

PAPER • OPEN ACCESS

Mathematical Programming Computational for Solving NP-Hardness Problem

To cite this article: Ahmed Alridha and Ahmed Sabah Al-Jilawi 2021 *J. Phys.: Conf. Ser.* **1818** 012137

View the [article online](#) for updates and enhancements.

You may also like

- [An augmented Lagrangian trust region method for inclusion boundary reconstruction using ultrasound/electrical dual-modality tomography](#)
Guanghui Liang, Shangjie Ren and Feng Dong
- [A regularization method based on level sets and augmented Lagrangian for parameter identification problems with piecewise constant solutions](#)
J P Agnelli, A De Cezaro and A Leitão
- [Formation of ZnO Nanoparticles by Atomic Layer Deposition for the Nonvolatile Memory Thin-Film Transistor Applications](#)
Gi-Ho Seo, Da-Jeong Yun, Won-Ho Lee et al.



The Electrochemical Society
Advancing solid state & electrochemical science & technology

241st ECS Meeting

May 29 – June 2, 2022 Vancouver • BC • Canada

Abstract submission deadline: Dec 3, 2021

Connect. Engage. Champion. Empower. Accelerate.
We move science forward



Submit your abstract



Mathematical Programming Computational for Solving NP-Hardness Problem

Ahmed ALRIDHA^{1*}, Dr Ahmed Sabah Al-Jilawi ²

¹ Ministry of Education, General Directorate of Education in Babylon.

²Department of Mathematics -University of Babylon.

E-mail: amqa92@yahoo.com, aljelawy2000@yahoo.com

Abstract. In this paper we will introduce a new approach for solving K-cluster problem which is one of the NP-hardness problem, in combinatorial optimization problems. In addition, P is NP-hardness if and only if the polynomial time of each NP problem is reduced to P. Actually, our study was focused on the two methods which is Penalty and Augmented Lagrangian methods base on the numerical result. Moreover, we tested the K-cluster problem and found the Augmented Lagrangian Method faster than Penalty method. Finally, our research is not just focus on the numerical computational but also improving the theoretical converges properties.

Keywords: Semidefinite programming, nonlinear programming, K-cluster problem, Penalty and Augmented Lagrangian methods.

1. Introduction of Combinatorial Optimization Problems

The general goal of problems with optimization is to achieve a solution with optimum objective function values [1]. In a brief sentence, optimization means finding the best solution of the problem. Optimization is an important branch that, in addition to being an important mathematics branch, is commonly applied in our public life. A combinatorial problem consists of finding one that satisfies a set of constraints among a finite set of objects [1,3,6]. The optimization of many technically important combinatorial issues is the main goal. Optimization problems can be seen as generalizations of decision problems, where the solutions are additionally evaluated by an objective function and the goal is to find solutions with optimal objective function values [7]. The objective function is often defined on candidate solutions as well as on solutions; the objective function value of a given candidate solution (or solution) is also called its solution quality.

Depending on whether the given objective function is to be minimized or maximized, any combinatorial optimization problem can be stated as a minimization problem or as a maximization question [18]. Sometimes, one of the two formulations is more natural, but problems of minimization and maximization are viewed equivalently algorithmically [15]. We distinguish two variants for each combinatorial optimization problem: the search variant: given a problem example, find a solution with minimal (or maximal, respective) solutions. On the other hand, in the evolutionary computing community, numerical optimization problems are very common; all major evolutionary techniques (genetic algorithm, evolutionary strategies, evolutionary programming) on these issues are very popular. Many of these tools, however, are struggling to solve certain real-world problems that include non-



trivial limitations (as well as other classic optimization methods). The global solution also lies within the limits of the feasible field for problems like these. It is therefore important to investigate some of the problem-specific factors that effectively explore these limits [2]. Class P consists of all solvable decision problems in polynomial-time. NP to be classify in two misunderstandings as follows:

(1) The class of problems that cannot be solved in polynomial time is NP.

(2) If its answer can be verified in polynomial-time, the decision problem belongs to class NP. The misconception (1) stems from the clarification of the short name NP for "Not Polynomial-time savable", but in a general sense of computation, non-deterministic computation, that is, NP is the class of all non-deterministic polynomials. Thus, NP is the brief name of "Nondeterministic Polynomial-time" [3]. Now we introduce some common types of NP-hardness:

1- Max-Cut problem

2- k-cluster problem

3- Maximum independent set problem

and others

The focus of our research paper will be on K- cluster problem. These three problems generally binary quadratic problem are NP hardness problem and often difficult to solve with normal application.

2. Background Mathematical (Basic Concept)

In this section, we provide the necessary definitions, notation, and results for Euclidean space. We also provide some basic concepts and facts for real mathematical analysis. For more see [1,5,12,8,4]

2.1. Convex set

Let $T \subseteq \mathbb{R}^n$. If the line segment between any two points in T lies in T , i.e. $\alpha x_1 + (1 - \alpha)x_2 \in T, \forall x_1, x_2 \in T, \forall \alpha \in [0,1]$ then T is said to be convex. It can be shown that a set $T \subseteq \mathbb{R}^n$ is convex if and only if for any $x_1, \dots, x_n \in T$, as in Figure (1,2,3), for more details see [1].

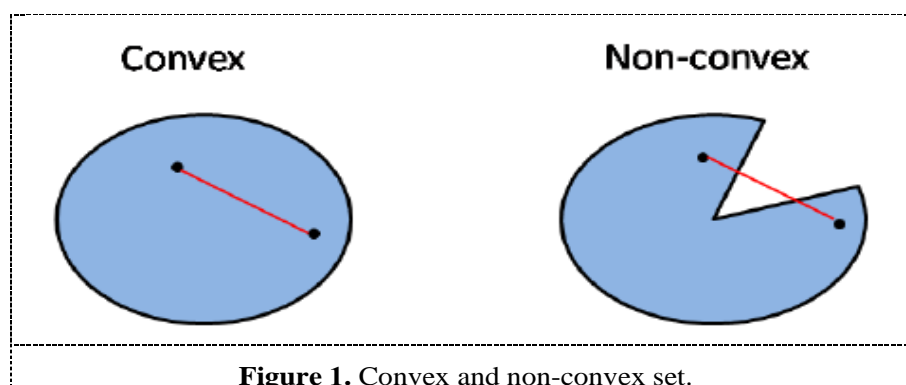
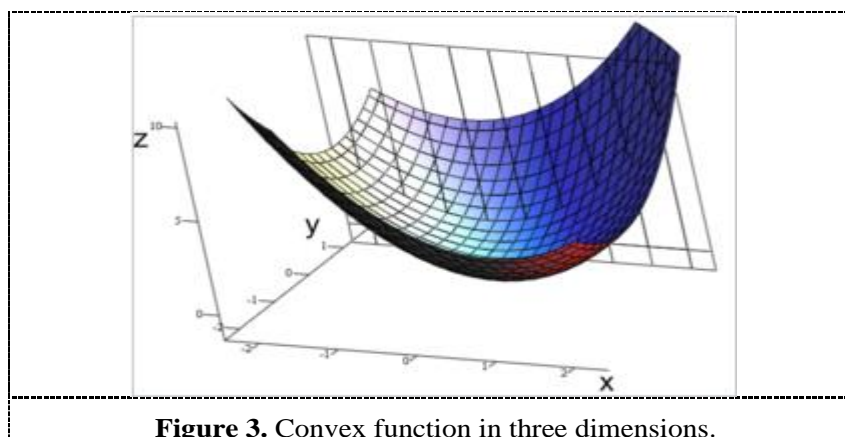
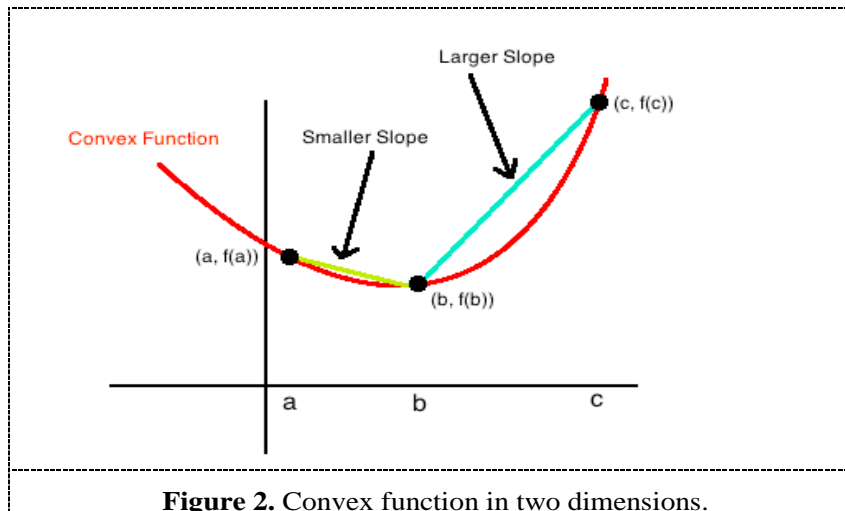


Figure 1. Convex and non-convex set.

2.2. Convex Functions

Let $T \subseteq \mathbb{R}^n$ be a nonempty convex set if $f: T \rightarrow \mathbb{R}$ satisfies

$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2), \forall x_1, x_2 \in T, \forall \alpha \in [0,1]$ Then f is said to be a convex function on T .



2.3. Local minimizer

Let $f: T \rightarrow \mathbb{R}$ and x is called a local minimizer of f if there is a scalar $\alpha > 0$ such that $f(x^*) \leq f(x)$ for all $x \in \beta(x^*, t) = \{x \in \mathbb{R}^n, \|x - x^*\| \leq t\}$.

2.4. Global minimizer

We call x^* is a global minimizer if $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$.

1. Symmetric Matrices

[11] Let $X = [x_{ij}]$ be a square matrix then the matrix is called symmetric if $X = X^T$.

Let S^n stand for the set of all of symmetric $n \times n$ matrices:

$$S = \{X \in \mathbb{R}^{n \times n} \mid X = X^T\}$$

Furthermore $X > 0$ ($X \geq 0$) means that X is symmetric and positive definite (semidefinite); that is, for all nonzero $u \in \mathbb{R}^n$ ($u^T X u = \sum_{ij} X_{ij} u_i u_j$ for all $u \in \mathbb{R}^n$).

Example 2.4.1

We will provide a simple example for positive semidefinite

Let $A = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$ When calculating the determinant of this matrix is equal to zero, which means that the matrix is positive semidefinite

Example 2.4.2

Let $B = \begin{bmatrix} 1 & -1 \\ 3 & 4 \end{bmatrix}$

Similar to the previous example when computing the determinant of this matrix is equal to 7, which means the matrix is positive semidefinite.

1.1. Norm

[1, 13] A function $\|\cdot\|: \mathbf{R}^n \rightarrow \mathbf{R}$ is said to be a norm, if it satisfies the following properties:

1. $\|x\| \geq 0, \forall x \in \mathbf{R}^n; \|x\| = 0$ if and only if $x = 0$
2. $\|\alpha x\| = |\alpha| \|x\|, \forall \alpha \in \mathbf{R}, \forall x \in \mathbf{R}^n$
3. $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbf{R}^n$

3. Summary of Canonical Convex Problem

(LP) Linear programming: $c^T x$ subject to $Qx \leq h, Ax = b$

(QP) Quadratic program: such LP but with quadratic objective function

(SDP) Semidefinite program: such LP but with matrices

Conic program: the most general form of convex problem

Second order cone problem (SOCP): $\min c^T x$ subject to $\|D_i x + d_i\|_2 \leq e_i^T x + f_i, i = 1, \dots, p$

The following diagram shows the relationship between these convex [14,17].

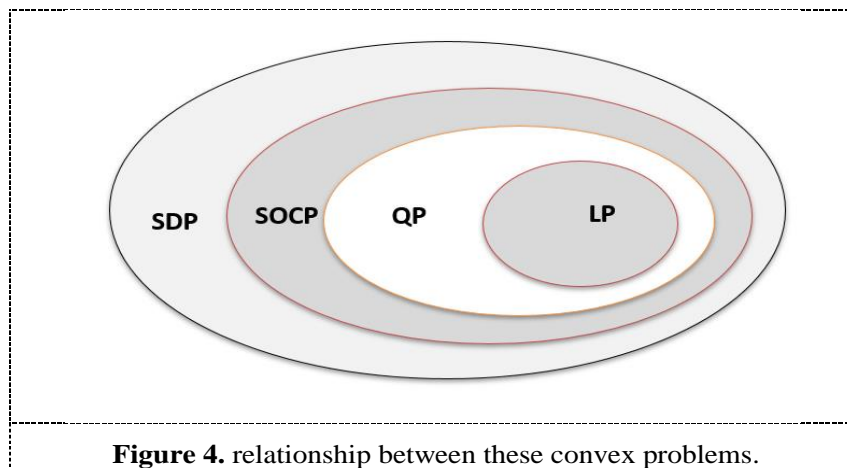


Figure 4. relationship between these convex problems.

4. Mathematical Programming

4.1 Linear programming (LP)

The standard LP problem is given by

$$(LP) \begin{cases} \text{minimizing } \langle c, x \rangle \\ \text{subject to } \langle a_i, x \rangle = b_i, i = 1, \dots, m \\ x \geq 0, x \in \mathbf{R}^n \end{cases} \quad \dots\dots\dots (1)$$

Where \mathbf{R} is the set of real number, \mathbf{R}^n is the set of vectors of length n with real entries and the problem data is given by vectors $a_i \in \mathbf{R}^n, b_i \in \mathbf{R}$ and $c \in \mathbf{R}^n$.

4.2 Semidefinite Programming (SDP)

The standard semidefinite programming (SDP) problem is given by:

$$(SDP) \begin{cases} \min < C, X > \\ \text{subject to } < A_i, X > \geq b_i, \forall i = 1, \dots, m, \\ X \succeq 0, X \in S^n \end{cases} \dots\dots\dots (2)$$

Where S^n is the set of $n \times n$ symmetric matrices, the matrices $A_i, C \in S^n$, the number $b_i \in \mathbb{R}$ for $i = 1, \dots, m$.

The dual problem of the SDP problem is given by:

$$(SDD) \begin{cases} \max < b, y > \\ \text{subject to } \sum_{i=1}^m y_i A_i + S = C \\ S \succeq 0 \end{cases} \dots\dots\dots (3)$$

Example 4.2.1

The matrices below show us the semidefinite programming from $n=3, m=2$

$$A_1 = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 4 & 3 \\ 2 & 3 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 4 & 7 \\ 2 & 7 & 9 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 5 \\ 1 & 5 & 0 \end{bmatrix}, \quad \text{and } b = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

And the X matrix represent 3×3 symmetric matrix

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}, \quad \text{and let}$$

$$C.X = 0x_{11} + x_{13} + 4x_{22} + x_{31} + 5x_{32} + 0x_{33} = 2$$

Then we can write the semidefinite programming as follows

$$(eSDP) \begin{cases} \text{maximize } 0x_{11} + x_{13} + 4x_{22} + x_{31} + 5x_{32} + 0x_{33} \\ \text{subject to } 0x_{11} + 2x_{13} + 4x_{22} + 3x_{31} + 4x_{32} + 0x_{33} = 2 \\ 2x_{11} + 2x_{13} + 4x_{22} + 2x_{31} + 7x_{32} + 9x_{33} = 1 \\ X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \succeq 0, X \in S^n \end{cases} \dots\dots\dots (4)$$

5. Duality Theory

Theorem 5.1 (Weak Duality)

If $x \in \mathbb{R}^n$ is feasible for linear programming (LP) and $y \in \mathbb{R}^m$ is feasible for its duality then

$$c^T x \leq y^T A x \leq b^T y$$

If LP is unbounded, then the duality D is necessarily infeasible, if D is unbounded then P is necessarily infeasible. On the other hand, if $c^T x' = b^T y'$ such that x' , feasible for P and D respectively, then x' must solve P and y' must solve D [11].

Theorem 5.2 (Strong Duality)

If either P or D has a finite optimal value, then so does the other, the optimal values coincide, and optimal solutions to both P and D exist [6].

Theorem 5.3 (Strong Duality For The Dual)

If there exists \mathbf{y}^* strictly feasible for (SDD), then there exists an optimal solution \mathbf{x}^* for (SDP) and $\langle \mathbf{C}, \mathbf{x}^* \rangle = d^*$ [18].

6. K- Cluster Problem

In this section we will review some basics about a problem K- cluster problem which is one of NP-hardness problems. Also, we briefly introduce the k-cluster problem, the combinatorial optimization problem we consider in this work. We refer to the recent [LMZ08] for complete references, and for discussions on the SDP approach for this problem. Consider an undirected, weighted graph $G = (V, E)$ of n vertices v_1, \dots, v_n , and a non-negative weight w_{ij} on each edge (v_i, v_j) . For a given integer k in $\{1, \dots, n\}$, the k-cluster problem consists in determining a subset S of k vertices such that the total edge weight of the subgraph induced by S is maximized. In unweighted graphs, the problem is also called the densest subgraph problem. Denoting $W = (w_{ij})_{ij}$ the weight-matrix of the graph G , the problem can be written as a $\{0, 1\}$ -quadratic optimization problem with one linear constraint. The general form of k - cluster problem

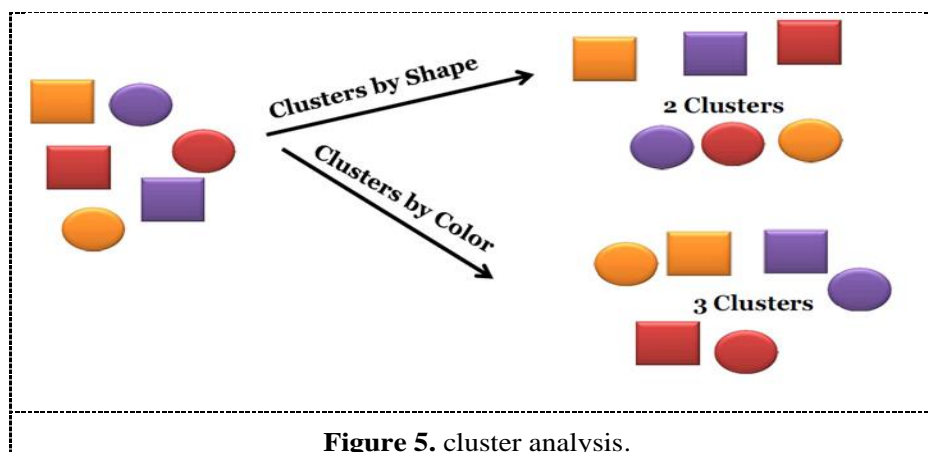
$$\frac{\sum_{ij} w_{ij} g_i g_j}{2} \dots\dots\dots (5)$$

Subject to $\sum_{i=1}^n g_i = k$

$$g \in \{0,1\}^n.$$

6.1 The cluster analysis

The process of breaking a series of input data into potentially Overlaps, subsets, where elements are present in each subset considered connected by some measure of similarity



6.2 Graph Clustering

A related problem is that of graph clustering. While graph partitioning works by partitioning a graph into agiven number of chunks of roughly same sizes, this constraint is not to be found in graph clustering. Instead, the goal is to detect strongly connected groups of vertices within the graph, i.e. groups that have a significantly higher edge-density within themselves than towards the outside. Figure 7 shows an example for a clustering of a small graph. The minimum cut constraint holds, but the clusters are of

different sizes. Graph clustering is often referred to as community detection in the spirit of social networks

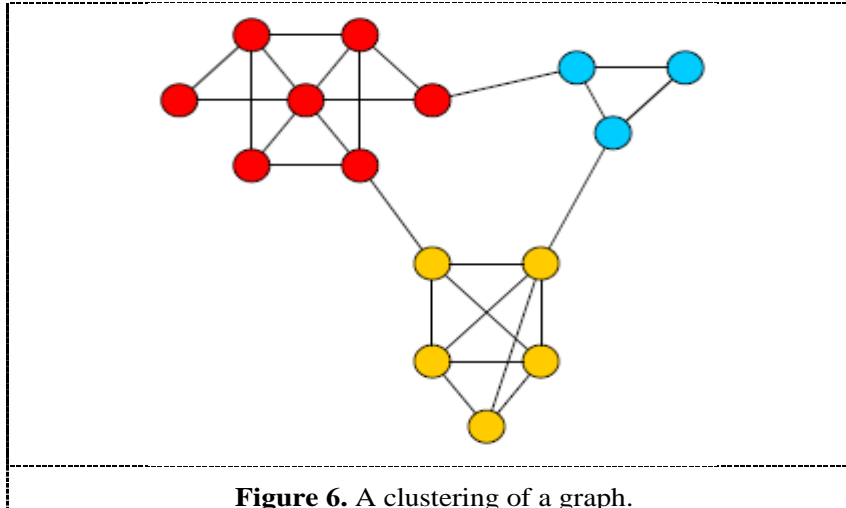


Figure 6. A clustering of a graph.

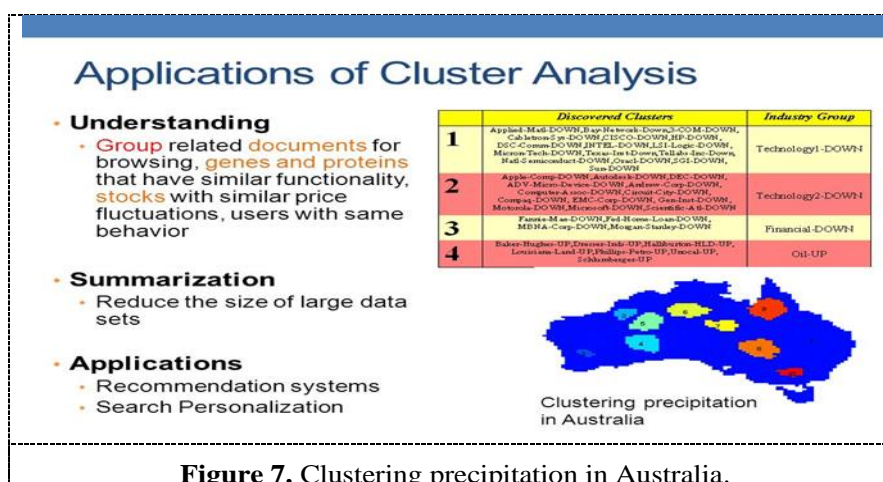
6.3 Types of Cluster in Graph

Between-graph: for example, a series of graphs describing chemical compounds may be able to base on their structural similarities; they can be grouped into clusters.

Within-graph: for example, these clusters may be in a social networking graph, Representing people with similar / same hobbies [3, 7, 9].

7. Applications of Cluster Analysis in Our Life

For classifying the water spring data, cluster analysis can be used. It is also used to categorize evidence that gives a single-purpose approach within electronic computing [9, 10].



special technique With a special technique by adding one term to the objective function and then add constraint to the objective function to become unconstrained i.e (that is, we transfer the problem from constraint to unconstrained) [1, 14,16].

Let we have the following linear equality constrained problem:

$$\begin{aligned} &\text{Minimize} && \langle c, x \rangle \\ &\text{Subject to} && \langle a_i, x \rangle = b_i, i = 1, \dots, m \\ &&& x \geq 0, x \in R^n \end{aligned} \quad \dots\dots\dots (6)$$

By replace previous problem with a penalized version

$$\begin{aligned} &\text{minimize} && \langle c, x \rangle + \mu^k \sum_{i=1}^m (\langle a_i, x \rangle - b_i) \\ &\text{subject to} && x \in X \end{aligned} \quad \dots\dots\dots (7)$$

The scalar μ^k is a positive penalty parameter, and decreasing μ^k to 0, the solution x^k of the penalized problem tends to decrease the constraint violation, consequently providing an increasingly accurate approximation to the main problem. A significant practical point here is that μ^k should be progressively decreased, using the optimal solution of each approximate problem to start the iteration of the next approximate problem. One choice for P_q can be the

$$\begin{aligned} &\text{minimize} && \langle c, x \rangle + \frac{1}{2\mu^k} \|Ax - b\|^2 \\ &\text{subject to} && x \in X \end{aligned} \quad \dots\dots\dots (8)$$

With a note that $Ax = b$ represents the system of equation $\langle a_i, x \rangle = b_i, i = 1, \dots, m$. The expanded Lagrangian structure, where we add a direct term to $P_q(y)$; including a multiplier vector $y^k \in \mathbb{R}^n$, is a significant improvement to the punishment work approach. Now we solve the follows problem instead of problem (8.2)

$$\begin{aligned} &\text{minimize} && \langle c, x \rangle + (y^k)^T (Ax - b) + \frac{1}{2\mu^k} \|Ax - b\|^2 \\ &\text{subject to} && x \in X \end{aligned} \quad \dots\dots\dots (9)$$

The multiplier vector y^k is modified by some formula that seeks to approximate an optimal dual solution after the solution of the above problem μ^k is obtained, so that

$$y^{k+1} = y^k + \frac{1}{\mu^k} (Ax^k - b).$$

This is called the augmented Lagrangian form of the first order. Finally, Penalty and augmented Lagrangian techniques can usually be used for inequalities and restrictions on equality.

9. Algorithmic of the Penalty Function Method

Now, we will show an algorithmic of the penalty function method

Algorithm

1 Given t° , and $\delta^\circ > 0$.

2 Find t^{k+1} such that

$$t^{n+1} = \operatorname{argmin} P_q(t, \delta^n).$$

3 Choose $\delta^{n+1} \leq \delta^n$

Set $n=n+1$ and repeat

9.1 Numerical Results

In our work, we have chosen some problems which is obtained from the Big Mac Library. Also, in this test, we chose two methods, namely the penalty methods and the Lagrangian methods. Moreover, we compared the results and took a new approach and mixed count, which is a new technique for selecting the fastest result as shown below.

When we took the problems (g05-60.4), (g05-60.7) as in Tables (1,2,3), graphs as shown in Figures 8 the mixed show as faster than the Penalty method and in figure 9 we had the effect of Lagrangian and Penalty increased by the two methods. It was shown to us in the Tables (3) that the Augmented Lagrangian Method is stronger and more precise than the Penalty Method, which means that the Augmented Lagrangian Method is the best. Its success was stronger and faster in order to find the best solutions.

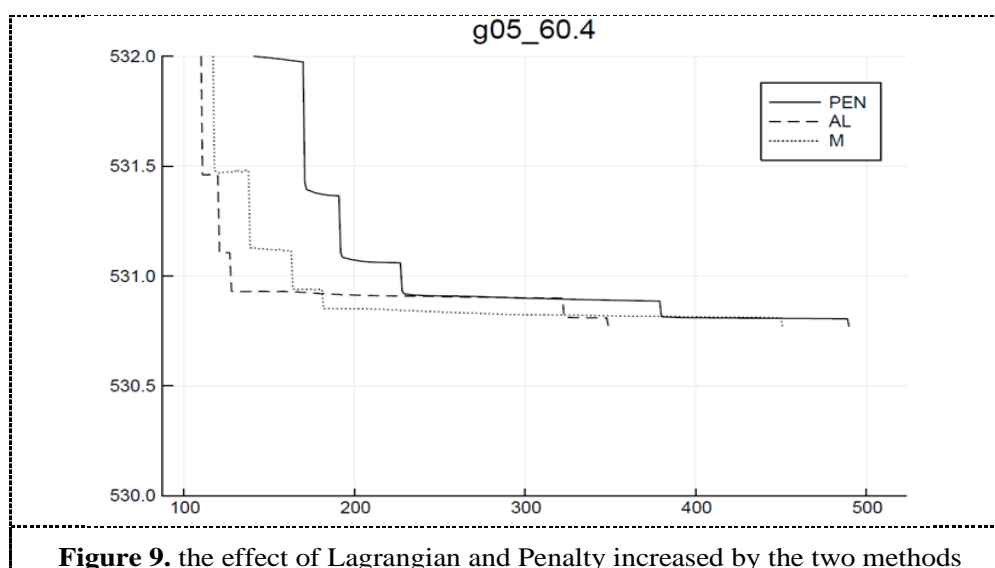
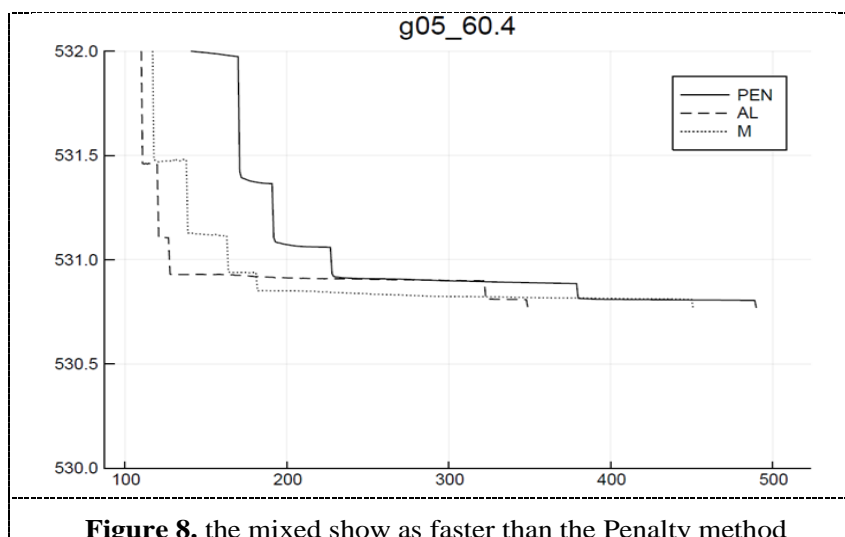


Table (1) :Our results in Graphs g05_60.0 to g05_60.9			
Graphs	Penalty Function call	Augmented Function call	Mixed Function call
g05_60.0	754	588	530
g05_60.1	850	444	494
g05_60.2	410	253	226
g05_60.3	891	850	524
g05_60.4	490	350	466
g05_60.5	874	322	445
g05_60.6	504	238	269
g05_60.7	788	515	480
g05_60.8	446	230	215
g05_60.9	518	271	393

Table (2) :Our results in Graphs g05_100.0 to g05_100.9			
Graphs	Penalty Function call	Augmented Function call	Mixed Function call
g05_100.0	401	156	232
g05_100.1	261	204	188
g05_100.2	517	240	228
g05_100.3	296	182	163
g05_100.4	509	266	284
g05_100.5	552	221	226
g05_100.6	484	320	255
g05_100.7	525	371	253
g05_100.8	438	200	314
g05_100.9	296	279	195

Table (3) :Our results in Graphs g05_80.0 to g05_80.9			
Graphs	Penalty Function call	Augmented Function call	Mixed Function call
g05_80.0	476	254	391
g05_80.1	1043	499	381
g05_80.2	713	285	325
g05_80.3	320	157	173
g05_80.4	680	273	269
g05_80.5	423	192	269
g05_80.6	459	256	238
g05_80.7	413	210	254
g05_80.8	445	213	190
g05_80.9	337	215	187

10. Conclusions

The main objective achieved from this paper is summarized as follows:

Some basic definitions that sought to achieve the objective of the research were revised

2. The problem of K-cluster is described in a simple mathematical way and what are its applications in our general life.

3. We conducted a study on choosing the best solution in two ways, the penalty and augmented lagrangian methods, by examining the numerical results and examining the best behavior of the solution.

4. We selected some problems obtained from the Big Mac library and made a special technique and compared the numerical results to get the best solution.

References:

- [1] Al-Jilawi, A. (2019). Solving the Semidefinite Programming Relaxation of Max-cut Using an Augmented Lagrangian Method. Northern Illinois University.
- [2] Schoenauer, M., & Michalewicz, Z. (1996, September). Evolutionary computation at the edge of feasibility. In *International Conference on Parallel Problem Solving from Nature* (pp. 245-254). Springer, Berlin, Heidelberg.
- [3] Belloni, A. (2005). Lecture notes for iap 2005 course introduction to bundle methods. Operation Research Center, MIT, Version of February, 11.
- [4] Bertsekas, D. P., & Scientific, A. (2015). *Convex optimization algorithms*. Belmont: Athena Scientific.
- [5] Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.
- [6] Burer, S., Monteiro, R. D., & Zhang, Y. (2002). Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12(2), 503-521.
- [7] Dempe, S., & Bard, J. F. (2001). Bundle trust-region algorithm for bilinear bilevel programming. *Journal of Optimization Theory and Applications*, 110(2), 265-288.
- [8] Bertsekas, D. P., Hager, W., & Mangasarian, O. (1999). *Nonlinear Programming*. Athena Scientific Belmont. Massachusetts, USA.
- [9] Du, Y., & Ruszczyński, A. (2017). Rate of convergence of the bundle method. *Journal of Optimization Theory and Applications*, 173(3), 908-922.
- [10] Buriol, L. S., Figueiredo, C., Resende, M. G., & Uchoa, E. (2020). THE GUIDE TO NP-COMPLETENESS IS 40 YEARS OLD: AN HOMAGE TO DAVID S. JOHNSON. *Pesquisa Operacional*, 40.
- [11] Hochba, D. S. (Ed.). (1997). *Approximation algorithms for NP-hard problems*. ACM Sigact News, 28(2), 40-52.
- [12] Jumaa, B. R., & Al-Jilawi, A. S. (2020, July). Solving Max-Cut Optimization Problem. In *Journal of Physics: Conference Series* (Vol. 1591, No. 1, p. 012051). IOP Publishing.
- [13] Hestenes, M. R. (1969). Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5), 303-320.
- [14] de Oliveira, W., & Tcheou, M. P. (2019). An inertial algorithm for DC programming. *Set-Valued and Variational Analysis*, 27(4), 895-919.
- [15] Joki, K. (2013). *Nonsmooth Optimization: Bundle Methods*.
- [16] Kanzow, C., Steck, D., & Wachsmuth, D. (2018). An augmented Lagrangian method for optimization problems in Banach spaces. *SIAM Journal on Control and Optimization*, 56(1), 272-291.
- [17] King, R. P. (1966). Necessary and sufficient conditions for inequality constrained extreme values. *Industrial & Engineering Chemistry Fundamentals*, 5(4), 484-489.
- [18] Kiwiel, K. C. (1990). Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical programming*, 46(1-3), 105-122.