Yugoslav Journal of Operations Research
31 (2021), Number 3, 319–327
DOI: https://doi.org/10.2298/YJOR200515002C

MODIFIED PROJECTED NEWTON SCHEME FOR NON-CONVEX FUNCTION WITH SIMPLE CONSTRAINTS

Suvra Kanti CHAKRABORTY
Department of Mathematics, Sir Gurudas Mahavidyalaya, Kolkata, 700067, India
suvrakanti89@gmail.com

Geetanjali PANDA
Department of Mathematics, Indian Institute of Technology Kharagpur, 721302, India
geetanjali@maths.iitkgp.ernet.in

Received: May 2020 / Accepted: January 2021

Abstract: In this paper, a descent line search scheme is proposed to find a local minimum point of a non-convex optimization problem with simple constraints. The idea ensures that the scheme escapes the saddle points and finally settles for a local minimum point of the non-convex optimization problem. A positive definite scaling matrix for the proposed scheme is formed through symmetric indefinite matrix factorization of the Hessian matrix of the objective function at each iteration. A numerical illustration is provided, and the global convergence of the scheme is also justified.

Keywords: Projected Newton Scheme, Non-convex Function, Simple Constraints, Saddle Point, Local Minimum Point.

MSC: 90C26, 90C30.

1. INTRODUCTION

The non-convex optimization problem is always an area of keen interest for mathematicians and engineers. Even finding a local minimum of a non-convex function is not an easy task. On the other hand, studying different properties of saddle points and escaping from the same is also an area of interest for the researchers in recent times ([1], [2]). In 1982, Bertsekas ([3]) proposed the projected Newton method for simple constraints and it had been widely studied in [4], [5], [6], [7] etc, so far. These schemes are applied to convex functions for which the Hessian matrix is positive semi-definite. We modify the projected Newton method by introducing a positive definite scaling matrix to find a local minimum of the optimization problem involving non-convex objective function with simple constraints. The global convergence of the scheme is also established under mild assumptions. The advantage of the proposed scheme is illustrated through examples, which justify the fact that the scheme escapes from a saddle point utilizing a slight perturbation and finally settles for a local minimum point. Another advantage of the scheme is that the second-order sufficiency test at the solution point is also not required in this process.

Consider the following optimization problem:

$$(P) \quad \min_{x \in \mathbb{R}^n} \; f(x)$$
$$\text{subject to } x \geq 0, \tag{1}$$

$f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function and the vector inequality $x \geq 0$ is componentwise.

Consider the following iterative process for solving (P) due to [3]:

$$x_{k+1} = [x_k - \alpha_k D_k \nabla f(x_k)]^+, \; k = 0, 1, 2, \ldots, \tag{2}$$

where α_k is a positive scalar, D_k is a positive definite symmetric matrix and for a vector $z = (z^1, z^2, \ldots, z^n) \in \mathbb{R}^n$,

$$[z]^+ = \begin{bmatrix} \max(0, z^1) \\ \ldots \\ \max(0, z^n) \end{bmatrix}.$$

For $x \geq 0$ denote $I^+(x) = \left\{ i : x^i = 0, \frac{\partial f(x)}{\partial x^i} > 0 \right\}$.

Theorem 1. *(Prop. 1, [3]) Let $x \geq 0$ and D be a positive definite symmetric matrix which is diagonal with respect to $I^+(x)$ and denote*

$$x(\alpha) = [x - \alpha D \nabla f(x)]^+ \quad \forall \alpha \geq 0$$

(a) x is a critical point w.r.t. problem (P) if and only if $x = x(\alpha)$ for all $\alpha \geq 0$.
(b) If x is not a critical point w.r.t. problem (P), then there exists a scalar $\bar{\alpha} > 0$ such that

$$f(x(\alpha)) < f(x) \quad \forall \alpha \in (0, \bar{\alpha}].$$

Based on the above theory, Bertsekas ([3]) proposed a practical algorithm as follows.

- Consider a small real number $\varepsilon > 0$ and a fixed diagonal positive definite matrix M and two parameters $\beta = 0.9$ and $\sigma = 0.4$, that will be used in connection with an Armijo-like step-size rule.
- An initial vector $x_0 \geq 0$ is chosen and at k-th iteration of the algorithm we have a vector $x_k \geq 0$. Denote $w_k = |x_k - [x_k - M\nabla f(x_k)]^+|$, $\varepsilon_k = \min\{\varepsilon, w_k\}$.

$(k+1)^{st}$ iteration of the algorithm is explained in the following algorithm.

Algorithm 1 $(k+1)^{st}$ **iteration of Newton projection scheme**

Select two parameters $\beta = 0.9$ and $\sigma = 0.4$ and a positive definite matrix D_k which is diagonal with respect to the set I_k^+, where

$$\begin{aligned} I_k^+ =& \Big\{ i : 0 \leq x_k^i \leq \varepsilon_k, \frac{\partial f(x_k)}{\partial x^i} > 0 \Big\} \\ p_k =& D_k \nabla f(x_k), \\ x_k(\alpha) =& [x_k - \alpha p_k]^+ \quad \forall \alpha \geq 0 \end{aligned} \tag{3}$$

Then x_{k+1} is given by $x_{k+1} = x_k(\alpha_k)$, where $\alpha_k = \beta^{m_k}$ and m_k is the first non-negative integer m such that

$$f(x_k) - f(x_k(\beta^m)) \geq \sigma \Big\{ \beta^m \sum_{i \notin I_k^+} \frac{\partial f(x_k)}{\partial x^i} p_k^i + \sum_{i \in I_k^+} \frac{\partial f(x_k)}{\partial x^i} p_k^i (x_k^i - x_k^i(\beta^m)) \Big\} \tag{4}$$

Theorem 2. *([3]) Under the following assumptions, (A) and (B), every limit point of a sequence $\{x_k\}$ generated by (3) is a critical point of problem (P).*

Assumption A: ∇f is Lipschitz continuous on each bounded set of $\mathbb{R}^n$.
Assumption B: There exist positive scalars λ_1 and λ_2 and non-negative integers q_1 and q_2 such that

$$\begin{aligned} &\lambda_1 w_k^{q_1} |z|^2 \leq z' D_k z \leq \lambda_2 w_k^{q_2} |z|^2 \quad \forall z \in \mathbb{R}^n, \quad k = 0, 1, 2, \ldots \\ &w_k = |x_k - [x_k - M\nabla f(x_k)]^+|. \end{aligned} \tag{5}$$

The following theorem demonstrates an important fact that under mild condition the sequence $\{x_k\}$ is attracted by a local minimum x^* satisfying Assumptions C and identifies the set of active constraints at x^* in a finite number of iterations.

Assumption C: f is twice continuously differentiable in the open ball $B(x^*) = \{x : \|x - x^*\| < \delta_1\}$ for $\delta_1 > 0$, x^* as local minimum point of (P) and there exists positive scalar m_1, m_2 such that

$$m_1|z|^2 \leq \nabla^2 f(x)z \leq m_2|z|^2 \quad \forall x$$
$$\text{satisfying } \; |x - x^*| < \delta_1 \text{ and } z \neq 0 \text{ such that } \; z^i = 0 \; \forall i \in B(x^*). \tag{6}$$

Theorem 3. *(Prop. 3, [3]) Let x^* be local minimum of problem (P) satisfying Assumption C. Assume also that Assumption B holds in the stronger form, where in addition to (5) the diagonal elements d^k_{ii} of the elements of the matrices D_k satisfy for some scalar $\bar{\lambda_1} > 0$, $\bar{\lambda_1} \leq d^k_{ii}$, $\forall k = 0, 1, 2, \ldots,$ $i \in I^+_k$.*

Then there exists a scalar $\bar{\delta_1} > 0$ such that if $\{x_k\}$ is a sequence generated by iteration (3) and for some index $\bar{k}$, we have $|x_k - x^*| \leq \bar{\delta_1}$, then $\{x_k\}$ converges to x^* and we have $I^+_k = B(x_k) = B(x^*)$ for all $k \geq \bar{k} + 1$.

2. MODIFIED PROJECTED NEWTON SCHEME

The algorithms described in the previous section hold for a convex programming problem. Here we modify the scheme for the optimization problem involving non-convex objective function with simple constraints. We extend the new scheme over box constraints, which is discussed in Subsection 2.3. Since box-constraint gives a closed bounded domain, then obtaining the local minimum is ensured. This scheme aims to reach some local minimum and not to get stuck at a saddle point.

Since f is non-convex, $\nabla^2 f(x_k)$ may not be positive definite in general, and so $\nabla^2 f$ can be replaced by a symmetric positive definite matrix, as close as possible to the original matrix. A real symmetric matrix A can be expressed as $PAP^T = LBL^T$, where L is a lower triangular matrix, P is a permutation matrix, and B is a block diagonal matrix that allows at most 2×2 blocks. This requires a pivot block initially. There are several pivoting strategies available [8] to preserve the sparsity of the matrix. The symmetric indefinite factorization [9] allows us to determine inertia of the matrix A and inertia of B remains equal to the inertia of A. An indefinite factorization can be modified to ensure that the modified factors are the factors of a positive definite matrix. This idea is briefed in the following Algorithm (See [10]), and for this purpose, MATLAB in-built command *ldl*() is used in this paper since it is less expensive. This algorithm can be found in [11], given as follows.

Algorithm 2 Modifying symmetric indefinite matrix to a positive definite matrix

1: Compute the factorization $PAP^T = LBL^T$.

2: Perform the spectral decomposition of B as $B = Q\Lambda Q^T$.

3: Construct a modification matrix F such that LBL^T is sufficiently positive definite as follows.

Suppose λ_i are the eigenvalues of B. Choose parameter $\delta = 0.000001$ and define F as $F = Q\,\mathrm{diag}(\tau_i)\,Q^T$, where

$$\tau_i = \begin{cases} 0, \text{ if } \lambda_i \geq \delta \\ \delta - \lambda_i, \text{ if } \lambda_i < \delta \end{cases}$$

4: A matrix E has to be added to A to make it positive definite. $P(A+E)P^T = L(B+F)L^T$ provides $E = P^T LFL^T P$. So $\lambda_{\text{min}}(A+E) \approx \delta$.

Output:$\bar{A} \triangleq A + E$ is the positive definite matrix.

Using idea of the logic of positive definite approximation given above, we can get a positive definite approximate of $\nabla^2 f(x_k)$ as $\nabla^2 f(x_k) + E_k$ at the current iteration point x_k. Let $G_k = [\nabla^2 f(x_k) + E_k]^{-1}$. Choose D_k, which is used in (3) as

$$D_k^{ij} = \begin{cases} 0 & \text{if } i \neq j \text{ and either } i \in I_k^+ \ or \ j \in I_k^+, \\ G_k^{ij} & \text{otherwise .} \end{cases} \tag{7}$$

With this choice of above D_k, if we apply Algorithm 1 to generate $\{x^{(k)}\}$, then the algorithm produces a critical point $\bar{x} \in \mathbb{R}^n$ of (P). Consider the following example.

Example 1: $\min\limits_{x_1 \geq 0, x_2 \geq 0, x_3 \geq 0} \phi(x_1, x_2, x_3) = (x_1 - 1)^2 - (x_2 - 1)^2 + x_3^3$

Take the starting point as $(2, 1, 0)$. Using D_k from (7), Algorithm 1 is executed and the next iteration point $(1, 1, 0)$ is obtained. The gradient vector at $(2, 1, 0)$ lies parallel to x_1 axis. Hence, the increasing directions do not affect determining the next direction vector at $(2, 1, 0)$. So the algorithm reaches the critical point $(1, 1, 0)$, which is, in fact, the saddle point of ϕ. The iteration process with Algorithm 1 cannot be progressed further due to the presence of this saddle point. Now to escape from this saddle point, we modify the scheme as follows with rational analysis.

2.1. Modification to escape the saddle point

In the previous section, it is justified that the projected Newton scheme (3) with positive definite modification D_k, provided in (7), converges to a critical point $\bar{x}$ at k^{th} iteration, i.e., $x_k = \bar{x}$. This point is either a saddle point or a minimum point. A small perturbation is considered at $\bar{x}$ in the following way.

First, we take a point $\bar{x}_k$ as $\bar{x}_k = \lambda x_k + (1 - \lambda)x_{k-1}$ for some small positive

λ . Let $v_1 = x_k - x_{k-1}$.
Suppose v_1 is written in the explicit form as $v_1 = (v_{11}, v_{12}, \ldots, v_{1n})$. Now at $\bar{x}_k$, we consider two cases.

- In the first case, let at least two entries of v_1 are non-zero, without loss of generality, say v_{11}, v_{12}. Then the perturbed point is taken simply as $x_{k+1} = \bar{x}_k + \epsilon(-v_{12}, v_{11}, 0, \ldots, 0)$, with ϵ being a small positive number.
- In the second case, let exactly one entry of v_1 be non-zero, without loss of generality, say v_{11}. Then the perturbed point is taken in the direction $x_{k+1} = \bar{x}_k + \epsilon(0, 1, 1, \ldots, 1, 1)$, with ϵ being a small positive number.

In both of the cases, the perturbed point, viz. x_{k+1} lies in the orthogonal to v_1 at $\bar{x}_k$.

Now if the critical point $\bar{x}$ is a local minimum point, the descent scheme (3) at the perturbed point x_{k+1} drags it back to $\bar{x}$ due to its local attraction using Theorem 3. But, if $\bar{x}$ is a saddle point, the Hessian matrix at $\bar{x}$ possesses both positive and negative eigenvalues. Since eigenvalue varies continuously on the variables of the matrix and $f \in \mathscr{C}^2$, there exists a neighbourhood in which the eigenvalue retains its sign (neighbourhood property). Since we can choose the perturbed point x_{k+1} sufficiently close to $\bar{x}$, $\nabla^2 f(x_{k+1})$ has both positive and negative eigenvalues. Also we note that, since $\bar{x}$ is a saddle point, the vector v_1 must be an eigenvector of $\nabla^2 f(x_{k-1})$ corresponding to its positive eigenvalue. Since x_{k+1} is chosen from the orthogonal complement of v_1, the descent direction computed at x_{k+1} will have the effect of both positive and negative curvature. As a result of the negative curvature, the next iteration point is repelled from $\bar{x}$. In a nutshell, once a critical point is reached, we store it in memory. Then applying Algorithm 1 at the perturbed point, if the same critical point is reached, it can be concluded that the same is a local minimum point and hence, we stop.

2.2. Algorithm of the proposed scheme

Algorithm 3 Modified projection Newton Scheme for the non-convex objective function and simple constraints

```
1: At current iteration point use Algorithm 1 to reach a critical point $\bar{x}$.
2: The perturbation is computed by the logic of subsection 2.1 and apply Algorithm 1 on the perturbed point to reach next critical point.
3: Stop if the next critical point is again $\bar{x}$. Else Go to step 2.
Output: $\bar{x}$ is a local minimum point of the problem.
```

2.3. Extension to box constraints

This idea can be now extended in case of problems with box constraints.

$$\begin{aligned} &\min f(x) \\ &\text{subject to } b_1 \leq x \leq b_2, \end{aligned}$$

where b_1 and b_2 are given vectors of lower and upper bounds with $b_1 \leq b_2$. Hence, the set I_k^+ becomes

$$\begin{aligned} I_k^\sharp &= \left\{ i : b_1^i \leq x_k^i \leq b_1^i + \varepsilon_k, \frac{\partial f(x_k)}{\partial x^i} > 0 \right\} \\ \text{or} \quad I_k^\sharp &= \left\{ i : b_2^i - \varepsilon_k \leq x_k^i \leq b_2^i, \frac{\partial f(x_k)}{\partial x^i} < 0 \right\} \end{aligned}$$

and the definition of $x_k(\alpha)$ is changed to

$$x_k(\alpha) = [x_k - \alpha D_k \nabla f(x_k)]^\sharp, \tag{8}$$

where for all $z \in \mathbb{R}^n$ we denote by $[z]^\sharp$ the vector with co-ordinates

$$[z]^{\sharp^i} = \begin{cases} b_2^i & \text{if } b_2^i \leq z^i, \\ z^i & \text{if } b_1^i \leq z^i \leq b_2^i, \\ b_1^i & \text{if } z^i \leq b_1^i. \end{cases}$$

The scalar ε_k is given by $\min\{\varepsilon, |[x_k - M\nabla f(x_k)]^\sharp|\}$. The matrix D_k is positive definite by (7) and diagonal with respect to $I_k^\sharp$ and M is fixed diagonal positive definite matrix. The iteration is given by

$$x_{k+1} = x_k(\alpha_k),$$

where α_k is chosen by (4) with $[x_k^i - x_k^i(\beta^m)]^+$ replaced by $[x_k^i - x_k^i(\beta^m)]^\sharp$. The perturbation will be given to the inactive constraints, and similarly, the next iteration will be repelled if the critical point is a saddle point. So it (box constraints case) is a natural extension of $x \geq 0$ case.

3. NUMERICAL ILLUSTRATIONS

Choose Example 1 with box constraints. The bounds of the coordinates are taken as $0 \leq x_i \leq 5$, for $i = 1, 2, 3$. With this starting point, Algorithm 3 initially produces the critical point $(1, 1, 0)$. Next, we select a point $(1.001, 1, 0)$ in a close neighbourhood of $(1, 1, 0)$. The gradient vector at this point is $(1, 0, 0)$. Now employing the method described in Subsection 2.1 and choosing $\epsilon = .001$, the next perturbed point becomes $(1.001, 1.001, 0.001)$. In a nutshell, we perform a perturbation of $(0.001, 0.001, 0.001)$ at the saddle point $(1, 1, 0)$ along the coordinates. Next, by Step 2 of Algorithm 3, we reach another critical point $(1, 5, 0)$, a desired local minimum point. We have used MATLAB R2015a with user defined tolerance limit for the numerical computations as 10^{-6}. The details of numerical computations are summarized in Table 1. Moreover, Figure 1 shows the iteration points' movement starting from the initial guess to the local minimum point.

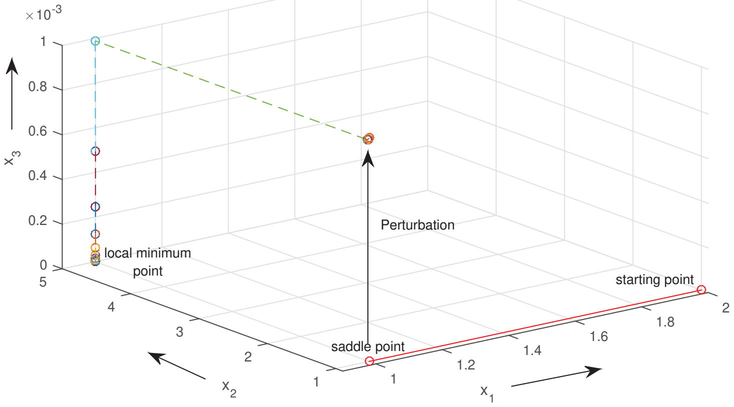

Figure 1: Movements of the iteration points for Algorithm 3 on Example 1

Table 1: Iteration points for Algorithm 3 on Example 1

k	x_k	$-\alpha_k p_k$	$x_{k+1} = x_k - \alpha_k p_k$
0	$(2, 1, 0)$	$(-1, 0, 0)$	(1,1,0)
...	Perturbation	...	...
1	$(1.001, 1.001, 0.001)$	$(-0.002, -0.002, 0)$	$(0.9990, 1.003, 0.000997)$
2	$(0.9990, 1.027, 0.000997)$	$(0.002, -.017999, -0.000003)$	$(1.001, 1.008999, 0.000994)$
3	$(1.001, 1.008999, 0.000994)$	(-0.002,0.017999, -0.00000299)	$(0.999, 1.026999, 0.000991)$
4	$(0.999, 1.026999, 0.000991)$	$(0.000003, 3.973, -0.000002)$	$(0.999003381, 5, 0.000989)$
5	$(0.999, 5, 0.000989)$	$(0.000999, 0, -0.000494)$	$(1, 5, 0.000494)$
6	$(1, 5, 0.000494)$	$(0, 0, -0.000247)$	$(1, 5, 0.000247)$
7	$(1, 5, 0.000247)$	$(0, 0, -0.000123)$	$(1, 5, 0.00012)$
8	$(1, 5, 0.00012)$	$(0, 0, -0.00006)$	$(1, 5, 0.00006)$
9	$(1, 5, 0.00006)$	$(0, 0, -0.00003)$	$(1, 5, 0.00003)$
10	$(1, 5, 0.00003)$	$(0, 0, -0.000015)$	$(1, 5, 0.000001)$
11	$(1, 5, 0.000001)$	$(0, 0, -0.000001)$	$(1, 5, 0)$
...	Perturbation	...	...
12	$(1.001, 5.001, 0.001)$	$(-0.001, -0.001, 0.00024)$	$(1, 5, 0.00025)$
13	$(1, 5, 0.00025)$	$(0, 0, -000129)$	$(1, 5, 0.00012)$
14	$(1, 5, 0.00012)$	$(0, 0, -000059$	$(1, 5, 0.00006)$
15	$(1, 5, 0.00006)$	$(0, 0, -0.00003)$	$(1, 5, 0.00003)$
16	$(1, 5, 0.00003)$	$(0, 0, -0.000015)$	$(1, 5, 0.000001)$
17	$(1, 5, 0.000001)$	$(0, 0, -0.00001)$	$(1, 5, 0)$

3.1. Advantages of the scheme

- Escapes from a critical point if it is a saddle point.
- Finally, a local minimum is ensured to be attained.
- Second-order sufficiency test is not required.

4. CONCLUSIONS

This paper proposes a projected Newton-like descent scheme, which searches for a local minimum of a non-convex optimization problem. This scheme never sticks to the saddle point even in the slightest possibility; rather, it reaches a local minimum point. At the solution point, a second-order sufficiency test for constrained optimization is also not required. One may further use this concept to escape the saddle point in the existing non-convex optimization schemes.

REFERENCES

[1] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y., "Identifying and attacking the saddle point problem in highdimensional non-convex optimization", *Advances in neural information processing systems*, 27 (2014) 2933–2941.

[2] Jayswal, A., Ahmad, I., and Banerjee, J., "Nonsmooth interval-valued optimization and saddle-point optimality criteria", *Bulletin of the Malaysian Mathematical Sciences Society*, 39 (4) (2016) 1391–1411.

[3] Bertsekas, D. P., "Projected Newton methods for optimization problems with simple constraints", *SIAM Journal on Control and Optimization*, 20 (2) (1982) 221–246.

[4] Bardsley, J. M., and Vogel, C. R., "A nonnegatively constrained convex programming method for image reconstruction", *SIAM Journal on Scientific Computing*, 25 (4) (2004) 1326–1343.

[5] Chi, E. C., Zhou, H., and Lange, K., "Distance majorization and its applications", *Mathematical Programming*, 146 (1-2) (2014) 409–436.

[6] Kuang, D., Park, H., and Ding, C. HQ, "Symmetric nonnegative matrix factorization for graph clustering", *SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, California, USA, 106-117, 2012.

[7] Schmidt, M., Kim, D., and Sra, S., "Projected Newton-type methods in machine learning", *Optimization for Machine Learning*, MIT Press, Cambridge, MA, USA, 2011.

[8] Golub, G. H., and Van Loan, C. F., "Matrix Computations", *Johns Hopkins University Press*, Baltimore, USA, 2012.

[9] Cheng, S. H., and Higham, N. J., "A modified Cholesky algorithm based on a symmetric indefinite factorization", *SIAM Journal on Matrix Analysis and Applications*, 19 (4) (1998) 1097–1110.

[10] Nocedal, J., and Wright, S. J., "Numerical Optimization", *Springer Series in Operations Research and Financial Engineering*, Springer, New York, USA, 2006.

[11] Chakraborty, S. K., and Panda, G., "Newton like line search method using q-calculus", *Mathematics and Computing, ICMC 2017, Communications in Computer and Information Science, vol. 655*, Springer, Singapore, 196-208, 2017.