Convex Optimization Project

Authors

Zahra Maleki - 400110009
Mohammad MohammadianBisheh - 401110342
Mohammad Armin Dehghan - 400101175

2025-02-06

Contents

Introduction

In recent years, the *Semidefinite Relaxation (SDR)* technique has emerged as a pivotal tool in addressing complex optimization challenges, particularly in the domains of signal processing and communications. This method, as discussed in the article **"Semidefinite Relaxation of Quadratic Optimization Problems"**, offers a computationally efficient approximation framework for tackling nonconvex optimization problems, especially quadratically constrained quadratic programs (QCQPs).

In this report, we focus on the application of SDR to two specific problems. These applications highlight the versatility and effectiveness of SDR in solving optimization problems in both medical imaging and wireless communication systems.

1 MIMO Detection

MIMO (Multiple-Input Multiple-Output) systems play a vital role in modern wireless communication, enabling increased data rates and improved reliability. In a MIMO communication system, the received signals can be mathematically modeled as:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$$

Here, $\mathbf{y} \in C^M$ is the received signal vector, $\mathbf{H} \in C^{M\times N}$ is the channel matrix, $\mathbf{s} \in \mathcal{S}^N$ is the transmitted symbol vector where $\mathcal{S}$ represents the modulation symbol set and $\mathbf{n} \in C^M$ is the additive noise vector.

The goal of MIMO detection is to estimate the transmitted symbols $\mathbf{s}$ from the received signals $\mathbf{y}$, given knowledge of the channel matrix $\mathbf{H}$.

The detection task can be formulated as an optimization problem. The objective is to find the transmitted symbol vector $\mathbf{s}$ that minimizes the Euclidean distance between the received signal $\mathbf{y}$ and the reconstructed signal $\mathbf{Hs}$:

$$\min_{\mathbf{s}\in\mathcal{S}^N} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2$$

This problem is inherently non-convex due to the discrete nature of $\mathcal{S}$, making it computationally challenging to solve exactly, especially for large-scale systems.

To make the problem more tractable, it can be reformulated in a real-valued domain. Letting $\mathbf{y}, \mathbf{s}, \mathbf{H}$ be expressed in terms of their real and imaginary parts:

$$\mathbf{y} = \begin{bmatrix}\text{Re}(\mathbf{y})\\ \text{Im}(\mathbf{y})\end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix}\text{Re}(\mathbf{s})\\ \text{Im}(\mathbf{s})\end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix}\text{Re}(\mathbf{H}) & -\text{Im}(\mathbf{H})\\ \text{Im}(\mathbf{H}) & \text{Re}(\mathbf{H})\end{bmatrix}$$

The problem can then be rewritten as:

$$\min_{\mathbf{s}\in R^{2N}, t\in R} \quad \|t\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \quad \text{subject to } t^2 = 1,\ s_i^2 = 1,\ i = 1, \ldots, 2N.$$

To further simplify, this formulation can be expressed as a homogeneous Quadratic Constrained Quadratic Program (QCQP):

$$\min_{\mathbf{s},t} \quad \begin{bmatrix}\mathbf{s}\\ t\end{bmatrix}^T \begin{bmatrix}\mathbf{H}^T\mathbf{H} & -\mathbf{H}^T\mathbf{y}\\ -\mathbf{y}^T\mathbf{H} & \|\mathbf{y}\|^2\end{bmatrix} \begin{bmatrix}\mathbf{s}\\ t\end{bmatrix} \quad \text{subject to } t^2 = 1,\ s_i^2 = 1,\ i = 1, \ldots, 2N.$$

This approach transforms the original detection problem into a form that can be solved or approximated using advanced optimization techniques like Semidefinite Relaxation (SDR).

To address the detection problem, various techniques have been developed:

1. Zero Forcing (ZF): This linear method estimates $\mathbf{s}$ by inverting the channel matrix $\mathbf{H}$:

$$\mathbf{s}_{\text{ZF}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y}.$$

   While computationally efficient, ZF is sensitive to noise amplification, especially when $\mathbf{H}$ is ill-conditioned.

2. Minimum Mean Square Error (MMSE): This method balances noise amplification and interference suppression by incorporating noise statistics:

$$\mathbf{s}_{\text{MMSE}} = (\mathbf{H}^H\mathbf{H} + \sigma_n^2\mathbf{I})^{-1}\mathbf{H}^H\mathbf{y}.$$

3. Sphere Decoding (SD): A non-linear method that searches for the optimal solution within a reduced subset of symbols. It provides higher accuracy but can be computationally expensive for large systems.

4. Semidefinite Relaxation (SDR): By relaxing the original non-convex problem into a convex semidefinite program, SDR offers efficient approximations. Randomization techniques can further refine the solutions obtained via SDR.

5. Lattice Reduction Aided Detection: This preprocessing method transforms the channel matrix $\mathbf{H}$ to make it more orthogonal, simplifying subsequent detection steps.

   The results demonstrate the trade-offs in performance and computational complexity for various MIMO detection techniques. Specifically, SDR achieves superior detection accuracy compared to techniques like Zero Forcing (ZF) and MMSE but at the cost of significantly higher computational complexity. This complexity becomes especially pronounced as the problem size increases due to the intensive optimization involved in SDR.

   The use of only 20 symbols in the simulation impacts both the accuracy and statistical reliability of the results. With a smaller dataset, the bit error probability (BER) metrics may show more variability, and the statistical trends may not be as robust as those obtained with larger symbol counts. However, increasing the number of symbols would lead to smoother BER curves and more accurate assessments of each methodâs performance. For SDR in particular, the accuracy benefits from larger datasets, as the randomized approaches used in SDR refine their approximations more effectively with increased data points.

   On the computational side, as the number of symbols increases, the computational cost for methods like SDR grows significantly, emphasizing the need to balance between accuracy and feasibility in practical implementations.

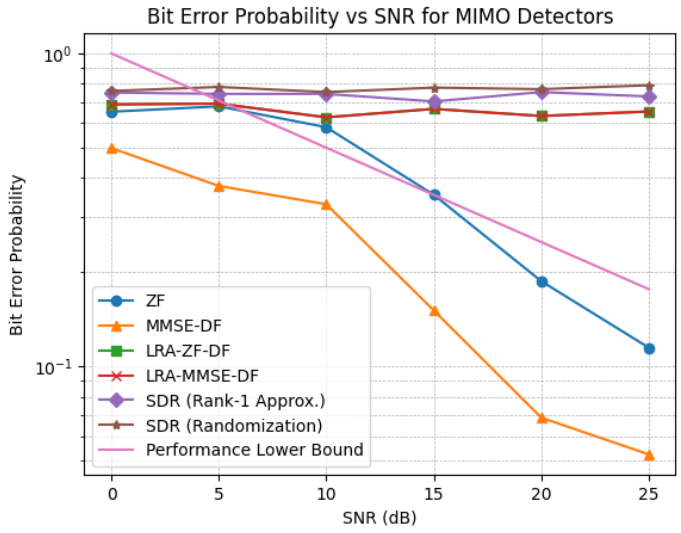

Figure 1: Bit Error Probability vs SNR for various MIMO detectors.

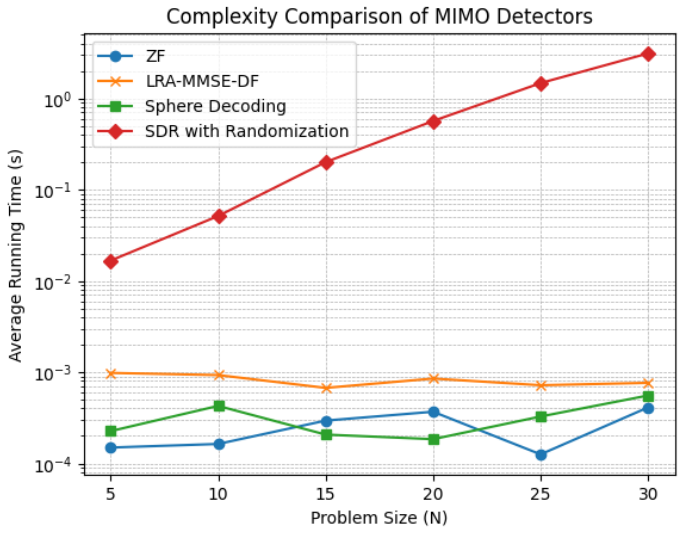

Figure 2: Average Running Time vs Problem Size for various MIMO detectors.

2 Max-Cut Problem

2.1 The problem

The **Max-Cut Problem** is a fundamental NP-hard problem in combinatorial optimization and graph theory, with applications in clustering, circuit design, statistical physics, and machine learning. Given an undirected graph $G = (V, E)$ with edge weights $w : E \to R^+$, the objective is to partition the vertices into two disjoint sets S and T such that the sum of the weights of edges crossing the cut is maximized:

$$\max_{S \subseteq V} \sum_{(u,v) \in E, u \in S, v \in T} w(u, v).$$

This problem is computationally challenging, as finding an exact solution is infeasible for large graphs.

For example,in practice, clustering in machine learning can be categorized into two types: clustering by difference and clustering by similarity. To measure similarity, we construct a complete graph where the edges are weighted based on the similarity between data points. The goal is to partition this graph into smaller subgraphs while minimizing the overall cut cost.

$$cut(C_1, C_2, ..., C_k) = \sum_{i=1}^{k} \sum_{r \in C_i, s \notin C_i} W_{r,s}$$

Minimizing this cost is equal to maximizing the in-graph weights.For large datasets this problem is computationally challenging!

Now consider the $k = 2$ case Max-Cut problem:

Let G be a graph on n vertices. The Max-Cut consists in separating the vertex set V of G into two sets $S, V \setminus S$ such that the number of edges between the set S and the set $V \setminus S$ is as large as possible.

Now let us denote A the adjacency matrix of G, the Max-Cut problem (MC) can be formalized as follows :

$$\max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \tag{MC}$$

where $\delta(S)$ is the set of edges ij such that $i \in S$, $j \notin S$

Now using the ideas we introduced in part 1.1, let us model subsets $S \subseteq V$ by characteristic vectors $x \in \{-1, 1\}^n$ where $x_i = 1$ if $i \in S$

We note that edge $= ij \in \delta(S)$ if $x_i x_j = -1$, thus we can rewrite our objective function in (MC) :

$$\sum_{ij \in \delta(S)} a_{ij} = \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2}$$

Now, a way to model the objective function makes use of the Laplacian associated to A :

$$L_A = \text{Diag}(Ae) - A$$

where $e = (1, \dots, 1)^T$

Using L_A, we obtain a new equality for the objective function :

$$\sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2} = \frac{1}{4} x^T L_A x$$

This can be proved by developing the right-hand side of the equality.

We now use those results in ((MC)) and get :

$$\max_{x \in \{-1,1\}^n} \frac{1}{4} x^T L x$$

Now the of relaxation idea introduced in paper **Semidefinite Relaxation of Quadratic Optimization Problem**, we relax this problem :

$$\begin{aligned} &\max \frac{1}{4} x^T L x \\ &\text{s.c} : x_i^2 = 1, i = 1, ..., n \end{aligned} \tag{1}$$

And now we introduce $X = xx^T$ and change the objective to $x^T L x = tr(x^T L x) = tr(L x x^T) = tr(LX)$:

$$\begin{aligned} &\max \frac{1}{4} tr(LX) \\ &\text{s.c} : x_i^2 = 1, i = 1, ..., n \end{aligned} \tag{2}$$

Now to get better constraint now that $x_i^2 = 1$ is equal to $Diag(X) = e$:

$$\begin{aligned} &\max \frac{1}{4} tr(LX) \\ &\text{s.c} : Diag(X) = e \end{aligned} \tag{3}$$

Given that $x_i \in \{-1, 1\}$, we note that Diag$(X) = e$

But in the last problem X is not well-constrainted , because we have $X = xx^T$, X is rank one symmetric positive semidefinite (PSD) matrix , so we add this constraint too.

We can now conclude with the SDP relaxation of Max-Cut :

$$\begin{aligned} &\max \frac{1}{4} tr(LX) \\ &\text{s.c} : \text{Diag}(X) = e \\ &X \succeq 0 \end{aligned} \tag{MC SDP}$$

We can solve this problem using cvxpy.

2.2 Hyperplane rounding for Max-Cut

Now assume that we derived the SDP relaxation of Max-Cut(MC SDP) and obtained X^* . Note that we used the SDP relaxation as a modeling tool and this matrix is not necessarily give us the integers. In this part we want to generate good integer solutions for the problems using the solution X^* with Hyperplane rounding.

Now since X^* is a symmetric matrix we can write it $X^* = P^T L P$ where L is a diagonal matrix with the all positive eigenvalues of X^* and P the matrix formed by the associated eigenvectors.

$$X^* = P^T L P = P^T (L^{\frac{1}{2}})^T L^{\frac{1}{2}} P = V^T V$$

We note that as Diag$(X^*) = e$, having written X^* as $V^T V$ we deduce that the Euclidian norm of v_i for any $i \in \{1, \ldots, n\}$ is 1 meaning that v_i are on the unit sphere.

Indeed, we recall that are trying to find an optimal hyperplane, going through the origin, separating the v_i in two, one side in S and the other in $V \setminus S$ as defined in 3.1.1, one side being encoded as 1 and the other as -1

The geometric intuition behind this method is based on the idea of separating the unit vectors v_i using a random hyperplane. Suppose we choose a normal vector r that defines this hyperplane. Each vector v_i in the unit sphere forms an angle with r. If this angle is **acute** (i.e., less than $90°$), then the inner product $r^T v_i$ is **positive**, and we assign that node to one partition. Conversely, if the angle is **obtuse** (i.e., greater than $90°$), the inner product is **negative**, and the node is placed in the other partition.

Mathematically, this separation is determined by the sign of the inner product:

$$r^T v_i = \|r\|\|v_i\| \cos(\text{angle between } r \text{ and } v_i).$$

Since both r and v_i have unit norm, this simplifies to $\cos(\theta)$, where θ is the angle between the two vectors. This explains why the Hyperplane Rounding method assigns nodes based on the **sign** of $r^T v_i$. If $\cos(\theta)$ is positive, the node belongs to one side of the cut; otherwise, it belongs to the other. This simple yet powerful geometric argument ensures that the cut reflects the structure of the SDP solution while maintaining a good approximation to the optimal Max-Cut.

Now that we have the intuition in mind, the whole algorithm is given by :

Algorithm 1 Hyperplane Rounding Algorithm

Require: $X \succeq 0$ and $\text{Diag}(X) = e$ given by $X = V^T V$, where V is an $n \times n$ matrix with columns $V = (v_1, \ldots, v_n)$.

Ensure: A bisection $\xi \in \{-1, 1\}^n$.

Randomly draw a vector $r \in R^n$ uniformly from the unit sphere.

for $i = 1$ to n **do**

Set $\xi_i = 1$ if $r^T v_i \geq 0$, otherwise set $\xi_i = -1$.

end for

return ξ

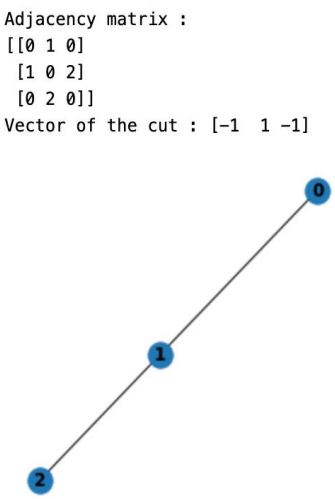

Figure 3: GW algorithm for a graphs with three vertices with two connections

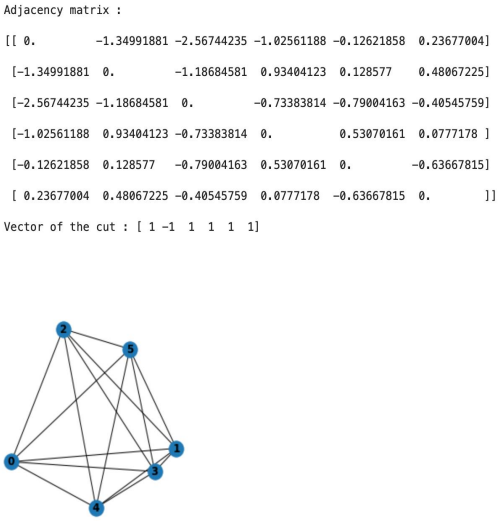

Figure 4: GW algorithm for a graph with a random adjacency matrix

3 Graph Coloring

3.1 SDP Relaxation - Chromatic Number

Let G be a graph and $(V_1, \ldots, V_k)$ be a k-partition of its vertex set $V(G)$. We recall that a vertex set V_i is stable if the vertices in V_i are pairwise nonadjacent. A k-coloring of G is such that each V_i of the k-partition is stable.

With that in mind, we define the quantity of interest, the chromatic number: $\chi(G)$ is the smallest value k such that G has a k-coloring.

Now, using the idea we introduced earlier, let us model the k-partition of the vertex set by characteristic vectors $s_i \in \{0,1\}^n$ where $(s_i)_u = 1$ if $u \in V_i$.

Following this idea, we introduce the partition matrix associated with the k-partition $(V_1, \ldots, V_k)$:

$$M = \sum_i s_i s_i^T \tag{PMat}$$

Now, a way to model the objective $\chi(G)$ is through the following result on M. Let M be a symmetric $0,1$ matrix. M is a k-partition matrix if and only if:

- $\text{Diag}(M) = e$
- $\text{rank}(M) = k$
- $M \succeq 0$

To get a sense of where this result is coming from, let us check that the partition matrix we introduced earlier indeed satisfies the rank condition. First, looking at (PMat), it is obvious that M is a symmetric $0,1$ matrix. Since $s_i \neq 0_{1\times n}$ for any V_i in the partition and that if $u \in V_i$, then $(s_{j\neq i})_u = 0$, we conclude that $\text{rank}(M) = k$.

Other arguments allow refining the previous equivalence result on M and notably eliminating the non-convex rank constraint and the semidefiniteness of M. Instead, we now have:

Let M be a symmetric $0,1$ matrix. M is a k-partition matrix if and only if:

- $\text{Diag}(M) = e$
- $(tM - J \succeq 0 \iff t \geq k)$, where J is the all-ones matrix.

Using these two results, we now write our objective as the following SDP program:

$$\chi(G) = \min\{t : tM - J \succeq 0; \text{Diag}(M) = e; m_{ij} = 0, \forall ij \in E(G); m_{ij} \in \{0,1\}\} \quad (\chi(G) \text{ SDP bin})$$

3.2 Hyperplane Rounding for Graph Coloring

As the hyperplane rounding algorithm used for Max-Cut, we will now dive into an approach used for the graph-coloring problem. The graph-coloring problem consists in assigning a color to each {vertex, edge} of a graph such that no two adjacent {vertices, edges} are of the same color.

This problem too is NP-hard, and as such, non-trivial bounds for the number of colors are often hard to get, let alone $\chi(G)$. In 1983, Wigderson focused on the particular set-up of three-colorable graphs and ended up with an algorithm coloring any three-colorable graph on n vertices with at most $3\sqrt{n}$ colors.

The choice of three-colorable graphs was not haphazard. Indeed, Wigderson used the two following facts:

- For any $v \in V(G)$ where G is such that $\chi(G) = 3$, the neighborhood $N(v)$ is a bipartite graph, therefore colorable with at most two colors .
- If G has a maximum vertex degree Δ, then G can be colored in polynomial time with at most $\Delta + 1$ colors.

We quickly recall that the neighborhood $N(v)$ of a vertex $v \in V(G)$ is the set of all vertices adjacent to v, and that the vertex degree Δ of a vertex v is the number of edges incident to v.

Those two facts in mind, Wigderson suggests using property 1 stated above and coloring the neighborhood of a vertex of G with only two colors. Furthermore, if the maximum vertex degree Δ is such that $\Delta > \sqrt{n}$, then, abiding by the constraints of the problem, one can color at least $\sqrt{n}$ vertices with only two colors. The rest is backed by the second property stated above. The whole algorithm is now the following:

Algorithm 2 Wigderson Algorithm for Three-Colorable Graphs

Require: G is a graph on n vertices such that $\chi(G) = 3$
Ensure: A coloring of G with at most $3\sqrt{n}$ colors
  while there exists a vertex $v \in G$ with degree $\Delta \geq \sqrt{n}$ **do**
    Color v with color 1
    Color $N(v)$ with at most two new colors
    Remove v and $N(v)$ from G (the new graph is still called G)
  end while
  Color the remaining graph G with at most $\Delta + 1$ colors
  return A valid coloring of G using at most $3\sqrt{n}$ colors

Output: A coloring of G with at most $3\sqrt{n}$ colors.

3.2.1 Semidefinite Programming and Lovasz Number

The Wigderson algorithm we just saw does not make use of the relaxation we introduced , but one improvement of the algorithm does so.

Starting back from the $\chi(G)$ SDP formulation, we acknowledge that the $0, 1$ condition is not tractable and introduce a new form for M that can be found in the Rendl article. From this new form emerges the famous Lovasz number $\vartheta(G)$.

The Lovasz number has a lot of different uses and relations in various graph-related NP-hard problems like finding optimal stable sets and cliques. Notably, for the graph coloring problem, $\vartheta(G)$ is a lower bound on $\chi(G)$.

$\vartheta(G)$ can be formulated as a semidefinite program and numerically approximated by the ellipsoid method in polynomial time in the number of vertices of G .

One semidefinite program for $\vartheta(G)$ is the following:

$$\min\{\lambda : V \succeq 0, v_{ii} = 1 \forall i, v_{ij} = \lambda \forall ij \in E(G)\} \qquad (\vartheta(G) \text{ SDP})$$

3.3 Refined Algorithm Using SDP Relaxation

We now use the Lovasz number in a refined algorithm.

As with the Max-Cut problem, we can adapt the hyperplane rounding idea to obtain a solution for the graph coloring problem from a solution X of its SDP relaxation. This is what Karger, Motwani, and Sudan (KMS) proposed in 1998, adapting the idea of Goemans-Williamson.

They proposed generating t random vectors on the unit sphere. The t hyperplanes constructed from these vectors define 2^t regions. In a similar manner to the Goemans-Williamson algorithm, for a well-chosen t, if we apply identical colors to the vectors V_i that are in the same region, we obtain a semi-coloring with high probability, i.e., there exists a subgraph of size $(n/2)$ where there are no monochromatic edges.

The KMS algorithm combines this method with Wigdersonâs while-loop to achieve a coloring with $O(n^{0.387})$ colors.

Algorithm 3 KMS Algorithm for Three-Colorable Graphs

Require: G is a graph on n vertices such that $\chi(G) = 3$
Ensure: A coloring of G with at most $O(n^{0.387})$ colors
  while there exists a vertex $v \in G$ with degree $\Delta \geq n^{0.613}$ **do**
    Color v with color 1
    Color $N(v)$ with at most two new colors
    Remove v and $N(v)$ from G (the new graph is still called G)
  end while
  while there exist non-colored vertices **do**
    Compute the solution of the SDP relaxation of the problem
    Compute a semi-coloring from this solution with the hyperplane rounding method
    Remove the colored vertices from G
  end while
  return A valid coloring of G using at most $O(n^{0.387})$ colors

```
End of the Widgerson loop ; current coloring :
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
New semicoloring added via hyperplane rounding ; current coloring :
[1. 3. 2. 0. 4. 0. 2. 0. 4. 3.]
New semicoloring added via hyperplane rounding ; current coloring :
[1. 3. 2. 5. 4. 0. 2. 5. 4. 3.]
New semicoloring added via hyperplane rounding ; current coloring :
[1. 3. 2. 5. 4. 6. 2. 5. 4. 3.]
coloring : [1. 3. 2. 5. 4. 6. 2. 5. 4. 3.]
The KMS algo is here correct.
```

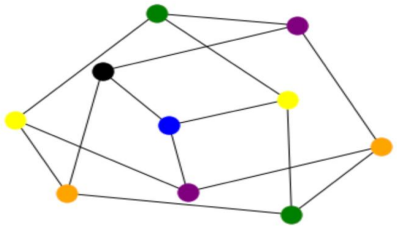

Figure 5: KMS algorithm on the Petersen graph that has 10 vertices and 15 edges and that is known to be 3-colorable

4 References

- Asratian, Armen S.; Denley, Tristan M. J.; HÃ¤ggkvist, Roland (1998), *Bipartite Graphs and their Applications*, Cambridge Tracts in Mathematics, 131, Cambridge University Press, ISBN 9780521593458.
- [link]Dunja Pucher, Franz Rendl; Practical Experience with Stable Set and Coloring Relaxations