Quadratic Maximum-Weight Independent Set Problems (Q-MWIS)

Eldar Keller

Master Thesis

Supervisors: PD Dr. Bogdan Savchynskyy
Interdisciplinary Center for Scientific Computing
Ruprecht-Karls-University Heidelberg

Prof. Dr. Ekaterina A. Kostina
Faculty of Mathematics and Computer Science
Ruprecht-Karls-University Heidelberg

January 31, 2024

Abstract

This thesis introduces the Quadratic Maximum-Weight Independent Set (Q-MWIS) problem as the quadratic extension of the Maximum-Weight Independent Set (MWIS) problem. Its analysis is motivated by subproblems arising in algorithms addressing an NP-hard multi-graph matching problem [1].
Following the discussion of its properties, two linearizations of the problem are constructed as part of a potential solution algorithm: A "trivial" linearization and one based on a "reformulation-linearization technique" (RLT) of Sherali and Adams [2], which leads to varying degrees of tightness of the corresponding LP relaxation. We show that some of the resulting constraints for the latter can be redundant, suggesting a more concise formulation and confirm that it is indeed a valid reformulation of the original problem, with a tighter LP relaxation than the one of the "trivial" linearization.
Lastly, we look deeper into the practical part of solving the Q-MWIS problem and focus on two approaches: A primal-dual algorithm, which creates approximate solutions by solving Lagrange dual problems and recombining them in an "optimized crossover" heuristic, and employing off-the-shelf ILP solvers. We finish up with tests on which of the previously constructed formulations is preferable for problems of varying size and structure, when using the optimization software Gurobi [3] – finding that the Sherali-Adams linearization outperforms the trivial linearization in almost all cases and yields results quicker than the default QIP formulation in most problem instances, particularly with nonnegative costs.

Zusammenfassung

Diese Masterarbeit führt das quadratische Maximum-Weight Independent Set (Q-MWIS) Problem ein – eine Erweiterung des linearen MWIS Problems um quadratische Terme. Die Analyse dieses Problems wird durch Subprobleme motiviert, welche in Algorithmen zum NP-schweren multi-graph matching Problem auftreten [1].
Nach der Einführung des Problems werden, als potenzieller Teil eines Algorithmus für Q-MWIS Probleme, zwei Linearisierungen konstruiert: Eine "triviale" und eine basierend auf die Reformulierungs-Linearisierungstechnik von Sherali und Adams [2]. Nachdem wir Letztere auf unser Problem angewandt haben, zeigen wir, dass manche der resultierenden Ungleichungen in bestimmten Fällen redundant sind und formulieren eine prägnantere Linearisierung.
Im darauffolgenden praktischen Teil, behandeln wir, wie Q-MWIS Probleme gelöst oder approximiert werden können. Hierfür, ziehen wir einen primal-dual Algorithmus in Betracht, welcher beim lösen eines Lagrange dual Problems suboptimale Lösungen erzeugt und diese auf optimale Weise rekombiniert. Anschließend prüfen wir, welche der zuvor konstruierten Reformulierungen des Q-MWIS Problems beim Einsatz üblicher Optimierungssoftware die besten Resultate liefert. Hierfür werden Testprobleme mit unterschiedlichen Strukturen generiert und mit Hilfe von Gurobi [3] unter Nutzung der verschiedenen (Re-)Formulierungen gelöst. Dabei zeigt sich, dass die Sherali-Adams Linearisierung fast durchgehend bessere Resultate liefert als die triviale Linearisierung oder die Originalformulierung – mit Ausnahme von Problemen, mit teilweise negativen Kostentermen.

Table of Contents

Notation

$\{0,1\}^n$; $[0,1]^n$	sets of elements $(x_1,\dots,x_n)$, respectively with x_i binary or $x_i \in [0,1]$, $\forall i = 1,\dots,n$
$\langle c, x\rangle$	scalar product of vectors $c, x \in \mathbb{R}^n$, given by $\sum_{i=1}^n c_i x_i$
$\lvert\mathcal{V}\rvert$	cardinality of a set $\mathcal{V}$
$\mathbf{vrtx}(P)$	set of vertices for a polytope P
$\mathbf{conv}(X)$	convex hull of the set X
$\mathbb{R}_{\geq 0}$	space of nonnegative real numbers
$\partial f(x^0)$	set of subgradients for convex function f in x^0, a point in its domain
$[n]$	$\{1,\dots,n\}$, enumeration set for $n \in \mathbb{N}$ elements
$[[n]]^2$	$\{ik \mid i,k \in [n], i < k\}$ index pair set, with sub-indices from set $[n]$
N_Z	set of all label pairs with non-zero pairwise cost, i.e. $N_Z \subseteq [[n]]^2$, with $c_{ik} \neq 0,\ \forall ik \in N_Z$
$F_d(J_1,J_2)$	polynomials of the form $\left(\prod_{j\in J_1} x_j\right)\left(\prod_{j\in J_2}(1-x_j)\right)$, where (J_1,J_2) are subsets *of order* d, i.e. $J_1, J_2 \subseteq [n]$, $J_1 \cap J_2 = \emptyset$ and $\lvert J_1 \cup J_2\rvert = d$
N_d	total number of subset pairs (J_1,J_2) of order d
$f_d(J_1,J_2)$	polynomials $F_d(J_1,J_2)$, after applying the relationship $x_i^2 = x_i,\ \forall i \in [n]$ and linearizing the remaining quadratic terms
X	feasible set of the original problem; $X \subseteq \mathbb{R}^n$ with $n \in \mathbb{N}$
X_d	polyhedral set of feasible elements after Sherali-Adams transformations of order d on the original feasible set $X \subseteq \mathbb{R}^n$; $X_d \subseteq \mathbb{R}^n \times \mathbb{R}^{m_d}$ with $n, m_d \in \mathbb{N}$
X_{P^d}	projection of set $X_d \subseteq \mathbb{R}^n \times \mathbb{R}^{m_d}$, for $n, m_d \in \mathbb{N}$, on the realm of the original feasible set $X \subseteq \mathbb{R}^n$, given by $\{x \mid (x,w) \in X_d,\ x \in \mathbb{R}^n,\ w \in \mathbb{R}^{m_d}\}$
$\{A_i\}_{i\in[n]}$	set of sets $A_1,\dots,A_n$
$(x_i)_{i\in[n]}$	$(x_1,\dots,x_n)$, the vector indexed by the set $[n]$

Abbreviations

s.t.	subject to
w.l.o.g.	without loss of generality
RLT	Reformulation-Linearization Technique
MWIS	Maximum-Weight Independent Set
Q-MWIS	Quadratic Maximum-Weight Independent Set
ILP	(0-1) Integer Linear Program(ming)
MIP	Mixed Integer Program(ming)
QIP	Quadratic Integer Program(ming)
LP relaxation	Linear Programming relaxation, acquired by relaxing the integrality constraints of an ILP
QUBO	Quadratic Unconstrained Binary Optimization
QAP	Quadratic Assignment Problem
MAP-Inference	Maximum A Posteriori (MAP) inference for graphical models

0 Preliminaries

Before heading on to the main contents of the thesis, we state a few definitions of important notions used therein.

Definition 0.1 (Linear Programming)**.** Let P be a (convex) polyhedron of the form $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, with $A \in \mathbb{R}^{m+n}$ and $b \in \mathbb{R}^m$. Then, optimization problems of the form

$$\max_{x \in P} \ \langle c, x \rangle \quad \text{or} \quad \min_{x \in P} \ \langle c, x \rangle \tag{0.1}$$

with a cost vector $c \in \mathbb{R}^n$, are called *linear programs* (LP). The maximized/minimized term $\langle c, x \rangle$ is its *objective function*, P its *feasible set* and any $x \in P$ is called a *feasible solution* to it.

Hence, the goal of such linear programs is to find a solution $x \in \mathbb{R}^n$, which satisfies the (in-)equality constraints defined by the polyhedral set P and maximizes/minimizes the objective function. We can see that this constitutes a convex optimization problem, since feasible set P is convex and the objective function is linear – thus both concave and convex. This leads to some favorable properties, such as locally optimal solutions (local maxima/minima) also being globally optimal.

When the feasible set of a linear program is restricted to binary solutions, one yields the following concept:

Definition 0.2 (Integer Linear Programming)**.** Consider a linear program as defined in (0.1). The combinatorial optimization problem that is acquired by adding *integrality constraints* of the form $x \in \{0, 1\}^n$, which restrict the feasible solutions to be binary, is given by

$$\max_{x \in P \cap \{0,1\}^n} \ \langle c, x \rangle \quad \text{or} \quad \min_{x \in P \cap \{0,1\}^n} \ \langle c, x \rangle \tag{0.2}$$

and called a *(0-1) integer linear program* (ILP).

It is clear that restricting the feasible set to binary values leads to non-convexity, which makes the problem harder to solve. While linear programs can generally be solved "efficiently" in polynomial time (see for example [4]), the same doesn't hold for integer linear programs, which can be polynomially solvable in special cases, but are NP-hard in the general case. Some NP-hard problem instances, which can be represented as an integer linear program, are the binary knapsack and maximum a posteriori (MAP) inference problem for graphical models, as described in [5].

A widely used relaxation of ILPs is given by relaxing their integrality constraints:

Definition 0.3 (Linear Programming Relaxation)**.** The *linear programming* (LP) *relaxation* of the integer linear programs described in (0.2) is given by

$$\max_{x \in P \cap [0,1]^n} \langle c, x \rangle \quad \text{or} \quad \min_{x \in P \cap [0,1]^n} \langle c, x \rangle. \tag{0.3}$$

It is acquired by replacing the integrality constraints $x \in \{0,1\}^n$ with *box constraints* $x \in [0,1]^n$.

Since both P and the set $\{x \mid x \in [0,1]^n\}$ are convex, the same holds for their intersection. Therefore, as the name suggests, the LP relaxation is a linear program, making it easier to solve than the original integer linear program. Its feasible set is a *polytope* (a bounded polyhedron).

We note the following properties of integer linear programs and their LP relaxations, all of which are covered in chapter 3 of [5]:

1. Optimal solutions to linear programs, where the feasible set is a polytope, can be acquired in the vertices thereof. Exemplary for the LP relaxation in (0.3), this means

$$\max_{x \in P \cap [0,1]^n} \langle c, x \rangle = \max_{x \in \mathrm{vrtx}(P \cap [0,1]^n)} \langle c, x \rangle. \tag{0.4}$$

2. Optimal solutions for the LP relaxation can incorporate fractional values, which are infeasible for the underlying integer linear program. However, if an optimal solution to the LP relaxation contains only binary values, it is also optimal for the ILP. In this case, the LP relaxation is said to be *LP-tight*.

3. Let $\mathrm{conv}(X)$ denote the *convex hull* of a finite set X, i.e. the smallest convex set containing X. For any integer linear program as in (0.2), it holds that

$$\max_{x \in P \cap \{0,1\}^n} \langle c, x \rangle = \max_{x \in \mathrm{conv}(P \cap \{0,1\}^n)} \langle c, x \rangle, \tag{0.5}$$

which means any integer linear program could be solved as a linear program, by instead considering the convex hull of its feasible set $P \cap \{0,1\}^n$. This doesn't conflict with the differing computational complexities of ILP and LP problems, as describing the convex hull may take an exponential amount of linear inequalities, which prevents finding a solution in polynomial time for the linear program described in (0.5).

4. Combining the previous properties, we can follow: An integer linear program as described in (0.2) can be exactly solved by its LP relaxation, if the following property holds for its feasible set:

$$P \cap \{0,1\}^n = \mathrm{vrtx}(P \cap [0,1]^n), \tag{0.6}$$

i.e. when the vertices of the feasible set for the LP relaxation are exactly the solutions feasible to the ILP. While $P \cap \{0,1\}^n \subseteq \mathrm{vrtx}(P \cap [0,1]^n)$ holds for any polyhedron P, the opposite direction does not hold generally.

As the set $P \cap \{0,1\}^n$ can be equivalent for many different polyhedra P, these properties highlight the importance of the ILP problem description: While many problem descriptions can be used for the same problem instance, some yield a tighter LP relaxation than others, by allowing fewer fractional vertices, which are infeasible for the original ILP.
This fact naturally leads to algorithms, which aim to restrict the polyhedron P by removing non-integer elements, such that problems become easier to solve through their LP relaxation:

Definition 0.4 (Cutting-plane Method)**.** Consider the LP relaxation as described in Definition 0.3. Algorithms, which aim to eliminate fractional vertices from its feasible set $P \cap [0,1]^n$ through the addition/manipulation of constraints describing the Polyhedron P, are referred to as *cutting-plane methods*. Constraints added for this purpose are called *cuts*.

The first such cutting-plane method to solve ILPs was suggested in [6] by Gomory. While methods using only Gomory cuts have limited use in practice, due to poor convergence properties to the solution, they are still widely used in modern branch-and-cut algorithms [7], where cuts are used in a branch and bound framework, as a means to decrease the number of branching nodes.

Since then, several other methods to generate cuts have been suggested, one of which is closely related to a linearization technique we will employ in the thesis – namely, *lift-and-project cuts*. A cutting-plane algorithm using such cuts to solve mixed integer programs (MIP), which may contain both integer and real variables, has been formulated in [8] and shown to outperform Gomory cuts. Similarly, lift-and-project cuts have been incorporated in branch-and-cut algorithms [9].
In general, these methods aim to create a problem description, which results in an LP relaxation that is as tight as possible, without being prohibitively large to use.

Besides the fairly straightforward LP relaxation, another commonly used relaxation technique for optimization problems removes some constraints and instead adds them with a cost to the objective function. We define it exemplary for a maximization ILP problem:

Definition 0.5 (Lagrange Relaxation)**.** Let P be some polytope and $Ax \leq b$ define a system of linear inequalities with $A \in \mathbb{R}^{m,n}$, $b \in \mathbb{R}^m$, for an ILP with costs $c \in \mathbb{R}^n$, given by

$$\begin{aligned} &\max_{x \in P \cap \{0,1\}^n} \langle c, x \rangle \\ &\text{subject to (s.t.)} \quad Ax \leq b. \end{aligned} \tag{0.7}$$

The relaxations to this problem, which we acquire by dualizing the constraints $Ax \leq b$, i.e. omitting them and instead adding them to the objective function of (0.7), are called *Lagrange relaxations*

$$\max_{x \in P \cap \{0,1\}^n} \langle c, x \rangle + \langle \lambda, b - Ax \rangle =: \mathcal{D}(\lambda), \tag{0.8}$$

with (nonnegative) *Lagrange multipliers* $\lambda \in \mathbb{R}^m_{\geq 0}$. The function mapping all possible Lagrange multipliers to the relaxations (0.8) is defined as the *Lagrange dual function* $\mathcal{D}(\lambda)$.

As it is favorable to find the tightest relaxation possible, one considers the problem of finding the Lagrange multiplier λ that minimizes the upper bounds, which we acquire from solving the Lagrange relaxations. It is called the *Lagrange dual problem* and defined as

$$\min_{\lambda \in \mathbb{R}^m_{\geq 0}} \mathcal{D}(\lambda) = \min_{\lambda \in \mathbb{R}^m_{\geq 0}} \max_{x \in P \cap \{0,1\}^n} \langle c, x \rangle + \langle \lambda, b - Ax \rangle. \tag{0.9}$$

The Lagrange dual problem (0.9) is equivalent to the *primal relaxed problem*, which is given by

$$\max_{\substack{x \in \operatorname{conv}(P \cap \{0,1\}^n) \\ Ax \leq b}} \langle c, x \rangle \tag{0.10}$$

and generally tighter than the LP relaxation, unless it is LP-tight, in which case they are equivalent (see corollary 5.44 in [5]).

When considering a primal-dual algorithm later in this work, we will incorporate the subgradient method to solve the Lagrange dual problem. Thus, we briefly cover the definitions of subgradient and subgradient method here, then state its convergence properties.

Definition 0.6 (Subgradient)**.** Let f be a convex function and x^0 any point on its domain. Then, a vector g is called a *subgradient of* f *in* x^0, if

$$\forall x \in \text{dom}(f): \ f(x) \geq f(x^0) + \langle g, x - x^0 \rangle \tag{0.11}$$

holds and we denote the set of subgradients of f in x^0 as $\partial f(x^0)$.

Subgradients can be used to find minima of convex functions, without requiring them to be differentiable:

Definition 0.7 (Subgradient Method)**.** Let $x^0 \in \mathbb{R}^n$ be any starting vector and $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. For steps $t \in \mathbb{N}_0$ and a sequence of stepsizes $\{\alpha^t\}_{t=0}^{\infty}$, which satisfies the criteria

$$\lim_{t\to\infty} \alpha^t = 0 \quad \text{and} \quad \sum_{t=0}^{\infty} \alpha^t = \infty, \tag{0.12}$$

the *subgradient method* describes the process

$$x^{t+1} = x^t - \alpha^t g(x^t), \tag{0.13}$$

with subgradients $g \in \partial f(x^t)$.

As shown in [10, Thm. 3.2.2], the subgradient method converges to the minima of convex functions, which are locally lipschitz-continuous around their optimum, when choosing certain stepsizes α^t. Generally, they should always fulfill the diminishing stepsize rules (0.12) for the method to converge.
In practice, choosing an appropriate stepsize is of great importance, but the best choice of stepsizes may vary between problem types and instances. Therefore, they usually have to be tested and adjusted for the problems at hand, in order to ensure good performance of the subgradient method.

Following these definitions and basic properties of (integer) linear programs and their relaxations, as well as the subgradient method, we proceed with the introduction of the MWIS problem.

1 Introduction

In this section, we will lay the foundation for the upcoming work, by starting out with the definition of the Maximum-Weight Independent Set Problem (MWIS), its computational complexity and current approaches for solving such problems.

Following that, we motivate the quadratic extension thereof in Section 2, coining the term Quadratic Maximum-Weight Independent Set Problem (Q-MWIS). As a preparation for the practical part, where we consider means to efficiently solve Q-MWIS problems or get good approximations, we will apply two different linearization methods. While the first, "trivial" linearization simply adds variables and enforces them to be equal to the quadratic terms through extra constraints, the second technique also aims to tighten the LP relaxation in the process – without losing sight of the feasibility in practical settings. To achieve this, we apply a *reformulation-linearization technique* (RLT), suggested by Sherali-Adams in [2] and clear the outcome of redundant constraints for a more concise linearization of the Q-MWIS problem.

We finish in Section 3, by considering ways how to solve Q-MWIS problems in practice. The initial idea is to incorporate the previously constructed linearization in a primal-dual algorithm, where we aim to solve the corresponding Lagrange dual problem through the subgradient method and use the approximate solutions it supplies for primal heuristics. As this proves to be unlikely to provide an efficient solution algorithm, we briefly contemplate on alternatives, involving off-the-shelf ILP solvers.
For the latter, we test which of the different problem (re-)formulations we construct in Section 2 performs best, when used with the optimization software Gurobi [3], on problems of varying size and structure.

1.1 The Maximum-Weight Independent Set Problem (MWIS)

As indicated by the name, the *Maximum-Weight Independent Set (MWIS) problem* describes the problem of finding an independent subset of elements with maximum weight/cost. Formally, it can be defined as an integer linear programming (ILP) problem:

Definition 1.1 (Maximum-Weight Independent Set)**.** Let $[n] := \{1, \dots, n\}$ be the numbering for the elements we can choose from, which we will refer

to as *labels* and K_j be subsets thereof, with $j \in [m]$ indexing the m so-called *conflict sets*. If an element with label i is chosen, the corresponding binary decision variable x_i is set to 1 and the cost $c_i \in \mathbb{R}$ is added to the total resulting cost.
In total, for a cost vector $c \in \mathbb{R}^n$, the maximization problem is given by

$$\max_{x \in \{0,1\}^n} \langle c, x \rangle = \max_{x \in \{0,1\}^n} \sum_{i=1}^{n} c_i x_i \tag{1.1}$$

$$\text{s.t.} \sum_{i \in K_j} x_i \leq 1, \ \forall j \in [m]. \tag{1.2}$$

We can see from the constraints in (1.2) that feasible solutions can only pick at most one of the labels contained in each conflict set. A subset of all labels $S \subseteq [n]$, with $x_i = 1$ for all $i \in S$ and zero otherwise, which fulfills the constraints (1.2) for all conflict sets K_j, is called an *independent set*.

To sum it up: In order to solve these problems, we have to find such independent sets not violating the conflict set constraints, while maximizing cost. It is evident that if $c_i < 0$ for any label $i \in [n]$, $x_i = 0$ holds for any optimal solution. Therefore, these terms could theoretically be omitted and the corresponding decision variables set to 0 by default.

Remark 1.2 (Conflict set representations)**.** It should be mentioned that the MWIS problem usually does not have a unique set of conflict sets, but can have different conflict set representations. For example, three sets $K_1 = \{1, 2\}$, $K_2 = \{2, 3\}$ and $K_3 = \{1, 3\}$ could equivalently be represented by $K = \{1, 2, 3\}$ and vice versa. While these differing formulations don't change the set of feasible solutions for the original integer linear program, they do affect the set of feasible solutions for the corresponding LP relaxation, as we have noted in the preliminaries.

We observe the following statement on conflict set representations, which can be seen as a generalization of Remark 1.2:

Proposition 1.3. *Let K_j with $j \in [m]$, and $K \subseteq [n]$ be conflict sets with non-zero $n, m \in \mathbb{N}$ and $K_j \subseteq K$, $\forall j \in [m]$. If all label pairs contained in K are also both contained in some conflict set K_j, i.e. if*

$$\forall i, k \in K, \ \exists j \in [m] : i, k \in K_j \tag{1.3}$$

holds, the following inequalities are equivalent for binary decision variables $x_i \in \{0, 1\}$, $\forall i \in [n]$:

$$\sum_{i \in K} x_i \leq 1 \quad \Leftrightarrow \quad \sum_{i \in K_j} x_i \leq 1, \ \forall j \in [m]. \tag{1.4}$$

Proof. " $\Rightarrow$ " : From $K_j \subseteq K$, $\forall j \in [m]$ and the decision variables being binary (i.e. nonnegative), we can immediately see that

$$\sum_{i \in K_j} x_i \leq \sum_{i \in K} x_i \leq 1, \ \forall j \in [m]. \tag{1.5}$$

" $\Leftarrow$ " : (Proof by contradiction) Assume that $\sum_{i \in K} x_i > 1$ holds. Since the decision variables are binary, we have at least one label pair $i, k \in K$, with $x_i = x_k = 1$ and $i \neq k$. From condition (1.3), we know that there also exists some $j \in [m]$, such that $i, k \in K_j$, resulting in $\sum_{i \in K_j} x_i > 1$, since $x_i \geq 0$, $\forall i \in [n]$. This contradicts the assumption $\sum_{i \in K_j} x_i \leq 1, \ \forall j \in [m]$ and $\sum_{i \in K} x_i \leq 1$ follows. □

Proposition 1.3 can be used as a means of constructing a single larger conflict set, in place of multiple smaller ones. This both yields a tighter LP-relaxation and simultaneously reduces the number of constraints, resulting in a preferable MWIS problem formulation.

Remark 1.4 (Graph theoretic interpretation)**.** The MWIS problem can also be seen as a graph theoretic problem:
Consider a graph $(\mathcal{V}, \mathcal{E})$, with set of vertices $\mathcal{V}$ and edges $\mathcal{E}$ connecting the vertices. Additionally, every vertex has a weight/cost attached to it. The problem then consists of finding the subset S of $\mathcal{V}$ that sums to the largest total weight, such that there is no edge connecting any of its vertices. In this scenario, each edge can be thought of as a conflict set containing two elements, namely, the vertices it connects. Alternatively, the conflict sets can be constructed as clique edge covers, resulting in fewer, but larger conflict sets. It may be noted that the subset S also forms an independent set in the graph $(\mathcal{V}, \mathcal{E})$.

In the following work, we will also use an alternative way to formulate the constraints (1.2), replacing the inequality constraints with equations. When incorporating quadratic terms later on, this will make the constraints easier to work with and allow us to connect the quadratic MWIS problem to other similar combinatorial optimization problems.

To this end, we will introduce *slack decision variables* $x_{n+1}, \ldots, x_{n+m}$ with zero costs $c_{n+1}, \ldots, c_{n+m}$ to (1.1) – one for every constraint. Each label to these slack elements is added to exactly one conflict set of the original formulation, resulting in new conflict sets $K'_j := K_j \cup \{n + j\}$ for $j \in [m]$.

Definition 1.5 (MWIS with equality constraints)**.** Let $x_{n+1}, \ldots, x_{n+m}$ be additional decision variables with costs $c_{n+1} = \cdots = c_{n+m} = 0$ and define the conflict sets $K'_j = K_j \cup \{n+j\}, \ \forall j \in [m]$. The *equality constraint reformulation of the MWIS problem* described in Definition 1.1 reads:

$$\begin{aligned} &\max_{x \in \{0,1\}^{n+m}} \langle c, x \rangle \\ \text{s.t. } &\sum_{i \in K'_j} x_i = 1, \ \forall j \in [m]. \end{aligned} \tag{1.6}$$

Since they are unweighted and exactly one such slack variable is added per conflict set, it can be seen that the the objective function value, as well as the optimal solution to the problem, remain unchanged.

Indeed, we observe that depending on the summed decision variables in the underlying conflict set, the slack variables are set to zero or one, allowing us to turn the conflict set inequalities into equations:

$$\forall j \in [m] \colon x_{n+j} = \begin{cases} 0, & \text{if } \sum_{i \in K_j} x_i = 1, \\ 1, & \text{otherwise.} \end{cases} \tag{1.7}$$

1.2 Computational Complexity of the MWIS problem

In order to solve MWIS problems, one has to find an independent set of maximum possible weight. Considering the exponential 2^n possible combinations of n decision variables, it can be seen that brute-force algorithms are at most feasible for small-scale problems. Indeed:

Proposition 1.6. *The MWIS problem is* ***NP-hard*** *in the general case.*

Proof. As noted in [11], the independent set decision problem is NP-complete. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the problem consists of answering the question, whether there is an independent subset $\mathcal{V}' \subseteq \mathcal{V}$ with cardinality $|\mathcal{V}'| \geq J$ for some positive integer $J \leq |\mathcal{V}|$. It is readily observed that the independent set problem can be reduced to the MWIS problem in polynomial time, using its graph theoretic interpretation mentioned in Remark 1.4:

Let $x_1, \ldots, x_n$ be the decision variables for the $n = |\mathcal{V}|$ elements of the MWIS problem and set the costs of all elements to 1. The $m = |\mathcal{E}|$ conflict sets are defined by the edges:

$$\forall j \in [m] : K_j = \{k, l\}, \text{ for } (v_k, v_l) \in \mathcal{E}. \tag{1.8}$$

In particular, an independent set in the MWIS problem corresponds to an independent set in the independent set problem and vice versa. Because the costs are identical, an independent set with maximum cost is also a maximum independent set (i.e., the largest independent set in $\mathcal{V}$). Additionally, as the costs are 1 for every element, the optimal value of the MWIS problem we described is equal to the cardinality of the maximum independent set J_{max}. Therefore, the answer to the independent set problem is "yes", if $J \leq J_{max}$ and "no" otherwise.
Since the NP-complete independent set problem thus can be reduced to the MWIS problem, the latter is shown to be NP-hard. □

Despite it not being proven, due to the NP-hardness of the MWIS problem it is unlikely that there exists a polynomial time algorithm that can exactly solve all problem instances, as this would imply that P = NP.

Furthermore, it has been shown in [12] that (under the assumption that NP≠ZPP) there is no polynomial time algorithm that can approximate the max-clique problem for a graph with n vertices within a factor of $n^{1-\epsilon}$, for some constant $\epsilon > 0$. This means that for the maximum clique size X_{opt}, there is no polynomial time algorithm that returns at most X_{opt} and at least $\frac{X_{opt}}{n^{1-\epsilon}}$. Similar results are derived for an approximation factor of $n^{1/2-\epsilon}$, if one instead assumes that P≠NP.
As the max-clique problem can be easily transformed to the unweighted maximum independent set problem by complementing the graph – i.e. connecting all vertices that weren't connected by an edge previously and removing the old edges – these theorems extend to the MWIS problem.

In summary, we can see that the MWIS problem is both hard to solve exactly and to approximate. However, there are different approaches attempting to do so as efficiently as possible.

1.3 Finding exact or approximate solutions for the MWIS problem

Some current algorithms that aim at exactly solving general MWIS problems employ branch-and-bound or branch-and-reduce techniques [13, 14]. When exact methods are infeasible due to the large scale of a problem instance, heuristic algorithms using local search methods can be used to find approximate solutions [15].

In addition to this, some algorithms are tailored to special MWIS problem classes, which can be solved in polynomial time. These classes include claw-free [16, 17], fork-free [18] or P_6-free graphs [19].

Aside from these algorithms specifically tailored to the MWIS problem, one may also take a more general approach using ILP solvers, such as Gurobi [3]. The efficiency of such solvers heavily depends on the formulation of the constraints used in the problem description. Indeed, as mentioned in the preliminaries and Remark 1.2, there are many ways to describe the same problem, resulting in different amounts of constraints with varying tightness for the corresponding LP relaxation.

To illustrate this, consider the following basic example:

Example 1.7. Let $K_1 = \{1, 2\}$, $K_2 = \{2, 3\}$ and $K_3 = \{1, 3\}$ be a conflict set representation for a MWIS problem with costs $c_1 = 2$, $c_2 = 2$ and $c_3 = 3$. As previously described, these three conflict sets could be replaced by a single one, namely $\hat{K} = \{1, 2, 3\}$. We can see that there is a unique independent set of maximum cost for both representations, namely the single element set $\{3\}$ with a cost of 3. But, while both yield the same result for the original integer linear programming problem, they result in different LP relaxations. The relaxations are given by

$$\begin{aligned} \max_{x\in[0,1]^3} \sum_{i=1}^{3} c_i x_i,\ \text{subject to (s.t.)} \\ x_1 + x_2 \leq 1, \\ x_2 + x_3 \leq 1, \\ x_1 + x_3 \leq 1 \end{aligned} \tag{1.9}$$

and

$$\begin{aligned} \max_{x\in[0,1]^3} \sum_{i=1}^{3} c_i x_i,\ \text{s.t.} \\ x_1 + x_2 + x_3 \leq 1 \end{aligned} \tag{1.10}$$

respectively.

It is evident that the single constraint in (1.10) implies all three constraints of (1.9), but the same doesn't hold in reversed order: For instance, values $x_1 = x_2 = x_3 = 0.5$ would be a feasible solution for the LP relaxation in (1.9), while they're infeasible for the single constraint LP relaxation. This implies that the relaxation in (1.10) is strictly tighter for the original ILP, than the one described in (1.9).

One can also observe this, when solving these linear programs: The LP relaxation (1.9) yields an optimal value of 3.5 for the optimal solution given by $x_1 = x_2 = x_3 = 0.5$, while the linear program in (1.10) is a tight relaxation with an optimal value of 3, and optimal solution $x_1 = x_2 = 0$, $x_3 = 1$. As such, the single constraint representation of the problem is clearly preferable here, when solving the integer linear programming problem using its LP relaxation.

Remark 1.8 (Connection of the constraint formulation and LP relaxation)**.** As illustrated in Example 1.7, the use of appropriate constraints is of relevance, when trying to solve MWIS or ILP problems through their LP relaxation, since they affect the tightness thereof. In order to eliminate fractional solutions from the LP relaxation, one may resort to adding inequalities, which are violated by them, without affecting the integer solutions. As mentioned in the preliminaries, constraints added for this purpose are referred to as *cuts* and there are several methods that allow the building of progressively tighter relaxations – at the cost of introducing more constraints.

One of such methods was described in [2] by Sherali and Adams, where they employ a reformulation-linearization technique (RLT), which adds constraints akin to the ones that were later used in the lift-and-project cutting plane method introduced in [8]. Through multiplication of the constraints with certain polynomials and subsequent linearization, relaxations of varying tightness can be constructed, resulting in a hierarchy of relaxations. This hierarchy ranges from the basic LP relaxation of the original problem to the convex hull of the feasible solutions for the ILP problem – therefore, solving the latter would also solve the ILP problem.
In practice, constructing the tightest relaxation of the hierarchy in such a way is infeasible for real-world applications, considering the exponentially increasing number of constraints. However, we will use this technique later on, constructing a lower-ranking relaxation in order to tighten the constraints, while retaining efficiency.

After this brief introduction of the MWIS problem, its computational complexity and current means of approaching such problems, we proceed with the introduction of its quadratic extension.

2 Quadratic Maximum-Weight Independent Set (Q-MWIS) Problems

While MWIS problems only contain a linear cost term for every decision variable by itself, some problems require the modeling of pairwise relations, such as the quadratic cost subproblems arising in the work of [1], where they address multi-graph matching problems. To this end, we will augment the MWIS problem with quadratic cost terms, transforming it from an integer linear programming problem into a *quadratic integer programming* (QIP) problem.

Similarly to the MWIS problem (see definitions 1.1, 1.5), we will define an inequality and equality constraint formulation – and mostly proceed working with the latter.

Definition 2.1 (Quadratic MWIS)**.** Let $[n]$ be the set of *labels*, $K_j \subseteq [n]$ with $j \in [m]$ be m *conflict sets* and $[[n]]^2 := \{ik \mid i,k \in [n], i < k\}$ be the set of (ordered) *label pairs*. For a subset $N_Z \subseteq [[n]]^2$ thereof, the *Quadratic Maximum-Weight Independent Set (Q-MWIS) problem* is defined as

$$\begin{aligned} \max_{x \in \{0,1\}^n} & \sum_{i=1}^{n} c_i x_i + \sum_{ik \in N_Z} c_{ik} x_i x_k \\ \text{s.t.} & \sum_{i \in K_j} x_i \leq 1, \ \forall j \in [m], \end{aligned} \tag{2.1}$$

with $c_i, c_{ik} \in \mathbb{R}$ being *unary* and *pairwise costs* respectively, for $i \in [n]$ and $ik \in N_Z$.

When a label pair $ik \in [[n]]^2$ is contained in some conflict set, a solution setting both to 1 can not be feasible for the problem. As such, these terms $c_{ik}x_ix_k$ are omitted from the objective function by setting $c_{ik} = 0$ and N_Z can be seen as a representation of label pairs with *non-zero* pairwise costs c_{ik}.

Analogously to the MWIS problem, we can reformulate Q-MWIS problems to include only equality constraints, without affecting its optimal solution:

Definition 2.2 (Q-MWIS with equality constraints)**.** We introduce m slack decision variables $x_{n+1}, \dots, x_{n+m}$ to the Q-MWIS problem (2.1). Their unary and pairwise costs are zero: $c_{n+1} = \cdots = c_{n+m} = 0$ and $c_{ik} = 0$, if $i > n$ or $k > n$. Additionally, let $K'_j = K_j \cup \{n + j\}, \ \forall j \in [m]$ be the new conflict sets. The *equality constraint reformulation of the Q-MWIS problem* is then given by:

$$\begin{aligned} \max_{x \in \{0,1\}^{n+m}} \sum_{i=1}^{n} c_i x_i + \sum_{ik \in N_Z} c_{ik} x_i x_k \\ \text{s.t.} \sum_{i \in K'_j} x_i = 1, \ \forall j \in [m]. \end{aligned} \tag{2.2}$$

Note that we omitted the slack decision variable indices in the sums of the objective function of (2.2), since their unary and pairwise costs are set to zero anyways.
Like in the MWIS problem equality constraint reformulation, these slack variables don't affect the original decision variables or the objective function. The value assigned to them depends on the values of the decision variables in the underlying conflict sets, as described in (1.7).

Additionally, we define the following form, in which Q-MWIS problems are usually considered in practice:

Definition 2.3 (Q-MWIS standard form)**.** A Q-MWIS problem with set of labels $[n]$ and conflict sets K_j with $j \in [m]$, is said to be in *standard form*, if it holds that

$$\forall i \in [n], \ \exists j \in [m] : \ i \in K_j, \tag{2.3}$$

i.e. all labels are contained in at least one conflict set.

Remark 2.4 (Transformation to standard form)**.** It is clear that the decision variables are unconstrained, when their label isn't contained in some conflict set.
We make the following observations for such decision variables x_i, with $i \in [n]$ and $\nexists j \in [m] : \ i \in K_j$:

- If the cost terms connected to label i, namely the unary costs c_i and pairwise cost terms c_{si}, c_{it}, $\forall s, t \in [n]$ with $s < i$ and $t > i$, are all nonnegative, with at least one term strictly greater zero, all optimal solutions will contain label i, i.e. have $x_i = 1$.

- Similarly, if all cost terms associated with i are nonpositive, with at least one term strictly smaller than zero, all optimal solutions will have the decision variable $x_i = 0$.

- If all costs connected to label i were zero, the variable choice doesn't affect the optimal solution and x_i can be chosen at random.

Such decision variables can be fixed in the Q-MWIS problem, as a part of the optimal solution already is trivially known.

However, if the costs associated with labels i are both negative and positive, the optimal choice of x_i is non-trivial. Indeed, it corresponds to the *quadratic unconstrained binary optimization* (QUBO) problem, which is known to be NP-hard and can be used to describe many combinatorial optimization problems (see e.g. [20]).

In this case, the Q-MWIS problem could be transformed to standard form, by adding conflict sets $K_{j+1}, \ldots, K_{j+N}$ with cardinality 1, which each contain one of the N labels that aren't contained in any conflict set. Effectively, this adds the trivially holding bounding constraints $x_i \leq 1$ to our Q-MWIS problem, for all such labels i. The reason for considering these seemingly pointless pseudo conflict set constraints will become evident later in this work, when we apply linearization techniques on Q-MWIS problems that are not necessarily in standard form.

Remark 2.5 (Connection to QAP and MAP-Inference)**.** It can be seen that the Q-MWIS problem is tightly connected to other known combinatorial optimization problems, such as the quadratic assignment problem (QAP) and the maximum a posteriori (MAP) inference problem for graphical models.

For the QAP, which has several similar formulations and was originally introduced in [21], this is most readily seen with the format used in [22], where it is given in form of a quadratic integer program:

$$\min_{(x_{ij})_{1\leq i,j\leq n}\in\mathbb{R}^{n\times n}} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} c_{ijkl}x_{ik}x_{jl} + \sum_{i,j=1}^{n} b_{ij}x_{ij}, \tag{2.4}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^{n} x_{ij} = 1, \ \forall j = 1, \ldots, n, \\ & \sum_{j=1}^{n} x_{ij} = 1, \ \forall i = 1, \ldots, n \\ \text{and} \quad & x_{ij} \in \{0,1\}, \ \forall i,j = 1, \ldots, n, \end{aligned} \tag{2.5}$$

with cost terms $c_{ijkl} \in \mathbb{R}$, $\forall i,j,k,l \in [n]$ and $b_{ij} \in \mathbb{R}$, $\forall i,j \in [n]$. Since the objective function (2.4) is quadratic and the assignment constraints (2.5) directly present $2n$ conflict set like constraints of a specific form, its format is very close to the Q-MWIS problem with equality constraints as defined in (2.2). The minimization problem could be turned into a maximization, by simply switching the sign of the costs.

Similarly, the MAP-inference problem very closely matches the Q-MWIS problem with equality constraints. For a graphical model $(\mathcal{G}, \mathcal{Y}_\mathcal{V}, \theta)$ with graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of nodes $\mathcal{V}$ and edges $\mathcal{E}$, space of labelings $\mathcal{Y}_\mathcal{V}$ and cost vector θ, the MAP-inference problem is given by

$$\min_{y \in \mathcal{Y}_\mathcal{V}} \sum_{u \in \mathcal{V}} \theta_u(y_u) + \sum_{uv \in \mathcal{E}} \theta_{uv}(y_u, y_v), \tag{2.6}$$

where $\theta_u(\cdot)$ and $\theta_{uv}(\cdot)$ are unary and pairwise cost functions respectively, matching costs to labels for all nodes $u \in \mathcal{V}$ and edges $uv \in \mathcal{E}$.
To transform this into a Q-MWIS problem, one may assign decision variables $x_s^{(u)}$, $\forall u \in \mathcal{V}$ and $\forall s \in \mathcal{Y}_u$, where $\mathcal{Y}_u$ represents the set of labels for node u – i.e., each decision variable $x_s^{(u)}$ corresponds to whether label s is selected in node u, or not. They can be combined with the unary and pairwise cost terms of the graphical model, yielding a quadratic objective function with the same function value as the one of (2.6).
In order to enforce that the choices for $x_s^{(u)}$ result in a labeling, i.e. exactly one label is picked for each node, equality constraints can be added, resulting in the problem formulation

$$\begin{aligned} \min_{x \in \{0,1\}^N} \quad & \sum_{u \in \mathcal{V}} \sum_{s \in \mathcal{Y}_u} \theta_u(s) x_s^{(u)} + \sum_{uv \in \mathcal{E}} \sum_{s \in \mathcal{Y}_u} \sum_{t \in \mathcal{Y}_v} \theta_{uv}(s,t) x_s^{(u)} x_t^{(v)}, \\ \text{s.t.} \quad & \sum_{s \in \mathcal{Y}_u} x_s^{(u)} = 1, \ \forall u \in \mathcal{V}, \end{aligned} \tag{2.7}$$

where $N = \sum_{v \in V} |\mathcal{Y}_v|$ is the total number of labels of all nodes. Just as with the QAP, the problem (2.7) can be turned into maximization by considering costs with flipped signs, closely matching Q-MWIS problem with equality constraints.

However, the MAP-inference problem could also be transformed into a Q-MWIS problem with inequality constraints: To this end, we would have to enforce the costs to be negative, by subtracting a large enough constant from all unary and pairwise costs. While this affects the optimal value of the objective function, the optimal solution is unchanged. All costs would become strictly positive on changing their sign and considering a maximization problem instead. Lastly, due to the non-intersecting structure of the conflict set constraints (2.7) and since all costs are positive, the equality constraints can be replaced by inequalities, without affecting the optimal solution. Solving the thus constructed Q-MWIS problem with inequality constraints would also yield an optimal solution for the original MAP-inference problem.

Following this introduction of the Q-MWIS problem, we proceed with contemplations on different Q-MWIS problem reformulations, which may prove useful in the practical part thereafter – namely ones that transform the quadratic integer programming problem into a linear one.

2.1 The "trivial" Q-MWIS linearization

To reformulate a Q-MWIS problem as an integer linear program, one has to linearize the quadratic pairwise cost terms and add appropriate constraints, such that the outcome is equivalent to the original problem.

The most straightforward way of doing so, would be to introduce auxiliary variables $y_{ik} = x_i x_k$ for all $ik \in N_Z$ and add constraints, which guarantee that

$$\forall ik \in N_Z \colon y_{ik} = \begin{cases} 0, & \text{if } x_i = 0 \text{ or } x_k = 0, \\ 1, & \text{otherwise.} \end{cases} \tag{2.8}$$

A Q-MWIS problem with equality constraints, as described in Definition 2.2, linearized in such a way, results in the following problem description:

Example 2.6 (Trivial linearization of Q-MWIS)**.** We combine the pairwise cost term factors $x_i x_k$ from the objective function of (2.2) into an auxiliary decision variable y_{ik} for all $ik \in N_Z$ and add the constraints needed, such that $y_{ik} = x_i x_k$ is guaranteed.
Note that for an unambiguous index labeling, we consider label pairs $ik \in [[n]]^2$ to always satisfy $i < k$, by definition of $[[n]]^2$.
This yields the following ILP:

$$\begin{aligned} \max_{\substack{x \in \{0,1\}^{n+m} \\ y \in \{0,1\}^{|N_Z|}}} \sum_{i=1}^{n} c_i x_i + \sum_{ik \in N_Z} c_{ik} y_{ik} \\ \text{s.t.} \sum_{i \in K'_j} x_i = 1,\ \forall j \in [m], \\ y_{ik} \leq x_i,\ \forall ik \in N_Z, \\ y_{ik} \leq x_k,\ \forall ik \in N_Z, \\ y_{ik} \geq x_i + x_k - 1,\ \forall ik \in N_Z. \end{aligned} \tag{2.9}$$

In total, it contains $n + m + |N_Z|$ variables and $m + 3|N_Z|$ constraints, not including the integrality and bounding constraints on the variables.

Obviously, an equivalent linearization can be applied to the original Q-MWIS problem with inequality constraints, but we proceed focusing on this

one for now.
We note that the size of this linearization mostly depends on the cardinality of $N_Z \subseteq [[n]]^2$, which can contain up to $\frac{n(n-1)}{2}$ label pairs. In the worst case, this nearly squares the number of constraints and greatly increases the number of variables – with the upside of removing the quadratic objective function terms, thus allowing ILP methods and algorithms to be applied.

To confirm the equivalence of this linearization (2.9) to the original problem, we note the following:

Proposition 2.7. *For any $x \in \{0,1\}^{n+m}$ feasible to the QIP problem described in* (2.2)*, there exists a tuple (x, y), with $y \in \{0,1\}^{N_Z}$, feasible to the ILP problem in Example 2.6 and vice versa. The solutions yield the same value in the objective functions of the respective problems.*

Proof. Firstly, we can see that any tuple (x, y), which is a feasible solution to the ILP problem (2.9), contains the solution x, feasible to the original QIP problem, since the conflict set constraints $\sum_{i \in K'_j} x_i = 1$ are fulfilled, $\forall j \in [m]$.
As we have noted for any y_{ik} satisfying the inequality constraints of (2.9), its value is 1, if $x_i = x_k = 1$ and 0, if x_i or x_k are 0, taking the same value as the quadratic term $x_i x_k$ when the decision variables are binary. It follows that for any x, which is a feasible solution to the QIP problem (2.2), there exists exactly one $y \in \{0,1\}^{N_Z}$, such that the tuple (x, y) satisfies the inequality constraints of the ILP problem and is therefore a feasible solution to it. Since these y_{ik} take the same value as the term $x_i x_k$, $\forall ik \in N_Z$, when the decision variables are binary, the objective function value of x for the QIP problem is identical to the one of (x, y) for the ILP problem. □

While this linearization is very straightforward, it's not necessarily the most efficient way of formulating the problem – both, in terms of number of constraints and tightness of the corresponding LP relaxation. It is not clear, whether removing the quadratic terms really is worth the vastly increased complexity (in terms of number of constraints and variables) for problems with large sets N_Z.

We have repeatedly noted, how the more closely the feasible set of the LP relaxation matches the convex hull of feasible solutions to the ILP problem, the better are the approximate solutions one may acquire from it. Therefore, we will consider another approach in the following subsection, suggested by Sherali and Adams in [2], which additionally aims to tighten the LP relaxation of the resulting integer linear programming problem during the reformulation process.

2.2 The Sherali-Adams linearization for Q-MWIS problems

As mentioned in Remark 1.8: When the goal is to construct tighter LP relaxations, one may consider the *Sherali-Adams relaxation hierarchy* described in [2]. It allows for a tightening of the basic LP relaxation to various degrees, by using a *reformulation-linearization technique* (RLT).

The general procedure for ILP problems is as follows:
Consider any integer linear program with feasible set X and n variables $x \in \{0,1\}^n$, where X is defined by a number of inequality, equality and integrality constraints:

$$\max_{x \in X} \langle c, x \rangle, \text{ with feasible set} \tag{2.10}$$

$$X = \left\{ (x_1, \ldots, x_n) \in \{0,1\}^n \middle| \begin{aligned} &\sum_{j=1}^{n} \alpha_{rj} x_j \geq \beta_r, r = 1, \ldots, R_1; \\ &\sum_{j=1}^{n} a_{rj} x_j = b_r, r = 1, \ldots, R_2 \end{aligned} \right\}. \tag{2.11}$$

Let $d \in [n]$ and denote certain polynomials of order d by $F_d(J_1, J_2)$, where $J_1, J_2 \subseteq [n]$ are non-intersecting subsets covering d elements of $[n]$ when joined, i.e. $J_1 \cap J_2 = \emptyset$ and $|J_1 \cup J_2| = d$. They have the following form:

$$F_d(J_1, J_2) = \left(\prod_{j \in J_1} x_j \right) \left(\prod_{j \in J_2} (1 - x_j) \right). \tag{2.12}$$

Any pair of subsets (J_1, J_2) satisfying the above conditions is said to be *of order d* and N_d is the total number of such subset pairs.
Using these definitions, we can construct relaxations X_{Pd} for any $d \in [n]$, by applying the following steps:

1. Multiply all R_1 inequality constraints and R_2 equality constraints of X with each polynomial $F_d(J_1, J_2)$ individually. This results in a number of $(R_1 + R_2) \cdot N_d$ mostly non-linear (in-)equalities.

2. For $D := \min\{d+1; n\}$, add N_D constraints $F_D(J_1, J_2) \geq 0$ for all subset pairs (J_1, J_2) of order D. Note: This is akin to multiplying the bounding

constraints $0 \leq x_i \leq 1,\ \forall i \in [n]$ with all polynomials $F_d(J_1, J_2)$ for all subsets (J_1, J_2) of order d.

3. Lastly, as a means to linearize the previously constructed (in-)equalities in step 1 and 2, we first apply the relationship $x_i^2 = x_i$ or equivalently $x_i(1 - x_i) = 0\ \forall i = 1, \ldots, n$, which holds trivially for all ILPs with binary decision variables. The remaining non-linear terms of the form $\prod_{j \in J} x_j$, for some subset $J \subseteq [n]$ with $|J| \geq 2$, are substituted by variables w_J. Polynomials $F_d(J_1, J_2)$ reformulated in such a way are denoted as $f_d(J_1, J_2)$.

The set of variables $(x, w) \in \mathbb{R}^n \times \mathbb{R}^{m_d}$ fulfilling these constraints, where m_d depends on d and n, is defined as X_d and given by:

$$
\begin{aligned}
X_d = \Bigg\{ (x, w) \Bigg| & \text{for all } N_d \text{ subset pairs } (J_1, J_2) \text{ of order } d \text{, it holds that} \\
& \left(\sum_{j \in J_1} \alpha_{rj} - \beta_r \right) f_d(J_1, J_2) + \sum_{j \in [n] \setminus (J_1 \cup J_2)} \alpha_{rj} f_{d+1}(J_1 \cup \{j\}, J_2) \geq 0, \\
& \forall r = 1, \ldots, R_1, \\
& \left(\sum_{j \in J_1} a_{rj} - b_r \right) f_d(J_1, J_2) + \sum_{j \in [n] \setminus (J_1 \cup J_2)} a_{rj} f_{d+1}(J_1 \cup \{j\}, J_2) = 0, \\
& \forall r = 1, \ldots, R_2 \\
& \text{and for all } N_D \text{ subsets } (J_1, J_2) \text{ of order } D = \min\{d+1, n\}, \\
& \text{it holds that: } f_D(J_1, J_2) \geq 0 \Bigg\}.
\end{aligned}
\tag{2.13}
$$

Then, the relaxation for our ILP problem described in (2.10) and (2.11) is acquired by projecting the (x, w) tuples of the set X_d to x, i.e. they are given by

$$X_{P^d} = \{x \mid (x, w) \in X_d\}. \tag{2.14}$$

In [2, Thm. 1, 3], it is proven that the relaxations constructed in such a way form a hierarchy from the basic LP relaxation to the convex hull of X:

$$X_0 \equiv X_{P^0} \supseteq X_{P^1} \supseteq \cdots \supseteq X_{P^n} \equiv \operatorname{conv}(X), \tag{2.15}$$

where X_0 denotes the feasible set of the basic LP-relaxation and $\operatorname{conv}(X)$ is the convex hull of X. The equivalence of X_0 and X_{P^0} follows, when setting

$(J_1, J_2) = (\emptyset, \emptyset)$ as the only subset pair of order 0 and $f_0(\emptyset, \emptyset) = 1$.

We can see that, since X_{P^n} is equal to the convex hull of X, solving the corresponding relaxed problem would also solve the original ILP problem (see (0.5)). However, it is evident that solving larger ILP problems this way is infeasible, as the number of constraints grows linearly in the number of subset pairs (J_1, J_2) of order d and D, which is equal to $\binom{n}{d}2^d$ factors. Nevertheless, Sherali-Adams relaxations of lower order might be of use as a tighter alternative to the LP relaxation of ILP problems.

Remark 2.8. It should be stressed that the constraints acquired through the previously described polynomial multiplications and nonnegativity inequalities don't remove (integer) solutions from the original problem or add any to it.
As proven in [2, Cor. 1], a tuple (x, w) with binary variables $x \in \{0, 1\}^n$ is an element of X_d, $d \in [n]$, if and only if $x \in X$ and the linearized terms all correspond to their quadratic counterparts: $w_J = \prod_{j \in J} x_j$, for all sets J with $|J| = 2, \ldots, D$ for $D = \min\{d + 1, n\}$. Indeed, if the constraints were to remove any feasible solution to X, this would conflict with the hierarchy described in (2.15).

Intuitively, this becomes evident when realizing that the multiplication with all such factors $F_d(J_1, J_2)$ essentially amounts to a multiplication of the constraints with a nonnegative factor, which can only take the values 0 or 1 in the case of binary variables. Each polynomial therefore "fixes" constraints to only consider, what happens when the d decision variables with indices in $J_1 \cup J_2$ are $x_i = 1$ for $i \in J_1$ and $x_j = 0$ for $j \in J_2$. Hence, terms x_i and $(x_j - 1)$ are also termed *bound-factors* in [23]. This also explains the requirement of sets J_1 and J_2 to be non-intersecting, since otherwise the process would essentially just amount to a multiplication with 0, which serves no purpose.

Instead, raising the polynomial order of constraints allows this procedure to apply the relationship $x_i^2 = x_i, \forall i \in [n]$ to the constraints, which obviously holds for binary values x_i, but is violated by any non-binary variable. It is this step, which yields the tightening of the relaxation.

We also note the following:

Remark 2.9 (Interpretation as lift-and-project method)**.** The lift-and-project cutting plane method suggested in [8] is similar to the RLT by Sherali and Adams, in that it uses similar bound-factors x_i, $(1 - x_i)$ in the lifting process.

However, they are only applied for one variable at a time.
More generally, we can see that the RLT by Sherali-Adams works by lifting the dimensionality of the original problem, through replacement of higher order terms with linear variables. The resulting problem can then be solved and projected back to the space of original decision variables.

Example 2.10. As an illustration of the use of the Sherali-Adams transformations on a minimal example, consider the feasible set of an ILP

$$X = \left\{(x_1, x_2) \in \{0,1\}^2 \mid 2x_1 + 2x_2 \geq 1\right\}. \tag{2.16}$$

Since X contains exactly the three points $(0,1), (1,0), (1,1)$, its convex hull is given by

$$\text{conv}(X) = \left\{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 + x_2 \geq 1,\ x_1 \leq 1,\ x_2 \leq 1\right\}, \tag{2.17}$$

the convex combination of those points.

Firstly, the LP-relaxation of X is equivalent to X_{P^0} and acquired by relaxing the integrality constraints of X:

$$X_{P^0} \equiv X_0 = \left\{(x_1, x_2) \in [0,1]^2 \mid 2x_1 + 2x_2 \geq 1\right\}. \tag{2.18}$$

To construct the set X_1 according to the Sherali-Adams scheme, the single inequality constraint of X has to be multiplied by all polynomials $F_d(J_1, J_2)$ of order $d = 1$, namely $x_1, x_2, (1 - x_1)$ and $(1 - x_2)$. Exemplary for the first polynomial x_1 we get:

$$2x_1^2 + 2x_2x_1 \geq x_1. \tag{2.19}$$

Using the relationship $x_1^2 = x_1$ and substituting the leftover non-linear term $x_1x_2 = w_{12}$ yields:

$$x_1 + 2w_{12} \geq 0. \tag{2.20}$$

A similar process with the other polynomials of order 1 results in the following constraints:

$$\begin{aligned} x_2 + 2w_{12} &\geq 0, \\ x_1 + 2x_2 - 2w_{12} &\geq 1, \\ 2x_1 + x_2 - 2w_{12} &\geq 1. \end{aligned} \tag{2.21}$$

Additionally, the nonnegativity constraints for polynomials $F_2(J_1, J_2)$ have to be included, $\forall(J_1, J_2)$ of order 2:

$$\begin{aligned} w_{12} &\geq 0, \\ x_1 - w_{12} &\geq 0, \\ x_2 - w_{12} &\geq 0, \\ 1 - x_1 - x_2 + w_{12} &\geq 0. \end{aligned} \tag{2.22}$$

Then X_1 is given by the set of tuples $(x_1, x_2, w_{12}) \in \mathbb{R}^3$ fulfilling above constraints (2.20)-(2.22). X_2 can be constructed in a similar fashion, applying the same steps for $d = 2$. Its projected set would be equivalent to the convex hull of X.

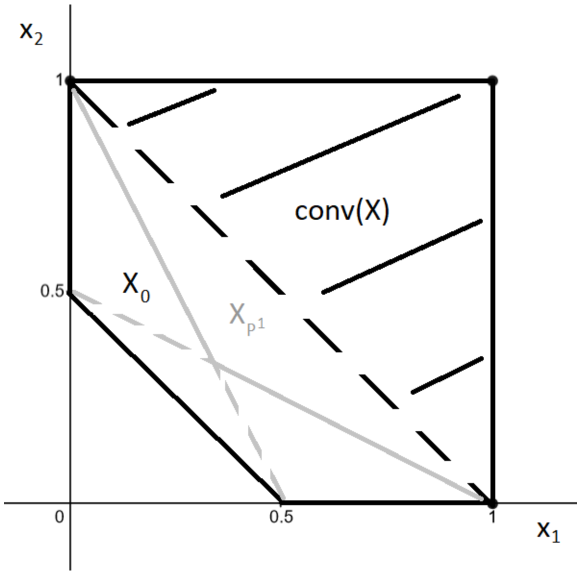

Figure 1: Illustration of the example sets. X is given by the points $(0, 1)$, $(1, 0)$ and $(1, 1)$, the convex combination of which yields the convex hull conv(X). It is a subset of the LP relaxation X_0 and the projected set X_{P^1}. In total, it is evident that conv$(X) \subseteq X_{P^1} \subseteq X_0$.

Note: All constraints of X_0, i.e. the single set constraint and the bounding constraints on x_1 and x_2, can be acquired by combining constraints of X_1, which implies $X_{P^1} \subseteq X_0$.
However, we can see that $X_0 \not\subseteq X_{P^1}$, since the constructed constraints eliminate possible solutions from X_0: For example, $x_1 = 0$, $x_2 = 0.5$ is a feasible solution for X_0, but not for X_{P^1}, since there is no tuple $(0, 0.5, w_{12})$ fulfilling the constraints of X_1. This can be seen due to the fact that the second constraint in (2.21) and the first nonnegativity constraint in (2.22) imply $w_{12} = 0$, which

violates the third constraint of (2.21), as $(x_1, x_2, w_{12}) = (0, 0.5, 0)$ yields the contradiction $0.5 \geq 1$.
Additionally, it is evident that X_{P^1} is not equivalent to the convex hull of X: While the tuple $(x_1, x_2, w_{12}) = (\frac{1}{3}, \frac{1}{3}, 0)$ satisfies the constraints (2.20), (2.21) and (2.22), meaning that $(\frac{1}{3}, \frac{1}{3}) \in X_{P^1}$, it is not an element of the convex hull of X, since it violates the constraint $x_1 + x_2 \geq 1$.
By considering the constraints for $w_{12} = 0$ and how they move when increasing it, we can see that the projected set X_{P^1} is given by:

$$X_{P^1} = \left\{(x_1, x_2) \in \mathbb{R}^2 \mid 2x_1 + x_2 \geq 1,\ x_1 + 2x_2 \geq 1,\ x_1 \leq 1,\ x_2 \leq 1\right\}. \tag{2.23}$$

The intersection of its first two constraints reveals an extra (non-integer) vertex not present in the convex hull of X, namely $(\frac{1}{3}, \frac{1}{3})$. This implies that the convex hull of X is strictly contained in X_{P^1}, but not equivalent to it.

After this description of how the Sherali-Adams reformulation-linearization technique works, we will apply this method to attempt to build a more efficient linearization of the Q-MWIS problem – both in terms of number of constraints and tightness of the corresponding LP relaxation.

Constructing a linearization of the Q-MWIS problem with the Sherali-Adams scheme

The idea behind incorporating the Sherali-Adams method in the construction of an efficient linearization of the Q-MWIS problem, comes from the fact that both include quadratic terms, which will subsequently be linearized. For the Q-MWIS problem (2.2) these terms are in the objective function, while the RLT by Sherali-Adams includes them in the constraints.
This allows us to linearize the quadratic terms in the objective function and constraints simultaneously, while tightening the latter – resulting in a linearization for our QIP problem, whose LP relaxation potentially performs better than the one of the trivial linearization described in Example 2.6. As previously mentioned, the number of added constraints for the sets X_d increases with the number of subset pairs of order d, given by $\binom{n}{d}2^d$, exhibiting an exponential growth in d. In order to retain the feasibility of our approach for problems of larger scale, we will therefore focus on the Sherali-Adams relaxation of degree $d = 1$.

We will construct the Sherali-Adams linearization for the general Q-MWIS problem with equality constraints, introduced in Definition 2.2, which is given by

$$\begin{aligned} \max_{x\in\{0,1\}^{n+m}} \sum_{i=1}^{n} c_i x_i + \sum_{ik\in N_Z} c_{ik} x_i x_k \\ \text{s.t.} \sum_{i\in K'_j} x_i = 1, \ \forall j \in [m], \end{aligned} \tag{2.24}$$

for a conflict set representation $\{K'_j\}_{j\in[m]}$, unary cost terms $c_i \in \mathbb{R}, \ \forall i \in [n]$ and pairwise costs $c_{ik} \in \mathbb{R}, \ \forall ik \in N_Z \subseteq [[n]]^2$. Most of the following steps can be equivalently applied to the original Q-MWIS problem with inequality constraints, though.

In order to construct the set X_1, as defined in (2.13), we start out by multiplying all conflict set constraints with each polynomial $F_1(J_1, J_2)$, where $J_1, J_2 \subseteq [n]$ are two disjoint subsets of order 1 – i.e. all polynomials $x_1, \ldots, x_{n+m}$ and $(1 - x_1), \ldots, (1 - x_{n+m})$.
This yields the following $m \cdot 2(n + m)$ constraints:

$$\begin{aligned} \left(\sum_{i\in K'_j} x_i x_1\right) - x_1 = 0, \forall j \in [m], \\ \vdots \\ \left(\sum_{i\in K'_j} x_i x_{n+m}\right) - x_{n+m} = 0, \forall j \in [m], \\ \left(\sum_{i\in K'_j} x_i - x_i x_1\right) + x_1 - 1 = 0, \forall j \in [m], \\ \vdots \\ \left(\sum_{i\in K'_j} x_i - x_i x_{n+m}\right) + x_{n+m} - 1 = 0, \forall j \in [m]. \end{aligned} \tag{2.25}$$

Using the relationships $x_k^2 = x_k, \forall k \in [n+m]$ on all equations it is applicable, namely the ones with $k \in K'_j$, we get:

$$\begin{aligned}
&\forall j \in [m], k \in [n+m] : \begin{cases} \left(\sum_{i \in K'_j} x_i x_k \right) - x_k = 0, & \text{if } k \notin K'_j, \\ \sum_{i \in K'_j \setminus \{k\}} x_i x_k = 0, & \text{if } k \in K'_j \end{cases} \\
&\text{and} \begin{cases} \left(\sum_{i \in K'_j} x_i - x_i x_k \right) + x_k - 1 = 0, & \text{if } k \notin K'_j, \\ \left(\sum_{i \in K'_j \setminus \{k\}} x_i - x_i x_k \right) + x_k - 1 = 0, & \text{if } k \in K'_j. \end{cases}
\end{aligned} \tag{2.26}$$

The remaining nonlinear terms are linearized, by substituting $w_{ik} = x_i x_k$ with all the quadratic terms left in the equations.
In addition to the equality constraints acquired through multiplication of the original conflict set constraints with polynomials, we require nonnegativity constraints on the polynomials $F_2(J_1, J_2)$ as described in (2.12), for all possible subsets $J_1, J_2 \subseteq [n]$ of order 2.
These are given by:

$$\begin{gathered}
\forall i, k \in [n+m], \text{ with } i \neq k : \\
x_i x_k \geq 0, \\
x_i(1 - x_k) \geq 0, \\
(1 - x_i)(1 - x_k) \geq 0,
\end{gathered} \tag{2.27}$$

resulting in a total of $2(n+m-1)(n+m)$ inequality constraints – a quarter defined in the first and third constraint set with $\frac{1}{2}(n+m-1)(n+m)$ constraints each and half of them in the middle expression with $(n+m-1)(n+m)$ constraints. Just as for the equations acquired through polynomial multiplication in (2.26), the quadratic terms are then replaced by $w_{ik} = x_i x_k$. They are indexed by the pair set $[[n+m]]^2$, resulting in a total of $\frac{(n+m-1)(n+m)}{2}$ terms.

Before passing on to combine the thus acquired constraints in the set X_1, we make the following observations:

Remark 2.11. Firstly, it should be noted that the constraints acquired by polynomial multiplication still imply the original constraints – this holds for both the bounding constraints on our decision variables $0 \leq x_i \leq 1$, $\forall i \in [n+m]$ and the conflict set constraints of the problem formulation (2.24), $\sum_{i \in K'_j} x_i = 1,\ \forall j \in [m]$.

The former are implied by the nonnegativity constraints (2.27): We can see that the first constraint of (2.27) implies that all decision variables are either nonnegative, or nonpositive. The option that they are nonpositive can be discarded by considering the second constraint of (2.27), hence $x_i \geq 0,\ \forall i \in [n+m]$. Using this fact with the third constraint of (2.27), we can deduct that $0 \leq x_i \leq 1,\ \forall i \in [n+m]$.

Similarly, we can see that the $m \cdot 2(n+m)$ reformulated constraints still imply the original conflict set constraints of (2.24). Indeed, as the original constraints are equalities that were multiplied by terms x_k and $(1 - x_k)$, for $k \in [n+m]$, the resulting equations can be transformed to their original version by simply adding the equations multiplied with x_k to the ones multiplied with $(1 - x_k)$.

This last observation highlights the possibility for a more concise set of constraints, than the one described in (2.26). As the equations we acquired from multiplying the conflict set constraints with $(1 - x_k),\ \forall k \in [n+m]$, are implied by the original constraints and the equations we got by multiplication with terms $x_k,\ \forall k \in [n+m]$, we can replace the former with the conflict set constraints. The resulting set of constraints is equivalent to the previous one in (2.26) – even when considering relaxations later on – but only consists of $m(n+m)+m$ equations, yielding a more efficient formulation:

$$\forall j \in [m], k \in [n+m] : \begin{cases} \left(\sum_{i \in K'_j} x_i x_k \right) - x_k = 0, & \text{if } k \notin K'_j, \\ \sum_{i \in K'_j \setminus \{k\}} x_i x_k = 0, & \text{if } k \in K'_j \end{cases} \tag{2.28}$$
$$\text{and } \forall j \in [m] : \sum_{i \in K'_j} x_i = 1.$$

We note that this reduction of constraints only works for the Q-MWIS problem with equality constraints. A linearization of the original problem instance with inequality constraints must therefore contain both constraint sets – the ones we acquire by multiplication with x_k and $(1 - x_k),\ \forall k \in [n]$.

After thus constructing the new equality and nonnegativity constraints, the constraints from (2.28) and (2.27) are linearized by setting $w_{ik} = x_i x_k$, $\forall ik \in [[n+m]]^2$ (i.e. $i < k$) and combined to define the polyhedral set X_1, which contains tuples $(x, w) \in \mathbb{R}^{n+m} \times \mathbb{R}^{\frac{1}{2}(n+m-1)(n+m)}$:

$$X_1 = \Bigg\{ (x,w) \Bigg| \forall j \in [m], k \in [n+m] \text{ and } k \notin K'_j :$$

$$\left(\sum_{\substack{s \in K'_j \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \\ t>k}} w_{kt} \right) - x_k = 0; \tag{2.29}$$

$$\forall j \in [m], k \in [n+m] \text{ and } k \in K'_j :$$

$$\sum_{\substack{s \in K'_j \setminus \{k\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \setminus \{k\} \\ t>k}} w_{kt} = 0; \tag{2.30}$$

$$\forall j \in [m] : \sum_{i \in K'_j} x_i = 1; \tag{2.31}$$

$$\text{Additionally, } \forall ik \in [[n+m]]^2 :$$

$$w_{ik} \geq 0, \tag{2.32}$$

$$\begin{aligned} x_i - w_{ik} &\geq 0, \\ x_k - w_{ik} &\geq 0, \end{aligned} \tag{2.33}$$

$$w_{ik} - x_i - x_k + 1 \geq 0 \Bigg\}. \tag{2.34}$$

Remark 2.12 (Connection to trivial linearization)**.** It is apparent at this point that the set X_1 contains all constraints of the trivial linearization described in Example 2.6 – namely the nonnegativity and conflict set constraints. As such, the Sherali-Adams linearization is bound to provide a tighter LP relaxation, than the trivial linearization we considered prior, when projected on the original space of decision variables $\mathbb{R}^{n+m}$ – at the cost of a higher number of variables and constraints.
However, we will see that some redundant constraints can be omitted, under a weak assumption.

Due to the structure of the conflict set constraints, we can simplify X_1 further. To do so, we only have to assume that the problem is in standard form (see Definition 2.3) and add nonnegativity constraints for the decision variables x_i, $\forall i \in [n+m]$.

Lemma 2.13. *Let the original Q-MWIS problem* (2.24) *that we consider here be in standard form. Then, the* $(n+m-1)(n+m)$ *constraints of* X_1 *described in* (2.33)*, given* $\forall ik \in [[n+m]]^2$ *by*

$$\begin{aligned} x_i - w_{ik} &\geq 0 \text{ and} \\ x_k - w_{ik} &\geq 0, \end{aligned} \tag{2.35}$$

are implied by nonnegativity constraints $x_i \geq 0$*,* $\forall i \in [n+m]$ *and the other constraints of* X_1*, excluding* (2.34)*.*

Proof. Firstly, it should be noted that the nonnegativity constraints on the decision variables are implied by constraints (2.32) and (2.33). Thus, they can be added to X_1, without affecting the properties of the linearization later on.

Let $ik \in [[n+m]]^2$ be any label pair (with $i < k$, as per definition). Since the problem is in standard form, it holds for said pair that

$$\forall i \in [n+m],\ \exists j \in [m]:\ i \in K'_j \text{ and either } k \in K'_j \text{ or } k \notin K'_j. \tag{2.36}$$

In the case that $k \in K'_j$, we have from the second constraint (2.30) that

$$\left(\sum_{\substack{s \in K'_j \setminus \{k,i\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \setminus \{k,i\} \\ t>k}} w_{kt} \right) + w_{ik} = 0. \tag{2.37}$$

Coupled with the first nonnegativity constraint (2.32), given by $w_{st} \geq 0$, $\forall st \in [[n+m]]^2$, this means that $w_{ik} = 0$ and the constraints (2.35) reduce to nonnegativity constraints $x_i, x_k \geq 0$, which we assumed to hold.

If $k \notin K'_j$, the first constraint (2.29) reads

$$\left(\sum_{\substack{s \in K'_j \setminus \{i\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \setminus \{i\} \\ t>k}} w_{kt} \right) + w_{ik} - x_k = 0. \tag{2.38}$$

Again, as $w_{st} \geq 0$, $\forall st \in [[n+m]]^2$, this implies the sums in the parentheses are greater equal zero and thus $w_{ik} - x_k$ can't be strictly greater 0. Therefore, $w_{ik} - x_k \leq 0$, i.e. $x_k - w_{ik} \geq 0$ holds.

A similar reasoning deducts $x_i - w_{ik} \geq 0$, if the roles of i and k are reversed in the steps above.

□

Under the same assumptions, we can omit another set of constraints, which can be acquired by combining the others:

Lemma 2.14. *Let the original Q-MWIS problem* (2.24) *that we consider here be in standard form. Then, the* $\frac{1}{2}(n+m-1)(n+m)$ *constraints* (2.34), *given* $\forall ik \in [[n+m]]^2$ *by*

$$w_{ik} - x_i - x_k + 1 \geq 0, \tag{2.39}$$

are already implied by the other constraints of X_1 *and nonnegativity constraints* $x_i \geq 0$, $\forall i \in [n+m]$.

Proof. Firstly, we can see that $x_i \in [0,1]$ holds $\forall i \in [n+m]$: $x_i \geq 0$, $\forall i \in [n+m]$ was added through constraints and due to the original problem being in standard form, every decision variable is contained in at least one conflict set. Thus, $\forall i \in [n+m]$, $\exists j \in [m]$, such that $i \in K'_j$ and

$$\sum_{s \in K'_j \setminus \{i\}} x_s + x_i = 1. \tag{2.40}$$

Coupled with the nonnegativity of decision variables, this implies that $x_i \leq 1$ holds $\forall i \in [n+m]$.

Now, let $ik \in [[n+m]]^2$ be any label pair. If there exists any conflict set containing both labels, we have $x_i + x_k \leq 1$ from the respective conflict set constraint (2.31). When combining this with the constraints $w_{st} \geq 0$, $\forall st \in [[n+m]]^2$ from (2.32), inequality (2.39) directly follows.

In the case that there exists no conflict set containing both labels i, k, we know from the problem being in standard form that there exists a conflict set containing i, i.e. $\exists j \in [m]$, such that $i \in K'_j$ and $k \notin K'_j$.
Thus, the first type of constraints (2.29) in X_1 yields

$$\left(\sum_{\substack{s \in K'_j \setminus \{i\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \setminus \{i\} \\ t>k}} w_{kt} \right) + w_{ik} - x_k = 0. \tag{2.41}$$

Additionally, as has been proven in Lemma 2.13, without using (2.39), constraints (2.35) are implied by the other constraints used here, i.e.

$$\begin{aligned} x_s &\geq w_{sk},\ \forall s \in K'_j \setminus \{i\},\ s<k \text{ and} \\ x_t &\geq w_{kt},\ \forall t \in K'_j \setminus \{i\},\ t>k, \end{aligned} \tag{2.42}$$

hold.
Lastly, we consider the conflict set constraints (2.31), which yield

$$\sum_{s \in K'_j \setminus \{i\}} x_s = 1 - x_i. \tag{2.43}$$

Plugging the inequalities (2.42) into the equation (2.43) results in the inequality

$$\sum_{\substack{s \in K'_j \setminus \{i\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \setminus \{i\} \\ t>k}} w_{kt} \leq 1 - x_i. \tag{2.44}$$

Combining (2.44) with (2.41), we get

$$0 = \left(\sum_{\substack{s \in K'_j \setminus \{i\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \setminus \{i\} \\ t>k}} w_{kt} \right) + w_{ik} - x_k \leq 1 - x_i + w_{ik} - x_k, \tag{2.45}$$

i.e. $w_{ik} - x_i - x_k + 1 \geq 0$.

□

As mentioned in Remark 2.4, any Q-MWIS problem can be reduced to one in standard form, so it's an easily satisfiable assumption. The observation that we can arrive at a more concise set of constraints, if the problem is assumed to be in standard form, is bundled in the following:

Theorem 2.15 (Concise constraint set for Q-MWIS linearization with equality constraints in standard form)**.** *Let the original Q-MWIS problem* (2.2) *be in standard form and consider the set* X_1*, as defined in constraints* (2.29)-

(2.34). *Then, the polyhedral set*

$$X_1' = \Bigg\{ (x,w) \Bigg| \forall j \in [m], k \in [n+m] \text{ and } k \notin K_j' : \\ \left(\sum_{\substack{s \in K_j' \\ s<k}} w_{sk} + \sum_{\substack{t \in K_j' \\ t>k}} w_{kt} \right) - x_k = 0; \\ \forall j \in [m], k \in [n+m] \text{ and } k \in K_j' : \\ \sum_{\substack{s \in K_j' \setminus \{k\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K_j' \setminus \{k\} \\ t>k}} w_{kt} = 0; \\ \forall j \in [m] : \sum_{i \in K_j'} x_i = 1; \\ \forall ik \in [[n+m]]^2 : \; w_{ik} \geq 0, \\ \forall i \in [n+m] : x_i \geq 0 \Bigg\}. \tag{2.46}$$

is equivalent to X_1.

Proof. The fact that constraints (2.33) and (2.34) are implied by the constraints (2.29)-(2.32), coupled with nonnegativity constraints on variables $x_i \geq 0$, $\forall i \in [n+m]$, has been proven in Lemmas 2.13 and 2.14. Therefore, X_1' is a subset of X_1. The opposite also holds, since the constraints (2.32) and (2.33) of X_1 imply the nonnegativity constraints $x_i \geq 0$, $\forall i \in [n+m]$. Hence, X_1 is equivalent to X_1', if the Q-MWIS problem is in standard form. □

In total, the set X_1 contains $3m^2 + 2n^2 + 5mn - m - 2n$ constraints and $\frac{n^2+n+m^2+m}{2} + mn$ variables, while X_1' consists of $\frac{3}{2}m^2 + \frac{1}{2}n^2 + 2mn + \frac{3}{2}m + \frac{1}{2}n$ constraints and the same amount of variables. Since the sets are equivalent, the formulation of X_1' is clearly preferable.

Comparing this to the trivial linearization described in Example 2.6 that consisted of $m + 3|N_Z| + 2(n+m)$ constraints (including the bounding constraints on decision variables $0 \leq x_i \leq 1, \; \forall i \in [n+m]$) with $n + m + |N_Z|$ variables, we can see that X_1' contains fewer constraints if $N_Z = [[n+m]]^2$, with the same amount of variables. However, if the number of non-zero pairwise costs is very low, the trivial linearization contains fewer constraints in fewer variables. Which constraint set is preferable might therefore depend on the specific problem instance.

We can use the more concise constraint representation of X_1' in place of X_1, in order to define a linearization of the original Q-MWIS problem, when said problem is in standard form. It is given by

$$\begin{aligned} \max_{\substack{x\in\{0,1\}^{n+m} \\ w\in\{0,1\}^{\frac{1}{2}(n+m-1)(n+m)}}} \quad & \sum_{i=1}^{n} c_i x_i + \sum_{ik\in N_Z} c_{ik} w_{ik} \\ \text{s.t. } & (x,w) \in X_1', \end{aligned} \tag{2.47}$$

with $x = (x_i)_{i\in[n+m]} := (x_1, \dots, x_{n+m})$ and $w = (w_{ik})_{ik\in[[n+m]]^2}$.

To ensure that the resulting ILP formulation (2.47) is indeed a valid linearization of the original QIP formulation of the Q-MWIS problem in (2.2), we have to prove that solving the ILP problem also solves the original QIP problem and the other way around.

Proposition 2.16. *If the original Q-MWIS problem* (2.2) *is in standard form, any feasible solution* $x \in \mathbb{R}^{n+m}$ *can be matched to some tuple* (x, w) *feasible to the ILP problem* (2.47) *and vice versa, with both yielding the same objective value in their respective objective function.*

Proof. Let X_Q be the feasible set for the QIP formulation of the Q-MWIS problem defined in (2.2) and $X_1'^f$ the feasible set to the corresponding ILP problem (2.47), given by $X_1'^f := X_1' \cap \{0,1\}^{n+m+\frac{1}{2}(n+m-1)(n+m)}$.

So as to prove the equivalence of the problems, we first show that $\forall x \in X_Q$, $\exists w \in \{0,1\}^{\frac{1}{2}(n+m-1)(n+m)}$, s.t. $(x,w) \in X_1'^f$. Since the linearization is based on replacing the terms $x_i x_k$, it can be seen that setting $w = (w_{ik})_{ik\in[[n+m]]^2}$, with $w_{ik} = x_i x_k$, $\forall ik \in [[n+m]]^2$, works for any $x \in X_Q$: In this case, the equality constraints of X_1' defined in (2.46) revert back to their original form, namely the conflict set constraints $\sum_{i\in K_j'} x_i = 1$, $\forall j \in [m]$, multiplied by x_k, $\forall k \in [n+m]$ and the conflict set constraints themself. Because $x \in X_Q$ implies that the latter hold – and therefore the constraints reformulated with polynomial multiplication, too – such tuples (x,w) fulfill the equality constraints of X_1'. The nonnegativity constraints of X_1' are also satisfied, which follows directly from the fact that we picked $w_{ik} = x_i x_k$, $\forall ik \in [[n+m]]^2$ and $x_i \in \{0,1\}$, $\forall i \in [n+m]$. In summary, all constraints of X_1' hold and $(x,w) \in \{0,1\}^{n+m+\frac{1}{2}(n+m-1)(n+m)}$, therefore $(x,w) \in X_1'^f$.

The other direction, stating that for all tuples $(x,w) \in X_1'^f$, x is also an element of X_Q, follows immediately from the fact that such $x \in \{0,1\}^{n+m}$

must also satisfy the conflict set constraints contained in X_1'.

Lastly, in order to see that the objective function values of the problems are equal for the respective feasible solutions, we need to show that $\forall x \in X_Q$, $\exists! w$, such that $(x,w) \in X_1'^f$, namely $w = (w_{ik})_{ik \in [[n+m]]^2}$, with $w_{ik} = x_i x_k$, $\forall ik \in [[n+m]]^2$, since in that case the objective functions are equivalent.
It can be noted that the inequality constraints, which we used in the trivial linearization of the Q-MWIS problem defined in (2.9), are also present in the set X_1 in form of the nonnegativity constraints. As X_1' is equivalent to X_1, these constraints are also satisfied for X_1'. Thus, it is easy to see that tuples (x,w) feasible to $X_1'^f$ have to satisfy $w_{ik} = x_i x_k$, $\forall ik \in [[n+m]]^2$, following the same reasoning as in Example 2.6. Therefore, the objective function values of the original QIP and reformulated ILP problems are equal for the respective feasible solutions x and (x,w).

□

After establishing the equivalence of the ILP reformulation (2.47) to the initial problem, we follow up with a comparison of its LP relaxation to the LP relaxation of the initially considered linearization:

Proposition 2.17. *The LP relaxation of the linearization according to the RLT by Sherali-Adams* (2.47) *is tighter than the LP relaxation of the trivial linearization* (2.9), *when projected on the space of the decision variables* $x_i,\ i \in [n+m]$.

Proof. Let $\hat{X}$ denote the feasible set of the LP relaxation based on the trivial linearization (2.9), containing tuples $(x,y) \in \mathbb{R}^{n+m} \times \mathbb{R}^{Nz}$ and define the respective projected sets as:

$$\hat{X}_P = \left\{ x \in \mathbb{R}^{n+m} \mid (x,y) \in \hat{X} \right\} \text{ and} \tag{2.48}$$

$$X'_{P1} = \left\{ x \in \mathbb{R}^{n+m} \mid (x,w) \in X_1' \right\}. \tag{2.49}$$

In order to prove that the Sherali-Adams linearization results in an LP relaxation for the decision variables, which is at least as tight as the one of the trivial linearization, we have to prove that its projected set X'_{P1} is a subset of $\hat{X}_P$. To this end, it is sufficient to see that all constraints contained in $\hat{X}$ are implied by constraints of X_1'. Indeed, all nonnegativity constraints and the conflict set constraints of $\hat{X}$ are also contained in X_1 and thus in X_1'. Therefore, it holds that $\forall (x,w) \in X_1',\ \exists y \in \mathbb{R}^{Nz}$, such that $(x,y) \in \hat{X}$. This means $\hat{X}_P \subseteq X_{P1}$ holds and that the LP relaxation of the linearization in (2.47) is tighter than the trivial linearization given in (2.9), when projected on the space of decision variables.

□

Due to the structure of the constraints involving slack variables, we can note the following:

Proposition 2.18. *Let the original Q-MWIS problem with equality constraints* (2.24) *that we consider here be in standard form.*
Then, the constraints

$$\forall j \in [m], k \in [n+m] \text{ and } k \in K'_j : \quad \sum_{\substack{s \in K'_j \setminus \{k\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \setminus \{k\} \\ t>k}} w_{kt} = 0 \quad (2.50)$$

from the feasible set X'_1 in (2.46) *can be shortened to only consider $k \in [n]$, since the constraints for the slack labels $k \in \{n+1, \dots, n+m\}$ are already implied by the other constraints of X'_1.*

Proof. We consider any conflict set K'_p, $p \in [m]$ and some arbitrary slack label $\hat{k} \in \{n+1, \dots, n+m\}$.
First, let our slack label $\hat{k}$ be an element of K'_p, i.e. $\hat{k} = n+p$ holds. Coupled with the nonnegativity constraints of X'_1 on the linearized terms, namely $w_{ab} \geq 0$, $\forall ab \in [[n+m]]^2$, the corresponding constraint

$$\sum_{s \in K'_p \setminus \{\hat{k}\}} w_{s\hat{k}} = 0 \quad (2.51)$$

from (2.50) is equivalent to $w_{s\hat{k}} = 0$, $\forall s \in K'_p \setminus \{\hat{k}\}$ (note that $s < \hat{k}$ holds for any such s).
It can be observed that for every $s \in K'_p \setminus \{\hat{k}\}$, the term $w_{s\hat{k}}$ is also contained in the sum of the constraint of (2.52), when considering $j = p$ and $k = s$. From the nonnegativity of the linearized terms it directly follows that $w_{s\hat{k}} = 0$ holds $\forall s \in K'_p \setminus \{\hat{k}\}$, without resorting to any constraint of (2.50) or (2.52) with $k \in \{n+1, \dots, n+m\}$, showing the desired. □

Remark 2.19 (Nonnegative cost case)**.** When additionally assuming the Q-MWIS problem in Proposition 2.18 to only have nonnegative cost terms $c_i \geq 0$, $\forall i \in [n]$ and $c_{ik} \geq 0$, $\forall ik \in N_Z$, it appears (from empirical testing) that constraints

$$\forall j \in [m], k \in [n+m] \text{ and } k \notin K'_j : \quad \left(\sum_{\substack{s \in K'_j \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \\ t>k}} w_{kt} \right) - x_k = 0 \quad (2.52)$$

can be reduced to only consider non-slack labels $k \in [n]$, without affecting the optimal objective function value of the corresponding integer linear program.

While this remains to be shown in a more rigorous proof, heuristically this is most likely due to the constraint holding trivially in any optimal solution, implied by the remaining constraints.

These results mostly focused on linearizations for Q-MWIS problems with equality constraints, however some of them also apply to the Q-MWIS problem with inequality constraints. In preparation for the upcoming practical part, where we will also consider it as an alternative problem linearization, we can note:

Proposition 2.20 (Concise constraint set for Q-MWIS linearization with inequality constraints in standard form)**.** *Consider the Q-MWIS problem with inequality constraints as defined in* (2.1) *and let it be in standard form. Then, the RLT according to Sherali-Adams for order* $d = 1$ *yields the constraint set*

$$
\begin{aligned}
Z_1 = \Bigg\{ (x,w) \Bigg| \; & \forall j \in [m], k \in [n] \textit{ and } k \notin K'_j : \\
& \left(\sum_{\substack{s \in K'_j \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \\ t>k}} w_{kt} \right) - x_k \leq 0 \quad \textit{and} \\
& \left(\sum_{i \in K'_j} x_i - \sum_{\substack{s \in K'_j \\ s<k}} w_{sk} - \sum_{\substack{t \in K'_j \\ t>k}} w_{kt} \right) + x_k - 1 \leq 0; \\
& \forall j \in [m], k \in [n] \textit{ and } k \in K'_j : \\
& \sum_{\substack{s \in K'_j \setminus \{k\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K'_j \setminus \{k\} \\ t>k}} w_{kt} \leq 0 \quad \textit{and} \\
& \left(\sum_{i \in K'_j} x_i - \sum_{\substack{s \in K'_j \setminus \{k\} \\ s<k}} w_{sk} - \sum_{\substack{t \in K'_j \setminus \{k\} \\ t>k}} w_{kt} \right) - 1 \leq 0; \\
& \textit{Additionally, } \forall ik \in [[n]]^2 : \\
& w_{ik} \geq 0, \\
& x_i - w_{ik} \geq 0, \\
& x_k - w_{ik} \geq 0, \\
& w_{ik} - x_i - x_k + 1 \geq 0 \Bigg\}
\end{aligned}
\tag{2.53}
$$

and Z_1 can equivalently be reformulated to

$$
\begin{aligned}
Z_1' = \Bigg\{ (x,w) \Bigg| & \forall j \in [m], k \in [n] \ \textit{and} \ k \notin K_j' : \\
& \left(\sum_{\substack{s \in K_j' \\ s<k}} w_{sk} + \sum_{\substack{t \in K_j' \\ t>k}} w_{kt} \right) - x_k \leq 0 \quad \textit{and} \\
& \left(\sum_{i \in K_j'} x_i - \sum_{\substack{s \in K_j' \\ s<k}} w_{sk} - \sum_{\substack{t \in K_j' \\ t>k}} w_{kt} \right) + x_k - 1 \leq 0; \\
& \forall j \in [m], k \in [n] \ \textit{and} \ k \in K_j' : \\
& \sum_{\substack{s \in K_j' \setminus \{k\} \\ s<k}} w_{sk} + \sum_{\substack{t \in K_j' \setminus \{k\} \\ t>k}} w_{kt} \leq 0 \quad \textit{and} \\
& \left(\sum_{i \in K_j'} x_i - \sum_{\substack{s \in K_j' \setminus \{k\} \\ s<k}} w_{sk} - \sum_{\substack{t \in K_j' \setminus \{k\} \\ t>k}} w_{kt} \right) - 1 \leq 0; \\
& \forall ik \in [[n]]^2 : \ w_{ik} \geq 0, \\
& \forall i \in [n] : \ x_i \geq 0 \Bigg\}.
\end{aligned} \tag{2.54}
$$

Proof. The procedure to acquire these results is completely analogous to the case with equality constraints, with only one small difference: Since Remark 2.11 doesn't apply for inequality constraints, we can't replace the constraint set that is acquired through multiplication with polynomials $(1 - x_k)$, $\forall k \in [n]$, with conflict set constraints.
However, as long as the problem is in standard form, we can see that Lemmas 2.13 and 2.14 still work exactly the same, despite some of the equality constraints being inequalities, resulting in the more concise constraint set Z_1'. □

In summary, we have thus far introduced the Q-MWIS problem and various reformulations thereof – namely the trivial linearization and the linearization according to the Sherali-Adams scheme, both with and without equality constraints. In the upcoming part, we will focus more on the practical part of solving Q-MWIS problems, attempting to put to use our alternative formulations.

3 Solving Q-MWIS Problems

Since the Q-MWIS problem is a generalization of the NP-hard MWIS problem (see Proposition 1.6), it is also NP-hard. As such, unless P=NP, there can't be a polynomial-time algorithm solving all problem instances. However, this doesn't necessarily imply intractability for problems of larger scale. Indeed, one may observe in practice that the MWIS problem is efficiently solvable for problem instances with $\leq 10^6$ variables and constraints, by employing off-the-shelf ILP solvers.
Due to the number of quadratic terms, Q-MWIS problems may scale differently though. In the worst case, if all label pairs $[[n]]^2$ have non-zero costs, the size of the problem grows rapidly – adding up to $\frac{n(n-1)}{2}$ quadratic terms to the objective function.

In Section 1.3, we have already discussed current means of solving MWIS problems: Besides the algorithms specifically tailored to it, they can also be directly fed into generic ILP solvers. While we could linearize the quadratic terms in the Q-MWIS problems, the constraints we have to add for this purpose change the structure of the problem. Hence, these MWIS algorithms can't be applied to Q-MWIS problems without adjustments. The possibility to use ILP solvers still remains, though.
For instance, the solver Gurobi [3] supports both linear and quadratic integer programs, which would even allow us to directly input Q-MWIS problems. Whether it is preferable to do so, or instead reformulate the problem first, is not clear yet, since both have their up- and downsides: While the original Q-MWIS problem is fairly straightforward in terms of variables and constraints, it still contains quadratic terms, which might impede the optimization process. On the other hand, the linearizations add a vast number of variables and constraints, but get rid of the quadratic terms, turning the problem into a pure ILP. In the case of the Sherali-Adams linearization, we additionally have the advantage of a tightened LP-relaxation.

This last section will focus on two topics: First, discussing ideas on how to construct a specialized algorithm for the Q-MWIS problem and second, which of the previously constructed reformulations of the Q-MWIS problem perform best, when solving simulated problem instances of varying size and structure with Gurobi.

3.1 An algorithm for the Q-MWIS problem

An initial idea for a solution algorithm for Q-MWIS problems, was the construction of a primal-dual algorithm. While the primal part includes methods to improve approximate solutions we acquire, the dual part produces said approximations, when solving the Lagrange dual problem. However, in the process of formulating the algorithm it became evident that its convergence speed is unlikely to prove satisfying. Nevertheless, we will briefly describe the intended methodology, as parts of it could still be used in a different algorithm.
The primal-dual algorithm is composed of the following two general parts:

1. On the dual domain, we consider the Lagrange dual problem that is acquired by dualizing the constraints of our Sherali-Adams linearization, which enforce $w_{ik} = x_i x_k$, $\forall ik \in [[n+m]]^2$. The resulting ILPs form closely matches MWIS problems and is solved for the current Lagrange multipliers in each update step using Gurobi. This yields a solution that will then be used to compute a subgradient for the Lagrange multipliers, which are subsequently updated.

2. On the primal domain, we use the approximate solutions we acquired in the dual part and attempt to improve them using a recombination heuristic, which merges two approximate solutions, resulting in an objective value that is at least as good as the one of the formerly best approximation. Since finding the optimal crossover of two solutions is an NP-hard problem in itself, we will look into ways to approximate it.

Following this general description of the algorithm, we first go into detail with both the dual and primal methods we apply.

The Lagrange dual problem

To this end, we start out by constructing the Lagrange dual for the Q-MWIS problem, reformulated according to the Sherali-Adams scheme. Generally speaking, the purpose of the lagrange relaxation is to loosen problematic constraints, without which the problem would become easier to solve (at the cost of accuracy). In our case, this would be the constraints binding the new linear terms of our linearization with the quadratic ones $x_i x_k$.

In the following, we consider the non-trivial linearization with constraint set X_1 described in (2.29) - (2.34) that we acquired after applying the Sherali-Adams scheme on the Q-MWIS problem.

We dualize the $4\cdot\left(\frac{1}{2}(n+m-1)(n+m)\right)$ constraints ensuring that $w_{ik} = x_i x_k$ holds, $\forall ik \in [[n+m]]^2$, namely

$$\begin{aligned} \forall ik \in [[n+m]]^2,\ \text{i.e. } i<k:& \\ w_{ik} &\geq 0, \\ x_i - w_{ik} &\geq 0, \\ x_k - w_{ik} &\geq 0, \\ w_{ik} - x_i - x_k + 1 &\geq 0. \end{aligned} \tag{3.1}$$

To do this, the constraints are moved to the objective function with a Lagrange multiplier (see preliminaries), yielding the Lagrange dual function

$$\begin{aligned} \mathcal{D}(\lambda) = & \max_{\substack{x\in\{0,1\}^{n+m} \\ w\in\{0,1\}^{\frac{1}{2}(n+m-1)(n+m)}}} \sum_{i=1}^{n} c_i x_i + \sum_{ik\in N_Z} c_{ik} w_{ik} + \\ & + \sum_{\substack{jl\in[[n+m]]^2 \\ (j<l)}} \lambda^{(1)}_{jl} w_{jl} + \lambda^{(2)}_{jl}(x_j - w_{jl}) + \lambda^{(3)}_{jl}(x_l - w_{jl}) + \lambda^{(4)}_{jl}(w_{jl} - x_j - x_l + 1), \end{aligned} \tag{3.2}$$

$$\begin{aligned} \text{s.t. } \left(\sum_{i\in K'_j} w_{ik}\right) - x_k = 0,\ & \forall j \in [m], k \in [n+m] \text{ and } k \notin K'_j, \\ \sum_{i\in K'_j\setminus\{k\}} w_{ik} = 0,\ & \forall j \in [m], k \in [n+m] \text{ and } k \in K'_j, \\ \sum_{i\in K'_j} x_i = 1,\ & \forall j \in [m], \end{aligned} \tag{3.3}$$

with $\lambda = (\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)}, \lambda^{(4)}) \in \mathbb{R}^{2(n+m-1)(n+m)}_{\geq 0}$, where $\lambda^{(i)}$, with $i = 1, \ldots, 4$, are the respective Lagrange multipliers for each of the four types of constraints described in (3.1), which we dualize.
By reformulating the Lagrange dual function to emphasize the cost terms associated with the decision variables (x, w) and considering that label pairs

$ik \in [[n+m]]^2$ are always taken to be $i < k$, we get

$$\mathcal{D}(\lambda) = \max_{\substack{x \in \{0,1\}^{n+m} \\ w \in \{0,1\}^{\frac{1}{2}(n+m-1)(n+m)}}} \sum_{i=1}^{n+m} \left(c_i + \left(\sum_{l=i+1}^{n+m} \lambda_{il}^{(2)} - \lambda_{il}^{(4)} \right) + \left(\sum_{j=1}^{i-1} \lambda_{ji}^{(3)} - \lambda_{ji}^{(4)} \right) \right) x_i + \sum_{jl \in [[n+m]]^2} \left(c_{jl} + \lambda_{jl}^{(1)} - \lambda_{jl}^{(2)} - \lambda_{jl}^{(3)} + \lambda_{jl}^{(4)} \right) w_{jl} + \sum_{jl \in [[n+m]]^2} \lambda_{jl}^{(4)}, \tag{3.4}$$

s.t. the constraints (3.3) hold.

It should be noted that we have $c_i = 0$, $\forall i = n+1, \ldots, n+m$ for the unary cost terms and $c_{jl} = 0$ for all pairwise cost terms with $jl \notin N_Z$.
To solve the Lagrange dual problem, we have to find the $\lambda \in \mathbb{R}_{\geq 0}^{2(n+m-1)(n+m)}$, which gives us the tightest upper bound of all Lagrange relaxations, minimizing the Lagrange dual function:

$$\min_{\lambda \in \mathbb{R}_{\geq 0}^{2(n+m-1)(n+m)}} \mathcal{D}(\lambda). \tag{3.5}$$

As the maximum over linear functions, the Lagrange dual is convex piecewise linear (see chapter 5.2 of [5]) and therefore not differentiable. Since we are looking for the minimum of a convex function, we could still resort to the subgradient method in order to solve the Lagrange dual problem, which also produces approximate solutions with every iteration for the primal part of the algorithm.

However, due to the large number of Lagrange multipliers and the generally fairly slow convergence speed of the subgradient method, this seems unlikely to provide satisfying results in practical settings.
As an alternative for regular subgradient methods, using a conjugate subgradient method is considered in [23], with the method being described in [24]. Another option is based on the *method of weighted dual averages* in [25] and has very low requirements on the function that is to be optimized, in particular not requiring it to be differentiable or Lipschitz continuous, while still achieving the optimal convergence rate for first-order methods [26]. Whether this would sufficiently increase the convergence speed isn't clear though.

Following this outline of the dual part, we briefly describe how the primal part of an algorithm could look like – which could also easily be re-used as a building-block in any other algorithm that produces approximate solutions.

Recombination of approximate solutions

On the primal part, we consider an "optimized crossover" heuristic, akin to the ones used in genetic algorithms, which aims to optimally merge two approximate solutions into one, with at least as high objective function value as both of the parent solutions.
The general procedure for genetic algorithms, which were originally proposed in [27], is to pick some approximate solutions from a pool and recombine them, using a crossover operator: If a variable choice coincides in both original solutions, it remains the same in the recombined one. For differing solution variables, one value is selected according to the crossover operator. Additionally, so as to to generate new solution candidates, solutions are mutated in some way at specific intervals
Among others, such heuristic algorithms have since been employed for the maximum-weight clique [28] and independent set problems [29], using an "optimized crossover" operator. Said operator is optimal in the sense that it returns a recombination of the two parent solutions with the highest possible objective value.

As we already have the means of producing approximate solutions from the dual part, only the recombination part of such algorithms is required here. Furthermore, we note that the Lagrange dual problem in (3.4) could be seen as a MWIS problem with reduced costs, potentially allowing us to apply the crossover method of [28] for maximum-weight clique problems, which can be easily reduced to MWIS problems, by considering the complement graph instead.
In practice, the optimal recombination of two solutions for MWIS problems can be efficiently implemented in polynomial time, by reducing it to a min-st-cut problem, as described in [30].

3.2 Performance of the Q-MWIS problem formulations in ILP solvers

Following these considerations on an algorithm addressing Q-MWIS problems, we proceed with a test on how the different problem formulations we constructed in Section 2 perform, when they're fed into off-the-shelf ILP solvers. For this purpose, we consider six different formulations, generate some problem instances of varying size and structure and solve them with Gurobi [3], in order to see which one performs best for certain problem structures.

In practice, when trying to solve problem instances as fast as possible, we would prepend some transformations of the conflict set structure, according to Proposition 1.3. This would be to exploit the benefit of larger conflict sets, namely a tighter LP relaxation, as has been described in Example 1.7 for instance. As the focus here is on comparing the different formulations, we refrain from doing this.

We also note that, if all unary and pairwise costs associated with some label i, which is not contained in any conflict set, are positive/negative, the corresponding decision variable could be set to 1/0 respectively, before solving the model. Since this just increases the model setup time, while reducing the optimization time uniformly for all models only in special cases, we forgo this process in the implementation.

The Q-MWIS problem formulations

In Section 2, we considered two ways of writing the conflict sets, using either inequalities or a reformulation to equality constraints. Additionally, we constructed two linearizations of the standard formulation, resulting in a total of six possible ways to (re-)formulate Q-MWIS problems:

Q-MWIS1: The original quadratic integer problem with inequality constraints, as stated in Definition 2.1.

Q-MWIS2: Also a QIP like Q-MWIS1, but with slack variables, which turn the inequality conflict sets constraints into equations, as given in Definition 2.2.

Q-MWIS3: The "trivial linearization" akin to the one of Example 2.6, except without the reformulation to equality conflict set constraints. As the quadratic terms are linearized, this yields an ILP.

Q-MWIS4: Same as Q-MWIS3, except with equality conflict set constraints, see Example 2.6.

Q-MWIS5: The ILP we acquire after applying the RLT according to Sherali-Adams on the Q-MWIS problem with inequality constraints, yielding the constraint set stated in Proposition 2.20.

Q-MWIS6: Another linearization according to the Sherali-Adams scheme, except applied on the Q-MWIS problem reformulation with equality constraints, resulting in the ILP problem given in (2.47).

Since the formulations Q-MWIS5 and 6 require the problem to be in standard form, which we don't enforce in our problem instance generation, we partially add the constraints

$$\begin{aligned} x_i - w_{ik} &\geq 0, \\ x_k - w_{ik} &\geq 0 \text{ and} \\ w_{ik} - x_i - x_k + 1 &\geq 0 \end{aligned} \tag{3.6}$$

back to the formulation for all $i, k \in [n]$, where i or k (or both) are not contained in any conflict set.
Additionally, our implementation applies the results of Proposition 2.18 and use Remark 2.19 in the case of nonnegative costs, in order to reduce the number of constraints that are not essential to the problem.

An overview of the models used is given in the following table:

Problem formulation	**Conflict set type**	**Linearization**
Q-MWIS1	inequalities	none
Q-MWIS2	equations	none
Q-MWIS3	inequalities	trivial
Q-MWIS4	equations	trivial
Q-MWIS5	inequalities	Sherali-Adams
Q-MWIS6	equations	Sherali-Adams

Table 1: Q-MWIS problem (re-)formulations used in the Gurobi comparison.

Performance comparison of problem reformulations

In order to compare how the reformulations of Q-MWIS problems perform in ILP-solvers, we generate example problem instances with Python and solve them using Gurobi [3], with each problem formulation. The computations are done on a system with 2.50GHz dual core CPU (Intel Core i7-6500U)

and 8GB RAM.
Unless specified otherwise, the optimizations are carried out with Gurobi's default settings and are set to time out after 5 minutes. Gurobi's general optimization routine is:

1. Very briefly generate approximate solutions with heuristics.

2. Presolve the model, reducing its dimensionality in terms of decision variables and constraints.

3. Solve the LP relaxation of the presolved model ("root relaxation"), typically using barrier/interior point methods in the case of QIPs and a dual simplex algorithm for (smaller) ILPs. If the size of the root relaxation is large, the default choice "deterministic concurrent" runs several algorithms on multiple threads at the same time and chooses the one that finishes first.

4. Apply a branch-and-cut algorithm until the problem is solved (MIP gap of 0%, i.e. the objective value of the optimal solution and its upper bound coincide) or timeout occurs.

Problem set 1 – smaller problems with conflict sets of size 2

In this set of problem instances, the focus is on smaller problems that have a fixed conflict set size of $|K_j| = 2$, $\forall j \in [m]$, with the number of conflict sets m being equal to the number of (unary) decision variables n. Therefore, the problems are unlikely to be in standard form.
The unary and pairwise costs are randomly distributed integers in the interval $[1, 10]$ and the pairwise cost matrix density is 100%, i.e. $N_Z = [[n]]^2$ holds for the original Q-MWIS problem.

We can see in Table 2 that the default QIP formulations yield results fastest for the smaller problem instances, while the Sherali-Adams linearization with equality conflict set constraints Q-MWIS6 outperforms the others in larger problems. The trivial linearizations perform worse than the Sherali-Adams one in every case, but better than the default formulation for larger problems.
It also appears that the equality constraint reformulations perform marginally better, with the Sherali-Adams linearization showing the largest improvement due to it.

n	m	$\lvert K_j \rvert$	Average solution time of model (in s) Q-MWIS	1	2	3	4	5	6
20	20	2		**0.03**	**0.03**	0.05	0.05	0.08	0.05
40	40	2		**0.09**	**0.09**	0.37	0.37	0.32	0.2
60	60	2		0.36	**0.34**	1.03	1.09	0.82	0.6
80	80	2		2.28	2.19	2.95	2.86	1.69	**1.12**
100	100	2		16.73	16.81	6.64	6.26	4.84	**2.54**

Table 2: Average time it took to solve the respective Gurobi models to optimality, i.e. until the MIP gap is 0%. The solution time is an average for 25 problem instances and the fastest models are highlighted.

Problem set 2 – smaller problems with scaling conflict set size

So as to investigate the impact of the conflict set size/number on the performance of our problem formulations, we consider problems similar to problem set 1 here, except that we set the number of conflict sets $m = \frac{n}{4}$ and their size $|K_j| = \frac{n}{10}$, for n (unary) decision variables. The other parameters are the same, except for the costs, which are set to be integer in the range $[1, 3]$. The results are:

n	m	$\lvert K_j \rvert$	Average solution time of model (in s) Q-MWIS	1	2	3	4	5	6
20	5	2		**0.01**	**0.01**	0.04	0.03	0.05	0.04
40	10	4		**0.04**	**0.04**	0.21	0.23	0.15	0.13
60	15	6		**0.27**	**0.27**	0.86	0.96	0.35	**0.27**
80	20	8		12.62	12.96	3.05	3.48	0.68	**0.5**

Table 3: Average time it took to solve the respective Gurobi models to optimality, i.e. until the MIP gap is 0%. The solution time is an average for 25 problem instances and the fastest models are highlighted.

While the picture is very similar to the results of problem set 1, the difference between the linearized problem formulations and the default QIP ones appears to be increasing with larger conflict set sizes. Indeed, this would make sense, considering how the total number of constraints, particularly for the Sherali-Adams linearization, increases with the number of conflict set constraints in the original Q-MWIS problem.

Problem set 3 – mid-sized problems with various conflict set sizes and numbers

Next, we increase the size of the problems to $n = 150$ (unary) decision variables and consider various combinations of conflict set numbers m and sizes $|K_j|$, $\forall j \in [m]$. The cost terms are integers in the interval $[1, 10]$ and $N_Z = [[n]]^2$ still holds for the original Q-MWIS problem, i.e. all pairwise costs are non-zero.
This yields the following results:

			Average solution time of model (in s), Number of models solved to optimality and Average MIP gap (in %)						
n	**m**	$\|K_j\|$	**Q-MWIS**	**1**	**2**	**3**	**4**	**5**	**6**
				206.05	206.17	33.49	34.29	20.22	**5.25**
150	150	2		2	2	3	3	3	**3**
				0.83	0.91	0	0	0	**0**
				300+	300+	271.96	271.54	300+	**231.04**
150	150	4		0	0	1	1	0	**1**
				29.61	29.65	21.58	21.61	1000+	**2.34**
				56.4	57.57	300+	300+	300+	300+
150	150	10		**3**	3	0	0	0	0
				0	0	61.72	66.32	1000+	45.87
				19.79	20.28	4.43	4.74	2.44	**2.19**
150	50	2		3	3	3	3	3	**3**
				0	0	0	0	0	**0**
				300+	300+	94.56	98.07	60.74	**4.81**
150	50	10		0	0	3	3	3	**3**
				89.02	89.57	0	0	0	**0**
				39.8	40.7	141.28	155.52	116.01	82.55
150	50	20		**3**	3	3	3	3	3
				0	0	0	0	0	0
				300+	300+	16.5	16.56	1.69	**1.39**
150	10	20		0	0	3	3	3	**3**
				6.39	7.0	0	0	0	**0**
				233.86	237.79	14.94	14.89	1.58	**1.12**
150	10	30		3	3	3	3	3	**3**
				0	0	0	0	0	**0**
				10.15	9.96	9.06	9.06	1.42	**0.91**
150	10	50		3	3	3	3	3	**3**
				0	0	0	0	0	**0**

Table 4: Average time it took to solve the respective Gurobi models, with a time limit of 5 minutes. Additionally, the number of models solved to optimality and the average MIP gap is displayed. For each parameter combination, 3 problems were solved and the formulation that performed best in all factors was highlighted.

It is interesting to see that the linearizations, particularly Q-MWIS6, perform very well up to a certain point of constraint size and number. When looking into the console output, one can see that this is due to the solver struggling to solve the root relaxation for problems with large numbers of conflict sets – if it can't be solved in the set time limit, the best solution returned is the (very bad) initial heuristic solution, resulting in a huge MIP gap. As the QIP formulations Q-MWIS1 and 2 have a relatively simple LP relaxation, they are not affected by this, but also can't close the MIP gap to optimality very well.
Potentially, the performance of the Q-MWIS linearizations could be improved here, by fine-tuning parameters or running the computations on a setup with more RAM. The latter might prevent memory issues, when the barrier method is used to solve the elaborate root relaxations of the linearizations, allowing them to proceed quicker to the branch-and-cut phase.

Problem set 4 – mid-sized problems with different cost ranges

So far, the considered problem instances all had non-negative costs, allowing particularly the Sherali-Adams linearization to shine. To see how partially negative costs affect the performance of the different formulations, we generate mid-sized problems with $n = 150$ decision variables, $m = 50$ conflict sets of size $|K_j| = 5$, $\forall j \in [m]$ and various (integer) cost ranges, where almost all pairwise cost terms are non-zero.

(integer) cost range	Average solution time of model (in s), Number of models solved to optimality and Average MIP gap (in %) Q-MWIS 1	2	3	4	5	6
$[1, 10]$	300+ 0 12.48	300+ 0 12.55	28.12 10 0	28.86 10 0	4.52 10 0	**2.16** **10** **0**
$[-10, 10]$	**300+** **0** **30.19**	300+ 0 30.35	300+ 0 232.24	300+ 0 233.26	300+ 0 257.98	300+ 0 319.96
$[-100, 100]$	**300+** **0** **23.57**	300+ 0 23.66	300+ 0 223.46	300+ 0 218.41	300+ 0 234.71	300+ 0 256.41
$[1, 1000]$	300+ 0 16.05	300+ 0 16.17	39.74 10 0	39.68 10 0	7.47 10 0	**3.7** **10** **0**

Table 5: Average results for problem instances with parameters $n = 150$, $m = 50$, $|K_j| = 5$ and different cost ranges. For each cost range, 10 problems were solved and the formulation that performed best across all factors was highlighted.

We can derive the following two conclusions from the results in Table 5: First, that the cost range doesn't matter too much, as long as it doesn't include negative costs and second that the introduction of negative cost terms makes a problem significantly harder to solve, particularly for the linearizations.

Looking into the optimization logs, it appears that the negative costs vastly reduce the effectivity of the linearizations, by reducing the tightness of their root relaxations – negating the purpose of these linearizations. Due to this, it seems unlikely that even fine-tuning the Gurobi parameters would yield a significant improvement.

Problem set 5 – large problem instances

After testing the problem formulations on mostly small and mid-sized problem instances, we proceed with larger ones. To lower the computational load here, we only consider the two strongest formulations so far, namely Q-MWIS1 and Q-MWIS6. As we have seen, the latter struggles to yield any solution for larger problems, since the root relaxation can't be solved using the barrier method (for memory reasons) and the simplex algorithms take too long. Thus, we omit the solving of the root relaxation and the subsequent branch-and-cut procedure, relying purely on heuristics.

The original Q-MWIS problem parameters are $n = 1500$ decision variables, $m = 100$ conflict sets of size $|K_j| = 50$, $\forall j \in [m]$ and integer costs in the ranges $[1, 100]$ and $[-100, 100]$. Like for the previous problem, almost all pairwise cost terms are non-zero, yielding a total of over a million pairwise terms. Due to the large problem size, we increased the optimization time limit to 15 minutes per model.

(integer) cost range	**Average MIP gap (in %)**		
	Q-MWIS1	**Q-MWIS1 (heur)**	**Q-MWIS6 (heur)**
$[1, 100]$	**387.57**	11230.34	501.69
$[-100, 100]$	**174.4**	47269.81	38745.9

Table 6: Average MIP gap for problems with $n = 1500$ decision variables, $m = 100$ conflict sets of size $|K_j| = 50$, $\forall j \in [m]$ and two different cost ranges. The MIP gap is an average over 3 problem instances and the fastest models are highlighted.

As expected, the results of Table 6 show a large MIP gap across all considered models. Looking into the optimization log for the non-negative cost range problems, it can be seen that Q-MWIS1 yields a better objective value and bound than the other models fairly quickly, but stagnates at this level with little to no improvement. On the other hand, the Sherali-Adams linearization with equality constraints Q-MWIS6 starts out with fairly weak solutions, but improves them at a quicker rate than Q-MWIS1. When the time limit is increased to e.g. one hour, Q-MWIS6 outperforms Q-MWIS1 with heuristics only, in terms of the MIP gap.
However, this is probably not the case, when negative cost terms are included, considering the greatly larger MIP gap of Q-MWIS6 with heuristics. When employing heuristic methods only, it can be noted that the Sherali-Adams linearization performs better than the default Q-MWIS1 with heuristics – since the Q-MWIS1 model performs significantly worse using just heuristics, instead of solving the root relaxation and using the branch-and-cut procedure, there is no point in choosing this method, though.

Problem set 6 – problems with differing pairwise cost matrix densities

Lastly, we look into the effect that the amount of non-zero pairwise cost terms has on the performance of our problem formulations. To this end, we consider mid-sized problems with $n = 200$ decision variables, $m = 100$ conflict sets of size $|K_j| = 10$, $\forall j \in [m]$ and nonnegative integer costs in the range of $[1, 20]$, with varying amounts of non-zero pairwise cost terms. For density $d \in [0, 1]$, they are given by

$$|N_Z| = d \cdot |[[n]]^2| = \frac{d \cdot n(n-1)}{2}. \tag{3.7}$$

We can see from the results in Table 7 that the Sherali-Adams linearization with equality constraints Q-MWIS6 noticeably outperforms the other models for high density values, but falls off for low density values, where the non-linearized formulations solve the problems fastest. If not accounting for the model setup time for Q-MWIS6 of roughly 1 second, its performance becomes comparable to formulations Q-MWIS1 to Q-MWIS4 even for very low pairwise cost density rates.
Q-MWIS5 however, consistently performs worst throughout all density values, since the solver fails to solve the root relaxation in the time limit for all but one problem instance. This result is not surprising, considering how the Sherali-Adams linearization adds terms w_{ik}, $\forall ik \in [n]^2$ regardless of whether

d (density)	Average solution time of model (in s), Number of models solved to optimality and Average MIP gap (in %)						
	Q-MWIS	**1**	**2**	**3**	**4**	**5**	**6**
100%		300+	300+	282.43	279.08	292.41	**99.39**
		0	0	1	1	1	**3**
		288.08	287.23	32.58	41.4	1000+	**0**
50%		300+	300+	165.11	159.74	300+	**64.38**
		0	0	3	3	0	**3**
		210.43	213.9	0	0	1000+	**0**
20%		43.21	46.87	22.64	22.8	300+	**21.14**
		3	3	3	3	0	**3**
		0	0	0	0	1000+	**0**
10%		**6.14**	6.83	6.4	6.79	300+	6.47
		3	3	3	3	0	3
		0	0	0	0	1000+	0
5%		**1.19**	1.46	2.61	2.68	300+	2.76
		3	3	3	3	0	3
		0	0	0	0	1000+	0
1%		0.18	**0.16**	0.29	0.25	300+	1.48
		3	**3**	3	3	0	3
		0	**0**	0	0	1000+	0

Table 7: Average time it took to solve the respective Gurobi models, with a time limit of 5 minutes. Additionally, the number of models solved to optimality and the average MIP gap is displayed. Problem instance parameters were $n = 200$, $m = 100$, $|K_j| = 10$ with random integer cost terms in the interval [1,20]. For each pairwise cost density rate d, 3 problems were solved and the formulation that performed best in all factors was highlighted.

there is a corresponding pairwise cost term in the objective function or not. Since this is not the case for the trivial linearizations, they perform noticeably better here.
The large solving speed difference of Q-MWIS6 over Q-MWIS5 possibly is attributable to the constraints omitted according to Proposition 2.18 and Remark 2.19, which prevent cluttering the problem with some non-essential constraints.

Summary of results

Generally, it appears that the best choice usually is either the original formulation Q-MWIS1 or the reformulation according to Sherali-Adams with equality constraints, i.e. Q-MWIS6. Moreover, the difference between the inequality and equality conflict set variants seems to be negligible for all but the formulations Q-MWIS5 and 6, where the latter commonly performs noticeably better.

When also considering the structure of a problem instance, we find the following indications, which problem formulation should be used:
If the problem at hand has nonnegative costs, a medium amount of decision variables n, preferably fewer, larger conflict sets and fewer zero pairwise cost terms, the formulation Q-MWIS6 performs strictly better than all other formulations, usually by a margin.
For large problems this might still hold true, but requires further fine-tuning of the optimization parameters, since the root relaxation becomes difficult and very time consuming to solve. Doing the computations on a setup with more memory could help, too.
In the cases, where the costs can be both positive and negative, the problem is noticeably harder to solve with all formulations – particularly however, for the linearizations. Hence, using the default formulation Q-MWIS1 could be the best course then.

After looking into how the different problem formulations perform in practice, when employing the off-the-shelf solver Gurobi, we finish up with a brief summary and outlook.

4 Conclusion

Over the course of this thesis, we introduced the Quadratic Maximum-Weight Independent Set (Q-MWIS) problem and looked into ways of solving it, partially through linearization and tightening of the corresponding LP relaxation.
To this end, after laying the foundation in the Preliminaries 0, we followed up with the Introduction 1, where we started out with the MWIS problem – looking into its computational complexity, the current state of algorithms addressing it, as well as the influence of the conflict set structure describing the problem, particularly on its LP relaxation.
Next, we motivated and defined the Q-MWIS problem in Section 2, looking into how it connects to other optimization problems like the QAP and MAP-Inference for graphical models, as well as introducing notions that are used later on. As part of a potential solution algorithm and to see, whether it yields a computational advantage, when used with off-the-shelf ILP solvers, we consider two linearizations of the quadratic MWIS problem: A "trivial" one, which simply replaces the quadratic terms with new decision variables and binds the two using constraints, and one that is according to the reformulation-linearization technique (RLT) by Sherali-Adams [2], which provides a tighter LP relaxation. After looking into how said RLT works, we apply it on the Q-MWIS problem and prove that some of the resulting constraints can be redundant, suggesting some more concise formulations.
In the final, practical Section 3, we first briefly considered a potential algorithm for the Q-MWIS problem in form of a primal-dual algorithm and lastly tested, how the different reformulations we constructed in Section 2 perform, when used to solve Q-MWIS problems with Gurobi [3]. This was done generating several Q-MWIS problem instance sets with different problem structures, in order to see how factors like decision variable number, conflict set size and cost term range affect the performance of the respective formulations.
We found that the reformulation according to the RLT of Sherali-Adams, applied to the Q-MWIS formulation with equality constraints, usually outperforms all others, as long as the costs are nonnegative. If the problem is so large in size that solving the LP relaxation of the linearization becomes too computationally expensive, optimization parameters have to be adjusted, possibly to employ purely heuristic methods.

Further work on this topic could be twofold: First, one might look deeper into which constraints added in the RLT according to Sherali-Adams are of use for various problem structures. Indeed, particularly for extremely sparse

pairwise cost matrices, it seems inefficient to add new variables for every possible pairwise label. An approach applying more targeted lift-and-project cuts according to [8] could prove more efficient.
Secondly, the algorithmic part can be greatly extended, looking into how the structure resulting from the Sherali-Adams linearization can be better exploited or how the Q-MWIS problem could be solved or approximated, without using linearizations. Some ideas for this might be derived from the algorithms addressing the linear MWIS problem, discussed in Subsection 1.3.

5 Appendix

The Python code used to generate and solve the problems of Subsection 3.2 can be found at `https://gitlab.com/homul/q-mwis`.

Bibliography

[1] P. Swoboda, D. Kainmüller, A. Mokarian, C. Theobalt, and F. Bernard, "A convex relaxation for multi-graph matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11156–11165, 2019.

[2] H. D. Sherali and 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, no. 3, pp. 411–430, 1990.

[3] "Gurobi Optimizer: The state-of-the-art mathematical programming solver." See `http://www.gurobi.com/`.

[4] P. Vaidya, "Speeding-up linear programming using fast matrix multiplication," in *30th Annual Symposium on Foundations of Computer Science*, pp. 332–337, 1989.

[5] B. Savchynskyy, "Discrete graphical models — an optimization perspective," *Foundations and Trends® in Computer Graphics and Vision*, vol. 11, no. 3-4, pp. 160–429, 2019.

[6] R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society*, vol. 64, no. 5, pp. 275 – 278, 1958.

[7] A. Zanette, M. Fischetti, and E. Balas, "Lexicography and degeneracy: can a pure cutting plane algorithm work?," *Mathematical Programming*, vol. 130, pp. 153–176, 2011.

[8] E. Balas, S. Ceria, and G. Cornuéjols, "A lift-and-project cutting plane algorithm for mixed 0–1 programs," *Mathematical Programming*, vol. 58, pp. 295–324, 1993.

[9] E. Balas, S. Ceria, and G. Cornuéjols, "Mixed 0-1 programming by lift-and-project in a branch-and-cut framework," *Management Science*, vol. 42, pp. 1229–1246, 1996.

[10] Y. Nesterov, *Lectures on Convex Optimization.* Springer Optimization and Its Applications, Springer International Publishing, 2nd ed., 2018.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Series of Books in the Mathematical Sciences, W. H. Freeman, first ed., 1979.

[12] J. Håstad, "Clique is hard to approximate within $n^{1-\epsilon}$," *Acta Mathematica*, vol. 182, no. 1, pp. 105 – 142, 1999.

[13] H. Jiang, C.-M. Li, and F. Manya, "An exact algorithm for the maximum weight clique problem in large graphs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, Feb. 2017.

[14] S. Lamm, C. Schulz, D. Strash, R. Williger, and H. Zhang, "Exactly solving the maximum weight independent set problem on large real-world graphs," in *2019 Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 144–158, SIAM, 2019.

[15] Y. Dong, A. V. Goldberg, A. Noe, N. Parotsidis, M. G. Resende, and Q. Spaen, "A Local Search Algorithm for Large Maximum Weight Independent Set Problems," in *30th Annual European Symposium on Algorithms (ESA 2022)* (S. Chechik, G. Navarro, E. Rotenberg, and G. Herman, eds.), vol. 244 of *Leibniz International Proceedings in Informatics (LIPIcs)*, (Dagstuhl, Germany), pp. 45:1–45:16, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[16] G. J. Minty, "On maximal independent sets of vertices in claw-free graphs," *Journal of Combinatorial Theory, Series B*, vol. 28, no. 3, pp. 284–304, 1980.

[17] P. Nobili and A. Sassano, "An $\mathcal{O}(n^2 \log(n))$ algorithm for the weighted stable set problem in claw-free graphs," *Math. Program.*, vol. 186, no. 1, pp. 409–437, 2021.

[18] V. V. Lozin and M. Milanič, "A polynomial algorithm to find an independent set of maximum weight in a fork-free graph," *Journal of Discrete Algorithms*, vol. 6, no. 4, pp. 595–604, 2008. Selected papers from the 1st Algorithms and Complexity in Durham Workshop (ACiD 2005).

[19] A. Grzesik, T. Klimošová, M. Pilipczuk, and M. Pilipczuk, "Polynomial-time algorithm for maximum weight independent set on p6-free graphs," *ACM Trans. Algorithms*, vol. 18, jan 2022.

[20] F. Glover, G. Kochenberger, R. Hennig, and Y. Du, "Quantum bridge analytics I: A tutorial on formulating and using QUBO models," *Annals of Operations Research*, vol. 314, no. 1, pp. 141–183, 2022.

[21] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica: Journal of the Econometric Society*, pp. 53–76, 1957.

[22] R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis, "The quadratic assignment problem," in *Handbook of Combinatorial Optimization: Volume1–3* (D.-Z. Du and P. M. Pardalos, eds.), pp. 1713–1809, Boston, MA: Springer US, 1998.

[23] H. D. Sherali and W. P. Adams, *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, vol. 31. Springer Science & Business Media, 2013.

[24] H. D. Sherali and O. Ulular, "A primal-dual conjugate subgradient algorithm for specially structured linear and convex programming problems," *Applied Mathematics and Optimization*, vol. 20, pp. 193–221, 1989.

[25] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical programming*, vol. 120, no. 1, pp. 221–259, 2009.

[26] M. R. Metel and A. Takeda, "Primal-dual subgradient method for constrained convex optimization problems," *Optimization Letters*, vol. 15, no. 4, pp. 1491–1504, 2021.

[27] J. H. Holland, "Adaptation in natural and artificial systems. an introductory analysis with applications to biology, control and artificial intelligence," *Ann Arbor: University of Michigan Press*, 1975.

[28] E. Balas and W. Niehaus, "Optimized crossover-based genetic algorithms for the maximum cardinality and maximum weight clique problems," *Journal of Heuristics*, vol. 4, pp. 107–122, 1998.

[29] C. C. Aggarwal, J. B. Orlin, and R. P. Tai, "Optimized crossover for the independent set problem," *Operations research*, vol. 45, no. 2, pp. 226–234, 1997.

[30] A. V. Eremeev and J. V. Kovalenko, “Optimal recombination in genetic algorithms for combinatorial optimization problems: Part i,” *Yugoslav Journal of Operations Research*, vol. 24, no. 1, pp. 1–20, 2014.

Declaration of Authorship

I hereby certify that this thesis has been composed by me and is based on my own work, unless stated otherwise. No other person's work has been used without due acknowledgment in this thesis. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged.

Place, Date: Signature: