Constrained Optimization: Implementation of a Penalty Method

Adrià Fenoy
February 7, 2019

1 PENALTY METHOD IMPLEMENTATION

Given the constrained problem

$$\begin{gathered}\min f(\mathbf{x}) \\ s.t.\;\; g_i(\mathbf{x}) \leq 0 \;\; \forall i \in I\end{gathered} \tag{1.1}$$

can be transformed into an unconstrained form [1]

$$\min\big(f(\mathbf{x}) + c \cdot p(\mathbf{x})\big) \quad , \tag{1.2}$$

where $p(\mathbf{x})$ is a penalty function and $c > 0$ is a constant. $p(\mathbf{x})$ must be continuous and positive. A penalty function that satisfies this, and it is commonly used [2] is

$$p(\mathbf{x}) = \sum_i \big(\max[0, g_i(\mathbf{x})]\big)^2 \quad . \tag{1.3}$$

The algorithm to find the minimum of (1.1) is based on iterating over the unconstrained formula

$$\mathbf{x}_{k+1} = \min\big(f(\mathbf{x}_k) + c_k \cdot p(\mathbf{x}_k)\big) \quad , \tag{1.4}$$

where c_k is obtained with a recursive formula like

$$c_{k+1} = \eta c_k \quad , \tag{1.5}$$

where $\eta \geq 0$ is the rate c_k is being updated.

The min in the recursive formula (1.4) can be computed using any method for constrained optimization. In this implementation, it has been used a Quasi-Newton Method [3], which consists in computing the gradient vector and the hessian matrix using the expressions:

$$\nabla_{(x_0,y_0)} q(x,y) \approx \begin{bmatrix} \frac{q(x_0+h,y_0)-q(x_0,y_0)}{h} \\ \\ \frac{q(x_0,y_0+h)-q(x_0,y_0)}{h} \end{bmatrix} \quad , \tag{1.6}$$

$$\mathbf{H}_{(x_0,y_0)} q(x,y) \approx \begin{bmatrix} \frac{q(x_0+2h,y_0)-2q(x_0+h,y_0)+q(x_0,y_0)}{h^2} & \frac{q(x_0+h,y_0+h)-q(x_0+h,y_0)-q(x_0,y_0+h)+q(x_0,y_0)}{h^2} \\ \\ \frac{q(x_0+h,y_0+h)-q(x_0+h,y_0)-q(x_0,y_0+h)+q(x_0,y_0)}{h^2} & \frac{q(x_0,y_0+2h)-2q(x_0,y_0+h)+q(x_0,y_0)}{h^2} \end{bmatrix} \quad , \tag{1.7}$$

where h represents the proximity of the points used to compute the gradient, and $q(\mathbf{x}) = f(\mathbf{x}) + c \cdot p(\mathbf{x})$. In this case, the update rule is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_{x_k} q(x,y) \cdot \nabla_{x_k} q(x,y) \quad . \tag{1.8}$$

2 Algorithm

The Penalty Method algorithm performs like follows:

1. Input $f(\mathbf{x})$ and all $g_i(\mathbf{x})$.
2. Define hyper-parameters h, ϵ (used to determine the stopping criteria) and η, as well as define the initial point $x_{opt} \leftarrow (x_0, y_0)$.
3. Compute $q(\mathbf{x}) = f(\mathbf{x}) + c \cdot p(\mathbf{x})$.
4. Store the current optimal point $x_{old} \leftarrow x_{opt}$ and compute the new optimal point with (1.4) using Quasi-Newton Method.
5. Update η with (1.5).
6. If $|f(x_{old}) - f(x_{opt})| < \epsilon$ output x_{out} and $f(x_{out})$. Else, go to step 4.

3 Exercises

1. Solve

$$\min\ x\sin(x) + y\sin(y)$$
$$\text{s.t.}\begin{cases} x \geq \frac{1}{3} \\ y \geq \frac{3}{4} \\ x - \sin(y) \geq 0 \\ x^2 + y^2 \leq 5 \end{cases}$$

Penalty Method Result:

$x_{opt} = (0.6816350232273357, 0.7499996085200379)$
$f(x_{opt}) = 0.9407019589023642$

Wolfram Alpha Answer [4]:

$x_{opt} = (0.681639, 0.75)$
$f(x_{opt}) = 0.940707$

2. Solve

$$\min\ (x+1)^2 + \frac{1}{2}y^2$$
$$\text{s.t.}\begin{cases} x \leq 3 \\ y \geq 0 \\ \frac{1}{8}x^3 - y \geq 0 \end{cases}$$

Penalty Method Result:

$x_{opt} = (-0.0043112873843365665, -1.000843969927903e-08)$
$f(x_{opt}) = 0.9913960124302371$

Wolfram Alpha Answer [4]:

$x_{opt} = (0, 0)$
$f(x_{opt}) = 1$

4 Conclusions

As it can be seen, penalty method is an easy to implement way of solving constrained optimization problems. The results obtained are good, specially in the first exercise. The results for the second exercise are a bit worse, probably due to the power three of the third constrain which makes the gradient for the Quasi-Newton Method very high in comparison to the previous case.

References

[1] D. G. Luenberger, Y. Ye *et al.*, *Linear and nonlinear programming.* Springer, 1984, vol. 2.

[2] Penalty Method - Wikipedia. https://en.wikipedia.org/wiki/Penalty_method.

[3] Quasi-Newton methods - optimization. https://optimization.mccormick.northwestern.edu/index.php/Quasi-Newton_methods.

[4] Wolfram|Alpha: Computational Intelligence. https://www.wolframalpha.com/.