PKU-IAI Technical Report: TR-PKU-IAI-2024-0006

Differentiable Engine for Tri-level Optimization

Yuntian Gu
Yuanpei College
Peking University
guyuntian@stu.pku.edu.cn

Xuzheng Chen
Yuanpei College
Peking University
2000017764@stu.pku.edu.cn

Junqing Chen
Yuanpei College
Peking University
chenjunqing@stu.pku.edu.cn

Abstract

Bi-level optimization, a concept extensively explored in economics, mathematics, and computer science, has recently gained renewed interest in machine learning. It shows promise in various machine learning applications, including hyperparameter tuning and continual learning. This article provides an overview of two principal forms of bi-level optimization, distinguished by whether the lower level optimization is parameterized by upper level parameters. We discuss the conventional gradient-based solutions and propose their extension to tri-level optimization, potentially applicable in multi-stage game scenarios. We conduct extensive experiments and show that our algorithm indeed outperforms vanilla alternative.

1 Introduction

Bi-level optimization was first introduced by Stackelberg in 1934 as a concept in economic game theory [3]. Recently, with the rapid development of deep learning and machine learning techniques, bi-level optimization has been widely exploited in reinforcement learning [7], hyperparameter finetuning and meta-learning [4].

Bi-level optimization is a hierarchical process consisting of upper level optimization and inner level optimization. The general form of bilevel optimization is

$$\begin{aligned} &\text{minimize}_{\mathbf{x}} && f^U(\mathbf{x}, \mathbf{y}) \\ &\text{subject to} && \mathbf{y} \in \arg\min_{\mathbf{y}'} f^L(\mathbf{x}, \mathbf{y}') \end{aligned} \tag{1}$$

where U and L mean upper level and lower level respectively, $\mathbf{x} \in \mathbb{R}^{\mathbf{m}}$, $\mathbf{y} \in \mathbb{R}^{\mathbf{n}}$, and $f^U, f^L : \mathbb{R}^{m+n} \to \mathbb{R}$. Bi-level optimization can be viewed as a leader-follower game. The lower level follwer always optimizes its own utility function f^L based on the leader's action. And the upper level leader will determine its optimal action for utility function f^U with the knowledge of follower's policy.

When the inner optimization is not parameterized by upper level parameters, bi-level optimization degenerates to a special form called simple bi-level optimization [8].

$$\begin{aligned} &\text{minimize}_{\mathbf{x}} && f^U(\mathbf{x}) \\ &\text{subject to} && \mathbf{x} \in \arg\min_{\mathbf{x}} f^L(\mathbf{x}) \end{aligned} \tag{2}$$

For the simple bi-level optimization, the f^L must not be strict convex, as otherwise the minimum of f^L is unique and the bi-level optimization degenerates to single level optimization, with the lower level optimization only.

Generally, the solutions to bi-level optimization can be classified into 4 categories. The first approach tries to explicitly figure out the function from x to y, $y(x)$ from the lower level optimization problem. The upper level optimization then turns into unconstrained optimization

$$\text{minimize}_{\mathbf{x}} \quad f^U(\mathbf{x}, \mathbf{y}(\mathbf{x})) \tag{3}$$

which is easy to solve through analytic methods or numerical methods like gradient decent.

The second approach solves the problem by replacing the lower level problem with equivalent forms, like inequalities or KKT conditions. For example, the Eq. (1) can be equivalently represented as

$$\begin{aligned} &\text{minimize}_{\mathbf{x}} && f^U(\mathbf{x}, \mathbf{y}) \\ &\text{subject to} && f^L(\mathbf{x},\mathbf{y}) \leq (f^L)^* \end{aligned} \tag{4}$$

where $(f^L)^*$ is the minimum of $f^L(\mathbf{x}, \mathbf{y})$.

The third approach is an approximate method called Tikhonovtype regularization [1]. It combines two optimization targets together with a coefficient parameter σ

$$\text{minimize}_{\mathbf{x},\mathbf{y}} \quad f^U(\mathbf{x}, \mathbf{y}) + \sigma * f^L(\mathbf{x}, \mathbf{y}) \tag{5}$$

In simple bi-level optimization scene, it is known that when $\sigma \to 0$, the solution to the coupled optimization target is also valid in original bi-level optimization.

The last approach is gradient-based. The gradient of $\mathbf{x}$ to the upper level function $f^U(\mathbf{x}, \mathbf{y})$ is

$$\frac{\mathrm{d}f^U}{\mathrm{d}\mathbf{x}} = \frac{\partial f^U}{\partial \mathbf{x}} + \frac{\partial f^U}{\partial \mathbf{y}} \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\mathbf{x}} \tag{6}$$

In the equation, $\frac{\partial f^U}{\partial \mathbf{x}}$ and $\frac{\partial f^U}{\partial \mathbf{y}}$ are known. If somehow we can calculate or approximate the gradient of $\mathbf{x}$ to $\mathbf{y}$, i.e. $\frac{d\mathbf{y}}{d\mathbf{x}}$, we will be able to optimize the upper level function through gradient decent as follows

$$\mathbf{x}' \leftarrow \mathbf{x} - \alpha(\frac{\partial f^U}{\partial \mathbf{x}} + \frac{\partial f^U}{\partial \mathbf{y}} \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\mathbf{x}})\big|_{\mathbf{x},\mathbf{y}^*} \tag{7}$$

Note that the gradient is calculated at $(\mathbf{x}, \mathbf{y}^*)$ due to the lower level constraint.

In the following sections, we will first explain approaches to calculate $\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\mathbf{x}}$ in bi-level optimization and then extend it to tri-level situations.

2 Related Work

Bi-level optimization, originating from von Stackelberg's 1934 thesis on market structure [6], involves hierarchical decision-making between a leader and a follower. The field has evolved into various directions, presenting challenges like whether follower's decision is well-posed and transforming bi-level problems into equivalent single-level ones.Current bi-level optimization encompasses diverse problem types, employing both deterministic algorithms and metaheuristics, despite its NP-hard nature.

Several approaches to bi-level optimization include constraint-based and gradient-based algorithms. Recent works propose stochastic bi-level algorithms with momentum recursive [16, 2] and variance reduction techniques[13, 12, 9], claiming improved computational complexity. Additionally, a single-loop momentum-based recursive bi-level optimizer (MRBO) [18] is introduced, demonstrating lower complexity than existing stochastic optimizers.

In the realm of bi-level optimization, first-order gradient-based techniques [11] gain significance, especially with the prevalence of deep neural network models in machine learning and computer vision. The focus is on differentiating parameterized argmin and argmax problems, with an emphasis on revisiting and presenting results in the context of first-order gradient procedures for solving bi-level optimization problems, considering the growth of deep learning.

3 Preliminaries

As mentioned above, we can see that the key to getting the gradients is to obtain the derivative of y with respect to x because f is known and the partial derivatives of f with respect to x and y can be calculated easily. One intuitive idea is to utilize the function $f^L(x, y)$ to get the relationship between x and y using chain rule[14]: we know that the constraint that we need to minimize $f^L(x, y)$ can provide the relationship $y = y(x)$ so that we can replace y with x in f^U.

The derivative of f^L may be an unsolvable differential equation and may not directly provide an explicit expression $y = y(x)$. But we can still derive the derivative of y with respect to x using lemmas and theorems provided in [5]. However, the proof of lemma 3.1 in [5] is not entirely correct. It neither explains whether $g(x)$ is differentiable nor considers whether f_{YY} is zero (the notion here is consistent with lemma 3.2 in this paper). We provide another proof here:

Lemma 3.1. *Let $F(x, y)$ be a binary function with respect to variables x and y, which satisfies the following conditions in the neighborhood $U((x_0, y_0), \delta)$:*

1. $F(x_0, y_0) = 0$.
2. $F'_x(x, y)$ *and* $F'_y(x, y)$ *are continuous in* $U((x_0, y_0), \delta)$.
3. $F'_y(x_0, y_0) \neq 0$

Then $\exists\ 0 \leq \delta_0 \leq \delta$ such that there exists a unique continuous function in the neighborhood that satisfies the following conditions:

1. $y_0 = f(x_0)$.
2. $F(x, f(x)) = 0$ *in the neighborhood* $U(x_0, \delta_0)$.
3. $f(x)$ *has continuous derivation in the neighborhood* $U(x_0, \delta_0)$, *and*

$$f'(x) = -\frac{F'_x(x, f(x))}{F'_y(x, f(x))}$$

Detailed proof can be seen in Appendix.

By using lemma 3.1, we can obtain the same result as in the paper and be more rigorous:

Lemma 3.2. *Let $f : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ be a continuous with continuous second derivatives and $F'_y(x, y) \neq 0$. Then the derivative of g with respect to x is*

$$\frac{\mathrm{d}g(x)}{\mathrm{d}x} = -\frac{f_{XY}(x, g(x))}{f_{YY}(x, g(x))}$$

Based on the lemma and theorem above, we have a method to calculate the derivative of y with respect to x, which allows us to obtain gradients easily. But they are still within the bi-level framework. However, in some cases, there may not be only two levels, which means there are higher-level "leaders" to make decisions over leaders. In the follow-up of the article, we will discuss higher-level frameworks, mainly focusing on tri-level and providing n-level conclusions without proof.

4 Gradient-based Optimization Algorithm for Tri-level Problems

In this section, we propose a new algorithm for tri-level optimization and shed light on solving a general n-level optimization optimization problem.

4.1 The procedure of Tri-level Optimization

Let us go back to the bi-level structure mentioned above first. As discussed in 1, bi-level optimization can be viewed as a leader-follower game. Essentially, this is a special type of sequential game involving two players. Sequential game refers to a game form in which players choose strategies in a sequential order. Therefore, some opponents may take actions first and others may take actions later.

In bi-level optimization, the follower makes his own optimal decision $y(x)$ for each fixed decision x of the leader. With the optimal decision $y(x)$ of the follower, the leader can select x that maximize his utility, which makes $f(x, y)$ to the minimum here. Further speaking, when there are 3 players making decision in a specific order, it can be seen as leaders of different levels. Specifically, the third player gives $\arg\min_z f_3(x, y, z)$ given fixed x and y. The second player gives $\arg\min_y f_2(x, y, z)$ given fixed x, with perfect understanding of how z will change according to y. The first player gives $\arg\min_x f_1(x, y, z)$, with perfect understanding of how y and z changes according to x.

4.2 The gradient of Tri-level Optimization

We start by considering the solution of lower problems $g(x, y) = \arg\min_{z\in\mathbb{R}} f_3(x, y, z), h(x) = \arg\min_{y\in\mathbb{R}} f_2(x, y, g(x, y))$. In our results, we assume that the minimum over y or z over the above functions exists over the domain of y or z. When the minimum is not unique, $g(x, y)$ or $h(x)$ can be taken as any one of the minimum points. Moreover, we do not require $g(x, y)$ or $h(x)$ to have a closed-form formula.

Lemma 4.1. *Let $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a continuous function with third derivatives. Let $g(x, y) = \arg\min_{z\in\mathbb{R}} f(x, y, z)$, then the following properties about g holds:*
(a)

$$\frac{\partial g}{\partial x} = -\frac{f_{XZ}}{f_{ZZ}}$$

(b)

$$\frac{\partial g}{\partial y} = -\frac{f_{YZ}}{f_{ZZ}}$$

(c)

$$\frac{\partial^2 g}{\partial x \partial y} = -\frac{\frac{\partial g}{\partial x}\frac{\partial g}{\partial y} f_{ZZZ} + \frac{\partial g}{\partial y} f_{XZZ} + \frac{\partial g}{\partial x} f_{YZZ} + f_{XYZ}}{f_{ZZ}}$$

(d)

$$\frac{\partial^2 g}{\partial y^2} = -\frac{(\frac{\partial g}{\partial x})^2 f_{ZZZ} + 2\frac{\partial g}{\partial y} f_{YZZ} + f_{YYZ}}{f_{ZZ}}$$

where $f_{AB} = \dfrac{\partial f^2}{\partial A \partial B}(x, y, g(x, y))$ and $f_{ABC} = \dfrac{\partial f^3}{\partial A \partial B \partial C}(x, y, g(x, y))$.

Detailed proof can be seen in Appendix.

Lemma 4.2. *Let $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a continuous function with third derivatives. Let $h(x) = \arg\min_{y\in\mathbb{R}} f(x, y, g(x, y))$, then the derivative of h with respect to x is:*

$$\frac{\partial h}{\partial x} = -\frac{f_{XY} + \frac{\partial g}{\partial x} f_{YZ} + \frac{\partial g}{\partial y} f_{XZ} + \frac{\partial g}{\partial x}\frac{\partial g}{\partial y} f_{ZZ} + \frac{\partial^2 g}{\partial x \partial y} f_Z}{f_{YY} + 2\frac{\partial g}{\partial y} f_{YZ} + (\frac{\partial g}{\partial y})^2 f_{ZZ} + \frac{\partial^2 g}{\partial y^2} f_Z}$$

where $f_{AB} = \dfrac{\partial f^2}{\partial A \partial B}(x, h(x), g(x, h(x)))$, $f_Z = \dfrac{\partial f}{\partial Z}(x, h(x), g(x, h(x)))$, and all of the derivative of g is calculated at $(x, h(x))$.

Detailed proof can be seen in Appendix.

Now, we are ready to bring out the main theorem, which gives us the gradient of the upper problem, enabling us with all kinds of gradient-based optimization algorithm like Adam [10].

Theorem 4.3. *Let $f_1 : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a continuous function $f_1(x, y, z)$ with first-order derivatives, $f_2 : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a continuous function $f_2(x, y, z)$ with second-order derivative, $f_3 : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a continuous function $f_3(x, y, z)$ with third-order derivatives. Let $g(x, y) = \arg\min_{z\in\mathbb{R}} f_3(x, y, z)$, $h(x) = \arg\min_{y\in\mathbb{R}} f_2(x, y, g(x, y))$. Then the derivative of f_1 with respect to x can be written as:*

$$\frac{\mathrm{d}f_1}{\mathrm{d}x} = \frac{\partial f_1}{\partial x} + \frac{\partial h}{\partial x}\frac{\partial f_1}{\partial y} + (\frac{\partial g}{\partial x} + \frac{\partial g}{\partial y}\frac{\partial h}{\partial x})\frac{\partial f_1}{\partial z}$$

where all the derivatives are calculated at $y = h(x), z = g(x, h(x))$. The derivative of g and h can be seen in lemma 4.1 and lemma 4.2.

Detailed proof can be seen in Appendix.

5 Experiments

In the last section, we discuss how to compute the full gradient with respect to first input. On the other hand, it is still essential to check whether taking the full gradient surpasses the naive alternative of only taking the partial gradient with respect to the first input. Below, we will consider some social reasoning tasks and prove our methods can indeed achieve better performance.

5.1 Experimental Setting

To verify the efficiency of our proposed method, we conducted three experiments: the first experiment compares the effectiveness of our method and existing methods, the second experiment compare compares the performance under the settings of [15], and the third experiment compares the performance on the social reasoning problem we constructed. In all the experiments, we compare Vanilla Gradient Descent(VGD), Iterative Differentiation (ITD)[4], and our method. We implement all codes with Python 3.8.18 and JAX 0.4.13 for automatic differentiation and executed them on a computer with 8 cores of Apple M2 CPU, 8 GB RAM.

5.2 Experiment of Efficiency

To validate the effectiveness of our proposed method, we conduct numerical experiments on artificial problems.

Algorithm	VGD	ITD	Implicit Differentiation (ours)
Time per update	1	7	1.25

We find that the running efficiency of our method is very close to vanilla gradient descent, far exceeding iterative differentiation method.

5.3 Experimental of Simple Situation

In this experiment, we follow the situation in [15], where

$$f_1(x, y, z) = x^2 + (z - x)^2$$
$$f_2(x, y, z) = (y - x)^2$$
$$f_3(x, y, z) = (z - x)^2$$

Clearly, the optimal solution is $x = y = z = 0$. The performance of the three methods in this situation is shown in the figure 1.

Experimental results show that all the three methods are approximately the same. We consider that it is because of the selection of f_1, f_2, f_3 here is too simple, so we design the following social reasoning problem for the experiment.

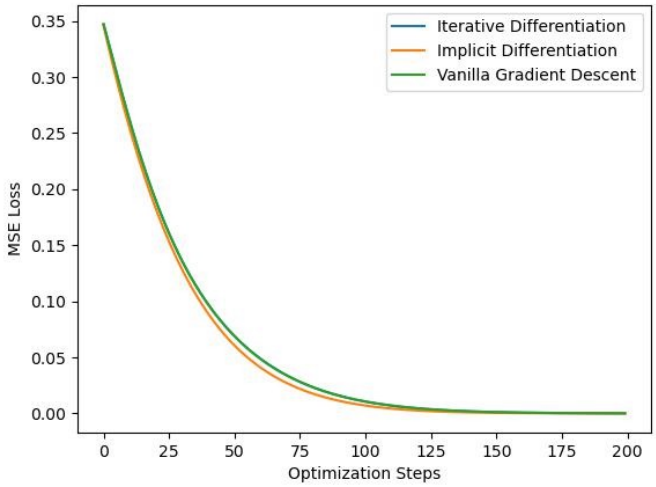


Figure 1: The performance in a simple situation

5.4 Experimental of Social Reasoning

Consider there are three generals all want to conquer a same city. Define the number of soldier each generals sent to the battlefield as x, y and z respectively. Sending troops will have a cost c, and the benefit one general can enjoy decrease with the number of soldiers from the enemy side. Formally, we can write down the utility functions of the three generals:

$$f_1(x, y, z) = x(1 - y - z) + c(x)$$
$$f_2(x, y, z) = y(1 - x - z) + c(y)$$
$$f_3(x, y, z) = z(1 - x - y) + c(z)$$

Assume that, under the hard work of spies, every generals know each other utility perfectly and they make decision in sequences.

Convergence rate comparison of tractable problems. Let $c(x) = x^2$, we can solve the optimal $x^* = 0.5$. We plot the mean absolute distance with 0.5 during training. Although theoretically when training to converge, partial gradient optimization algorithm will achieve global optimal, the experimental result shown below clearly illustrates full gradient optimization perform significant better than partial gradient.

Empirical Performance on intractable problems. Let $c(x) = exp(x) - 1$, then $g(x, y) = \arg\min_{z\in\mathbb{R}} f_3(x, y, z)$ and $h(x) = \arg\min_{y\in\mathbb{R}} f_2(x, y, g(x, y))$ do not have close-form solutions. In the below table, we plot $f_1(x, h(x), g(x, y))$ over training iterations.

Table 1: $f_1(x, h(x), g(x, y))$ over training iterations. Full gradient optimization consistently outperform the partial gradient alternative.

Algorithm	100	200	300	400	500
Full-Gradient	-0.0614	-0.0626	-0.0626	-0.0626	-0.0626
Partial-Gradient	-0.0601	-0.0599	-0.0598	-0.0598	-0.05986

Performance of different on social reasoning problems. To demonstrate the efficiency of our method, we repeat the experiment in more complex situation than that in [15]. We define

$$f_1(x, y, z) = -x(1 - y - z) + c(x)$$
$$f_2(x, y, z) = -y(1 - x - z) + c(y)$$
$$f_3(x, y, z) = -z(1 - x - y) + c(z)$$

where $c(x) = x^2$ for tractable problems and $c(x) = xe^x - x$ for intractable problems. The results are shown in figure 3. Through the results, we find that in more complex situations (whether tractable or intractable), our method outperforms Vanilla Gradient Descent and Iterative Descent, where the latter two methods converge to local optima.

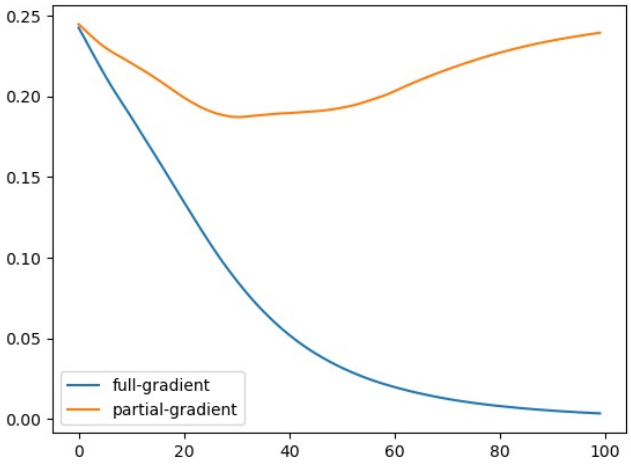
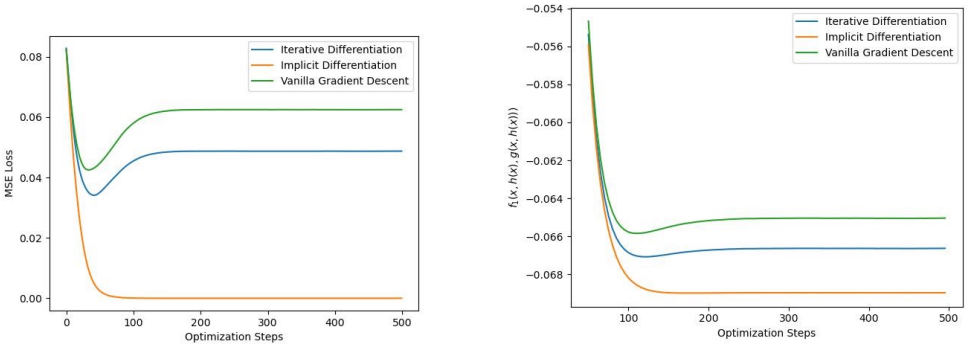


Figure 2: The mean absolute distance over training iterations.



(a) $c(x) = x^2$ (b) $c(x) = xe^x - x$

Figure 3: The performance on tractable and intractable problems

6 Discussion

6.1 Expansion of n-st-Level

Generally speaking, when there are n players, they will make decisions in a specific order, such as from player 1 x_1 to player n x_n. Their decision-making structure can also be seen as follower-leader pattern in a sense, but they may be seen as "leaders of different levels". In this case, the low-level players will give the optimal decisions when facing the fixed decisions of high-level players, while high-level players will choose their own optimal decisions based on the optimal decisions of low-level players and fixed decisions of higher-level players.

Moreover, we can extend theorem 4.3 to n variables:

Theorem 6.1. *Let $f_1 : \mathbb{R}^n \to \mathbb{R}$ be a continuous function $f_1(x_1, x_2, \cdots, x_n)$ with first-order partial derivatives, $f_2 : \mathbb{R}^n \to \mathbb{R}$ be a continuous function $f_2(x_1, x_2, \cdots, x_n)$ with second-order partial derivatives, and so on. $f_n : \mathbb{R}^n \to \mathbb{R}$ be a continuous function $f_n(x_1, x_2, \cdots, x_n)$ with n-st-order partial derivatives. Then the derivative of f_1 with respect to x_1 can be written as:*

$$\frac{\mathrm{d}f_1}{\mathrm{d}x_1} = \sum_{k=1}^{n} \frac{\partial f_1}{\partial x_k} \frac{\mathrm{d}x_k}{\mathrm{d}x_1} \tag{8}$$

The $\frac{\mathrm{d}x_k}{\mathrm{d}x_1}$ *in formula 8 can be obtained from*

$$\frac{\mathrm{d}x_k}{\mathrm{d}x_1} = \sum_{j<k} \frac{\partial x_k}{\partial x_j} \frac{\mathrm{d}x_j}{\mathrm{d}x_1}$$

recursively, where $x_2, x_3, \cdots, x_n$ *are determined by the following system of equations:*

$$\begin{cases} x_n^\star = x_n(x_1, \cdots, x_{n-1}) = \arg\min_{x_n} f_n(x_1, \cdots, x_n) \\ x_{n-1}^\star = x_{n-1}(x_1, \cdots, x_{n-2}) = \arg\min_{x_{n-1}} f_{n-1}(x_1, \cdots, x_n^\star) \\ \vdots \\ x_3^\star = x_3(x_1, x_2) = \arg\min_{x_3} f_3(x_1, \cdots, x_n^*) \\ x_2^\star = x_2(x_1) = \arg\min_{x_2} f_2(x_1, x_2, x_3^\star \cdots, x_n^\star) \end{cases}$$

Then $\frac{\partial x_k}{\partial x_j}$ *can also be further derived with a procedure greatly resemble lemma 4.1 and lemma 4.2.*

In fact, the above equation system is equivalent to $\frac{\mathrm{d}f_i}{\mathrm{d}x_i} = 0, \forall\ 2 \leq i \leq n$ for fixed $x_j (j < i)$ and optimal $x_k (k > i)$.

6.2 Convergence Analysis For Tri-level

When analyzing the convergence of gradient-based optimization algorithm for tri-level problems, we find it difficult to estimate the difference between the f_1 value in step N and the optimal value. So we switched to analyzing the average of the derivatives, which is to prove that the gradient $\frac{\mathrm{d}f_1}{\mathrm{d}x}$ will become smaller and smaller during the update process of x.

For simplicity, we assume that f_1, f_2, f_3 satisfy all the conditions of theorem 4.3 and the second-order derivative of $f_1(x, y, z)$ with respect to x is bounded, i.e.

$$\left|\frac{\mathrm{d}^2 f_1}{\mathrm{d}x^2}\right| \leq M$$

We have

$$\mathbb{E}\left(\frac{\mathrm{d}f_1}{\mathrm{d}x}\right)^2 \sim \mathcal{O}\left(\frac{1}{N}\right) \tag{9}$$

Detailed proof of formula 9 can be seen in Appendix. In this way, we know that as the number of update rounds increases, the expected gradient gradually decreases, which represents the convergence of the algorithm.

6.3 Conclusion

Overall, we have proposed and implemented an algorithm for the tri-level problem based on implicit differentiation in this project. When the entire problem has an optimal solution (it is correct most of the time for social reasoning problems, but may require f_i to be convex for general situation), our algorithm can converge to the optimal value, which goes beyond previous work. We have demonstrated the efficiency of our method on some artificial problems, and we have also demonstrated that this algorithm has convergence.

6.4 Future Work

The future work can be roughly divided into two parts:

The first part is the number of people making decisions at each level in tri-level or n-st-level problems. In our previous discussion, there was only one person making decisions at each level, but we know that in reality, there may be more than one person at each level, which means there may be multiple "leaders" working and deciding together at each level. This means that the x, y, z we discussed above

will become vectors, and we will provide corresponding gradient theorems and algorithms in this case later.

The other part is a more accurate convergence conclusion. In this project, we have set a very strong condition for the second-order derivative of f_1 with respect to x to be bounded, in order to obtain our conclusion. We will point out later that there is a relationship between the optimal M and Lipschitz constants of the derivatives of f_1, f_2, f_3. so we will weaken our conditions to obtain better convergence conclusions.

References

[1] John B Bell. Solutions of ill-posed problems., 1978. 2

[2] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019. 3

[3] Stephan Dempe and Alain Zemkoho. Bilevel optimization. In *Springer optimization and its applications*, volume 161. Springer, 2020. 1

[4] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pages 1568–1577. PMLR, 2018. 1, 5

[5] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016. 3

[6] John R Hicks. Marktform und gleichgewicht, 1935. 2

[7] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023. 1

[8] Ruichen Jiang, Nazanin Abolfazli, Aryan Mokhtari, and Erfan Yazdandoost Hamedani. A conditional gradient-based method for simple bilevel optimization with convex lower-level problem. In *International Conference on Artificial Intelligence and Statistics*, pages 10305–10323. PMLR, 2023. 1

[9] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013. 3

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 3

[12] Zhize Li and Jian Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. *Advances in neural information processing systems*, 31, 2018. 3

[13] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017. 3

[14] Kegan GG Samuel and Marshall F Tappen. Learning optimized map estimates in continuously-valued mrf models. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 477–484. IEEE, 2009. 3

[15] Ryo Sato, Mirai Tanaka, and Akiko Takeda. A gradient method for multilevel optimization. *Advances in Neural Information Processing Systems*, 34:7522–7533, 2021. 5, 6

[16] Quoc Tran-Dinh, Nhan H Pham, Dzung T Phan, and Lam M Nguyen. Hybrid stochastic gradient descent algorithms for stochastic nonconvex optimization. *arXiv preprint arXiv:1905.05920*, 2019. 3

[17] Shengjian Wu. Mathematical analysis. pages 80–83, 2019. 11

[18] Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682, 2021. 3

A Appendix

Lemma A.1. *4.1 Let $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a continuous function with third derivatives. Let $g(x, y) = \arg\min_{z\in\mathbb{R}} f(x, y, z)$, then the following properties about g holds:*
(a)

$$\frac{\partial g}{\partial x} = -\frac{f_{XZ}}{f_{ZZ}}$$

(b)

$$\frac{\partial g}{\partial y} = -\frac{f_{YZ}}{f_{ZZ}}$$

(c)

$$\frac{\partial^2 g}{\partial x \partial y} = -\frac{\frac{\partial g}{\partial x}\frac{\partial g}{\partial y} f_{ZZZ} + \frac{\partial g}{\partial y} f_{XZZ} + \frac{\partial g}{\partial x} f_{YZZ} + f_{XYZ}}{f_{ZZ}}$$

(d)

$$\frac{\partial^2 g}{\partial y^2} = -\frac{(\frac{\partial g}{\partial x})^2 f_{ZZZ} + 2\frac{\partial g}{\partial y} f_{YZZ} + f_{YYZ}}{f_{ZZ}}$$

where $f_{AB} = \dfrac{\partial f^2}{\partial A \partial B}(x, y, g(x, y))$ and $f_{ABC} = \dfrac{\partial f^3}{\partial A \partial B \partial C}(x, y, g(x, y))$.

Proof. Since $g(x, y) = \arg\min_z f(x, y, z)$, $\dfrac{\partial f}{\partial z} = 0$.

Differentiate with respect to x, we get $\dfrac{\mathrm{d}}{\mathrm{d}x}\dfrac{\partial f}{\partial z} = f_{XZ} + f_{ZZ}\dfrac{\partial g}{\partial x} = 0$.

Then $\dfrac{\partial g}{\partial x} = -\dfrac{f_{XZ}}{f_{ZZ}}$.

Similarly, we can easily obtain $\dfrac{\partial g}{\partial y} = -\dfrac{f_{YZ}}{f_{ZZ}}$.

Also, we can have $\dfrac{\mathrm{d}^2}{\mathrm{d}x\,\mathrm{d}y}\dfrac{\partial f}{\partial z} = 0$

Results in $\dfrac{\partial^2 g}{\partial x \partial y} f_{ZZ} + \dfrac{\partial g}{\partial y}[f_{XZZ} + f_{ZZZ}\dfrac{\partial g}{\partial x}] + f_{XYZ} + f_{YZZ}\dfrac{\partial g}{\partial x} = 0$ Summarizing the results, we have $\dfrac{\partial^2 g}{\partial x \partial y} = -\dfrac{\frac{\partial g}{\partial x}\frac{\partial g}{\partial y} f_{ZZZ} + \frac{\partial g}{\partial y} f_{XZZ} + \frac{\partial g}{\partial x} f_{YZZ} + f_{XYZ}}{f_{ZZ}}$.

Similarly, we have $\dfrac{\partial^2 g}{\partial y^2} = -\dfrac{(\frac{\partial g}{\partial x})^2 f_{ZZZ} + 2\frac{\partial g}{\partial y} f_{YZZ} + f_{YYZ}}{f_{ZZ}}$ □

Lemma A.2. *4.2 Let $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a continuous function with third derivatives. Let $h(x) = \arg\min_{y\in\mathbb{R}} f(x, y, g(x, y))$, then the derivative of h with respect to x is:*

$$\frac{\partial h}{\partial x} = -\frac{f_{XY} + \frac{\partial g}{\partial x} f_{YZ} + \frac{\partial g}{\partial y} f_{XZ} + \frac{\partial g}{\partial x}\frac{\partial g}{\partial y} f_{ZZ} + \frac{\partial^2 g}{\partial x \partial y} f_Z}{f_{YY} + 2\frac{\partial g}{\partial y} f_{YZ} + (\frac{\partial g}{\partial y})^2 f_{ZZ} + \frac{\partial^2 g}{\partial y^2} f_Z}$$

where $f_{AB} = \dfrac{\partial f^2}{\partial A \partial B}(x, h(x), g(x, h(x)))$, $f_Z = \dfrac{\partial f}{\partial Z}(x, h(x), g(x, h(x)))$, and all of the derivative of g is calculated at $(x, h(x))$.

Proof. Similarly from the previous proof, $\dfrac{\mathrm{d}}{\mathrm{d}x}\dfrac{\mathrm{d}f}{\mathrm{d}y} = 0$

Then we have $\dfrac{\mathrm{d}}{\mathrm{d}x}[\dfrac{\partial f}{\partial y} + \dfrac{\partial f}{\partial z}\dfrac{\partial g}{\partial y}] = f_{XZ} + f_{YY}\dfrac{\partial h}{\partial x} + f_{YZ}[\dfrac{\partial g}{\partial x} + \dfrac{\partial g}{\partial y}\dfrac{\partial h}{\partial x}] + \dfrac{\partial g}{\partial y}[f_{XZ} + f_{YZ}\dfrac{\partial h}{\partial x} +$

$$f_{ZZ}(\frac{\partial g}{\partial x}+\frac{\partial g}{\partial y}\frac{\partial h}{\partial x})]+f_Z(\frac{\partial^2 g}{\partial x\partial y}+\frac{\partial^2 g}{\partial y^2}\frac{\partial h}{\partial x})=0$$

Summarize the results, we have $\dfrac{\partial h}{\partial x}=-\dfrac{f_{XY}+\frac{\partial g}{\partial x}f_{YZ}+\frac{\partial g}{\partial y}f_{XZ}+\frac{\partial g}{\partial x}\frac{\partial g}{\partial y}f_{ZZ}+\frac{\partial^2 g}{\partial x\partial y}f_Z}{f_{YY}+2\frac{\partial g}{\partial y}f_{YZ}+(\frac{\partial g}{\partial y})^2 f_{ZZ}+\frac{\partial^2 g}{\partial y^2}f_Z}$ □

The proof of lemma 3.1 The detailed proof is from [17]. Here we strengthen $F(x,y)$ is continuous in $U((x_0,y_0),\delta)$ to $F'_x(x,y)$ is continuous in $U((x_0,y_0),\delta)$ to ensure that $f(x)$ has a continuous derivative.

The proof of theorem 4.3 Let $f_1:\mathbb{R}\times\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ be a continuous function $f_1(x,y,z)$ with first-order derivatives, $f_2:\mathbb{R}\times\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ be a continuous function $f_2(x,y,z)$ with second-order derivative, $f_3:\mathbb{R}\times\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ be a continuous function $f_3(x,y,z)$ with third-order derivatives. Let $g(x,y)=\arg\min_{z\in\mathbb{R}} f_3(x,y,z), h(x)=\arg\min_{y\in\mathbb{R}} f_2(x,y,g(x,y))$. Then the derivative of f_1 with respect to x can be written as:

$$\frac{\mathrm{d}f_1}{\mathrm{d}x}=\frac{\partial f_1}{\partial x}+\frac{\partial h}{\partial x}\frac{\partial f_1}{\partial y}+(\frac{\partial g}{\partial x}+\frac{\partial g}{\partial y}\frac{\partial h}{\partial x})\frac{\partial f_1}{\partial z}$$

where all the derivatives are calculated at $y=h(x), z=g(x,h(x))$. The derivative of g and h can be seen in lemma 4.1 and lemma 4.2.

Proof. Here $y=h(x)$ and $z=g(x,h(x))$ should represent the optimal response considered in the tri-level problems, so when calculating the derivative of f_1 with respect to x, we should note that both y and z are functions of x. Therefore, according to the chain rule, we have

$$\begin{aligned}\frac{\mathrm{d}f_1}{\mathrm{d}x}(x,y,z)&=\frac{\partial f_1}{\partial x}+\frac{\partial f_1}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x}+\frac{\partial f_1}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}x}\\&=\frac{\partial f_1}{\partial x}+\frac{\partial h}{\partial x}\frac{\partial f_1}{\partial y}+(\frac{\partial g}{\partial x}+\frac{\partial g}{\partial y}\frac{\partial h}{\partial x})\frac{\partial f_1}{\partial z}\end{aligned}$$

□

The proof of formula 9 Let f_1,f_2,f_3 satisfy the conditions of theorem 4.3 and

$$|\frac{\mathrm{d}^2 f_1}{\mathrm{d}x^2}|\le M$$

we will have

$$\mathbb{E}(\frac{\mathrm{d}f_1}{\mathrm{d}x})^2\sim\frac{1}{N}\sum_{i=0}^{N-1}(\frac{\mathrm{d}f_1}{\mathrm{d}x}|_{x=x_i})^2=\mathcal{O}(\frac{1}{N})$$

Proof. The basic proof approach is to compute the difference of f_1 value between two steps using Lagrange mean value theorem:

$$\begin{aligned}&f_1(x_{i+1},y^\star(x_{i+1}),z^\star(x_{i+1},y^\star(x_{i+1})))-f_1(x_i,y^\star(x_i),z^\star(x_i,y^\star(x_i)))\\=&(x_{i+1}-x_i)\frac{\mathrm{d}f_1}{\mathrm{d}x}(x_i,y^\star(x_i),z^\star(x_i,y^\star(x_i)))+\frac{1}{2}(x_{i+1}-x_i)^2\frac{\mathrm{d}^2 f_1}{\mathrm{d}x^2}(\xi,y^\star(\xi),z^\star(\xi,y^\star(\xi)))\end{aligned}$$

where $y^\star$ and $z^\star$ are the optimal value as discussed and ξ is some value between x_i and x_{i+1}. Note that we update x by

$$x=x-\beta\frac{\mathrm{d}f_1}{\mathrm{d}x}(x,y,z)$$

and the second-order derivative of f_1 with respect to x is bounded, we can replace $x_{i+1}-x_i$ and the second-order derivative above:

$$\begin{aligned}&(x_{i+1}-x_i)\frac{\mathrm{d}f_1}{\mathrm{d}x}(x_i,y^\star(x_i),z^\star(x_i,y^\star(x_i)))+\frac{1}{2}(x_{i+1}-x_i)^2\frac{\mathrm{d}^2 f_1}{\mathrm{d}x^2}(\xi,y^\star(\xi),z^\star(\xi,y^\star(\xi)))\\\le&-\beta(\frac{\mathrm{d}f_1}{\mathrm{d}x}(x_i,y^\star(x_i),z^\star(x_i,y^\star(x_i))))^2+\frac{\beta^2 M}{2}(\frac{df_1}{dx}(x_i,y^\star(x_i),z^\star(x_i,y^\star(x_i))))^2\\=&(\frac{\beta^2 M}{2}-\beta)(\frac{\mathrm{d}f_1}{\mathrm{d}x}(x_i,y^\star(x_i),z^\star(x_i,y^\star(x_i))))^2\end{aligned}$$

So we get

$$\begin{aligned}
&f_1(x_{i+1}, y^\star(x_{i+1}), z^\star(x_{i+1}, y^\star(x_{i+1}))) - f_1(x_i, y^\star(x_i), z^\star(x_i, y^\star(x_i))) \\
\leq&(\frac{\beta^2 M}{2} - \beta)(\frac{\mathrm{d}f_1}{\mathrm{d}x}(x_i, y^\star(x_i), z^\star(x_i, y^\star(x_i))))^2
\end{aligned}$$

After summing both sides, we find that the sum of the squared derivatives for each step can be bounded by a constant which is related to the initial point we select:

$$\begin{aligned}
&(\beta - \frac{\beta^2 M}{2}) \sum_{i=0}^{N-1} (\frac{\mathrm{d}f_1}{\mathrm{d}x}(x_i, y^\star(x_i), z^\star(x_i, y^\star(x_i))))^2 \\
\leq& f_1(x_0, y^\star(x_0), z^\star(x_0, y^\star(x_0))) - f_1(x_N, y^\star(x_N), z^\star(x_N, y^\star(x_N))) \\
\leq& f_1(x_0, y^\star(x_0), z^\star(x_0, y^\star(x_0)))
\end{aligned}$$

So when $\beta \leq \frac{2}{M}$, the expectation of the squared derivative can be approximated as

$$\mathbb{E}(\frac{\mathrm{d}f_1}{\mathrm{d}x})^2 \sim \frac{1}{N} \sum_{i=0}^{N-1} (\frac{\mathrm{d}f_1}{\mathrm{d}x}\big|_{x=x_i})^2 \leq \frac{1}{N}(\beta - \frac{\beta^2 M}{2})^{-1} f_1\big|_{x=x_0} = \mathcal{O}(\frac{1}{N})$$

by Central Limit Theorem.

□