# **The Structure of Optimal Solutions in the Motzkin-Straus Program: From Characteristic Vectors to Spurious Solutions**

## **Introduction**

The maximum clique problem (MCP) stands as a cornerstone of combinatorial optimization and computational complexity theory. Given a simple, undirected graph, the problem seeks to find the largest subset of vertices wherein every two distinct vertices are adjacent.1 Since its inclusion in Karp's seminal list of 21 NP-complete problems, the MCP has been the subject of intense theoretical and algorithmic investigation, owing to its wide-ranging applications in fields as diverse as social network analysis, bioinformatics, scheduling, and information retrieval.1 The inherent computational difficulty of the MCP has motivated the development of a vast array of solution methodologies, from exact branch-and-bound algorithms to sophisticated heuristics.

A particularly elegant and fruitful line of inquiry was opened in 1965 by T. S. Motzkin and E. G. Straus, who established a profound connection between the discrete, combinatorial nature of the MCP and the continuous world of quadratic optimization.2 They formulated the problem as a quadratic program (QP) over the standard simplex:

x∈Δn​max​fG​(x)=xTAx

where A is the adjacency matrix of the graph G on n vertices, and Δn​={x∈Rn∣∑i=1n​xi​=1,xi​≥0} is the standard simplex in Rn. The cornerstone result of their work is that the global optimal value of this program is directly related to the clique number of the graph, ω(G), by the simple formula:

x∈Δn​max​xTAx=1−ω(G)1​

This theorem provides a remarkable bridge between a discrete graph invariant and the solution to a continuous optimization problem, a result that has since been leveraged to derive new proofs of classical theorems in extremal graph theory, such as Turán's theorem, and to develop novel algorithmic approaches for the MCP.2

While the relationship between the optimal *value* of the Motzkin-Straus program and the clique number is well-established, the nature of the optimal *vector* x∗ that achieves this value is far more intricate and is the central subject of this report. The fundamental question is: what information does the optimal solution vector x∗ provide about the vertices of the maximum clique(s)? A naive hope would be a simple, direct correspondence, where the non-zero components of x∗ uniquely identify the vertices of a maximum clique. However, the reality is substantially more complex. The set of global optimizers can be a high-dimensional convex set, and the program can exhibit so-called "spurious" solutions—local or even global optima whose support does not correspond to a clique at all.11 These phenomena present significant challenges for algorithms that rely on the Motzkin-Straus formulation to find maximum cliques.

This report will provide an exhaustive analysis of the structure of optimal solutions to the Motzkin-Straus program. We will begin by examining the ideal case, where a direct correspondence between an optimizer and a maximum clique can be established through the logic of the original proof. We will then delve into the complex geometry of the full solution set, exploring the impact of multiple maximum cliques and the nature of spurious solutions. A deeper analysis through the lens of Karush-Kuhn-Tucker (KKT) optimality conditions will reveal the underlying algebraic reasons for these complexities, linking them to structural properties of the graph, such as regularity. Subsequently, we will present regularization as a powerful and definitive remedy to eliminate spurious solutions, thereby restoring a clean, one-to-one correspondence between optimizers and cliques. Finally, the report will explore broader horizons, including generalizations of the theorem to weighted graphs and hypergraphs, its profound connection to the theory of copositive programming, and its practical implications for designing effective clique-finding algorithms and its applications in fields such as computer vision. This comprehensive journey will deconstruct the intricate relationship between the continuous optimizers of the Motzkin-Straus program and the discrete clique structures they represent, providing a nuanced and complete understanding of the information held within the optimal solution vector x∗.

## **Section 1: The Ideal Case: A Direct Correspondence**

The foundational 1965 paper by Motzkin and Straus, "Maxima for Graphs and a New Proof of a Theorem of Turán," was originally motivated by a question concerning the maximum of a square-free quadratic form on a simplex.14 In addressing this question, they uncovered a deep structural property of the quadratic program's optimizers that directly links them to the clique structure of the underlying graph. This section dissects this foundational result, focusing on the idealized scenario where the optimal solution vector cleanly identifies a maximum clique.

### **1.1 The Foundational Result of Motzkin and Straus (1965)**

The theorem establishes a precise quantitative link between the maximum clique problem and a continuous quadratic program. Let G=(V,E) be a simple undirected graph with n vertices, and let A be its adjacency matrix. The Motzkin-Straus theorem considers the optimization problem of maximizing the quadratic form fG​(x)=xTAx=∑(i,j)∈E​2xi​xj​ over the standard simplex Δn​.

Theorem (Motzkin-Straus, 1965): Let G be a graph with clique number ω(G)=k. The global maximum of the quadratic program

x∈Δn​max​fG​(x)=xTAx

is given by 1−k1​.6

The true power of this theorem, especially concerning the structure of the optimal vector x∗, is revealed not just in the statement itself, but in the logic of its proof.

### **1.2 The Proof and the Clique Support Property**

The classical proof of the Motzkin-Straus theorem is a masterclass in optimization arguments, employing a proof by contradiction that hinges on the concept of a minimality condition on the optimizer's support. This argument is central to understanding how and when an optimal vector x∗ can be expected to identify the vertices of a clique.9

The proof proceeds as follows:

1. **Existence and Selection of the Optimizer:** The standard simplex Δn​ is a compact (closed and bounded) set in Rn. The objective function fG​(x) is continuous. By the Extreme Value Theorem, a global maximum must exist. Let M be the set of all global maximizers of fG​(x) on Δn​. From this set, we select a vector y∈M that has the **smallest possible support**, where the support is defined as the set of indices corresponding to its non-zero components: supp(y)={i∈V∣yi​>0}.7
2. **The Central Claim:** The core of the proof is to show that the subgraph induced by the support of this minimally-supported optimizer, G[supp(y)], must be a clique.
3. **Proof by Contradiction:** We assume the contrary: that the support of y, let's call it K=supp(y), is not a clique. This implies there must exist at least one pair of vertices, say i,j∈K, that are not adjacent, i.e., (i,j)∈/E. Since i,j∈K, we have yi​>0 and yj​>0.
4. **The "Mass Transportation" Argument:** The key insight is to construct a new vector y′ by transferring the "mass" (or weight) from one of the non-adjacent vertices to the other, without decreasing the objective function's value. This process is a continuous analogue of Zykov's symmetrization, a technique from extremal graph theory used to simplify graph structures while preserving or increasing certain properties.9 Let's compare the "local connectivity" of vertices  
   i and j with respect to the current solution y. We define the weighted degree of a vertex v as (Ay)v​=∑k∼v​yk​. Without loss of generality, let's assume vertex i is at least as "well-connected" as vertex j, i.e., (Ay)i​≥(Ay)j​.  
   We now define a new vector y′∈Δn​ by moving the entire weight of yj​ onto yi​:  
   $$ y'\_k = \begin{cases} y\_i + y\_j & \text{if } k = i \ 0 & \text{if } k = j \ y\_k & \text{otherwise} \end{cases} $$  
   This new vector y′ is still in the simplex since all its components are non-negative and their sum remains 1.
5. Analyzing the Change in Objective Value: Let's examine the difference fG​(y′)−fG​(y). The objective function is fG​(x)=∑(u,v)∈E​2xu​xv​. The change is due to the modification of components i and j:  
   $$ f\_G(y') - f\_G(y) = 2(y\_i + y\_j) \sum\_{k \sim i, k \neq j} y\_k - (2y\_i \sum\_{k \sim i, k \neq j} y\_k + 2y\_j \sum\_{k \sim j, k \neq i} y\_k) $$  
   Note that since (i,j)∈/E, there is no 2yi​yj​ term in fG​(y). The expression simplifies to:  
   $$ f\_G(y') - f\_G(y) = 2y\_j \left( \sum\_{k \sim i} y\_k - \sum\_{k \sim j} y\_k \right) = 2y\_j \left( (Ay)\_i - (Ay)\_j \right) $$  
   Since we chose yj​>0 and assumed (Ay)i​≥(Ay)j​, we have fG​(y′)−fG​(y)≥0.
6. **The Contradiction:** Because y is a global maximizer, we must have fG​(y′)≤fG​(y). Combined with the above, this implies fG​(y′)=fG​(y), meaning y′ is also a global maximizer. However, the support of y′ is supp(y′)=K∖{j}, which is strictly smaller than the support of y. This contradicts our initial choice of y as the global maximizer with the *minimal possible support*.
7. **Conclusion:** The initial assumption that supp(y) is not a clique must be false. Therefore, the support of any global maximizer chosen with minimal support must induce a clique in G.9

### **1.3 Characteristic Vectors as Global Optimizers**

The proof establishes that there exists at least one global maximizer y∗ whose support K=supp(y∗) is a clique. Let the size of this clique be ∣K∣=k′. When we restrict the optimization problem to the vertices in K, the problem becomes:

$$ \max\_{x\_K \in \Delta\_{k'}} \sum\_{(i,j) \in E(G[K])} 2x\_i x\_j = \max\_{x\_K \in \Delta\_{k'}} \sum\_{i,j \in K, i \neq j} 2x\_i x\_j $$

since G[K] is a complete graph (a clique). This is equivalent to maximizing the function for a complete graph Kk′​. It is a standard result that this maximum is achieved when the weights are distributed uniformly across all vertices in the clique.7

This unique optimizer on the sub-problem is the characteristic vector of the clique K, denoted xK​, defined as:

(xK​)i​={1/k′0​if i∈Kotherwise​

For this vector, the objective function evaluates to fG​(xK​)=(2k′​)⋅2⋅(k′1​)2=k′(k′−1)⋅(k′)21​=1−k′1​.

Since the global maximum value of the program over all of Δn​ is 1−1/ω(G), it must be that the value achieved by our minimal-support optimizer y∗ is 1−1/k′=1−1/ω(G). This implies that k′=ω(G). Therefore, the clique identified by the support of the minimal-support optimizer is not just any clique, but a **maximum clique**.17

In this idealized scenario, the optimal solution vector x∗ provides unambiguous information. For any maximum clique C in the graph, its characteristic vector xC​ is a global optimizer. The indices of the non-zero components of xC​ directly identify the vertices of the maximum clique C, and the values of these components are uniform, equal to the reciprocal of the clique number, 1/ω(G). This clean, direct correspondence forms the theoretical bedrock for many algorithms and applications based on the Motzkin-Straus theorem. However, the crucial qualifier "minimal support" is the linchpin of this entire argument. Relaxing this assumption, or considering graphs with more complex clique structures, reveals a far more intricate solution landscape, as the following sections will explore.

## **Section 2: The Complex Reality: The Full Geometry of the Optimizer Set**

The elegant correspondence between a minimal-support optimizer and a single maximum clique, as detailed in Section 1, represents an idealized view. The reality of the solution set for the Motzkin-Straus quadratic program is significantly more complex. When a graph possesses multiple maximum cliques, or when one considers local optima, the set of solutions expands beyond simple characteristic vectors. This section delves into the true geometry of the optimizer set, exploring the phenomena of convex combinations of solutions and the emergence of "spurious" optima that challenge the direct interpretation of the solution vector.

### **2.1 The Challenge of Multiple Maximum Cliques**

If a graph G has more than one maximum clique, say C1​,C2​,…,Cq​, all of size k=ω(G), then the argument from Section 1 applies to each of them. The characteristic vector of each maximum clique, xC1​​,xC2​​,…,xCq​​, is a global optimizer of the Motzkin-Straus program, and each achieves the maximum objective value of 1−1/k.19 This immediately tells us that the global optimizer is not necessarily unique. The set of global optimizers contains, at a minimum, the characteristic vectors of all maximum cliques.

### **2.2 The Convex Hull of Characteristic Vectors**

A far more profound complication arises from the quadratic nature of the objective function fG​(x)=xTAx. Since the Hessian matrix, 2A, is not strictly negative definite, the objective function is not strictly concave over the feasible set Δn​. This lack of strict concavity opens the door for the existence of "plateaus" of optimal solutions, where an entire continuum of points, not just a discrete set, achieves the global maximum.

Research has shown that under specific structural conditions, the set of global optimizers is not merely the discrete collection of characteristic vectors but their entire **convex hull**.20 Let's consider two distinct maximum cliques,

C and D, both of size k. Let xC​ and xD​ be their respective characteristic vectors. Now consider a convex combination of these two vectors:

$$x(\lambda) = \lambda x\_C + (1-\lambda) x\_D \quad \text{for } \lambda \in $$

The objective value at this point is:

$$ f\_G(x(\lambda)) = x(\lambda)^T A x(\lambda) = \lambda^2 f\_G(x\_C) + (1-\lambda)^2 f\_G(x\_D) + 2\lambda(1-\lambda) x\_C^T A x\_D $$

Since fG​(xC​)=fG​(xD​)=1−1/k, the value of fG​(x(λ)) depends on the cross term xCT​AxD​. It has been proven that fG​(x(λ)) is constant and equal to 1−1/k for all $\lambda \in $ if and only if the number of edges between the disjoint vertex sets C∖D and D∖C is maximal.21 Let

m=∣C∖D∣=∣D∖C∣. This maximal number of edges is m(m−1).

When this condition holds, the entire line segment connecting xC​ and xD​ consists of global optimizers. A point on this segment, for λ∈(0,1), is not a characteristic vector. Its support is the union of the two cliques, supp(x(λ))=C∪D. The values of its components are non-uniform; for instance, a vertex i∈C∩D would have a value of (λ/k)+((1−λ)/k)=1/k, whereas a vertex j∈C∖D would have a value of λ/k.

This result generalizes to more than two maximum cliques. If a set of maximum cliques {C1​,…,Cq​} satisfies the requisite pairwise edge conditions, their entire convex hull, conv(xC1​​,…,xCq​​), forms a set of global optimizers.20 These solutions, whose supports are unions of cliques and whose components are non-uniform, are the first type of deviation from the ideal one-to-one correspondence.

### **2.3 "Spurious" Solutions: Optima Without Clique Support**

Perhaps the most significant challenge to interpreting the optimal vector x∗ is the existence of what are commonly termed **spurious solutions**. These are local, and sometimes even global, optima of the Motzkin-Straus program whose support does not form a clique.6

The existence of such solutions is a major practical drawback for algorithms attempting to use the Motzkin-Straus formulation to find cliques. If an optimization heuristic converges to a spurious solution, it is not straightforward how to recover a valid clique from the resulting vector x∗.12 The support of the vector does not represent a feasible solution to the original combinatorial problem.

A canonical example is the cherry graph, which consists of three vertices {1,2,3} and two edges {(1,3),(2,3)}. The maximum clique size is ω(G)=2, corresponding to the cliques {1,3} and {2,3}. The maximum value of the objective function is 1−1/2=1/2. This value is achieved by the characteristic vectors x{1,3}​=(1/2,0,1/2) and x{2,3}​=(0,1/2,1/2). However, the vector

x~=(1/4,1/4,1/2)

is also a local maximizer (and a KKT point) of the program.11 Its objective value is

fG​(x~)=2(41​21​)+2(41​21​)=1/4+1/4=1/2, which is the global maximum. The support of this vector is {1,2,3}, which is not a clique. This x~ is a classic example of a global spurious solution. It arises because the optimization process finds a balance by placing most of the weight on the central, high-degree vertex (3) and distributing the remaining weight equally among the non-adjacent leaf vertices (1 and 2). The quadratic program is optimizing a continuous measure of "weighted density," and while cliques represent an ideal form of density, other non-clique weight distributions can also achieve high scores.

### **2.4 Local Maxima and Maximal Cliques**

The theory connecting optima to cliques extends, with added complexity, to the local level. While global optima are related to *maximum* cliques, local optima are related to *maximal* cliques. A clique is maximal if it cannot be extended by adding any other vertex from the graph.

It has been shown that the characteristic vector xC​ of a maximal clique C satisfies the first-order necessary conditions for optimality (the KKT conditions, discussed in the next section).19 This suggests that maximal cliques are candidates for local maxima.

A sharper result establishes a correspondence between *strict local maximizers* and *strictly maximal cliques*.19 A maximal clique

C is called strictly maximal if it is not possible to swap a vertex in the clique for a vertex outside the clique to form a new clique of the same size. Formally, for any i∈C and j∈/C, the set (C∖{i})∪{j} is not a clique. If a clique is strictly maximal, its characteristic vector is a strict local maximizer of the Motzkin-Straus program.

However, this correspondence is not perfect. A maximal clique that is *not* strictly maximal may have a characteristic vector that is not a local maximizer of the program.19 This introduces yet another layer of subtlety, where the specific topological properties of how a maximal clique is embedded in the larger graph determine whether its characteristic vector is a stable point in the continuous optimization landscape.

In summary, the true geometry of the optimizer set is far from the simple picture of isolated points corresponding to maximum cliques. It can be a convex, continuous set, and it can include spurious points whose supports are not cliques at all. These complexities reveal a fundamental limitation of the original formulation: it captures a more general notion of a "cohesive" or "dense" weighted subgraph, for which cliques are the discrete ideal but not the only possible outcome. To understand the algebraic origins of this behavior, we must turn to a deeper analysis of the program's optimality conditions.

## **Section 3: A Deeper Analysis Through First-Order Optimality Conditions**

To fully grasp why the Motzkin-Straus program admits such a complex solution space, including spurious optima and convex sets of maximizers, we must move beyond the global proof argument and analyze the problem through the lens of local optimality conditions. The Karush-Kuhn-Tucker (KKT) conditions, which are necessary for a point to be a local optimum in a constrained nonlinear program, provide a precise algebraic characterization of the solution vectors. This analysis reveals that the program's optima are intimately tied to a structural property of graphs—regularity—which is more general than cliqueness, thereby explaining the emergence of non-clique solutions.

### **3.1 The Karush-Kuhn-Tucker (KKT) Conditions for the Simplex-Constrained QP**

The Motzkin-Straus problem is a quadratic program with one equality constraint (∑xi​=1) and n non-negativity constraints (xi​≥0). For a feasible point x∗ to be a local maximizer, it must satisfy the KKT conditions.5 These state that there must exist a Lagrange multiplier

λ∈R for the equality constraint and non-negative multipliers μi​≥0 for each inequality constraint xi​≥0, such that the following conditions hold:

1. Stationarity: The gradient of the Lagrangian function with respect to x must be zero. The Lagrangian is L(x,λ,μ)=xTAx−λ(∑xi​−1)−∑μi​(−xi​). The gradient condition is:  
   ∇x​L=2Ax∗−λ1+μ=0  
     
   where 1 is the vector of all ones and μ=(μ1​,…,μn​)T. This gives a system of equations: 2(Ax∗)i​−λ+μi​=0 for each i∈V.
2. **Primal Feasibility:** x∗ must be in the simplex: ∑xi∗​=1 and xi∗​≥0 for all i.
3. **Dual Feasibility:** μi​≥0 for all i.
4. **Complementary Slackness:** For each i, μi​xi∗​=0.

### **3.2 The Equi-Payoff Property**

The complementary slackness condition is particularly revealing. It dictates that if a component xi∗​ is strictly positive (i.e., vertex i is in the support of the solution), then its corresponding dual multiplier μi​ must be zero. Applying this to the stationarity condition, we arrive at a crucial structural property of any KKT point x∗:

* For any vertex i∈supp(x∗) (where xi∗​>0), we have μi​=0, which implies 2(Ax∗)i​=λ.
* For any vertex j∈/supp(x∗) (where xj∗​=0), we have μj​≥0, which implies 2(Ax∗)j​≤λ.

This establishes the **equi-payoff property**: for any KKT point x∗, the "payoff" or "weighted degree" of each vertex in its support, given by (Ax∗)i​=∑j∼i​xj∗​, must be constant and equal to λ/2. For vertices outside the support, this weighted degree must be less than or equal to this same constant value.5 This property is the algebraic foundation for the structure of all optima, both "good" and "spurious."

### **3.3 The Induced Regular Subgraph Property**

The link between the KKT conditions and graph structure becomes explicit when we consider KKT points that are **characteristic vectors**. Let xS​ be the characteristic vector of a vertex set S⊆V, where (xS​)i​=1/∣S∣ if i∈S and 0 otherwise. For xS​ to be a KKT point, the equi-payoff property must hold for all vertices i∈S.

Let's compute the weighted degree for a vertex i∈S:

$$ (A x\_S)i = \sum{j \sim i} (x\_S)j = \sum{j \in S, j \sim i} \frac{1}{|S|} = \frac{\text{deg}{G}(i)}{|S|} $$

where $\text{deg}{G}(i)$ is the degree of vertex i within the subgraph induced by S, G.

The equi-payoff condition, (AxS​)i​=λ/2 for all i∈S, therefore implies:

∣S∣degG​(i)​=constant∀i∈S

This means that degG​(i) must be the same for all vertices i∈S. A graph in which every vertex has the same degree is known as a regular graph.

Thus, we have a powerful result: **A characteristic vector xS​ is a KKT point of the Motzkin-Straus program if and only if the induced subgraph G is regular**.5

### **3.4 Re-interpreting Spurious Solutions and Maximality**

This KKT analysis provides a clear and compelling explanation for the existence of a class of spurious solutions.

* A k-clique is a (k−1)-regular graph. Therefore, its characteristic vector satisfies the KKT conditions and is a candidate for a local optimum.
* However, any other induced subgraph G that is regular but *not* a clique also has a characteristic vector xS​ that satisfies the KKT conditions. For example, a cycle graph is 2-regular. If a graph G contains an induced cycle Ck​, its characteristic vector xCk​​ is a KKT point and thus a candidate for a spurious local optimum.

This reveals a fundamental mismatch: the global argument of the Motzkin-Straus proof identifies maximum cliques, but the local, first-order necessary conditions only enforce the much weaker structural property of regularity (for characteristic vectors). The optimization landscape has stationary points corresponding to all regular induced subgraphs, not just cliques.

Furthermore, the KKT conditions also elegantly encode the notion of maximality. For a KKT point x∗=xC​ where C is a clique, the second part of the condition is 2(AxC​)j​≤λ for all j∈/C. This translates to:

2∣C∣∣N(j)∩C∣​≤2∣C∣∣C∣−1​⟹∣N(j)∩C∣≤∣C∣−1

This inequality is always true. The condition for C to be a maximal clique is that for any vertex j∈/C, it is not adjacent to at least one vertex in C. This corresponds to the strict inequality ∣N(j)∩C∣<∣C∣−1, which in turn corresponds to a strict local optimum where the gradient conditions are strictly satisfied for vertices outside the support. Thus, the KKT framework perfectly captures the combinatorial definition of maximality.

The analysis of KKT points illuminates the deep algebraic reasons behind the complex geometry of the solution set. The program is, in a sense, "blind" to the difference between a clique and other regular structures when viewed through the lens of first-order optimality. This inherent limitation of the original formulation necessitates a modification to the problem itself if one wishes to guarantee that the optimizers have the desired clique structure. This leads directly to the concept of regularization.

## **Section 4: Restoring Order: Regularization and the Elimination of Spurious Solutions**

The analysis in the preceding sections has made it clear that the original Motzkin-Straus quadratic program, while mathematically elegant, is algorithmically problematic. The existence of spurious solutions, non-clique local optima, and entire convex sets of global maximizers means that a standard optimization heuristic applied to the problem is not guaranteed to return a vector from which a maximum clique can be easily identified. The root of this issue lies in the objective function fG​(x)=xTAx not being strictly concave on the simplex, which allows for this complex and undesirable solution geometry.12

The definitive solution to this problem is **regularization**. By adding a suitable strictly convex term to the objective function, one can reshape the optimization landscape to eliminate these unwanted optima and enforce a clean, one-to-one correspondence between the program's solutions and the graph's clique structures.

### **4.1 The Rationale for Regularization**

The core idea behind regularization is to perturb the original objective function to enforce desirable properties on its optima. In this context, the goal is to eliminate the "flat" regions and non-clique "bumps" in the optimization landscape that correspond to spurious solutions and convex hulls of optima. This is achieved by adding a strictly convex function Φ(x) to the objective, creating a new regularized program 11:

x∈Δn​max​xTAx+Φ(x)

The addition of a strictly convex term makes the new objective function strictly concave (or at least "more" concave). A strictly concave function optimized over a convex set (like the simplex) has at most one global maximum and all local maxima are strict and isolated. The key is to choose Φ(x) carefully so that it breaks the undesirable symmetries and degeneracies of the original problem while preserving the fundamental connection to cliques.

### **4.2 The Canonical Regularization (Bomze, 1997)**

The most widely studied and canonical form of regularization for the Motzkin-Straus program was introduced by I. M. Bomze in 1997.12 It involves adding a simple quadratic term:

Φ(x)=2c​∥x∥22​=2c​i=1∑n​xi2​

where c is a positive regularization parameter. The regularized objective function becomes:

freg​(x)=xTAx+2c​xTx=xT(A+2c​I)x

where I is the identity matrix. From a graph-theoretic perspective, this is equivalent to adding a weighted self-loop of weight c/2 to each vertex in the graph. The term 2c​∑xi2​ is strictly convex, and its addition fundamentally alters the optimization landscape. It penalizes solutions where the weight is spread out over many components and favors solutions where the weight is concentrated on fewer components, as the sum of squares is minimized for a fixed sum when the components are more uniform. This implicitly encourages sparser solutions, aligning with the goal of finding a support that is a small, dense clique.

### **4.3 Achieving a One-to-One Correspondence**

With an appropriate choice of the parameter c>0 (a common choice is c=1), this regularization achieves a remarkable result: it completely resolves the issues of spurious solutions and non-uniqueness.12 The regularized program exhibits a perfect correspondence between its optima and the cliques of the graph:

* **Local Optima and Maximal Cliques:** There is a **one-to-one correspondence** between the local maximizers of the regularized program and the maximal cliques of the graph G. Every local maximizer is the characteristic vector of a maximal clique, and the characteristic vector of every maximal clique is a local maximizer.11
* **Global Optima and Maximum Cliques:** The global maximizers of the regularized program correspond precisely and uniquely to the characteristic vectors of the **maximum** cliques of G.12
* **Elimination of Spurious Solutions:** Crucially, all local and global maximizers of the regularized program are **strict** and are **characteristic vectors**.5 This means that spurious solutions (both those that are not characteristic vectors and those whose supports are regular but not cliques) are no longer optima. The convex hulls of solutions also disappear, as the strict concavity ensures that all optima are isolated points.

This regularization technique transforms the Motzkin-Straus program from a theoretically interesting but algorithmically flawed formulation into a robust tool. By solving the regularized QP, one can be certain that any local optimum found by a heuristic corresponds to a valid maximal clique, and the global optimum corresponds to a maximum clique. The vertices of the clique are simply given by the support of the resulting optimal vector x∗.

### **4.4 Survey of Advanced Regularization Functions**

While the quadratic L2​-norm regularization is canonical, it is not the only possibility. A broader class of functions Φ(x) can be used to achieve the same goal, provided they satisfy a set of general conditions.19 These conditions typically require

Φ(x) to be, for all x∈Δn​:

1. Twice continuously differentiable.
2. Sufficiently convex (e.g., ∇2Φ(x)≻0).
3. Bounded in a certain sense (e.g., ∥∇2Φ(x)∥2​<2).
4. Permutation-invariant (i.e., the function value does not change if the components of x are permuted).

Researchers have explored other regularization functions that satisfy these criteria and offer different theoretical or algorithmic advantages 19:

* **p-norm Regularizers:** A generalization of the quadratic term, of the form Φ1​(x)=α1​∑i=1n​(xi​+ϵ)p for p>2. This provides a family of regularizers whose "strength" can be tuned by the parameter p.
* **Exponential Regularizers:** A function of the form Φ2​(x)=α2​∑i=1n​(e−βxi​−1). This term is particularly interesting as it is motivated by encouraging **sparsity** in the solution vector x. Maximizing this regularized objective is closely related to finding a solution with the smallest possible support. This provides a direct link back to the minimal-support argument in the original Motzkin-Straus proof, showing a beautiful consistency between the combinatorial proof technique and the algebraic fix provided by regularization.

In conclusion, regularization is the key to unlocking the full algorithmic potential of the Motzkin-Straus formulation. It reshapes the optimization landscape in a principled way, using a strictly convex "penalty" to break the degeneracies that lead to spurious solutions. This process purifies the solution set, ensuring that the continuous optima align perfectly with the desired discrete combinatorial structures—maximal and maximum cliques—and making the information contained in the optimal vector x∗ direct and unambiguous.

## **Section 5: Broader Horizons: Generalizations and Connections**

The Motzkin-Straus theorem is not an isolated result but rather the foundation of a rich theoretical framework. Its principles have been extended to more general graph settings, such as those with weighted vertices, and to higher-order structures like hypergraphs. Furthermore, the theorem serves as a crucial link between combinatorial optimization and the modern field of conic optimization, specifically through the lens of copositive programming. These generalizations and connections reveal the depth and versatility of the original insight.

### **5.1 Generalization 1: The Maximum Weight Clique Problem**

A natural extension of the maximum clique problem is the **maximum weight clique problem**, where each vertex i∈V is assigned a positive weight wi​, and the goal is to find a clique whose sum of vertex weights is maximized. This value is denoted ω(G,w).4

The Motzkin-Straus theorem can be generalized to this setting. While several formulations exist, a particularly direct generalization modifies the quadratic program to incorporate the weights.4 Let

W be the diagonal matrix with Wii​=wi​. One such formulation is to maximize f(x)=xT(WAW)x over the standard simplex. A different generalization, which preserves the original form more closely, modifies the objective and the constraint space. For instance, consider the program:

$$ \max \sum\_{(i,j) \in E} w\_i w\_j x\_i x\_j \quad \text{subject to} \quad \sum\_{i=1}^n w\_i x\_i = 1, \quad x \geq 0 $$

For such formulations, the optimal value is directly related to the maximum clique weight. A theorem by Busygin shows that for a related program, the optimal value is 1−wmin​/ω(G,w), where wmin​ is the minimum vertex weight in the graph.4

Most importantly for the present discussion, the structure of the optimal solution vector x∗ is also modified in a predictable way. For a maximum weight clique Q∗, the corresponding global optimizer x∗ is no longer a uniform characteristic vector. Instead, its components are proportional to the vertex weights within the clique:

xi∗​={wi​/ω(G,w)0​if i∈Q∗otherwise​

This result shows that the core principle holds: the support of the optimizer identifies the vertices of the optimal structure (the maximum weight clique), while the values of the non-zero components now encode additional information—the relative importance (weights) of the vertices within that structure. Issues of spurious solutions can still arise, necessitating regularization techniques analogous to the unweighted case.28

### **5.2 Generalization 2: The Motzkin-Straus Theorem for Hypergraphs**

Extending the theorem to k-uniform hypergraphs (or k-graphs), where edges connect k vertices at a time, is significantly more challenging. A direct generalization, where one maximizes the hypergraph's Lagrangian polynomial over the simplex, is **false**.29 The global maximum of the Lagrangian for a hypergraph is not necessarily achieved on a vector whose support is a clique.

The failure of a direct extension highlights the special role of pairwise interactions in the original theorem's proof. The "mass transportation" argument relies on a simple, linear comparison of the local environments of two non-adjacent vertices. In a hypergraph, shifting weight between two vertices affects hyperedge terms in a complex, multilinear fashion, breaking the simple logic of the proof.

A valid generalization can, however, be achieved by fundamentally reformulating the optimization problem.17 A successful approach involves:

1. **Switching to the Complement:** Instead of the graph's Lagrangian, one considers the Lagrangian of the *complementary* hypergraph, LGˉ​(x).
2. **Adding Regularization:** A higher-order regularization term, such as τ∑xik​, is added.
3. Minimization instead of Maximization: The problem is framed as minimizing the new objective function:  
   x∈Δn​min​hG​(x)=LGˉ​(x)+τi=1∑n​xik​  
     
   For an appropriate range of the parameter τ (e.g., 0<τ≤k(k−1)1​), this modified formulation restores the desired properties. A one-to-one correspondence is established between the local/global minimizers of hG​(x) and the characteristic vectors of maximal/maximum cliques of the hypergraph G.17 This demonstrates that while the original theorem's structure does not directly carry over, its spirit can be preserved through a more sophisticated problem construction that anticipates and resolves the issues of higher-order interactions.

### **5.3 A Deeper Connection: Copositive Programming**

The Motzkin-Straus theorem provides a crucial bridge to the powerful modern framework of **copositive programming**. A symmetric matrix M is defined as **copositive** if the quadratic form xTMx≥0 for all non-negative vectors x≥0. The set of all such matrices forms a convex cone, COPn​. Deciding if a matrix is *not* copositive is an NP-complete problem, highlighting the complexity of this cone.16

The stability number of a graph, α(G) (which is the clique number of the complement graph, ω(Gˉ)), can be formulated exactly as a linear program over the copositive cone. The seminal formulation by de Klerk and Pasechnik is 26:

α(G)=min{t∈R∣t(AG​+I)−J∈COPn​}

where AG​ is the adjacency matrix of G and J is the all-ones matrix. This formulation recasts the discrete combinatorial problem of finding the largest independent set (or maximum clique in the complement) into a continuous conic optimization problem: finding the smallest scalar t that makes a certain matrix pencil, t(AG​+I)−J, enter the cone of copositive matrices.

This profound result is a direct consequence of the Motzkin-Straus theorem for the stability number, which states:

α(G)1​=x∈Δn​min​xT(AG​+I)x

This equality implies that for any x∈Δn​, xT(AG​+I)x≥α(G)1​. Multiplying by (∑xi​)2=1 and rearranging gives xT(α(G)(AG​+I))x≥(∑xi​)2=xTJx, which means xT(α(G)(AG​+I)−J)x≥0 for all x∈Δn​. This can be extended to all x≥0, proving that the matrix α(G)(AG​+I)−J is indeed copositive.31

This connection elevates the Motzkin-Straus theorem from a specific result about a quadratic program to a fundamental principle within conic optimization. It provides an exact, though computationally intractable, continuous formulation for the maximum clique problem, placing it within the powerful duality theory of convex cones and inspiring hierarchies of tractable relaxations (e.g., using semidefinite programming) to approximate the clique number.34

## **Section 6: From Theory to Algorithm: Practical Implications**

The rich and complex theory surrounding the optimal solutions of the Motzkin-Straus program has significant practical consequences for anyone wishing to use it to solve the maximum clique problem. The choice of formulation—original, regularized, or weighted—directly impacts how the resulting optimal vector x∗ should be interpreted and used. This section synthesizes the preceding analysis into practical guidance, discusses common algorithmic approaches based on the theorem, and explores its application in other domains.

### **6.1 A Synthesis: How to Interpret and Use the Optimal Vector x∗**

The central question of this report is what information the optimal vector x∗ contains. The answer depends critically on which version of the Motzkin-Straus program is being solved.

* **If using the original Motzkin-Straus QP (max xᵀAx):** One must be cautious. A globally optimal solution x∗ is not guaranteed to have a support that forms a clique. As shown in Section 2, it could be a spurious solution or a convex combination of characteristic vectors of multiple maximum cliques. If a numerical solver returns an optimizer x∗, a post-processing step is required. A simple heuristic is to examine the support, S=supp(x∗), and check if the induced subgraph G is a clique. If not, one might try to extract a clique from it, for example, by iteratively removing vertices that are part of non-edge pairs within the support.19 However, this is not guaranteed to yield a maximum clique.
* **The Recommended Approach (max xᵀ(A+cI)x):** For practical clique-finding, the use of a **regularized formulation** is strongly recommended. As established in Section 4, adding a term like 2c​∥x∥22​ for c>0 eliminates all spurious solutions and ensures a one-to-one correspondence between optima and cliques.12 With this formulation, the interpretation of the optimal vector is direct and unambiguous:
  + If x∗ is a **global** optimum of the regularized program, its support, {i∣xi∗​>0}, is guaranteed to be the vertex set of a **maximum clique**.
  + If x∗ is a **local** optimum, its support is guaranteed to be the vertex set of a **maximal clique**.
  + In both cases, the optimal vector x∗ will be the characteristic vector of the corresponding clique. Therefore, the node indices of the clique are simply the indices of the non-zero components of the solution vector.

### **6.2 Heuristics Based on the Motzkin-Straus Formulation: Replicator Dynamics**

The regularized Motzkin-Straus program provides a well-behaved continuous optimization problem that can be tackled with a variety of numerical methods. One of the most natural and historically significant heuristics is based on **replicator dynamics**, a concept originating from evolutionary game theory.12

In this context, the components of the vector x are treated as proportions of different "strategies" (vertices) in a population. The fitness of each strategy is determined by its connections in the graph. The replicator dynamics equations describe how the proportions of strategies evolve over time, with fitter strategies growing at the expense of less fit ones. For the regularized program with objective freg​(x)=xT(A+cI)x, the discrete-time replicator update rule is a simple, iterative, and computationally inexpensive multiplicative update 41:

xi​(t+1)=xi​(t)x(t)T(A+cI)x(t)[(A+cI)x(t)]i​​

where x(t) is the vector at iteration t. The term [(A+cI)x(t)]i​ represents the "fitness" of strategy i, and the denominator is the average fitness of the population, serving as a normalization factor.

**Convergence Properties:** A key result, known as the "Fundamental Theorem of Natural Selection" in this context, states that the objective function freg​(x) is a strict Lyapunov function for the dynamics. This means the objective value is strictly increasing along any non-constant trajectory, guaranteeing that the dynamics will "climb" the optimization landscape and converge to a stationary point.12 Since regularization ensures that the stable stationary points are precisely the characteristic vectors of maximal cliques, the replicator dynamics provides an effective heuristic for finding large maximal cliques.11 While convergence to a

*global* optimum (a maximum clique) is not guaranteed, in practice, the basins of attraction for global optima are often large, making it a successful method.12

### **6.3 Performance Considerations and Comparisons**

Algorithms based on the Motzkin-Straus continuous formulation represent a significant class of heuristics for the maximum clique problem.8 Their performance is often compared against purely combinatorial algorithms, which typically rely on sophisticated branch-and-bound or backtracking search procedures (e.g., the well-known Cliquer algorithm).2

* Continuous methods, like those based on replicator dynamics or other gradient-ascent techniques, can be very fast per iteration. Their strength lies in quickly finding good local optima (large maximal cliques).
* Combinatorial methods are often better at exhaustively searching the solution space and are typically used in exact solvers, but their runtime can be prohibitive for large, dense graphs.
* Heuristics based on the Motzkin-Straus formulation have demonstrated strong empirical performance. For example, the QUALEX-MS algorithm, which uses a trust-region method inspired by a generalization of the Motzkin-Straus theorem, has been shown to perform exceptionally well on the standard DIMACS benchmark graphs, finding optimal or best-known solutions for a large number of instances and outperforming other continuous methods.4 Similarly, other continuous formulations have been shown to be competitive with, and sometimes superior to, other Motzkin-Straus based methods and even sophisticated combinatorial heuristics on certain graph classes.2

The choice between continuous and combinatorial approaches often depends on the specific characteristics of the graph (size, density) and the goal (finding a provably maximum clique vs. finding a large clique quickly).

### **6.4 Applications in Computer Vision and Image Segmentation**

The principles of the Motzkin-Straus theorem, particularly its generalization to weighted graphs, have found powerful applications in computer vision and pattern recognition, most notably in the area of **image segmentation** and clustering.46

The core idea is to represent an image as a weighted graph, where pixels (or pre-segmented superpixels) are the nodes. The weight of an edge between two nodes represents the similarity between them, based on features like color, texture, or spatial proximity.49 In this graph, a visually coherent segment of the image corresponds to a subgraph where nodes are highly similar to each other—a dense, highly-weighted cluster.

The concept of a **dominant set** was introduced as a generalization of a maximal clique to edge-weighted graphs. A dominant set is a subset of vertices that is both cohesive (high internal similarity) and distinct from the rest of the graph (low similarity to external vertices). It turns out that dominant sets can be characterized as the local maximizers of a generalized, regularized Motzkin-Straus program.51 Finding these dominant sets, often using replicator dynamics, has become a powerful, parameter-free method for pairwise data clustering and unsupervised image segmentation.40 The optimal vector

x∗ in this context represents a "soft" assignment of pixels to a dominant cluster, with the support of x∗ identifying the pixels belonging to the most coherent segment in the image.

This application demonstrates the versatility of the theorem's core idea: maximizing a quadratic form on a simplex is a fundamental way to identify the most cohesive substructures within a network, a principle that extends far beyond the original context of unweighted cliques.

## **Conclusion: A Unified View of the Motzkin-Straus Solution Space**

The Motzkin-Straus theorem, at first glance, presents a beautifully simple connection between the clique number of a graph and the optimal value of a quadratic program. However, as this report has detailed, the structure of the optimal solution vector x∗ that achieves this value is a subject of considerable complexity and nuance. The journey from the idealized case of a single characteristic vector to the complex reality of spurious solutions and convex solution sets, and finally to the order restored by regularization, provides a comprehensive understanding of the information encoded within the optimizer.

The initial proof of the theorem reveals that an optimal solution with minimal support must have that support be a maximum clique. This establishes the ideal correspondence: the non-zero components of such a vector identify the vertices of a maximum clique, and their values are uniform, equal to 1/ω(G). However, this ideal is quickly complicated. For the original, unregularized quadratic program, the set of global optimizers can be the entire convex hull of the characteristic vectors of multiple maximum cliques, and the program admits "spurious" local and global optima whose supports are not cliques at all. The information in an arbitrary optimal vector x∗ is therefore ambiguous; its support may be a union of cliques or a non-clique structure altogether.

A deeper analysis via the Karush-Kuhn-Tucker conditions provides the crucial explanation for this behavior. The first-order optimality conditions only enforce an "equi-payoff" property, which for characteristic vectors, corresponds to the induced subgraph being regular—a far weaker condition than being a clique. This fundamental mismatch between the local optimality conditions and the desired global combinatorial structure is the root cause of the formulation's algorithmic challenges.

Ultimately, it is **regularization** that provides the definitive answer to interpreting the optimal vector. By adding a simple, strictly convex term to the objective function, the optimization landscape is reshaped to eliminate all spurious solutions and enforce a strict one-to-one correspondence between optima and cliques. For a properly regularized Motzkin-Straus program, the optimal solution vector x∗ becomes an unambiguous informant:

* A **global optimum** is always the characteristic vector of a **maximum clique**.
* A **local optimum** is always the characteristic vector of a **maximal clique**.
* The support of the optimal vector, supp(x∗), directly and reliably identifies the vertices of the corresponding clique structure.

The "flaws" of the original formulation are not mere technicalities but are themselves informative, revealing that the quadratic program captures a more generalized notion of graph cohesiveness. Understanding these flaws motivates the use of regularization and leads to a deeper appreciation for the interplay between the combinatorial properties of a graph and the geometric properties of its associated continuous optimization problem. The Motzkin-Straus theorem is thus more than a single result; it is an entry point into a rich theory connecting graph structure, continuous optimization, and conic programming, with powerful extensions and practical applications. The optimal solution vector x∗ is a complex object whose structure reflects the symmetries, regularities, and multiplicities of the densest subgraphs within a network, and its interpretation depends critically and precisely on the formulation of the quadratic program being solved.

The following table provides a concise summary of these findings, contrasting the properties of the optimal solution vector across the different formulations discussed.

**Table 1: Properties of Optimal Solutions for Variants of the Motzkin-Straus Program**

| **Formulation** | **Objective Function f(x)** | **Global Optima x\*** | **Nature of x\*** | **Support supp(x\*)** | **Spurious Solutions Possible?** |
| --- | --- | --- | --- | --- | --- |
| **Original Motzkin-Straus** | $x^T A x$ | Corresponds to maximum cliques C of size $k=\omega(G)$. | Can be $x\_C$, a point in $\text{conv}(\{x\_C\})$, or a spurious solution. | Can be $C$, $\cup C\_i$, or a non-clique set. | Yes, both local and global. |
| **Regularized (Bomze)** | $x^T(A + cI)x$, $c>0$ | Corresponds one-to-one with maximum cliques C. | Always the characteristic vector $x\_C$. | Always a maximum clique C. | No. Local optima correspond one-to-one with maximal cliques. |
| **Weighted** | $x^T(W A W)x$ (example) | Corresponds to maximum weight cliques $Q^\*$. | Proportional to weights: $x\_i = w\_i / \omega(G,w)$. | Always a maximum weight clique $Q^\*$. | Yes, similar issues to original can arise without regularization. |
| **Hypergraph (Regularized)** | $\min L\_{\bar{G}}(x) + \tau \sum x\_i^k$ | Corresponds one-to-one with maximum cliques C. | Always the characteristic vector $x\_C$. | Always a maximum clique C. | No (by construction of the formulation). |

#### Works cited

1. Finite convergence of sum-of-squares hierarchies for the stability number of a graph - arXiv, accessed July 7, 2025, <https://arxiv.org/pdf/2103.01574>
2. Solving the Maximum Clique Problem with Symmetric Rank-One Non-negative Matrix Approximation - ORBi UMONS, accessed July 7, 2025, <https://orbi.umons.ac.be/bitstream/20.500.12907/41911/1/23%20Solving%20the%20Maximum%20Clique%20Problem%20with%20Symmetric%20Rank-One%20Non-negative%20Matrix%20Approximation.pdf>
3. A review on algorithms for maximum clique problems - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/278716368_A_review_on_algorithms_for_maximum_clique_problems>
4. A New Trust Region Technique for the Maximum Weight Clique Problem\* | Optimization Online, accessed July 7, 2025, <https://optimization-online.org/wp-content/uploads/2002/01/430.pdf>
5. (PDF) A Note on the KKT Points for the Motzkin-Straus Program - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/370775374_A_Note_on_the_KKT_Points_for_the_Motzkin-Straus_Program>
6. Continuous Cubic Formulations for Cluster Detection Problems in Networks - Austin Buchanan, accessed July 7, 2025, <https://austinlbuchanan.github.io/files/Continuous%20cubic.pdf>
7. My favorite application using eigenvalues - UCSD Math Department, accessed July 7, 2025, <https://www.math.ucsd.edu/~fan/teach/262/13/262notes/Craig_Midterm.pdf>
8. Continuous Characterizations of the Maximum Clique Problem | Mathematics of Operations Research - PubsOnLine - INFORMS.org, accessed July 7, 2025, <https://pubsonline.informs.org/doi/10.1287/moor.22.3.754>
9. Lecture 1. Turán problem 1, accessed July 7, 2025, <https://www.ibs.re.kr/ecopro/wp-content/uploads/2022/03/topic-comb-lecture1.pdf>
10. Lecture 4: Turán's theorem and Erd˝os-Stone theorem 1 Turán theorem - 4th proof, accessed July 7, 2025, <https://homepages.inf.ed.ac.uk/hguo/files/16.fall-adv.comb/Lecture_4_17.10.2016.pdf>
11. On Generalized KKT Points for the Motzkin-Straus Program - arXiv, accessed July 7, 2025, <https://arxiv.org/pdf/2305.08519>
12. Replicator Equations, Maximal Cliques, and Graph Isomorphism, accessed July 7, 2025, <https://proceedings.neurips.cc/paper/1512-replicator-equations-maximal-cliques-and-graph-isomorphism.pdf>
13. Replicator Equations, Maximal Cliques, and Graph Isomorphism - Dipartimento di Scienze Ambientali, Informatica e Statistica, accessed July 7, 2025, <https://www.dsi.unive.it/~pelillo/papers/NeuralComputation%201999.pdf>
14. Maxima for Graphs and a New Proof of a Theorem of Turán ..., accessed July 7, 2025, <https://www.cambridge.org/core/journals/canadian-journal-of-mathematics/article/maxima-for-graphs-and-a-new-proof-of-a-theorem-of-turan/AC3CC45896B053B75C856F25829CA95C>
15. [2305.08519] On generalized KKT points for the Motzkin-Straus program - arXiv, accessed July 7, 2025, <https://arxiv.org/abs/2305.08519>
16. Lecture 22: - The following claim was used in the proof of, accessed July 7, 2025, <https://www.tcs.tifr.res.in/~kavitha/22-Lecture-07-Apr-2022.pdf>
17. A generalization of the Motzkin–Straus theorem to hypergraphs, accessed July 7, 2025, <https://www.dsi.unive.it/~pelillo/papers/OL09.pdf?ref=binfind.com/web>
18. A generalization of the Motzkin–Straus theorem to hypergraphs - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/215990775_A_generalization_of_the_Motzkin-Straus_theorem_to_hypergraphs>
19. A GENERAL REGULARIZED CONTINUOUS FORMULATION FOR ..., accessed July 7, 2025, <https://optimization-online.org/wp-content/uploads/2017/09/6203.pdf>
20. On the maxima of motzkin-straus programs and cliques of graphs - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/360918379_On_the_maxima_of_motzkin-straus_programs_and_cliques_of_graphs>
21. (PDF) Feasible and Infeasible Maxima in a Quadratic Program for ..., accessed July 7, 2025, <https://www.researchgate.net/publication/2699614_Feasible_and_Infeasible_Maxima_in_a_Quadratic_Program_for_Maximum_Clique>
22. POLITECNICO DI TORINO Repository ISTITUZIONALE, accessed July 7, 2025, <https://iris.polito.it/bitstream/11583/2982472/1/2305.08519.pdf>
23. arXiv:1709.02486v1 [math.OC] 7 Sep 2017, accessed July 7, 2025, <https://arxiv.org/pdf/1709.02486>
24. Analysis of a regularized continuous formulation for the maximum s-plex problem - UNIVERSITÀ DEGLI STUDI DI PADOVA, accessed July 7, 2025, <https://thesis.unipd.it/retrieve/e96207bc-3520-46da-b608-6cf6aaf71a8b/Bellesso_Eleonora.pdf>
25. A detail in the proof of the Motzkin-Straus theorem - MathOverflow, accessed July 7, 2025, <https://mathoverflow.net/questions/130479/a-detail-in-the-proof-of-the-motzkin-straus-theorem>
26. Continuous Characterizations of the Maximum Clique Problem - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/2259188_Continuous_Characterizations_of_the_Maximum_Clique_Problem>
27. A General Regularized Continuous Formulation for the Maximum Clique Problem | Request PDF - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/319622193_A_General_Regularized_Continuous_Formulation_for_the_Maximum_Clique_Problem>
28. A new trust region technique for the maximum weight clique problem, accessed July 7, 2025, <http://www.stasbusygin.org/writings/qualex-ms.pdf>
29. ON CLIQUES AND LAGRANGIANS OF HYPERGRAPHS 1. Introduction Given a graph G, the Motzkin and Straus formulation of the maximum cli - Korea Science, accessed July 7, 2025, <https://koreascience.kr/article/JAKO201916842430378.pdf>
30. Sum of Squares (SOS) Techniques: An Introduction, accessed July 7, 2025, <https://www.princeton.edu/~aaa/Public/Teaching/ORF523/S16/ORF523_S16_Lec15.pdf>
31. Copositive Programming – a Survey - Optimization Online, accessed July 7, 2025, <https://optimization-online.org/wp-content/uploads/2009/11/2464.pdf>
32. Sum-of-Squares Representations for Copositive Matrices and Independent Sets in Graphs - CWI, accessed July 7, 2025, <https://ir.cwi.nl/pub/33433/33433D.pdf>
33. This document is downloaded from DR-NTU, Nanyang Technological University Library, Singapore. - MIT, accessed July 7, 2025, <http://mit.edu/~a_a_a/Public/Publications/refs_for_seb_blog/DeKlerk_Pasechnik_stable_set_sos.pdf>
34. Exploiting symmetry in copositive programs via semidefinite hierarchies - Optimization Online, accessed July 7, 2025, <https://optimization-online.org/wp-content/uploads/2013/12/4181.pdf>
35. Approximation of the Stability Number of a Graph via Copositive Programming | SIAM Journal on Optimization, accessed July 7, 2025, <https://epubs.siam.org/doi/10.1137/S1052623401383248>
36. Copositive Programming and Combinatorial Optimization, accessed July 7, 2025, <https://www.math.uwaterloo.ca/~hwolkowi/henry/teaching/gradstudiesandStudentsPostdocs.d/pprsgradseminarF08.d/eth08.pdf>
37. Copositive matrices, sums of squares and the stability number of a graph - Centrum Wiskunde & Informatica, accessed July 7, 2025, <https://ir.cwi.nl/pub/33270/33270.pdf>
38. Improved Dynamics for the Maximum Common Subgraph Problem - arXiv, accessed July 7, 2025, <https://arxiv.org/html/2403.08703v1>
39. Approximating the maximum weight clique using replicator dynamics - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/5602212_Approximating_the_maximum_weight_clique_using_replicator_dynamics>
40. REPLICATOR DYNAMICS IN COMBINATORIAL OPTIMIZATION ..., accessed July 7, 2025, <https://www.dsi.unive.it/~pelillo/papers/EoO-replicator.pdf>
41. Solving the Maximum Clique Problem with Symmetric Rank-One Nonnegative Matrix Approximation - arXiv, accessed July 7, 2025, <https://arxiv.org/pdf/1505.07077>
42. The replicator equation and other game dynamics - PubMed, accessed July 7, 2025, <https://pubmed.ncbi.nlm.nih.gov/25024202/>
43. A review on algorithms for maximum clique problems - Université Angers, accessed July 7, 2025, <https://leria-info.univ-angers.fr/~jinkao.hao/papers/WuHaoEJOR2014.pdf>
44. Analysis of a Maximal Clique Finding Algorithm with respect to Runtime and E ectiveness in High Dimensional Data, accessed July 7, 2025, <https://epub.ub.uni-muenchen.de/70669/1/MA_Litzka.pdf>
45. [1505.07077] Solving the Maximum Clique Problem with Symmetric Rank-One Nonnegative Matrix Approximation - arXiv, accessed July 7, 2025, <https://arxiv.org/abs/1505.07077>
46. Generalizing the Motzkin-Straus Theorem to Edge-Weighted Graphs, with Applications to Image Segmentation | Request PDF - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/220728725_Generalizing_the_Motzkin-Straus_Theorem_to_Edge-Weighted_Graphs_with_Applications_to_Image_Segmentation>
47. On the maxima of motzkin-straus programs and cliques of graphs - OUCI, accessed July 7, 2025, <https://ouci.dntb.gov.ua/en/works/loXLGYj9/>
48. Image Segmentation in Computer Vision and Review of Various Segmentation Techniques | Request PDF - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/275637547_Image_Segmentation_in_Computer_Vision_and_Review_of_Various_Segmentation_Techniques>
49. Efficient Graph-Based Image Segmentation - Brown Computer Science, accessed July 7, 2025, <https://cs.brown.edu/people/pfelzens/papers/seg-ijcv.pdf>
50. Unsupervised Image Segmentation based Graph Clustering Methods - SciELO México, accessed July 7, 2025, <https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462020000300969>
51. Dominant sets clustering for image retrieval - ResearchGate, accessed July 7, 2025, <https://www.researchgate.net/publication/220228831_Dominant_sets_clustering_for_image_retrieval>
52. (PDF) Dominant Sets for "Constrained" Image Segmentation, accessed July 7, 2025, <https://www.researchgate.net/publication/318528049_Dominant_Sets_for_Constrained_Image_Segmentation>
53. A density-based enhancement to dominant sets clustering | IET Computer Vision, accessed July 7, 2025, <https://digital-library.theiet.org/doi/10.1049/iet-cvi.2013.0072>