Assignment 7 of MATP4820

Xinshi Wang 661975305

Problem 1

Consider the one-dimensional minimization problem

$$\min_x 2x - 1, \text{ s.t. } x \leq 2,\ x + 1 \geq 0 \tag{1}$$

1. Formulate the quadratic penalty problem with a penalty parameter $\mu > 0$ by penalizing **both** inequality constraints.
   $\underset{x}{min} \quad 2x - 1 + \frac{\mu}{2}((x-2)_+^2 + (-x-1)_+^2)$

2. Derive the optimal solution of the quadratic penalty problem. Note that the solution should depend on μ.
   Gradient is equal to $2 + \mu(x-2)_+ + \mu(-x-1)_+$

   (a) if $x < -1$, gradient $= 2 - \mu(x+1)$
   Setting gradient to 0 gives us $2 - \mu x - \mu = 0 \implies x = \frac{2}{\mu} - 1$

   (b) if $-1 \leq x \leq 2$, gradient $= 2$
   This is a invalid range as gradient can never be 0.

   (c) if $x > 2$, gradient $= 2 + \mu(x-2)$
   Setting gradient to 0 gives us $2 + \mu x - 2\mu = 0 \implies x = 2 - \frac{2}{\mu}$

3. Prove that the solution of the quadratic penalty problem converges to the optimal solution $x^* = -1$ of Problem (1).

   As $\mu \to \infty$, we have $\frac{2}{\mu} - 1 \to -1$ for the first case and $2 - \frac{2}{\mu} \to 2$ for the second case. Since -1 is the lover objective value, we have $x^* = -1$ as the optimal value.

Problem 2

Consider the problem

$$\min_{x,y\in\mathbb{R}} f(x,y) = -\frac{1}{2}x^2 - xy + \frac{1}{2}y^2 - 2x - 2y, \text{ s.t. } x + y = 1.$$

1. Argue that $(1,0)$ is the unique minimizer.

   To find the unique minimizer, we can rewrite the constraint as $y = 1-x$ and substitute it into the function:

   $f(x, 1-x) = -\frac{1}{2}x^2 - x(1-x) + \frac{1}{2}(1-x)^2 - 2x - 2(1-x)$

   Now, we can find the critical points by differentiating with respect to x and setting the derivative to zero:

   $\frac{df}{dx} = -x - (1-2x) + (1-x) - 2 + 2 = 0$

   Solving for x, we get $x = 1$. Substituting back into the constraint, we get $y = 0$. Thus, the unique minimizer is $(1,0)$.

2. The augmented Lagrangian function with penalty parameter $\beta > 0$ is

   $$\mathcal{L}(x,y,v) = -\frac{1}{2}x^2 - xy + \frac{1}{2}y^2 - 2x - 2y + v(x+y-1) + \frac{\beta}{2}(x+y-1)^2.$$

   Argue that for any $\beta \in (0,1]$, the augmented Lagrangian method with initial multiplier $v^{(0)} = 0$ does not work for this instance, i.e., the generated sequence $(x^{(k)}, y^{(k)})$ does not converge to a KKT point.
   The gradient of the augmented Lagrangian function with respect to x and y is:

   $$\partial_x\mathcal{L} = -x - y - 2 + v^{(k-1)} + \beta(x+y-1)$$
   $$\partial_y\mathcal{L} = -x + y - 2 + v^{(k-1)} + \beta(x+y-1)$$

   Setting the gradient to zero and solving the system of linear equations, we get:

   $$y = 0$$
   $$-x - 2 + v^{(k-1)} + \beta(x-1) = 0$$

   Solving this system, we find:

   $$x^{(k)} = \frac{\beta + 2 - v^{(k-1)}}{\beta - 1}$$
   $$y^{(k)} = 0$$

If $x^{(k)}$ converges to 1, we need to have $v^{(k-1)}$ converges to 3. The update formula for $v^{(k)}$ is $v^{(k)} = v^{(k-1)} + \beta(\frac{\beta+2-v^{(k-1)}}{\beta-1})$ If we start from $v^{(0)} = 0$ and $\beta \in (0,1]$, we have $v^{(1)} < 0$ as the numerator is positive and the denominator is negative and $v^0 = 0$. The numerator is always positive and denominator is always negative in this case. Therefore the dangling $\beta(\frac{\beta+2-v^{(k-1)}}{\beta-1})$ term is always negative. Therefore $v^{(k)}$ would always be negative and thus not convering to 3. Therefore we would not have $x^{(k)}$ converging to 1 and thus not reaching the KKT point.

3. For $\beta = 2$, do by hand two iterations of the augmented Lagrangian method with initial multiplier $v^{(0)} = 0$.

   The augmented Lagrangian function for $\beta = 2$ is:

   $\mathcal{L}(x, y, v) = -\frac{1}{2}x^2 - xy + \frac{1}{2}y^2 - 2x - 2y + v(x + y - 1) + (x + y - 1)^2$

   First iteration:

   $v^{(0)} = 0$

   $\frac{\partial \mathcal{L}}{\partial x} = -x - y - 2 + v + 2(x + y - 1)$
   $\frac{\partial \mathcal{L}}{\partial y} = -x + y - 2 + v + 2(x + y - 1)$

   Setting the partial derivatives to zero:

   $-x - y - 2 + 2(x + y - 1) = 0$
   $-x + y - 2 + 2(x + y - 1) = 0$

   Solving the system of equations, we get $x^{(1)} = 4$ and $y^{(1)} = 0$.

   Now, we update the Lagrange multiplier:

   $v^{(1)} = v^{(0)} + \beta(x^{(1)} + y^{(1)} - 1) = 0 + 2(4 + 0 - 1) = 6$

   Second iteration:

   $v^{(1)} = 6$

   $\frac{\partial \mathcal{L}}{\partial x} = -x - y - 2 + v + 2(x + y - 1)$
   $\frac{\partial \mathcal{L}}{\partial y} = -x + y - 2 + v + 2(x + y - 1)$

   Setting the partial derivatives to zero:

   $-x - y + 4 + 2(x + y - 1) = 0$
   $-x + y + 4 + 2(x + y - 1) = 0$

   Solving the system of equations, we get $x^{(2)} = -2$ and $y^{(2)} = 0$.

   The Lagrange multiplier remains unchanged:

   $v^{(2)} = v^{(1)} + \beta(x^{(2)} + y^{(2)} - 1) = 6 + 2(-2 + 0 - 1) = 0$

   After two iterations, we have $(x^{(2)}, y^{(2)}) = (-2, 0)$

4. (**bonus question**) For any $\beta > 2$, does the generated sequence $(x^{(k)}, y^{(k)})$ converge to the unique minimizer $(1, 0)$, starting from any $v^{(0)}$? If yes, prove it; if not, give a value of $\beta > 2$ and $v^{(0)}$ such that the sequence does not converge to $(1, 0)$.

Problem 3

Consider the nonnegative quadratic program:

$$\underset{\mathbf{x}\in X}{\text{minimize}}\, f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} - \mathbf{c}^\top \mathbf{x}, \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b} \tag{2}$$

where $\mathbf{Q}$ is a symmetric and positive semidefinite matrix, and $X = \{\mathbf{x} \in \mathbb{R}^n : x_i \geq 0, i = 1, \ldots, n\}$.

Use the instructor's provided file `alm_qp.m` to write a Matlab function `alm_qp` with input $\mathbf{Q}, \mathbf{c}, \mathbf{A}$, $\mathbf{b}$, initial vector $\mathbf{x}0$, stopping tolerance `tol`, and penalty parameter $\beta > 0$. Also test your function by running the provided test file `test_alm_qp.m` and compare to the instructor's function. Print your code and the results you get.

```
function [x, hist_obj, hist_res] = alm_qp(Q, c, A, b, tol, beta, x0)
% augmented Lagrangian method for the quadratic programming
% min_x 0.5*x'*Q*x - c'*x
% s.t.  x >= 0, A*x == b

[m, n] = size(A);

% initialization
v = zeros(m, 1);

x = x0;

% compute the residual for the constraint A*x == b
r = A * x - b;

res = norm(r);
grad_err = 1;
hist_res = res;
hist_obj = 0.5 * x' * Q * x - c' * x;

% use constant stepsize
alpha = 1 / norm(Q + beta * A' * A);

while res > tol || grad_err > tol
    % compute the gradient
    grad = Q * x - c + A' * (v + beta * r);
```

```
% compute violation of optimality condition
grad_err = 0;
for i = 1:500
    if x(i) == 0
        grad_err = grad_err + max(0, - grad(i));
    else
        grad_err = grad_err + abs(grad(i));
    end
end

% update x
while grad_err > tol
    x = x - alpha * grad;
    x = max(0,x);
    r = A * x - b;
    % compute the gradient
    grad = Q * x - c + A' * (v + beta * r);

    % compute violation of optimality condition
    grad_err = 0;

    for i = 1:500
        if x(i) == 0
            grad_err = grad_err + max(0, - grad(i));
        else
            grad_err = grad_err + abs(grad(i));
        end
    end
end

% compute the residual
r = A * x - b;
res = norm(r);
obj = 0.5 * x' * Q * x - c' * x;

% save res and obj
```

```
    hist_res = [hist_res; res];
    hist_obj = [hist_obj; obj];

    % update multiplier
    v = v + beta * r;
end
end
```

Student solver: Total running time of instructor code is 4.1239

Instructor solver: Total running time of instructor code is 4.1005