Mathematical Optimization Techniques in Computational of Biological Models

Huda Amer Hadi, Ahmed Sabah Al-Jilawi
University of Babylon, College of Education for Pure Sciences, Mathematics Department, Hilla, Iraq

E-mail: huda_amir80@yahoo.com, aljelawy2000@yahoo.com

Abstract. The aim of this research is focused on the mathematical optimization in computational systems biological models based on find the good method that gives converges faster. Our strategy is to use Bundle Method instead of using Subgradient Methods. Also, we improve the convergence of theoretical properties of the methods. The basic idea of approximation is the subdifferential of the objective function by using subgradients from previous iterations of a bundle method.

Keywords: Bundle method, Subdifferential, Sub gradient.

1. Introduction
Optimization is an essential mathematical tool that aims to find the Best solution that provides the minimizers or maximizers value of an objective function subject to equality or inequality constraints. The optimization algorithm is an important tool to find the solution, usually with the help of a computer, by means of an iterative procedure that begins with an initial guess of the value of the variables and generates the sequence of improved evaluate, or iterates, until we terminate at an optimal solution (Oster, 2014). A good algorithm should be robust, efficient, and accurate; that is, it should always work, it should be fast, and it should provide a better approximation of an optimal solution (Wright & Nocedal, 1999). The application, optimal control, economics, applied mathematics, computational chemistry and physics (Bertsekas, 2014). Bundle Methods, Derivative Free Methods, Subgradient Methods, Gradient Sampling Methods Hybrid Methods, Special Methods The reasons we are interested in Bundle Method that the whole subdifferentiable of the function but only one arbitrary subgradient at each point (Bertsekas, Nedi, Ozdaglar, et al., 2003).

2. Subdifferential and Subgradient
Subdifferential calculus is a powerful tool to hold convex optimization where the objective function is nondifferentiable. We notices the subgradient and the subdifferential of a convex functions, see [(McCormick, 1983), (Wright & Nocedal, 1999), (Cook, Cunningham, Pulleyblank, & Schrijver, 2009)].
Definition 2.1 (Wright & Nocedal, 1999) Let the convex function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is the set \( \partial f (x) = \{ \lambda \in \mathbb{R}^n | f(y) \geq f(x) + \lambda^T(y - x) \text{ for all } y \in \mathbb{R}^n \} \).

Definition 2.2 (Cook et al., 2009) Let \( S \subseteq \mathbb{R}^n \) such that the statements hold,
\[
\zeta x_1 + (1 - \zeta) x_2 \in S, \forall x_1, x_2 \in S, \forall \zeta \in [0, 1],
\]
Therefore \( S \) is denoted to be convex. Shown that a set \( S \subseteq \mathbb{R}^n \) is convex if and only if for any \( x_1, \ldots, x_n \in S \), the convex combination
\[
\sum_{i=1}^{n} \zeta_i x_i,
\]
such that \( \sum_{i=1}^{n} \zeta_i = 1, \zeta_i \geq 0, i = 1, \ldots, n \) belongs to \( S \).

Definition 2.3 (Cook et al., 2009) Let \( S \subseteq \mathbb{R}^n \) be a nonempty convex set. If \( f : S \rightarrow \mathbb{R} \) satisfies
\[
f(\zeta x_1 + (1 - \zeta) x_2) \leq \zeta f(x_1) + (1 - \zeta) f(x_2), \quad \forall x_1, x_2 \in S, \forall \zeta \in [0, 1],
\]
therefore \( f \) is said convex function on \( S \). If the inequality is strict inequality for all \( x_1 \neq x_2 \) and for all \( \zeta \in (0, 1) \), thus \( f \) is called a strictly convex function on \( S \). If we have a constant \( b > 0 \) such that for all \( x_1, x_2 \in S \), and for all \( \alpha \in [0, 1] \), therefore \( f \) is said strongly convex function on \( S \). A function \( f \) is concave if \( -f \) is convex.

3. Differentiable of Vector Functions

In this section, we begin with various definitions concerning differentiability and the derivative of a scalar-valued function, see (Papadimitriou & Steiglitz, 1998) (McCormick, 1983) (Bertsekas, 1999).

Definition 3.4 (Bertsekas, 1999) Assume \( \psi : \mathbb{R}^n \rightarrow \mathbb{R} \) and \( x \in \mathbb{R}^n \). Then the partial derivative of \( \psi \) on \( x \) with respect to \( x_i \) is defined to be
\[
\frac{\partial \psi}{\partial x_i} = \lim_{t \to 0} \frac{\psi(x + te_i) - \psi(x)}{t}
\]
where \( e_i \) is ith unit vector. The gradient of \( \psi \) at \( x \) is defined as the column vector
\[
\nabla \psi(x) = \begin{bmatrix} \frac{\partial \psi(x)}{\partial x_1} \\ \vdots \\ \frac{\partial \psi(x)}{\partial x_n} \end{bmatrix}
\]
The **Hessian matrix** is defined to be the \( n \times n \) symmetric matrix

\[
\nabla^2 \psi(x) = \begin{pmatrix}
\frac{\partial^2 \psi}{\partial x_1 \partial x_1} & \frac{\partial^2 \psi}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \psi}{\partial x_1 \partial x_n} \\
\frac{\partial^2 \psi}{\partial x_2 \partial x_1} & \frac{\partial^2 \psi}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 \psi}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 \psi}{\partial x_n \partial x_1} & \frac{\partial^2 \psi}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \psi}{\partial x_n \partial x_n}
\end{pmatrix}
\]

The **directional derivative formula** of the function \( \psi \) at \( x \) in the direction \( d \) given by

\[
\psi'(x, d) = \lim_{t \to 0^+} \frac{\psi(x + td) - \psi(x)}{t}.
\]

We said the function \( \psi \) is **differentiable** at \( x \) if the gradient \( \nabla \psi(x) \) exists and satisfies \((\nabla \psi(x), d) = \psi'(x, d), \forall d \in \mathbb{R}^n \). Moreover, we say the function \( \psi \) is **differentiable on** \( S \) of \( \mathbb{R}^n \) if it is differentiable on \( x \in S \), and \( \psi \) is **continuously differentiable** over \( S \), if

\[
\lim_{d \to 0} \frac{\psi(x + d) - \psi(x) - (\nabla \psi(x), d)}{\|d\|} = 0 \quad \forall x \in S,
\]

where \( \| \cdot \| \) is an arbitrary vector norm.

**Definition 3.5** (Bertsekas, 1999) Let \( \psi : \mathbb{R}^n \to \mathbb{R} \) be differentiable at \( x \in \mathbb{R}^n \). We can say that \( d \in \mathbb{R}^n \) is a **descent direction** of the function \( \psi \) at \( x \) if

\[
(\nabla \psi(x), d) < 0.
\]

**Definition 3.6** We define \( f \) is **little-oh** of \( h \) as \( x \) approaches \( a \) and write

\[
f'(x) = o(h(x)) \text{ as } x \to a,
\]

that's mean

\[
\lim_{x \to a} \frac{f(x)}{h(x)} = 0
\]

In cases where there is a third function, \( g(x), f'(x) = g(x) + o(h(x)) \) which implies

\[
\lim_{x \to a} \frac{f(x) - g(x)}{h(x)} = 0
\]
That is, for \( h : IR^n \rightarrow IR \), we have

\[
\lim_{k \to \infty} \frac{o(h(x^k))}{h(x^k)} = 0,
\]

for all sequences \( \{x^k\} \) such that \( x^k \to a \) and \( h(x^k) \neq 0 \) for all \( k \).

**Theorem 3.1** (Taylor Expansion) (Bertsekas, 1999) Assume \( \psi : IR^n \rightarrow IR \) be continuously differentiable. Then, for all \( x_1, x_2 \in IR^n \), there is an \( \alpha \in [0, 1] \), such that

\[
(x_2) = (x_1) + \nabla \psi(x_1) (x_2 - x_1) + \frac{1}{2} (x_2 - x_1)^T \nabla^2 \psi(x_1) (x_2 - x_1).
\]

Also, if \( \psi \) is twice continuously differentiable, then, for all \( x_1, x_2 \in IR^n \), there is \( \alpha \in [0, 1] \), such that

\[
\psi(x_2) = \psi(x_1) + \nabla \psi(x_1)^T (x_2 - x_1) + \frac{1}{2} (x_2 - x_1)^T \nabla^2 \psi(x_1) (x_2 - x_1).
\]

In addition, if \( x, u \in IR^n \) and \( \eta \in IR \), such that

\[
\psi(x + \eta u) = \psi(x) + \eta u^T \nabla \psi(x) + \frac{\eta^2}{2} u^T \nabla^2 \psi(x) u + O(\eta^2) \quad \text{as} \ \eta \to 0.
\]

**Definition 3.7** (Bertsekas, 2014) The function \( \psi : IR^n \rightarrow IR^m \), with component functions \( \psi_1, \ldots, \psi_m \), is called differentiable if each component is differentiable. The gradient matrix of \( \psi \), denoted \( \nabla \psi(x) \), is the \( n \times m \) matrix with ith column is the gradient \( \nabla \psi_i(x) \) of \( \psi_i \):

\[
\nabla \psi(x) = [\nabla \psi_1(x) \cdots \nabla \psi_m(x)].
\]

Then the **Jacobian** of \( \psi \) at \( x \) is defined

\[
D(x) = [\nabla \psi(x)]^T = \begin{pmatrix}
\nabla \psi_1(x)^T \\
\vdots \\
\nabla \psi_m(x)^T
\end{pmatrix} = \begin{pmatrix}
\frac{\partial \psi_1(x)}{\partial x_1} & \cdots & \frac{\partial \psi_1(x)}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial \psi_m(x)}{\partial x_1} & \cdots & \frac{\partial \psi_m(x)}{\partial x_n}
\end{pmatrix}.
\]

**3.1 The Optimal Conditions of The Unconstrained Optimization**

Here review the unconstrained optimization problem. If \( X = IR^n \), i.e., minimize \( f \) without constraints (Bertsekas, 1999), it can be expressed as:

\[
\text{Minimize } f'(x) \quad \text{subject to } x \in IR^n \quad (1)
\]

- If \( f' \) is continuous differentiable, then a necessary condition for \( x^* \in IR^n \) to be a solution of problem (1) is
\[
\n\nabla f' (x^*) = 0.
\]

- If \( f' \) is twice continuously differentiable, then a necessary conditions for \( x^* \in \mathbb{R}^n \) to be a solution of problem (1) is

\[
\nabla f' (x^*) = 0, \quad \nabla^2 f' (x^*) \geq 0.
\]

- The sufficient conditions for \( x^* \in \mathbb{R}^n \) is said to be a local solution of problem (1) are

\[
\nabla f' (x^*) = 0, \quad \nabla^2 f' (x^*) > 0.
\]

**Theorem 3.2 (The First Order Necessary Conditions)** (Bertsekas, 1999)

Let \( f' : \mathbb{R}^n \rightarrow \mathbb{R} \) be differentiable. Let \( x^* \) is a local minimum of \( f' \), then \( \nabla f' (x^*) = 0 \).

**Proof:** Define \( h : \mathbb{R} \rightarrow \mathbb{R} \) as \( h(\eta) = f'(x^* + \eta u) \) for some \( u \in \mathbb{R}^n \), then \( h'(\eta) = u^T \nabla f' (x^* + \eta u) \).

If \( \eta = 0 \), we get \( h(0) = u^T \nabla f' (x^*) \).

By definition,

\[
 h'(\eta) = \lim_{\eta \to 0} \frac{f'(x^* + \eta u) - f'(x^*)}{\eta}
\]

we know that \( x^* \) is the local minimum, such that there exist \( t > 0 \), implies \( f'(x^* + \eta u) \geq f'(x^*) \) for all \( 0 < \eta \leq t \), therefore we get \( u^T \nabla f' (x^*) \geq 0 \). Since \( u \) is an arbitrary, we can replace \( u \) by \(-u\), and thus \(-u^T \nabla f' (x^*) \geq 0\). Therefore, \( u^T \nabla f' (x^*) = 0 \), for all \( u \in \mathbb{R}^n \). Thus, \( \nabla f' (x^*) = 0 \).

**Example:** Consider the following simple linear programming problem:

\[
(P) \begin{align*}
\text{minimize} & \quad x_1 + 3x_2 + 5x_3 \\
\text{subject} & \quad x_1 + x_2 + x_3 = 3 \\
& \quad x \geq 0
\end{align*}
\]

The dual of problem (2) is given by

\[
(D) \begin{align*}
\text{minimize} & \quad 3y_1 \\
\text{subject} & \quad y_1 \geq 1, \\
& \quad y_1 \geq 3, \\
& \quad y_1 \geq 5.
\end{align*}
\]
The optimal solution of problem (2) is \( x_1 = x_2 = 0, x_3 = 3 \) and the optimal value is \( x_1 + 3 x_2 + 5 x_3 = 15 \);
The optimal solution of problem (3) is \( y_1 = 5 \), also the optimal value \( 3 y_1 = 15 \).

### 3.2 Computational of Biological Models

In this section, we introduce the Biological Models which will be fundamental in the algorithm. In table (1) notices the CPU time vs the number of function calls for different graphs. We observe the Bundle method reaches the optimality solution in less time than the Subgradient methods. Furthermore, we can see that the number of iterations (reported as fcallsBudl and fcallsSubgr) required by the Bundle method is less. In Figure (5), we plot the bounds against the number of function calls. Since the number is deterministic, while the CPU time can vary between runs, we chose.

**Algorithm 1**: The Bundle Method

1. Given \( y^0 \), and \( \alpha^0 > 0 \).
2. Find \( y^{k+1} \) such that
   \[
   y^{k+1} = \arg \min B_q(y, \alpha^k).
   \]
3. Choose
   \[
   \alpha^{k+1} \leq \alpha^k.
   \]
4. Set \( k = k + 1 \) and repeat.
Table 1: CPU Time and number of function calls for Bundle method and Subgradient method.

| Problem | timeBudl | timeSubgr | fcallsBudl | fcallsSubgr |
|---------|----------|-----------|------------|-------------|
| g05 50.0 | 2.1      | 4.0       | 86         | 166         |
| g05 50.1 | 2.9      | 3.5       | 90         | 136         |
| g05 50.2 | 2.7      | 4.8       | 87         | 192         |
| g05 50.3 | 3.0      | 5.8       | 103        | 221         |
| g05 50.4 | 3.5      | 7.2       | 109        | 239         |
| g05 50.5 | 3.0      | 5.1       | 96         | 197         |
| g05 50.6 | 3.5      | 4.5       | 92         | 164         |
| g05 50.7 | 2.5      | 4.5       | 84         | 154         |
| g05 50.8 | 3.1      | 6.4       | 95         | 183         |
| g05 50.9 | 2.3      | 4.5       | 76         | 149         |

4. Conclusion

Initially, introduce the Biological Models which will be fundamental in the algorithm, then implemented Bundle Method and Subgradient method for linear programming problems. We used them on several graphs. Our results show that Bundle method reaches the optimal solution in approximately half the number of function calls as the Subgradient method, and approximately 1.7 times faster in CPU time.

References

[1] Bertsekas, D. P. (1999). *Nonlinear programming*. Athena scientific Belmont.
[2] Bertsekas, D. P. (2014). *Constrained optimization and lagrange multiplier methods*. Academic press.
[3] Bertsekas, D. P., Nedi, A., Ozdaglar, A. E., et al. (2003). Convex analysis and optimization.
[4] Cook, W. J., Cunningham, W., Pulleyblank, W., & Schrijver, A. (2009). Combinatorial optimization. Oberwolfach Reports, 5 (4), 2875–2942.
[5] McCormick, G. P. (1983). *Nonlinear programming; theory, algorithms, and applications* (Tech. Rep.).
[6] Oster, M. R. (2014). *Two applications of combinatorial optimization*. Rutgers TheState University of New Jersey-New Brunswick.
[7] Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. Courier Corporation.
[8] Wright, S., & Nocedal, J. (1999). Numerical optimization. Springer Science, 35(67- 68), 7.