How to Cite:
Salman, A. M., & Al-Jilawi, A. S. (2022). Solving nonlinear optimization problem using approximation methods. *International Journal of Health Sciences*, *6*(S3), 1578–1586.
https://doi.org/10.53730/ijhs.v6nS3.5699

Solving nonlinear optimization problem using approximation methods

Abbas Musleh Salman
Mathematics Department, University of Babylon
Email: abbas.mmm2019@gmail.com

Ahmed Sabah Al-Jilawi
Mathematics Department, University of Babylon
Email: aljelawy2000@yahoo.com

Abstract---The goal of this paper is to find a better method for integrating the optimization problem faster, and we did so by using a relaxation of the solution area, which is one of the methodologies in the correlation between the penalty method and the Augmented Lagrange method, and we also built default properties for combining these two methods.

Keywords---optimization, nonlinear optimization, relaxation.

Introduction

Nonlinear programming (NLP) is a mathematical technique for tackling optimization problems involving unknown constraints or objective functions. The primary goal of optimization is to determine the variables' values so that a mathematical function has a minimum and maximum value [3],[5]. The structure of optimization problems is divided into two types: continuous problems, where we have continuous variables, and discrete problems, which are also known as combinatorial optimization citation needed. There are numerous applications for optimization theory and methods in the fields of applied mathematics, computer science, business management, and military and space technology. The subject is concerned with the optimal solution of mathematically defined problems, i.e., the "best" solution to a practical problem can be found from numerous schemes using scientific methods and tools [11,12,13].

Optimization problems

The general basic formula of the optimization problem:
Minimize $f(x)$ ...(1)
subject to $x \in \Omega$

Corresponding author: *Salman, A. M.; Email: abbas.mmm2019@gmail.com*
Manuscript submitted: 18 Nov 2021, Manuscript revised: 09 Feb 2022, Accepted for publication: 27 March 2022

The function $f : R^n \longrightarrow R$ that we wish to minimize is a real-valued function called the objective function or "cost function". The vector x is an n-vector of independent variables $x = [x_1, x_{2,\dots,}x_n]^T \in R^n$·: The variables $x_1, x_{2,\dots,}x_n$ are often referred to as decision variables. The set $R^n$is a subset of R^n called the "constraint set" or "feasible set". The optimization problem above can be viewed as a decision problem that involves finding the "best" vector x of the decision variables over all possible vectors in R^n By the "best" vector we mean the one that results in the-smallest value of the objective function. This vector is called the minimizer of f over R^n. There could be a large number of minimizers. Finding any of the minimizers will do in this situation. Maximizes are also sought in optimization issues where the objective function must be maximized. Extremizers are people who are either maximizes for minimizers. If you're trying to solve a maximization problem, you can use the minimization $-f$ form above to get the same result. As a result, we can focus on minimization problems while still maintaining generality.

Mathematical optimization

A mathematical optimization problem, or simply an optimization problem, is of the following structure:
Minimize $f_0(x)$
subject to $f_i(x) \leq a_i, i = 1,\dots,m.$ ...(2)
The function $f_0 : R^n \rightarrow R$ signifies the objective function, the functions $f_i : R^n \rightarrow R$,$i = 1,\dots,m$ denote the (inequality) constraint functions, and the constants $a_1,\dots,a_m$ denote the constraints' restrictions, or bounds. For each z with $f_1(z) \leq a_1,\dots,f_m(z) \leq a_m$, we have optimum, or a solution to the problem (2), if a vector x* has the least objective value of all vectors that match the conditions.

$$f_0(z) \geq f_0(x^*).$$

Optimality conditions for constrained optimization

The optimization problem is called the general nonlinear programming problem if we also have some equality (or inequality $hj(x) \leq 0$, j = 1,…., I) constraints, or both of them, and it can be expressed as [11]:
minimize $f(x)$
subject to $g_i(x) = 0, i = 1,\dots,E$ ...(3)

$$x \in X$$

Where f and g_i defined from $X \subseteq R^n$ into R functions are considered to be continuously differentiable. The set of problem (3) that is feasible is denoted by

$$s = \{x \in X |\, g(x) = 0\}$$

Where g is the function with component functions $g_1,\dots,g_E$ In our study we assume X = R^nThe Lagrangian function $\mathcal{L}: R^n \times E \longrightarrow R$ is represented by

$$\mathcal{L}(x,\delta) = f(x) + \delta^T g(x),$$

where $\delta = (\delta_1,\dots,\delta_E)^T$is called the Lagrange multiplier vector. We have the following optimality conditions (See [3, 5, 11]).

The NP-Hard problem

A nondeterministic polynomial time problem known as the NP-hard problem is one of the most challenging. "On the Computational Complexity of Algorithms" by

June Hartman's and Richard Stearns was awarded the Turing Prize in 1965. There were hundreds more NP-complete problems discovered in the 1970s by theoretical computer scientists. Because there are so many steps simplex approach, introduced in 1947 by American mathematician George Dantzig, increases in direct proportion to the input, linear programming problems are NP-complete. There was discovered a polynomial-time algorithm in 1979 by Russian mathematician Leonid Khachian. He claims to have solved the puzzle of P and NP in August 2010 [11,14] in a move that could fundamentally change the way we utilize computers.

Karush-Kuhn-Tucker (KKT) conditions

The four conditions listed below are referred to as KKT conditions (for problems with differentiable f_i, g_i):

- Primal constraints: $f_i(x) \leq 0, i = 1, \ldots, I$, and $g_i(x) = 0, i = 1, \ldots, E$
- Complementary slackness: $\beta_i f_i(x) = 0, i = 1, \ldots, I$:
- Dual constraints: $\beta \geq 0$.

Gradient of Lagrangian with respect to x vanishes:
$\nabla f_0(x) + \sum_{i=1}^{I} \beta_i \nabla f_i(x) + \sum_{i=1}^{E} \gamma_i \nabla g_i(x)=0$

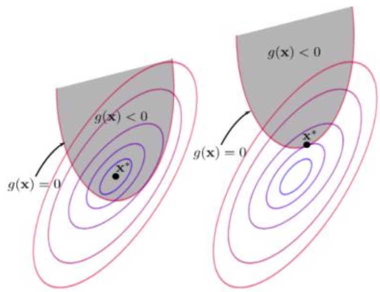

Figure 1. Graphical explanation for the KKT conditions

Example:
Minimize $2x_1^2 + x_2^2$
Subject to $x_1 + x_2 = 1$
Let us first consider the unconstrained case
Differentiate with respect to x_1 and x_2

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 4x_1$$
$$\frac{\partial f(x_1, x_2)}{\partial x_2} = 2x_2$$

These yield the solution $x_1 = x_2 = 0$. Does not satisfy the constraint

Nonlinear optimization

Nonlinear optimization (or nonlinear programming) is employed when the objective or constraint functions are nonlinear yet are not known to be convex. problem of the use of nonlinear programming (3) can't be solved using any effective methods. When it comes to solving issues with 10 variables and hundreds of variables, even the most straightforward situations can be difficult to solve. As a result, there are numerous techniques to solving the nonlinear programming problem, each of which requires some compromise. An optimization problem is to choose n variables $x_1, x_2, \dots\dots, x_n$ from a feasible region in order to optimize (minimize or maximize) a specified objective function $f(x_1, x_2, \dots\dots, x_n)$ of the choice variables, which is a general optimization problem in general. There are two types of NLPs: those with nonlinear objectives and/or feasible regions that are nonlinear restrictions constrain the options available. As a result, a nonlinear program in general can be expressed as follows in maximizing form:

$$\begin{cases} \text{Maximize} \quad f(x_1, x_2, \dots\dots, x_n) \\ \text{subject to:} \quad g_1\,(x_1, x_2, \dots\dots, x_n) \le \, b_1; \\ \quad \vdots \\ \quad g_m\,(x_1, x_2, \dots\dots, x_n) \le \, b_m \end{cases} \qquad \dots\dots(4)$$

where each of the constraint functions g_1 through g_m is given. A special case is the linear program that has been treated previously. The obvious association for this case is

$f(x_1, x_2, \dots\dots, x_n) = \sum_{j=1}^{n} c_j x_j$ and $g_1\,(x_1, x_2, \dots\dots, x_n = \sum_{j=1}^{n} a_{ij} x_j$, $(i\ =\ 1,2, \dots, m)$

Approximation Methods

In this section, we begin by discussing methods of approximation, concepts, and The terms we used. Furthermore, the Penalty and Augmented Lagrange method A method for both nonlinear programming problem and we offer the settlement. pproximation methods are a very active area of improvement. Consider minimizing Convex post $f: R^n \longrightarrow R$ on a convex array X. The objective of the approximation methods is Replace f and X with the approximation $f^k$and X^k. The approximation method only works if the approximation is easier than the original problem [7,11,15]. For every k iteration we tried to the second

X^{k+1}=$arg\ min_{x \in X^k} \quad f^k(x)$

Relaxed problem

There are several relaxations that could be considered while designing approximation methods for a combinatorial optimization problem. There is no one standard way to express a relaxation given a combinatorial optimization problem, for example. Different challenges necessitate varying degrees of restraint in the relaxation process. Multiple relaxations have been utilized to derive approximation techniques for similar combinatorial optimization problems in various circumstances. Find largest real k s.t. there is a vector X with real entries values in [0, 1] (i.e., $0\ \le x_i \le 1$), $\sum x_i\ =\ k$, and each edge $e\ =\ \{e_i, e_j\} \in E$ satisfies $x_i + x_j \le 1$.

Quadratic programming

Quadratic programming (QP) is a non-linear programming problem in which a quadratic objective function is optimized [9,10]. For a bilinear objective function, the constraints might be equalities or inequalities up to a second order polynomial. Image and signal processing, financial portfolio optimization, least-squares regression, chemical plant scheduling, and sequential quadratic programming are all examples of applications where QP is applied. When Wolfe and Frank devised the theoretical foundation for the problem in the early 1950s, and Markowitz applied it to portfolio optimization, an area of finance, it was the first time the subject had been studied in depth. A quadratic program (QP) is written in standard form as:

$$\begin{cases} \text{Minimize} & {}^{1}/_{2}\ x^T P x + q^T x \\ \text{Subject to} & Gx \preccurlyeq h \\ & Ax = b \end{cases} \quad \text{.....(5)}$$

The penalty function methods

A penalty method is a technique that involves solving a series of unconstrained problems. By adding a penalty element to the goal function, this technique converts limited issues to unconstrained problems [10, 11]. Courant studied the first penalty function in 1943 [5,16,17]. However, in the 1960s, sequential unconstrained minimization procedures became popular for tackling optimization issues (see [3]), and these approaches were employed to solve the original restricted problems. In the manner described in [11], we consider the standard penalty function (quadratic penalty function) for problem (2).

$$P_q(x,\theta) = f(x) + \frac{1}{2\theta}\|g(x)\|^2 \qquad \text{.......(6)}$$

where $\theta > 0$. Let $x(\theta)$ be a minimizer of $P_q(x,\theta)$ for $\theta > 0$. The penalty methods produce a sequence of infeasible solutions. Each iterate $x(\theta^k)$ is either necessarily infeasible or a local optimal solution of problem (5). These methods are beneficial because of their comparative simplicity as they can use powerful methods for solving unconstrained problems. Although they have a solid theoretical background, they are not efficient in practice since the sequence of unconstrained problems does not produce an exact solution [5]. The penalty parameter of the penalty function method must go to zero as you can see in Figure (2) (for more details, See [10, 11,18]). The penalty function method is summarized in Algorithm 1.

Algorithm 1: The Penalty Function Method
1 − $Given\ x^0, and\ \theta^0 > 0$
2 − $Find\ x^{k+1} such\ that$
$x^{k+1} = argmin\ Pq(x,\theta^k)$
3 − $Choose\ \theta^{k+1} \leq \theta^k$
4 − $Set\ k = k + 1\ and\ repeat.$

The Augmented Lagrangian Method (Multiplier Methods)

This technique became popular in the 1970s. Initially, it was referred to as the multiplier approach. This technique is now referred to as the augmented Lagrangian method. This strategy is intended to be used to tackle constrained optimization problems. This is accomplished by substituting a sequence of unconstrained issues for a constrained problem [11]. The enhanced Lagrangian approach is analogous to the penalty method in that both involve the addition of a penalty term to the aim. The enhanced Lagrangian technique differs simply in that it incorporates a Lagrange multiplier factor. Hestenes pioneered the augmented Lagrangian technique. To provide an overview of the enhanced Lagrangian approach, we can change the constraint $g_i(x) = 0$ of problem (2) to the constraint $g_i(x) + 2\theta\beta = 0$, as you can see in Figure (2, 3). Thus, we obtain the problem

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{subject to} && g_i(x) + 2\theta\beta = 0 \ , i = 1, \dots\dots, E \dots\dots\dots(7) \\ & && x \in X \end{aligned}$$

We may derive the augmented Lagrangian function as follows by applying the penalty technique to problem (6). We begin with the problem of the punishment for the problem.

$$argmin\, f(x) + \frac{1}{2\theta}(g(x) + 2\theta\beta\,)^T(g(x) + 2\theta\beta\,)$$

Which expands to

$$argmin\, f(x) + \frac{1}{2\theta}\ (\,g(x)^T g(x) + 4\theta\beta^T g(x) + 4\theta^2\beta^T\beta)$$

and simplifies to

$$argmin\, f(x) + 2\beta^T g(x) + \frac{1}{2\theta}\ \|g(x)\|^2$$

Therefore, the augmented Lagrangian function is

$$\mathcal{L}(x,\beta,\theta) = f(x) + 2\beta^T g(x) + \frac{1}{2\theta}\ \|g(x)\|^2$$

At each iteration, the multiplier approach updates an estimate of the Lagrange multiplier and, in some cases, the penalty parameter. Algorithm 2 summarizes the multiplier approach.

Algorithm 2: The Augmented Lagrangian Methods

1- Choose x^0, and $\theta^0 > 0$, choose β^0:

2- Find x^{k+1} such that

$$x^{k+1} = argmin\mathcal{L}(x,\beta^k,\theta^k):$$

3- Update $\beta^k$and θ^k:

4- Set $k = k + 1$ and repeat.

Numerical Result

The augmented Lagrangian approach was utilized in this article to create numerical results, which were then validated using the penalty method. A novel algorithm combining the augmented Lagrangian and the penalty methods was proposed, and the results appeared to be more beneficial than the results obtained using the two approaches alone. Comprehensive comparisons of the three numerical models' outcomes are made .See Figure (2,3,4) to compare the performance of the inequality augmented Lagrangian and penalty approaches. This experiment was conducted using a graph imported from the Biq Mac library. This graph contains 100 nodes and 2475 edges. The punishment system is more expedient. When the solution approaches the SDP bound, however, it is obvious that the enhanced Lagrangian method yields a faster convergence.As seen in Table(1), the inequality augmented Lagrangian technique arrived at the optimal solution more quickly than the penalty method.

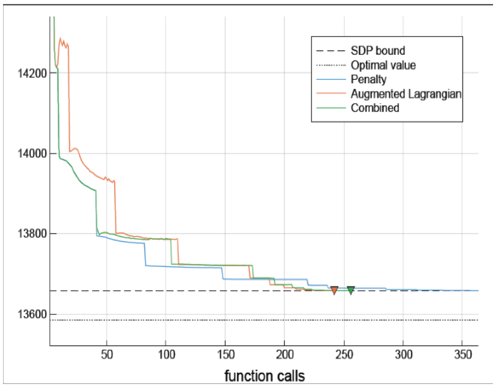

Figure 2. Combined Penalty and Augmented Lagrangian of Problem Pw01-100

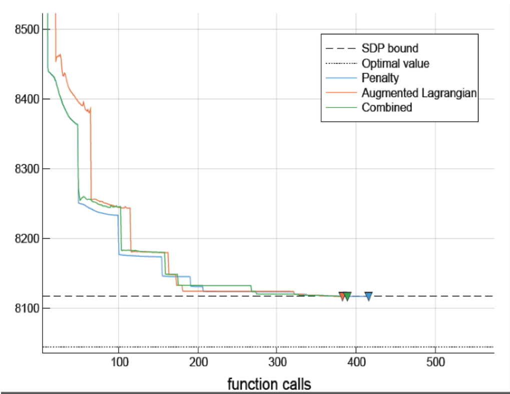

Figure 3. Combined Penalty and Augmented Lagrangian of Problem Pw05-100

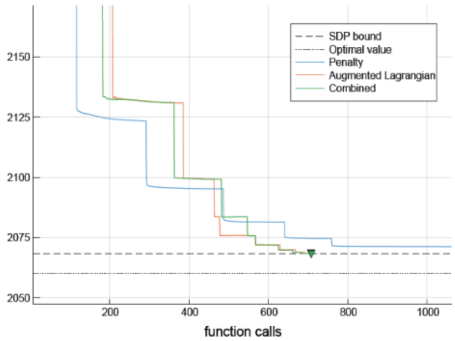

Figure 4. Combined Penalty and Augmented Lagrangian of Problem Pw09-100

Table 1

Our results					
Graphs	Biq Crunch Goal bound	Optimal value Biq Maclib	Penalty fcall	Aug fcall	Combined fcall
Pw01-100.0	2024.86045	2019	1228	650	549
Pw01-100.1	2068.33902	2060	1666	806	802
Pw01-100.2	2040.44551	2032	968	547	816
Pw01-100.3	2079.97438	2067	783	587	573
Pw01-100.4	2039.01451	2039	1572	1278	1253

Pw01-100.5	2108.05997	2108	1604	850	928
Pw01-100.6	2040.06693	2032	1036	866	768
Pw01-100.7	2082.95324	2074	1394	521	939
Pw01-100.8	2022.01521	2022	835	671	692
Pw01-100.9	2022.18848	2005	1049	463	555
Pw05-100.0	8284.62754	8190	489	345	313
Pw05-100.1	8117.05291	8045	460	431	428
Pw05-100.2	8116.48548	8039	588	811	580
Pw05-100.3	8189.39981	8139	703	367	391
Pw05-100.4	8201.37564	8125	460	298	375
Pw05-100.5	8224.13918	8169	544	513	357
Pw05-100.6	8308.47846	8217	714	377	441
Pw05-100.7	8313.06874	8249	509	657	391
Pw05-100.8	8230.34665	8199	811	427	888
Pw05-100.9	8160.27337	8099	659	509	637
Pw09-100.0	13657.78513	13585	426	293	301
Pw09-100.1	13494.91415	13417	489	331	471
Pw09-100.2	13519.47082	13461	604	336	360
Pw09-100.3	13706.17606	13656	784	342	387
Pw09-100.4	13575.50698	13514	576	375	338
Pw09-100.5	13639.05029	13574	583	647	402
Pw09-100.6	13703.98302	13640	600	342	331
Pw09-100.7	13579.47914	13501	461	380	317
Pw09-100.8	13650.86654	13593	1128	632	490
Pw09-100.9	13713.99428	13658	880	386	338

Conclusion

The paper's purpose was met, and the following points were discussed:

- We compared and contrasted two methods: the penalty technique and the Lagrange enhanced approach..
- We demonstrate the properties of hypothetical approximation and compare the two ways' algorithms.
- We also evaluated the approaches using the graphs included in the Big Mac library.

References

1. A. Fiacco and G. McCormick, Sequential unconstrained minimization techniques for nonlinear programming, A primal dual method. Munagement Sci, 10 (1968).
2. Angelo Miele, EE Cragg, RR Iyer, and AV Levy. Use of the augmented penalty function inmathematical programming problems, part 1. *Journal of Optimization Theory and Applications*, 8(2):115–130, 1971.
3. Andrzej Ruszczynski. *Nonlinear optimization*. Princeton university press, 2011.
4. Christian Kanzow, Daniel Steck, and Daniel Wachsmuth. An augmented lagrangian method for optimization problems in banach spaces. *SIAM Journal on Control and Optimization*, 56(1):272–291, 2018.
5. Dimitri P Bertsekas and Athena Scientific. *Convex optimization algorithms*. Athena Scientific Belmont, 2015.
6. D. P. Bertsekas, *Nonlinear programming*, Athena scienti_c Belmont, 1999.

7. Judah Ben Rosen, Olvi L Mangasarian, and Klaus Ritter. *Nonlinear Programming: Proceedings of a Symposium Conducted by the Mathematics Research Center, the University of Wisconsin, Madison, May 4-6, 1970*. Number 25. Elsevier, 2014.
8. JB Hiriart-Urruty and C Lemaréchal. Convex analysis and minimization algorithms. no. 305- 306 in grund. der math. wiss, 1993.
9. M. M. El-Alem, *A global convergence theory for a class of trust region algorithms for constrained optimization*, PhD thesis, Rice University, 1988.
10. Michael James Lowe. *Nonlinear programming: augmented Lagrangian techniques for constrained minimization*. PhD thesis, Montana State University-Bozeman, College of Engineering, 1974.
11. P Dimitri Bertsekas. Nonlinear programming. 1999. *Athena, Scientific, Belmont CA*.
12. Al-Jilawi, A. S., & Abd Alsharify, F. H. (2022). Review of Mathematical Modelling Techniques with Applications in Biosciences. Iraqi Journal For Computer Science and Mathematics, 3(1), 135-144.
13. Alridha, A., Wahbi, F. A., & Kadhim, M. K. (2021). Training analysis of optimization models in machine learning. International Journal of Nonlinear Analysis and Applications, 12(2), 1453-1461.
14. Kadhim, M. K., Wahbi, F. A., & Hasan Alridha, A. (2022). Mathematical optimization modeling for estimating the incidence of clinical diseases. International Journal of Nonlinear Analysis and Applications, 13(1), 185-195.
15. Alridha, A., Salman, A. M., & Al-Jilawi, A. S. (2021, March). The Applications of NP-hardness optimizations problem. In Journal of Physics: Conference Series (Vol. 1818, No. 1, p. 012179). IOP Publishing.
16. Salman, A. M., Alridha, A., & Hussain, A. H. (2021, March). Some Topics on Convex Optimization. In Journal of Physics: Conference Series (Vol. 1818, No. 1, p. 012171). IOP Publishing.
17. Alridha, A., & Al-Jilawi, A. S. (2021, March). Mathematical Programming Computational for Solving NP-Hardness Problem. In Journal of Physics: Conference Series (Vol. 1818, No. 1, p. 012137). IOP Publishing.
18. Alridha, A. H., & Al-Jilawi, A. S. (2022). Solving NP-hard problem using a new relaxation of approximate methods. International Journal of Health Sciences, 6(S3), 523–536. https://doi.org/10.53730/ijhs.v6nS3.5375