrent [45] et [46, Chapitre 6]. Les méthodes de réduction se répartissent en trois catégories principales : celles basées sur l'ordre, sur le clustering et sur l'optimisation. Les deux premières, de nature heuristique, n'assurent pas une réduction optimale. Les méthodes basées sur l'ordre ordonnent les arbres selon un critère et sélectionnent, de façon gloutonne, un sous-ensemble [21,47,48], tandis que celles basées sur le clustering regroupent les arbres similaires et retiennent un représentant par groupe [49]. Les approches d'optimisation formulent la réduction comme un problème d'optimisation. [50] propose un problème quadratique en nombres entiers avec une relaxation semi-définie, et [51, 52] étendent cette approche en intégrant des mesures de diversité. Une autre stratégie consiste à compresser

l'ensemble pour minimiser l'utilisation de mémoire [53], [54] étudiant l'encodage des forêts aléatoires en code binaire sans perte d'information. Enfin, [55] propose de simplifier les ensembles en réduisant le nombre de décisions distinctes afin de limiter le nombre de caractéristiques utilisées.

Méthodes de réduction optimale

Les méthodes de réduction optimale se sont avérées efficaces pour diminuer la taille des ensembles d'arbres de décision. [56] propose un modèle de programmation linéaire en nombres entiers pour sélectionner un sous-ensemble de nœuds par arbre, tandis que [23] présente une approche de suppression de nœuds dans une forêt aléatoire à l'aide d'un algorithme de descente de coordonnées. La structure arborescente se prête bien aux techniques d'optimisation, ce qui permet d'améliorer la justesse et l'interprétabilité des modèles [57].

Méthodes de réduction fidèles

Les méthodes de réduction fidèles garantissent que le modèle réduit conserve le comportement de l'ensemble initial. [58] propose de construire un arbre de décision fidèle à l'ensemble en utilisant un algorithme de programmation dynamique. Cet arbre reproduit le comportement prédictif de l'ensemble sur toute observation, bien que sa taille puisse croître de manière exponentielle par rapport aux arbres de l'ensemble initial et que le temps de calcul requis soit élevé pour des ensembles de grande taille.

2.3 Optimisation linéaire

L'optimisation linéaire est un domaine de recherche en mathématiques appliquées qui consiste à trouver le maximum ou le minimum d'une fonction linéaire sous contraintes linéaires. Les problèmes d'optimisation linéaire sont souvent modélisés sous la forme suivante :

$$\begin{aligned} \min \quad & c^\intercal x \\ & Ax = b \\ & x \geq 0 \end{aligned} \tag{2.12}$$

où x est le vecteur des variables de décision, c est le vecteur des coûts, A est la matrice des contraintes, et b est le vecteur des membres droits. Ces problèmes proviennent principalement des relaxations linéaires de problèmes d'optimisation en nombres entiers, lesquels

représentent un grand nombre de problèmes rencontrés dans l'industrie et la recherche opérationnelle. Souvent, ces problèmes industriels formulés en nombres entiers sont résolus en utilisant des méthodes de branchement [59] ou des coupes [60] reposant sur de nombreuses relaxations linéaires. Il est donc important de résoudre ces problèmes de manière efficace.

Lemme 2.7 (Théorème de la programmation linéaire)**.** *Si x^* est une solution optimale du problème d'optimisation linéaire* (2.12)*, alors x^* est un sommet ou se trouve sur une face du polyèdre défini par les contraintes du problème.*

Le Théorème 2.7 permet de résoudre les problèmes d'optimisation linéaire en vérifiant uniquement les sommets du polyèdre défini par les contraintes du problème.

Dualité Le dual du problème d'optimisation linéaire (2.12) est donné par :

$$\begin{aligned} \max \quad & b^\intercal y \\ & A^\intercal y + s = c \\ & s \geq 0 \end{aligned} \tag{2.13}$$

Théorème 2.8 (Théorème de dualité)**.** *Si x^* et (y^*, s^*) sont des solutions optimales respectives des problèmes d'optimisation linéaire* (2.12) *et* (2.13)*, alors $c^\intercal x^* = b^\intercal y^*$ et $x_j^* s_j^* = 0$ pour tout j.*

Le Théorème 2.8 est un résultat direct des conditions de Karush-Kuhn-Tucker [61], qui sont des conditions nécessaires pour l'optimalité d'un problème d'optimisation convexe.

2.3.1 Méthodes de résolution

La méthode du simplexe

Le simplexe primal [25] fut l'une des premières méthodes de résolution de problèmes d'optimisation linéaire. Il consiste à se déplacer d'un sommet à un autre du polyèdre défini par les contraintes du problème jusqu'à trouver un sommet optimal. À chaque itération du simplexe, les variables sont divisées en deux groupes : les variables de base et les variables hors base. Les variables hors base sont fixées à zéro et les variables de base sont calculées en fonction des contraintes. Pour effectuer une itération du simplexe, l'algorithme cherche à trouver une variable hors base qui peut remplacer une variable de base sans détériorer la valeur de la fonction objectif. Cette sélection de variable est appelée *pivot*. Plusieurs variantes des règles de pivot existent, la première étant la règle de Dantzig [25] qui consiste

à choisir la variable hors base ayant le coût réduit négatif le plus important [1]. Cette règle de pivot ne garantit pas la convergence à cause d'une propriété appelée *cycling*. Pour éviter le cycling, plusieurs règles de pivot ont été proposées, la plus connue étant la règle de Bland [62] qui consiste à choisir la variable hors base avec le plus petit indice lexicographique parmi les variables dont le coût réduit est négatif. Une autre règle de pivot est la règle de *steepest edge* [63] qui consiste à choisir la variable hors base produisant la meilleure amélioration de la fonction objectif. Une revue exhaustive des règles de pivot est donnée par [64].

La méthode du simplexe peut être utilisée pour résoudre le problème primal (2.12) ou le problème dual (2.13). Dans le cas primal, on parle de simplexe primal et, dans le cas dual, de simplexe dual.

La méthode des points intérieurs

La méthode des points intérieurs [65] est une autre méthode de résolution de problèmes d'optimisation linéaire. Elle consiste à suivre un chemin central défini par les solutions d'un système non linéaire d'équations. En effet, le Théorème 2.8 permet de définir un système d'équations non linéaires qui permet de trouver une solution optimale aux problèmes (2.12) et (2.13) :

$$Ax = b \tag{2.14a}$$
$$A^\intercal y + s = c \tag{2.14b}$$
$$x_j s_j = 0 \tag{2.14c}$$
$$(x, s) \geq 0 \tag{2.14d}$$

Si l'on remplace le second membre de l'équation (2.14c) par une constante κ ainsi que la contrainte (2.14d) avec $(x, s) > 0$ et que l'on résout le système d'équations, on obtient une solution $(x^\kappa, y^\kappa, s^\kappa)$ telle que $x^\kappa \to x^*$ et $s^\kappa \to s^*$ quand $\kappa \to 0$.

La méthode de descente de gradient

La méthode de descente de gradient [66] est une méthode de résolution de problèmes d'optimisation qui consiste à suivre le gradient de la fonction objectif pour trouver un minimum. Elle est généralement utilisée lorsque la fonction objectif n'est pas linéaire ; toutefois, des

1. Si tous les coûts réduits sont positifs, alors la solution est optimale.

travaux récents [67] ont montré que cette méthode peut également être appliquée aux problèmes d'optimisation linéaire.

2.3.2 Complexité et analyse

La complexité de la méthode du simplexe avec les règles de pivot connues jusqu'à présent est exponentielle dans le pire cas [68]. Cependant, en pratique et pour des problèmes non dégénérés, la méthode du simplexe reste relativement efficace. La méthode des points intérieurs présente une complexité polynomiale pour atteindre une solution ϵ-optimale [65]. Quant à la méthode de descente de gradient, elle n'offre pas, en général, de garantie de convergence et peut même rester bloquée dans un minimum local.

Comme mentionné précédemment, les problèmes d'optimisation linéaire sont souvent utilisés pour résoudre des problèmes d'optimisation en nombres entiers. On sera donc amené à effectuer des branchements sur les variables de décision. Ces branchements sont d'autant plus efficaces si la solution obtenue pour le problème linéaire est extrémale. C'est le cas lorsqu'on utilise la méthode du simplexe. Cependant, les méthodes des points intérieurs et de descente de gradient fournissent des solutions centrales non extrémales. Pour retrouver une solution extrémale, un algorithme de crossover est utilisé. En pratique, il est observé que cette étape de crossover est coûteuse en temps de calcul.

2.3.3 Dégénérescence

Définition 2.7 (Dégénérescence)**.** Un problème d'optimisation linéaire est dit dégénéré si une ou plusieurs variables de base sont nulles.

La dégénérescence est un problème courant en programmation linéaire. Elle peut entraîner plusieurs pivots sans amélioration de la fonction objectif, ce qui peut ralentir la convergence de la méthode du simplexe. Plusieurs techniques ont été proposées pour gérer la dégénérescence, qui se basent principalement sur le choix de la variable entrante et sortante [69] ou sur l'introduction de perturbations mineures [70] afin d'éliminer la dégénérescence. Une autre approche consiste à utiliser l'intuition géométrique qui dit qu'un sommet dégénéré d'un polyèdre de dimension n est l'intersection d'au moins $n-1$ faces du polyèdre [71]. La dégénérescence correspond donc à une redondance de l'information dans les contraintes du problème, et ainsi une version simplifiée du problème peut être obtenue en éliminant ces contraintes redondantes. Dans [72], les auteurs proposent une méthode basée sur la décomposition LU de la matrice de base pour détecter les contraintes redondantes. Dans [73], les auteurs proposent de reformuler le problème en considérant que la base est formée

uniquement des variables non nulles, afin d'éliminer la dégénérescence.

2.3.4 Algorithme du simplexe amélioré

Une autre approche permettant de surmonter les problèmes de dégénérescence est l'algorithme du simplexe amélioré [26]. Cet algorithme consiste à décomposer le problème initial en deux sous-problèmes plus simple à résoudre : un *problème réduit* et un *problème complémentaire*. Le problème réduit est un problème linéaire non dégénéré dont la taille est relativement inférieure à celle du problème original. Le problème complémentaire est un problème linéaire qui permet soit de retrouver une direction de descente pour la solution courante, soit de détecter une solution optimale. Formellement, en considérant que $\underline{x}$ est une solution réalisable du problème (2.12), $\mathcal{B}$ les variables de base et $\mathcal{N}$ les variables hors base, on peut définir $\mathcal{P} \coloneqq \{j : \underline{x}_j > 0\}$, l'ensemble des indices des variables de base non nulles, et $\mathcal{Z} \coloneqq \{j : \underline{x}_j = 0\}$, l'ensemble des indices des variables de base nulles. L'ensemble des variables est ainsi divisé en deux sous-ensembles : les variables compatibles avec $\mathcal{P}$, notées $\mathcal{C}$, et les variables incompatibles avec $\mathcal{P}$, notées $\mathcal{I}$. Une variable est dite compatible avec $\mathcal{P}$ si la colonne correspondante dans la matrice A peut s'exprimer comme une combinaison linéaire des colonnes associées aux variables de base non nulles ; sinon, elle est dite incompatible. Le problème réduit est défini comme suit :

$$\begin{aligned} \min \quad & c_{\mathcal{C}}^{\intercal} x_{\mathcal{C}} \\ & A_{:,\mathcal{C}} x_{\mathcal{C}} = b \\ & x_{\mathcal{C}} \geq 0. \end{aligned} \tag{2.15}$$

Résultat 2.3. Le problème réduit (2.15) est non dégénéré et peut être résolu efficacement en utilisant la méthode du simplexe.

D'autre part, le problème complémentaire est défini comme suit :

$$\min \quad -c_{\mathcal{P}}^{\intercal} u + c_{\mathcal{I}}^{\intercal} v, \tag{2.16a}$$

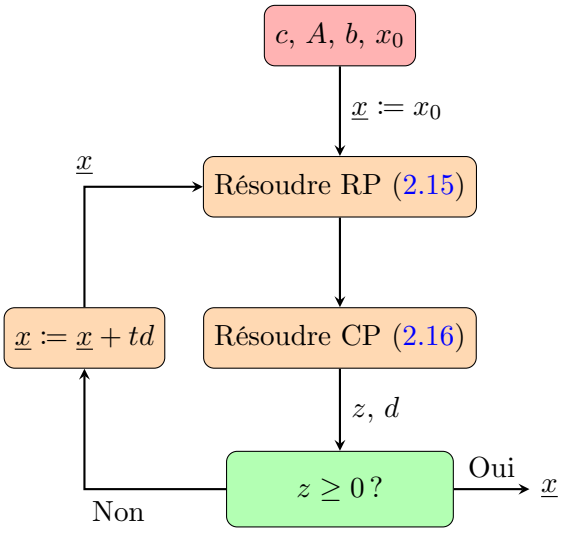
$$-A_{:,\mathcal{P}} u + A_{:,\mathcal{I}} v = 0, \tag{2.16b}$$

$$v \geq 0, \tag{2.16c}$$

$$\mathbf{1}^{\intercal} v = 1. \tag{2.16d}$$

On cherche à trouver une direction de descente pour la solution courante $\underline{x}$ en résolvant le problème complémentaire (2.16). Puisque les seules variables non nulles dans $\underline{x}$ sont celles

FIGURE 2.2 Algorithme du simplexe amélioré



de $\mathcal{P}$, on recherche une direction modifiant ces variables (représentées par u). Le problème réduit garantit que la modification des variables compatibles n'améliore pas la fonction objectif. Il convient donc de modifier les variables incompatibles, représentées par v, pour trouver une direction de descente. La contrainte (2.16b) assure que la direction recherchée est réalisable, permettant ainsi de considérer le cône des directions réalisables. Comme ce cône est non borné, on ajoute la contrainte de normalisation (2.16d) afin de le rendre borné.

Résultat 2.4. Le problème complémentaire (2.16) est non dégénéré, ou faiblement dégénéré (du point de vue dual), et peut être résolu efficacement à l'aide de la méthode du simplexe dual.

Le problème complémentaire (2.16) fournit une direction de descente pour la solution courante $\underline{x}$ si sa valeur optimale est strictement négative. Dans ce cas, on peut effectuer un pivot pour améliorer la solution courante. Si la valeur optimale du problème complémentaire est nulle ou positive, alors la solution courante est optimale.

Pour résumer l'algorithme du simplexe amélioré, on commence par une solution initiale x_0 et on résout le problème réduit (2.15) à l'optimalité à l'aide de la méthode du simplexe primal. En utilisant la solution optimale du problème réduit, on formule le problème complémentaire (2.16) et on le résout à l'optimalité à l'aide de la méthode du simplexe dual. Si la valeur optimale du problème complémentaire est strictement négative, on effectue un pivot pour améliorer la solution courante et on répète le processus. Sinon, la solution courante est optimale et on la retourne. L'algorithme du simplexe amélioré est résumé dans la Figure 2.2.

CHAPITRE 3 ORGANISATION DU TRAVAIL

Dans les chapitres précédents, une revue exhaustive de la littérature a été présentée, couvrant les fondements théoriques et les avancées significatives relatives aux thématiques abordées par cette thèse. Ce chapitre a pour objectif de délimiter la structure globale de cet ouvrage doctoral et d'exposer de manière systématique les contributions principales issues de nos travaux de recherche. Chaque section subséquente détaillera une contribution majeure, soulignant son originalité, sa méthodologie et ses résultats significatifs, souvent issus de publications dans des revues ou conférences de premier plan.

Nous débuterons par l'étude approfondie présentée dans le Chapitre 4. Ce chapitre est consacré à une analyse comparative des formulations polynomiales alternatives du problème du plus grand sous-ensemble stable d'un graphe. Une contribution essentielle de ce segment réside dans la démonstration, via un article publié, de l'impact significatif du choix de la formulation polynomiale sur la qualité des bornes obtenues par la méthode des hiérarchies de Lasserre. Notre démarche a d'abord impliqué une étude numérique rigoureuse, qui a révélé des variations notables dans les performances des bornes en fonction des différentes modélisations. Cette observation empirique a ensuite été étayée par une étude théorique approfondie, permettant de valider et de justifier mathématiquement ces phénomènes pour des graphes généraux. Ces résultats ouvrent des perspectives importantes pour l'optimisation et l'application plus efficace des hiérarchies de Lasserre dans la résolution de problèmes complexes.

Le Chapitre 5 présentera un deuxième axe de nos recherches, concrétisé par un article récemment publié. Ce chapitre introduira une nouvelle méthode innovante de réduction des forêts aléatoires. Notre approche vise à simplifier et à compresser ces ensembles d'arbres de décision, tout en garantissant qu'ils demeurent fonctionnellement identiques aux modèles originaux. Cette technique se distingue par sa capacité à préserver l'intégralité des performances prédictives des forêts, tout en réduisant considérablement leur complexité, ce qui est crucial pour des applications nécessitant une interprétabilité accrue, une inférence rapide ou un déploiement dans des environnements aux ressources limitées. L'efficacité et la robustesse de notre approche ont été validées par une étude numérique exhaustive menée sur divers ensembles de données, confirmant son comportement favorable dans des scénarios variés.

Enfin, le Chapitre 6 mettra en lumière notre troisième contribution majeure, également le sujet d'un de nos articles. Ce chapitre exposera des résultats théoriques fondamentaux qui

améliorent significativement la complexité algorithmique de la méthode du simplex primal amélioré, un algorithme central en programmation linéaire. En développant de nouvelles pistes théoriques et en proposant des raffinements conceptuels, nous démontrerons comment ces avancées se traduisent par des gains de performance substantiels dans la résolution de problèmes d'optimisation linéaire de grande envergure. Ces améliorations ont le potentiel d'étendre la portée et l'efficacité de cette méthode dans de multiples domaines d'application.

CHAPITRE 4 ARTICLE 1 : A NOTE ON THE LASSERRE HIERARCHY FOR DIFFERENT FORMULATIONS OF THE MAXIMUM INDEPENDENT SET PROBLEM

Authors: Youssouf Emine, Miguel F. Anjos, Zhao Sun and Andrea Lodi

Journal: Optimization Letters, 2021, Vol. 49, No. 1, pp. 30-34

Date de parution: 17 novembre 2020

Abstract In this note, we consider several polynomial optimization formulations of the maximum independent set problem and the use of the Lasserre hierarchy with these different formulations. We demonstrate using computational experiments that the choice of formulation may have a significant impact on the resulting bounds. We also provide theoretical justifications for the observed behavior.

Keywords: Maximum Independent Set; Polynomial Optimization; Dual Bounds; Lasserre Hierarchy

4.1 Introduction

Polynomial optimization and its close connections with semidefinite and conic optimization have attracted a lot of attention in recent years [8]. It is well known that semidefinite optimization has had a tremendous impact on combinatorial optimization, particularly with the groundbreaking results of Lovász and Schrijver [9] and Goemans and Williamson [10]. This motivates the study of the application of polynomial optimization to combinatorial optimization problems.

Moreover, combinatorial problems can typically be formulated in different ways, and it is known that different formulations of the same combinatorial problem may lead to different semidefinite relaxations and hence to different global bounds; the example of the maximum cut problem is explored from this perspective in [11].

Finally, various approaches have been proposed to construct hierarchies of semidefinite relaxations for (binary) combinatorial optimization problems (and applied to the stable set polytope). Lovász and Schrijver [9] used a sequence of lift-and-project operations to construct their hierarchy, while Lasserre's work [12] starts with a polynomial formulation and progressively refines it by providing another hierarchy. De Klerk and Pasechnik [13] used copositive programming to construct another hierarchy and yet another one follows by

the Reformulation-Linearization Technique approach by Sherali and Adams [14]. All these hierarchies have in common the property of converging to the optimal solution in a finite number of steps and a comparison among Sherali-Adams, Lovász-Schrijver and Lasserre hierarchies was carried out by Laurent [15].

In this note, we focus on the Lasserre approach and how it is impacted by using different polynomial formulations of the same problem. Specifically, we consider several polynomial optimization formulations of the maximum independent set problem and the use of the Lasserre hierarchy with these different formulations.

We demonstrate using computational experiments that the choice of formulation may have a significant impact on the resulting bounds. We also provide theoretical justifications for the observed behavior.

A specific comparison among hierarchies for the maximum independent set problem has been considered by Gvozdenović and Laurent [74]. They proved that the Lasserre's is tighter than the Lovász-Schrijver's, and tighter than De Klerk and Pasechnik's as well. However, all the comparisons in the literature used the Lasserre's hierarchy of one polynomial formulation. Our results show that this particular formulation is the best among a set of polynomial formulations, thus confirming the interest of the analysis in [74].

The paper is organized as follows. In Section 4.2, we give some preliminaries about polynomial optimization, while Section 4.3 introduces the maximum independent set problem and computationally motivates the interest of looking at different polynomial formulations for the problem. Section 4.4 provides the theoretical content of the paper and in Section 4.5 we draw some concluding remarks.

4.2 Preliminaries

Polynomial optimization is NP-hard in general, and the Lasserre hierarchy has great theoretical and practical appeal because it provides a sequence of tractable relaxations whose optimal objective values converge to the global optimum. We briefly review the construction of the Lasserre hierarchy (in the dual form). For more details about the Lasserre hierarchy, see e.g. [8, 12, 27, 75].

Given polynomials f, $g_1, \ldots, g_m$, we consider the following general polynomial optimization problem:

$$f_{min} = \min \ f(x) : \ g_j(x) \geq 0, \forall j = 1, \ldots, m. \tag{4.1}$$

Let $\{x^\alpha\}_{\alpha\in\mathbb{N}^n}$ be a canonical basis for $\mathbb{R}[x]$. Given $y = \{y_\alpha\} \in \mathbb{R}^{\mathbb{N}^n}$, we define $L_y : \mathbb{R}[x] \to \mathbb{R}$ as the linear functional which maps a polynomial function $f = \sum_{\alpha\in\mathbb{N}^n} f_\alpha x^\alpha$ (f_α are the coefficients of f in the canonical basis) to the real value

$$L_y(f) = \sum_{\alpha\in\mathbb{N}^n} f_\alpha y_\alpha.$$

The *moment matrix* $M_d(y)$ is the matrix of $\mathbb{R}^{\mathbb{N}^n_d\times\mathbb{N}^n_d}$ such that its entries are:

$$M_d(y)\,(\alpha,\beta) = y_{\alpha+\beta},\ \forall\alpha,\ \beta\ \in\ \mathbb{N}^n_d.$$

Given a polynomial function $\theta(x) = \sum_{\gamma\in\mathbb{N}^n} \theta_\gamma x^\gamma$, the *localizing matrix* $M_d\,(\theta\star y) \in \mathbb{R}^{\mathbb{N}^n_d\times\mathbb{N}^n_d}$ is the matrix of $\mathbb{R}^{\mathbb{N}^n_d\times\mathbb{N}^n_d}$ such that its entries are:

$$M_d\,(\theta\star y)\,(\alpha,\beta) = \sum_{\gamma\in\mathbb{N}^n} \theta_\gamma y_{\alpha+\beta+\gamma},\ \forall\alpha,\ \beta\ \in\ \mathbb{N}^n_d.$$

Let the degree of g_j be $2v_j$ or $2v_j - 1$. Then, for problem (4.1), the Lasserre relaxation of order d provides a lower bound ρ_d for f_{min}:

$$\rho_d = \min L_y(f) : \begin{cases} M_d(y) \succeq 0, \\ M_{d-v_j}(g_j \star y) \succeq 0,\ \forall j = 1,\dots,m, \\ L_y(1) = 1. \end{cases} \tag{4.2}$$

The sequence of Lasserre relaxations of increasing order $d = 1, 2, 3, \dots$ forms the Lasserre hierarchy.

4.3 Maximum Independent Set Problem

Given a graph $G = (V, E)$, the maximum independent set problem consists of determining the maximum cardinality of any subset of vertices such that no two vertices in that subset are connected by an edge of G. We consider four different formulations of this problem using quadratic polynomials. The formulations are:

$$\rho = \max \sum_{i\in V} x_i :$$

$$x_i x_j = 0, \quad \forall (i,j) \in E, \quad x_i^2 - x_i = 0, \quad \forall i \in V. \tag{4.3}$$

$$\text{or} \quad x_i x_j \leq 0, \quad \forall (i,j) \in E, \quad x_i^2 - x_i = 0, \quad \forall i \in V. \tag{4.4}$$

$$\text{or} \quad x_i + x_j \leq 1, \quad \forall (i,j) \in E, \quad x_i^2 - x_i = 0, \quad \forall i \in V. \tag{4.5}$$

$$\text{or} \quad x_i x_j = 0, \quad \forall (i,j) \in E, \quad 0 \leq x_i \leq 1, \quad \forall i \in V. \tag{4.6}$$

Let us compute the upper bounds arising from the Lasserre relaxation (4.2) of order $d = 1$ for each of the above four formulations. The bounds are reported in Table 4.1, where C_n denotes the cycle with n vertices and K_4 denotes the complete graph with 4 vertices.

Table 4.1 Bounds from the Lasserre relaxation of order $d = 1$ for different formulations

Graph	Optimal bound	Bound from (4.3)	Bound from (4.4)	Bound from (4.5)	Bound from (4.6)
C_3	1	1	1	1.5	3
C_4	2	2	2	2	4
K_4	1	1	1	2	4
C_5	2	2.236	2.236	2.5	5
C_6	3	3	3	3	6
C_7	3	3.318	3.318	3.5	7
Petersen graph	4	4	4	5	10

We observe that the bounds obtained using (4.3) and (4.4) are always equal, and are the best for all of these graphs. On the other hand, the bounds from (4.6) are consistently the weakest; indeed the bound obtained using (4.6) is always equal to the trivial upper bound $|V|$, as we prove formally in Theorem 4.1 below. Finally, the bounds from (4.5) are equal or moderately weaker than those from (4.3) and (4.4).

Although these results are only for 7 small graphs, they clearly show that the choice of formulation dramatically impacts the quality of the resulting bound. The remainder of this note is concerned with providing some theoretical justification for the results in Table 4.1.

4.4 Independent Set Formulations and Lasserre Relaxations

4.4.1 Notation

Let us consider the following polynomials of $\mathbb{R}^{|V|}$ for $(i,j) \in E$, $i \in V$:

$$g_{ij}^{+}(x) = x_i x_j,$$

$$
\begin{aligned}
g_{ij}^-(x) &= -x_i x_j, \\
l_{ij}(x) &= 1 - x_i - x_j, \\
h_i^+(x) &= x_i^2 - x_i, \\
h_i^-(x) &= -x_i^2 + x_i, \\
q_i^+(x) &= x_i, \\
q_i^-(x) &= 1 - x_i, \\
f(x) &= \sum_{i \in V} x_i.
\end{aligned}
$$

Let $(\mathbf{e}_i)_{i \in V}$ be the canonical basis of $\mathbb{R}^{|V|}$. For a fixed degree d, the Lasserre relaxations of order d of the above formulations are:

$$
\rho_d^{(1)} = \max\ L_y(f)\ : \begin{cases} M_d(y) \succeq 0, \\ M_{d-1}(g_{ij}^+ \star y) = 0,\ \forall (i,j) \in E, \\ M_{d-1}(h_i^+ \star y) = 0,\ \forall i \in V, \\ L_y(1) = 1. \end{cases}
$$

$$
\rho_d^{(2)} = \max\ L_y(f)\ : \begin{cases} M_d(y) \succeq 0, \\ M_{d-1}(g_{ij}^- \star y) = 0,\ \forall (i,j) \in E, \\ M_{d-1}(h_i^+ \star y) = 0,\ \forall i \in V, \\ L_y(1) = 1. \end{cases}
$$

$$
\rho_d^{(3)} = \max\ L_y(f)\ : \begin{cases} M_d(y) \succeq 0, \\ M_{d-1}(l_{ij} \star y) = 0,\ \forall (i,j) \in E, \\ M_{d-1}(h_i^+ \star y) = 0,\ \forall i \in V, \\ L_y(1) = 1. \end{cases}
$$

$$
\rho_d^{(4)} = \max\ L_y(f)\ : \begin{cases} M_d(y) \succeq 0, \\ M_{d-1}(g_{ij}^+ \star y) = 0,\ \forall (i,j) \in E, \\ M_{d-1}(q_i^+ \star y) = 0,\ \forall i \in V, \\ M_{d-1}(q_i^- \star y) = 0,\ \forall i \in V, \\ L_y(1) = 1. \end{cases}
$$

4.4.2 Value of $\rho_1^{(4)}$ for every graph G

Our first result is the proof that the optimal value of the Lasserre relaxation of order $d = 1$ using formulation (4.6) is equal to $|V|$ for every graph G.

Proposition 4.1. *For every graph G, $\rho_1^{(4)} = |V|$.*

Proof. For every feasible solution $\{y_\alpha\}$, we have $0 \leq y_{\mathbf{e}_i} \leq 1,\ \forall i \in V$, therefore

$$\sum_{i \in V} y_{\mathbf{e}_i} = L_y(f) \leq |V|.$$

To show attainment, consider $y^* = \{y^*_\alpha\}$ such that:

$$\begin{cases} y^*_0 = 1, & \\ y^*_{\mathbf{e}_i} = 1, & \forall i \in V, \\ y^*_{2\mathbf{e}_i} = |V| + 1, & \forall i \in V, \\ y^*_{\mathbf{e}_i + \mathbf{e}_j} = 0, & \forall (i,j) \in V^2,\ i \neq j. \end{cases}$$

It is straightforward to check that y^* is a feasible solution of the Lasserre relaxation of order $d = 1$ for formulation (4.6), and that this solution achieves the objective value $|V|$. □

4.4.3 Relationship between $\rho_d^{(1)}$ and $\rho_d^{(2)}$

The next proposition shows that the set of feasible solutions of the d order Lasserre relaxation for formulation (4.3) is a subset of the set of feasible solutions the relaxation with the same order for formulation (4.4). Moreover, for $d \geq 2$, the two feasible sets are equal, and hence so are the bounds.

Proposition 4.2. *For every graph G and order $d \geq 1$, $\rho_d^{(1)} \leq \rho_d^{(2)}$. Moreover, if $d \geq 2$, then $\rho_d^{(1)} = \rho_d^{(2)}$.*

Proof. The first claim follows from the observation that $M_{d-1}(g_{ij}^+ \star y) = 0$ implies $M_{d-1}(g_{ij}^- \star y) \succeq 0$.

To prove the second claim, let $y = \{y_\alpha\}$ be a feasible solution of the Lasserre hierarchy of order d for formulation (4.4). We know that for every $(\alpha, \beta) \in \mathbb{N}^{|V|}_{d-1} \times \mathbb{N}^{|V|}_{d-1}$

$$M_{d-1}(g_{ij}^- \star y)(\alpha, \beta) = -y_{\alpha+\beta+\mathbf{e}_i+\mathbf{e}_j}, \qquad \forall (i,j) \in E,$$

$$M_{d-1}(h_i^+ \star y)(\alpha,\beta) = y_{\alpha+\beta+2\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_i} = 0, \qquad \forall i \in V.$$

Then for every $\alpha \in \mathbb{N}^{|V|}_{d-2}$ and $k \in V$,

$$\begin{aligned} M_{d-1}(g_{ij}^- \star y)(\alpha,\alpha) &= -y_{2\alpha+\mathbf{e}_i+\mathbf{e}_j}, \\ &= -y_{2\alpha+2\mathbf{e}_i+\mathbf{e}_j}, \\ &= -y_{2\alpha+2\mathbf{e}_i+2\mathbf{e}_j}, \\ &= -M_d(y)(\alpha+\mathbf{e}_i+\mathbf{e}_j, \alpha+\mathbf{e}_i+\mathbf{e}_j). \end{aligned}$$

and

$$\begin{aligned} \det\left[M_{d-1}(g_{ij}^- \star y)_{\{\alpha,\alpha+\mathbf{e}_k\},\{\alpha,\alpha+\mathbf{e}_k\}}\right] &= \begin{vmatrix} -y_{2\alpha+\mathbf{e}_i+\mathbf{e}_j} & -y_{2\alpha+\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} \\ -y_{2\alpha+\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} & -y_{2\alpha+2\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} \end{vmatrix} \\ &= \begin{vmatrix} -y_{2\alpha+\mathbf{e}_i+\mathbf{e}_j} & -y_{2\alpha+\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} \\ -y_{2\alpha+\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} & -y_{2\alpha+\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} \end{vmatrix} \\ &= y_{2\alpha+\mathbf{e}_i+\mathbf{e}_j} y_{2\alpha+\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} - y^2_{2\alpha+\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j}. \end{aligned}$$

Since $M_{d-1}(g_{ij}^- \star y)$ and $M_d(y)$ are semi-definite positive matrices then

$$\begin{cases} M_{d-1}(g_{ij}^- \star y)(\alpha,\alpha) \geq 0, \\ M_d(y)(\alpha+\mathbf{e}_i+\mathbf{e}_j, \alpha+\mathbf{e}_i+\mathbf{e}_j) \geq 0, \\ \det\left[M_{d-1}(g_{ij}^- \star y)_{\{\alpha,\alpha+\mathbf{e}_k\},\{\alpha,\alpha+\mathbf{e}_k\}}\right] \geq 0. \end{cases}$$

Which implies that

$$\begin{cases} y_{2\alpha+\mathbf{e}_i+\mathbf{e}_j} = 0, \\ -y^2_{2\alpha+\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} \geq 0. \end{cases} \quad \text{and so} \quad \begin{cases} y_{2\alpha+\mathbf{e}_i+\mathbf{e}_j} = 0, \\ y^2_{2\alpha+2\mathbf{e}_k+\mathbf{e}_i+\mathbf{e}_j} = 0. \end{cases}$$

This proves that $M_{d-1}(g_{ij}^- \star y)(\alpha,\alpha) = 0$ for every $\alpha \in \mathbb{N}^{|V|}_{d-1}$. Therefore $M_{d-1}\left(g_{ij}^- \star y\right)$ is a semi-definite positive matrix with zero on the diagonal. It is the zero matrix, and this proves that y is also a feasible solution of the level d of the Lasserre hierarchy for formulation (4.3). □

4.4.4 Relationship between $\rho_d^{(2)}$ and $\rho_d^{(3)}$

The next result is that the feasible set of the Lasserre relaxation of order d using the formulation (4.4) is a subset of the feasible set of the relaxation of the same order for formulation (4.5). Hence, the bound $\rho_d^{(3)}$ is always dominated by the bound $\rho_d^{(2)}$.

Proposition 4.3. *For every graph G and order $d \geq 1$, $\rho_d^{(2)} \leq \rho_d^{(3)}$.*

Proof. Let $y = \{y_\alpha\}$ be a feasible solution of the relaxation of (4.4) of order d. We know that for every $(\alpha, \beta) \in \mathbb{N}^{|V|}_{d-1} \times \mathbb{N}^{|V|}_{d-1}$:

$$\begin{aligned}
M_{d-1}(g_{ij}^- \star y)(\alpha,\beta) &= -y_{\alpha+\beta+\mathbf{e}_i+\mathbf{e}_j} && \forall (i,j) \in E,\\
M_{d-1}(h_i^+ \star y)(\alpha,\beta) &= y_{\alpha+\beta+2\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_i} && \forall i \in V,\\
M_{d-1}(l_{ij} \star y)(\alpha,\beta) &= y_{\alpha+\beta} - y_{\alpha+\beta+\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_j} && \forall (i,j) \in E.
\end{aligned}$$

Let $A \in \mathbb{R}^{\mathbb{N}^{|V|}_{d-1} \times \mathbb{N}^{|V|}_{d}}$ such that:

$$A(\alpha,\gamma) = \begin{cases} 1 & \text{if } \gamma = \alpha, \\ -1 & \text{if } \gamma = \alpha + \mathbf{e}_i, \\ -1 & \text{if } \gamma = \alpha + \mathbf{e}_j, \\ 0 & \text{otherwise.} \end{cases}$$

For every $(\alpha, \beta) \in \mathbb{N}^{|V|}_{d-1} \times \mathbb{N}^{|V|}_{d-1}$:

$$\begin{aligned}
\left[AM_d(y)A^\intercal\right](\alpha,\beta) &= \sum_{\gamma \in \mathbb{N}^{|V|}_d} \sum_{\delta \in \mathbb{N}^{|V|}_d} A(\alpha,\gamma) M_d(y)(\gamma,\delta) A^\intercal(\delta,\beta) \\
&= \sum_{\gamma \in \mathbb{N}^{|V|}_d} \sum_{\delta \in \mathbb{N}^{|V|}_d} A(\alpha,\gamma) M_d(y)(\gamma,\delta) A(\beta,\delta) \\
&= \sum_{\gamma \in \mathbb{N}^{|V|}_d} \sum_{\delta \in \mathbb{N}^{|V|}_d} A(\alpha,\gamma) y_{\gamma+\delta} A(\beta,\delta) \\
&= \; y_{\alpha+\beta} - y_{\alpha+\beta+\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_j} \\
&\quad - y_{\alpha+\mathbf{e}_i+\beta} + y_{\alpha+\mathbf{e}_i+\beta+\mathbf{e}_j} + y_{\alpha+\mathbf{e}_i+\beta+\mathbf{e}_j} \\
&\quad - y_{\alpha+\mathbf{e}_j+\beta} + y_{\alpha+\mathbf{e}_j+\beta+\mathbf{e}_i} + y_{\alpha+\mathbf{e}_j+\beta+\mathbf{e}_j}
\end{aligned}$$

$$= \underbrace{y_{\alpha+\beta} - y_{\alpha+\beta+\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_j}}_{=M_{d-1}(l_{ij}\star y)(\alpha,\beta)} + \underbrace{y_{\alpha+\beta+2\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_i}}_{=M_{d-1}(h_i^+\star y)(\alpha,\beta)}$$
$$+ \underbrace{y_{\alpha+\beta+2\mathbf{e}_j} - y_{\alpha+\beta+\mathbf{e}_j}}_{=M_{d-1}(h_j^+\star y)(\alpha,\beta)} + 2 \underbrace{y_{\alpha+\beta+\mathbf{e}_i+\mathbf{e}_j}}_{=-M_{d-1}(g_{ij}^-\star y)(\alpha,\beta)}$$

which implies that

$$M_{d-1}(l_{ij} \star y) = AM_d(y)A^\intercal + 2M_{d-1}(g_{ij}^- \star y) - M_{d-1}(h_j^+ \star y) - M_{d-1}(h_i^+ \star y).$$

Since $M_d(y) \succeq 0$ then $AM_d(y)A^\intercal \succeq 0$. Moreover

$$\begin{cases} M_{d-1}(g_{ij}^- \star y) \succeq 0, \\ M_{d-1}(h_j^+ \star y) = M_{d-1}(h_i^+ \star y) = 0. \end{cases}$$

Then $M_{d-1}(l_{ij} \star y) \succeq 0$ for all $(i,j) \in E$, and thus y is a feasible solution of the d-order Lasserre relaxation for formulation (4.5).

□

4.4.5 Relationship between $\rho_d^{(1)}$ and $\rho_d^{(4)}$

The next result is that the feasible set of the Lasserre relaxation of order d using (4.3) is a subset of the feasible set of the relaxation of the same order for formulation (4.6). Hence, the bound $\rho_d^{(4)}$ is always dominated by the bound $\rho_d^{(1)}$.

Proposition 4.4. *For every graph G and order $d \geq 1$, $\rho_d^{(1)} \leq \rho_d^{(4)}$.*

Proof. Let $y = \{y_\alpha\}$ be a feasible solution of the relaxation of (4.3) of order d. We know that for every $(\alpha, \beta) \in \mathbb{N}_{d-1}^{|V|} \times \mathbb{N}_{d-1}^{|V|}$:

$$\begin{aligned}
M_{d-1}(g_{ij}^- \star y)(\alpha,\beta) &= -y_{\alpha+\beta+\mathbf{e}_i+\mathbf{e}_j} && \forall (i,j) \in E, \\
M_{d-1}(h_i^+ \star y)(\alpha,\beta) &= y_{\alpha+\beta+2\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_i} && \forall i \in V, \\
M_{d-1}(q_i^+ \star y)(\alpha,\beta) &= y_{\alpha+\beta+\mathbf{e}_i} && \forall i \in V, \\
M_{d-1}(q_i^- \star y)(\alpha,\beta) &= y_{\alpha+\beta} - y_{\alpha+\beta+\mathbf{e}_i} && \forall i \in V,
\end{aligned}$$

Let $A \in \mathbb{R}^{\mathbb{N}^{|V|}_{d-1} \times \mathbb{N}^{|V|}_{d}}$ such that:

$$A(\alpha, \gamma) = \begin{cases} 1 & \text{if } \gamma = \alpha + \mathbf{e}_i, \\ 0 & \text{otherwise.} \end{cases}$$

For every $(\alpha, \beta) \in \mathbb{N}^{|V|}_{d-1} \times \mathbb{N}^{|V|}_{d-1}$:

$$\begin{aligned}
\left[AM_d(y)A^\intercal\right](\alpha, \beta) &= \sum_{\gamma \in \mathbb{N}^{|V|}_{d}} \sum_{\delta \in \mathbb{N}^{|V|}_{d}} A(\alpha, \gamma) M_d(y)(\gamma, \delta) A^\intercal(\delta, \beta) \\
&= \sum_{\gamma \in \mathbb{N}^{|V|}_{d}} \sum_{\delta \in \mathbb{N}^{|V|}_{d}} A(\alpha, \gamma) M_d(y)(\gamma, \delta) A(\beta, \delta) \\
&= \sum_{\gamma \in \mathbb{N}^{|V|}_{d}} \sum_{\delta \in \mathbb{N}^{|V|}_{d}} A(\alpha, \gamma) y_{\gamma+\delta} A(\beta, \delta) \\
&= y_{\alpha+\mathbf{e}_i+\beta+\mathbf{e}_i} \\
&= y_{\alpha+\beta+2\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_i} + y_{\alpha+\beta+\mathbf{e}_i} \\
&= M_{d-1}(h_i^+ \star y)(\alpha, \beta) + M_{d-1}(q_i^+ \star y)(\alpha, \beta)
\end{aligned}$$

which implies that

$$M_{d-1}(q_i^+ \star y) = AM_d(y)A^\intercal - M_{d-1}(h_i^+ \star y).$$

Since $M_d(y) \succeq 0$ then $AM_d(y)A^\intercal \succeq 0$. Moreover $M_{d-1}(h_i^+ \star y) = 0$. Then $M_{d-1}(q_i^+ \star y) \succeq 0$ for all $i \in V$. On the other hand, let $B \in \mathbb{R}^{\mathbb{N}^{|V|}_{d-1} \times \mathbb{N}^{|V|}_{d}}$ such that:

$$B(\alpha, \gamma) = \begin{cases} 1 & \text{if } \gamma = \alpha, \\ -1 & \text{if } \gamma = \alpha + \mathbf{e}_i, \\ 0 & \text{otherwise.} \end{cases}$$

For every $(\alpha, \beta) \in \mathbb{N}^{|V|}_{d-1} \times \mathbb{N}^{|V|}_{d-1}$:

$$\begin{aligned}
\left[BM_d(y)B^\intercal\right](\alpha, \beta) &= \sum_{\gamma \in \mathbb{N}^{|V|}_{d}} \sum_{\delta \in \mathbb{N}^{|V|}_{d}} B(\alpha, \gamma) M_d(y)(\gamma, \delta) B^\intercal(\delta, \beta) \\
&= \sum_{\gamma \in \mathbb{N}^{|V|}_{d}} \sum_{\delta \in \mathbb{N}^{|V|}_{d}} B(\alpha, \gamma) M_d(y)(\gamma, \delta) B(\beta, \delta) \\
&= \sum_{\gamma \in \mathbb{N}^{|V|}_{d}} \sum_{\delta \in \mathbb{N}^{|V|}_{d}} B(\alpha, \gamma) y_{\gamma+\delta} B(\beta, \delta) \\
&= y_{\alpha+\beta} - y_{\alpha+\beta+\mathbf{e}_i} - y_{\alpha+\mathbf{e}_i+\beta} + y_{\alpha+\mathbf{e}_i+\beta+\mathbf{e}_i}
\end{aligned}$$

$$= y_{\alpha+\beta} - y_{\alpha+\beta+\mathbf{e}_i} - y_{\alpha+\beta+\mathbf{e}_i} + y_{\alpha+\beta+2\mathbf{e}_i}$$
$$= M_{d-1}(q_i^- \star y)(\alpha,\beta) + M_{d-1}(h_i^+ \star y)(\alpha,\beta).$$

which implies that

$$M_{d-1}(q_i^- \star y) = BM_d(y)B^\intercal - M_{d-1}(h_i^+ \star y).$$

Since $M_d(y) \succeq 0$ then $BM_d(y)B^\intercal \succeq 0$. Moreover $M_{d-1}(h_i^+ \star y) = 0$. Then $M_{d-1}(q_i^- \star y) \succeq 0$ for all $i \in V$. Since $M_{d-1}(g_{ij}^- \star y) = 0$, y is a feasible solution of the d-order Lasserre relaxation for formulation (4.6). □

4.4.6 Relationships with the linear programming relaxations of maximum independent set

In this section, we establish the relationship among some of the formulations discussed above and two famous linear programming (LP) relaxations for the maximum independent set problem. More precisely, we consider two LP formulations and we refer to them as "weak" and "strong". The weak formulation is that with constraints

$$x_i + x_j \leq 1 \quad \forall (i,j) \in E \tag{4.7}$$

and nonnegativity, whereas the strong formulation replaces constraints (4.7) with constraints

$$\sum_{i \in C} x_i \leq 1 \quad \forall C \in \mathcal{C}, \tag{4.8}$$

where $\mathcal{C}$ is the set of all maximum cliques in G.

Proposition 4.5. *For every graph $G = (V, E)$, the optimal value of the order 1 of the Lasserre hierarchy for formulation (4.5) is equal to the value of the LP relaxation of the weak formulation of the independent set problem.*

Proof. Let $y = \{y_\alpha\}_\alpha$ be a feasible solution of the order 1 of the Lasserre hierarchy of formulation (4.5), then

$$\begin{cases} y_0 = 1, & \\ y_{\mathbf{e}_i} = y_{2\mathbf{e}_i}, & \forall i \in V, \\ y_{\mathbf{e}_i} + y_{\mathbf{e}_j} \leq 1, & \forall (i,j) \in E. \end{cases} \implies \begin{cases} y_0 = 1, & \\ 0 \leq y_{\mathbf{e}_i} \leq 1, & \forall i \in V, \\ y_{\mathbf{e}_i} + y_{\mathbf{e}_j} \leq 1, & \forall (i,j) \in E, \end{cases}$$

and $(y_{\mathbf{e}_i})_{i \in V}$ is a feasible solution of the LP relaxation of the weak formulation of the

independent set problem. Conversely, let $(x_i)_{i\in V}$ be a feasible solution of the LP relaxation of the weak formulation of the independent set problem, let $X \in \mathbb{R}^{|V|+1}$ be defined as

$$\begin{cases} X_0 = 1, \\ X_i = x_i, \quad \forall i \in V, \end{cases}$$

and let $A \in \mathbb{R}^{(|V|+1)\times(|V|+1)}$ be defined as

$$\begin{cases} A \text{ is diagonal}, \\ A_{0,0} = 0, \\ A_{i,i} = x_i(1-x_i) \quad \forall i \in V. \end{cases}$$

Let $y = \{y_\alpha\}_{\alpha\in\mathbb{N}_2^{|V|}}$ be defined as

$$\begin{cases} y_0 = 1, \\ y_{\mathbf{e}_i} = y_{2\mathbf{e}_i} = x_i, \quad \forall i \in V, \\ y_{\mathbf{e}_i+\mathbf{e}_j} = x_i \cdot x_j, \quad \forall (i,j) \in V^2 \quad i \neq j. \end{cases}$$

Then,

$$\begin{cases} L_y(1) = y_0 = 1, \\ M_0(l_{ij} \star y) = 1 - y_{\mathbf{e}_i} - y_{\mathbf{e}_j} = 1 - x_i - x_j \geq 0, & \forall (i,j) \in E, \\ M_0(h_i^+ \star y) = y_{2\mathbf{e}_i} - y_{\mathbf{e}_i} = 0, & \forall i \in V, \\ M_1(y) = XX^\intercal + A \succeq 0. \end{cases}$$

This proves that y is a feasible solution of the level 1 of the Lasserre hierarchy of formulation (4.5) with the value equal to $\sum_{i\in V} y_{\mathbf{e}_i} = \sum_{i\in V} x_i$. □

In [76, Theorem 10.4] the authors proved that $\rho_1^{(1)}$ is smaller than the value of the LP relaxation of the strong formulation of the independent set problem. In the following proposition, we extend this result to prove a stronger relationship, namely that between the LP relaxation of the strong formulation and the order 1 of the Lasserre hierarchy for formulation (4.4). Such a result, with a different proof, was given independently by Szegedy [77].

Proposition 4.6 ([77]). *For every graph $G = (V, E)$, the optimal value of the order 1 of the Lasserre hierarchy for formulation (4.4) is smaller than the value of the LP relaxation*

of the strong formulation of the independent set problem.

Proof. Let $y = \{y_\alpha\}_\alpha$ be a feasible solution of the level one of the Lasserre hierarchy of formulation (4.4), then

$$\begin{cases} y_0 = 1, \\ y_{\mathbf{e}_i} = y_{2\mathbf{e}_i}, & \forall i \in V, \\ y_{\mathbf{e}_i+\mathbf{e}_j} \leq 0, & \forall (i,j) \in E. \end{cases}$$

Let W be a clique of G and $X \in \mathbb{R}^{|V|+1}$ such that:

$$\begin{cases} X_0 = 1, \\ X_i = -1, & \text{if } i \in W, \\ X_i = 0, & \text{otherwise.} \end{cases}$$

Then,

$$\begin{aligned} 0 \leq X^\intercal M_1(y) X &= y_0 - 2\sum_{i\in W} y_{\mathbf{e}_i} + \sum_{i\in W} y_{2\mathbf{e}_i} + \sum_{\substack{(i,j)\in W^2, \\ i\neq j}} y_{\mathbf{e}_i+\mathbf{e}_j} \\ &\leq y_0 - 2\sum_{i\in W} y_{\mathbf{e}_i} + \sum_{i\in W} y_{\mathbf{e}_i} \leq 1 - \sum_{i\in W} y_{\mathbf{e}_i}. \end{aligned}$$

This proves that $(y_{\mathbf{e}_i})_{i\in V}$ is a feasible solution of the LP relaxation of the strong formulation of the independent set problem. □

4.5 Summary of Results and Future Research

Using the notation previously defined, we summarize our results as follows:

For $d = 1$:

$$\rho_1^{(1)} \leq \rho_1^{(2)} \leq \mathrm{LP}_{\text{strong}} \leq \mathrm{LP}_{\text{weak}} = \rho_1^{(3)} \leq |V| = \rho_1^{(4)},$$

and for $d \geq 2$:

$$\rho_d^{(1)} = \rho_d^{(2)} \leq \rho_d^{(3)}, \text{ and } \rho_d^{(1)} = \rho_d^{(2)} \leq \rho_d^{(4)}.$$

We believe these results give an interesting, initial perspective on evaluating the quality of a formulation not only in terms of its relaxation but also with respect to the Lasserre

relaxations originated by it.

In future research, it would be interesting to further study this question for other combinatorial problems and for the other hierarchies in Section 4.1.

Acknowledgements

We are indebted to the Area Editor and an anonymous referee for a careful reading and useful suggestions to improve the manuscript.

CHAPITRE 5 ARTICLE 2 : FREE LUNCH IN THE FOREST : FUNCTIONALLY IDENTICAL PRUNING OF TREE ENSEMBLES

Authors: Youssouf Emine, Alexandre Forel, Idriss Malek and Thibaut Vidal

Accepted: AAAI Conference on Artificial Intelligence, 2025

Date de parution: 3 mars 2025

Tree ensembles, including boosting methods, are highly effective and widely used for tabular data. However, large ensembles lack interpretability and require longer inference times. We introduce a method to prune a tree ensemble into a reduced version that is "functionally identical" to the original model. In other words, our method guarantees that the prediction function stays unchanged for any possible input. As a consequence, this pruning algorithm is lossless for any aggregated metric. We formalize the problem of functionally identical pruning on ensembles, introduce an exact optimization model, and provide a fast yet highly effective method to prune large ensembles. Our algorithm iteratively prunes considering a finite set of points, which is incrementally augmented using an adversarial model. In multiple computational experiments, we show that our approach is a "free lunch", significantly reducing the ensemble size without altering the model's behavior. Thus, we can preserve state-of-the-art performance at a fraction of the original model's size.

5.1 Introduction

Ensembles remain one of the most effective and commonly utilized machine learning models. Among them, tree ensembles such as random forests and boosting are known to achieve state-of-the-art performance on tabular data [16,17]. They also offer greater interpretability compared to large neural networks. Theory and practice indicate that superior performance is achieved with large ensembles, i.e., forests with many trees [18–20]. However, large ensembles lead to large memory requirements and inference times, which can quickly become a bottleneck when embedding models into hardware such as microcontrollers or smartphones. Further, large ensemble models lack interpretability due to the complex interaction of their components.

Ensemble pruning denotes the process of reducing the size of a trained ensemble model. This can be understood as removing nodes from the trees or removing trees entirely from the forest. Early approaches studied how to prune random forests made of shallow trees, i.e., trees that were themselves pruned beforehand [21, 22]. These works showed that pruning

could improve the test error and out-of-sample generalization. However, this is not true with boosted ensembles, since the trees are typically already shallow. Hence, most recent studies seek a trade-off between the amount of pruning and the impact on test error [23,24]. This trade-off may be profitable in some situations but there is no guarantee that the model will perform well on new data.

In this paper, we completely avoid this trade-off by pruning ensemble models *without any change in their behavior*. We call this a "functionally-identical" pruning since the prediction functions of the pruned ensemble and the original one are identical. This has two main advantages. First, as demonstrated in this paper, the resulting optimization problems can be very efficiently solved. Second, such pruned models give a "free lunch": they achieve the exact same performance as the original model at a fraction of its size. This provides multiple advantages regarding memory, inference time, and interpretability. An example ensemble that can be pruned without any change in its prediction function is shown in Figure 5.1.

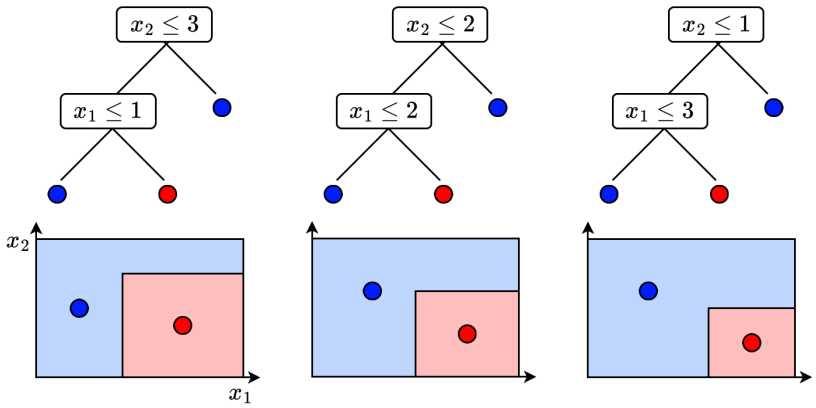


Figure 5.1 A small ensemble made of three trees with equal weights. This ensemble can be pruned without any change in its prediction function by removing the first and third trees.

This paper introduces the algorithm `FIPE`, standing for **F**unctionally **I**dentical **P**runing of **E**nsembles. `FIPE` iterates between a *pruning model* and a *separation oracle*. The pruning model identifies candidate pruned ensembles on a finite set of points. The separation oracle checks whether the pruned model is functionally identical to the original model on the entire feature space. If this is not the case, new points are added to the pruning set. Our iterative algorithm is such that the final pruned ensemble is certifiably equivalent to the original one on the entire feature space.

We make the following contributions:

(i) We formalize the problem of functionally identical pruning of tree ensembles and char-

acterize its computational complexity.

(ii) We present `FIPE`, our iterative algorithm and its two main components: the pruner, and the oracle. We prove that the algorithm terminates in a finite number of steps.

(iii) We investigate two variants of the pruning model: an exact combinatorial optimization model based on the $\|\cdot\|_0$ norm that guarantees that the pruned model is of minimal size, and a fast approximate algorithm based on the $\|\cdot\|_1$ norm that is not necessary of minimal size but scales to large datasets. We further provide an efficient pruning oracle inspired by recent works on counterfactual explanations.

(iv) We apply `FIPE` on boosted ensembles `AdaBoost`, `LightGBM` and `XGBoost` as well as random forests. Our extensive experiments demonstrate that significantly smaller ensembles can be identified across various real-world datasets. A central finding of this study is identifying that many base learners of boosted ensembles are superfluous. Consequently, an integrated pruning and reweighting approach can substantially reduce their size while maintaining their predictive performance.

5.2 Problem Statement

Let $\{(x_i, y_i)\}_{i=1}^n$ be a labeled set of observations, where $x \in \mathcal{X} \subseteq \mathbb{R}^p$ is a feature vector. We focus on classification problems with C classes. Denote by $[\![a, b]\!]$ the set of integers between a and b. Any class y belongs to $[\![1, C]\!]$.

5.2.1 Classification ensembles

A classification ensemble is a weighted set of classifiers $\{(h_m, \alpha_m)\}_{m=1}^M$ where each classifier $h_m : \mathcal{X} \to [0, 1]^C$ provides a score vector for any input x, and α_m is the weight of classifier m. The ensemble prediction function is a map $H : \mathcal{X} \times \mathbb{R}_{\geq 0}^M \to [\![1, C]\!]$ that assigns a class to any input x using a voting scheme, expressed generally as:

$$H(x; \alpha) = \underset{c \in [\![1,C]\!]}{\operatorname{argmax}} \sum\nolimits_{m=1}^{M} \alpha_m h_m^{(c)}(x), \tag{5.1}$$

where $h_m^{(c)}(x)$ is the score predicted by classifier of index m for class c. Any deterministic tie-breaking rule can be used to break the ties in Equation (5.1) if the argmax is set-valued.

The computation of the score of a tree depends on the nature of the ensemble. For instance, AdaBoost [78] and the original random forest algorithm of [41] use the so-called hard-voting criterion in which each classifier prediction $h_m^{(c)}(x) \in \{0, 1\}$ is binary.

5.2.2 Functionally-identical pruning

We study how to prune classification ensembles while guaranteeing that the pruned model is functionally identical to the original model. This is formalized in the following definition.

Definition 5.1. A pruned ensemble with weights w is *functionally identical* to the original ensemble on the entire feature space $\mathcal{X}$ if $H(x; w) = H(x; \alpha)$ for all $x \in \mathcal{X}$.

The pruned ensemble with minimum size is obtained by solving:

$$\min_{w \geq 0} \|w\|_0 \ : \ H(x; w) = H(x; \alpha), \ \forall x \in \mathcal{X}. \tag{5.2}$$

Since any learner with a zero weight is effectively pruned, Problem (5.2) minimizes the number of active learners. The minimizer of Problem (5.2) is not only certifiably functionally identical, but it is also of minimal size. That is, it is guaranteed to be the smallest reweighted ensemble that is functionally identical to the original one.

The constraint in Problem (5.2) ensures that the pruned model is functionally identical to the original model on the entire space. Functionally-identical pruning is also known as "faithful" pruning in [58]. Hence, we use equivalently both words in the rest of the paper.

Proposition 5.1. *For additive tree ensembles, i.e., a broad class including random forests and boosting methods, Problem* (5.2) *is NP-hard.*

The proof is given in Section B.1. Theorem 5.1 states that optimal faithful pruning is a difficult task in general. Nevertheless, this paper shows that Problem (5.2) can be solved efficiently for large ensembles on real-world datasets, and provides effective heuristics otherwise. Our algorithms are presented in the next section.

5.3 Pruning Algorithms

To solve the faithful pruning problem, `FIPE` iterates between solving a *pruning problem* on a finite set of points and a *separation oracle*. This is illustrated in Figure 5.2.

`FIPE` is thus made of two essential components: (i) a faithful pruner `Pruner` that returns the set of weights $\{w_m\}_{m=1}^M$ of the pruned ensemble faithful to the original ensemble on a given finite set of points $\mathcal{D} \subset \mathcal{X}$, and (ii) a separation oracle `Oracle` that returns a set of point $\mathcal{S} \subset \mathcal{X}$ for which the pruned ensemble with weights w and the original ensemble with weights α differ. If the oracle cannot find a separating point, the two ensembles are

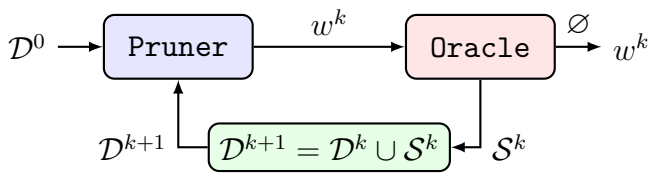


Figure 5.2 `FIPE` iterates between the pruning model and the separation oracle until it returns the set of weights of the pruned model.

Algorithm 5.1: Faithful pruning algorithm

Input: An arbitrary finite set of points of $\mathcal{X}$
Output: w^k

$\mathcal{D}^0 \leftarrow$ An arbitrary finite set of points of $\mathcal{X}$;
$w^0 \leftarrow$ `Pruner`$(\mathcal{D}^0)$;
$k \leftarrow 0$;
while `Oracle`$(w^k) \neq \varnothing$ **do**
  $\mathcal{S}^k \leftarrow$ `Oracle`(w^k);
  $\mathcal{D}^{k+1} \leftarrow \mathcal{D}^k \cup \mathcal{S}^k$;
  $w^{k+1} \leftarrow$ `Pruner`$(\mathcal{D}^{k+1})$;
  $k \leftarrow k + 1$;
end
return w^k;

functionally identical over the entire feature space. The algorithm then returns the set of weights of the pruned ensemble. This algorithm is presented in Algorithm 5.1.

Theorem 5.2. *For tree ensembles,* `FIPE` *terminates after a finite number of calls to the* `Oracle`.

The proof is given in Section B.1.

5.3.1 Pruning on a finite set of points

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ a finite set of points $\mathcal{D} \subset \mathcal{X}$. The pruning model returns the set of weights of the pruned ensemble faithful to the original ensemble on $\mathcal{D}$.

Denote by $c_i = H(x_i, \alpha)$ the class of the point x_i given by the original ensemble with weights α. Note that the classes c_i are not necessarily equal to the original classes y_i depending on the training of the ensemble H. Pruning on a finite set of points is a special case of Problem (5.2) when $\mathcal{X}$ is finite. The constraint of Problem (5.2) can be reformulated to ensure that the pruned ensemble is faithful to the original ensemble on $\mathcal{D}$. It can be written

as:

$$\sum_{m=1}^{M} w_m \left(h_m^{(c_i)}(x_i) - h_m^{(c)}(x_i)\right) > 0. \tag{5.3}$$

It ensures that the pruned ensemble gives a higher weighted score for the class c_i than for the class c for each point x_i and each class $c \neq c_i$.

Equation (5.3) is difficult to handle numerically since it involves a strict inequality. However, using the same idea as in a support vector machine, we express it equivalently as:

$$\sum_{m=1}^{M} w_m \left(h_m^{(c_i)}(x_i) - h_m^{(c)}(x_i)\right) \geq 1. \tag{5.4}$$

Minimal-size faithful pruner. The faithful pruning problem on a finite set of points is a combinatorial optimization problem given by:

$$\min_{w \geq 0,\, u \in \{0,1\}^M} \sum_{m=1}^{M} u_m \tag{5.5a}$$

$$w_m \leq W u_m, \ \forall m \in [\![1, M]\!], \tag{5.5b}$$

$$\sum_{m=1}^{M} w_m \left(h_m^{(c_i)}(x_i) - h_m^{(c)}(x_i)\right) \geq 1, \tag{5.5c}$$

$$\forall i \in [\![1, N]\!], \ \forall c \in [\![1, C]\!], \ c \neq c_i.$$

This problem minimizes the $\|\cdot\|_0$ norm of the weights. It uses the binary variables u_m to indicate whether the estimator m is active or not, and a parameter W, which is an upper bound on the weights of the active estimators. A minimizer of Problem (5.5) is both faithful to the original ensemble on $\mathcal{D}$ and of certifiably minimal size according to the $\|\cdot\|_0$ norm.

Efficient approximation. While Problem (5.5) ensures finding the sparsest model, it might be expensive to solve numerically since its objective function with $\|\cdot\|_0$ is hard to linearize. Hence, we propose an efficient approximation by replacing the $\|\cdot\|_0$ norm with the $\|\cdot\|_1$ norm. This problem is a linear programming problem given by:

$$\min_{w \geq 0} \|w\|_1 = \sum_{m=1}^{M} w_m \tag{5.6a}$$

$$\sum_{m=1}^{M} w_m \left(h_m^{(c_i)}(x_i) - h_m^{(c)}(x_i)\right) \geq 1, \tag{5.6b}$$

$$\forall i \in [\![1, N]\!], \ \forall c \in [\![1, C]\!], \ c \neq c_i.$$

Problem (5.6) can be solved very efficiently using standard linear programming solvers. The faithfulness constraint of the problem is unchanged so that any solution of Problem (5.6) is guaranteed to remain faithful to the original ensemble on $\mathcal{D}$.

5.3.2 Separation oracle for tree ensembles

This section presents the separation oracle used in `FIPE`. The goal of the oracle is twofold: (i) it certifies that a pruned model is functionally equivalent to the original model on the entire feature space, or (ii) it provides a set of points for which the two models differ. These two goals can be achieved jointly by formulating an optimization model that maximizes the misclassification between the pruned and original model over the feature space.

Let c and y be two fixed classes. The separation problem over these two classes can be formulated as:

$$\max_{x \in \mathcal{X}} \sum_{m=1}^{M} w_m \left(h_m^{(c)}(x) - h_m^{(y)}(x) \right) \; : \; y = H(x, \alpha) \,. \tag{5.7}$$

This problem maximizes the difference between the score of class c and class y according to the pruned ensemble. The constraint of Problem (5.7) ensures that the original ensemble predicts class y for the point x. If the value of Problem (5.7) is positive, the pruned ensemble predicts class c whereas the original ensemble predicts y. Hence, a separating point is such that it has a positive value according to Problem (5.7).

Remark 5.2. Since Problem (5.7) maximizes the difference in prediction between two ensembles, it can be interpreted as a variation of the adversarial example or counterfactual explanation problem.

If the objective function is non-positive across all points in the feature space, both models are certified to be equivalent on the entire feature space. In practice, we solve the separation problem for each pair of classes c and y. Depending on the algorithm used to solve Problem (5.7), several points may be found that have a positive score difference. In `FIPE`, we add all points with a positive score difference $\mathcal{S}$. When $\mathcal{S}$ is empty, the algorithm terminates.

The complexity of Problem (5.7) lies in representing the classification function of the trees with linear variables and constraints. To ensure that the original ensemble predicts class y for point x, we can replace the constraint of Problem (5.7) by the following inequality:

$$\sum\nolimits_{m=1}^{M} \alpha_m \left(h_m^{(y)}(x) - h_m^{(y')}(x) \right) > 0, \; \forall y' \neq y. \tag{5.8}$$

These constraints ensure that the original ensemble gives a higher score for class y than for any other class $y' \neq y$, making y the predicted class for point x by the original ensemble. Again, this constraint is ill defined because of the strict inequality. By introducing a

sufficiently small $\varepsilon > 0$, the separation problem can be equivalently formulated as:

$$\max_{x \in \mathcal{X}} \sum_{m=1}^{M} w_m \left(h_m^{(c)}(x) - h_m^{(y)}(x) \right) \tag{5.9a}$$

$$\sum_{m=1}^{M} \alpha_m \left(h_m^{(y)}(x) - h_m^{(y')}(x) \right) \geq \varepsilon, \forall y' \neq y. \tag{5.9b}$$

Separation for tree ensembles. Problem (5.9) is nonlinear in general. However, for additive tree ensembles, we can linearize it efficiently using a procedure similar to the one used in [79]. This consists in introducing a set of variables and constraints to track the path of the separating point x in all trees of the ensembles.

For each tree m, let $\mathcal{V}_m$ be the set of its internal nodes, $\mathcal{L}_m$ the set of its leaves, and d_m its maximum depth. For each node $v \in \mathcal{V}_m$, let $\textbf{left}(v)$ and $\textbf{right}(v)$ be its left and right children respectively, and let $\textbf{depth}(v)$ be its depth in the tree m. Let $\textbf{root}(m)$ denote the root of tree m. Let $\mathrm{f}_{m,v}$ be a binary variable indicating if node $v \in \mathcal{V}_m \cup \mathcal{L}_m$ is in the path of the point x in the tree m.

To linearize the tree prediction functions, we introduce a binary variable $\lambda_{m,d}$ to indicate whether point x goes to the left child at depth d of the tree m. The path consistency constraints in the tree m can then be expressed as:

$$\mathrm{f}_{m,\textbf{root}(m)} = 1, \tag{5.10a}$$

$$\mathrm{f}_{m,\textbf{left}(v)} + \mathrm{f}_{m,\textbf{right}(v)} = \mathrm{f}_{m,v}, \ \forall v \in \mathcal{V}_m, \tag{5.10b}$$

$$\sum_{v \in \mathcal{V}_m : \textbf{depth}(v) = d} \mathrm{f}_{m,\textbf{left}(v)} = \lambda_{m,d}, \forall d \in [\![0, d_m]\!]. \tag{5.10c}$$

Constraint (5.10a) ensures that the root of tree m is in the path of point x. Constraint (5.10b) ensures that one of the children of node v is in the path of point x if node v is in the path. It also ensures that the path of point x in tree m at depth d goes to the left child if $\lambda_{m,d} = 1$. Using these variables, we can formulate Problem (5.9) equivalently as:

$$\max_{\mathrm{f} \geq 0} \sum_{m=1}^{M} \sum_{v \in \mathcal{L}_m} w_m \left(h_m^{(c)}(x) - h_m^{(y)}(x) \right) \mathrm{f}_{m,v} \tag{5.11}$$

$$\sum_{m=1}^{M} \sum_{v \in \mathcal{L}_m} \alpha_m \left(h_m^{(y)}(x) - h_m^{(y')}(x) \right) \mathrm{f}_{m,v} \geq \varepsilon, \forall y' \neq y.$$

Remark 5.3. Problem (5.11) is not optimizing over the variable x anymore, but only over the variables that track the path over the trees. Indeed, for tree ensembles, the prediction

function is piecewise-constant. Hence, it is sufficient to identify the set of leaves to separate the pruned and original ensembles. A separating point is directly determined by taking the center of the cell defined by the leaf variables.

Further, observe that the binary variables $\mathrm{f}_{m,v}$ can be defined as continuous in $[0,1]$, as they will be forced to binary values due to Constraints (5.10a)–(5.10c).

Feature consistency. To ensure the value of x is consistent across the trees, we need to add feature consistency variables and constraints. Each feature type has to be handled separately. We explain below how to ensure the consistency of continuous features, and in Section B.1 how to handle categorical and binary features.

Let j be a continuous feature and let $\left\{t_{j,1}, t_{j,2}, \ldots, t_{j,R_j}\right\}$ be the set of thresholds that the nodes of the original ensemble are splitting on. We consider without loss of generality that this set is sorted. For each continuous feature j and $r \in [\![1, R_j]\!]$, we define the continuous auxiliary variables $\mu_{j,r} \in [0,1]$. These variables indicate whether the new point is on the left or right-side of a level. They are constrained such that $\mu_{j,r} = 0 \Leftrightarrow x_j \in \,]-\infty, t_{j,r}]$ by imposing:

$$\mu_{j,r} \geq \mu_{j,r+1}, \forall r \in [\![1, R_j - 1]\!] \tag{5.12}$$

For each tree m, let $\mathcal{V}_m^{j,r}$ be the set of nodes that split on feature j at threshold $t_{j,r}$ for $r \in [\![1, R_j]\!]$. The following constraints ensure that the value of $\mu_{j,r}$ is consistent across the nodes of tree m:

$$\mathrm{f}_{m,\mathbf{left}(v)} \leq 1 - \mu_{j,r}, \forall r \in [\![1, R_j]\!], \forall v \in \mathcal{V}_m^{j,r}, \tag{5.13a}$$

$$\mathrm{f}_{m,\mathbf{right}(v)} \leq \mu_{j,r}, \qquad \forall r \in [\![1, R_j]\!], \forall v \in \mathcal{V}_m^{j,r}. \tag{5.13b}$$

Constraint (5.13a) ensures that if the value of x_j is in the interval $]t_{j,r}, \infty[$, then at node v we cannot go to the left child. Similarly, constraint (5.13b) ensures that if the value of x_j is in the interval $]-\infty, t_{j,r}]$, then at node v we can not go to the right child.

Our separation oracle resembles closely the problem of finding counterfactual explanations [79] or classifier evasions [80]. Yet, optimizing over the set of leaves (see Definition 5.3) instead of the continuous variable x offers numerical benefits. Determining the exact value of x, as needed in counterfactual explanation, requires checking for strict inequalities when going right on a threshold.

Table 5.1 Comparison of FIPE$-\|\cdot\|_0$ and FIPE$-\|\cdot\|_1$ for pruning AdaBoost ensembles with 50 learners. The table shows the number of learners in the pruned ensemble $\|w\|_0$, test faithfulness to the original ensemble Fi, test accuracy Acc, and rounded number of oracle calls K. All results are averaged over five repetitions with different train/test splits.

	FIPE$-\|\cdot\|_0$				FIPE$-\|\cdot\|_1$				
Dataset	$\|w\|_0$	K	**Fi**	**Time (s)**	$\|w\|_0$	K	**Fi**	**Time (s)**	**Acc**
Breast-Cancer	20	32	100%	166	20	25	100%	13	95.33%
COMPAS	13	16	100%	72	13	16	100%	14	66.66%
ELEC2	12	2	100%	3	12	2	100%	7	77.30%
FICO	16	22	100%	456	16	19	100%	30	71.42%
HTRU2	9	18	100%	25	9	13	100%	26	97.75%
House-16H	35	42	100%	643	35	40	100%	235	85.57%
Ionosphere	27	57	100%	4 475	27	47	100%	132	90.70%
Pima-Diabetes	25	28	100%	189	25	24	100%	16	77.92%
PoL	16	6	100%	6	16	6	100%	8	90.90%
Seeds	9	6	100%	5	10	4	100%	4	89.05%
Spambase	26	48	100%	765	26	43	100%	118	93.38%

5.4 Computational Experiments

We now study the value of FIPE for pruning large training ensembles while maintaining faithfulness. We investigate how much FIPE can prune ensembles and its scalability over several datasets. We analyze the behavior of FIPE in terms of what trees are removed, and how pruning is impacted by the size of the original ensemble. Finally, we compare our approach to baselines from the recent literature.

In this section, we focus our experiments on AdaBoost classifiers. Experiments with other boosted ensembles, such as XGBoost and LightGBM, are provided in Section B.2 with essentially the same conclusions. We also study random forests in this appendix. We perform our analyses across 11 datasets commonly used in previous studies. In each experiment, we split the data set into a training dataset (80%) and a test dataset (20%) to measure faithfulness and test accuracy. This process is repeated over five different random seeds. The characteristics of the datasets are presented in Table 5.2: their number of samples n, number of features p including the number of numerical p_N and binary p_B features, and number of classes C.

All experiments are implemented in python. We used scikit-learn for training the ensembles. All optimization problems are solved to global optimality using the commercial solver Gurobi v11.0. The experiments are conducted on a computing grid. Each experiment utilizes a single core of an Intel Xeon Gold 6258R CPU running at 2.7 GHz and is

Table 5.2 Characteristics of the data sets

Dataset	n	p	p_N	p_B	C
Seeds	210	7	7	0	3
Ionosphere	351	33	32	1	2
Breast-Cancer-Wisconsin	683	9	9	0	2
Pima-Diabetes	768	8	8	0	2
Spambase	4 601	57	57	0	2
COMPAS-ProPublica	6 907	12	0	12	2
PoL	10 082	26	26	0	2
FICO	10 459	17	0	17	2
House-16H	13 488	16	16	0	2
HTRU2	17 898	8	8	0	2
ELEC2	38 474	7	7	0	2

provided with 4 GB RAM. `FIPE` is provided as a standalone `python` package at https://www.github.com/eminyous/fipe. To reproduce the results of this paper, we provide the code and the datasets at https://www.github.com/eminyous/fipe-experiments.

5.4.1 Main results

First, we evaluate how much `FIPE` is able to prune tree ensembles while certifying faithfulness to the original model. We compare both the certifiable minimal-size and fast but approximate, respectively denoted by `FIPE`$-\|\cdot\|_0$ and `FIPE`$-\|\cdot\|_1$. We measure the number of learners in the pruned ensemble, the faithfulness to the original ensemble on the test set denoted Fi, the test accuracy Acc, and the number of oracle calls rounded to its upper integer denoted by K. All performance metrics are average over the five repetitions.

The results are presented in Table 5.1 for ensembles of 50 learners. They show that `FIPE` is consistently able to prune `AdaBoost` ensembles while maintaining faithfulness. Further, we observe that `FIPE`$-\|\cdot\|_1$ achieves the same pruned size as `FIPE`$-\|\cdot\|_0$ for all but one dataset, while having significantly faster pruning times.

Result 5.1. Using $\|\cdot\|_1$ in `FIPE` typically yields pruning performance similar to $\|\cdot\|_0$ but with significantly faster runtimes.

Because `FIPE`$-\|\cdot\|_0$ struggles to scale to large datasets when the ensemble size increases, we restrict our study to `FIPE`$-\|\cdot\|_1$ and investigate its scalability to large ensembles. Table 5.3 shows that `FIPE`$-\|\cdot\|_1$ is consistently able to reduce the number of active estimators while maintaining the fidelity of the original ensemble. Further, the results show that it scales well to large datasets and ensemble sizes. Another remarkable result shown by Table 5.3 is that the number of oracle calls K is relatively small, even for large ensembles. Indeed, it

is possible to certify that the pruned model is functionally identical to the original one on the entire feature space with less than 200 calls.

Table 5.3 Pruning with FIPE$-\|\cdot\|_1$ on AdaBoost ensembles with 100, 200, 500, and 1000 base learners.

	$M = 100$					$M = 200$				
Dataset	$\|w\|_0$	K	**Fi**	**Acc**	**Time (s)**	$\|w\|_0$	K	**Fi**	**Acc**	**Time (s)**
Breast-Cancer	26	34	100%	95.33%	37	31	42	100%	95.47%	110
COMPAS	13	16	100%	66.70%	39	13	17	100%	66.73%	99
ELEC2	20	6	100%	79.77%	120	40	32	100%	80.40%	1 272
FICO	18	24	100%	71.54%	107	18	24	100%	71.61%	208
HTRU2	29	34	100%	97.77%	267	40	46	100%	97.78%	752
House-16H	55	66	100%	86.29%	1 520	86	116	100%	86.86%	6 114
Ionosphere	42	66	100%	91.55%	835	60	85	100%	90.70%	2 184
Pima-Diabetes	36	38	100%	76.49%	61	49	57	100%	76.62%	282
PoL	18	9	100%	92.97%	40	27	21	100%	93.16%	194
Seeds	10	4	100%	89.05%	7	10	4	100%	89.05%	11
Spambase	42	72	100%	93.64%	1 624	63	92	100%	94.31%	2 867

	$M = 500$					$M = 1000$				
Dataset	$\|w\|_0$	K	**Fi**	**Acc**	**Time (s)**	$\|w\|_0$	K	**Fi**	**Acc**	**Time (s)**
Breast-Cancer	40	53	100%	95.18%	384	45	62	100%	95.33%	904
COMPAS	13	15	100%	66.73%	218	13	16	100%	66.73%	616
ELEC2	79	70	100%	81.10%	8 762	104	97	100%	81.20%	26 493
FICO	18	26	100%	71.63%	593	18	25	100%	71.63%	1 434
HTRU2	70	92	100%	97.80%	5 517	99	128	100%	97.81%	18 637
House-16H	124	155	100%	87.18%	14 381	151	172	100%	87.12%	27 127
Ionosphere	91	130	100%	91.27%	4 890	114	156	100%	90.42%	7 014
Pima-Diabetes	70	88	100%	76.49%	1 264	96	126	100%	74.68%	4 145
PoL	52	61	100%	93.17%	1 840	75	88	100%	93.64%	8 059
Seeds	10	4	100%	89.05%	22	10	4	100%	89.05%	44
Spambase	103	124	100%	94.55%	6 294	148	204	100%	94.64%	19 040

Result 5.2. FIPE$-\|\cdot\|_1$ consistently provides small pruned ensembles even when the dataset and original ensemble are large.

5.4.2 Analysis of the pruned ensembles

Our previous experiment shows that close to 90% of the base learners of AdaBoost ensembles are superfluous. That is, these learners can be effectively pruned without any change in the prediction function, if the remaining learners are reweighted adequately. We now study how FIPE is able to prune by jointly removing and reweighting the base learners. Figure 5.3

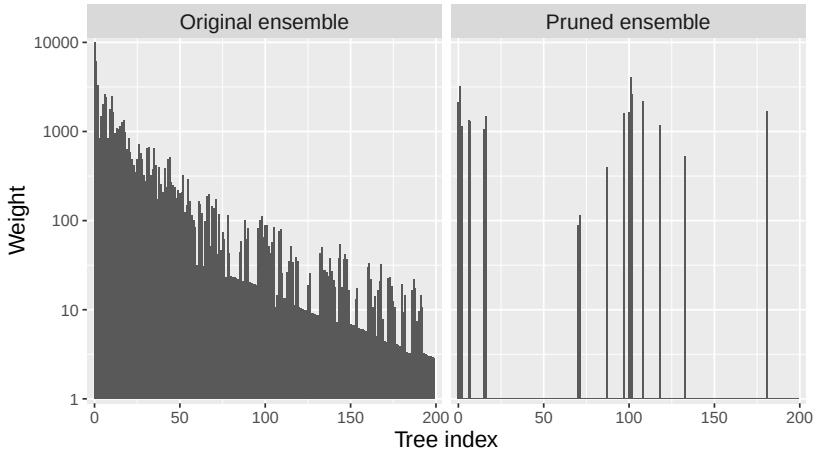


Figure 5.3 Weights of learners in the original and pruned ensembles on the FICO dataset with $M = 200$.

shows the weights of the active estimators in the original and pruned ensembles. It shows that some learners that seem to have a low predictive power in the original ensemble because they are created later in the training process (Figure 5.3 left) in fact have a large predictive power after pruning and reweighting (Figure 5.3 right). For instance, tree index 180 has a weight of around 30 in the original ensemble and a weight of around 1300 in the pruned model. Hence, it is possible to prune learners created early in the training process by assigning a large weight to later learners.

It is also interesting to observe that the size of the pruned ensemble grows significantly slower than the size of the original ensemble. This can be seen in Table 5.3, which shows the size of the pruned ensemble for varying sizes of the original ensemble. Yet, the test accuracy tends to increase with the size of the ensemble. This suggests that only a small percentage of the newly created learners have a significant predictive power.

Result 5.3. As boosted ensembles grow, their accuracy increases but they tend to have more superfluous base learners.

5.4.3 Comparison with "non-faithful" baselines

We compare `FIPE` with several baselines from other recent studies. To the best of our knowledge, we are the first to study functionally-identical pruning of ensembles. Hence, all existing baselines cannot guarantee faithfulness to the original ensemble. We implement the method of [47] denoted as `IMD`, [48] denoted as `IC` and [51] denoted as `DREP`. We use the implementation from the `pypruning` package [20]. A key difference with our approach is that these methods require the size of the pruned ensemble as a hyperparameter. To

ensure a fair comparison, we set the value of this hyperparameter as the size of the pruned ensemble identified by FIPE.

Table 5.4 shows the faithfulness Fi of the pruned model to the original model as well as the test accuracy Acc of the pruned model. The results show that the baselines have significantly lower accuracy and faithfulness to the original model on average, regardless of the size of the original ensemble. Hence, FIPE outperforms all baselines w.r.t. both faithfulness and accuracy.

Table 5.4 Comparison of FIPE$-\|\cdot\|_1$ and non-faithful baselines for varying size of the original ensemble M. Results are averaged over all datasets and repetitions.

M	$\|w\|_0$		Fi	Acc
100	28.13	FIPE	100.00%	85.55%
		DREP	87.56%	81.68%
		IC	87.50%	80.56%
		IMD	37.08%	41.08%
200	39.55	FIPE	100.00%	85.70%
		DREP	86.71%	81.25%
		IC	87.33%	80.77%
		IMD	32.92%	37.57%
500	60.95	FIPE	100.00%	85.83%
		DREP	85.98%	80.72%
		IC	86.40%	80.29%
		IMD	28.06%	32.75%
1000	79.33	FIPE	100.00%	85.66%
		DREP	85.72%	80.70%
		IC	82.85%	79.04%
		IMD	34.95%	38.30%

Result 5.4. Functionally identical pruning guarantees that there is no decrease in test accuracy. In contrast, existing pruning methods have no guarantee and decrease the test accuracy in practice.

5.5 Related Literature

Many approaches have been proposed for ensemble pruning. We provide an overview of the closely related literature in this section, and refer an interested reader to [45] and Chapter 6 of [46] for a more exhaustive review.

Pruning tree ensembles. We can distinguish three main frameworks for pruning: order-based, cluster-based, and optimization-based. The first two are approximate and therefore do not guarantee that the pruned ensembles are of minimal size. Early approaches rank the elements of the ensemble and select them in a greedy fashion [21]. This can be regularized by encouraging the diversity of the selected classifiers [47, 48]. Cluster-based pruning is performed in two steps: first, cluster the classifiers in representative groups, then select a representative element from these groups [49].

A few optimization-based approaches have also been proposed. They are based on mixed-integer quadratic formulations, which typically do not scale to large models and datasets. [50] present a quadratic formulation and propose a relaxation based on semi-definite programming. [51] and [52] extend this idea with diversity measures.

Finally, an interesting stream of literature focuses on building models with small memory requirements [53]. [54] study how to compress random forests by encoding the trees into binary codes. [55] simplify tree ensembles by reducing the number of distinct splitting rules and sharing the rules across trees.

Optimal pruning. Optimal methodologies have demonstrated effectiveness across several pruning tasks. [56] present an optimal algorithm for pruning nodes of decision trees. [23] show how to jointly prune nodes of random forest ensembles using a coordinated-descent algorithm. The tree-based structure of tree ensembles is especially suitable for combinatorial optimization methods, allowing to adapt the training algorithm or post-process ensembles to improve fairness or interpretability [57]. Combinatorial optimization methods have also proven successful for pruning large deep neural networks [81–84].

Faithfulness. Functionally identical pruning introduces a significant shift in perspective compared to existing works. Traditionally, the question asked was "*given a target size, what is the best subset I should select from my ensemble?*". In contrast, we look for the smallest reweighted subset of trees that provide the same prediction function. The closest work to ours is likely [58], who show how to transform a tree ensemble into a decision tree with an identical prediction function on the entire feature space. However, this born-again tree is computationally expensive to obtain and might grow exponentially large in some cases as some ensembles are not efficiently representable as single trees.

5.6 Conclusion

This paper proposed new methodological tools to prune additive tree ensembles while guaranteeing that their prediction function remains unchanged. Through extensive experimental analyses, we have demonstrated that boosted ensembles can be significantly compressed. This suggests that boosted ensembles have many superfluous learners, which can be removed entirely from the ensemble on the condition that the remaining learners are adequately reweighted. A significant advantage of such pruning is that, contrary to existing pruning methods, it guarantees that the test accuracy of the pruned ensemble does not decrease.

This work opens numerous research perspectives. First, while we focused on being certifiably identical to the original model on the entire feature space, it is worthwhile to study how to impose faithfulness on a subspace, such as the space of plausible observations as [79]. This might allow stronger compression. For most practical situations, it is possible to avoid using the combinatorial optimization oracle. Using an approximate oracle may lead to a faster algorithm that lacks the faithfulness certificate but might empirically be very close to the original model. Last but not least, pursuing a disciplined analysis of model compression with faithfulness guarantees for other machine learning models is a promising direction for future work.

Acknowledgements

This work was supported by the Canada Excellence Research Chair in Data Science for Real-Time Decision-Making and the SCALE-AI Research Chair in Data-Driven Supply Chains. The authors would like to thank the anonymous reviewers for their valuable suggestions.

CHAPITRE 6 ARTICLE 3 : IMPROVED PRIMAL SIMPLEX WITH POLYNOMIAL NUMBER OF CALLS TO THE COMPLEMENTARY SUBPROBLEM

Authors: Youssouf Emine, François Soumis and Issmail El Hallaoui

Submitted: Journal of Optimization Theory and Applications

Date de soumission: 10 avril 2025

The primal simplex algorithm is still one of the most used algorithms by the operations research community to solve various problems in the industry that can be modeled as integer linear program of mixed integer linear program. It moves from basis to adjacent one until optimality. The number of bases can be very huge, even exponential, due to degeneracy or when we have to go through all of the extreme points very close to each other. The improved primal simplex algorithm (IPS) is efficient against degeneracy but when there is no degeneracy, it behaves exactly as a primal simplex and consequently, it may suffer from the same limitations.

We present a new formulation of the complementary problem, i.e., the auxiliary subproblem used by the IPS algorithm to find descent directions, that guarantees a significant improvement of the objective value at each iteration until we reach an $\varepsilon-$approximation of the optimal value. We prove theoretically that the number of needed descent directions is polynomial.

6.1 Introduction

Consider the linear program (LP) in standard form

$$\begin{aligned} \max_{x} \quad & c^{\intercal}x \\ & Ax = b \\ & x \geq 0 \end{aligned} \tag{6.1}$$

where $A \in \mathbb{R}^{m\times n}$ is of full rank, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. Usually the x variables or some of them are integer and the LP problem is obtained by relaxing the integrality constraints of an integer linear program (ILP) or a mixed integer linear program (MILP) which are a common class of optimization problems in the industry. To efficiently solve the ILP or MILP problems, we use a branch-and-bound [59] or branch-and-cut algorithm [60] to solve

the LP relaxation of the problem at each node of the tree. It takes a large amount of LP problems to solve the ILP or MILP problems. So the efficiency of the LP solver is crucial to solve the ILP or MILP problems.

In nondegenerate or slightly degenerate cases, the standard primal simplex algorithm efficiently solves the problem. Its aptitude for re-optimization (especially in column generation contexts) renders it highly valuable in applications such as transportation and scheduling, where the number of variables is vast and extreme degeneracy is common. Many algorithms have been developed to solve LPs, including the simplex [25], interior-point [65], and gradient descent [66] methods. It is well known that an optimal solution of LP is always at an extreme point of the feasible region (see [85] for a detailed proof).

The simplex method uses a sequence of *pivots* to move from one extreme point to another, keeping the objective value nondecreasing until an optimal solution is reached. At each iteration, this algorithm divides the variables into basic and nonbasic sets and selects a nonbasic variable to enter the basis and a basic variable to leave. The choice of these variables is crucial, as it determines the direction in which the algorithm moves. A rule for selecting these variables is called a *pivot rule*. One of the most popular pivot rules is Dantzig's rule [25], which computes the reduced cost of each variable and selects the one with the most negative reduced cost. To avoid cycling, Bland's rule [62] can be used, which selects the variable with the smallest index among those with the most negative reduced cost. The steepest edge rule [63] is another pivot rule that selects the variable that leads to the best improvement in the objective value. For an extensive review of pivot rules, see [64]. The simplex method can be implemented in either the primal or dual form, with the primal simplex being the most common in practice.

The interior-point method is another popular algorithm for solving LPs. It is based on the central path, which is a trajectory in the feasible region that approaches an optimal solution as the barrier parameter tends to zero. The algorithm iteratively solves a sequence of barrier subproblems, each of which is a nonlinear problem. Indeed, the barrier subproblem is

$$\min_{x} \quad c^{\intercal}x - \kappa \sum_{i=1}^{m} \log(x_i) \tag{6.2a}$$

$$Ax = b \tag{6.2b}$$

$$x > 0 \tag{6.2c}$$

where $\kappa > 0$ is the barrier parameter. When the barrier parameter is small, the solution of the barrier subproblem is not exactly an extreme point of the feasible region, but it is close

to it. To obtain an extreme point, the algorithm uses a crossover step. The interior-point method is known for its polynomial complexity, but in practice, the crossover step can be computationally expensive.

The gradient descent method is a first-order optimization algorithm that iteratively moves in the direction of the negative gradient of the objective function. It is widely used in machine learning and deep learning, where the objective function is typically nonconvex. Recent works show that the gradient descent method can be used to solve LPs in practice which was not common in the past [67]. The algorithm is simple to implement and can be parallelized, but it may converge slowly or get stuck in local minima. Both the interior-point and gradient descent methods require a crossover step to identify an exact extreme point of the feasible region.

While the interior-point and gradient descent methods are efficient for large-scale problems, the simplex method is still one of the most widely used algorithms for solving LPs in practice. However, two main difficulties affect the primal simplex method. First, *degeneracy* can cause the algorithm to perform many pivots without any improvement in the objective value, since multiple bases may correspond to the same extreme point. Although several techniques have been proposed to address this issue, many focus solely on variable selection [69] or introduce minor perturbations [70] to break the degeneracy. Another approach aims to take advantage of degeneracy rather than just to minimize its negative effects. By exploiting insights from a geometrical perspective, a degenerate vertex of the n-dimensional polyhedron described by the LP's constraints is the crossing point of more than $n-1$ facets of the polyhedron [71]. Degeneracy thus corresponds to a local excess of information, which suggests that a smaller problem may be considered locally to make progress from a degenerate solution. In [72], they introduce a particular degeneracy structure in the LU decomposition of the basis, which involves fewer calculations when performing degenerate pivots. The paper [73] generalizes the definition of basis to include deficient bases containing p independent columns, with $p < m$. When degeneracy occurs, a deficient basis, smaller than the usual square basis matrix, may be used to perform the pivot calculations. Second, even when an improving pivot is found, the improvement can be very small if all neighboring extreme points are very close. In such cases, the algorithm may require a large number of iterations to reach optimality.

The Improved Primal Simplex (IPS) [26] algorithm was developed to overcome degeneracy by ensuring that each pivot leads to a strictly better extreme point. Nevertheless, when degeneracy is absent, IPS behaves essentially like the standard primal simplex, and the step improvements remain marginal. This algorithm identifies the compatible variables,

i.e., those in the span of non-degenerate variables, reformulates a reduced problem, and solves it to find a strictly improving pivot in the compatible subspace. This ensures that no compatible variable with a negative reduced cost is left in the basis. The algorithm then reformulates a complementary problem to find a strictly improving direction moving to a strictly better extreme point. The complementary problem is observed to be dual *nondegenerate* or *weakly* degenerate and it can be solved efficiently using the dual simplex method. The algorithm is guaranteed to certify the optimality of the solution if the complementary problem has a nonnegative objective value. IPS can be viewed as a generalization of the previous work on dynamic constraints aggregation [86] showing clear success for solving set partitioning problems (see [26, 87, 88]). It has already been shown in [89] that the IPS algorithm is a special case of a Dantzig-Wolfe decomposition algorithm. In practice, IPS reduces the computation time by a factor of more than 4 on driver and bus scheduling problems of 2000 constraints and $6000 - 10000$ variables with in 50% of degeneracy [89]. It also reduces the computation time by a factor of more than 10 on real life fleet assignment and aircraft routing problems of 5000 constraints and $17000 - 25000$ variables with in $65\% - 70\%$ of degeneracy [89].

Even though the IPS algorithm is efficient in practice and always provides a strictly improving pivot when degeneracy is present, it may still require a large number of iterations (call to complementary problem) to progress when neighboring extreme points are very close. Using steepest edge as pricing criterion selects the best improvement but it is time consuming and anyway, at some extreme points, the best improvement can be small when all neighbors in the polyhedron are very close. For example, when a very degenerate problem is perturbed to escape degeneracy, the huge number of bases at an extreme point is transformed into a huge number of very close extreme points. In the worst case, as we all know, the primal simplex algorithm could need an exponential number of pivots to solve the problem [68]. Finding a polynomial adaptation of the primal simplex is classified as an important problem for the 21^{st} century [68].

In this theoretical paper, we introduce a new variant of the complementary problem within IPS that guarantees a significant decrease in the objective value at every iteration even in highly degenerate scenarios or when neighboring extreme points offer only minor improvements. This enhancement enables the algorithm to jump to a nonadjacent extreme point when beneficial, thus reaching an ε-optimal solution in a polynomial number of calls to the complementary problem solver.

The remainder of the paper is organized as follows. In Section 6.2, we present the standard formulation of the Improved Primal Simplex algorithm. Section 6.3 introduces our new

parametric formulation of the complementary problem and discusses the parameter choices that ensure the desired improvement. Section 6.4 details the theoretical results and their proofs.

6.2 Improved Primal Simplex Generic Framework

For $\mathcal{I} \subset \{1, \ldots, m\}$ and $\mathcal{J} \subset \{1, \ldots, n\}$, let $A_{\mathcal{I},\mathcal{J}}$ denote the submatrix of A with rows indexed by $\mathcal{I}$ and columns indexed by $\mathcal{J}$. When $\mathcal{I} = \{i\}$ and $\mathcal{J} = \{j\}$, we write $A_{i,j}$ for the element in the i-th row and j-th column of A. We also use ":" to denote all indices of a set; for example, $A_{\mathcal{I},:}$ is the submatrix with rows in $\mathcal{I}$ and all columns, $A_{:,\mathcal{J}}$ is the submatrix with all rows and columns in $\mathcal{J}$ while $x_{\mathcal{J}}$ denotes the subvector of x with components indexed by $\mathcal{J}$. Throughout the paper, $\mathcal{B}$ and $\mathcal{N}$ denote the sets of basic and nonbasic indices, respectively.

The central idea behind the Improved Primal Simplex (IPS) algorithm is to decompose the standard LP (6.1) into two subproblems easier to solve: the *reduced problem* and the *complementary problem* that will be defined below. The algorithm alternates between these subproblems until an optimal solution is reached, with each iteration yielding a new basic feasible solution. Let $\underline{x}$ be a basic feasible solution of (6.1) with basis $\mathcal{B}$ and nonbasic set $\mathcal{N}$. Define $\mathcal{P} \coloneqq \{j \in \mathcal{B} : \underline{x}_j > 0\}$ as the set of indices corresponding to positive components of $\underline{x}$ and $\mathcal{Z} \coloneqq \{j \in \mathcal{B} : \underline{x}_j = 0\}$ as the indices corresponding to zero components.

Definition 6.1 (Compatible Variables)**.** Given $\mathcal{P}$ and $\mathcal{Z}$ defined as above, the variable x_j is *compatible* with $\mathcal{P}$ if $A_j \in \text{Span}\{A_{:,\mathcal{P}}\}$; otherwise, it is *incompatible*.

In essence, compatibility means that the column of A corresponding to x_j can be expressed as a linear combination of the columns indexed by $\mathcal{P}$. Let $\mathcal{C}$ and $\mathcal{I}$ denote the sets of compatible and incompatible variables, respectively.

The *reduced problem* RP is defined as

$$\begin{aligned} \max_{x_{\mathcal{C}}} \quad & c_{\mathcal{C}}^{\intercal} x_{\mathcal{C}}, \\ & [A_{\mathcal{B}}]^{-1}_{\mathcal{P},:} \, A_{:,\mathcal{C}} \, x_{\mathcal{C}} = [A_{\mathcal{B}}]^{-1}_{\mathcal{P},:} \, b, \\ & x_{\mathcal{C}} \geq 0. \end{aligned} \tag{6.3}$$

This linear program involves only $|\mathcal{C}|$ variables, a significant reduction from n, and its constraint matrix has rank $|\mathcal{P}|$, the number of positive components in $\underline{x}$. As shown in [26], any pivot on a compatible variable with a negative reduced cost is nondegenerate, ensuring that each pivot strictly improves the objective.

The RP arises by partitioning the variables into compatible ($\mathcal{C}$) and incompatible ($\mathcal{I}$) sets and setting $x_{\mathcal{I}} = 0$. Writing the original constraints as

$$A_{:,\mathcal{C}}\, x_{\mathcal{C}} + A_{:,\mathcal{I}}\, x_{\mathcal{I}} = b, \tag{6.4}$$

and imposing $x_{\mathcal{I}} = 0$ yields

$$A_{:,\mathcal{C}}\, x_{\mathcal{C}} = b. \tag{6.5}$$

Multiplying (6.5) by $[A_{\mathcal{B}}]^{-1}_{\mathcal{P},:}$ recovers the constraints in (6.3). The *complementary problem* CP is defined as

$$\begin{aligned}
\min_{u\in\mathbb{R}^{\mathcal{P}},\, v\in\mathbb{R}^{\mathcal{I}}} \quad & -c_{\mathcal{P}}^{\intercal} u + c_{\mathcal{I}}^{\intercal} v, && \text{(6.6a)}\\
& -A_{:,\mathcal{P}}\, u + A_{:,\mathcal{I}}\, v = 0, && \text{(6.6b)}\\
& v \geq 0, && \text{(6.6c)}\\
& \mathbf{1}^{\intercal} v = 1, && \text{(6.6d)}
\end{aligned}$$

where $\mathbf{1}$ is the vector of ones of dimension dictated by the context. This formulation captures the cone of feasible directions that can be added to $\underline{x}$ to obtain a new feasible solution of (6.1). The CP is unbounded in the absence of the normalization constraint (6.6d), which ensures that the feasible region is a bounded polyhedron. The CP is solved to find a feasible descent direction that improves the objective value of the LP. The algorithm terminates when the optimal value of the CP is nonnegative, indicating that the current basic feasible solution is optimal for the LP.

Proposition 6.1 ([26]). *If the optimal value of the CP (6.6) is nonnegative, if and only if $\underline{x}$ is optimal for LP (6.1).*

Although the CP in (6.6) involves $|\mathcal{P}| + |\mathcal{I}|$ variables, the u variables can be eliminated by multiplying the constraint (6.6b) by the matrix $[A_{\mathcal{B}}]^{-1}_{\mathcal{P},:}$ leading to $u = [A_{\mathcal{B}}]^{-1}_{\mathcal{P},:}\, A_{:,\mathcal{I}}\, v$. and we recover a reduced version of the CP as

$$\begin{aligned}
\min_{v\in\mathbb{R}^{\mathcal{I}}} \quad & \left(c_{\mathcal{I}}^{\intercal} - c_{\mathcal{P}}^{\intercal}\, [A_{\mathcal{B}}]^{-1}_{\mathcal{P},:}\, A_{:,\mathcal{I}}\right) v, && \text{(6.7a)}\\
& [A_{\mathcal{B}}]^{-1}_{\mathcal{Z},:}\, A_{:,\mathcal{I}}\, v = 0, && \text{(6.7b)}\\
& v \geq 0, && \text{(6.7c)}\\
& \mathbf{1}^{\intercal} v = 1. && \text{(6.7d)}
\end{aligned}$$

The complementary problem (6.7) can also be reformulated using the *compatibility matrix* $M = \begin{bmatrix} A_{\mathcal{Z},\mathcal{P}} \left[A_{\mathcal{B}}\right]^{-1}_{\mathcal{P},\mathcal{Z}} & -\mathbf{I} \end{bmatrix}$ where $\mathbf{I}$ is the identity matrix of size $|\mathcal{Z}| \times |\mathcal{Z}|$. The CP (6.7) can then be written as

$$\min_{v \in \mathbb{R}^{\mathcal{I}}} \quad \left(c_{\mathcal{I}}^{\intercal} - c_{\mathcal{P}}^{\intercal} \left[A_{\mathcal{B}}\right]^{-1}_{\mathcal{P},:} A_{:,\mathcal{I}}\right) v, \tag{6.8a}$$

$$M A_{:,\mathcal{I}} v = 0, \tag{6.8b}$$

$$v \geq 0, \tag{6.8c}$$

$$\mathbf{1}^{\intercal} v = 1. \tag{6.8d}$$

The formulation using the compatibility matrix has been studied in [90] where the authors used it to define the *compatibility degree* of a variable. The compatibility degree allows to use *multiphase technique* to accelerate the solving process of the CP (6.8) as shown in [90]. It is also instructive to consider the dual of the CP (6.6). Its formulation is given by

$$\max_{y \in \mathbb{R},\, \pi \in \mathbb{R}^m} \quad y, \tag{6.9a}$$

$$c_{\mathcal{I}} - \pi^{\intercal} A_{:,\mathcal{I}} \geq y\, \mathbf{1}, \tag{6.9b}$$

$$c_{\mathcal{P}} - \pi^{\intercal} A_{:,\mathcal{P}} = 0. \tag{6.9c}$$

The normalization constraint in the primal can be generalized by replacing $\mathbf{1}$ with an arbitrary vector $\lambda \in \mathbb{R}^{\mathcal{I}}_{>0}$. In that case, the formulation of the primal CP becomes

$$\min_{u \in \mathbb{R}^{\mathcal{P}},\, v \in \mathbb{R}^{\mathcal{I}}} \quad - c_{\mathcal{P}}^{\intercal} u + c_{\mathcal{I}}^{\intercal} v, \tag{6.10a}$$

$$- A_{:,\mathcal{P}}\, u + A_{:,\mathcal{I}}\, v = 0, \tag{6.10b}$$

$$v \geq 0, \tag{6.10c}$$

$$\lambda^{\intercal} v = 1. \tag{6.10d}$$

Its dual is then given by

$$\max_{y \in \mathbb{R},\, \pi \in \mathbb{R}^m} \quad y, \tag{6.11a}$$

$$c_{\mathcal{I}} - \pi^{\intercal} A_{:,\mathcal{I}} \geq y\, \lambda, \tag{6.11b}$$

$$c_{\mathcal{P}} - \pi^{\intercal} A_{:,\mathcal{P}} = 0. \tag{6.11c}$$

Various studies have investigated the impact of different choices of λ on the performance of the IPS algorithm. In [26], the authors recommend $\lambda = \mathbf{1}$ to ensure a bounded feasible

region, while [91] demonstrates that alternative choices for λ can improve the algorithm's convergence properties. Furthermore, the authors have explored adaptive strategies for selecting λ based on problem-specific characteristics and the behavior observed during the iterative process. These strategies aim to dynamically adjust λ to better capture the geometry of the dual space, thereby accelerating convergence. To understand the impact of the choice of λ on the performance of the algorithm, we can present an equivalent non linear formulation of the CP (6.10) as follows:

$$\min_{v \in \mathbb{R}^{\mathcal{I}}} \quad \frac{\overline{c}^{\intercal} v}{\lambda^{\intercal} v} \tag{6.12a}$$

$$[A_{\mathcal{B}}]^{-1}_{\mathcal{Z},:} A_{:,\mathcal{I}} v = 0, \tag{6.12b}$$

$$v > 0. \tag{6.12c}$$

where $\overline{c} = c_{\mathcal{I}} - c_{\mathcal{P}}^{\intercal} [A_{\mathcal{B}}]^{-1}_{\mathcal{P},:} A_{:,\mathcal{I}}$. The objective function of the CP (6.12) is a ratio of the reduced cost of the linear combination of the incompatible variables to the normalization constraint.

To summarize, the IPS algorithm alternates between solving the RP (6.3) and the CP (6.7) until the latter yields a nonnegative optimal value. The algorithm is guaranteed to terminate in a finite number of iterations, and each iteration strictly improves the objective value. In the next section, we introduce a new variant of the CP that ensures a significant decrease in the objective value at every iteration, even in highly degenerate scenarios or when neighboring extreme points offer only minor improvements.

6.3 New formulation of the complementary problem

The standard CP (6.7) is designed to capture the cone of feasible descent directions to reach a new improving extreme point. However, the objective improvement obtained by solving the CP can be marginal, especially when the neighboring extreme points are very close. In such cases, as mentioned earlier, the algorithm may require a large number of solving calls to the CP to reach an optimal solution or even an ε-optimal solution. To address this issue, we introduce a new way of partitioning the basic variables into three subsets: $\mathcal{P}_{\geq\delta}$, $\mathcal{P}_{<\delta}$, and $\mathcal{Z}$, where δ is a small positive constant. The set $\mathcal{P}_{\geq\delta}$ contains the indices of basic variables with values larger or equal to δ, while $\mathcal{P}_{<\delta}$ contains those with values smaller than δ. The set $\mathcal{Z}$ remains unchanged and contains the indices of basic variables with value zero. We also redefine the set of compatible variables $\mathcal{C}$ as those compatible with $\mathcal{P}_{\geq\delta}$ instead of $\mathcal{P}$ in the standard formulation of the CP. Finally the incompatible variables set $\mathcal{I}$ is redefined as the set of variables incompatible with $\mathcal{P}_{\geq\delta}$. The main idea behind this new partinioning is

to consider that the variables with value smaller than δ, $\mathcal{P}_{<\delta}$ as *weakly* degenerate variables since they don't provide a large step improvement. We then use the same decomposition idea as in the standard CP (6.7) to find a strictly improving direction. This idea can be formulated as follows:

$$\begin{aligned}
\min_{u,w,v} \quad & -c_{\mathcal{P}_{\geq\delta}}^{\intercal} u - c_{\mathcal{P}_{<\delta}}^{\intercal} w + c_{\mathcal{I}}^{\intercal} v, && \text{(6.13a)}\\
& - A_{:,\mathcal{P}_{\geq\delta}}\, u - A_{:,\mathcal{P}_{<\delta}}\, w + A_{:,\mathcal{I}}\, v = 0, && \text{(6.13b)}\\
& v \geq 0,\ v \in \mathbb{R}^{\mathcal{I}}, && \text{(6.13c)}\\
& w \geq 0,\ w \in \mathbb{R}^{\mathcal{P}_{<\delta}}, && \text{(6.13d)}\\
& u \in \mathbb{R}^{\mathcal{P}_{\geq\delta}}, && \text{(6.13e)}\\
& \mathbf{1}^{\intercal} v + \mathbf{1}^{\intercal} w \leq 1. && \text{(6.13f)}
\end{aligned}$$

The u variables represent the changes in the varibles of $\mathcal{P}_{\geq\delta}$ while w and v represent the increases in the variables of $\mathcal{P}_{<\delta}$ and $\mathcal{I}$ respectively. The objective function represents the reduced cost of the direction defined by (u, v, w). Similar to the standard CP (6.7), the new CP (6.13) captures the cone of feasible descent directions to reach a new improving extreme point. We can also eliminate the u variable by using the following substitution:

$$u = [A_{\mathcal{B}}]^{-1}_{\mathcal{P}_{\geq\delta},:}\ A_{:,\mathcal{I}}\, v - [A_{\mathcal{B}}]^{-1}_{\mathcal{P}_{\geq\delta},:}\ A_{:,\mathcal{P}_{<\delta}}\, w.$$

Using this substitution, the new CP (6.13) can be reformulated as follows:

$$\begin{aligned}
\min_{w,v} \quad & -\left(c_{\mathcal{P}_{<\delta}}^{\intercal} - c_{\mathcal{P}_{\geq\delta}}^{\intercal}\,[A_{\mathcal{B}}]^{-1}_{\mathcal{P}_{\geq\delta},:}\ A_{:,\mathcal{P}_{<\delta}}\right) w + \left(c_{\mathcal{I}}^{\intercal} - c_{\mathcal{P}_{\geq\delta}}^{\intercal}\,[A_{\mathcal{B}}]^{-1}_{\mathcal{P}_{\geq\delta},:}\ A_{:,\mathcal{I}}\right) v, && \text{(6.14a)}\\
& [A_{\mathcal{B}}]^{-1}_{\mathcal{Z},:}\ A_{:,\mathcal{I}}\, v - [A_{\mathcal{B}}]^{-1}_{\mathcal{Z},:}\ A_{:,\mathcal{P}_{<\delta}}\, w = 0, && \text{(6.14b)}\\
& v \geq 0,\ v \in \mathbb{R}^{\mathcal{I}}, && \text{(6.14c)}\\
& w \geq 0,\ w \in \mathbb{R}^{\mathcal{P}_{<\delta}}, && \text{(6.14d)}\\
& \mathbf{1}^{\intercal} v + \mathbf{1}^{\intercal} w \leq 1. && \text{(6.14e)}
\end{aligned}$$

The number of constraints in the new CP (6.14) is $m - |\mathcal{P}_{\geq\delta}|$ and the number of variables is $|\mathcal{P}_{<\delta}| + |\mathcal{I}|$. We can see that the new CP (6.14) has a larger number of constraints and variables compared to the standard CP (6.7). However, as we will show in the remainder of the paper, the new CP (6.14) gives better directions to reach a new improving extreme point.

Proposition 6.2. *If the optimal value of the new CP (6.13) is nonnegative, if and only if the current basic feasible solution is optimal for the LP (6.1).*

Proof. The proof is similar to the proof of the standard CP. We can show that the objective value of the new CP is equal to the reduced cost of the current basic feasible solution. If the optimal value of the new CP is nonnegative, then the reduced cost of the current basic feasible solution is nonnegative, which implies that the current basic feasible solution is optimal for the LP (6.1). □

Similar to the standard CP, the new CP (6.13) captures the cone of feasible descent directions to reach a new improving extreme point and it is unbounded in the absence of the normalization constraint. In the formulation above we used the default normalization constraint $\mathbf{1}^\intercal v + \mathbf{1}^\intercal w = 1$. However, we propose to use the more general normalization constraint $\lambda^\intercal v + \mu^\intercal w = 1$ where $\lambda \in \mathbb{R}^{\mathcal{I}}_{>0}$ and $\mu \in \mathbb{R}^{\mathcal{P}_{<\delta}}_{>0}$. The dual of the new CP is given by

$$\max_{y\in\mathbb{R},\, \pi\in\mathbb{R}^m} \quad -y, \tag{6.15a}$$

$$c_{\mathcal{I}} - \pi^\intercal A_{:,\mathcal{I}} \geq -y\,\lambda, \tag{6.15b}$$

$$c_{\mathcal{P}_{<\delta}} - \pi^\intercal A_{:,\mathcal{P}_{<\delta}} \leq y\mu, \tag{6.15c}$$

$$c_{\mathcal{P}_{\geq\delta}} - \pi^\intercal A_{:,\mathcal{P}_{\geq\delta}} = 0, \tag{6.15d}$$

$$y \geq 0. \tag{6.15e}$$

In this paper, we will discuss the theoretical results and how to choose the parameter δ, the weights λ and μ in the normalization constraint to ensure that the new CP (6.13) yields a significant decrease in the objective value at every iteration. To understand the impact of the choice of the weights λ and μ on the optimal solution of the new CP (6.13), we can present an equivalent non linear formulation of the new CP (6.13) as follows:

$$\min_{w,v} \quad \frac{\overline{s}^\intercal w + \overline{q}^\intercal v}{\lambda^\intercal v + \mu^\intercal w} \tag{6.16a}$$

$$[A_{\mathcal{B}}]^{-1}_{\mathcal{Z},:}\, A_{:,\mathcal{I}}\, v - [A_{\mathcal{B}}]^{-1}_{\mathcal{Z},:}\, A_{:,\mathcal{P}_{<\delta}}\, w = 0, \tag{6.16b}$$

$$v > 0,\ v \in \mathbb{R}^{\mathcal{I}}, \tag{6.16c}$$

$$w > 0,\ w \in \mathbb{R}^{\mathcal{P}_{<\delta}}. \tag{6.16d}$$

where $\overline{s} = c_{\mathcal{P}_{<\delta}} - c^\intercal_{\mathcal{P}_{\geq\delta}}\, [A_{\mathcal{B}}]^{-1}_{\mathcal{P}_{\geq\delta},:}\, A_{:,\mathcal{P}_{<\delta}}$ and $\overline{q} = c_{\mathcal{I}} - c^\intercal_{\mathcal{P}_{\geq\delta}}\, [A_{\mathcal{B}}]^{-1}_{\mathcal{P}_{\geq\delta},:}\, A_{:,\mathcal{I}}$.

6.4 Theoretical results

In this section, we present the main theoretical results of the paper, we also provide the choice of the parameter δ and the weights λ and μ in the normalization constraint to ensure the desired behavior. We start by this lemma:

Lemma 6.3. *Let (y, π) be a feasible solution of the dual of the new CP (6.15). Then,*

$$c_{\mathcal{C}} - \pi^{\intercal} A_{:,\mathcal{C}} \geq 0 \tag{6.17}$$

Proof. The proof is straightforward as

$$\begin{aligned} c_{\mathcal{C}} - \pi^{\intercal} A_{:,\mathcal{C}} &= c_{\mathcal{C}} - \pi^{\intercal} A_{\mathcal{B}} \left[A_{\mathcal{B}}\right]^{-1} A_{:,\mathcal{C}} \\ &= c_{\mathcal{C}} - \pi^{\intercal} A_{:,\mathcal{P}_{\geq\delta}} \left[A_{\mathcal{B}}\right]^{-1}_{\mathcal{P}_{\geq\delta},:} A_{:,\mathcal{C}} \\ &= c_{\mathcal{C}} - c^{\intercal}_{\mathcal{P}_{\geq\delta}} \left[A_{\mathcal{B}}\right]^{-1}_{\mathcal{P}_{\geq\delta},:} A_{:,\mathcal{C}} \\ &\geq 0. \end{aligned}$$

□

We then prove the following proposition that gives a bound on how far is the current basic feasible solution from the optimal solution of the LP (6.1). We will assume that the set of feasible solutions of the LP (6.1) is bounded. We define Δ as the upper bound on components of the feasible solutions:

$$\Delta \coloneqq \max \left\{ \|x\|_{\infty} : Ax = b, x \geq 0 \right\}.$$

In the case where the LP (6.1) is a relaxation of a binary linear program, we can set $\Delta = 1$.

Proposition 6.4. *Let $\underline{x}$ be a basic feasible solution, x^* be the optimal solution of the LP (6.1), and z the optimal value of the CP (6.13). Then,*

$$c^{\intercal} \left(\underline{x} - x^*\right) \leq \left[m\Delta \left\|\lambda\right\|_{\infty} + \mu^{\intercal} \underline{x}_{\mathcal{P}_{<\delta}}\right] |z|. \tag{6.18}$$

Proof. Let (y, π) be a feasible solution of the dual of the new CP (6.15).

$$\begin{aligned} c^{\intercal}_{\mathcal{I}} x^*_{\mathcal{I}} &\geq -y\, \lambda^{\intercal} x^*_{\mathcal{I}} + \pi^{\intercal} A_{:,\mathcal{I}} x^*_{\mathcal{I}} \\ &= -y \lambda^{\intercal} x^*_{\mathcal{I}} + \pi^{\intercal} \left(b - A_{:,\mathcal{C}} x^*_{\mathcal{C}}\right) \\ &= -y \lambda^{\intercal} x^*_{\mathcal{I}} + \pi^{\intercal} b - \pi^{\intercal} A_{:,\mathcal{C}} x^*_{\mathcal{C}}. \end{aligned}$$

Using the fact that $c_{\mathcal{C}} - \pi^\intercal A_{:,\mathcal{C}} \geq 0$ (Theorem 6.3), we have that

$$c^\intercal x^* \geq -y\lambda^\intercal x^*_{\mathcal{I}} + \pi^\intercal b$$

On the other hand, we have that

$$b = A_{:,\mathcal{P}_{\geq\delta}} \underline{x}_{\mathcal{P}_{\geq\delta}} + A_{:,\mathcal{P}_{<\delta}} \underline{x}_{\mathcal{P}_{<\delta}}.$$

Therefore,

$$\begin{aligned}
\pi^\intercal b &= \pi^\intercal A_{:,\mathcal{P}_{\geq\delta}} \underline{x}_{\mathcal{P}_{\geq\delta}} + \pi^\intercal A_{:,\mathcal{P}_{<\delta}} \underline{x}_{\mathcal{P}_{<\delta}} \\
&= c^\intercal_{\mathcal{P}_{\geq\delta}} \underline{x}_{\mathcal{P}_{\geq\delta}} + \pi^\intercal A_{:,\mathcal{P}_{<\delta}} \underline{x}_{\mathcal{P}_{<\delta}} \\
&\geq c^\intercal_{\mathcal{P}_{\geq\delta}} \underline{x}_{\mathcal{P}_{\geq\delta}} + c^\intercal_{\mathcal{P}_{<\delta}} \underline{x}_{\mathcal{P}_{<\delta}} - y\mu^\intercal \underline{x}_{\mathcal{P}_{<\delta}} \\
&= c^\intercal \underline{x} - y\mu^\intercal \underline{x}_{\mathcal{P}_{<\delta}}.
\end{aligned}$$

And so,

$$c^\intercal x^* \geq -y\lambda^\intercal x^*_{\mathcal{I}} + c^\intercal \underline{x} - y\mu^\intercal \underline{x}_{\mathcal{P}_{<\delta}}$$

This implies that

$$\begin{aligned}
c^\intercal \left(\underline{x} - x^*\right) &\leq y \left(\|\lambda\|_\infty \|x^*_{\mathcal{I}}\|_\infty + \mu^\intercal \underline{x}_{\mathcal{P}_{<\delta}} \right) \\
&\leq \left[m\Delta \|\lambda\|_\infty + \mu^\intercal \underline{x}_{\mathcal{P}_{<\delta}} \right] |z| \,.
\end{aligned}$$

□

In the following we state a proposition quantifying the decrease in the objective value of the LP (6.1) at every iteration of the IPS algorithm when using the new CP (6.13). We define

$$\beta_j := \min_{i \in \mathcal{I}} \left\{ \frac{\lambda_i}{[A_{\mathcal{B}}]^{-1}_{j,:} A_{:,i}} \right\}.$$

Proposition 6.5. *Consider the new CP (6.13) with $\underline{x}$ the current basic feasible solution. If the optimal value of the new CP z is negative then the improvement in the objective value of the LP (6.1) at the next iteration is at least $z \times \alpha$ where*

$$\alpha \geq \min \left\{ \gamma^{\geq\delta}, \gamma^{<\delta} \right\},$$

with

$$\gamma^{<\delta} \coloneqq \min_{j \in \mathcal{P}_{<\delta}} \mu_j \underline{x}_j \quad \textit{and} \quad \gamma^{\geq \delta} \coloneqq \min_{j \in \mathcal{P}_{\geq \delta}} \beta_j \underline{x}_j.$$

Proof. Let (u, w, v) be the optimal solution of the new CP (6.13) and z its optimal value. The maximum step size in the direction of the optimal solution of the new CP is given by

$$\alpha \coloneqq \max \left\{ t \geq 0 : \begin{array}{l} \underline{x}_{\mathcal{P}_{\geq \delta}} - tu \geq 0, \\ \underline{x}_{\mathcal{P}_{<\delta}} - t\left(w - v_{\mathcal{P}_{<\delta}}\right) \geq 0 \end{array} \right\}.$$

On one hand, for $j \in \mathcal{P}_{\geq \delta}$, using the fact that for all $i \in \mathcal{I}$, $[A_{\mathcal{B}}]^{-1}_{j,:} A_{:,i} \geq \frac{\lambda_i}{\beta_j}$ and the fact that $v_i \geq 0$, we have

$$\begin{aligned} u_j &= \sum_{i \in \mathcal{I}} [A_{\mathcal{B}}]^{-1}_{j,:} A_{:,i} v_i \\ &\leq \sum_{i \in \mathcal{I}} \frac{\lambda_i}{\beta_j} v_i = \frac{1}{\beta_j} \sum_{i \in \mathcal{I}} \lambda_i v_i = \frac{1}{\beta_j} \lambda^\intercal v \leq \frac{1}{\beta_j}. \end{aligned}$$

$$\underline{x}_j - \gamma^{\geq \delta} u_j \geq \underline{x}_j - \beta_j \underline{x}_j u_j = \underline{x}_j \left(1 - \beta_j u_j\right) \geq 0.$$

On the other hand, for $j \in \mathcal{P}_{<\delta}$, if $w_j - v_j \geq 0$, then $\mu_j \left(w_j - v_j\right) \leq \mu_j w_j \leq \mu^\intercal w \leq 1$ and so

$$\begin{aligned} \underline{x}_j - \gamma^{<\delta} \left(w_j - v_j\right) &\geq \underline{x}_j - \mu_j \underline{x}_j \left(w_j - v_j\right) \\ &= \underline{x}_j \left(1 - \mu_j \left(w_j - v_j\right)\right) \geq 0. \end{aligned}$$

This implies that $\alpha \geq \min\left\{\gamma^{\geq \delta}, \gamma^{<\delta}\right\}$. The improvement in the objective value of the LP (6.1) at the next iteration is at least $z \times \alpha$. □

Henceforth, we assume that $c \geq 0$. This assumption is satisfied by a large class of linear programs, including but not limited to scheduling problems, assignment problems, and network flow problems. It is also worth noting that the assumption $c \geq 0$ is without loss of generality, as we can always consider the variable $y_j = \Delta - x_j$ instead of the variable x_j. In the following proposition, we provide a choice of the parameter δ and the weights λ and μ in the normalization constraint to ensure that the new complementary problem (6.13) yields a significant decrease in the objective value at every iteration. We define $\overline{x}$ as the feasible solution obtained by moving in the direction of the optimal solution of the new CP (6.13) from a basic feasible solution $\underline{x}$ and x^* the optimal solution of the LP (6.1).

Proposition 6.6. *Let $\varepsilon > 0$ be a small positive constant, then there exists a choice of the*

parameter δ and the weights λ and μ in the normalization constraint such that either,

$$c^\intercal \underline{x} \leq \frac{1}{1-\varepsilon} c^\intercal x^* \quad \textit{or} \quad c^\intercal \overline{x} \leq \left(1-\varepsilon^2\right) c^\intercal \underline{x}.$$

Proof. Let $\lambda_i \coloneqq \left\| [A_{\mathcal{B}}]^{-1} A_{:,i} \right\|_\infty$ for $i \in \mathcal{I}$, $\mu_j \coloneqq \frac{\delta}{\underline{x}_j}$ for $j \in \mathcal{P}_{<\delta}$ and z the optimal value of the CP (6.13). Using Theorem 6.4, we have that

$$c^\intercal \left(\underline{x} - x^*\right) \leq m \left[\Delta \left\|\lambda\right\|_\infty + \delta\right] |z| \,.$$

Moreover the definition of β_j implies that,

$$\beta_j = \min_{i \in \mathcal{I}} \left\{ \frac{\left\| [A_{\mathcal{B}}]^{-1} A_{:,i} \right\|_\infty}{[A_{\mathcal{B}}]^{-1}_{j,:} A_{:,i}} \right\} \geq 1.$$

and the definition of $\gamma^{<\delta}$ and $\gamma^{\geq\delta}$ implies that, $\gamma^{<\delta} \geq \delta$ and $\gamma^{\geq\delta} \geq \delta$. Therefore, using Theorem 6.5 then,

$$c^\intercal \left(\overline{x} - \underline{x}\right) \leq z\delta$$

In the case where $z \geq -\frac{\varepsilon}{m(\Delta\|\lambda\|_\infty + \delta)} c^\intercal \underline{x}$ then we have that, $c^\intercal \left(\underline{x} - x^*\right) \leq \varepsilon c^\intercal \underline{x}$ and so $c^\intercal \underline{x} \leq \frac{1}{1-\varepsilon} c^\intercal x^*$. In the case where $z < -\frac{\varepsilon}{m(\Delta\|\lambda\|_\infty + \delta)} c^\intercal \underline{x}$ then we have that,

$$c^\intercal \left(\overline{x} - \underline{x}\right) \leq -\frac{\varepsilon\delta}{m\left(\Delta \left\|\lambda\right\|_\infty + \delta\right)} c^\intercal \underline{x} = -\varepsilon^2 c^\intercal \underline{x}$$

for the choice of the parameter δ. □

Since the formulation of the new CP (6.13) is based on the fact that $\underline{x}$ is a basic feasible solution, it would be great if $\overline{x}$ is a basic solution as well. In some cases the new CP (6.13) will provide a basic solution, for example when the value of w is zero. However, this is not always the case. [92] proved that any decomposition method different from the standard IPS will not always provide a basic solution. However, from $\overline{x}$, we can easily construct a better cost basic solution with a procedure of low polynomial complexity. We then can restart the algorithm by reformulating the CP with the new basic solution. Before introducing the procedure, we prove the following lemma:

Lemma 6.7. $|\{j \colon \overline{x}_j > 0\}| \leq m + |\mathcal{P}_{<\delta}| \leq 2m.$

Proof. The number of nonzeros in the direction of the optimal solution of the new complementary problem is at most $m - |\mathcal{P}_{\geq\delta}| + 1$. After moving from the current basic feasible

solution in the direction of the optimal solution of the new CP, the number of nonzeros in the new basic solution is at most $|\mathcal{P}_{\geq\delta}| + |\mathcal{P}_{<\delta}| + m - |\mathcal{P}_{\geq\delta}| + 1 - 1 = m + |\mathcal{P}_{<\delta}|$. The last inequality is due to the fact that $|\mathcal{P}_{<\delta}| \leq m$. □

It is worth mentioning that the number of nonzeros in the new basic solution is at most $2m$, which is significantly smaller than the number obtained using the interior point method which is typically equal to n. In an extremely degenerate problem where the number of variables n greatly exceeds the number of constraints m, the new basic solution still contains at most $m + |\mathcal{P}_{<\delta}|$ nonzeros, making it much smaller than n. For example, in the case of a crew scheduling problem, the number of variables is typically in the order of tens of thousands to hundreds of thousands, while the number of constraints is in the order of hundreds to thousands. Also the number of nonzeros in a basic solution is typically extremely smaller than the number of constraints.

Crossover procedure. In this paragraph, we explain how to do a simple crossover algorithm from a non basic feasible solution to a better cost basic feasible solution. For instance, let x be a non-basic feasible solution of the LP (6.1). Since x is not a basic feasible solution, the matrix $A_{:,\mathcal{P}}$ is not full column rank where $\mathcal{P} \coloneqq \{j \colon x_j > 0\}$. We can then find a vector an index $i \in \mathcal{P}$ such that $A_{:,i}$ can be expressed as a linear combination of the columns of $A_{:,\mathcal{P}\setminus\{i\}}$, i.e., $A_{:,i} = \sum_{j\in\mathcal{P}\setminus\{i\}} \alpha_j A_{:,j}$ for some $\alpha_j \in \mathbb{R}$. Consider the vector $d \in \mathbb{R}^n$ such that $d_i = 1$, $d_j = -\alpha_j$ for $j \in \mathcal{P}\setminus\{i\}$ and $d_j = 0$ otherwise. It is obvious that $Ad = 0$. Without loss of generality, we can assume that $c^\intercal d \leq 0$ (otherwise we can consider $-d$ instead of d). Let $x^t \coloneqq x + td$ for $t > 0$. Since $Ad = 0$, we have $Ax^t = Ax$; therefore, x^t is a feasible solution of the LP (6.1). Moreover, we have that $c^\intercal x^t = c^\intercal x + tc^\intercal d \leq c^\intercal x$. Therefore, by choosing t such that at least one component of x^t becomes zero, we have a new feasible solution with a better cost and with fewer nonzeros variables than x. We can then repeat this procedure until we obtain a basic feasible solution. This procedure is summarized in the Algorithm 6.1.

As mentionned before, the crossover procedure has a polynomial complexity. Indeed, the complexity of the procedure is dominated by the complexity of solving the linear system $A_{:,\mathcal{P}}d = 0$ which is $O(m^3)$. The number of iterations of the while loop is at most $2m$ (Theorem 6.7) and so the complexity of the crossover procedure is $O(m^4)$.

The diagram in Figure 6.1 shows the flow of the new variant of the IPS algorithm. The algorithm starts by solving the RP (6.3) and then computes the parameter δ and the weights λ and μ in the normalization constraint using the procedure described in the proof of the Theorem 6.5. The algorithm then solves the CP (6.13) and compares its optimal value to the

Algorithm 6.1: Crossover procedure

Input: A non basic feasible solution x, the matrix A, the vector b, the vector c
Output: A basic feasible solution with a better cost

while *x is not a basic feasible solution* **do**
  $\mathcal{P} \leftarrow \{j : x_j > 0\}$;
  Solve the linear system $A_{:,\mathcal{P}} d = 0$ to find a nonzero vector d;
  if $c_{\mathcal{P}}^{\intercal} d > 0$ **then**
    $d \leftarrow -d$;
  end
  $t \leftarrow \min\left\{\frac{x_j}{d_j} : j \in \mathcal{P}, d_j < 0\right\}$;
  $x_{\mathcal{P}} \leftarrow x_{\mathcal{P}} + td$;
end
return x;

threshold $-\frac{\varepsilon}{m(\Delta\|\lambda\|_\infty+\delta)} c^{\intercal}\underline{x}$. If the optimal value is greater than the threshold, the algorithm returns the current basic feasible solution and it is guaranteed by the Theorem 6.6 that the current basic feasible solution is ε-optimal. Otherwise, using the result of Theorem 6.6 we can improve the cost of the current basic feasible. However, the new solution is not a basic feasible solution. We then use the crossover procedure to construct a better cost basic feasible solution as described in the Algorithm 6.1 and restart the algorithm by solving the RP (6.3) with the new basic feasible solution. It would be interesting to see how many times the algorithm needs to solve the CP to reach an ε-optimal solution.

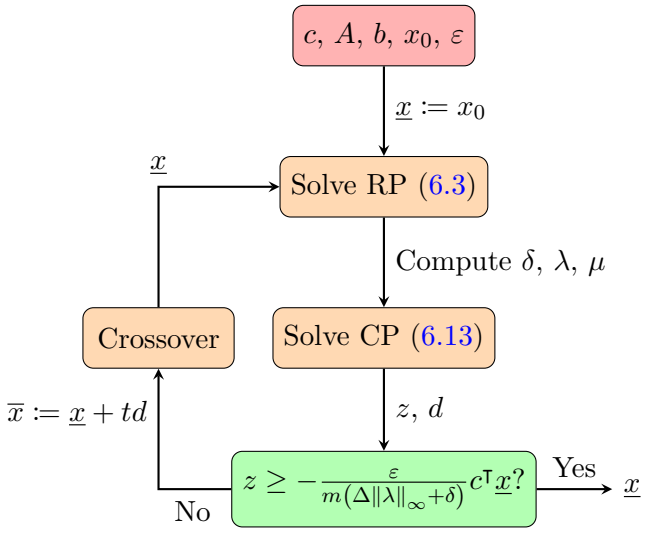
Proposition 6.8. *The number of times the algorithm needs to solve the CP (6.13) to reach an ε-optimal solution is at most $O\left(\frac{1}{\varepsilon^2}\right)$.*

Proof. The proof is straightforward. The cost of the basic feasible solution is improved by at least a factor $1-\varepsilon^2$ at every iteration of the algorithm (Theorem 6.6). Since we assume that the problem is bounded, then the number of iterations is at most $\left|\frac{\ln(K)}{\ln(1-\varepsilon^2)}\right| = O\left(\frac{1}{\varepsilon^2}\right)$ where K is the ratio of the cost of the initial basic feasible solution to the cost of the optimal solution. □

6.5 Conclusion

In summary, we have introduced a new variant of the IPS algorithm that guarantees a significant decrease in the objective value at every iteration, even in highly degenerate scenarios or when neighboring extreme points offer only minor improvements. This enhancement enables the algorithm to jump to a nonadjacent extreme point when beneficial,

Figure 6.1 Flow of the new variant of the IPS algorithm



thus reaching an ε-optimal solution in a polynomial number of calls to the CP solver. The proposed algorithm is a significant step towards finding a polynomial adaptation pivot rule of the primal simplex method, which is classified as an important problem for the 21$^{\text{st}}$ century [68]. The theoretical results presented in this paper provide a solid foundation for future research on the development of efficient iterative algorithms for solving LPs, while remaining extremal at each iteration. The numerical experiments conducted for IPS need to be extended to the new variant to validate the theoretical results and assess the performance of the algorithm on real-world problems. Future work will focus on the implementation of the new variant and its comparison with existing algorithms on a wide range of LP instances.

6.6 Acknowledgements

This research was supported by the Canada Excellence Research Chair in Data Science, the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Fonds de Recherche du Québec - Nature et Technologies (FRQNT). The authors would like to thank Andrea Lodi for his valuable support and insightful discussions.

CHAPITRE 7 DISCUSSION GÉNÉRALE

Cette thèse a exploré en profondeur divers aspects de l'optimisation mathématique et ses applications pour résoudre des problèmes complexes en recherche opérationnelle et en apprentissage automatique. L'objectif principal a été de proposer de nouvelles méthodes, d'analyser leurs performances théoriques et pratiques, et d'ouvrir des perspectives pour des recherches futures dans ces domaines en pleine expansion.

Dans un premier temps, nous avons étudié l'impact du choix de la formulation polynomiale dans la hiérarchie de Lasserre appliquée au problème du plus grand sous-ensemble indépendant d'un graphe. Ce problème, reconnu comme NP-difficile en théorie des graphes, possède de nombreuses applications, notamment en conception de réseaux, en ordonnancement, en allocation de ressources et en bio-informatique. Nous avons démontré que certaines formulations polynomiales permettent d'obtenir des bornes plus serrées et une convergence plus rapide vers la solution optimale, ce qui se traduit par une réduction significative du temps de calcul et une amélioration de la qualité des solutions approchées. Cette observation met en lumière l'importance du choix de la formulation dans la performance des hiérarchies de relaxation semi-définies. À l'avenir, il serait particulièrement intéressant d'examiner l'influence de la formulation polynomiale sur la qualité des hiérarchies de Lasserre pour d'autres problèmes combinatoires, tels que le problème de la coupe maximale, le coloriage de graphes ou encore le problème du voyageur de commerce. De plus, une analyse fine de la structure des polynômes utilisés pourrait permettre de concevoir des formulations encore plus efficaces, adaptées à la nature spécifique de chaque problème.

Dans la deuxième contribution de cette thèse, nous avons proposé une technique innovante de réduction fonctionnelle des ensembles d'arbres de décision, qui sont des modèles de prédiction largement utilisés en apprentissage automatique pour leur capacité à modéliser des relations complexes et non linéaires. Nous avons formulé cette réduction comme un problème d'optimisation exacte et utilisé une stratégie itérative de type adversaire pour simplifier ces modèles sans altérer leur fonction de prédiction. Cette approche permet de réduire la taille et la complexité des modèles, facilitant ainsi leur interprétabilité, leur déploiement et leur utilisation dans des environnements contraints en ressources. Sur des jeux de données réels, et pour divers types de modèles d'ensembles d'arbres de décision, notre approche a permis de réduire significativement la taille des modèles sans perte de précision, tout en maintenant, voire en améliorant, leur robustesse face au bruit et aux perturbations. Cette contribution ouvre la voie à de nouvelles méthodes de compression et de simplifica-

tion des modèles d'apprentissage automatique, avec des applications potentielles en *edge computing*, en Internet des objets et dans des contextes où la transparence des modèles est cruciale, comme la santé ou la finance.

Enfin, la dernière contribution a présenté une nouvelle formulation de la méthode du simplexe amélioré, un outil fondamental en programmation linéaire. Le simplexe amélioré est une méthode itérative visant à trouver une solution optimale à un programme linéaire en contournant les limitations du simplexe classique face à la dégénérescence. Toutefois, dans le pire des cas, cette méthode peut s'avérer lente, car elle ne garantit pas une amélioration significative de la fonction objectif à chaque itération. Nous avons donc proposé une nouvelle formulation garantissant des progrès notables de la fonction objectif à chaque itération jusqu'à l'obtention d'une solution ϵ-approximative optimale. Néanmoins, cette contribution demeure essentiellement théorique et nécessitera des expérimentations complémentaires pour en confirmer l'efficacité en pratique, en particulier sur des problèmes de grande dimension ou présentant une structure particulière. Une perspective future intéressante consisterait à explorer des variantes de notre méthode pour des problèmes de programmation linéaire en nombres entiers, ainsi qu'à étudier son intégration avec d'autres techniques d'accélération, telles que la méthode du multiphase ou les méthodes de points intérieurs.

CHAPITRE 8 CONCLUSION ET PERSPECTIVES

Dans cette thèse, nous avons étudié trois problèmes en optimisation mathématique et en apprentissage automatique. Nous avons apporté trois contributions distinctes, chacune visant à résoudre un problème spécifique de manière innovante et efficace.

Dans la première contribution, nous avons étudié l'impact du choix de la formulation polynomiale dans la hiérarchie de Lasserre appliquée au problème du plus grand sous-ensemble indépendant d'un graphe. L'étude a montré numériquement que le choix de la formulation affecte la qualité des bornes. Nous avons appuyé l'étude numérique par des résultats théoriques pour comparer les bornes des différentes formulations.

La deuxième contribution de cette thèse modélise la réduction fonctionnelle des ensembles d'arbres de décision. Cette réduction permet de diminuer la taille des modèles sans perte de précision. Nous avons présenté un algorithme de réduction et nous avons montré, sur un ensemble de données, l'efficacité de cet algorithme pour réduire significativement la taille de ces ensembles.

Dans la troisième contribution, notre objectif est d'améliorer la méthode du simplexe amélioré. Cette méthode, connue pour son efficacité à résoudre la programmation linéaire dans un contexte dégénéré, peut souffrir de la pseudo-dégénérescence et ainsi être amenée à faire un nombre exponentiel d'appels au problème complémentaire. Nous avons donc proposé une nouvelle méthode de décomposition similaire à celle du simplexe amélioré, qui garantit un nombre d'appels polynomial au problème complémentaire avant d'arriver à une solution presque optimale.

Bien que ces contributions soient diverses, elles partagent un point commun : l'utilisation d'outils d'optimisation avancés pour résoudre des problèmes complexes en recherche opérationnelle et en apprentissage automatique. Cette approche transversale permet non seulement de proposer des solutions innovantes à des problèmes spécifiques, mais aussi de renforcer les liens entre différentes disciplines, favorisant ainsi l'émergence de nouvelles idées et de collaborations interdisciplinaires.

Les perspectives ouvertes par cette thèse sont nombreuses et variées. Tout d'abord, approfondir l'étude de l'effet des formulations polynomiales sur la hiérarchie de Lasserre pour d'autres problèmes combinatoires, tels que le problème de la coupe maximale, constituerait une suite naturelle à nos travaux. L'impact du choix de la formulation polynomiale sur la qualité et la rapidité de convergence des hiérarchies de Lasserre reste en effet lar-

gement inexploré pour de nombreux problèmes classiques. Par ailleurs, l'application de ces hiérarchies à des problèmes variationnels ou à des modèles probabilistes complexes représente également un axe de recherche prometteur, notamment dans le contexte de l'optimisation globale et de l'apprentissage automatique probabiliste. Concernant la réduction fonctionnelle des ensembles d'arbres de décision, il serait pertinent d'étudier son influence sur d'autres métriques de performance, telles que la robustesse, l'explicabilité ou l'équité des modèles, afin d'en évaluer pleinement le potentiel en apprentissage automatique. L'intégration de ces techniques dans des pipelines de *machine learning* automatisés, ou leur adaptation à d'autres types de modèles, comme les réseaux de neurones ou les modèles graphiques, constitue également une direction de recherche intéressante. Enfin, la nouvelle formulation de la méthode du simplexe amélioré, principalement étudiée ici d'un point de vue théorique, gagnerait à être implémentée et testée en pratique, notamment en combinaison avec des techniques d'accélération existantes comme la méthode du multiphase. Il serait également intéressant d'analyser son impact sur la résolution de problèmes de programmation linéaire en nombres entiers, où le simplexe amélioré sert souvent de point de départ aux algorithmes de résolution. L'ensemble de ces perspectives souligne la richesse des interactions entre optimisation, recherche opérationnelle et apprentissage automatique, et ouvre la voie à de nombreux développements futurs, tant sur le plan théorique que pratique.

En conclusion, les travaux présentés dans cette thèse témoignent du potentiel considérable des méthodes d'optimisation pour aborder des problématiques variées et complexes. Ils invitent à poursuivre l'exploration de ces outils, à la fois pour approfondir la compréhension des phénomènes sous-jacents et pour concevoir des solutions innovantes répondant aux défis actuels et futurs de la recherche opérationnelle et de l'apprentissage automatique.

RÉFÉRENCES

[1] D. Henrion, J.-B. Lasserre et J. Löfberg, "Gloptipoly 3 : moments, optimization and semidefinite programming," *Optimization Methods & Software*, vol. 24, n°. 4-5, p. 761–779, 2009.

[2] M. E. Dyer et A. M. Frieze, "On the complexity of computing the volume of a polyhedron," *SIAM Journal on Computing*, vol. 17, n°. 5, p. 967–974, 1988.

[3] M. Dyer, A. Frieze et R. Kannan, "A random polynomial-time algorithm for approximating the volume of convex bodies," *Journal of the ACM (JACM)*, vol. 38, n°. 1, p. 1–17, 1991.

[4] B. Cousins et S. Vempala, "A practical volume algorithm," *Mathematical Programming Computation*, vol. 8, p. 133–160, 2016.

[5] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[6] D. Henrion, J. B. Lasserre et C. Savorgnan, "Approximate volume and integration for basic semialgebraic sets," *SIAM review*, vol. 51, n°. 4, p. 722–743, 2009.

[7] A. Macdonald, "Stokes' theorem," *Real Analysis Exchange*, vol. 27, n°. 2, p. 739–748, 2002.

[8] M. F. Anjos et J. B. Lasserre, "Introduction to semidefinite, conic and polynomial optimization," *Handbook on semidefinite, conic and polynomial optimization*, p. 1–22, 2012.

[9] L. Lovász et A. Schrijver, "Cones of matrices and set-functions and 0–1 optimization," *SIAM journal on optimization*, vol. 1, n°. 2, p. 166–190, 1991.

[10] M. X. Goemans et D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, n°. 6, p. 1115–1145, 1995.

[11] H. Wolkowicz et M. F. Anjos, "Semidefinite programming for discrete optimization and matrix completion problems," *Discrete Applied Mathematics*, vol. 123, n°. 1-3, p. 513–577, 2002.

[12] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on optimization*, vol. 11, n°. 3, p. 796–817, 2001.

[13] E. De Klerk et D. V. Pasechnik, "Approximation of the stability number of a graph via copositive programming," *SIAM Journal on Optimization*, vol. 12, n°. 4, p. 875–892, 2002.

[14] H. D. Sherali et W. P. Adams, "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems," *SIAM Journal on Discrete Mathematics*, vol. 3, n°. 3, p. 411–430, 1990.

[15] M. Laurent, "A comparison of the sherali-adams, lovász-schrijver, and lasserre relaxations for 0–1 programming," *Mathematics of Operations Research*, vol. 28, n°. 3, p. 470–496, 2003.

[16] L. Grinsztajn, E. Oyallon et G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data ?" *Advances in Neural Information Processing Systems*, 2022.

[17] D. McElfresh, S. Khandagale, J. Valverde, V. Prasad C, G. Ramakrishnan, M. Goldblum et C. White, "When do neural nets outperform boosted trees on tabular data ?" *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[18] G. Biau et E. Scornet, "A random forest guided tour," *Test*, vol. 25, n°. 2, p. 197–227, 2016.

[19] P. Probst et A.-L. Boulesteix, "To tune or not to tune the number of trees in random forest," *Journal of Machine Learning Research*, vol. 18, n°. 1, p. 6673–6690, 2017.

[20] S. Buschjäger et K. Morik, "Improving the accuracy-memory trade-off of random forests via leaf-refinement," *arXiv preprint arXiv:2110.10075*, 2021.

[21] D. D. Margineantu et T. G. Dietterich, "Pruning adaptive boosting," dans *International Conference on Machine Learning*, 1997, p. 211–218.

[22] G. Martínez-Muñoz et A. Suárez, "Pruning in ordered bagging ensembles," dans *International Conference on Machine Learning*, 2006, p. 609–616.

[23] B. Liu et R. Mazumder, "ForestPrune : Compact depth-pruned tree ensembles," dans *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, p. 9417–9428.

[24] A. H. Amee, M. I. Hossain, S. F. Khan, D. M. Farid *et al.*, "A novel pessimistic decision tree pruning approach for classification," dans *International Conference on Electrical Information and Communication Technology*. IEEE, 2023, p. 1–6.

[25] G. B. Dantzig et M. N. Thapa, *The simplex method*. Springer, 1997.

[26] I. Elhallaoui, A. Metrane, G. Desaulniers et F. Soumis, "An improved primal simplex algorithm for degenerate linear programs," *INFORMS Journal on Computing*, vol. 23, n°. 4, p. 569–577, 2011.

[27] J. B. Lasserre, *Moments, positive polynomials and their applications*. World Scientific, 2009, vol. 1.

[28] M. Putinar, "Positive polynomials on compact semi-algebraic sets," *Indiana University Mathematics Journal*, vol. 42, n°. 3, p. 969–984, 1993.

[29] C. Josz et D. K. Molzahn, "Lasserre hierarchy for large scale polynomial optimization in real and complex variables," *SIAM Journal on Optimization*, vol. 28, n°. 2, p. 1017–1048, 2018.

[30] J. B. Lasserre, D. Henrion, C. Prieur et E. Trélat, "Nonlinear optimal control via occupation measures and lmi-relaxations," *SIAM journal on control and optimization*, vol. 47, n°. 4, p. 1643–1666, 2008.

[31] E. Pauwels, D. Henrion et J.-B. Lasserre, "Inverse optimal control with polynomial optimization," dans *53rd IEEE Conference on Decision and Control*. IEEE, 2014, p. 5581–5586.

[32] J. B. Lasserre, "The moment-sos hierarchy : Applications and related topics," *Acta Numerica*, vol. 33, p. 841–908, 2024.

[33] D. Henrion, M. Korda et J. B. Lasserre, *Moment-sos Hierarchy, The : Lectures In Probability, Statistics, Computational Geometry, Control And Nonlinear Pdes*. World Scientific, 2020, vol. 4.

[34] K. Fang et H. Fawzi, "The sum-of-squares hierarchy on the sphere and applications in quantum information theory," *Mathematical Programming*, vol. 190, n°. 1, p. 331–360, 2021.

[35] M. Laurent et L. Slot, "An effective version of schmüdgen's positivstellensatz for the hypercube," *Optimization Letters*, vol. 17, n°. 3, p. 515–530, 2023.

[36] L. Baldi et B. Mourrain, "On the effective putinar's positivstellensatz and moment approximation," *Mathematical Programming*, vol. 200, n°. 1, p. 71–103, 2023.

[37] L. Baldi et L. Slot, "Degree bounds for putinar's positivstellensatz on the hypercube," *SIAM Journal on Applied Algebra and Geometry*, vol. 8, n°. 1, p. 1–25, 2024.

[38] E. L. Lawler, J. K. Lenstra et A. Rinnooy Kan, "Generating all maximal independent sets : Np-hardness and polynomial-time algorithms," *SIAM Journal on Computing*, vol. 9, n°. 3, p. 558–565, 1980.

[39] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, p. 81–106, 1986.

[40] L. Breiman, J. Friedman, R. A. Olshen et C. J. Stone, *Classification and regression trees*. Routledge, 2017.

[41] L. Breiman, "Random forests," *Machine Learning*, vol. 45, n°. 1, p. 5–32, 2001.

[42] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, n°. 4, p. 367–378, 2002.

[43] T. Chen et C. Guestrin, "XGBoost : A scalable tree boosting system," dans *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, p. 785–794.

[44] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye et T.-Y. Liu, "LightGBM : A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[45] G. Tsoumakas, I. Partalas et I. Vlahavas, "An ensemble pruning primer," *Applications of Supervised and Unsupervised Ensemble Methods*, p. 1–13, 2009.

[46] Z.-H. Zhou, *Ensemble methods : Foundations and algorithms*, 1er éd. CRC press, 2012.

[47] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo et M. Xu, "Margin & diversity based ordering ensemble pruning," *Neurocomputing*, vol. 275, p. 237–246, 2018.

[48] Z. Lu, X. Wu, X. Zhu et J. Bongard, "Ensemble pruning via individual contribution ordering," dans *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, p. 871–880.

[49] G. Giacinto, F. Roli et G. Fumera, "Design of effective multiple classifier systems by clustering of classifiers," dans *International Conference on Pattern Recognition*, vol. 2. IEEE, 2000, p. 160–163.

[50] Y. Zhang, S. Burer, W. Nick Street, K. P. Bennett et E. Parrado-Hernández, "Ensemble pruning via semi-definite programming." *Journal of Machine Learning Research*, vol. 7, no. 7, 2006.

[51] N. Li, Y. Yu et Z.-H. Zhou, "Diversity regularized ensemble pruning," dans *Machine Learning and Knowledge Discovery in Databases : European Conference*. Springer, 2012, p. 330–345.

[52] G. D. Cavalcanti, L. S. Oliveira, T. J. Moura et G. V. Carvalho, "Combining diversity measures for ensemble pruning," *Pattern Recognition Letters*, vol. 74, p. 38–45, 2016.

[53] A. Kumar, S. Goyal et M. Varma, "Resource-efficient machine learning in 2 kb ram for the internet of things," dans *International Conference on Machine Learning*. PMLR, 2017, p. 1935–1944.

[54] A. Painsky et S. Rosset, "Lossless compression of random forests," *Journal of Computer Science and Technology*, vol. 34, p. 494–506, 2019.

[55] A. Nakamura et K. Sakurada, "Simplification of forest classifiers and regressors," *arXiv preprint arXiv:2212.07103*, 2022.

[56] H. D. Sherali, A. G. Hobeika et C. Jeenanunta, "An optimal constrained pruning strategy for decision trees," *INFORMS Journal on Computing*, vol. 21, n°. 1, p. 49–61, 2009.

[57] E. Carrizosa, C. Molero-Río et D. Romero Morales, "Mathematical optimization in classification and regression trees," *TOP*, vol. 29, n°. 1, p. 5–33, 2021.

[58] T. Vidal, T. Pacheco et M. Schiffer, "Born-again tree ensembles," dans *International Conference on Machine Learning*. PMLR, 2020, p. 9743–9753.

[59] D. R. Morrison, S. H. Jacobson, J. J. Sauppe et E. C. Sewell, "Branch-and-bound algorithms : A survey of recent advances in searching, branching, and pruning," *Discrete Optimization*, vol. 19, p. 79–102, 2016.

[60] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of applied optimization*, vol. 1, n°. 1, p. 65–77, 2002.

[61] B. Ghojogh, A. Ghodsi, F. Karray et M. Crowley, "Kkt conditions, first-order and second-order optimization, and distributed optimization : tutorial and survey," *arXiv preprint arXiv:2110.01858*, 2021.

[62] D. Avis et V. Chvátal, "Notes on bland's pivoting rule," dans *Polyhedral Combinatorics : Dedicated to the memory of DR Fulkerson*. Springer, 2009, p. 24–34.

[63] J. J. Forrest et D. Goldfarb, "Steepest-edge simplex algorithms for linear programming," *Mathematical programming*, vol. 57, n°. 1, p. 341–374, 1992.

[64] I. Adler, C. Papadimitriou et A. Rubinstein, "On simplex pivoting rules and complexity theory," dans *Integer Programming and Combinatorial Optimization : 17th International Conference, IPCO 2014, Bonn, Germany, June 23-25, 2014. Proceedings 17*. Springer, 2014, p. 13–24.

[65] F. A. Potra et S. J. Wright, "Interior-point methods," *Journal of computational and applied mathematics*, vol. 124, n°. 1-2, p. 281–302, 2000.

[66] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[67] S. Borgwardt et C. Viss, "An implementation of steepest-descent augmentation for linear programs," *Operations Research Letters*, vol. 48, n°. 3, p. 323–328, 2020.

[68] S. Smale, "Mathematical problems for the next century," *The mathematical intelligencer*, vol. 20, n°. 2, p. 7–15, 1998.

[69] H. J. Greenberg, *Design and implementation of optimization software*. Sijthoff & Noordhoff [International Publishers], 1978.

[70] M. Bénichou, J.-M. Gauthier, G. Hentges et G. Ribiere, "The efficient solution of large-scale linear programming problems-some algorithmic techniques and computational results," *Mathematical Programming*, vol. 13, n°. 1, p. 280–322, 1977.

[71] A. Charnes, "Optimality and degeneracy in linear programming," *Econometrica : Journal of the Econometric Society*, p. 160–170, 1952.

[72] A. F. Perold, "A degeneracy exploiting lu factorization for the simplex method," *Mathematical Programming*, vol. 19, n°. 1, p. 239–254, 1980.

[73] P.-Q. Pan, "A primal deficient-basis simplex algorithm for linear programming," *Applied mathematics and computation*, vol. 196, n°. 2, p. 898–912, 2008.

[74] N. Gvozdenović et M. Laurent, "Semidefinite bounds for the stability number of a graph via sums of squares of polynomials," *Mathematical Programming*, vol. 110, p. 145–173, 2007.

[75] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," dans *Emerging applications of algebraic geometry*. Springer, 2009, p. 157–270.

[76] M. Conforti, G. Cornuéjols, G. Zambelli *et al.*, *Integer programming*. Springer, 2014, vol. 271.

[77] F. Silvestri, "A variant of the lov\'asz-theta number based on projection matrices," *arXiv preprint arXiv:1708.06563*, 2017.

[78] T. Hastie, S. Rosset, J. Zhu et H. Zou, "Multi-class AdaBoost," *Statistics and its Interface*, vol. 2, n°. 3, p. 349–360, 2009.

[79] A. Parmentier et T. Vidal, "Optimal counterfactual explanations in tree ensembles," dans *International Conference on Machine Learning*. PMLR, 2021, p. 8422–8431.

[80] A. Kantchelian, J. D. Tygar et A. Joseph, "Evasion and hardening of tree ensemble classifiers," dans *International Conference on Machine Learning*. PMLR, 2016, p. 2387–2396.

[81] T. Serra, A. Kumar et S. Ramalingam, "Lossless compression of deep neural networks," dans *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, 2020, p. 417–430.

[82] X. Yu, T. Serra, S. Ramalingam et S. Zhe, "The combinatorial brain surgeon : Pruning weights that cancel one another in neural networks," dans *International Conference on Machine Learning*. PMLR, 2022, p. 25 668–25 683.

[83] R. Benbaki, W. Chen, X. Meng, H. Hazimeh, N. Ponomareva, Z. Zhao et R. Mazumder, "Fast as CHITA : Neural network pruning with combinatorial optimization," dans *International Conference on Machine Learning*. PMLR, 2023, p. 2031–2049.

[84] X. Meng, S. Ibrahim, K. Behdin, H. Hazimeh, N. Ponomareva et R. Mazumder, "OSSCAR : One-shot structured pruning in vision and language models with combinatorial optimization," dans *International Conference on Machine Learning*, 2024.

[85] H. Karloff, *Linear programming*. Springer Science & Business Media, 2008.

[86] I. Elhallaoui, D. Villeneuve, F. Soumis et G. Desaulniers, "Dynamic aggregation of set-partitioning constraints in column generation," *Operations Research*, vol. 53, n°. 4, p. 632–645, 2005.

[87] I. Elhallaoui, A. Metrane, F. Soumis et G. Desaulniers, "Multi-phase dynamic constraint aggregation for set partitioning type problems," *Mathematical Programming Series A*, 2008.

[88] M. Saddoune, G. Desaulniers, I. Elhallaoui et F. Soumis, "Integrated airline crew scheduling : A bi-dynamic constraint aggregation method using neighborhoods," *European Journal of Operational Research*, 2011.

[89] A. Metrane, F. Soumis et I. Elhallaoui, "Column generation decomposition with the degenerate constraints in the subproblem," *European Journal of Operational Research*, vol. 207, n°. 1, p. 37–44, 2010.

[90] A. Zaghrouti, I. El Hallaoui et F. Soumis, "Improved integral simplex using decomposition for the set partitioning problem," *EURO Journal on Computational Optimization*, vol. 6, n°. 2, p. 185–206, 2018.

[91] S. Rosat, I. Elhallaoui, F. Soumis et D. Chakour, "Influence of the normalization constraint on the integral simplex using decomposition," *Discrete Applied Mathematics*, vol. 217, p. 53–70, 2017, combinatorial Optimization : Theory, Computation, and Applications. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0166218X15005843

[92] J. Desrosiers, J. B. Gauthier et M. E. Lübbecke, "Row-reduced column generation for degenerate master problems," *European Journal of Operational Research*, vol. 236, n°. 2, p. 453–460, 2014.

[93] B. Bollobás, "Volume estimates and rapid mixing," *Flavors of geometry*, vol. 31, p. 151–182, 1997.

[94] B. Cousins et S. Vempala, "A cubic algorithm for computing gaussian volume," dans *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*. SIAM, 2014, p. 1215–1228.

[95] J. B. Lasserre, "Computing gaussian & exponential measures of semi-algebraic sets," *Advances in Applied Mathematics*, vol. 91, p. 137–163, 2017.

[96] N. Dunford et J. T. Schwartz, *Linear operators, part 1 : general theory*. John Wiley & Sons, 1988, vol. 10.

[97] M. Trnovska, "Strong duality conditions in semidefinite programming," *Journal of Electrical Engineering*, vol. 56, n°. 12, p. 1–5, 2005.

[98] J. B. Lasserre, "A new look at nonnegativity on closed sets and polynomial optimization," *SIAM Journal on Optimization*, vol. 21, n°. 3, p. 864–885, 2011.

[99] ——, "Lebesgue decomposition in action via semidefinite relaxations," *Advances in Computational Mathematics*, vol. 42, p. 1129–1148, 2016.

[100] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot et E. Duchesnay, "Scikit-learn : Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, p. 2825–2830, 2011.

ANNEXE A SEMIDEFINITE RELAXATIONS FOR LEBESGUE AND GAUSSIAN MEASURES OF UNIONS OF BASIC SEMIALGEBRAIC SETS

Authors: Youssouf Emine and Jean-Bernard Lasserre

Journal: Mathematics of Operations Research, 2019, Vol. 44, No. 4, pp. 1477-1493

Abstract Given a finite Borel measure μ on $\mathbb{R}^n$ and basic semi-algebraic sets $\mathbf{\Omega}_i \subseteq \mathbb{R}^n$, $i = 1, \ldots, p$, we provide a systematic numerical scheme to approximate as closely as desired $\mu\left(\bigcup_i \mathbf{\Omega}_i\right)$, when all moments of μ are available (and finite). More precisely, we provide a hierarchy of semidefinite programs whose associated sequence of optimal values is monotone and converges to the desired value from above. The same methodology applied to the complement $\mathbb{R}^n \setminus (\bigcup_i \mathbf{\Omega}_i)$ provides a monotone sequence that converges to the desired value from below. When μ is the Lebesgue measure we assume that $\mathbf{\Omega} := \bigcup_i \mathbf{\Omega}_i$ is compact and contained in a known box $\mathbb{B} := [-a, a]^n$ and in this case the complement is taken to be $\mathbb{B} \setminus \mathbf{\Omega}$. In fact, not only $\mu(\mathbf{\Omega})$ but also every finite vector of moments of $\mu_{\mathbf{\Omega}}$ (the restriction of μ on $\mathbf{\Omega}$) can be approximated as closely as desired, and so permits to approximate the integral on $\mathbf{\Omega}$ of any given polynomial.

Keywords: Lebesgue and Gaussian measure; semi-algebraic sets; moment problem and sums of squares; semidefinite programming; convex optimization

MSC: 44A60 28A75 90C05 90C22

A.1 Introduction

Given a set $\mathbf{\Omega} \subset \mathbb{R}^n$ and a finite Borel measure μ on $\mathbb{R}^n$, computing $\mu(\mathbf{\Omega})$ is a very challenging problem. In fact, even approximating the Lebesgue volume of a convex body $\mathbf{\Omega} \subset \mathbb{R}^n$ (e.g., a polytope) is difficult; see, for example, Bollobás [93] and Dyer and Frieze [2]. However, in the latter case, some efficient (non-deterministic) algorithms with probabilistic guarantees are available, and for more details, the interested reader is referred to, for example, Dyer et al. [3], Cousins and Vempala [4, 94], and the references therein. In the non-convex case, no such algorithm is available, and one is left with approximating $\mu(\mathbf{\Omega})$ using Monte Carlo (or Quasi-Monte-Carlo) type methods as described in, for example, Niederreiter [5]. That is, one first generates a sample of N points in $\mathbb{B}$ following the distribution μ on $\mathbb{B}$, and then one *counts* the number $N_{\mathbb{K}}$ of points that fall into $\mathbf{\Omega}$. This realization of the random variable $\frac{N_{\mathbb{K}}}{N}$ provides an estimate of $\mu(\mathbf{\Omega})$, but it is by no means

an upper or lower bound on $\mu(\mathbf{\Omega})$. Of course, this method is quite fast, especially in small dimensions. Yet, as $\mu(\mathbf{\Omega})$ is indeed very difficult to compute exactly, a less ambitious but still useful goal would be to provide *upper* and/or *lower bounds* on $\mu(\mathbf{\Omega})$. Even better, a converging sequence of upper (or lower) bounds would be highly desirable. This is the strategy proposed in Henrion et al. [6] when $\mathbf{\Omega}$ is a compact basic semi-algebraic set and μ is the Lebesgue measure. In [6], the authors have provided a (deterministic) numerical scheme that yields a monotone sequence of upper bounds converging to $\mu(\mathbf{\Omega})$. It consists of solving a hierarchy of semidefinite programs of increasing size. By repeating the procedure, but now with the complement $\mathbb{B} \setminus \mathbf{\Omega}$, one also obtains a monotone sequence of lower bounds converging to $\mu(\mathbf{\Omega})$. However, even on typical $2-$ or $3-$dimensional examples, the convergence was rather slow, and the authors proposed a slight modification which turned out to be much more efficient; the convergence was much faster but, unfortunately, no longer monotone.

A.1.1 Contribution

The purpose of this paper is to introduce a deterministic method to approximate (in principle as closely as desired) the measure $\mu(\mathbf{\Omega})$ of the union $\mathbf{\Omega} = \bigcup_i \mathbf{\Omega}_i$ of finitely many basic semi-algebraic sets. The finite Borel measure μ is any measure whose moments are all finite, e.g., the Lebesgue measure when $\mathbf{\Omega}$ is compact, or the Gaussian measure $d\mu = \exp(-\|\mathbf{x}\|^2)\, d\mathbf{x}$ for non-compact sets $\mathbf{\Omega}$.

The method is similar in spirit to the one in [6] for a compact basic semi-algebraic set and the one in [95] for computing Gaussian measures of basic closed semi-algebraic sets (not necessarily compact), but with two important novelties.

- In contrast to [6] and [95], we consider a finite *union* $\mathbf{\Omega}$ of (non-disjoint) basic semi-algebraic sets, which complicates matters significantly.
- We include a technique to accelerate the convergence, different from the one described in [6]. Indeed, in contrast to [6], it has the highly desirable feature of maintaining the *monotone* convergence to $\mu(\mathbf{\Omega})$, which is essential if one wishes to obtain upper and lower bounds. This technique consists of using moment constraints derived from a particular application of Stokes' theorem.

In fact, this numerical scheme allows the approximation of not only $\mu(\mathbf{\Omega})$, but also any fixed finite sequence of moments of the measure $\mu_{\mathbf{\Omega}}$ (where $\mu_{\mathbf{\Omega}}$ is the restriction of μ to $\mathbf{\Omega}$).

Remark .1. One might invoke the inclusion-exclusion principle, which states that

$$\mu\left(\bigcup_{i=1}^{p}\mathbf{\Omega}_i\right)=\sum_{j=1}^{p}(-1)^{j+1}\sum_{1\le i_1<\cdots<i_j\le p}\mu\left(\mathbf{\Omega}_{i_1}\cap\cdots\cap\mathbf{\Omega}_{i_j}\right), \tag{A.1}$$

so that in principle it suffices to compute (or approximate) $\mu\left(\mathbf{\Omega}_{i_1}\cap\cdots\cap\mathbf{\Omega}_{i_j}\right)$ for all possible intersections of the $\mathbf{\Omega}_j$'s, e.g., by the approach of [6] or [95]. But this approach has two major drawbacks. First, there are possibly 2^p such sets, and secondly, to compute an upper bound, one has to compute an upper bound for such intersections with an odd number of elementary sets $\mathbf{\Omega}_{i_j}$, and a lower bound for such intersections with an even number of elementary sets. The latter lower bound, in turn, is obtained by computing an upper bound for the complement. This makes the whole procedure tedious and complicated.

Finally, Bonferroni's inequalities also provide a (finite) sequence of upper and lower bounds on $\mu(\mathbf{\Omega})$, but computing those bounds involves sums similar to the right-hand side of (A.1), hence with the same drawbacks just mentioned. Our proposed technique is direct, with no partial computation on intersections of elementary sets $\mathbf{\Omega}_{i_j}$.

Of course, the technique described in this paper is computationally expensive. In particular, its applicability is limited by the performance of state-of-the-art semidefinite solvers because the size of the semidefinite programs increases quickly with the rank in the hierarchy. Therefore, it is limited to small-dimensional problems ($n \le 3, 4$). For higher dimensions, only a few steps in the hierarchy can be implemented, and therefore only upper and lower bounds (possibly crude) are expected. But the reader should keep in mind that the problem is very difficult, and to the best of our knowledge, we are not aware of an algorithm (at least at this level of generality) that provides certified upper and lower bounds with such convergence properties (even for convex sets, and in particular for non-compact sets $\mathbf{\Omega}$). In our opinion, this methodology should be viewed as complementary to (rather than competing with) probabilistic methods.

A.2 Notations, definitions and preliminary results

A.2.1 Notations and definitions

Let $\mathbb{R}[\mathbf{x}]$ be the ring of polynomials in the variables $\mathbf{x} = (x_1, \ldots, x_n)$. Denote by $\mathbb{R}[\mathbf{x}]_d \subset \mathbb{R}[\mathbf{x}]$ the vector space of polynomials of degree at most d, which has dimension $s(d) := \binom{n+d}{d}$, with the usual canonical basis $(\mathbf{x}^\gamma)_{\gamma\in\mathbb{N}^n_d}$ of monomials, where $\mathbb{N}^n_d := \{\gamma \in \mathbb{N}^n \mid |\gamma| \le d\}$, $\mathbb{N}$ is the set of natural numbers including 0, and $|\gamma| := \sum_{i=1}^n \gamma_i$.

Also, denote by $\Sigma[\mathbf{x}] \subset \mathbb{R}[\mathbf{x}]$ (resp. $\Sigma[\mathbf{x}]_d \subset \mathbb{R}[\mathbf{x}]_{2d}$) the cone of sums of squares (s.o.s.) polynomials (resp. s.o.s. polynomials of degree at most $2d$). If $f \in \mathbb{R}[\mathbf{x}]_d$, we write $f(\mathbf{x}) = \sum_{\gamma \in \mathbb{N}^n_d} f_\gamma \mathbf{x}^\gamma$ in the canonical basis and denote by $\mathbf{f} = (f_\gamma)_\gamma \in \mathbb{R}^{s(d)}$ its vector of coefficients.

Finally, let S^n denote the space of $n \times n$ real symmetric matrices, with inner product $\langle \mathbf{A}, \mathbf{B} \rangle = \operatorname{trace}(\mathbf{AB})$. We use the notation $\mathbf{A} \succeq 0$ (resp. $\mathbf{A} \succ 0$) to denote that $\mathbf{A}$ is positive semidefinite (definite). With $g_0 \coloneqq 1$, the quadratic module $Q(g_1, \ldots, g_m) \subset \mathbb{R}[\mathbf{x}]$ generated by polynomials $g_1, \ldots, g_m$ is defined by:

$$Q(g_1, \ldots, g_m) \coloneqq \left\{ \sum_{j=0}^{m} \sigma_j g_j \colon \sigma_j \in \Sigma[\mathbf{x}] \right\}.$$

Definition .2 (Archimedean Assumption)**.** The quadratic module $Q(g_1, \ldots, g_m)$ is said to be *Archimedean* if there exists $M > 0$ such that the quadratic polynomial

$$\mathbf{x} \mapsto g_{m+1}(\mathbf{x}) \coloneqq M - \|\mathbf{x}\|^2$$

belongs to $Q(g_1, \ldots, g_m)$. Notice that $g_{m+1} \in Q(g_1, \ldots, g_m)$ is an *algebraic certificate* that the set $\mathcal{K} \coloneqq \{\mathbf{x} \colon g_j(\mathbf{x}) \geq 0, j = 1, \ldots, m\}$ is compact.

If the set $\mathcal{K} \coloneqq \{\mathbf{x} \colon g_j(\mathbf{x}) \geq 0, j = 1, \ldots, m\}$ is compact, then $\|\mathbf{x}\|^2 \leq M$ for some $M > 0$, and one may always include the redundant quadratic constraint $\theta(\mathbf{x}) \coloneqq M - \|\mathbf{x}\|^2 \geq 0$ in the definition of $\mathcal{K}$ without changing $\mathcal{K}$. Then the quadratic module $Q(g_1, \ldots, g_{m+1})$ is Archimedean.

Moment and localizing matrices With a real sequence $\mathbf{y} = (y_\gamma)_{\gamma \in \mathbb{N}^n_d}$, one may associate the (Riesz) linear functional $L_{\mathbf{y}} \colon \mathbb{R}[\mathbf{x}]_d \to \mathbb{R}$ defined by

$$f \left(= \sum_\gamma f_\gamma \mathbf{x}^\gamma \right) \mapsto L_{\mathbf{y}}(f) \coloneqq \sum_\gamma f_\gamma y_\gamma,$$

Denote by $\mathbf{M}_d(\mathbf{y})$ the *moment* matrix associated with $\mathbf{y}$, the real symmetric matrix with rows and columns indexed in the basis of monomials $(\mathbf{x}^\gamma)_{\gamma \in \mathbb{N}^n_d}$, and with entries:

$$\mathbf{M}_d(\mathbf{y})(\alpha, \beta) \coloneqq L_{\mathbf{y}}\left(\mathbf{x}^{\alpha+\beta}\right) = y_{\alpha+\beta}, \qquad \forall\, \alpha, \beta \in \mathbb{N}^n_d.$$

Next, given $g \in \mathbb{R}[\mathbf{x}]$, denote by $\mathbf{M}_d(g\mathbf{y})$ the *localizing* moment matrix associated with $\mathbf{y}$ and g, the real symmetric matrix with rows and columns indexed in the basis of monomials

$(\mathbf{x}^\gamma)_{\gamma\in\mathbb{N}^n_d}$, and with entries:

$$\mathbf{M}_d(g\mathbf{y})(\alpha,\beta) := L_{\mathbf{y}}\left(g(\mathbf{x})\mathbf{x}^{\alpha+\beta}\right) = \sum_\gamma g_\gamma y_{\alpha+\beta+\gamma}, \qquad \forall\,\alpha,\beta\in\mathbb{N}^n_d.$$

If $\mathbf{y}=(y_\gamma)_{\gamma\in\mathbb{N}^n}$ is the sequence of moments of some Borel measure μ on $\mathbb{R}^n$, then $\mathbf{M}_d(\mathbf{y})\succeq 0$ for all $d\in\mathbb{N}$. However, the converse is not true in general and is related to the well-known fact that there are positive polynomials that are not sums of squares. Similarly, if the support of μ is contained in $\{\mathbf{x}\colon g(\mathbf{x})\geq 0\}$, then $\mathbf{M}_d(g\mathbf{y})\succeq 0$ for all $d\in\mathbb{N}$.

For more details, the interested reader is referred to e.g. [27, Chapter 3].

Given a Borel set $\mathbf{\Omega}\subset\mathbb{R}^n$, let $\mathcal{M}(\mathbf{\Omega})$ be the space of finite *signed* Borel measures on $\mathbf{\Omega}$, and let $\mathcal{M}(\mathbf{\Omega})_+\subset\mathcal{M}(\mathbf{\Omega})$ be the convex cone of finite Borel measures on $\mathbf{\Omega}$.

A.2.2 The measure of a basic semi-algebraic set

Let $\mathbf{B},\mathcal{K}\subset\mathbb{R}^n$ with $\mathbf{B}\supset\mathcal{K}$ and let μ be a finite Borel measure whose support is $\mathbf{B}$. (Typically, μ is the Lebesgue measure on a box $\mathbf{B}$, and one wishes to compute the Lebesgue volume $\mathrm{vol}(\mathcal{K})$; alternatively, $\mathbf{B}=\mathbb{R}^n$, μ is the Gaussian measure $d\mu=\exp\left(-\|\mathbf{x}\|^2\right)d\mathbf{x}$, and one wishes to compute $\mu(\mathcal{K})$.)

A.2.3 An infinite-dimensional linear program P

Let $f\in\mathbb{R}[\mathbf{x}]$ be positive almost everywhere on $\mathcal{K}$, and consider the following infinite-dimensional **LP** problem:

$$\mathbf{P}\colon \quad f^* = \sup_\phi\left\{\int_{\mathcal{K}} f\,d\phi\colon \lambda\leq\mu,\ \phi\in\mathcal{M}(\mathcal{K})_+\right\}$$

Theorem .1 ([6]). *The measure $\phi^*=\mu_{\mathcal{K}}$ (the restriction of μ to $\mathcal{K}$) is the unique optimal solution of* **P**. *In particular, if $f(\mathbf{x})=1$ for all $\mathbf{x}$, then $f^*=\mu(\mathcal{K})$.*

A.2.4 Semidefinite relaxations

Of course, problem **P** in Theorem .1 is infinite-dimensional and cannot be solved directly. However, when $\mathcal{K}$ is a basic semi-algebraic set, Theorem .1 can be further exploited. So, given $(g_j)_{j=1}^m\subset\mathbb{R}[\mathbf{x}]$, let $\mathcal{K}\subset\mathbb{R}^n$ be the basic semi-algebraic set:

$$\mathcal{K} := \{\mathbf{x}\in\mathbb{R}^n\colon g_j(\mathbf{x})\geq 0,\ j=1,\ldots,m\}, \tag{A.2}$$

assumed to be nonempty and compact. Let $\mathbf{B} \supset \mathcal{K}$, and let μ be a finite Borel measure whose all moments $\mathbf{z} = (z_\alpha)$ with

$$z_\alpha := \int_{\mathbf{B}} \mathbf{x}^\alpha \, d\mu(\mathbf{x}), \qquad \alpha \in \mathbb{N}^n,$$

are available in closed form or can be computed. To approximate f^* as closely as desired, the authors in [6] propose solving the following hierarchy $(\mathbf{Q}_d)_{d\in\mathbb{N}}$ of semidefinite programs [1] indexed by $d \in \mathbb{N}$:

$$\mathbf{Q}_d\colon \sup_{\mathbf{y}} \left\{ L_{\mathbf{y}}(f)\colon \mathbf{M}_d(\mathbf{y}) \succeq 0,\ \mathbf{M}_d(\mathbf{z}-\mathbf{y}) \succeq 0,\ \mathbf{M}_{d-r_j}(g_j\mathbf{y}) \succeq 0,\ j = 1,\ldots,m \right\}. \tag{A.3}$$

Observe that $\mathbf{Q}_d$ is a relaxation of $\mathbf{P}$, and so $\rho_d \geq \mu(\mathcal{K})$ for all d. In addition, the sequence $(\rho_d)_{d\in\mathbb{N}}$ is monotone non-increasing. The dual of (A.3) is the semidefinite program:

$$\mathbf{Q}_d^*\colon \rho_d^* = \inf_{p\in\mathbb{R}[\mathbf{x}]_{2d}} \left\{ \int_{\mathbf{B}} p\, d\mu\colon p - f \geq 0 \text{ on } \mathcal{K},\ p \in \Sigma[\mathbf{x}]_d \right\}, \tag{A.4}$$

and by weak duality, $\rho_d \leq \rho_d^* \leq f^*$ for all d.

Theorem .2 ([6]). *Assume that $Q(g_1,\ldots,g_m)$ is Archimedean. Then $\rho_d \to f^*$ as $d \to \infty$. If $\mathcal{K}$ has nonempty interior, then $\rho_d^* = \rho_d$, and (A.4) has an optimal solution $p^* \in \mathbb{R}[\mathbf{x}]_{2d}$.*

So when $f = 1$, $(\rho_d)_{d\in\mathbb{N}}$ provides us with a monotone sequence of upper bounds on $f^* = \mu(\mathcal{K})$. Unfortunately, the convergence is rather slow, as observed in several numerical examples. This is because, in the dual (A.4), the optimal solution $p^* \in \mathbb{R}[\mathbf{x}]_{2d}$ tries to approximate from above (in $L_1(\mathbf{B},\mu)$) the discontinuous function $1_{\mathcal{K}}$, which implies an annoying Gibbs' phenomenon [2]. To remedy this problem, the authors in [6] propose using a polynomial f, nonnegative on $\mathcal{K}$, that vanishes on $\partial\mathcal{K}$. In this case, the convergence $\rho_d \to \int_{\mathcal{K}} f\, d\mu$ as $d \to \infty$ is still monotone, and if $\mathbf{y}^d = (y_\alpha^d)_{\alpha\in\mathbb{N}^n_{2d}}$ denotes an optimal solution of (A.3), then $y_0^d \to \mu(\mathcal{K})$ as $d \to \infty$. However, while faster than with $f = 1$, the latter convergence of y_0^d to $\mu(\mathcal{K})$ is not monotone anymore, a rather annoying feature that prevents obtaining a non-increasing sequence of upper bounds.

1. A semidefinite program (SDP) is a conic convex optimization problem with a remarkable modeling power. It can be solved efficiently (in time polynomial in its input size) up to arbitrary precision fixed in advance; see e.g. Anjos and Lasserre [8]

2. The Gibbs' phenomenon appears at a jump discontinuity when one approximates a piecewise C^1 function with a continuous function, e.g. by its Fourier series.

A.3 Main results

The context. Let $\mathbf{B} \subset \mathbb{R}^n$ be a box, and for every $i = 1, \ldots, p$, let

$$\mathbf{\Omega}_i \coloneqq \{\mathbf{x} \in \mathbb{R}^n \colon g_{ij}(\mathbf{x}) \geq 0,\, j = 1, \ldots, m_i\}\,,$$

for some polynomials $(g_{ij}) \subset \mathbb{R}[\mathbf{x}]$. Assume that $\mathbf{B}$ has been chosen so as to satisfy:

$$\mathbf{\Omega} \coloneqq \bigcup_{i=1}^{p} \mathbf{\Omega}_i \subset \mathbf{B}. \tag{A.5}$$

The goal is to provide a numerical scheme to approximate as closely as desired the Lebesgue volume $\mu(\mathcal{K})$. (We will see how to adapt the methodology to also approximate as closely as desired $\mu(\mathcal{K})$ when $\mathcal{K}$ is not necessarily compact and μ is a Gaussian measure.)

One possible approach described below is to use the powerful inclusion-exclusion principle and/or the associated Bonferroni inequalities.

The inclusion-exclusion principle and Bonferroni inequalities. Let:

$$S_k \coloneqq \sum_{1 \leq i_1 < \cdots < i_k \leq p} \mu\left(\mathbf{\Omega}_{i_1} \cap \cdots \cap \mathbf{\Omega}_{i_k}\right), \quad k = 1, \ldots, p.$$

By the inclusion-exclusion principle,

$$\mu\left(\bigcup_{k=1}^{p} \mathbf{\Omega}_k\right) = \sum_{k=1}^{p} (-1)^{k+1} S_k,$$

which allows us to work with intersections of the $\mathbf{\Omega}_k$'s only. In addition, the Bonferroni inequalities state that

$$\mu\left(\bigcup_{i=1}^{p} \mathbf{\Omega}_i\right) \leq \sum_{j=1}^{2k+1} (-1)^{j+1} S_j \quad \forall\, 2k+1 \leq p$$

and

$$\mu\left(\bigcup_{i=1}^{p} \mathbf{\Omega}_i\right) \geq \sum_{j=1}^{2k} (-1)^{j+1} S_j \quad \forall\, 2k \leq p,$$

which provides sequences of (increasingly tighter) upper and lower bounds.

Therefore, to compute $\mu(\mathbf{\Omega})$, we only have to compute the measure of the intersection

$$\Theta_{i_1,\ldots,i_k} := \bigcap_{j=1}^{k} \mathbf{\Omega}_{i_j}, \quad \text{for all } 1 \le i_1 < \cdots < i_k \le p.$$

Notice that there are 2^p such sets. As each $\Theta_{i_1,\ldots,i_k} \subset \mathbf{B}$ is a compact basic semi-algebraic set, one may apply the methodology described in [6], to obtain a sequence $\left(\rho_d^{(i_1,\ldots,i_k)}\right)_{d\in\mathbb{N}}$ which converges to $\mu(\Theta_{i_1,\ldots,i_k})$ as $d \to \infty$, and therefore:

$$\lim_{d\to\infty} \left(\sum_{i=1}^{p} (-1)^{k+1} \sum_{1\le i_1<\cdots<i_k\le p} \rho_d^{i_1,\ldots,i_k} \right) = \mu(\mathbf{\Omega}).$$

Notice that the convergence is not monotone non-increasing even if one solves (A.3) with $f = 1$ because we sum up negative and positive terms. To maintain monotone convergence (when $f = 1$), it suffices to compute a lower bound on the complement $\mathbf{B} \setminus \Theta_{i_1,\ldots,i_k}$ when k is even. However, as already mentioned, the convergence is expected to be rather slow.

To accelerate the convergence, one may use $f = \prod_{j=1}^{k} \prod_{\ell=1}^{m_{i_j}} g_{i_j\ell}$ when solving (A.3) with $\mathbf{\Omega} = \Theta_{i_1,\ldots,i_k}$, as $f \ge 0$ on $\Theta_{i_1,\ldots,i_k}$ and $f = 0$ on $\partial\Theta_{i_1,\ldots,i_k}$.

But then the convergence

$$\lim_{d\to\infty} \sum_{i=1}^{p} (-1)^{k+1} \sum_{1\le i_1<\cdots<i_k\le p} y_{d,0}^{i_1,\ldots,i_k} = \mu(\mathbf{\Omega}), \tag{A.6}$$

(where $\mathbf{y}_d^{i_1,\ldots,i_k} = y_{d,\alpha}^{i_1,\ldots,i_k}$ is an optimal solution of (A.3) with $\mathbf{\Omega} = \Theta_{i_1,\ldots,i_k}$) is not monotone anymore.

A.3.1 A direct approach

In this section, we describe a direct approach with two distinguishing features:

- It does not use the inclusion-exclusion principle and the need to approximate

$$\mu\left(\bigcap_{j=1}^{k} \mathbf{\Omega}_{i_j}\right)$$

  for all 2^p such sets.
- The convergence to $\mu(\mathbf{\Omega})$ (and also to $\mu(\mathbf{B} \setminus \mathbf{\Omega})$) is monotone non-increasing, that is,

we can compute two sequences $(\overline{\omega}_d)_{d\in\mathbb{N}}$ and $(\underline{\omega}_d)_{d\in\mathbb{N}}$ such that:

$$\underline{\omega}_d \le \mu(\mathbf{\Omega}) \le \overline{\omega}_d, \quad d\in\mathbb{N}, \quad \mu(\mathbf{\Omega}) = \lim_{d\to\infty} \underline{\omega}_d = \lim_{d\to\infty} \overline{\omega}_d.$$

Recall that any finite number of moments

$$\mu_\alpha = \int_{\mathbb{B}} \mathbf{x}^\alpha d\mu(\mathbf{x}), \quad \alpha\in\mathbb{N}^n,$$

are either available in closed form or can be obtained numerically.

A multi infinite-dimensional linear program Q

As in [6], we first introduce an infinite-dimensional **LP** problem **Q**, whose unique optimal solution is the restriction of μ on $\mathbf{\Omega}$ (and whose dual has a clear interpretation).

Let $f\in\mathbb{R}[\mathbf{x}]$ be positive almost everywhere on $\mathbf{\Omega}$ and consider the following infinite-dimensional **LP** problem:

$$\mathbf{Q}: \quad f^* = \sup_{\phi_1,\ldots,\phi_p} \left\{ \sum_{i=1}^p \int_{\mathbf{\Omega}_i} f d\phi_i : \sum_{i=1}^p \phi_i \le \mu,\ \phi_i\in\mathcal{M}(\mathbf{\Omega}_i)_+,\ i=1,\ldots,p \right\}. \tag{A.7}$$

Theorem .3. *Problem* **Q** *has an optimal solution* $(\phi_1^*,\ldots,\phi_p^*)$*, and every optimal solution satisfies* $\sum_{i=1}^p \phi_i^* = \mu_{\mathbf{\Omega}}$*, where* $\mu_{\mathbf{\Omega}}$ *is the restriction of* μ *to* $\mathbf{\Omega}$*.*

Proof. We first prove that **Q** has an optimal solution. The set $\Delta_\mu := \{\phi\in\mathcal{M}(\mathbb{R}^n)_+ : \phi\le\mu\}$ is weakly sequentially compact; see e.g. Dunford & Schwartz [96, Theorem 1, p. 305]. Therefore, let $(\phi_1^k,\ldots,\phi_p^k)_{k\in\mathbb{N}}$ be a maximizing sequence of feasible solutions of **Q**. There exists a subsequence $(k_\ell)_{\ell\in\mathbb{N}}$ such that for every $i=1,\ldots,p$, $\phi_i^{k_\ell} \xrightarrow{w} \phi_i^*$ for some $\phi_i^*\in\mathcal{M}(\mathbb{R}^n)_+$. The above weak convergence and $\int_{\mathbf{\Omega}_i^c} d\phi_i^{k_\ell} = 0$ imply $\int_{\mathbf{\Omega}_i^c} d\phi_i^* = 0$, that is, $\phi_i^*\in\mathcal{M}(\mathbf{\Omega}_i)_+$ for all $i=1,\ldots,p$. Weak convergence again implies $\sum_{i=1}^p \phi_i^* \le \mu$, and so $(\phi_1^*,\ldots,\phi_p^*)$ is a feasible solution of **Q**. Finally, weak convergence also implies:

$$f^* = \lim_{\ell\to\infty} \sum_{i=1}^p \int f\, d\phi_i^{k_\ell} = \sum_{i=1}^p \lim_{\ell\to\infty} \int f\, d\phi_i^{k_\ell} = \sum_{i=1}^p \int f\, d\phi_i^*,$$

which proves that $(\phi_1^*,\ldots,\phi_p^*)$ is an optimal solution of **Q**.

We next prove that $f^* = \int f d\mu_{\mathbf{\Omega}}$. Indeed, first observe that for every feasible solution

$(\phi_1, \ldots, \phi_p)$ of $\mathbf{Q}$,

$$\sum_{i=1}^{p} \int f d\phi_i \leq \int_{\mathbf{\Omega}} f d\mu = \int f d\mu_{\mathbf{\Omega}}.$$

On the other hand, for every $i = 1, \ldots, p$, denote by θ_i the measurable function defined on $\mathbf{\Omega}$ by:

$$\mathbf{x} \mapsto \theta_i(\mathbf{x}) = \frac{1}{|\{j \in \{1, \ldots, p\} \colon \mathbf{x} \in \mathbf{\Omega}_j\}|} 1_{\mathbf{\Omega}_i}(\mathbf{x}), \quad \mathbf{x} \in \mathbf{\Omega}.$$

The (discontinuous) functions $(\theta_i)_{i=1}^p$ form a *partition of unity* subordinate to the open cover $\bigcup_i \operatorname{int}(\mathbf{\Omega}_i)$. For every $i = 1, \ldots, p$, let $\phi_i^* \in \mathcal{M}(\mathbf{\Omega}_i)_+$ be the finite Borel measure defined by:

$$\phi_i^*(C) \coloneqq \int_C \theta_i(\mathbf{x}) d\mu(\mathbf{x}), \quad \forall C \in \mathcal{B}(\mathbb{R}^n). \tag{A.8}$$

Hence,

$$\sum_{i=1}^{p} \phi_i^*(C) = \int_C \sum_{i=1}^{p} \theta_i(\mathbf{x}) d\mu(\mathbf{x}) = \int_C 1_{\mathbf{\Omega}}(\mathbf{x}) d\mu(\mathbf{x}) = \mu_{\mathbf{\Omega}}(C) \leq \mu(C).$$

Therefore, $(\phi_1^*, \ldots, \phi_p^*)$ is a feasible solution of $\mathbf{Q}$ such that $\sum_{i=1}^{p} \phi_i^* = \mu_{\mathbf{\Omega}}$, and so

$$\sum_{i=1}^{p} \int f d\phi_i^* = f^*,$$

i.e., $(\phi_1^*, \ldots, \phi_p^*)$ is an optimal solution of $\mathbf{Q}$. In fact, *every* optimal solution $(\phi_1^*, \ldots, \phi_p^*)$ of $\mathbf{Q}$ satisfies

$$\sum_{i=1}^{p} \int f d\phi_i^* = f^*,$$

and therefore $\phi^* \coloneqq \sum_{i=1}^{p} \phi_i^* \in \mathcal{M}(\mathbf{\Omega})_+$ is an optimal solution of

$$\sup_{\phi} \left\{ \int f d\phi \colon \phi \leq \mu, \, \phi \in \mathcal{M}(\mathbf{\Omega})_+ \right\}.$$

By Theorem .1, this solution ϕ^* is unique, which yields the desired result. □

A.3.2 A hierarchy of semidefinite programs

Let $\mathbf{z} = (z_\alpha)_{\alpha \in \mathbb{N}^n}$ be the sequence of all moments of μ, that is,

$$z_\alpha \coloneqq \int \mathbf{x}^\alpha \, d\mu(\mathbf{x}), \quad \alpha \in \mathbb{N}^n. \tag{A.9}$$

Let $\mathbf{B} \subset \mathbb{R}^n$ be a box, and $\mathbf{\Omega} \subset \mathbf{B}$ be a compact semi-algebraic set as in (A.5). With no loss of generality and possibly after scaling, we may and will assume that $\mathbf{B} \subset [-1,1]^n$, and μ is a probability measure. Therefore, $|z_\alpha| \le 1$ for all $\alpha \in \mathbb{N}^n$.

Let $r_{ij} = \lceil \deg(g_{ij})/2 \rceil$, and let $f \in \mathbb{R}[\mathbf{x}]$ be a given polynomial positive almost everywhere on $\mathbf{\Omega}$ (and define $r_{00} := \lceil \deg(f)/2 \rceil$). For $d \ge d_0 := \max_{i,j} r_{ij}$, consider the following hierarchy of semidefinite programs $(\mathbf{Q}_d)$ indexed by $d \in \mathbb{N}$:

$$\mathbf{Q}_d\colon \rho_d^f = \sup_{\mathbf{y}^1,\ldots,\mathbf{y}^p} \left\{ \sum_{i=1}^p L_{\mathbf{y}^i}(f)\colon \begin{array}{l} \mathbf{M}_d\left(\mathbf{z} - \sum_{i=1}^p \mathbf{y}^i\right) \succeq 0;\ \mathbf{M}_d(\mathbf{y}^i) \succeq 0, \quad i = 1,\ldots,p \\ \mathbf{M}_{d-r_{ij}}\left(g_{ij}\mathbf{y}^i\right) \succeq 0, \quad j = 1,\ldots,m_i;\ i = 1,\ldots,p \end{array} \right\}, \tag{A.10}$$

Observe that $\rho_d^f \ge f^*$ for all $d \in \mathbb{N}$. Indeed, if $(\mathbf{z}^1,\ldots,\mathbf{z}^p)$ is the sequence of moments of an optimal solution $(\phi_1^*,\ldots,\phi_p^*)$ of $\mathbf{Q}$ in (A.7), then $(\mathbf{z}^1,\ldots,\mathbf{z}^p)$ is also a feasible solution of $\mathbf{Q}_d$.

Theorem .4. *Consider the semidefinite programs $(\mathbf{Q}_d)$, $d \ge d_0$. Then:*

(i) $\mathbf{Q}_d$ has an optimal solution, and the associated sequence of optimal values $(\rho_d^f)_{d\in\mathbb{N}}$ is monotone non-increasing and converges to f^*, that is:

$$\rho_d^f \downarrow f^* = \int_{\mathbf{\Omega}} f\, d\mu, \quad \text{as } d \to \infty.$$

(ii) Let $(\mathbf{y}^{1,d},\ldots,\mathbf{y}^{p,d})$ be an optimal solution of $\mathbf{Q}_d$. Then for each $\alpha \in \mathbb{N}^n$:

$$\lim_{d\to\infty} \sum_{i=1}^p y_\alpha^{i,d} = z_\alpha^* = \int_{\mathbf{\Omega}} \mathbf{x}^\alpha\, d\mu, \tag{A.11}$$

and in particular,

$$\lim_{d\to\infty} \sum_{i=1}^p y_0^{i,d} = \mu(\mathbf{\Omega}).$$

Proof. For a sequence $\mathbf{y} = (y_\alpha)$, let $\tau_d(\mathbf{y}) = \max_{i=1,\ldots,n} L_{\mathbf{y}}(x_i^{2d})$, and recall that if $\mathbf{M}_d(\mathbf{y}) \succeq 0$, then $|y_\alpha| \le \max\left[y_0, \max_i L_{\mathbf{y}}(x_i^{2d})\right]$ for every $\alpha \in \mathbb{N}^n_{2d}$; see [27, Proposition 3.6].

Next, observe that from $\mathbf{M}_d\left(\mathbf{z} - \sum_{i=1}^p \mathbf{y}^{i,d}\right) \succeq 0$ and $\mathbf{M}_d(\mathbf{y}^i) \succeq 0$,

$$\mathbf{M}_d\left(\mathbf{z} - \mathbf{y}^{i,d}\right) \succeq \mathbf{M}_d\left(\sum_{j\ne i} \mathbf{y}^j\right) \succeq 0, \quad i = 1,\ldots,n.$$

Hence, the diagonal elements $z_{2\alpha} - y_{2\alpha}^{i,d}$ are all non-negative, which in turn implies $\tau_d(\mathbf{y}^{i,d}) \le$

$\tau_d(\mathbf{z}) \leq 1$ for all $i = 1, \ldots, n$. As $z_0 = 1$, then by [27, Proposition 3.6], $\left|y_\alpha^{i,d}\right| \leq 1$ for every $\alpha \in \mathbb{N}^n_{2d}$, and so the feasible set of the semidefinite program $\mathbf{Q}_d$ is closed, bounded, and hence compact. Therefore, $\mathbf{Q}_d$ has an optimal solution.

Next, let $(\mathbf{y}^{1,d}, \ldots, \mathbf{y}^{p,d})$ be an optimal solution of $\mathbf{Q}_d$, and by completing with zeros, make $(\mathbf{y}^{1,d}, \ldots, \mathbf{y}^{p,d})$ an element of the unit ball of $(\ell_\infty)^p$ (where ℓ_∞ is the Banach space of bounded sequences, equipped with the sup-norm). As $(\ell_\infty)^p$ is the topological dual of $(\ell_1)^p$, by the Banach-Alaoglu Theorem, there exists $(\mathbf{y}^{1,*}, \ldots, \mathbf{y}^{p,*}) \in (\ell_\infty)^p$ and a subsequence $\{d_k\}$ such that $(\mathbf{y}^{1,d_k}, \ldots, \mathbf{y}^{p,d_k}) \to (\mathbf{y}^{1,*}, \ldots, \mathbf{y}^{p,*})$ as $k \to \infty$, for the weak $\star$ topology $\sigma((\ell_\infty)^p, (\ell_1)^p)$. In particular,

$$\lim_{k\to\infty} y_\alpha^{i,d_k} = y_\alpha^{i,*}, \quad \forall \alpha \in \mathbb{N}^n, \forall i \in \{1, \ldots, p\}.$$

Next, let $d \in \mathbb{N}$ be fixed arbitrarily. From the pointwise convergence above, we also obtain $\mathbf{M}_d(\mathbf{y}^{i,*}) \succeq 0$ and $\mathbf{M}_d\left(\mathbf{z} - \sum_{i=1}^p \mathbf{y}^{i,*}\right) \succeq 0$ for every $i = 1, \ldots, p$. Similarly, $\mathbf{M}_{d-r_{ij}}\left(g_{ij}\mathbf{y}^{i,*}\right) \succeq 0$ for every i and j.

As d was arbitrary, by Putinar's Positivstellensatz [28], $\mathbf{y}^{i,*}$ has a representing measure ϕ_i supported on $\mathbf{\Omega}_i$ for all $i = 1, \ldots, p$, and $\sum_{i=1}^p \phi_i \leq \mu$. In particular, as $k \to \infty$,

$$f^* \leq \rho_{d_k}^f = \sum_{i=1}^p L_{\mathbf{y}^{i,d_k}}(f) \downarrow \sum_{i=1}^p L_{\mathbf{y}^{i,*}}(f) = \sum_{i=1}^p \int f \, d\phi_i.$$

Therefore, $(\phi_1, \ldots, \phi_p)$ is admissible for problem $\mathbf{Q}$ with value $\sum_{i=1}^p \int f \, d\phi_i \geq f^*$, and so $(\phi_1, \ldots, \phi_p)$ is an optimal solution of $\mathbf{Q}$. Finally, by Theorem .3, $\sum_{i=1}^p \phi_i = \mu_{\mathbf{\Omega}}$. And so, for each $\alpha \in \mathbb{N}^n$:

$$\lim_{k\to\infty} \sum_{i=1}^p y_\alpha^{i,d_k} = \sum_{i=1}^p y_\alpha^{i,*} = z_\alpha^* = \int_{\mathbf{\Omega}} \mathbf{x}^\alpha \, d\mu(\mathbf{x}).$$

As the converging subsequence $(d_k)_{k\in\mathbb{N}}$ was arbitrary, it follows that in fact the whole sequence $\left(\sum_{i=1}^p y_\alpha^{i,d}\right)_d$ converges to z_α for all $\alpha \in \mathbb{N}^n$, that is, (A.11) holds. □

A.3.3 The dual of $\mathbf{Q}_d$

Let $g_{i0}(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathbb{R}^n$, $i = 1, \ldots, p$. The dual of the semidefinite program $\mathbf{Q}_d$ is the semidefinite program:

$$\mathbf{Q}_d^*\colon (\rho_d^f)^* = \inf_{q,\sigma_{ij}} \left\{ \int_{\mathbf{B}} q \, d\mu \; : \; \begin{array}{c} q \in \Sigma[\mathbf{x}]_d, \; q - f = \sum_{j=0}^{m_i} \sigma_{ij} g_{ij}, \quad i = 1, \ldots, p \\ \sigma_{ij} \in \Sigma[\mathbf{x}]_{d-r_{ij}}, \quad j = 0, \ldots, m_i; \; i = 1, \ldots, p \end{array} \right\}. \tag{A.12}$$

Proposition .5. *Assume that for every $i = 1, \ldots, p$, both $\mathbf{\Omega}_i$ and $\mathbf{B} \setminus \mathbf{\Omega}_i$ have nonempty interior. Then there is no duality gap between (A.10) and its dual (A.12), that is, $\rho_d^f = (\rho_d^f)^*$ for all $d \geq d_0$. Moreover, (A.12) has an optimal solution $(q^*, (\sigma_{ij}^*))$.*

Proof. Let $(\phi_1^*, \ldots, \phi_p^*)$ be the measures defined in (A.8) from the proof of Theorem .3, and let $\mathbf{y}_d^i$ be the sequence of moments up to degree d of ϕ_i^*, $i = 1, \ldots, p$.

As every $\mathbf{\Omega}_i$ has nonempty interior, then clearly $\mathbf{M}_d(\mathbf{y}^i) \succ 0$ and $\mathbf{M}_{d-r_{ij}}(g_{ij}\mathbf{y}^i) \succ 0$ for every $j = 1, \ldots, m_i$ and $i = 1, \ldots, p$. As $\mathbf{B} \setminus \mathbf{\Omega}$ also has nonempty interior, then $\mathbf{M}_d\left(\mathbf{z} - \sum_{i=1}^p \mathbf{y}^i\right) \succ 0$. Therefore, Slater's condition holds for $\mathbf{Q}_d$. In addition, the set of admissible solutions of $\mathbf{Q}_d^*$ is nonempty (set $q = f$ and $\sigma_{ij} = 0$ for all i, j), and therefore a standard result in conic convex optimization yields the desired result [3]. □

As in the case of a basic closed semi-algebraic set, when f is the constant function 1, the convergence $\rho_d^f \to f^* = \mu(\mathbf{\Omega})$ is monotone non-increasing, a highly desirable feature. However, in typical examples, this convergence is rather slow. Again, one may take for f a function that is nonnegative on $\mathbf{\Omega}$ and which vanishes on $\partial\mathbf{\Omega}$. This accelerates the convergence both $\rho_d^f \to f^*$ and $\sum_i y_0^{i,d} \to \mu(\mathbf{\Omega})$ as $d \to \infty$, but if by construction the former is monotone non-increasing, the latter is not monotone anymore, which is a rather annoying feature if the goal is to obtain a converging sequence of *upper bounds*. In the next section, we describe a technique that allows for significant acceleration of the convergence $\sum_i y_0^{i,d} \to \mu(\mathbf{\Omega})$ as $d \to \infty$, while maintaining its monotone non-increasing character.

A.3.4 Convergence improvement using Stokes' formula

In this section, we show how to improve significantly the monotone non-increasing convergence of ρ_d^1 (i.e. ρ_d^f with $f = 1$) to $\mu(\mathbf{\Omega})$. To do this, we will use Stokes' theorem for integration and, to avoid technicalities, we assume that $\mathbf{\Omega} \subset \mathbb{R}^n$ is the closure of its interior, i.e., $\mathbf{\Omega} = \overline{\text{int}(\mathbf{\Omega})}$. The basic idea is simple to express in informal terms.

Since we know in advance that $(\phi_1^*, \ldots, \phi_p^*)$ in (A.8) is an optimal solution of problem $\mathbf{Q}$, every additional information in terms of linear constraints on the moments of ϕ_i^* can be included in $\mathbf{Q}$ without changing its optimal value. BUT when included in the relaxation $\mathbf{Q}_d$, it will provide useful additional constraints that restrict the feasible set of $\mathbf{Q}_d$ and so make its optimal value necessarily smaller.

Suppose for the moment that the set $\mathbf{\Omega}$ is compact with smooth boundary, and assume

3. In fact, as the set of optimal solutions of (A.10) is compact, the absence of a duality gap between (A.10) and (A.12) also follows from [97] without the conditions $\text{int}(\mathbf{\Omega}_i) \neq \varnothing$ and $\text{int}(\mathbf{B} \setminus \mathbf{\Omega}_i) \neq \varnothing$.

that the measure μ has a density h with respect to Lebesgue measure $d\mathbf{x}$ of the form $q(\mathbf{x})\exp(r(\mathbf{x}))\,\mathbf{1}_{\mathbf{B}}(\mathbf{x})$ for some polynomials $r, q \in \mathbb{R}[\mathbf{x}]$. Let X be a given vector field and $f \in \mathbb{R}[\mathbf{x}]$. Then Stokes' theorem states:

$$\int_{\mathbf{\Omega}} \mathrm{Div}(X)\, f(\mathbf{x})\, h(\mathbf{x}) d\mathbf{x} + \int_{\mathbf{\Omega}} \langle X, \nabla(f(\mathbf{x})\, h(\mathbf{x}))\rangle d\mathbf{x} = \int_{\partial\mathbf{\Omega}} \langle X, \vec{n}_{\mathbf{x}}\rangle f(\mathbf{x}) h(\mathbf{x}) d\sigma(\mathbf{x}),$$

where $\vec{n}_{\mathbf{x}}$ is the outward-pointing normal at $\mathbf{x} \in \partial\mathbf{\Omega}$, and σ is the $(n-1)$-dimensional Hausdorff measure on $\partial\mathbf{\Omega}$. In particular, if f vanishes on $\partial\mathbf{\Omega}$ and $\mathbf{X} = e_k \in \mathbb{R}^n$ (where $e_k(j) = \delta_{k=j}$), Stokes' formula becomes:

$$\int_{\mathbf{\Omega}} \frac{\partial}{\partial x_k} \left(f(\mathbf{x})\, h(\mathbf{x})\right) d\mathbf{x} = 0.$$

To exploit this in our particular context where $\mathbf{\Omega}$ is defined in (A.2), let $g = \prod_{i=1}^{p} \prod_{j=1}^{m_i} g_{ij}$ and let $\mathbf{x} \mapsto f(\mathbf{x}) = \mathbf{x}^\alpha g(\mathbf{x}) q(\mathbf{x})$ with $\alpha \in \mathbb{N}^n$ arbitrary. Then f vanishes on $\partial\mathbf{\Omega}$ and on $\partial\mathbf{\Omega}_{i_1,\ldots,i_s} := \mathbf{\Omega}_{i_1} \cap \cdots \cap \mathbf{\Omega}_{i_s}$ for all $1 \le i_1 < \cdots < i_s \le p$, $s = 1, \ldots, p$. Hence, by Stokes' theorem:

$$\int_{\mathbf{\Omega}} p_{\alpha,k}(\mathbf{x}) \underbrace{q(\mathbf{x})\, \exp(r(\mathbf{x}))\, d\mathbf{x}}_{d\mu(\mathbf{x})} = 0 \tag{A.13a}$$

$$\int_{\mathbf{\Omega}_{i_1,\ldots,i_s}} p_{\alpha,k}(\mathbf{x}) \underbrace{q(\mathbf{x})\, \exp(r(\mathbf{x}))\, d\mathbf{x}}_{d\mu(\mathbf{x})} = 0, \tag{A.13b}$$

for all $1 \le i_1 < \cdots < i_s \le p$, $s = 1, \ldots, p$, where

$$p_{\alpha,k}(\mathbf{x}) = q(\mathbf{x}) \frac{\partial}{\partial_{x_k}} (\mathbf{x}^\alpha g(\mathbf{x})) + 2\mathbf{x}^\alpha g(\mathbf{x}) \frac{\partial}{\partial_{x_k}} q(\mathbf{x}) + \mathbf{x}^\alpha g(\mathbf{x}) q(\mathbf{x}) \frac{\partial}{\partial_{x_k}} r(\mathbf{x}).$$

Recalling how ϕ_i^* is defined in (A.8), it can be written as

$$\phi_i^* = \sum_{s=1}^{p} \sum_{1 \le i_1 < \cdots < i_s \le p} \phi^*_{i,i_1,\ldots,i_s},$$

where each $\phi^*_{i,i_1,\ldots,i_s}$ is supported on $\mathbf{\Omega}_i \cap \mathbf{\Omega}_{i_1,\ldots,i_s}$ and has a constant density with respect to μ. Therefore, for every $i = 1, \ldots, p$:

$$\int_{\mathbf{\Omega}_i} p_{\alpha,k}(\mathbf{x})\, d\phi_i^* = 0, \quad \forall \alpha \in \mathbb{N}^n,\ k = 1, \ldots, n.$$

Hence, this provides additional useful information on the optimal solution $(\phi_1^*, \ldots, \phi_p^*)$ of

$\mathbf{Q}$ defined in (A.8). Namely, it translates into:

$$L_{\mathbf{y}^i}(p_{\alpha,k}) = 0, \quad \forall \alpha \in \mathbb{N}^n,\ k = 1,\dots,n,\ i = 1,\dots,p,$$

i.e., linear constraints on the moments of ϕ_i^*, for every $i = 1,\dots,p$.

Plugging these additional linear constraints on the moments of ϕ_i^* into the relaxation $\mathbf{Q}_d$ yields the following new hierarchy of SDP-relaxation $(\mathbf{Q}_d^{\text{Stokes}})_{d\geq d_0}$:

$$\rho_d^{\text{Stokes}} = \sup_{\mathbf{y}^1,\dots,\mathbf{y}^p} \left\{ \sum_{i=1}^p L_{\mathbf{y}^i}(1) : \begin{array}{l} \mathbf{M}_d\left(\mathbf{z} - \sum_{i=1}^p \mathbf{y}^i\right) \succeq 0;\ \mathbf{M}_d(\mathbf{y}^i) \succeq 0, \mathbf{M}_{d-r_{ij}}\left(g_{ij}\mathbf{y}^i\right) \succeq 0, \\ L_{\mathbf{y}^i}(p_{\alpha,k}) = 0,\ k = 1,\dots,n,\ |\alpha| \leq 2d - \deg(p_{\alpha,k}), \\ j = 1,\dots,m_i,\ i = 1,\dots,p \end{array} \right\}. \tag{A.14}$$

By construction, $\rho_d^1 \geq \rho_d^{\text{Stokes}} \geq \mu(\mathbf{\Omega})$ holds for every $d \geq d_0$, and the analogue of Theorem .4 (with $f = 1$) reads:

Theorem .6. *Consider the semidefinite programs $(\mathbf{Q}_d^{\text{Stokes}})$, $d \geq d_0$, defined in (A.14). Then:*

1. *$\left(\mathbf{Q}_d^{\text{Stokes}}\right)_{d\in\mathbb{N}}$ has an optimal solution, and the associated sequence of optimal values $(\rho_d^{\text{Stokes}})_{d\in\mathbb{N}}$ is monotone non-increasing and converges to $\mu(\mathbf{\Omega})$, that is:*

$$\rho_d^{\text{Stokes}} \downarrow \mu(\mathbf{\Omega}), \quad \text{as } d \to \infty.$$

2. *Let $(\mathbf{y}^{1,d},\dots,\mathbf{y}^{p,d})$ be an optimal solution of $\mathbf{Q}_d^{\text{Stokes}}$. Then for each $\alpha \in \mathbb{N}^n$:*

The important feature of Theorem .6 is that we now have the monotone non-increasing convergence $\rho_d^{\text{Stokes}} \downarrow \mu(\mathbf{\Omega})$, which improves upon the slower convergence seen in Theorem .4.

A.3.5 Gaussian measure of non compact sets Ω

So far Theorem .4 and Theorem .6 have been given for μ supported on a box $\mathbf{B}$, and so only for sets $\mathbf{\Omega}$ in (A.5) that are compact.

It turns out that for a Gaussian measure μ of (possibly non-compact) sets $\mathbf{\Omega} = \bigcup_i \mathbf{\Omega}_i$, Theorem .4 (resp. Theorem .6) can be extended to the non-compact case with exactly the same statement and exactly the same semidefinite relaxations (A.10) and (A.14) as before. The only difference is that now $\mathbf{z} = (z_\alpha)$ is the vector of moments of the Gaussian measure μ (instead of the moments of the Lebesgue measure on $\mathbf{B}$ previously). However, in

the Gaussian case the proof of Theorem .4 and Theorem .6 uses quite different arguments (some already used in [95] for a basic semi-algebraic set). Indeed, as $\mathbf{\Omega}$ is not necessarily compact:

- The uniform bound $\sup_\alpha |z_\alpha| \leq 1$ is not valid anymore for the relaxations $\mathbf{Q}_d$ and $\mathbf{Q}_d^{\text{Stokes}}$.
- One cannot invoke Putinar's Positivstellensatz [28] anymore.
- The standard version of Stokes' theorem, where $\mathbf{\Omega}$ is compact, cannot be invoked anymore either.

The new arguments that we need are the following:

- A crucial fact is that the Gaussian measure μ satisfies Carleman's condition:

$$\sum_{k=1}^{\infty} \left(\int_{\mathbb{R}^n} x_i^{2k} \, d\mu(\mathbf{x}) \right)^{-1/2k} = +\infty, \quad i = 1, \ldots, n. \tag{A.15}$$

  Then a sequence $\mathbf{y} = (y_\alpha)_{\alpha \in \mathbb{N}^n}$ such that $\mathbf{M}_d(\mathbf{y}) \succeq 0$ for all $d \in \mathbb{N}$, and

$$\sum_{k=1}^{\infty} L_{\mathbf{y}}(x_i^{2k})^{-1/2k} = +\infty, \quad i = 1, \ldots, n,$$

  has a unique representing measure ϕ on $\mathbb{R}^n$, which is moment determinate; see, for instance, [27, Proposition 3.5, p. 60].
- If in addition $\mathbf{M}_d(h\mathbf{y}) \succeq 0$ for all $d \in \mathbb{N}$ (where $h \in \mathbb{R}[\mathbf{x}]$), and as ϕ satisfies (A.15), then $h(\mathbf{x}) \geq 0$ for all $\mathbf{x}$ in the support of ϕ; see Lasserre [98]. This argument is used to show that ϕ is supported on $\mathbf{\Omega}$.
- To obtain a version of Stokes for a non-compact set $\mathbf{\Omega}$ with boundary $\partial\mathbf{\Omega}$, we invoke a limiting argument that uses (the standard) Stokes' theorem on the compact $\mathbf{\Omega} \cap \mathbf{B}(0, M)$ (where $\mathbf{B}(0, M) = \{\mathbf{x} : \|\mathbf{x}\| \leq M\}$). Letting $M \to \infty$ and using the Monotone and Bounded Convergence theorems yields the desired result.

For more details, the reader is referred to [95], where such arguments have been used in the case of a *basic* semi-algebraic set.

Finally, it is worth emphasizing that this methodology also works for *any* measure μ that satisfies (A.15) (and whose moments are known or can be computed); an important special case is the exponential measure on the positive orthant $\mathbb{R}^n_+$.

Remark .3. As mentioned above, in [95] the first author has already used Stokes' formula to accelerate the convergence of a hierarchy of semidefinite relaxations to approximate the

Gaussian measure $\mu(\mathbf{\Omega})$ of a *basic* semi-algebraic set $\mathbf{\Omega}$, not necessarily compact. The important and non-trivial novelty here is that

(i) $\mathbf{\Omega} = \bigcup_{i=1}^{p} \mathbf{\Omega}_i$ is now a *union* of basic semi-algebraic sets, and

(ii) even if this complicates matters significantly, we are still able to work with measures ϕ_i, each supported on $\mathbf{\Omega}_i$ (a basic semi-algebraic set). It turns out that $\mu(\mathbf{\Omega}) = \sum_{i=1}^{p} \phi_i^*$, where each ϕ_i^* has a piecewise constant density with respect to μ (constant on each of the possible intersections $\mathbf{\Omega}_i \cap \mathbf{\Omega}_{i_1,\ldots,i_p}$). By using a family of polynomials that all vanish on the boundary of *each* $\mathbf{\Omega}_i \cap \mathbf{\Omega}_{i_1,\ldots,i_p}$, we can exploit Stokes' theorem on each piece and sum up to obtain a family of linear constraints on the moments of ϕ_i^*.

A.4 Numerical Experiments

For illustration purposes, we have applied the methodology to a few (simple) examples. We report some numerical experiments carried out in MATLAB and GloptiPoly3 [1], a software package for manipulating and solving generalized problems of moments. The SDP problems were solved with SeDuMi 1.1R3.

A.4.1 Lebesgue volume of a union of two ellipsoids

We first consider a simple example of two ellipsoids in $\mathbb{R}^2$ where the exact value $\mu(\mathbf{\Omega})$ can be computed exactly so that we can compare it with our upper bounds. So we want to compute the Lebesgue measure of $\mathbf{\Omega} = \mathbf{\Omega}_1 \cup \mathbf{\Omega}_2$ with $\mathbf{\Omega}_1 = \{(x_1, x_2) \in \mathbb{R}^2 : \frac{x_1^2}{4} + x_2^2 \leq 1\}$ and $\mathbf{\Omega}_2 = \{(x_1, x_2) \in \mathbb{R}^2 : \frac{x_2^2}{4} + x_1^2 \leq 1\}$. In this example, we take $\mathbf{B} := [-2, 2]^2$.

The results are displayed in Figure A.1 with: in **orange** the approximation of the Lebesgue volume $\mu(\mathbf{\Omega})$ without using Stokes' formulas, in **red** the approximation when using Stokes' formulas, and in **blue** the exact value of $\mu(\mathbf{\Omega})$. We next consider a union of two ellipsoids in dimension $n = 3$. Let

$$\mathbf{\Omega}_1 = \left\{\mathbf{x} \in \mathbb{R}^3 \colon x_1^2 + 4x_2^2 + 4x_3^2 \leq 1\right\}, \quad \mathbf{\Omega}_2 = \left\{\mathbf{x} \in \mathbb{R}^3 \colon x_2^2 + 4x_1^2 + 4x_3^2 \leq 1\right\},$$

$\mathbf{\Omega} = \mathbf{\Omega}_1 \cup \mathbf{\Omega}_2$, and $\mathbf{B} = [-1, 1]^3$. Results are displayed in Figure A.2. In both examples, one can check that the convergence is much faster when using Stokes' formula.

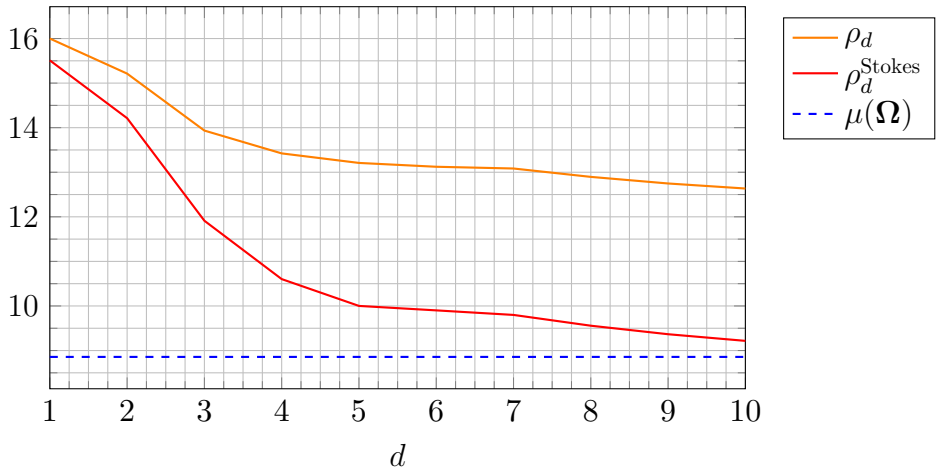


Figure A.1 $n = 2$: Lebesgue measure of a union of 2 ellipsoids

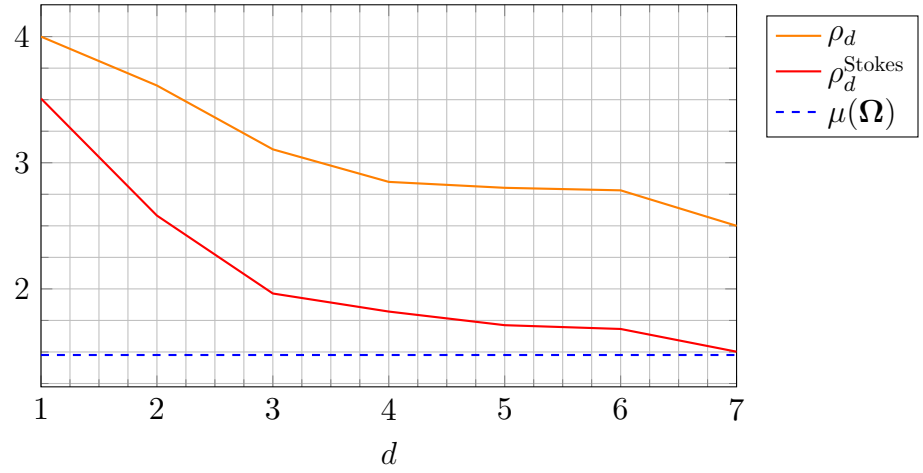


Figure A.2 $n = 3$: Lebesgue measure of a union of 2 ellipsoids

A.4.2 Lebesgue measure of a union of three ellipsoids

We next consider a union of three ellipsoids in dimension $n = 2$, with:

$$\mathbf{\Omega}_1 = \left\{ \mathbf{x} \in \mathbb{R}^2 \colon \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \frac{16}{9} & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq 1 \right\},$$

$$\mathbf{\Omega}_2 = \left\{ \mathbf{x} \in \mathbb{R}^2 \colon \frac{1}{9} \begin{bmatrix} x_1 - \frac{1}{10} & x_2 - \frac{1}{10} \end{bmatrix} \begin{bmatrix} 31 & 5\sqrt{3} \\ 5\sqrt{3} & 21 \end{bmatrix} \begin{bmatrix} x_1 - \frac{1}{10} \\ x_2 - \frac{1}{10} \end{bmatrix} \leq 1 \right\},$$

$$\mathbf{\Omega}_3 = \left\{ \mathbf{x} \in \mathbb{R}^2 \colon \frac{1}{9} \begin{bmatrix} x_1 + \frac{1}{10} & x_2 - \frac{1}{10} \end{bmatrix} \begin{bmatrix} 31 & -5\sqrt{3} \\ -5\sqrt{3} & 21 \end{bmatrix} \begin{bmatrix} x_1 + \frac{1}{10} \\ x_2 - \frac{1}{10} \end{bmatrix} \leq 1 \right\},$$

and $\mathbf{B} = [-1, 1]^2$. In Figure A.3, we also compare our results with those obtained using *Bonferroni* inequalities. In **red**, the upper bounds obtained by solving $\mathbf{Q}^{\text{Stokes}}$, in **orange**, the lower bounds obtained by solving $\mathbf{Q}^{\text{Stokes}}$ for the complement, and in **blue**, the upper bounds obtained by using Bonferroni inequalities. (For a fair comparison, for each relaxation in the Bonferroni case, we also use appropriate Stokes' constraints.)

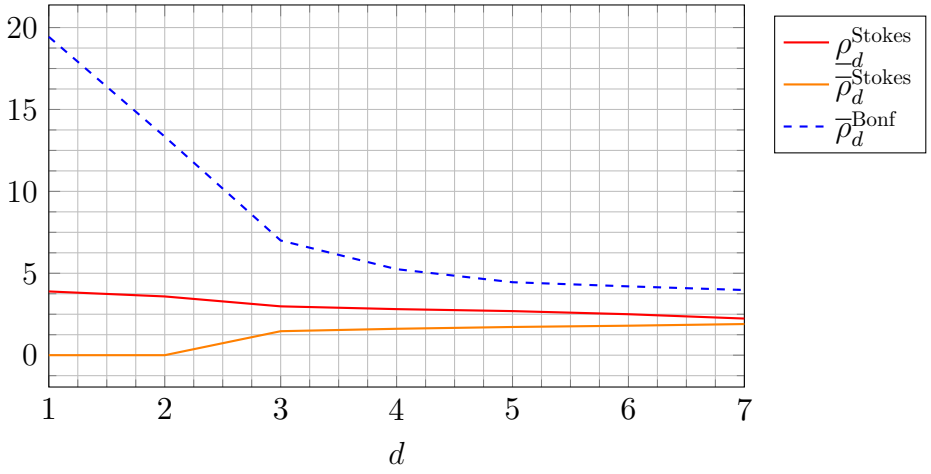


Figure A.3 $n = 2$: Lebesgue measure of a union of 3 ellipsoids

A.4.3 Examples for the Gaussian measure

In this section, we consider the Gaussian measure

$$d\mu = \exp\left(-\frac{\|\mathbf{x}\|^2}{\sigma^2}\right) d\mathbf{x}$$

with variance $\sigma^2 = 0.8$. For each example, we have computed two upper bounds and two lower bounds for $\mu(\mathbf{\Omega})$. The first (resp. second) upper bound $\overline{\rho}_d$ (resp. $\overline{\rho}_d^{\text{Stokes}}$) is obtained by solving the semidefinite relaxation $\mathbf{Q}_d$ (resp. $\mathbf{Q}_d^{\text{Stokes}}$). Similarly, the lower bounds $\underline{\rho}_d$ (resp. $\underline{\rho}_d^{\text{Stokes}}$) are obtained from upper bounds for the complement $\mathbb{R}^n \setminus \mathbf{\Omega}$. The respective relative error gaps are denoted by

$$\epsilon_d = \frac{\overline{\rho}_d - \underline{\rho}_d}{\overline{\rho}_d}, \quad \text{and} \quad \epsilon_d^{\text{Stokes}} = \frac{\overline{\rho}_d^{\text{Stokes}} - \underline{\rho}_d^{\text{Stokes}}}{\overline{\rho}_d^{\text{Stokes}}}.$$

Example .4 (Union of two ellipsoids in $\mathbb{R}^2$)**.** In this example, $\mathbf{\Omega}$ is the union of two

ellipsoids. Let $\mathbf{\Omega} \coloneqq \mathbf{\Omega}_1 \cup \mathbf{\Omega}_2$, with

$$\mathbf{\Omega}_1 = \left\{\mathbf{x} \in \mathbb{R}^2 \colon (\mathbf{x}-\mathbf{u})^\intercal \mathbf{A}_1 (\mathbf{x}-\mathbf{u}) \leq 1\right\}, \text{ and } \mathbf{\Omega}_2 = \left\{\mathbf{x} \in \mathbb{R}^2 \colon (\mathbf{x}-\mathbf{v})^\intercal \mathbf{A}_2 (\mathbf{x}-\mathbf{v}) \leq 1\right\}$$

for the values

$$\mathbf{v} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{4} \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & 1 \end{bmatrix}$$

and for multiple choice of $\mathbf{u} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \begin{pmatrix} 0.1 & 0.5 \end{pmatrix}, \begin{pmatrix} 0.5 & 0.5 \end{pmatrix}$.In this case, $\rho \coloneqq \mu(\mathbf{\Omega})$ can be computed exactly, and so we have displayed the values of the relative errors denoted by

$$\epsilon_d = \frac{\overline{\rho}_d - \rho}{\rho}, \quad \text{and} \quad \epsilon_d^{\text{Stokes}} = \frac{\overline{\rho}_d^{\text{Stokes}} - \rho}{\rho},$$

respectively, depending on whether or not we have used Stokes' formula. As one can see in Table A.1, for a reasonable value $d = 10$, the relative error (when using Stokes' formula) is quite good. The respective behaviors are displayed in Figure A.4.

Table A.1 Bounds and relative gap for $d = 10$

	$\mathbf{u} = \begin{pmatrix} 0 & 0 \end{pmatrix}$	$\mathbf{u} = \begin{pmatrix} 1/10 & 1/2 \end{pmatrix}$	$\mathbf{u} = \begin{pmatrix} 1/2 & 1/2 \end{pmatrix}$
$\overline{\rho}_{10}$	1.9649	1.9554	1.9484
$\underline{\rho}_{10}$	1.6129	1.5752	1.5369
ϵ_{10}	18%	19%	21%
$\overline{\rho}_{10}^{\text{Stokes}}$	1.8571	1.8308	1.8156
$\underline{\rho}_{10}^{\text{Stokes}}$	1.7948	1.7746	1.7618
$\epsilon_{10}^{\text{Stokes}}$	3.4%	3.1%	3%

Example .5 (Union of an ellipsoid and a hyperboloid in $\mathbb{R}^2$)**.** Consider $\mathbf{\Omega} = \mathbf{\Omega}_1 \cup \mathbf{\Omega}_2$ with

$$\mathbf{\Omega}_1 = \left\{\mathbf{x} \in \mathbb{R}^2 \colon (\mathbf{x}-\mathbf{u})^\intercal \mathbf{A}_1 (\mathbf{x}-\mathbf{u}) \leq 1\right\}, \text{ and } \mathbf{\Omega}_2 = \left\{\mathbf{x} \in \mathbb{R}^2 \colon (\mathbf{x}-\mathbf{v})^\intercal \mathbf{A}_2 (\mathbf{x}-\mathbf{v}) \leq 1\right\}$$

for the values

$$\mathbf{u} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} -2 & 0 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} \frac{1}{16} & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & -1 \end{bmatrix}.$$

In this case, $\mathbf{\Omega}$ is not a compact set as it is unbounded. The results for $d = 9$, displayed

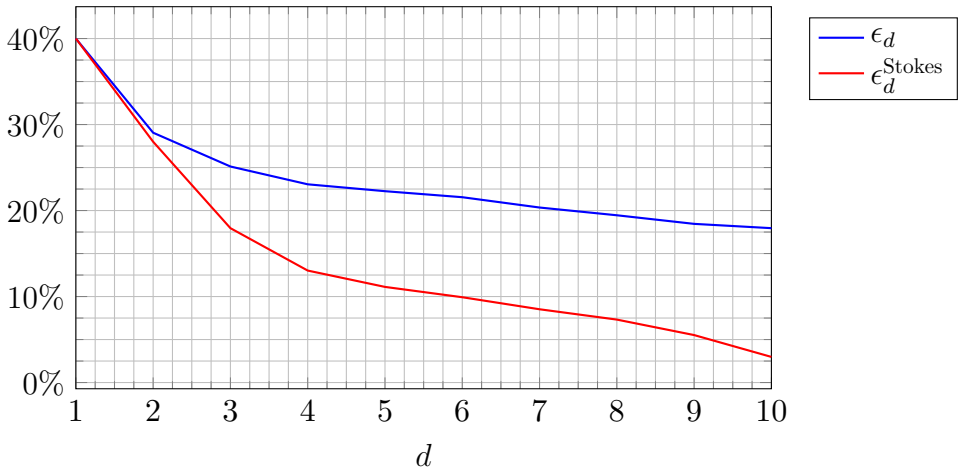


Figure A.4 Relative errors ϵ_d (**blue**) and $\epsilon_d^{\text{Stokes}}$ (**red**)

in Table A.2, show that a good value is already obtained when using Stokes' formula. The respective behaviors of ϵ_d and $\epsilon_d^{\text{Stokes}}$, displayed in Figure A.5, also show that using Stokes' formula yields a significant improvement.

Table A.2 Bounds and relative gaps for $d = 9$

$\overline{\rho}_9$	$\underline{\rho}_9$	ϵ_9	$\overline{\rho}_9^{\text{Stokes}}$	$\underline{\rho}_9^{\text{Stokes}}$	$\epsilon_9^{\text{Stokes}}$
2.0038	1.8252	8.9%	1.9347	1.9019	1.7%

Example .6 (Union of 2 hyperboloids in $\mathbb{R}^2$)**.** Consider $\mathbf{\Omega} = \mathbf{\Omega}_1 \cup \mathbf{\Omega}_2$ with

$$\mathbf{\Omega}_1 = \left\{\mathbf{x} \in \mathbb{R}^2 \colon (\mathbf{x}-\mathbf{u})^\intercal \mathbf{A}_1 (\mathbf{x}-\mathbf{u}) \leq 1\right\}, \text{ and } \mathbf{\Omega}_2 = \left\{\mathbf{x} \in \mathbb{R}^2 \colon (\mathbf{x}-\mathbf{v})^\intercal \mathbf{A}_2 (\mathbf{x}-\mathbf{v}) \leq 1\right\}$$

for the values

$$\mathbf{u} = \begin{pmatrix} 0 & 0 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} -2 & 0 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} -\frac{1}{16} & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & -1 \end{bmatrix}$$

Again, $\mathbf{\Omega}$ is not a compact set. The results for $d = 9$, displayed in Table A.3, confirm that using Stokes' formula yields a significant improvement. The respective behaviors of ϵ_d and $\epsilon_d^{\text{Stokes}}$, displayed in Figure A.6, also show the improvement.

Example .7 (Union of two hyperboloids in $\mathbb{R}^3$)**.** We next consider an example in dimension

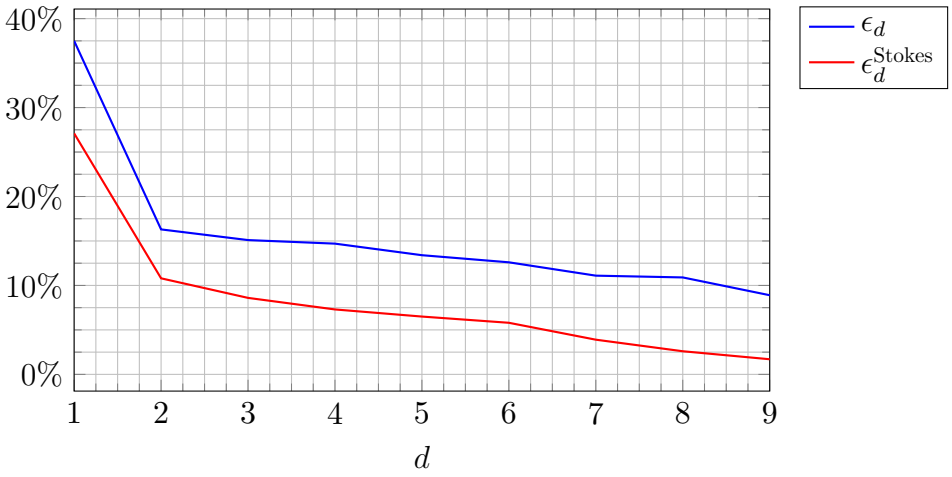


Figure A.5 Relative error ϵ_d (**blue**) and $\epsilon_d^{\text{Stokes}}$ (**red**)

Table A.3 Bounds and relative gaps for $d = 9$

$\overline{\rho}_9$	$\underline{\rho}_9$	ϵ_9	$\overline{\rho}_9^{\text{Stokes}}$	$\underline{\rho}_9^{\text{Stokes}}$	$\epsilon_9^{\text{Stokes}}$
2.0046	1.8342	8.5%	1.9542	1.9083	2.3%

$n = 3$. Let

$$\mathbf{\Omega}_1 = \left\{\mathbf{x} \in \mathbb{R}^2 \colon (\mathbf{x}-\mathbf{u})^\intercal \mathbf{A}_1 (\mathbf{x}-\mathbf{u}) \leq 1\right\}, \text{ and } \mathbf{\Omega}_2 = \left\{\mathbf{x} \in \mathbb{R}^2 \colon (\mathbf{x}-\mathbf{v})^\intercal \mathbf{A}_2 (\mathbf{x}-\mathbf{v}) \leq 1\right\}$$

for the values

$$\mathbf{u} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} -2 & 0 & -1 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} -\frac{1}{16} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & 0 \\ \frac{1}{2} & -1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}.$$

Results for $d = 6$ are displayed in Table A.4, and the relative errors ϵ_d and $\epsilon_d^{\text{Stokes}}$ are displayed in Figure A.7. The quality of results is comparable to that in Definition .5 and Definition .6 for $d = 6$.

Table A.4 Bounds and relative gaps for $d = 6$

$\overline{\rho}_6$	$\underline{\rho}_6$	ϵ_6	$\overline{\rho}_6^{\text{Stokes}}$	$\underline{\rho}_6^{\text{Stokes}}$	$\epsilon_6^{\text{Stokes}}$
2.8222	2.3123	18%	2.6856	2.5360	5.6%

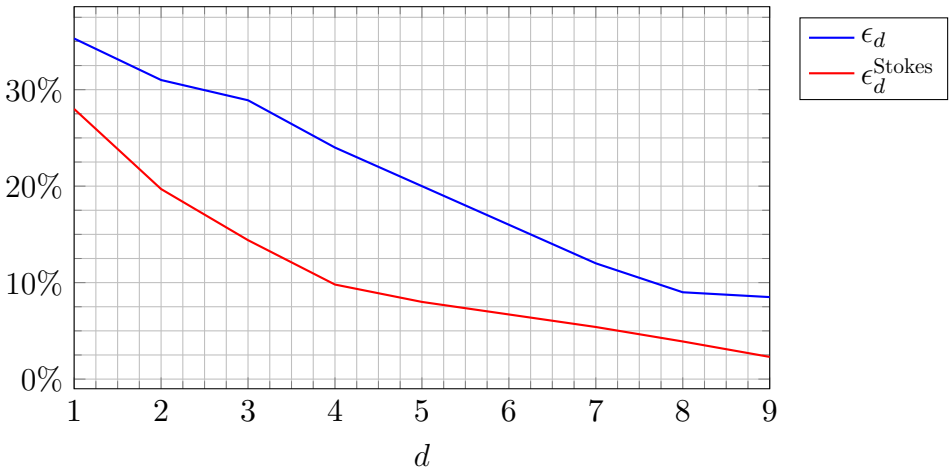


Figure A.6 Relative errors $\epsilon_d^{\text{Stokes}}$ (**red**) and ϵ_d (**blue**)

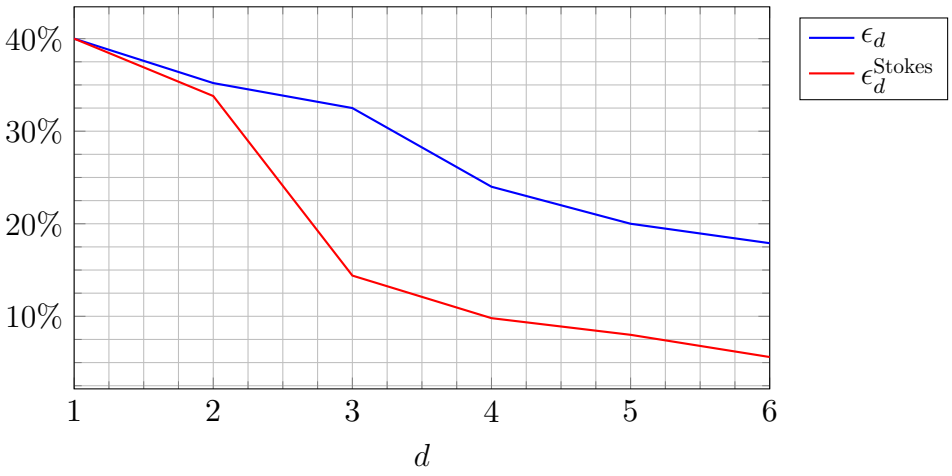


Figure A.7 Relative error $\epsilon_d^{\text{Stokes}}$ (**red**) and ϵ_d (**blue**)

Example .8 (Union of a halfplane and a hyperboloid in $\mathbb{R}^3$)**.** Still in dimension $n = 3$, let $\mathbf{\Omega} := \mathbf{\Omega}_1 \cup \mathbf{\Omega}_2$ with $\mathbf{\Omega}_1 = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x}^\intercal e \leq 1\}$ and $\mathbf{\Omega}_2 = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x}^\intercal \mathbf{A}\mathbf{x} \leq 1\}$, where $e = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ and

$$\mathbf{A} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & 0 \\ \frac{1}{2} & -1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}.$$

The relative errors ϵ_d and $\epsilon_d^{\text{Stokes}}$ are displayed in Figure A.8.

One can see that in all examples, quite good approximations are obtained with relatively few moments (up to order $2d \leq 18$ for $n = 2$ and $2d \leq 14$ for $n = 3$) provided that we use

Table A.5 Bounds and relative gaps for $d = 7$

$\overline{\rho}_7$	$\underline{\rho}_7$	ϵ_7	$\overline{\rho}_7^{\text{Stokes}}$	$\underline{\rho}_7^{\text{Stokes}}$	$\epsilon_7^{\text{Stokes}}$
2.8143	2.3494	17%	2.6887	2.5338	5.8%

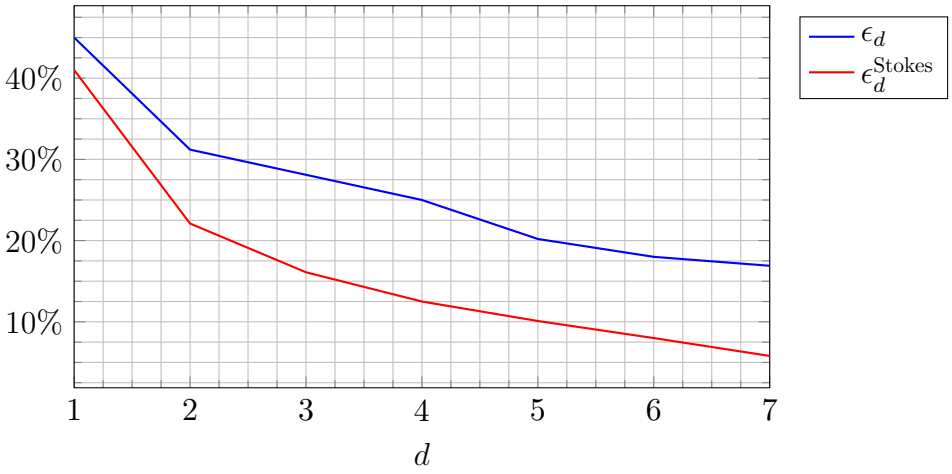


Figure A.8 Relative error $\epsilon_d^{\text{Stokes}}$ (**red**) and ϵ_d (**blue**)

the hierarchy $\mathbf{Q}_d^{\text{Stokes}}$ in (A.14) with the additional moment constraints induced by Stokes' formula. The convergence of the hierarchy $\mathbf{Q}_d$ in (A.10) (without those Stokes constraints) is indeed much slower.

For all the examples that we have treated, the (crucial) moment and localizing matrices involved in (A.10) and (A.14) have been expressed in the canonical basis $(\mathbf{x}^\alpha)_{\alpha\in\mathbb{N}^n}$ of monomials for simplicity and ease of implementation of the SDP relaxations. However, this choice is, in fact, the worst from a numerical point of view (in terms of stability and robustness), which prevented us from solving $\mathbf{Q}_d$ and $\mathbf{Q}_d^{\text{Stokes}}$ for $d \geq 7$ when $n = 3$. It is very likely that the basis of orthonormal polynomials w.r.t. μ (Legendre for the Lebesgue measure μ on $[-1, 1]$, and Hermite for the Gaussian measure μ) is a much better (and recommended) choice. Such a more sophisticated implementation was beyond the scope of this paper.

A.5 Conclusion

In this paper, we have provided a numerical scheme to approximate as closely as desired the measure $\mu(\mathbf{\Omega})$ of a finite *union* $\mathbf{\Omega} = \cup_{i=1}^p \mathbf{\Omega}_i$ of basic semi-algebraic sets (the case of a single basic semi-algebraic set was treated in [99]). Surprisingly, even though the case of

a union of semi-algebraic sets complicates matters significantly, we are still able to adapt the methodology developed in [95] and provide a monotone non-increasing (resp. non-decreasing) sequence of upper (resp. lower) bounds that converges to $\mu(\mathbf{\Omega})$ as the number of moments considered increases. In addition, we are also able to use additional moment constraints induced by an appropriate application of Stokes' Theorem, which permits us to significantly improve the convergence. In fact, these additional moment constraints are crucial to obtain good bounds rapidly as they permit a strong attenuation of a Gibbs' phenomenon that would otherwise appear. Our current implementation could be significantly improved by using a basis for polynomials more appropriate than the usual canonical basis of monomials (the worst choice from a numerical stability point of view). For instance, in doing so, it should be possible to implement step $d = 8, 9$ of the hierarchy in dimension $n = 3$, and step $d = 7$ for $n = 4$. As the convergence seems to be fast, each additional step of the hierarchy can yield a significant improvement. The methodology was presented for the Lebesgue measure μ when $\mathbf{\Omega}$ is compact, and the Gaussian measure for non-compact sets $\mathbf{\Omega}$, but in fact, and remarkably, the same methodology works for any measure μ that satisfies Carleman's condition, provided that all its moments are available (or can be computed easily).

Of course, the methodology proposed in this paper is computationally expensive, especially when compared with Monte Carlo-type methods. But the latter provide only an estimate of $\mu(\mathbf{\Omega})$ and by no means an upper or lower bound on $\mu(\mathbf{\Omega})$. Therefore, these two types of methods should be seen as complementary rather than competing. In its present form, it is also limited to small-dimension problems (typically $n \leq 3, 4$) because each upper (or lower) bound requires solving a semidefinite program whose size increases fast in the hierarchy. Thus, one is limited by the current efficiency of state-of-the-art semidefinite solvers. However, to the best of our knowledge, this is the first method that provides a sequence of upper and lower bounds with strong asymptotic guarantees, at least at this level of generality.

Acknowledgements

Research funded by by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement ERC-ADG 666981 TAMING)"

ANNEXE B MATÉRIEL SUPPLÉMENTAIRE POUR LE CHAPITRE 5

B.1 Supplementary material: Method and implementation

B.1.1 Proofs

Proof of Theorem 5.1. [58] have demonstrated how to transform a 3-SAT instance into an additive tree ensemble that predicts FALSE in the entire feature space if 3-SAT is FALSE, and otherwise has a non-trivial prediction function. Applying functionally identical pruning in the sense of Definition 5.1 to the forest henceforth leads to a compressed result with a single tree predicting FALSE if 3-SAT is FALSE, and to a different result otherwise. Given that 3-SAT is NP-complete, it follows that Problem (5.2) is NP-hard. □

Proof of Theorem 5.2. We consider the ensemble of trees $\mathcal{T}_1, \ldots, \mathcal{T}_M$ that are trained on the same dataset. Each tree $\mathcal{T}_m$ is a piecewise constant function that divides the input space into J_m disjoint regions, $R_{m,1}, \ldots, R_{m,J_m}$, with a constant prediction assigned to each region. This proves that the prediction function of any weighted ensemble will be identical across any region of the form $\bigcap_{m=1}^{M} R_{m,j_m}$ for $j_m \in [\![1, J_m]\!]$. The oracle will provide a new point only when there is a region where the predictions of the original and pruned ensembles differ. Consequently, the pruning process will terminate after a finite number of oracle calls at most equal to $\prod_{m=1}^{M} J_m$. □

B.1.2 Feature consistency for additive tree ensembles

This section provides the additional constraints required in the separation oracle to ensure that the feature values are consistent with all the branching decisions. The formulation is taken from [79].

Binary features. Let j be a binary feature. We assume that a split on the feature j is such that the tree branches to the left if the value of x_j is 0 and branches to the right otherwise. We consider the binary variable x_j that indicates if the value of x_j is 1. For each tree m, let $\mathcal{V}_m^j$ be the set of nodes that split on binary feature j. We ensure that the value of x_j is consistent across the nodes of tree m with the constraints:

$$\mathrm{f}_{m,\mathbf{left}(v)} \leq 1 - x_j, \forall v \in \mathcal{V}_m^j, \tag{B.1a}$$

$$\mathrm{f}_{m,\mathbf{right}(v)} \leq x_j, \quad \forall v \in \mathcal{V}_m^j. \tag{B.1b}$$

Constraint (B.1a) ensures that the value of x_j is 1 if we cannot go to the left child at node v. Similarly, Constraint (B.1b) ensures that the value of x_j is 0 if we cannot go to the right child at node v.

Categorical features. Let j be a categorical feature. We assume that the categories of the feature j are given by $\{1, 2, \ldots, Z_j\}$. We introduce the binary variable $\nu_{j,z}$ to indicates whether the value of x_j is z for categorical feature j for $z \in [\![1, Z_j]\!]$. First, we ensure that only a single category is active for a categorical feature using the constraint:

$$\sum_{z=1}^{Z_j} \nu_{j,z} = 1. \tag{B.2}$$

Constraint (B.2) ensures that the value of x_j is only one of the categories. For each tree m, let $\mathcal{V}_m^{j,z}$ be the set of nodes that split on categorical feature j and branch to the right child if the value of x_j is z. We ensure that the value of $\nu_{j,z}$ is consistent across the nodes of tree m with the constraints:

$$\mathrm{f}_{m,\mathbf{left}(v)} \leq 1 - \nu_{j,z}, \forall z \in [\![1, Z_j]\!], \forall v \in \mathcal{V}_m^{j,z}, \tag{B.3a}$$

$$\mathrm{f}_{m,\mathbf{right}(v)} \leq \nu_{j,z}, \qquad \forall z \in [\![1, Z_j]\!], \forall v \in \mathcal{V}_m^{j,z}. \tag{B.3b}$$

Constraint (B.3a) ensures that, if the value of x_j is z, the path of point x in tree m at node v does not go to the left child. Similarly, constraint (B.3b) ensures that if the value of x_j is not z, the path of point x in tree m at node v goes to the right child.

B.2 Supplementary material: Computational experiments

This section presents the results of applying FIPE to various additive tree ensembles. The analysis includes:

- gradient boosted trees (`Gradient Boosting`) using the implementation from the `scikit-learn` package [100],
- random forests (`Random Forest`) also from the `scikit-learn` package,
- `LightGBM` using the implementation from [44], and
- `XGBoost` using the implementation from [43].

All hyperparameters are kept at their default values, except for the maximum depth of the trees, which is set to one or three. The experimental settings and datasets are kept as in the experiments presented in Section 5.4. First, we provide a summary of FIPE's pruning effectiveness on these tree ensembles, followed by detailed results for each ensemble.

B.2.1 Summary of Results: Pruning and Accuracy

Table B.1 presents the relative size of the ensemble after pruning with FIPE$-\|\cdot\|_1$ calculated as $\|w\|_0 / M$ along with the test accuracy averaged across all datasets. The results show that FIPE effectively prunes the ensemble in most cases. On average, `Random Forests` tend to be pruned more effectively by FIPE. There is no significant difference in ensemble size after pruning between the different boosted tree ensembles. Ensembles with a smaller maximum depth tend to have smaller relative pruned sizes. For a fixed maximum depth, the relative size of the pruned ensemble decreases as the number of base learners increases.

Result B.1. FIPE can consistently prune various types of tree ensembles. The extent of pruning is relatively insensitive to the training algorithm and depends mostly on the maximum depth of each tree.

B.2.2 Detailed results

We now present detailed results for the different tree ensembles. First, we compare the performance of FIPE$-\|\cdot\|_0$ and FIPE$-\|\cdot\|_1$ in Table B.2 for varying maximum depth. The results are consistent with those presented in the main paper. That is, the amount of pruning of FIPE$-\|\cdot\|_1$ is often equal to the one of FIPE$-\|\cdot\|_0$ with only a fraction of the computation time. This is true for all the ensemble models and datasets. On average, the difference between FIPE$-\|\cdot\|_0$ and FIPE$-\|\cdot\|_1$ is larger as the maximum tree depth increases.

Table B.1 Relative size of the pruned ensemble using FIPE$-\|\cdot\|_1$ and test accuracy for different tree ensembles and maximum tree depths.

		$\frac{\|w\|_0}{M}$			Test Accuracy		
	M	50	100	200	50	100	200
Ensemble	max depth						
AdaBoost	1	37.15%	26.95%	18.67%	78.92%	77.22%	75.48%
Gradient Boosting	1	35.09%	27.81%	20.98%	84.57%	85.43%	85.73%
	3	79.71%	73.97%	67.92%	86.23%	86.47%	86.67%
LightGBM	1	35.05%	26.60%	19.53%	84.87%	85.73%	86.23%
	3	77.42%	73.34%	67.19%	86.61%	86.88%	86.96%
Random Forest	1	39.20%	28.67%	20.05%	80.41%	80.91%	81.15%
	3	77.29%	67.68%	58.77%	84.60%	84.68%	84.85%
XGBoost	1	33.29%	25.59%	18.94%	84.46%	85.32%	85.73%
	3	77.02%	71.98%	63.62%	86.11%	86.18%	86.52%

Table B.2 Comparison of FIPE− $\|\cdot\|_0$ and FIPE− $\|\cdot\|_1$ for pruning tree ensembles with 50 base learners.

			FIPE− $\|\cdot\|_0$				FIPE− $\|\cdot\|_1$				
Ensemble	max depth	Dataset	$\|w\|_0$	K	F1	Time (s)	$\|w\|_0$	K	F1	Time (s)	Acc
Gradient Boosting	1	Breast-Cancer	13	15	100%	3 704	16	11	100%	6	96.93%
		COMPAS	7	5	100%	105	7	5	100%	4	66.02%
		ELEC2	21	8	100%	77	21	7	100%	32	74.68%
		FICO	13	8	100%	3 979	13	9	100%	11	70.91%
		HTRU2	7	3	100%	3	8	3	100%	5	96.96%
		House-16H	36	27	100%	4 469	38	20	100%	86	84.05%
		Ionosphere	14	18	100%	3 302	15	15	100%	10	92.11%
		Pima-Diabetes	19	9	100%	2 001	19	9	100%	5	73.90%
		PoL	13	6	100%	133	13	6	100%	8	90.83%
		Seeds	10	11	100%	1 675	16	9	100%	15	91.90%
		Spambase	24	25	100%	8 319	26	19	100%	42	91.94%
	3	Breast-Cancer	27	28	100%	2 355	33	24	100%	673	96.20%
		COMPAS	15	17	100%	4 201	27	11	100%	48	66.56%
		ELEC2	48	22	100%	1 247	50	21	100%	948	81.24%
		FICO	35	22	100%	16 489	46	16	100%	229	71.54%
		HTRU2	33	31	100%	3 467	39	29	100%	2 452	96.91%
		House-16H	49	31	100%	9 637	50	29	100%	8 205	87.20%
		Ionosphere	27	44	100%	7 371	35	32	100%	4 174	92.96%
		Pima-Diabetes	33	34	100%	15 037	43	30	100%	3 658	73.12%
		PoL	35	29	100%	2 505	42	25	100%	1 002	95.86%
		Seeds	26	22	100%	4 444	29	19	100%	3 716	93.33%
		Spambase	36	34	100%	44 650	46	33	100%	8 829	93.64%
LightGBM	1	Breast-Cancer	14	14	100%	3 397	16	12	100%	6	95.33%
		COMPAS	9	5	100%	90	9	5	100%	4	66.27%
		ELEC2	21	9	100%	121	21	8	100%	38	74.66%
		FICO	14	9	100%	4 153	14	9	100%	9	71.11%
		HTRU2	7	6	100%	7	8	5	100%	8	97.80%
		House-16H	36	26	100%	3 210	38	19	100%	77	84.18%
		Ionosphere	13	14	100%	2 836	15	13	100%	7	93.80%
		Pima-Diabetes	15	7	100%	641	16	7	100%	3	76.23%
		PoL	11	6	100%	37	11	5	100%	6	90.39%
		Seeds	13	14	100%	3 174	20	9	100%	14	91.90%
		Spambase	23	20	100%	3 169	25	18	100%	30	91.92%
	3	Breast-Cancer	26	30	100%	2 286	32	22	100%	281	96.64%
		COMPAS	14	16	100%	3 731	24	10	100%	37	66.40%
		ELEC2	45	20	100%	964	49	18	100%	347	80.96%
		FICO	36	21	100%	15 850	46	15	100%	169	71.49%
		HTRU2	32	31	100%	3 375	36	25	100%	598	97.98%
		House-16H	49	28	100%	7 440	50	31	100%	8 813	87.00%
		Ionosphere	27	43	100%	6 373	35	37	100%	4 432	94.37%
		Pima-Diabetes	32	30	100%	8 523	42	27	100%	1 806	75.45%
		PoL	29	23	100%	1 800	36	17	100%	114	95.49%
		Seeds	26	19	100%	3 103	32	17	100%	2 215	93.33%
		Spambase	34	28	100%	28 477	45	29	100%	7 012	93.59%

Ensemble	max depth	Dataset	FIPE– $\|\cdot\|_0$ $\|w\|_0$	K	F1	Time (s)	FIPE– $\|\cdot\|_1$ $\|w\|_0$	K	F1	Time (s)	Acc
Random Forest	1	Breast-Cancer	9	6	100%	256	10	5	100%	5	94.01%
		COMPAS	8	7	100%	159	8	6	100%	8	65.44%
		ELEC2	16	14	100%	749	18	13	100%	79	74.19%
		FICO	8	9	100%	145	8	6	100%	10	69.88%
		HTRU2	25	34	100%	2 898	26	29	100%	105	97.01%
		House-16H	24	20	100%	2 348	26	17	100%	46	77.31%
		Ionosphere	23	49	100%	4 535	28	33	100%	388	84.23%
		Pima-Diabetes	18	18	100%	690	19	15	100%	20	71.43%
		PoL	17	18	100%	1 307	17	15	100%	33	85.01%
		Seeds	21	19	100%	1 222	24	14	100%	52	82.86%
		Spambase	26	35	100%	6 291	31	29	100%	295	83.11%
	3	Breast-Cancer	27	33	100%	3 006	34	27	100%	1 118	97.08%
		COMPAS	21	23	100%	5 245	33	14	100%	88	66.34%
		ELEC2	35	32	100%	8 519	41	31	100%	3 131	74.65%
		FICO	31	23	100%	5 539	37	19	100%	171	70.60%
		HTRU2	29	38	100%	23 797	39	35	100%	5 152	97.63%
		House-16H	48	32	100%	8 040	50	35	100%	8 710	82.93%
		Ionosphere	27	36	100%	6 413	36	26	100%	4 702	93.80%
		Pima-Diabetes	31	34	100%	38 345	43	29	100%	4 229	77.14%
		PoL	30	36	100%	6 473	34	37	100%	4 406	91.38%
		Seeds	26	24	100%	8 701	31	22	100%	7 438	88.57%
		Spambase	36	36	100%	55 919	47	31	100%	6 025	90.51%
XGBoost	1	Breast-Cancer	13	15	100%	3 497	15	11	100%	6	97.08%
		COMPAS	7	5	100%	87	7	5	100%	3	65.85%
		ELEC2	20	7	100%	77	20	6	100%	31	74.65%
		FICO	13	8	100%	3 623	14	8	100%	10	70.76%
		HTRU2	9	4	100%	5	10	4	100%	6	97.56%
		House-16H	38	26	100%	3 238	39	21	100%	100	84.14%
		Ionosphere	11	13	100%	1 785	12	9	100%	5	91.27%
		Pima-Diabetes	17	8	100%	1 809	18	7	100%	4	74.29%
		PoL	11	5	100%	36	11	5	100%	6	90.38%
		Seeds	9	9	100%	967	14	6	100%	8	91.43%
		Spambase	22	22	100%	6 454	24	18	100%	32	91.62%
	3	Breast-Cancer	24	28	100%	2 404	31	19	100%	117	96.93%
		COMPAS	14	16	100%	3 859	25	10	100%	43	66.44%
		ELEC2	46	19	100%	916	50	18	100%	400	80.88%
		FICO	35	22	100%	17 804	45	16	100%	179	71.47%
		HTRU2	32	28	100%	4 155	37	24	100%	421	97.31%
		House-16H	49	32	100%	9 384	50	27	100%	6 663	87.04%
		Ionosphere	27	42	100%	5 878	33	31	100%	3 351	92.39%
		Pima-Diabetes	31	28	100%	9 966	41	25	100%	1 295	73.25%
		PoL	31	24	100%	1 814	38	19	100%	127	95.56%
		Seeds	24	19	100%	1 641	29	15	100%	278	92.38%
		Spambase	34	34	100%	53 950	45	34	100%	8 309	93.51%

Finally, Table B.3 analyzes the scalability of FIPE– $\|\cdot\|_1$ to ensembles of size 100 and 200. Again, the results are consistent with those presented in Section 5.4. The amount of pruning as the number of base learners increases and tends to decrease with the maximum depth of the trees. Additionally, the relative size of the pruned ensemble is heavily dependent on

the dataset. For ensembles of size $M = 200$, FIPE$-\|\cdot\|_1$ can always prune the ensemble, even if the amount of pruning is relatively low for more complex datasets such as ELEC2 or House-16H.

Table B.3 Pruning with FIPE$-\|\cdot\|_1$ on tree ensembles with 100 and 200 base learners.

			$M = 100$					$M = 200$				
Ensemble	max depth	Dataset	$\|w\|_0$	K	F1	Acc	Time (s)	$\|w\|_0$	K	F1	Acc	Time (s)
Gradient Boosting	1	Breast-Cancer	21	17	100%	96.50%	22	24	23	100%	95.91%	73
		COMPAS	9	5	100%	66.54%	8	10	8	100%	66.53%	32
		ELEC2	31	11	100%	76.12%	159	48	16	100%	79.62%	492
		FICO	16	11	100%	71.02%	45	17	13	100%	71.28%	112
		HTRU2	22	12	100%	97.20%	78	40	25	100%	96.52%	380
		House-16H	67	45	100%	85.68%	1 122	108	66	100%	86.63%	4 122
		Ionosphere	25	28	100%	92.96%	113	41	40	100%	93.52%	571
		Pima-Diabetes	30	17	100%	73.77%	29	42	29	100%	73.25%	127
		PoL	18	9	100%	93.26%	34	29	15	100%	93.05%	130
		Seeds	29	13	100%	93.81%	52	37	14	100%	92.86%	131
		Spambase	39	37	100%	92.88%	558	66	59	100%	93.88%	2 584
	3	Breast-Cancer	63	39	100%	96.50%	5 202	113	63	100%	96.64%	43 245
		COMPAS	37	13	100%	66.35%	141	51	16	100%	66.34%	527
		ELEC2	99	44	100%	82.27%	15 053	194	64	100%	83.87%	48 243
		FICO	81	23	100%	71.44%	1 145	141	33	100%	71.20%	6 555
		HTRU2	78	53	100%	96.64%	20 307	153	71	100%	96.54%	31 339
		House-16H	100	57	100%	87.94%	54 952	191	79	100%	88.17%	9 291
		Ionosphere	62	59	100%	93.24%	24 024	114	69	100%	93.80%	42 281
		Pima-Diabetes	77	49	100%	71.82%	20 509	140	63	100%	71.69%	22 066
		PoL	80	50	100%	96.82%	18 574	149	87	100%	97.65%	9 716
		Seeds	54	29	100%	93.81%	32 568	103	50	100%	92.86%	28 115
		Spambase	83	32	100%	94.29%	9 570	145	50	100%	94.64%	26 551
LightGBM	1	Breast-Cancer	19	16	100%	95.77%	19	24	23	100%	95.62%	70
		COMPAS	10	6	100%	66.51%	11	11	8	100%	66.61%	30
		ELEC2	30	10	100%	75.89%	117	46	17	100%	79.52%	535
		FICO	16	11	100%	71.41%	33	17	13	100%	71.58%	105
		HTRU2	20	13	100%	97.84%	75	32	24	100%	97.86%	376
		House-16H	64	43	100%	85.57%	1 030	102	74	100%	86.56%	5 067
		Ionosphere	23	25	100%	93.80%	57	36	41	100%	93.80%	588
		Pima-Diabetes	27	15	100%	76.36%	22	38	24	100%	76.36%	92
		PoL	17	9	100%	92.85%	29	26	14	100%	93.29%	99
		Seeds	30	12	100%	93.81%	39	37	16	100%	93.33%	114
		Spambase	37	32	100%	93.16%	345	60	52	100%	94.03%	1 782
	3	Breast-Cancer	64	39	100%	96.93%	4 481	116	58	100%	96.50%	22 225
		COMPAS	40	13	100%	66.63%	142	51	15	100%	66.60%	513
		ELEC2	97	41	100%	81.71%	12 631	191	65	100%	83.03%	55 810
		FICO	81	23	100%	71.63%	1 227	136	31	100%	71.41%	7 335
		HTRU2	69	45	100%	98.01%	12 332	136	66	100%	98.03%	44 612
		House-16H	100	46	100%	87.92%	36 836	192	65	100%	88.22%	37 741
		Ionosphere	66	46	100%	93.80%	12 249	118	47	100%	93.80%	16 319
		Pima-Diabetes	78	42	100%	74.94%	10 830	141	58	100%	73.25%	32 897
		PoL	74	46	100%	96.31%	9 341	144	62	100%	97.28%	40 006
		Seeds	58	23	100%	93.33%	14 620	110	36	100%	93.33%	48 061
		Spambase	82	32	100%	94.44%	9 762	144	43	100%	95.07%	24 161

			$M = 100$					$M = 200$				
Ensemble	max depth	Dataset	$\|w\|_0$	K	F1	Acc	Time (s)	$\|w\|_0$	K	F1	Acc	Time (s)
Random Forest	1	Breast-Cancer	11	6	100%	94.31%	12	12	8	100%	94.74%	35
		COMPAS	9	6	100%	65.54%	17	9	8	100%	65.20%	44
		ELEC2	25	20	100%	74.27%	309	35	30	100%	74.26%	939
		FICO	10	6	100%	70.15%	24	11	7	100%	70.12%	60
		HTRU2	41	46	100%	97.08%	527	63	76	100%	97.07%	2 277
		House-16H	41	30	100%	78.32%	335	56	49	100%	78.41%	1 329
		Ionosphere	42	53	100%	83.10%	1 708	60	70	100%	83.10%	3 123
		Pima-Diabetes	30	26	100%	71.56%	87	42	35	100%	71.30%	368
		PoL	23	28	100%	88.97%	203	30	39	100%	88.98%	747
		Seeds	41	22	100%	83.81%	189	62	32	100%	86.19%	674
		Spambase	44	40	100%	82.93%	1 037	61	61	100%	83.24%	2 570
	3	Breast-Cancer	60	45	100%	97.23%	6 405	112	70	100%	96.93%	22 087
		COMPAS	48	16	100%	66.37%	241	58	16	100%	66.32%	640
		ELEC2	72	56	100%	74.73%	15 962	127	89	100%	74.61%	56 155
		FICO	65	28	100%	70.68%	655	116	36	100%	70.81%	2 083
		HTRU2	67	53	100%	97.70%	21 656	119	71	100%	97.69%	41 486
		House-16H	98	57	100%	82.88%	35 409	173	97	100%	82.89%	11 227
		Ionosphere	62	32	100%	94.37%	6 405	115	37	100%	94.65%	9 115
		Pima-Diabetes	73	47	100%	76.88%	18 180	126	63	100%	76.88%	43 545
		PoL	61	55	100%	91.20%	17 642	111	82	100%	91.34%	45 545
		Seeds	57	34	100%	88.57%	31 675	98	49	100%	90.21%	30 249
		Spambase	81	36	100%	90.92%	9 801	139	43	100%	90.99%	21 500
XGBoost	1	Breast-Cancer	19	16	100%	97.08%	17	21	18	100%	96.20%	45
		COMPAS	9	5	100%	66.53%	8	10	8	100%	66.54%	29
		ELEC2	29	9	100%	75.91%	123	44	16	100%	79.44%	472
		FICO	16	10	100%	71.01%	38	17	12	100%	71.35%	93
		HTRU2	18	10	100%	97.68%	62	36	21	100%	97.64%	311
		House-16H	65	46	100%	85.63%	1 179	102	63	100%	86.52%	3 743
		Ionosphere	22	23	100%	92.96%	54	35	39	100%	92.96%	440
		Pima-Diabetes	27	14	100%	74.16%	21	39	24	100%	73.38%	92
		PoL	17	11	100%	92.86%	39	26	15	100%	93.26%	111
		Seeds	21	10	100%	91.90%	31	28	12	100%	92.38%	75
		Spambase	38	36	100%	92.77%	503	58	47	100%	93.40%	1 444
	3	Breast-Cancer	55	32	100%	96.64%	1 241	93	46	100%	96.50%	5 514
		COMPAS	40	14	100%	66.47%	151	50	15	100%	66.43%	466
		ELEC2	97	39	100%	81.86%	11 331	188	66	100%	83.28%	50 598
		FICO	81	25	100%	71.39%	1 168	138	32	100%	71.33%	6 175
		HTRU2	75	50	100%	97.31%	15 827	144	64	100%	97.30%	23 872
		House-16H	100	52	100%	87.81%	45 517	190	96	100%	88.13%	11 554
		Ionosphere	61	49	100%	91.83%	12 418	97	50	100%	92.68%	13 430
		Pima-Diabetes	74	44	100%	72.34%	12 699	134	58	100%	71.69%	38 776
		PoL	75	47	100%	96.26%	8 204	142	67	100%	97.21%	39 919
		Seeds	54	25	100%	91.43%	2 615	82	35	100%	91.90%	11 310
		Spambase	81	34	100%	94.59%	11 413	142	45	100%	95.24%	26 334