International Journal of Mathematics and Computer Science
Volume **20,** Issue no. 2, (2025), 521–527
DOI: https://doi.org/10.69793/ijmcs/02.2025/zeina

Modern Nonlinear Optimization Algorithms by Using Approximate Methods for Combinatorial Numerical Analysis

Zeina Saleh Mehdi, Ahmed Sabah Ahmed Al-jilawi

Department of Mathematics
Faculty of Education College for Pure Sciences
Babylon University
Babylon, Iraq

email: edu770.zena.salah@student.uobabylon.edu.iq
aljelawy2000@yahoo.com

(Received January 3, 2025, Accepted January 29, 2025, Published February 4, 2025)

Abstract

Large-scale optimization has become an important research theme in the evolutionary computing field. To take advantage of this progress, sophisticated algorithms have been merged in commercially inexpensive software, like python, to increase the aptitude of large data handling. In this paper, we propose an advanced algorithm, to compare Augmented and Penalty methods for resolving large-scale constrained problems of optimization. We highlight the advantages of an Augmented method in terms of faster convergence, better numerical stability and more robust performance while noting the behavior of Penalty Methods like parameter sensitivity issues, degraded performance near optimal and numerical instabilities. Using numerical experiments, our algorithm is quickly convergent and is more reliable.

1 Introduction

Optimization is a process to access the best objective under specific constraints [1]. Optimization techniques are vastly utilized in all engineering

Key words and phrases: Optimization, Numerical Optimization, Approximate Method, Augmented Lagrange Method, Large-Scale Optimization.
AMS (MOS) Subject Classifications: 65-xx.
ISSN 1814-0432, 2025, https://future-in-tech.net

fields to pick the best design. Assume the aggregate numeral of factors is n [2]. The optimization theme of a single-objective may be characterized as follows [3]:
Select factors from an n-dimensional vector $x = [x_1, x_2, \ldots, x_n]^T$

$$\begin{aligned} &\text{minimize} && f(x_1, x_2, \ldots, x_n) \\ &\text{subject to} && y_j(x) \leq 0, j = 1, 2, \ldots, p_1 \\ & && g_u(x) = 0, u = 1, 2, \ldots, p_2. \end{aligned} \tag{1.1}$$

where $f(x) = f(x_1, x_2, \ldots, x_n)$ is an objective function, each $y_j(x) \leq 0$ is the inequality constraint and $g_u(x) = 0$ is the equality constraint, $(p_2 < n)$ within the factor range $x_i^L \leq x_i \leq x_i^U$, $i = 1, 2, \ldots, n$, with x_i^L and x_i^U denoting the lower and upper confine of a factor x_i [4].

2 Mathematical Model of Augmented Lagrange Method (ALM)

General Augmented Lagrange multipliers have been presented to address optimization problems constraints of the type [5]:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && h(x) = 0 \end{aligned} \tag{2.2}$$

where $f : R^n \longrightarrow R$ and $h : R^n \longrightarrow R^m$. An Augmented Lagrange Function can be defined as follows [6]:
$L(x, \lambda, \rho) = f(x) + \langle \lambda, h(x) \rangle + \frac{\rho}{2} \parallel h(x) \parallel_F^2$.
When the scalar ρ is positive, the Augmented Lagrange Multiplier can be utilized to solve optimization problems. When ρ_m is an increasing sequence and both f and h are continuously differentiable functions, the ALM determines an ideal step size for updating λ_m depending on the penalty parameter ρ_m.

Algorithm 1: Comparison of Augmented Lagrange and Penalty Methods

1. Problem Definition
Consider an n-dimensional convex objective function:
$f(x) = \sum(x_i^2) + \sum(sin(x_i)) + exp((mean(x))$
2. Augmented Lagrange Method Algorithm
Initialization: Input: Initial point $x_0 \in R^n$, parameter $\mu = 1$, $\lambda = 0$,$max - iter$, factor $\beta = 1.5$
1. For $k = 1$ to $max - iter$:
a. Solve the sub-problem: $minimize L_A(x) = f(x) + \lambda^T x + (\frac{\mu}{2}) \parallel x \parallel^2$ where $f(x)$ is the original objective function, $\lambda^T x$ is the Lagrangian term, $(\frac{\mu}{2}) \parallel x \parallel^2$ is the quadratic penalty term
b. Update Lagrange multipliers: $\lambda_{k+1} = \lambda_k + \mu_x$
c. Update the penalty parameter: $\mu_{k+1} = \beta_{\mu_k}$
d. Store iteration results: Current iteration k, Objective value $f(x)$, Current solution x
3. Penalty Method Algorithm
Initialization: Input: Initial point $x_0 \in R^n$, parameter $\rho = 1$, $max - iter$, factor $\gamma = 2$
1. For $k = 1$ to $max - iter$:
a. Solve the sub-problem: $minimize\ p(x) = f(x) + \rho \parallel x \parallel^2$ where $f(x)$is the original objective function, $\rho \parallel x \parallel^2$ is the penalty term
b. Update the penalty parameter $\rho_{k+1} = \gamma\rho_k$
c. Store iteration results: Current iteration k, objective value $f(x)$, current solution x

The algorithm presents two related but distinct optimization methods: ALM and Penalty method. The given objective function has three terms: A quadratic term $\sum(x_i^2)$ that grows quickly as variables move from zero, a term $\sum(sin(x_i))$ that adds periodic behavior, and an exponential term $exp(mean(x))$ that grows very quickly with the average of the variables. As ALM combines aspects of the Penalty and Lagrange Methods, it introduces Lagrange Multipliers λ to handle constraints, uses a quadratic penalty term $(\frac{\mu}{2}) \parallel x \parallel^2$ that helps with convergence, updates both the penalty parameter μ and the Lagrange Multipliers λ iteratively and the Penalty Parameter increases by a factor $\beta = 1.5$ at each iteration. The Penalty Method only uses $\rho \parallel x \parallel^2$ without Lagrange multipliers, increases the Penalty Parameter ρ aggressively (factor $\gamma = 2$) and generally requires more iterations but is easier to implement. ALM typically converges faster due to the Lagrange multiplier terms and provides better constraint due to the dual updates.

Table 1: The numerical results of comparison of Augmented Lagrange and Penalty Methods

i	**Aug.Obj. value**	**Solution**	**Pen.Obj. value**	**Solution**
1	-0.41335	[-0.3587 -0.3587 -0.3587 -0.3587 -0.3587]	-0.22911	[-0.2782 -0.2782 -0.2782 -0.2782 -0.2782]
2	-0.05352	[-0.222 -0.222 -0.222 -0.222 -0.222]	0.05870	[-0.1912 -0.1912 -0.1912 -0.1912 -0.1912]
3	0.39285	[-0.1131 -0.1131 -0.1131 -0.1131 -0.1131]	0.37387	[-0.1171 -0.1171 -0.1171 -0.1171 -0.1171]
4	0.73901	[-0.0454 -0.0454 -0.0454 -0.0454 -0.0454]	0.62900	[-0.0658 -0.0658 -0.0658 -0.0658 -0.0658]
5	0.91792	[-0.0139 -0.0139 -0.0139 -0.0139 -0.0139]	0.79635	[-0.0351 -0.0351 -0.0351 -0.0351 -0.0351]
6	0.98133	[-0.0031 -0.0031 -0.0031 -0.0031 -0.0031]	0.89306	[-0.0181 -0.0181 -0.0181 -0.0181 -0.0181]
7	0.99696	[-0.0005 -0.0005 -0.0005 -0.0005 -0.0005]	0.94517	[-0.0092 -0.0092 -0.0092 -0.0092 -0.0092]
8	0.99965	[-0.0001 -0.0001 -0.0001 -0.0001 -0.0001]	0.97223	[-0.0046 -0.0046 -0.0046 -0.0046 -0.0046]
9	0.99997	[-0. -0. -0. -0. -0.]	0.98602	[-0.0023 -0.0023 -0.0023 -0.0023 -0.0023]
10	0.99999	[-0. -0. -0. -0. -0.]	0.99299	[-0.0012 -0.0012 -0.0012 -0.0012 -0.0012]
11	1.00000	[-0. -0. -0. -0. -0.]	0.99649	[-0.0006 -0.0006 -0.0006 -0.0006 -0.0006]
12	1.00000	[-0. -0. -0. -0. -0.]	0.99824	[-0.0003 -0.0003 -0.0003 -0.0003 -0.0003]
13	1.00000	[-0. -0. -0. -0. -0.]	0.99912	[-0.0001 -0.0001 -0.0001 -0.0001 -0.0001]
14	1.00000	[-0. -0. -0. -0. -0.]	0.99956	[-0.0001 -0.0001 -0.0001 -0.0001 -0.0001]
15	1.00000	[-0. -0. -0. -0. -0.]	0.99978	[-0. -0. -0. -0. -0.]
16	1.00000	[0. 0. 0. 0. 0.]	0.99989	[-0. -0. -0. -0. -0.]
17	1.00000	[0. 0. 0. 0. 0.]	0.99994	[-0. -0. -0. -0. -0.]
18	1.00000	[-0. -0. -0. -0. -0.]	0.99997	[-0. -0. -0. -0. -0.]
19	1.00000	[-0. -0. -0. -0. -0.]	0.99998	[-0. -0. -0. -0. -0.]
20	1.00000	[-0. -0. -0. -0. -0.]	0.99999	[-0. -0. -0. -0. -0.]

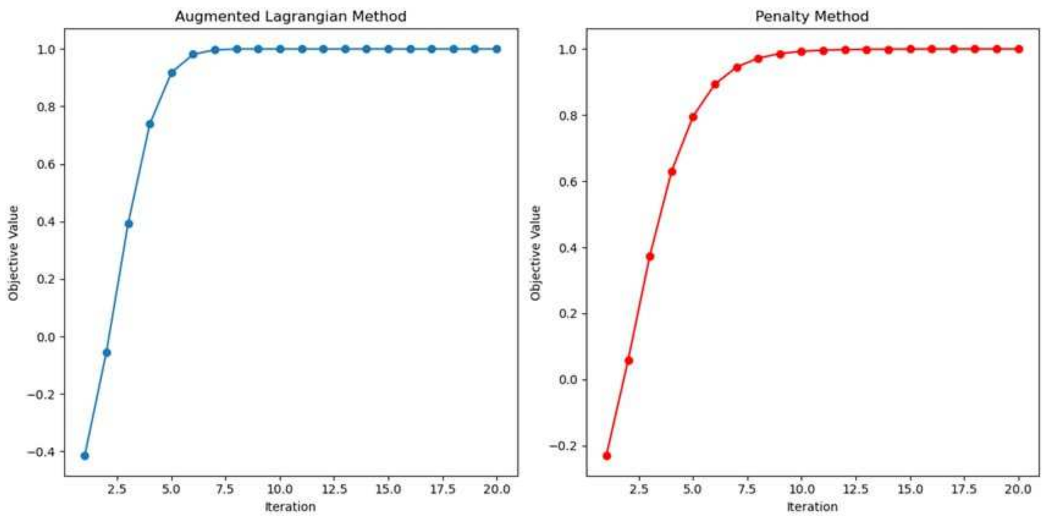

Figure 1: Comparison of Augmented Lagrange and Penalty Methods

These tables compare the performance of ALM and the Penalty method over 20 iterations each. As for convergence speed that ALM reaches the optimal objective value of 1.0 by iteration 11 and penalty method approaches but doesn't quite reach 1.0, getting to 0.999993 by iteration 20 then ALM shows faster convergence overall while initial behavior that ALM starts at a lower objective value -0.413359 but recovers more quickly and penalty method starts from a higher point -0.229118 but progresses more gradually then numerical stability that ALM stabilizes completely after iteration 11 and penalty method continues making very small improvements until the final iteration. This comparison explains that ALM is more efficient for this particular optimization problem, achieving better convergence in fewer iterations. This plot compares two optimization methods the Augmented Lagrange Method left in blue and the Penalty method right in red,both graphs show the convergence behavior of these methods over 20 iterations.The y-axis represents the objective value ranging from about-0.4 to 1.0,while the x-axis shows the number of iterations. Both methods start with negative objective values around -0.2 to -0.4 and show rapid initial improvement in the first 5-7 iterations then eventually converge to a value close to 1. ALM appears to reach its final value slightly faster,showing steeper initial improvement and the penalty method's curve is slightly smoother in its approach to convergence.

3 Conclusion

We have suggested, tested and analyzed an advanced algorithm which is include comparison between two approximate methods,which are Augmented and penalty, about it is a strategy for converting constrained problems into unconstrained ones.We have proved that ALM demonstrates slightly faster convergence and typically offers better numerical stability,making it particularly valuable for real-world applications like engineering design optimization,training neural networks with constraints and resource allocation in manufacturing while both methods can handle large-scale problems, ALM performs better due to its strong theoretical convergence properties and also it can handle equality and inequality constraints effectively and generally more robust than simple penalty methods may have difficulty meeting strict constraints and slower convergence for complex problems.Based on the results we obtained ALM showing generally faster convergence.

References

[1] Ahmed Sabah Ahmed Al-Jilawi, ”Modern Free-Derivative Numerical Optimization of Approximate Algorithms Convergence and Neutrosophic Convergence,” International Journal of Neutrosophic Science, **25,** no. 3, (2025), 115–122.

[2] Giuseppe C. Calafiore, Laurent El Ghaoui, Optimization models, Cambridge University Press, 2014.

[3] Ameer Dehyauldeen A. Al-Zamili, Ahmed Sabah Ahmed Aljilawi, ”Optimizing Logistic Regression with Enhanced Convergence: A Modified ADMM Approach,” Computer Science, **20,** no. 1, (2025), 9–15.

[4] John M. Mulvey, Robert J. Vanderbei, Stavros A. Zenios, ”Robust optimization of large-scale systems,” Operations research, **43,** no. 2, (1995), 264–281.

[5] Ameer Dehyauldeen A. Al-Zamili, Ahmed Sabah Ahmed Aljilawi, "Using Alternating Direction Method of Multipliers to solve optimization problem in Statistics," Computer Science, **19,** no. 1, (2024), 157–162.

[6] J.D. Lee et al., Practical large-scale optimization for max-norm regularization, Advances in Neural Information Processing Systems, (2010), 1297–1305.