

# Task Embedded Coordinate Update: A Realizable Framework for Multivariate Non-Convex Optimization

Yiyang Wang,<sup>1,2</sup> Risheng Liu,<sup>3,4,\*</sup> Long Ma,<sup>3,4</sup> Xiaoliang Song<sup>5</sup>

<sup>1</sup>Institute of Atmospheric Sciences, Fudan University

<sup>2</sup>Shanghai Ecological Forecasting and Remote Sensing Center

<sup>3</sup>DUT-RU International School of Information Science & Engineering, Dalian University of Technology

<sup>4</sup>Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology

<sup>5</sup>Department of Applied Mathematics, The Hong Kong Polytechnic University

yywerica@fudan.edu.cn rslu@dlut.edu.cn malone94319@gmail.com xiaoliang.song@polyu.edu.hk

## Abstract

We in this paper propose a realizable framework TECU, which embeds task-specific strategies into update schemes of coordinate descent, for optimizing multivariate non-convex problems with coupled objective functions. On one hand, TECU is capable of improving algorithm efficiencies through embedding productive numerical algorithms, for optimizing univariate sub-problems with nice properties. From the other side, it also augments probabilities to receive desired results, by embedding advanced techniques in optimizations of realistic tasks. Integrating both numerical algorithms and advanced techniques together, TECU is proposed in a unified framework for solving a class of non-convex problems. Although the task embedded strategies bring inaccuracies in sub-problem optimizations, we provide a realizable criterion to control the errors, meanwhile, to ensure robust performances with rigid theoretical analyses. By respectively embedding ADMM and a residual-type CNN in our algorithm framework, the experimental results verify both efficiency and effectiveness of embedding task-oriented strategies in coordinate descent for solving practical problems.

## Introduction

Over the past few decades, multivariate non-convex optimization has been widely concerned in the realms of pattern recognition and machine learning. Achieving well performances in the tasks such as matrix factorization (Vu and Monga 2016; Bao et al. 2016) and image enhancement task (Fu et al. 2016; Gharbi et al. 2017), multivariate non-convex problems have motivated a revived interest in designing and analyzing numerical algorithms.

Compared to univariate optimization, it is much more complicated to optimize multivariate problems with coupled objective functions. Taking two variables as instance, this kind of coupled problem can be formulated as:

$$\min_{\mathbf{z}:=\{\mathbf{x},\mathbf{y}\}} \Psi(\mathbf{z}) := f(\mathbf{x}) + g(\mathbf{y}) + H(\mathbf{x}, \mathbf{y}), \quad (1)$$

\*Corresponding author.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

with vectors/matrices  $\mathbf{x}$  and  $\mathbf{y}$ . Being employed to varieties of tasks, the coordinate descent (CD) (Luo and Tseng 1989) is widely used for solving problem (1), which optimizes the objective over each direction, while fixing the remaining one with its latest value, i.e., solving univariate optimization problems in a loop. Doing in this way, calculating the coordinate updates are much simpler than computing a full update, requiring less memory and computational cost. However, in addition to these benefits, few CD algorithms consider useful traits of univariate sub-problems for improving either convergence speeds or optimized results for solving the generic problem (1) with non-convex, non-smooth objective function.

In most cases, though  $\Psi(\mathbf{z})$  is non-convex and even non-smooth, it is quite likely to have univariate sub-problems with nice properties: e.g., the sub-problems can be optimized via convex optimization, or may have unique solutions. Moreover, the univariate problems usually have entirely distinct formations referred to different variables. For example, many literatures have posted superiorities on restricting dictionaries with normalized bases, meanwhile, constraining sparsity for the codes with various non-convex penalties for dictionary learning tasks (Gregor and Lecun 2010; Wang et al. 2016; Bao et al. 2016). Though these models are non-convex and non-smooth, their univariate sub-problems of dictionary can be efficiently solved, compared with the other sub-problem. Thus, in view of the nice traits and specificities of univariate problems, it is significant to specifically integrate effective algorithms for optimizing task-specific sub-problems, to improve the efficiency and effectiveness of CD schemes.

More critically, though there are not a few CD algorithms for solving multivariate optimization problems, converging to a critical point is still a nice result for generic non-convex and non-smooth problems (Bolte, Sabach, and Teboulle 2014; Xu and Yin 2015; Pock and Sabach 2017). While, we have noticed that, many univariate sub-problems of real-world image processing tasks are referred to specific application problems. E.g., in tasks like image deblurring and super-resolution, one univariate sub-problem can be regarded as an image denoising task. Rather than numerical algorithms, techniques such

as BM3D and CNN are effective for solving image denoising problem. Although such advanced techniques mostly lack theoretical support, they have the ability to efficiently project the variables on small neighborhoods of the desired solutions. Considering the effectiveness of these advanced techniques, it is significant to integrate them into CD schemes, expecting to get desired results with high probability (Zhang et al. 2017; Chan, Wang, and Elgandy 2017).

The above mentioned strategies, integrating either numerical algorithms or advanced techniques, have already appeared in applications for specific tasks, which will be briefly stated later. However, the success of those CD schemes, designed for specific problems, can not be straightforwardly replicated to other tasks. Moreover, there has not yet been proposed a unified CD framework, integrating both numerical algorithms and advanced techniques, for optimizing the generic multivariate non-convex problem (1). More importantly, few of them are able to provide rigid theoretical analyses on illuminating the properties of the final optimized results. Considering all the mentioned aspects, we in this paper propose a realizable algorithm framework, which embeds various task-oriented strategies in the update of CD scheme, for effectively solving the generic problem (1). We name our proposed algorithm as TECU (task embedded coordinate update), and the main contributions are sketched out as follows:

1. For optimizing the generic multivariate problem (1) with coupled non-convex objective, we propose an algorithm TECU, which embeds task-oriented techniques for optimizing specific univariate sub-problems of CD update. Moreover, we further provide a realizable condition to ensure robust performances of TECU with theoretical analyses.
2. Considering the nice properties of univariate sub-problems, TECU is able to improve the algorithm efficiency by embedding high-efficient numerical algorithms into its framework. We utilize the  $\ell_0$ -regularized dictionary learning task and design to embed ADMM to accelerate the convergence speed of the whole algorithm. Experiments conducted on synthetic data give verifications on the efficiency of TECU, in comparison with other existing numerical algorithms.
3. Through embedding advanced techniques, TECU is likely to obtain desired solutions with high probability, which is superior to most numerical algorithms for non-convex optimization. Taking low-light image enhancement as an example, we embed a residual-type CNN to optimize the univariate problem of illumination layer. Then, comparing to state-of-the-art methods, the experimental results show the superiority of embedding networks for real-world tasks, meanwhile, verify the effectiveness of integrating networks and CD schemes in a unified algorithm framework.

## Related Work

For solving general multivariate non-convex problems, the most classical case that adopts CD scheme is the proximal alternating method (PAM) (Attouch et al. 2010). However, it is limited to most coupled problems for requiring explicit solutions for every univariate sub-problems. To get around this limitation, the PALM linearizes the coupled function, in pursuit of explicit solutions (Bolte, Sabach, and Teboulle

2014). However, it requires computing exact Lipschitz constants during iterations, which sometimes is time-consuming even for estimating their tight upper bounds (Bao et al. 2016; Xu and Yin 2015). Moreover, improper upper bounds definitely slow down the convergence speeds of PALM. These troubles on estimating Lipschitz constants also exist in CD variants like BCU (Xu and Yin 2017) and iPALM (Pock and Sabach 2017). Besides the mentioned defects, the updates of existing CD algorithms utterly lose sight of task specificities, i.e., optimizing every univariate sub-problems in the same scheme, which is less efficient in practice.

Unlike algorithms for general problems, it is common to embed numerical algorithms for optimizing sub-problems in real-world applications (Guo and Ma 2014; Li and Brown 2014; Li et al. 2016; Yue et al. 2017). Such algorithms often make good uses on the nice traits of univariate sub-problems, thus they always possess high efficiencies and superior performances. However, their specificities give rise to less generalization: the well-designed algorithms usually cannot be borrowed to other models. Not only this, those specified algorithms mostly have relatively weak convergence in theory, thus their efficiencies are mostly lack of robustness.

Quite recently, fusing advanced techniques into optimization framework has been a hot research interest for real-world applications (Schmidt and Roth 2014; Liu et al. 2018; Zhang et al. 2017). For example, the authors in (Schmidt and Roth 2014) learn a cascade of shrinkage fields to replace artificially designed priors in a half-quadratic optimization for image restoration. Instead of designing complex regularizers, (Zhang et al. 2017) learns a CNN-based denoiser to replace corresponding sub-problem in their optimization framework. These novel methods usually have remarkable performances with the power of advanced techniques, however, their successes rely on completely replacing univariate sub-problems with advanced techniques, thus few of them are able to illuminate the properties on final results with rigid theoretical analyses.

## Preliminaries

In general, the objective function of problem (1) is assumed to have: (1)  $f$  and  $g$  are proper, lower semi-continuous (l.s.c); (2)  $H$  is a  $C^1$  function; its gradient and partial gradients are Lipschitz continuous on bounded sets; (3)  $\Psi$  is coercive, that is, it is bounded from below and  $\Psi(\mathbf{z}) \rightarrow \infty$  when  $\|\mathbf{z}\| \rightarrow \infty$ , where  $\|\cdot\|$  denotes the Frobenius norm. Meanwhile,  $\Psi$  is a Kurdyka-Łojasiewicz (KŁ) function.

Notice that, all semialgebraic functions and subanalytic functions satisfy the KŁ property. Typical semialgebraic functions include real polynomial functions,  $\|\cdot\|_p$  with  $p \geq 0$ , indicator functions of semialgebraic sets, Stiefel manifolds and constant rank matrices (Attouch et al. 2010).

## Task Embedded Coordinate Update

Corresponding to specific tasks, the univariate sub-problems of the generic model (1) usually either have desirable characteristics or can be corresponding to certain tasks with single variable. Considering these available traits, we embed powerful strategies in CD scheme, for optimizing task-oriented

---

**Algorithm 1** Task Embedded Coordinate Update

---

- 1: Set constants  $C_x, C_y, \eta_1, \eta_2$ .
- 2: Initialize variables  $\mathbf{x}^0, \mathbf{y}^0$ .
- 3: **while** not converged **do**
- 4:  $\mathbf{x}^t \leftarrow$  either  $\psi(\mathbf{x}; \mathbf{x}^{t-1}, \mathbf{y}^{t-1}, f, H, \eta_1, C_x, \epsilon_x^t)$  or the updates in Table 1.
- 5:  $\mathbf{y}^t \leftarrow$  either  $\psi(\mathbf{y}; \mathbf{y}^{t-1}, \mathbf{x}^t, g, H, \eta_2, C_y, \epsilon_y^t)$  or the updates in Table 1.
- 6: **if**  $t \geq 2$  **then**
- 7:  $\epsilon_x^t = \|\mathbf{x}^{t-1} - \mathbf{x}^{t-2}\|$  and  $\epsilon_y^t = \|\mathbf{y}^{t-1} - \mathbf{y}^{t-2}\|$ .
- 8: **end if**
- 9: **end while**

---

---

**Algorithm 2**  $\psi(\mathbf{u}; \mathbf{u}^{t-1}, \mathbf{v}, h, H, \eta, C_u, \epsilon_u^t)$ 

---

- 1: Initialize  $\mathbf{u}^{t,0} = \mathbf{u}^{t-1}$ .
- 2: **while**  $\|\mathbf{e}_u^{t,i}\| > C_u \epsilon_u^t$  **do**
- 3: Update  $\mathbf{u}^{t,i} = \mathcal{A}_u^i(\mathbf{u}^{t,i-1})$ .
- 4: Calculate  $\tilde{\mathbf{u}}^{t,i}$  and inexact error  $\mathbf{e}_u^{t,i}$  by Eq. (7).
- 5: **end while**
- 6:  $\mathbf{u}^t = \tilde{\mathbf{u}}^{t,i}$ . Output  $\mathbf{u}^t$ .

---

univariate sub-problems, and then improving the convergence speeds and optimized results of the whole algorithm.

The most basic CD scheme optimizes the objective  $\Psi(\mathbf{z})$  cyclically over each variable, that is, successively solving the following sub-problems to update  $\mathbf{x}^t$  and  $\mathbf{y}^t$  at iteration  $t$ .

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}) + H(\mathbf{x}, \mathbf{y}^{t-1}) + \frac{\eta_1}{2} \|\mathbf{x} - \mathbf{x}^{t-1}\|^2, \\ \min_{\mathbf{y}} g(\mathbf{y}) + H(\mathbf{x}^t, \mathbf{y}) + \frac{\eta_2}{2} \|\mathbf{y} - \mathbf{y}^{t-1}\|^2. \end{aligned} \quad (2)$$

Targeting to these univariate sub-problems, we respectively introduce numerical algorithms and advanced techniques to optimize them with a practical error control condition, which provides a criteria on optimization precisions, meanwhile, helps illuminating the properties of final optimized results.

### Task Embedded Strategies

Owing to the diversities of realistic tasks, the univariate sub-problems in Eq. (2) usually have distinct objective functions. Taking their specificities into consideration, we introduce two targeted strategies, i.e., numerical algorithms and advanced techniques, for optimizing these univariate sub-problems.

**Numerical Algorithms Embedding** Mostly, it makes no sense to employ extra numerical algorithms for optimizing sub-problems in Eq. (2) if their explicit solutions are easily obtained. However, for most cases with coupled  $H$ , it is common to adopt linearization for easy-to-solve sub-problems. But as mentioned in the section of related work, the linearization skill requires estimating Lipschitz constants during every iteration, which brings a series of time-consuming troubles.

We have noticed that, though the objective function  $\Psi(\mathbf{z})$  is non-convex and non-smooth, sub-problems corresponding to specific tasks may possess nice traits, e.g. they are convex sometimes even differentiable problems, or they have unique solutions. Thus there are plenty of high-efficient numerical

algorithms designed for solving such univariate problems, e.g., greedy algorithms (Elad 2010), PCG (Spillane 2016), FISTA (Kim and Fessler 2018) and ADMM (Wang, Yin, and Zeng 2018). Thus it is advantaged to embed efficient algorithms in the CD scheme, for improving the efficiency and effectiveness of the whole algorithm, which is one of the motivations for proposing TECU.

**Advanced Techniques Embedding** For real-world image processing tasks, most univariate sub-problems can be corresponding to specific applications. For example, in tasks of image deblurring and super-resolution, one univariate sub-problem can be seen as image denoising task (Chan, Wang, and Elgandy 2017; Zhang et al. 2017). While for low-light image enhancement, one of two sub-problems is for estimating the illumination layer, while the other one is for restoring the reflection image.

Except for optimization algorithms, there have been plenty advanced techniques such like BM3D (Chan, Wang, and Elgandy 2017) and variants of neural networks (Zhang et al. 2017; Gharbi et al. 2017) for solving single-variable tasks. Different from numerical algorithms, advanced techniques like neural networks, obtain the final results by a pre-trained propagations, rather than optimizing mathematical models. Such advanced techniques are mostly quite efficient, meanwhile, are able to propagate variables very close to the desired solutions. Taking these advantages into consideration, we propose to embed advanced techniques into CD scheme, to improve the convergence speeds of the whole algorithm, meanwhile, to get desired solutions with high probability.

Targeting to embed the above mentioned strategies into a unified framework, we adopt  $K_x^t$  and  $K_y^t$  steps of updates, starting at  $\mathbf{x}^{t-1}$  and  $\mathbf{y}^{t-1}$ , for optimizing problems of Eq. (2) to some extents, namely:

$$\begin{aligned} \mathbf{x}^{t,K_x^t} &= \mathcal{A}_x^{K_x^t} \circ \mathcal{A}_x^{K_x^t-1} \cdots \mathcal{A}_x^1 \circ \mathcal{A}_x^0(\mathbf{x}^{t-1}), \\ \mathbf{y}^{t,K_y^t} &= \mathcal{A}_y^{K_y^t} \circ \mathcal{A}_y^{K_y^t-1} \cdots \mathcal{A}_y^1 \circ \mathcal{A}_y^0(\mathbf{y}^{t-1}), \end{aligned} \quad (3)$$

where  $\circ$  denotes the composition operator. Each  $\mathcal{A}_u^i$  can be set as either one-step iteration of numerical algorithms or a propagation of advanced techniques.

This general framework covers various existing methods (Yue et al. 2017; Schmidt and Roth 2014; Zhang et al. 2017): e.g., for solvers like (Yue et al. 2017), each  $\mathcal{A}_u^i$  can be seen as one-step iteration of numerical algorithms; for others like (Zhang et al. 2017), there exists only one step propagation of advanced techniques. Besides, Eq. (3) also includes cases that employ numerical algorithms and advanced techniques in hybrid manners, which is far more flexible than existing solvers. Furthermore, considering two scenarios: 1) one problem of Eq. (2) has closed-form solution; 2) not all the sub-problems have efficient numerical algorithms, thus we propose TECU in a hybrid updating scheme (as shown in Alg. 1) with other two classical CD updates (see Table 1).

### Error Control and Estimation

Notice that our task embedded strategies do not require exactly optimizing problems in Eq. (2): updating  $\mathbf{x}^t$  and  $\mathbf{y}^t$

Table 1: Two classical CD updates for optimizing problem (1). Moreover, parameters should satisfy  $\zeta_1^t > 0$ ,  $\zeta_2^t > 0$ ,  $\gamma_1^t > L_1^t$  and  $\gamma_2^t > L_2^t$ , where  $L_1^t$  and  $L_2^t$  are Lipschitz constants of  $\nabla_x H(\mathbf{x}, \mathbf{y}^t)$  and  $\nabla_y H(\mathbf{x}^{t+1}, \mathbf{y})$ .

	Proximal update	Prox-linear update
$\mathbf{x}^{t+1} \in \arg \min_{\mathbf{x}}$	$f(\mathbf{x}) + H(\mathbf{x}, \mathbf{y}^t) + \frac{\zeta_1^t}{2} \ \mathbf{x} - \mathbf{x}^t\ ^2$	$f(\mathbf{x}) + \frac{\gamma_1^t}{2} \ \mathbf{x} - (\mathbf{x}^t - \nabla_x H(\mathbf{x}^t, \mathbf{y}^t)/\gamma_1^t)\ ^2$
$\mathbf{y}^{t+1} \in \arg \min_{\mathbf{y}}$	$g(\mathbf{y}) + H(\mathbf{x}^{t+1}, \mathbf{y}) + \frac{\zeta_2^t}{2} \ \mathbf{y} - \mathbf{y}^t\ ^2$	$g(\mathbf{y}) + \frac{\gamma_2^t}{2} \ \mathbf{y} - (\mathbf{y}^t - \nabla_y H(\mathbf{x}^{t+1}, \mathbf{y}^t)/\gamma_2^t)\ ^2$

by task embedded strategies bring errors  $\mathbf{e}_x^t$  and  $\mathbf{e}_y^t$  to the first-order optimality conditions of univariate sub-problems:

$$\begin{aligned} \mathbf{e}_x^t &= \mathbf{g}_x^t + \nabla_x H(\mathbf{x}^t, \mathbf{y}^{t-1}) + \eta_1(\mathbf{x}^t - \mathbf{x}^{t-1}), \\ \mathbf{e}_y^t &= \mathbf{g}_y^t + \nabla_y H(\mathbf{x}^t, \mathbf{y}^t) + \eta_2(\mathbf{y}^t - \mathbf{y}^{t-1}), \end{aligned} \quad (4)$$

where  $\mathbf{g}_x^t \in \partial f(\mathbf{x}^t)$  and  $\mathbf{g}_y^t \in \partial g(\mathbf{y}^t)$ , with  $\partial$  represents the Fréchet limiting-subdifferential (Attouch et al. 2010).

Apparently, imprecise task embedded calculations certainly slow down the convergence speed of the whole algorithm, while over-precise optimizations are time-consuming and unnecessary for practical use. Hence, we provide the following criterion to control the accuracies for optimizing univariate sub-problems.

**Criterion 1** *The errors  $\mathbf{e}_x^t$  and  $\mathbf{e}_y^t$  brought by task embedded strategies should be controlled by certain constants, i.e.,*

$$\|\mathbf{e}_x^t\| \leq C_x \epsilon_x^t, \quad \|\mathbf{e}_y^t\| \leq C_y \epsilon_y^t, \quad (5)$$

with  $\epsilon_x^t = \|\mathbf{x}^{t-1} - \mathbf{x}^{t-2}\|$  and  $\epsilon_y^t = \|\mathbf{y}^{t-1} - \mathbf{y}^{t-2}\|$ . Moreover,  $0 < 2C_x < \eta_1$  and  $0 < 2C_y < \eta_2$  should be satisfied.

Notice that the conditions in Eq. (5) are certainly attainable for converged algorithms since the inaccurate errors  $\mathbf{e}_x^t$  and  $\mathbf{e}_y^t$  are approaching to zero when the algorithms are identified as converged. However, since  $\mathbf{g}_x^t$  and  $\mathbf{g}_y^t$  should not be optionally selected from the sets of  $\partial f$  and  $\partial g$ , we in Prop. 1 provide an implementation for estimating the errors, to make our proposed TECU more practical<sup>1</sup>.

**Proposition 1** *Two intermediate variables  $\tilde{\mathbf{x}}^{t, K_x^t}$  and  $\tilde{\mathbf{y}}^{t, K_y^t}$  are calculated with respect to  $\mathbf{x}^{t, K_x^t}$  and  $\mathbf{y}^{t, K_y^t}$  as follows<sup>2</sup>:*

$$\begin{aligned} \tilde{\mathbf{x}}^{t, K_x^t} &= \text{prox}_f^1(\eta_1 \mathbf{x}^{t-1} + \mathcal{P}_x^{t-1}(\mathbf{x}^{t, K_x^t})), \\ \tilde{\mathbf{y}}^{t, K_y^t} &= \text{prox}_g^1(\eta_2 \mathbf{y}^{t-1} + \mathcal{P}_y^{t-1}(\mathbf{y}^{t, K_y^t})), \end{aligned} \quad (6)$$

with functions  $\mathcal{P}_x^{t-1}(\mathbf{u}) = (1 - \eta_1)\mathbf{u} - \nabla_x H(\mathbf{u}, \mathbf{y}^{t-1})$  and  $\mathcal{P}_y^{t-1}(\mathbf{u}) = (1 - \eta_2)\mathbf{u} - \nabla_y H(\mathbf{x}^t, \mathbf{u})$ . Then, the following

$$\begin{aligned} \mathbf{e}_x^{t, K_x^t} &= \mathcal{P}_x^{t-1}(\mathbf{x}^{t, K_x^t}) - \mathcal{P}_x^{t-1}(\tilde{\mathbf{x}}^{t, K_x^t}), \\ \mathbf{e}_y^{t, K_y^t} &= \mathcal{P}_y^{t-1}(\mathbf{y}^{t, K_y^t}) - \mathcal{P}_y^{t-1}(\tilde{\mathbf{y}}^{t, K_y^t}), \end{aligned} \quad (7)$$

are implementations of  $\mathbf{e}_x^t$  and  $\mathbf{e}_y^t$  in Eq. (4), by assigning  $\tilde{\mathbf{x}}^{t, K_x^t}$  to  $\mathbf{x}^t$  and  $\tilde{\mathbf{y}}^{t, K_y^t}$  to  $\mathbf{y}^t$ .

<sup>1</sup>All the proofs in this paper are presented in (Wang et al. 2018)

<sup>2</sup> $\text{prox}_\sigma^r(\mathbf{x}) \in \arg \min_{\mathbf{z}} \sigma(\mathbf{z}) + \frac{r}{2} \|\mathbf{z} - \mathbf{x}\|^2$  denotes proximal mapping to proper, l.s.c function  $\sigma$ .

So far, we have introduced the whole process of TECU and further provide its detailed procedures in Alg. 2. As follows, we will demonstrate that our error control conditions are more persuasive than previously used criteria (Li and Pong 2014; Yue et al. 2017), since it is well converged in theory.

## Theoretical Analyses

With properties of the objective function and the error control criterion, we in this section present some nice properties of TECU. Firstly, we demonstrate that along with the iteration progresses of TECU, there exists a bounded function that satisfies sufficient-descent property.

**Proposition 2** *Suppose that the sequence  $\{\mathbf{z}^t\}_{t \in \mathbb{N}}$  is generated by TECU, then with Criterion 1, there exist a bounded function  $\Phi(\mathbf{z}, \mathbf{w})$ , such that for  $\forall t \geq 0$ :*

$$\Phi(\mathbf{z}^t, \mathbf{z}^{t-1}) - \Phi(\mathbf{z}^{t+1}, \mathbf{z}^t) \geq a \|\mathbf{z}^{t+1} - \mathbf{z}^t\|^2, \quad (8a)$$

$$\text{dist}(0, \partial \Phi(\mathbf{z}^t, \mathbf{z}^{t-1})) \leq b(\|\mathbf{z}^t - \mathbf{z}^{t-1}\| + \|\mathbf{z}^{t-1} - \mathbf{z}^{t-2}\|), \quad (8b)$$

with definite positive constants  $a$  and  $b$ . In addition, the sequence  $\{\mathbf{z}^t\}_{t \in \mathbb{N}}$  generated by TECU is bounded.

Since our proposed TECU adopts a hybrid update scheme (presented in Alg. 1), thus the  $\Phi(\mathbf{z}, \mathbf{w})$ ,  $a$  and  $b$  have different concrete formations, corresponding to distinct combinations of updates (Wang et al. 2018). Then with this proposition, we are ready to illuminate the property of the final solution optimized by TECU.

**Theorem 3** *The  $\{\mathbf{z}^t\}_{t \in \mathbb{N}}$  generated by TECU is a Cauchy sequence, which converges to a critical point  $\mathbf{z}^* := (\mathbf{x}^*, \mathbf{y}^*)$  of the original objective function  $\Psi(\mathbf{z})$ .*

The Theorem 3 presents the property of the final optimized solution, meanwhile, demonstrates the robust performances of our proposed algorithm framework. Moreover, TECU has at least sub-linear convergence rate when the desingularising function  $\phi(s) = \frac{C}{\theta} s^\theta : [0, \mu) \rightarrow \mathbb{R}_+$  of the objective  $\Psi(\mathbf{z})$  is satisfied with positive constant  $C$  and  $\theta \in [0, 1)$ . Better yet, it will further have linear convergence rate if  $\theta \in [\frac{1}{2}, 1)$ . Though this convergence rate is accordance with previous CD schemes like (Attouch et al. 2010), our TECU is far more efficient and effective for realistic tasks.

## TECU for Realistic Tasks

As previously stated, TECU allows embedding both numerical algorithms and advance techniques in its algorithm framework. Hence, we consider two realistic tasks,  $\ell_0$ -regularized dictionary learning ( $\ell_0$ DL) and low-light image enhancement

Table 2: The comparison results on iteration number and whole computation time (s). The data dimensions  $(n, m, p)$  are set as  $(64, 600, 4000)$ ,  $(144, 900, 10000)$  and  $(256, 1600, 16000)$  respectively.

Algorithms	Iteration number / Total propagation steps					Computation time (s)				
	PALM	INV	BCU	iPALM	TECU	PALM	INV	BCU	iPALM	TECU
$n = 64$	21	109	19	21	<b>12 / 54</b>	4.45	16.20	3.87	4.68	<b>2.22</b>
$n = 144$	21	35	18	21	<b>14 / 152</b>	17.16	21.40	14.45	18.01	<b>11.96</b>
$n = 256$	22	39	18	22	<b>17 / 153</b>	121.47	137.27	100.19	124.67	<b>75.19</b>

Table 3: Computation time (s) of one-step iteration of CD algorithms, and one-step propagation of TECU on different data scales.

$n = 64$					$n = 144$					$n = 256$				
PALM	INV	BCU	iPALM	TECU	PALM	INV	BCU	iPALM	TECU	PALM	INV	BCU	iPALM	TECU
0.21	0.15	0.20	0.22	<b>0.04</b>	0.82	0.61	0.80	0.86	<b>0.08</b>	5.52	3.52	5.57	5.67	<b>0.49</b>

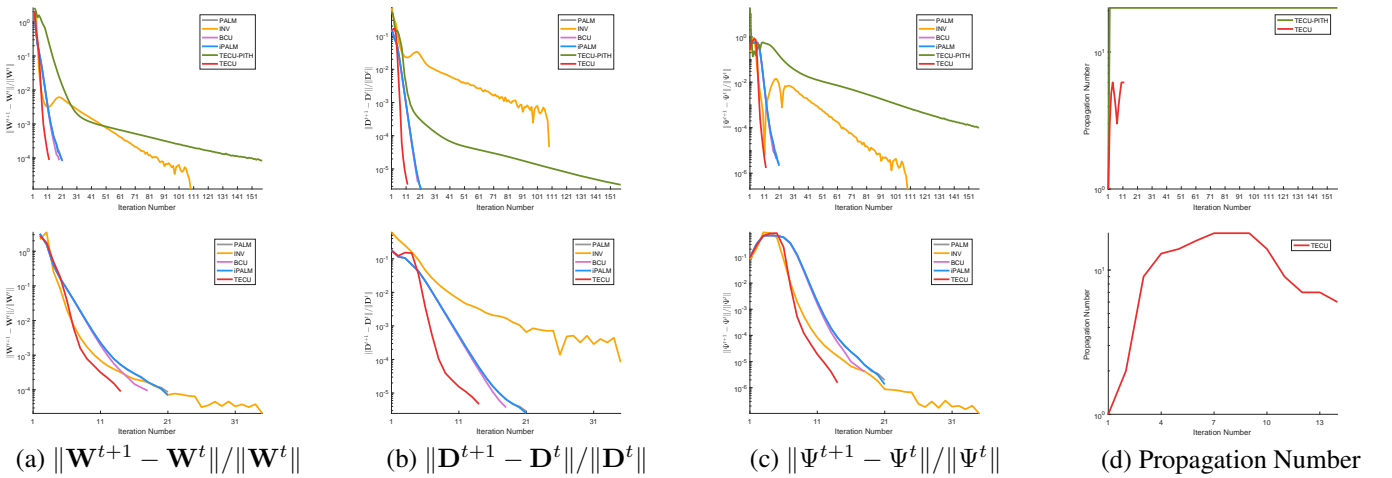


Figure 1: The compared convergence property of PALM, INV, BCU, iPALM and our TECU. The results in the first row belongs to the data with  $n = 64$ , while the comparison results in the second row belongs to  $n = 144$ .

(LIE), to verify both efficiency and effectiveness of embedding task-oriented strategies.

### $\ell_0$ -Regularized Dictionary Learning

Dictionary learning is a powerful tool for learning features from data. Its basic idea is to factorize the data  $\mathbf{Y}$  as  $\mathbf{D}\mathbf{W}^\top$ , where  $\mathbf{D}$  is the dictionary and  $\mathbf{W}$  is the corresponding coefficients. We consider the previously proposed  $\ell_0$ DL problem (Bao et al. 2016), which can be modeled as

$$\min_{\mathbf{W}, \mathbf{D}} \lambda \|\mathbf{W}\|_0 + \mathcal{X}_{\mathcal{D}}(\mathbf{D}) + \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{W}^\top\|^2, \quad (9)$$

where  $\|\cdot\|_0$  denotes the  $\ell_0$  penalty that counts the number of non-zero elements, and indicator function  $\mathcal{X}$  acts on the set  $\mathcal{D} = \{\mathbf{D} = \{\mathbf{d}_i\}_{i=1}^m : \|\mathbf{d}_i\| = 1, \forall i\}$  for normalized bases.

Notice that, the two sub-problems of  $\ell_0$ DL have entirely different characteristics. The sub-problem of  $\mathbf{W}$  is a sparse coding task with  $\ell_0$  penalty, which can be optimized by a proximal iterative hard-thresholding (PITH) method (Bach et al. 2011). While the sub-problem of  $\mathbf{D}$  minimizes a strongly convex quadratic function with unit ball constraint, thus it

can be efficiently solved. Considering this nice property, we embed ADMM for optimizing  $\mathbf{D}$  sub-problem, and the experimental results verify that embedding ADMM improve the efficiency of optimization. However, due to the non-convexity of the sub-problem of  $\mathbf{W}$ , additional experiments are conducted to show that embedding PITH is not a good choice for TECU, which also indicates the necessity of hybrid scheme.

Rather than TECU, the  $\ell_0$ DL problem can also be optimized by other numerical algorithms. However, as repeatedly stated, though the update of PALM can be simply computed, its request on estimating Lipschitz constants decelerate the overall convergence speeds, especially for large-scale data (see Table 3). Except for PALM, its two variants, i.e., BCU (Xu and Yin 2017) and iPALM (Pock and Sabach 2017) are also employed for solving the  $\ell_0$ DL problem. But these variants do not avoid estimating Lipschitz constants, even worse, their efficiencies are restricted by additional parameter conditions. To avoid these troubles, (Bao et al. 2016) utilize a strategy (named INV), which solves the sub-problem of  $\mathbf{D}$  by leaving out term  $\mathcal{X}_{\mathcal{D}}(\mathbf{D})$  first, and then directly projecting the results on  $\mathcal{X}_{\mathcal{D}}$ . However, we have noticed that the update

Table 4: Quantitative evaluations of NIQE (the lower, the better) on the NASA and Non-uniform dataset.

Dataset	HE	BPDHE	MSRCR	GOLW	NPEA
NASA	3.68	3.77	3.66	3.74	3.43
Non-uniform	4.28	3.09	3.04	2.99	2.99
Dataset	SRIE	WVM	JIEP	HDRNet	TECU
NASA	3.97	3.86	3.72	3.72	<b>3.41</b>
Non-uniform	3.02	2.94	2.97	3.11	<b>2.91</b>

of INV is inaccurate without theoretical supports, thus its performances are always lack of robustness (see Fig. 1).

### Low-light Image Enhancement

The purpose of LIE is to enhance the captured low-visibility images so that high-quality images can be obtained. For LIE, Retinex-based decomposition (Cai et al. 2017; Fu et al. 2016) is widely concerned:  $\mathbf{O} = \mathbf{I} \odot \mathbf{R}$ , with element-wise multiplication operator  $\odot$ . Thus LIE is to factorize the observed low-light image  $\mathbf{O}$  into an illumination layer  $\mathbf{I}$  that represents the light intensity, and a reflection layer  $\mathbf{R}$  which describes the physical characteristic of objects.

Considering the characteristics of illumination layer, many literatures (Fu et al. 2015; Cai et al. 2017) enforce a smooth constraint, i.e.,  $\|\nabla \mathbf{I}\|^2$  to represent the smooth changes of the illumination layer. Then together with the range constraints of both layers, we establish the following optimization model for LIE task:

$$\min_{\mathbf{I}, \mathbf{R}} \frac{\alpha}{2} \|\nabla \mathbf{I}\|^2 + \mathcal{X}_{\mathcal{I}}(\mathbf{I}) + \mathcal{X}_{\mathcal{R}}(\mathbf{R}) + \frac{1}{2} \|\mathbf{O} - \mathbf{I} \odot \mathbf{R}\|^2, \quad (10)$$

with  $\mathcal{I} = \{\mathbf{I} : \mathbf{I}_i \in [0, \mathbf{O}_i], \forall i\}$ ,  $\mathcal{R} = \{\mathbf{R} : \mathbf{R}_i \in [0, 1], \forall i\}$ .

Employing TECU for optimizing it, we embed a residual-type CNN to propagate the illumination layer very close to the desired solution. Specifically, we first randomly choose 800 image-pairs from ImageNet database (Krizhevsky, Sutskever, and Hinton 2012) and crop them into  $35 \times 35$  small patches, to train a neural network with only one residual block including 7 convolutional layers and ReLU activations. Then at each iteration, we first use the pre-trained network to propagate the latest value of  $\mathbf{I}$ . Then by considering that the pre-trained network may not always satisfy the error control conditions, thus we further employ prox-linear updates as the remaining propagations, until the Criterion 1 is satisfied. For the other sub-problem of reflection layer, we adopt proximal update to get the closed-form solution.

## Experimental Results

With the task embedded strategies respectively designed for  $\ell_0$ DL and LIE problems, we apply TECU to  $\ell_0$ DL with synthetic data to verify its nice convergence properties. While for LIE task, TECU is applied to real-world images. The experimental results of both two realistic tasks demonstrate the efficiency and effectiveness of embedding strategies into CD algorithm, by comparing with other state-of-the-art methods.

## $\ell_0$ -Regularized Dictionary Learning

We generate synthetic data with different sizes (see Table 2) to help analyze the convergence properties of TECU. Specifically, all the algorithms are terminated when satisfying the following condition:

$$\max \left\{ \frac{\|\mathbf{D}^{t+1} - \mathbf{D}^t\|}{\|\mathbf{D}^t\|}, \frac{\|\mathbf{W}^{t+1} - \mathbf{W}^t\|}{\|\mathbf{W}^t\|}, \frac{\|\Psi^{t+1} - \Psi^t\|}{\|\Psi^t\|} \right\} < 1e^{-4}. \quad (11)$$

**Comparisons of Different Embedded Strategies** Firstly, we conduct experiments to compare two particular cases of our proposed framework. Precisely, the sign ‘‘TECU’’ in the Fig. 1, Table 2 and Table 3 represents the case of embedding ADMM for updating  $\mathbf{D}$ , while using prox-linear update for  $\mathbf{W}$  sub-problem; the ‘‘TECU-PITH’’ in Fig. 1 refers to embedding PITH for updating  $\mathbf{W}$ , while applying prox-linear update for  $\mathbf{D}$  sub-problem.

From the first row of Fig. 1, it is distinct that these two cases of TECU have quite different convergence performances. From the top row of Fig. 1(d), we can see that TECU requires a few propagation steps, however, TECU-PITH reaches the maximum inner steps (set as 20) at almost every iteration. The excessive inner propagations definitely decelerate the overall convergence speed, i.e., TECU uses 2.22s but TECU-PITH takes nearly 600s to converge.

This comparison result from one side shows that ADMM is more productive for optimizing  $\mathbf{D}$  sub-problem, but PITH is less effective for solving the sub-problem of  $\mathbf{W}$ . On the other side, the different performances are also influenced by the characteristics of sub-problems. The sub-problem of  $\mathbf{D}$  minimizes strongly convex quadratic function with unit ball constraint, thus it can be efficiently solved. However, due to the  $\ell_0$  penalty, the sub-problem of  $\mathbf{W}$  is NP hard, which is more difficult to optimize. Therefore, this comparisons of different embedded strategies indicate the necessity of employing hybrid scheme in the framework of TECU, meanwhile, suggest embedding high-efficient numerical algorithms for sub-problem optimization.

**Comparisons with Other Algorithms** Comparing with other existing algorithms, we can see from Table 2 and Fig. 1 that TECU converges with less iteration steps, especially has better convergence performances on the optimization of  $\mathbf{D}$  sub-problem. Moreover, we give further comparisons in Table 3 to show that the computation time of one propagation in TECU is much less than the ones in other algorithms. Thus it is the reason why TECU totally adopts more propagations but has less computation time. Moreover, since all the PALM, BCU and iPALM require estimating Lipschitz constants at every iteration, thus they have similar one-step computation time on different data scales. It is obvious that estimating Lipschitz constants during iterations is extremely time-consuming especially for large scale data. Thus though embedding numerical algorithms brings more propagations, it avoids estimating Lipschitz constants thus is far more efficient than existing algorithms.





Figure 2: Comparisons between the classical CD algorithm, i.e., PAM (Attouch et al. 2010) and TECU. The PSNR scores are reported below (c) and (d) sub-figures. The PSNR curves with respect to iteration number is plotted in sub-figure (e).



Figure 3: Visual comparisons selected from the NASA dataset (top row) and the Non-uniform dataset (bottom row). Moreover, we post the NIQE / computation time (s) below each image.

### Low-light Image Enhancement

Firstly, we conduct an experiment in Fig. 2 to compare TECU with the classical CD algorithm, i.e., PAM (Attouch et al. 2010) on an example image from (Cai, Gu, and Zhang 2018). Since (Cai, Gu, and Zhang 2018) provides image pairs of low-light images and the references obtained by other techniques, we provide PSNR values with respect to the given reference to give quantitative evaluations. As shown in Fig. 2, TECU achieves superior performances in terms of visual effect and PSNR score. Moreover, we further plot the PSNR curves in Fig. 2(e). From which we can tell that embedding network into the classical CD scheme certainly produces an excellent growth trend than employing itself.

We compare TECU with state-of-the-art approaches including: HE (Cheng and Shi 2004), BPDHE (Sheet et al. 2010), MSRRCR (Rahman, Jobson, and Woodell 2004), GOLW (Shan, Jia, and Brown 2010), NPEA (Wang et al. 2013), SRIE (Fu et al. 2015), WVM (Fu et al. 2016), JIEP (Cai et al. 2017) and HDRNet (Gharbi et al. 2017), on the NASA dataset (NASA 2001) and the Non-uniform dataset (Wang et al. 2013). There are 23 images of different indoor and outdoor scenes in NASA dataset, while the Non-uniform dataset consists 130 low-quality images in different natural scenes including sunshine, overcast sky and nightfall scenarios.

For the lack of ground truth, it is impossible to give standard metrics (i.e., PSNR) to evaluate the quantitative performances for LIE task. In previous literatures (Wang et al. 2013; Fu et al. 2016; Cai et al. 2017), a blind image quality assessment called Natural Image Quality Evaluator (NIQE) is widely used to give quantitative evaluation for LIE. Following this, we also present the NIQE scores in Table 4, comparing with all these state-of-the-art methods on the two different benchmarks. The comparison results in Table 4 indicate that TECU with embedded network has the lowest NIQE score and thus achieves the highest image quality. We also provide a visual comparison on examples selected from both two datasets. It is obvious that TECU is able to enhance image quality with high contrast, but other results are still contain details in dark, which are hard to recognize. Thus from both quantitative and quality analyses, we can conclude that embedding networks in the framework of TECU is effective and competitive for the challenging LIE task.

### Conclusion

We propose a realizable algorithm framework TECU, which embeds both numerical algorithms and advance techniques for optimizing a generic multivariate non-convex problem. Through embedding task-oriented strategies, TECU is able to improve the convergence speed of the whole algorithm

and obtain desired solutions with high probability. Moreover, we further provide a realizable error control condition, to ensure robust performances with rigid theoretical supports. The experimental results on two practical problems verify the superiorities of our proposed algorithm.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant Nos. 61672125, 61733002, 61572096, 61632019 and 61806057), China Postdoctoral Science Foundation (Grant No. 2018M632018) and the Fundamental Research Funds for the Central Universities.

## References

Attouch, H.; Bolte, J.; Redont, P.; and Soubeyran, A. 2010. Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the kurdyka-lojasiewicz inequality. *Mathematics of Operations Research* 35:438–457.

Bach, F.; Jenatton, R.; Mairal, J.; and Obozinski, G. 2011. Optimization with sparsity-inducing penalties. *Foundations & Trends<sup>®</sup> in Machine Learning* 4(1):1–106.

Bao, C.; Ji, H.; Quan, Y.; and Shen, Z. 2016. Dictionary learning for sparse coding: Algorithms and convergence analysis. *IEEE TPAMI* 38(7):1356–1369.

Bolte, J.; Sabach, S.; and Teboulle, M. 2014. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming* 146:459–494.

Cai, B.; Xu, X.; Guo, K.; Jia, K.; Hu, B.; and Tao, D. 2017. A joint intrinsic-extrinsic prior model for retinex. In *CVPR*.

Cai, J.; Gu, S.; and Zhang, L. 2018. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE TIP* 27(4):2049–2062.

Chan, S. H.; Wang, X.; and Elgandy, O. A. 2017. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE TCI* 3(1):84–98.

Cheng, H., and Shi, X. 2004. A simple and effective histogram equalization approach to image enhancement. *Digital Signal Processing* 14(2):158–170.

Elad, M. 2010. *Sparse and Redundant Representations*. Springer New York.

Fu, X.; Liao, Y.; Zeng, D.; Huang, Y.; Zhang, X.-P.; and Ding, X. 2015. A probabilistic method for image enhancement with simultaneous illumination and reflectance estimation. *IEEE TIP* 24(12):4965–4977.

Fu, X.; Zeng, D.; Huang, Y.; Zhang, X.-P.; and Ding, X. 2016. A weighted variational model for simultaneous reflectance and illumination estimation. In *CVPR*.

Gharbi, M.; Chen, J.; T., B. J.; Hasinoff, S. W.; and Fredo, D. 2017. Deep bilateral learning for real-time image enhancement. *ACM ToG* 36(4):118.

Gregor, K., and Lecun, Y. 2010. Learning fast approximations of sparse coding. In *ICML*.

Guo, X. and Cao, X., and Ma, Y. 2014. Robust separation of reflection from multiple images. In *CVPR*.

Kim, D., and Fessler, J. A. 2018. Another look at the fast iterative shrinkage/thresholding algorithm (fista). *Siam J. on Optimization* 28(1).

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Li, Y., and Brown, M. S. 2014. Single image layer separation using relative smoothness. In *CVPR*.

Li, G., and Pong, T. K. 2014. Global convergence of splitting methods for nonconvex composite optimization. *arXiv:1407.0753*.

Li, C.; Guo, J.; Cong, R.; Pang, Y.; and Wang, B. 2016. Underwater image enhancement by dehazing with minimum information loss and histogram distribution prior. *IEEE TIP* 25(12):5664–5677.

Liu, R.; Ma, L.; Wang, Y.; and Zhang, L. 2018. Learning converged propagations with deep prior ensemble for image enhancement. *IEEE TIP*.

Luo, Z. Q., and Tseng, P. 1989. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications* 72:7–35.

NASA. 2001. Retinex image processing. <https://dragon.larc.nasa.gov/retinex/pao/news/>.

Pock, T., and Sabach, S. 2017. Inertial proximal alternating linearized minimization (ipalm) for nonconvex and nonsmooth problems. *Siam J. Imaging Sciences* 9(4):1756–1787.

Rahman, Z.-u.; Jobson, D. J.; and Woodell, G. A. 2004. Retinex processing for automatic image enhancement. *Journal of Electronic Imaging* 13(1):100–111.

Schmidt, U., and Roth, S. 2014. Shrinkage fields for effective image restoration. In *CVPR*.

Shan, Q.; Jia, J.; and Brown, M. S. 2010. Globally optimized linear windowed tone mapping. *IEEE TVCG* 16(4):663–675.

Sheet, D.; Garud, H.; Suveer, A.; Mahadevappa, M.; and Chatterjee, J. 2010. Brightness preserving dynamic fuzzy histogram equalization. *IEEE TCE* 56(4).

Spillane, N. 2016. An adaptive multipreconditioned conjugate gradient algorithm. *Siam J. on Scientific Computing* 38(3):A1896–A1918.

Vu, T. H., and Monga, V. 2016. Fast low-rank shared dictionary learning for image classification. *IEEE TIP* PP(99):1–1.

Wang, S.; Zheng, J.; Hu, H.-M.; and Li, B. 2013. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP* 22(9):3538–3548.

Wang, Y.; Liu, R.; Song, X.; and Su, Z. 2016. Linearized alternating direction method with penalization for nonconvex and nonsmooth optimization. In *AAAI*.

Wang, Y.; Liu, R.; Ma, L.; and Song, X. 2018. Task embedded coordinate update: A realizable framework for multivariate nonconvex optimization. *arXiv:1811.01587*.

Wang, Y.; Yin, W.; and Zeng, J. 2018. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing* 1–35.

Xu, Y., and Yin, W. 2015. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM J. Imaging Sciences* 6(3):1758–1789.

Xu, Y., and Yin, W. 2017. A globally convergent algorithm for nonconvex optimization based on block coordinate update. *Journal of Scientific Computing* 72(2):700–734.

Yue, H.; Yang, J.; Sun, X.; Wu, F.; and Hou, C. 2017. Contrast enhancement based on intrinsic image decomposition. *IEEE TIP* 26(8):3981–3994.

Zhang, K.; Zuo, W.; Gu, S.; and Zhang, L. 2017. Learning deep cnn denoiser prior for image restoration. In *CVPR*.