Question 2 – Option A: Constrained Optimization Using the Quadratic Penalty Method

Team Members:

BT2024066, BT2024144, BT2024166

1. Introduction

In this question, we formulate and solve a real-world constrained logistics optimization problem using the Quadratic Penalty Method, in which a company distributes goods using three transportation routes. Each route has an economically preferred operating level, a maximum capacity, and the company must meet a fixed daily demand.

The goal is to determine the shipment allocation that minimizes cost while respecting all constraints. To make the model flexible and realistic, all key problem parameters are provided by the user.

2. Problem Formulation

Let x_1, x_2, x_3 denote daily shipments through the three routes. The user specifies:

$$t_i = \text{preferred shipment level}, \quad w_i = \text{cost weight}, \quad c_i = \text{capacity}, \quad D = \text{daily demand}.$$

2.1 Objective Function

The cost increases quadratically when actual shipments deviate from preferred values:

$$\min_x f(x) = w_1(x_1 - t_1)^2 + w_2(x_2 - t_2)^2 + w_3(x_3 - t_3)^2$$

2.2 Constraints

Equality constraint (demand):

$$h(x) = x_1 + x_2 + x_3 - D = 0$$

Inequality constraints (capacities and non-negativity):

$$g_1 = x_1 - c_1, \quad g_2 = x_2 - c_2, \quad g_3 = x_3 - c_3$$

$$g_4 = -x_1, \quad g_5 = -x_2, \quad g_6 = -x_3$$

Thus, the constrained problem is:

$$\min_x f(x) \quad \text{s.t. } g_i(x) \le 0,\ i = 1, \dots, 6, \quad h(x) = 0.$$

2.3 User Input Feature

To ensure the model can be applied to multiple real-world logistics settings, the following values are entered by the user at runtime:

- **Preferred operating levels** t_1, t_2, t_3**:** These represent the economically optimal shipment levels for each route; deviations from these preferred levels increase operational cost.
- **Cost weights** w_1, w_2, w_3**:** These determine the severity of the penalty applied when shipments deviate from the preferred operating levels on each route.
- **Capacity limits** c_1, c_2, c_3**:** These specify the maximum allowable shipment volumes for each route due to physical or operational constraints.
- **Total daily demand** D**:** This is the required number of shipments that must be fulfilled collectively across all routes.
- **Initial starting point** $x_0 = (x_{1,0}, x_{2,0}, x_{3,0})$**:** This is the point from which the optimization algorithm begins its iterative search for the optimal solution.

The optimization algorithm then computes the optimal feasible shipment distribution that satisfies all constraints while minimizing total cost.

3. Quadratic Penalty Method

To solve the constrained problem, we convert it into a sequence of unconstrained penalized problems:

3.1 Penalized Objective

$$\Phi(x;\mu) = f(x) + \frac{\mu}{2}h(x)^2 + \frac{\mu}{2}\sum_{i=1}^{6} \max(0, g_i(x))^2$$

3.2 Gradient

$$\nabla\Phi(x) = \nabla f(x) + \mu h(x)\nabla h(x) + \mu \sum_{i:g_i(x)>0} g_i(x)\nabla g_i(x)$$

3.3 Hessian

$$\nabla^2\Phi(x) = \nabla^2 f + \mu \nabla h \nabla h^T + \mu \sum_{i:g_i(x)>0} \nabla g_i \nabla g_i^T$$

3.4 Newton Step for Penalized Subproblem

$$x_{k+1} = x_k - \left(\nabla^2\Phi(x_k;\mu)\right)^{-1} \nabla\Phi(x_k;\mu)$$

3.5 Penalty Parameter Update

$$\mu_{k+1} = \beta\mu_k, \qquad \beta > 1$$

3.6 Stopping Criteria

$$|h(x)| \leq 10^{-3}, \qquad \max(0, g_i(x)) \leq 10^{-3}$$

4. Numerical Results

For user-specified parameters, the algorithm generates iterates:

$$x^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)})$$

After several penalty updates, the solution converges to a feasible point $x^\star$ such that:

$$h(x^\star) \approx 0, \qquad g_i(x^\star) \leq 0$$

The solution lies close to the preferred shipment levels while satisfying demand and capacity constraints.

5. Interpretation and Trade-offs

- The penalty term forces feasibility as μ increases.
- Extremely large μ may lead to numerical instability.
- The trade-off between cost minimization and feasibility becomes tighter with higher penalties.
- The final solution balances operational preference with real-world limitations.

6. Conclusion

We solved a real-world constrained logistics optimization problem using the Quadratic Penalty Method. The solution respects all operational constraints and adjusts shipments minimally from their preferred levels. The iterative penalty updates progressively enforced feasibility, demonstrating how the method effectively balances constraint satisfaction with cost minimization.

7. Example Used for Demonstration

The following user inputs were used to generate the convergence plots and results:

- Preferred shipment levels: $t_1 = 1,\ t_2 = 2,\ t_3 = 3$
- Cost weights: $w_1 = 10,\ w_2 = 20,\ w_3 = 30$
- Route capacities: $c_1 = 3,\ c_2 = 3,\ c_3 = 3$
- Daily demand: $D = 5$
- Initial guess: $(x_1, x_2, x_3) = (5, 5, 5)$

8. Convergence Plots and Program Output

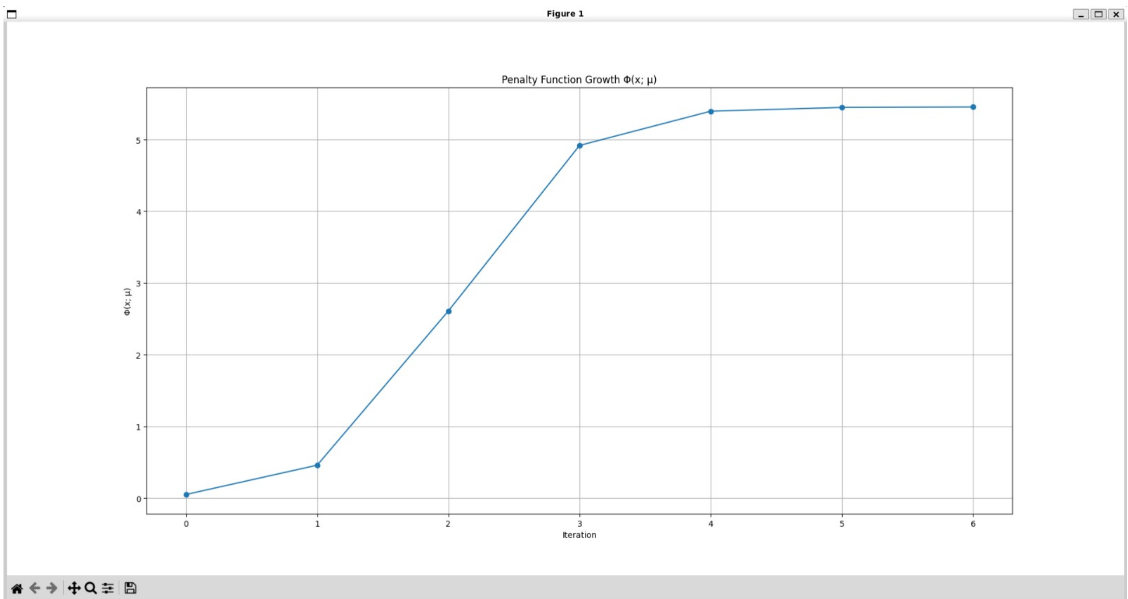


Figure 1: Penalty Function Growth $\Phi(x; \mu)$ Across Iterations

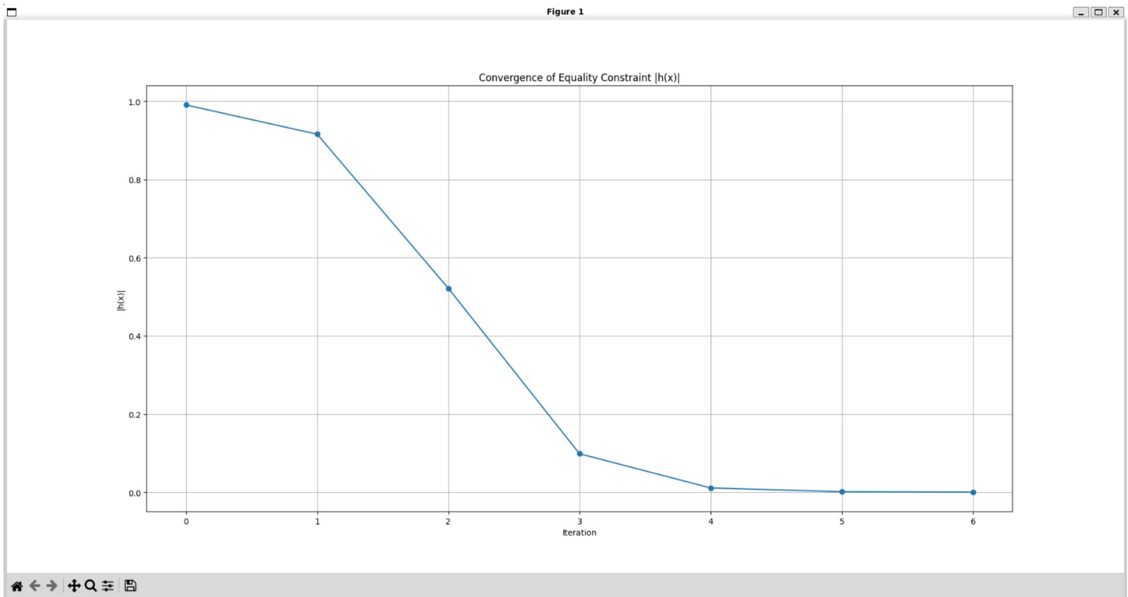
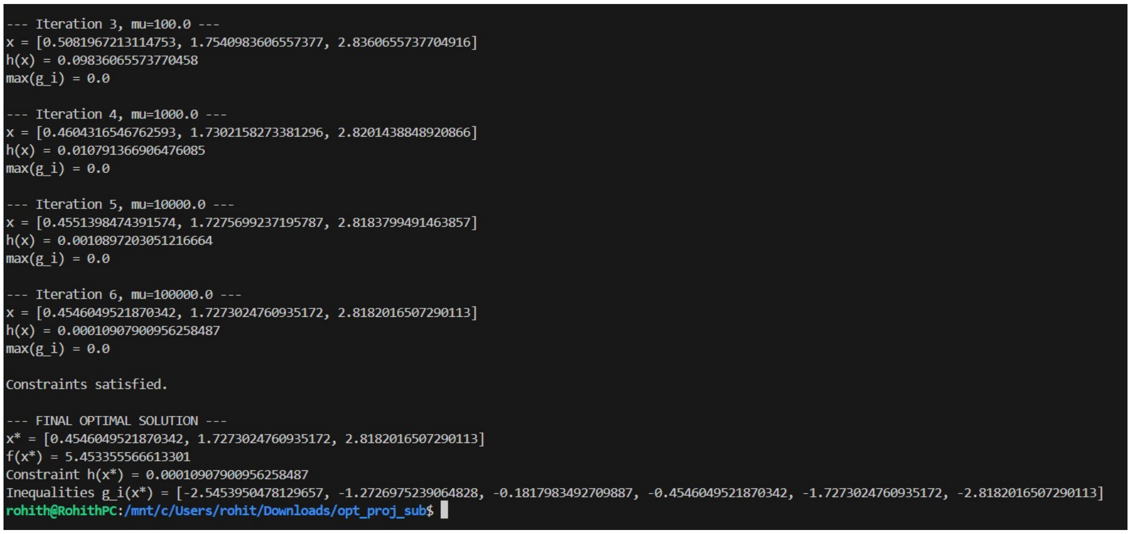


Figure 2: Equality Constraint Convergence $|h(x)|$

```
rohith@RohithPC:/mnt/c/Users/rohit/Downloads/opt_proj_sub$ python3 question2a.py
Enter preferred shipment levels:
t1: 1
t2: 2
t3: 3

Enter cost weights:
w1: 10
w2: 20
w3: 30

Enter route capacities:
Capacity c1: 3
Capacity c2: 3
Capacity c3: 3

Enter daily demand D: 5

Enter initial guess (x1, x2, x3):
x1_0: 5
x2_0: 5
x3_0: 5

=== CONSTRAINED PENALTY METHOD SOLUTION ===

--- Iteration 0, mu=0.1 ---
x = [0.9950454170107349, 1.9975227085053675, 2.998348472336912]
h(x) = 0.9909165978530137
max(g_i) = 0.0

--- Iteration 1, mu=1.0 ---
x = [0.9541984732824428, 1.9770992366412214, 2.984732824427481]
h(x) = 0.9160305343511457
max(g_i) = 0.0

--- Iteration 2, mu=10.0 ---
x = [0.7391304347826085, 1.8695652173913042, 2.9130434782608696]
h(x) = 0.5217391304347823
max(g_i) = 0.0
```

Figure 3: Program Output – Part 1



```
--- Iteration 3, mu=100.0 ---
x = [0.5081967213114753, 1.7540983606557377, 2.8360655737704916]
h(x) = 0.09836065573770458
max(g_i) = 0.0

--- Iteration 4, mu=1000.0 ---
x = [0.4604316546762593, 1.7302158273381296, 2.8201438848920866]
h(x) = 0.010791366906476085
max(g_i) = 0.0

--- Iteration 5, mu=10000.0 ---
x = [0.4551398474391574, 1.7275699237195787, 2.8183799491463857]
h(x) = 0.0010897203051216664
max(g_i) = 0.0

--- Iteration 6, mu=100000.0 ---
x = [0.4546049521870342, 1.7273024760935172, 2.8182016507290113]
h(x) = 0.00010907900956258487
max(g_i) = 0.0

Constraints satisfied.

--- FINAL OPTIMAL SOLUTION ---
x* = [0.4546049521870342, 1.7273024760935172, 2.8182016507290113]
f(x*) = 5.453355566613301
Constraint h(x*) = 0.00010907900956258487
Inequalities g_i(x*) = [-2.5453950478129657, -1.2726975239064828, -0.1817983492709887, -0.4546049521870342, -1.7273024760935172, -2.8182016507290113]
rohith@RohithPC:/mnt/c/Users/rohit/Downloads/opt_proj_sub$
```

Figure 4: Program Output – Part 2