British Journal of Mathematics & Computer Science
4(20): 2917-2928, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

SCIENCEDOMAIN

An Approximate Sequential Bundle Method for Solving a Convex Nondifferentiable Bilevel Programming Problem

Jie Shen[1*], Li-Ping Pang[2], Xi-Jun Liang[3] and Zun-Quan Xia[2]

[1] *School of Mathematics, Liaoning Normal University, Dalian 116029, 0086-0411-84258356, China.*
[2] *School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China.*
[3] *College of Science, China University of Petroleum, Qingdao 266555, China.*

Original Research Article

Received: 13 June 2014
Accepted: 09 July 2014
Published: 01 August 2014

Abstract

By combining two bundle methods, PBMASL (proximal bundle method with approximate subgradient linearizations) and DPLBM (descent proximal level bundle method), we present an approximate sequential bundle algorithm for solving a bilevel programming problem with a nondifferentiable convex objective function and two separable constraints. In the proposed algorithm, the values of the objective function in the constraints and its subgradients are computed approximately, the estimates of the tolerances are not required for convergence proof. The presented results improve and extend the earlier work.

Keywords: Nonsmooth optimization; Bilevel programming problem; Bundle method; Subgradient; Proximal bundle method

2010 Mathematics Subject Classification: 90C30; 90C25; 49M37; 90C59

1 Introduction

We consider a bilevel programming problem of the form

$$\text{(P)} \qquad \begin{cases} \min & f(x,y) \\ \text{s. t.} & (x,y) \in \Omega_1 \times \Omega_2 \subset R^m \times R^n, \end{cases} \tag{1.1}$$

where $f : R^{m+n} \to R^1$ is convex and nondifferentiable, Ω_1 is compact convex and Ω_2 is defined by $\Omega_2 = \text{Arg}\inf_{y\in S}\varphi(y) = \{y \,|\, \varphi(y) = \inf_{y\in S}\varphi(y)\}$, $\varphi : R^n \to R^1$ is convex and level bounded, S is a nonempty closed convex set in the Euclidean space R^n.

*Corresponding author: E-mail: tt010725@163.com. This research was partially supported by National Natural Science Foundation of China (Grants 11301246, 11171049 and 11171138).

For the case $S = R^n$, a bundle method for solving problem (P) is presented by Zun-Quan Xia, Jie Shen and Li-Ping Pang [1] , in which the function values and subgradients of φ are assumed to be computed exactly. The following algorithm framework is presented for solving (P) [1]:

Algorithm 1.1 :

Step 1 Take $\bar{x} \in \Omega_1$ and find $\bar{y} \in \Omega_2$ by solving

$$(\mathrm{P}_a) \qquad \min_{y \in S \subset R^n} \varphi(y) \qquad \text{via a bundle method.} \tag{1.2}$$

Step 2 Minimize f on $\Omega_1 \times \Omega_2$,

$$(\mathrm{P}_b) \qquad \min_{(x,y) \in \Omega_1 \times \Omega_2} f(x, y) \qquad \text{with } (\bar{x}, \bar{y}) \in \Omega_1 \times \Omega_2 \text{ as a starting point.} \tag{1.3}$$

End of Algorithm 1.1

In this article, we still use Alg.1.1 to solve problem (P), but this time we will focus on solving problem (P_a). There are many algorithms designed for solving problem (P_a) [2-8], but they all need to compute the exact function values, this necessity may bring much difficulty when it comes to constructing an implementable algorithm [9-11]. It was established that the proximal bundle algorithm based on the inexact linearizations of the objective function converges to an exact optimal solution, if ε, the approximation error of objective function values and its subgradients, satisfies $\varepsilon \to 0$ in the course of the iterations [9]. Solodov considered the proximal form of a bundle algorithm for minimizing a nonsmooth convex function, the algorithm assumed that the function values and its subgradients are evaluated approximately, and it answered the question that how the approximation error ε should be controlled in order to satisfy the desired optimality tolerance, that is, given some nonzero (and not tending to zero) approximation error ε, some kind approximate optimal solution can be obtained which depends on the given approximation error ε [11]. Kiwiel proposed a proximal bundle method which only requires evaluating the objective function values and its subgradients with an accuracy $\varepsilon > 0$, it asymptotically finds points that are $\varepsilon-$ optimal [10], this algorithm is denoted by PBMASL in our paper and will be used to solve problem (P_a). According to [10], for given $\varepsilon_f > 0, \varepsilon_g > 0$, one could obtain an ε-optimal solution y of problem (P_a), where $\varepsilon = \varepsilon_f + \varepsilon_g$, i.e., find a y satisfying

$$y \in \Omega_{2,\varepsilon} = \{y \mid \varphi(y) \leq \inf_{y \in S} \varphi(y) + \varepsilon\}, \tag{1.4}$$

and this algorithm can be slightly revised by modifying the initial parameter such that it terminates in finite steps to obtain an approximate solution to problem (P_a). The assumptions for using approximate subgradients and approximate values of the objective function are realistic in many applications, for instance, the Lagrangian relaxation problem: if f is a max-type function of the form

$$f(y) = \sup\{F_z(y) \mid z \in Z\}, \tag{1.5}$$

where each $F_z(y)$ is convex and Z is an infinite set, then it may be impossible to calculate exactly $f(y)$. However, we may still consider two cases. In the first case, for each positive $\varepsilon > 0$ one can find an element $z_y \in Z$ satisfying $F_{z_y}(y) \geq f(y) - \varepsilon$; in the second case, this may be possible only for some fixed (any possibly unknown) $\varepsilon < \infty$. In both cases we may set $\bar{f}_y = F_{z_y}(y) \geq f(y) - \varepsilon$. A special case of (1.5) arises from Lagrangian relaxation [12], where the problem $\min\{f(y) \mid y \in S\}$ with $S = R^n_+$ is the Lagrangian duality problem of the primal problem

$$\sup \ \psi_0(z) \quad \text{s.t. } \psi_j(z) \geq 0, \ \ j = 1, 2, \cdots, n, \ \ z \in Z$$

with $F_z(y) = \psi_0(z) + \langle y, \psi(z) \rangle$ for $\psi = (\psi_1, \psi_2, \cdots, \psi_n)$. Then, for each multiplier $y \geq 0$, we need only find $z_y \in Z$ such that $\bar{f}_y = F_{z_y}(y) \geq f(y) - \varepsilon$.

Level bundle methods is a class of bundle method variants based on minimizing a quadratic function subjected to some level set [13-16]. Among them a *descent proximal level bundle method* (DPLBM) [14] will be used in our paper to solve problem (P_b).

The problem

$$(\mathrm{P}_\varepsilon) \qquad \min_{(x,y)\in\Omega_1\times\Omega_{2,\varepsilon}} \quad f(x,y) \tag{1.6}$$

is an approximation to problem (P), where ε is a given nonnegative number. We firstly focus our attention on solving problem (P$_\varepsilon$) and then construct an approximate sequential bundle method for solving problem (P) by combing two bundle methods, PBMASL and DPLBM. The proposed approximate algorithm is easier to implement than [1] since it only requires the inexact information of the objective function in the constraints.

This paper is organized as follows. In Sections 2 and 3 two bundle methods, PBMASL and DPLBM, are used to solve problem (P$_a$) and problem (P$_b$), respectively. An approximate sequential bundle method for solving problem (P) is presented in Section 4, and its convergence analysis is given in Section 5. In Section 6, we report on numerical testing of the proposed algorithm. Finally, some conclusions are given in Section 7.

2 Solving Problem (P$_a$)

In this section we use PBMASL to get an ε-optimal solution to problem (P$_a$). The method PBMASL generates a sequence of trial points $\{y^{k_b}\}_{k_b=1}^{\infty} \subset S$, and at these trial points the approximate function values $\varphi_y^{k_b} = \varphi_{y^{k_b}}$, the approximate subgradients $g^{k_b} = g_{y^{k_b}}$ are computed and the linearization $\varphi_{k_b}(\cdot)$ is given such that

$$\begin{aligned} \varphi_{k_b}(\cdot) &= \varphi_y^{k_b} + \langle g^{k_b}, \cdot - y^{k_b}\rangle \le \varphi(\cdot) + \varepsilon_g, \\ \varphi_{k_b}(y^{k_b}) &= \varphi_y^{k_b} \ge \varphi(y^{k_b}) - \varepsilon_f \end{aligned} \tag{2.1}$$

hold for fixed $\varepsilon_f \ge 0$ and $\varepsilon_g \ge 0$. Thus, we have $\varphi_y^{k_b} \in [\varphi(y^{k_b}) - \varepsilon_f,\, \varphi(y^{k_b}) + \varepsilon_g]$ is an approximation to $\varphi(y^{k_b})$, and $g^{k_b} \in \partial_\varepsilon \varphi(y^{k_b})$ for $\varepsilon = \varepsilon_f + \varepsilon_g$. At the k_bth iteration, the cutting-plane model of φ

$$\check{\varphi}_{k_b}(\cdot) = \max_{j\in J^{k_b}} \varphi_j(\cdot), \quad J^{k_b} \subset \{1,2,\cdots,k_b\} \tag{2.2}$$

is used for finding

$$y^{k_b+1} = \text{argmin}\{h_{k_b}(\cdot) = \check{\varphi}_{k_b}(\cdot) + i_s(\cdot) + \frac{1}{2t_{k_b}}||\cdot - x^{k_b}||^2\}, \tag{2.3}$$

where $t^{k_b} > 0$ is a stepsize, at $x^{k_b} = y^{k_b(l)}$ one has $\varphi_x^{k_b} = \varphi_y^{k_b(l)}$ for some $k_b(l) \le k_b$ and $i_s(\cdot)$ denotes the indicator function associated with S (i.e., $i_s(x) = 0$, if $x \in S$ and $+\infty$ otherwise). The predicted descent is defined by

$$v_{k_b} = \varphi_x^{k_b} - \check{\varphi}_{k_b}(y^{k_b+1}). \tag{2.4}$$

Note that $0 \in \partial h_{k_b}(y^{k_b+1})$, there exist $p_\varphi^{k_b} \in \partial\check{\varphi}_{k_b}(y^{k_b+1})$ and $p_s^{k_b} \in \partial i_s(y^{k_b+1})$ such that

$$p_s^{k_b} = -(y^{k_b+1} - x^{k_b})/t_{k_b} - p_\varphi^{k_b} \tag{2.5}$$

and there are multipliers $v_j^{k_b},\ j \in J^{k_b}$ such that

$$\begin{aligned} &p_\varphi^{k_b} = \textstyle\sum_{j\in J^{k_b}} v_j^{k_b} g^j, \quad \sum_{j\in J^{k_b}} v_j^{k_b} = 1, \\ &v_j^{k_b}[\check{\varphi}_{k_b}(y^{k_b+1}) - \varphi_j(y^{k_b+1})] = 0, \quad j \in J^{k_b}. \end{aligned} \tag{2.6}$$

We define the following aggregate linearizations of $\check{\varphi}_{k_b}$, i_s, and $\varphi_s(\cdot) = \varphi(\cdot) + i_s(\cdot)$, respectively:

$$\begin{aligned} \bar{\varphi}_{k_b}(\cdot) &= \check{\varphi}_{k_b}(y^{k_b+1}) + \langle p_\varphi^{k_b}, \cdot - y^{k_b+1}\rangle \\ &\le \check{\varphi}_{k_b}(\cdot) \le \varphi(\cdot) + \varepsilon_g, \end{aligned} \tag{2.7}$$

$$\bar{i}_s^{k_b}(\cdot) = \langle p_s^{k_b}, \cdot - y^{k_b+1}\rangle \le i_s(\cdot), \tag{2.8}$$

$$\bar{\varphi}_s^{k_b}(\cdot) = \bar{\varphi}_{k_b}(\cdot) + \bar{i}_s^{k_b}(\cdot) \le \check{\varphi}_s^{k_b}(\cdot) = \check{\varphi}_{k_b}(\cdot) + i_s(\cdot) \le \varphi_s(\cdot) + \varepsilon_g. \tag{2.9}$$

Furthermore, we have

$$\varphi_x^{k_b} + \langle p^{k_b}, \cdot - x^{k_b}\rangle - \alpha_{k_b} = \bar{\varphi}_s^{k_b}(\cdot) \le \varphi_s(\cdot) + \varepsilon_g, \tag{2.10}$$

where $p^{k_b} = p_\varphi^{k_b} + p_s^{k_b} = (x^{k_b} - y^{k_b+1})/t_{k_b}$, $\alpha_{k_b} = \varphi_x^{k_b} - \bar{\varphi}_s^{k_b}(x^{k_b})$. Hence, it is not difficult to obtain that

$$\varphi_x^{k_b} \le \varphi(x) + \varepsilon_g + ||p^{k_b}||\,||x - x^{k_b}|| + \alpha_{k_b}, \quad \text{for all } x \in S. \tag{2.11}$$

Inequality (2.11) shows that x^{k_b} is ε-optimal (i. e., $\varphi(x^{k_b}) \le \varphi_* + \varepsilon$, $\varepsilon = \varepsilon_f + \varepsilon_g$) if the optimality measure

$$V_{k_b} = \max\{||p^{k_b}||, \alpha_{k_b}\} \tag{2.12}$$

is zero; x^{k_b} is approximately ε-optimal if V_{k_b} is small.

Algorithm 2.1 (PBMASL):

Step 0 Select $x^1 \in S$, $\kappa \in (0,1)$, a stepsize bound $T_1 > 0, t_1 \in (0, T_1]$. Set $k_b = k_b(0) = 1$, $y^1 = x^1$, $J^1 = \{1\}$, $\varphi_x^1 = \varphi_y^1, g^1 = g_{y^1}$, $i_t^1 = 0$, $l = 0$ ($k_b(l) - 1$ denotes the iteration of the lth descent step). Take $\varepsilon' > 0$ such that $0 < \varepsilon' < \varepsilon = \varepsilon_f + \varepsilon_g$.

Step 1 Compute y^{k_b+1} and $v_j^{k_b}$ such that (2.5) and (2.6) hold.

Step 2 If $V_{k_b} = 0$, stop ($\varphi_x^{k_b} \le \varphi_* + \varepsilon_g$).

Step 3 If $v_{k_b} < -\alpha_{k_b}$, then set $t_{k_b} = 10t_{k_b}, T_{k_b} = \max\{T_{k_b}, t_{k_b}\}$, $i_t^{k_b} = k_b$ and loop to Step 1, else set $T_{k_b+1} = T_{k_b}$.

Step 4 Evaluate $\varphi_y^{k_b+1}$ and g^{k_b+1}.
If the descent test holds:

$$\varphi_y^{k_b+1} \le \varphi_x^{k_b} - \kappa v_{k_b}, \tag{2.13}$$

then set $x^{k_b+1} = y^{k_b+1}, \varphi_x^{k_b+1} = \varphi_y^{k_b+1}, i_t^{k_b+1} = 0$, $k_b(l+1) = k_b + 1$ and increase l by one; else set $x^{k_b+1} = x^{k_b}, \varphi_x^{k_b+1} = \varphi_x^{k_b}$ and $i_t^{k_b+1} = i_t^{k_b}$.

Step 5 Choose $J^{k_b+1} \supset \{\hat{J}^{k_b} \cup \{k_b+1\}\}$, where $\hat{J}^{k_b} = \{j \in J^{k_b} : v_j^{k_b} \ne 0\}$.

Step 6 If $k_b(l) = k_b + 1$, select $t_{k_b+1} \in [t_{k_b}, T_{k_b+1}]$,
otherwise, either set $t_{k_b+1} = t_{k_b}$, or choose $t_{k_b+1} \in [0.1t_{k_b}, t_{k_b}]$ if $i_t^{k_b+1} = 0$ and $\varphi_x^{k_b} - \varphi_{k_b+1}(x^{k_b}) \ge V_{k_b}$.

Step 7 Increase k_b by one and go to Step 1.

End of Algorithm 2.1

Note that the parameter ε' will be replaced by ε^k in Alg. 4.1 and Alg. 4.2.

The loop between Step 1 and Step 3 is infinite iff $\varphi_x^{k_b} \le \inf \check{\varphi}_s^{k_b} < \check{\varphi}_{k_b}(x^{k_b})$, in which case the current iterate x^{k_b} is already ε'-optimal [10].

Lemma 2.1. *[10]If* $\liminf_{k_b} V'_{k_b} = 0$ (e. g., $\lim_{k_b} V_{k_b} = 0$)*, where* V'_{k_b} *denotes the minimum value of* V_{k_b} *at each iteration* k_b*, and* x^{k_b} *is bounded, then* $\varphi_x^\infty \le \varphi_* + \varepsilon_g$*, where* $\varphi_x^\infty = lim_k \varphi_x^k$*,* $\varphi_* = \inf\{\varphi(x) | x \in S\}$*.* □

Lemma 2.2. *[10] If infinitely many descent steps occur, then* $\varphi_x^\infty \le \varphi_* + \varepsilon_g$*.* □

Theorem 2.3. *[10] The following two assertions are true:*

1. $\varphi_x^{k_b} \downarrow \varphi_x^\infty \le \varphi_* + \varepsilon_g$;

2. $\limsup_{k_b} \varphi(x^{k_b}) \leq \varphi_* + \varepsilon'$ *for* $\varepsilon' = \varepsilon_f + \varepsilon_g$ *so that each cluster* x^* *of* $\{x^{k_b}\}$ *(if any) satisfies* $x^* \in S$ *and* $\varphi(x^*) \leq \varphi_* + \varepsilon'$. □

According to Theorem 2.3 above, we obtain the following results.

Theorem 2.4. *For given* $\varepsilon' > 0$*, if* $\varepsilon > \varepsilon'$*, then* $\{\varphi(y^{k_b})\}_{k_b \in N}$ *generated by Alg. 2.1 satisfies*

$$\limsup_{k_b \to \infty} \varphi(y^{k_b}) < \inf_{y \in S} \varphi(y) + \varepsilon,$$

which implies that there exists a $\bar{N} \in \aleph_\infty^\sharp$ *and a* $K \in \bar{N}$ *such that*

$$\varphi(y^{k_b}) \leq \inf_{y \in S} \varphi(y) + \varepsilon, \quad \forall\, k_b > K, \quad k_b \in \bar{N}, \tag{2.14}$$

where $\aleph_\infty^\sharp$ *denotes the collection of all infinite subsequences of nature number set* N*, i. e.,*

$$y^{k_b} \in \{y \,|\, \varphi(y) \leq \inf_{y \in S} \varphi(y) + \varepsilon\}, \quad \forall\; k_b > K, \quad k_b \in \bar{N}. \tag{2.15}$$

□

Corollary 2.5. *Alg. 2.1 terminates finitely at some approximate optimal solution to problem* (P$_a$).

□

3 Solving Problem (P$_b$)

In this section DPLBM is used to solve problem (P$_b$). Suppose that at the k_cth iteration one has generated linearizations

$$f^j(x, y) = f(\hat{x}^j, \hat{y}^j) + \langle g(\hat{x}^j, \hat{y}^j), (x, y) - (\hat{x}^j, \hat{y}^j)\rangle$$

of f at trial points $(\hat{x}^j, \hat{y}^j) \in \Omega_1 \times \Omega_{2,\varepsilon}$, where $g(\hat{x}^j, \hat{y}^j) \in \partial f(\hat{x}^j, \hat{y}^j)$, $j \in J^{k_c} \subset \{1, 2, ..., k_c\}$. We define $\check{f}^{k_c}(x, y) = \max_{j \in J^{k_c}} f^j(x, y)$ and let

$$(u^{k_c+1}, v^{k_c+1}) = \text{argmin}\{\frac{1}{2}\|(x, y) - (x^{k_c}, y^{k_c})\|^2 | (x, y) \in \Omega_1 \times \Omega_{2,\varepsilon}, \check{f}^{k_c}(x, y) \leq f_{\text{lev}}^{k_c}\},$$

where $f_{\text{lev}}^{k_c} < f(x^{k_c}, y^{k_c})$ is chosen to ensure that $f_{\text{lev}}^{k_c} \to f^* = \inf_{\Omega_1 \times \Omega_{2,\varepsilon}} f$ as $k_c \to \infty$. If a finite lower bound $f_{\text{low}}^{k_c} \leq f^*$ is already known, then we usually take

$$f_{\text{lev}}^{k_c} = k_l f_{\text{low}}^{k_c} + (1 - k_l) f(x^{k_c}, y^{k_c}) = f(x^{k_c}, y^{k_c}) - k_l \Delta^{k_c},$$

where $0 < k_l < 1$, $\Delta^{k_c} = f(x^{k_c}, y^{k_c}) - f_{\text{low}}^{k_c}$. The desired descent δ^{k_c} is defined by $\delta^{k_c} = f(x^{k_c}, y^{k_c}) - f_{\text{lev}}^{k_c}$.

Solving the problem

$$\begin{array}{ll} \text{min} & \frac{1}{2}\|(x, y) - (x^{k_c}, y^{k_c})\|^2 \\ \text{s.t.} & f^j(x, y) \leq f_{\text{lev}}^{k_c}, \quad \forall j \in J^{k_c}, \\ & (x, y) \in \Omega_1 \times \Omega_{2,\varepsilon} \end{array} \tag{3.1}$$

is required in the following algorithm.

We give two stopping criteria which will be used in Alg. 4.2.

TWO STOPPING CRITERIA

Stopping Criterion 1(SC1) : If $\Delta^{k_c} \leq \varepsilon_{\text{opt}}$, then stop;
Stopping Criterion 2(SC2) : If $\max\{\|p^{k_c}\|, \tilde{\alpha}_p^{k_c}\} \leq \varepsilon_{\text{opt}}$, then stop,
where $p^{k_c} \in \partial \check{f}^{k_c}_{\Omega_1 \times \Omega_{2,\varepsilon}}(u^{k_c+1}, v^{k_c+1})$,
$\check{f}^{k_c}_{\Omega_1 \times \Omega_{2,\varepsilon}}(\cdot,\cdot) = \check{f}(\cdot,\cdot) + \delta_{\Omega_1 \times \Omega_{2,\varepsilon}}(\cdot,\cdot)$,
$\tilde{\alpha}_p^{k_c} = f(x^{k_c}, y^{k_c}) - \bar{f}^{k_c}_{\Omega_1 \times \Omega_{2,\varepsilon}}(x^{k_c}, y^{k_c})$,
$\bar{f}^{k_c}_{\Omega_1 \times \Omega_{2,\varepsilon}}(\cdot,\cdot) = \check{f}^{k_c}(u^{k_c+1}, v^{k_c+1}) + \langle p^{k_c}, (\cdot,\cdot) - (u^{k_c+1}, v^{k_c+1})\rangle$.

TWO STOPPING CRITERIA

We put SC1 in Step 1 of Alg. 3.1, and SC2 in Step 2 of Alg. 3.1 if necessary.

Algorithm 3.1: (DPLBM)

$f^{k_c}_{lev} = f(x^{k_c}, y^{k_c}) - \delta^{k_c}$, $(d^{k_c}, e^{k_c}) = (u^{k_c+1}, v^{k_c+1}) - (x^{k_c}, y^{k_c})$.
Given positive numbers $t_{\text{max}} > 0$, k_d, k_l, $k_\delta \in (0,1)$.

Step 0 Choose $(x^1, y^1) \in \Omega_1 \times \Omega_{2,\varepsilon}$, $f^1_{\text{low}} \leq f^*$. If $\Delta^1 < \infty$, let $\delta^1 = k_l \Delta^1$, otherwise choose $\delta^1 > 0$. Set $J^1 = \{1\}$, $k_c = 1$, $l = 0$, $k_c(0) = 1$. J^{k_c} has at most N indices.

Step 1 If (3.1) is feasible, then go to Step 2, otherwise choose $f^{k_c}_{\text{low}} \in [f^{k_c}_{\text{lev}}, f^*]$, compute $\Delta^{k_c}, \delta^{k_c} = k_l \Delta^{k_c}$, and go to the beginning of this step.

Step 2 Find the solution (u^{k_c+1}, v^{k_c+1}) of (3.1) and multipliers $\lambda_j^{k_c}$ such that

$$\bar{J}^{k_c} = \{j \in J^{k_c} \mid \lambda_j^{k_c} > 0\}, \quad |\bar{J}^{k_c}| \leq N.$$

Set $t^{k_c} = \sum_{j \in J^{k_c}} \lambda_j^{k_c}$ and compute (d^{k_c}, e^{k_c}).

Step 3 If $t^{k_c} > t_{\max}$, then replace δ^{k_c} by $k_\delta \delta^{k_c}$ and go to Step 1.
If $f(u^{k_c+1}, v^{k_c+1}) \leq f(x^{k_c}, y^{k_c}) - k_d \delta^{k_c}$, then set $t_L^{k_c} = 1$, $k_c(l+1) = k_c + 1$ and $l = l + 1$, otherwise set $t_L^{k_c} = 0$. Compute $(x^{k_c+1}, y^{k_c+1}) = (x^{k_c}, y^{k_c}) + t_L^{k_c}(d^{k_c}, e^{k_c})$.

Step 4 Select $J_s^{k_c} \subset J^{k_c}$ such that $\bar{J}^{k_c} \subset J_s^{k_c}$. Set $J^{k_c+1} = J_s^{k_c} \cup \{k_c + 1\}$.

Step 5 Set $f^{k_c+1}_{\text{low}} = f^{k_c}_{\text{low}}$, compute Δ^{k_c+1}.
Set $\delta^{k_c+1} = \delta^{k_c}$ if $t_L^{k_c} = 0$, otherwise set $\delta^{k_c+1} \in [\min\{\delta^{k_c}, k_l \Delta^{k_c+1}\}, \Delta^{k_c+1}]$.
Let $k_c = k_c + 1$, go to Step 1.

End of Algorithm 3.1

Theorem 3.1. *[14] Either*

$$(x^{k_c}, y^{k_c}) \to (x^*, y^*) \in \Omega^* = \{(x,y) \mid f(x,y) = \inf_{\Omega_1 \times \Omega_{2,\varepsilon}} f(x,y)\}$$

or $\Omega^* = \emptyset$ *and* $\{||(x^{k_c}, y^{k_c})||\}_{k_c=1}^\infty \to \infty$*. In both cases, one has that* $\{f(x^{k_c}, y^{k_c})\}_{k_c=1}^\infty \downarrow f^* = \inf_{\Omega_1 \times \Omega_{2,\varepsilon}} f(x,y)$. □

4 Algorithm

By taking a descent sequence $\{\varepsilon^k\}_{k=1}^\infty \downarrow 0$, we obtain a sequence of solutions $\{(x_k, y_k)\}_{k=1}^\infty$ to problems $\{(P_{\varepsilon^k})\}_{k=1}^\infty$. At the beginning of each iteration for generating (x_k, y_k) one needs to provide a solution

$$y \in \Omega_{2,\varepsilon^k} = \{y \mid \varphi(y) \leq \inf_{y \in S} \varphi(y) + \varepsilon^k\}, \tag{4.1}$$

instead of providing a starting point that belongs to $\text{Arginf}_{y\in S}\varphi(y)$. Then a pair (x_k, y_k) will be generated at the kth iteration satisfying

$$(x_k, y_k) \in \{(x, y)\,|\,f(x, y) = \inf_{(x,y)\in\Omega_1\times\Omega_{2,\varepsilon^k}} f(x, y)\}. \tag{4.2}$$

If ε^k reaches the value that satisfies the stopping criterion, then stop and the solution (x_k, y_k) is an ε^k approximate solution to problem (P). Otherwise, repeat the process presented above until the stopping criterion is satisfied.

Definition 4.1. PBMASL(k) is defined by PBMASL in which ε is replaced by ε^k. DPLBM(k) is defined by DPLBM in which ε is replaced by ε^k. □

Algorithm 4.1: Solve problem (P)

Step 0 Initialization.
Take $\varepsilon^1 > 0,\ \gamma \in (0, 1)$, set $k = 1$.

Step 1 Compute initial point of DPLBM(k).
If $k = 1$, then find $\bar{x}^1 \in \Omega_1$, otherwise set $\bar{x}^k = x_{k-1}$.
Find a $\bar{y}^k \in \Omega_{2,\varepsilon^k}$ using PBMASL(k).

Step 2 Update iterate points of problem (1.1).
Find the kth iterate point (x_k, y_k) starting form $(\bar{x}^k, \bar{y}^k)$ using DPLBM(k).

Step 3 Update ε^k and k.
If $(x_k, y_k) \in \Omega_1 \times \Omega_2$ then stop, otherwise set $\varepsilon^{k+1} = \gamma\varepsilon^k,\ k = k + 1$ and loop to Step 1.

End of Algorithm 4.1

Remark: γ is a contraction parameter, in general, the smaller γ is, the faster the approximate algorithm will converge. γ plays an important role in controlling the accuracy degree of the approximate solution (x_k, y_k) to problem (P).

Lemma 4.1. *Let $\varepsilon_{\text{opt}} > 0$ be a constant. If $\Delta^{k_c} \le \varepsilon_{\text{opt}}$, then (x^{k_c}, y^{k_c}) is a ε_{opt} optimal solution to (4.2) for $\varepsilon = \varepsilon^k$.*

Proof. Since $\Delta^{k_c} = f(x^{k_c}, y^{k_c}) - f^{k_c}_{\text{low}}$ and $f^{k_c}_{\text{low}} < f^* = \inf_{\Omega_1\times\Omega_{2,\varepsilon^k}} f(x, y)$, we have

$$f(x^{k_c}, y^{k_c}) - f^* \le f(x^{k_c}, y^{k_c}) - f^{k_c}_{\text{low}} = \Delta^{k_c}.$$

The condition $\Delta^{k_c} \le \varepsilon_{\text{opt}}$ leads to $f(x^{k_c}, y^{k_c}) \le f^* + \varepsilon_{\text{opt}}$. The lemma is proved. □

Definition 4.2. [3] Given $\varepsilon \ge 0$, $\tilde{x}$ is called an $\varepsilon-$popt optimal solution to f on S if $\tilde{x}$ satisfies

$$f(\tilde{x}) \le f(x) + \varepsilon\|x - \tilde{x}\| + \varepsilon, \quad \forall x \in S.$$

□

Lemma 4.2. *If $\max\{\|p^{k_c}\|, \tilde{\alpha}^{k_c}_p\} \le \varepsilon_{\text{opt}}$, then (x^{k_c}, y^{k_c}) is an ε_{opt}-popt optimal solution to (4.2) for $\varepsilon = \varepsilon^k$.*

Proof. Since $f(x^{k_c}, y^{k_c}) + \langle p^{k_c}, (x, y) - (x^{k_c}, y^{k_c})\rangle - \tilde{\alpha}^{k_c}_p \le f(x, y)$ for all $(x, y) \in \Omega_1 \times \Omega_{2,\varepsilon^k}$ [14], we have

$$f(x^{k_c}, y^{k_c}) \le f(x, y) + \|p^{k_c}\|\|(x, y) - (x^{k_c}, y^{k_c})\| + \tilde{\alpha}^{k_c}_p, \quad \forall (x, y) \in \Omega_1 \times \Omega_{2,\varepsilon^k}.$$

It follows form $\max\{\|p^{k_c}\|, \tilde{\alpha}^{k_c}_p\} \le \varepsilon_{\text{opt}}$ that

$$f(x^{k_c}, y^{k_c}) \le f(x, y) + \varepsilon_{\text{opt}}\|(x, y) - (x^{k_c}, y^{k_c})\| + \varepsilon_{\text{opt}}, \quad \forall (x, y) \in \Omega_1 \times \Omega_{2,\varepsilon^k}.$$

This implies that (x^{k_c}, y^{k_c}) is a ε_{opt}-popt optimal solution to (4.2) for $\varepsilon = \varepsilon^k$. □

Algorithm 4.2: Approximately solve problem (P).

Step 0 Initialization.
Take $\varepsilon^1 > 0,\ \gamma \in (0,1),\ \varepsilon^* > 0$, set $k = 1$.

Step 1 Compute initial point of DPLBM(k).
The same as Step 1 in Algorithm 4.1.

Step 2 Update iterate points.
Solve (4.2) to get the kth iterate point (x_k, y_k) using DPLBM(k) with two stopping criteria SC1 and SC2 starting from $(\bar{x}^k, \bar{y}^k)$. If Alg. 3.1 stops at some index k_c, then set $(x_k, y_k) = (x^{k_c}, y^{k_c})$.

Step 3 Update ε^k and k.
If $\varepsilon^k < \varepsilon^*$, then stop, otherwise set $\varepsilon^{k+1} = \gamma\varepsilon^k,\ k = k+1$ and go to Step 1.

End of Algorithm 4.2

5 Convergence Analysis

Theorem 5.1. *Suppose the optimal solution set of problem (P) is nonempty. Then any accumulation point of the sequence $\{(x_k, y_k)\}_{k=1}^{\infty}$ generated by Algorithm 4.1 is an optimal solution of problem (P).*

Proof. According to the design of Algorithm 4.2, $(x_k, y_k) \in \Omega_1 \times \Omega_{2,\varepsilon^k}$ and

$$\begin{aligned} f(x_k, y_k) \leq &\quad f(x, y), \quad \forall (x, y) \in \Omega_1 \times \Omega_{2,\varepsilon^k}, \\ \varphi(y_k) \leq &\quad \inf_{y\in S}\ \varphi(y) + \varepsilon^k \end{aligned} \tag{5.1}$$

since $(x_k, y_k) \in \{(x, y)|f(x, y) = \inf_{\Omega_1\times\Omega_{2,\varepsilon^k}} f(x, y)\}$. Ω_{2,ε^k} is convex and compact according to the fact that $\varphi(y)$ is convex and level bounded. The sequence $\{(x_k, y_k)\}_{k=1}^{\infty}$ must have accumulation points since Ω_1 is convex and closed. Without loss of generality, we assume that $(x_k, y_k) \to (\hat{x}, \hat{y})$, where $\hat{x} \in \Omega_1$. Functions φ and f are continuous because they are finite and convex. In view of $\varepsilon^k \downarrow 0$, for the second inequality of (5.1), taking the limit we obtain

$$\varphi(\hat{y}) = \inf_{y\in S}\ \varphi(y). \tag{5.2}$$

For the first inequality of (5.1), taking the limit we have

$$\lim_{k\to\infty} f(x_k, y_k) = f(\hat{x}, \hat{y}) \leq f(x, y), \quad \forall (x, y) \in \Omega_1 \times \Omega_{2,\varepsilon^k}. \tag{5.3}$$

Therefore,

$$f(\hat{x}, \hat{y}) \leq f(x, y), \quad \forall (x, y) \in \Omega_1 \times \Omega_2,$$

i. e., $(\hat{x}, \hat{y})$ is an optimal solution of problem (P). □

Theorem 5.2. *The following conclusions hold:*

a. *If (x_k, y_k) is an $\varepsilon_{\text{opt}}-$optimal solution to (4.2) for $\varepsilon = \varepsilon^k$, generated by DPLBM with two stopping criteria, then any accumulation point $(\hat{x}, \hat{y})$ of $\{(x_k, y_k)\}_{k=1}^{\infty}$ satisfies*

$$(\hat{x}, \hat{y}) \in \{(\hat{x}, \hat{y})|f(\hat{x}, \hat{y}) \leq \inf_{(x,y)\in\Omega_1\times\Omega_2} f(x, y) + \varepsilon_{opt}\};$$

b. *If (x_k, y_k) is an $\varepsilon_{opt}-$ popt optimal solution to (4.2) for $\varepsilon = \varepsilon^k$, generated by DPLBM with two stopping criteria, then any accumulation point $(\hat{x}, \hat{y})$ of $\{(x_k, y_k)\}_{k=1}^{\infty}$ satisfies*

$$(\hat{x}, \hat{y}) \in \{(\hat{x}, \hat{y})\,|\,f(\hat{x}, \hat{y}) \leq f(x, y) + \varepsilon_{\text{opt}}\|(x, y) - (\hat{x}, \hat{y})\| + \varepsilon_{\text{opt}}, \forall (x, y) \in \Omega_1 \times \Omega_2\}.$$

Proof. (a) Since $(x_k, y_k) \in \{(x_k, y_k) | f(x_k, y_k) \leq \inf_{(x,y)\in\Omega_1\times\Omega_{2,\varepsilon^k}} f(x, y) + \varepsilon_{\text{opt}}\}$ by Algorithm 4.2, we have

$$f(x_k, y_k) \leq f(x, y) + \varepsilon_{\text{opt}}, \quad \forall (x, y) \in \Omega_1 \times \Omega_{2,\varepsilon^k}, \tag{5.4}$$

$$\varphi(y_k) \leq \inf_{y\in S} \varphi(y) + \varepsilon^k, \tag{5.5}$$

and $x^k \in \Omega_1$. The set Ω_{2,ε^k} is compact since φ is convex and level bounded, so $\{(x_k, y_k)\}_{k=1}^{\infty}$ has accumulation points. Without loss of generality, we may assume that $(x_k, y_k) \to (\hat{x}, \hat{y})$ as $k \to \infty$. For (5.5), taking the limit we obtain $\varphi(\hat{y}) = \inf_{y\in S} \varphi(y)$. It is clear from the compactness of Ω_1 that $\hat{x} \in \Omega_1$. Similarly, for (5.4), taking the limit we have

$$\lim_{k\to\infty} f(x_k, y_k) = f(\hat{x}, \hat{y}) \leq f(x, y) + \varepsilon_{\text{opt}}, \forall (x, y) \in \Omega_1 \times \Omega_{2,\varepsilon^k}. \tag{5.6}$$

Therefore,

$$f(\hat{x}, \hat{y}) \leq f(x, y) + \varepsilon_{\text{opt}}, \forall (x, y) \in \Omega_1 \times \Omega_2.$$

(b) The proof is omitted. □

6 Numerical Tests

We shall now report on numerical testing of Algorithm 4.2 with Matlab-code on the platform of Matlab (R2009b) in a computer with Intel (R) 2 Duo 2.93 GHz CPU and 2.0 GB Memory. All these examples can be found in [17] except for the constraints which are appended by ourselves.

Consider the following problem:

$$(\text{P}) \qquad \begin{cases} \min & f(x, y) \\ \text{s. t.} & (x, y) \in \Omega_1 \times \Omega_2 , \end{cases} \tag{6.1}$$

where $\Omega_1 := \{x \in \mathbb{R} \,|\, A^1 x \leq b^1\}$, $A^1 := 2$, $b^1 := 12$, $\Omega_2 =$Arg $\inf_{y\in S} \varphi(y) = \{y \,|\, \varphi(y) = \inf_{y\in S} \varphi(y)\}$, $\varphi(y) := \text{F2d}(y) := \max\{\, 0.5 * (y_1^2 + y_2^2) - y_2,\, y_2\}$, $S := \{y = (y_1, y_2)^T \in \mathbb{R}^2 \,|\, A^2 y \leq b^2\}$, $A^2 := \begin{bmatrix} -1 & -1 \\ 1 & -1.5 \end{bmatrix}$, $b^2 := (8.5 \quad 6.5)^T$. We take $f(z)$:=F3d_Uv(z) also from [17], with v $= 0, 1, 2, 3$, respectively, these four functions $\text{F3d\_U0}, \text{F3d\_U1}, \text{F3d\_U2}, \text{F3d\_U3}$ of $z = (x, y_1, y_2)^T$ are defined as

$$\text{F3d\_Uv}(z) := \max\{\, 0.5 * (0.1x^2 + y_1^2 + y_2^2) - e^T z - \beta_1^v,\, y_1^2 - 3y_1 - \beta_2^v,\, y_2 - \beta_3^v,\, y_2 - \beta_4^v\},$$

where $e = (0, 1, 1)^T$ and four parameter vectors $\beta^v \in \mathbb{R}^4$ are given with v $= 0, 1, 2, 3$, respectively, $\beta^0 := (0.5, \,-2, \,0, \,0)$, $\beta^1 := (0, \,10, \,0, \,0)$, $\beta^2 := (-5, \,10, \,0, \,10)$ and $\beta^3 := (-5.5, \,10, \,11, \,20)$.

The parameters have values: $k_d = k_l = k_\delta := 0.382$, $t_{max} := 1.0e15$, $\varepsilon_{opt} := 1.0e-6$, $\varepsilon^1 := 1.0e-4$, $\gamma = 0.1$, $\varepsilon^* := 1.0e-8$, $f_{low}^1 := -50$. The maximum iteration number in the PBMASL algorithm is set 1000.

In the subsequent Table 1, z^* and z^0 denote the optimal solution and the initial point, respectively, fz^* indicates the optimal function value.

Table 1: Problem data

Problem	z^*	fz^*	z^0
F3d-U0	(1 0 0)^T	0	(-1 0.9 1.9)^T
F3d-U1	(0 0 0)^T	0	(-1 0.9 1.9)^T
F3d-U2	(0 0 0)^T	5	(-1 0.9 1.9)^T
F3d-U3	(0 0 0)^T	5.5	(-1 0.9 1.9)^T

Table 2: Results of solving four examples

problem	z^T	$(z-z^*)^T$	$fz - fz^*$	seconds
F3d_U0	(1.000e0 -1.919e-7 3.748e-9)	(5.627e-7 -1.919e-7 3.748e-9)	7.508e-7	36.1
F3d_U1	(-1.127e-3 -2.605e-8 1.268e-8)	(-1.127e-3 -2.605e-8 1.268e-8)	6.483e-7	72.4
F3d_U2	(6.257e-4 2.131e-3 1.417e-6)	(6.257e-4 2.131e-3 1.417e-6)	-2.130e-3	105.5
F3d_U3	(-1.183e-4 3.570e-3 2.814e-6)	(-1.183e-4 3.570e-3 2.814e-6)	-3.566e-3	69.4

Table 2 shows the obtained solution z, the difference between z and the optimal solution z^*, the difference between the corresponding function values fz and fz^*, and at the same time the elapsed seconds are listed in Table 2.

Upon studying Table 1 and 2, it can be seen that all the obtained solutions by Algorithm 4.2 are near the optimal solutions. However, in problems $\mathrm{F3d\_U2}, \mathrm{F3d\_U3}$, the obtained objective function values are less than the optimal ones. A possible explanation is that $\Omega_{2,\varepsilon}$ is too larger than Ω_2. We also notice that although the problems considered are only three dimensions, they still cost much time in order to obtain the solution. In fact, some conclusions may be drawn: much time is cost on running the inner algorithm DPLBM, more precisely, on checking the feasibility and solving problem (3.1) which is implemented by the fmincon() function built in the Matlab, this is one drawback of algorithm DPLBM, and may be discussed in the future work.

The favorable testing results demonstrate that it is worthwhile to continue the development of the applications of bundle methods to MPEC problems.

7 Conclusions

In this paper we present an approximate sequential bundle method for solving a MPEC (Mathematical Programs with Equilibrium Constraints) problem (P) by combining two bundle methods PBMASL and DPLBM. The proposal of the algorithm is based on the construction of the approximate problem (P$_\varepsilon$) and by focusing our attention on solving the approximate problem (P$_\varepsilon$) step by step, we finally prove that the optimal solution (x_k, y_k) of problem (P$_{\varepsilon^k}$) converges to the (approximate) optimal solution of problem (P) as $k \to \infty$. We once used the similar technique to solve a MPEC problem with the constraint being an unconstrained optimization problem [1], but in that paper, the exact values of the objective function in the constraints are used. Just like the discussion in the first part of our paper, sometimes it is not so easy or even impossible to compute the exact function values. Our algorithm can be viewed as an improvement to [1]. The presented algorithm, utilizing the approximate function values and approximate subgradients, can be applied to the situations in which the exact objective function values are difficult or even impossible to be computed. For example, consider problem (P) with

$$\varphi(y) = \min_{z \in R^n} \{h(z) + \frac{1}{2\lambda}\|z-y\|^2\}, \tag{7.1}$$

where λ is a fixed positive parameter and $\|\cdot\|$ denotes the Euclidean norm, and we assume h is strongly convex. The function $\varphi(y)$ is called the Moreau-Yosida regularization of $h(z)$, it has the following properties [18]: the function $\varphi(y)$ is convex, everywhere finite and differentiable with Lipschitz continuous gradient given by

$$g(y) = \frac{1}{\lambda}(y - p(y)), \tag{7.2}$$

where $p(y)$ is the unique minimizer of problem (7.1), i.e.,

$$p(y) = \text{arg} \min_{z \in R^n} \{h(z) + \frac{1}{2\lambda} \|z - y\|^2\}. \quad (7.3)$$

Since the Moreau-Yosida regularization $\varphi(y)$ is defined through a minimization problem involving another function $h(z)$, the exact evaluation of the function values of φ and its gradients g at an arbitrary point y is practically impossible in general, therefore, we shall consider using their approximate values. Specifically, suppose that, for each $y \in R^n$ and $\varepsilon > 0$, we can find an approximation $p^a(y, \varepsilon)$ to $p(y)$ such that $h(p^a(y, \varepsilon)) + \frac{1}{2\lambda} \|p^a(y, \varepsilon) - y\|^2 \le \varphi(y) + \varepsilon$. Some implementable algorithms to find such an approximation $p^a(y, \varepsilon)$ to $p(y)$ for a general convex function can be found [2, 19, 20]. With $p^a(y, \varepsilon)$, we define the approximations $\varphi^a(y, \varepsilon)$ and $g^a(y, \varepsilon)$ to $\varphi(y)$ and $g(y)$, respectively, $\varphi^a(y, \varepsilon) = h(p^a(y, \varepsilon)) + \frac{1}{2\lambda} \|p^a(y, \varepsilon) - y\|^2$, $g^a(y, \varepsilon) = \frac{1}{\lambda}(y - p^a(y, \varepsilon))$. For these approximations, we have the following inequalities [20],

$$\varphi(y) \le \varphi^a(y, \varepsilon) \le \varphi(y) + \varepsilon,$$
$$\|g^a(y, \varepsilon) - g(y)\| \le \sqrt{2\varepsilon/\lambda}.$$

These inequalities indicate that the approximations $\varphi^a(y, \varepsilon)$ and $g^a(y, \varepsilon)$ can be made arbitrarily close to the exact values $\varphi(y)$ and $g(y)$ by choosing the parameter ε small enough.

Acknowledgment

The authors acknowledge the anonymous referees and the associate editor for their careful reading and valuable comments..

Competing Interests

The authors have declared that no competing interests exist.

References

[1] Zun-Quan Xia, Jie Shen, Li-Ping Pang: A sequential bundle method for solving a class of MPEC problems, Journal of Information and Computational Science. 2007;4(1):331-336.

[2] Auslender A. Numerical methods for nondifferentiable convex optimization, Mathematical Programming Study. 1987;30:102-126.

[3] Hintermüller M. A proximal bundle method based on approximate subgradients, Computational Optimization and Applications. 2001;20:245-266.

[4] Kiwiel KC. Proximity control in bundle methods for convex nondifferentiable optimization, Mathematical Programming. 1990;46:105-122.

[5] Lemaréchal C, Strodiot JJ, Bihain A. On a bundle algorithm for nonsmooth optimization, in: Nonlinear Programming OL. Magasarian, R.R. Meyer, S.M. Robinson (Eds), Academic Press, NY. 1981;4:245-282.

[6] Outrata J, Kocvara M, Zowe J. Nonsmooth Approach to Optimization Problems with Equilibrium Constraints, Kluwer Acad. Publ., Springer, Berlin; 1998.

[7] Rockafellar RT. Monotone Operators and the Proximal Point Algorithm, SIAM J. on Control and Optimization. 1976;14:877-898.

[8] Schramm H, Zowe J. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results, SIAM J. Optim. 1992;2;121-152.

[9] Kiwiel KC. Approximations in proximal bundle methods and decomposition of convex programs, Journal of Optimization Theory and Applications. 1995;84;529-548.

[10] Kiwiel KC. A proximal bundle method with approximate subgradient linearizations. SIAM J. Optim. 2006;2;1007-1023.

[11] Solodov MV. On approximation with finite precision in bundle methods for nonsmooth optimizatioon, Journal of Optimization Theory and Applications. 2003;119(1):151-165.

[12] Bertsekas DP. Nonlinear programming, Athena Scientific, Belmont, MA; 1999.

[13] Brännlund U. A descent method with relaxation type step, In: J. Henry and J. P. Yvon (Eds), Lecture Notes in Control and Information Sciences, Springer-Verlag, New York. 1994;177-186.

[14] Brännlund U, Kiwiel KC, Lindberg PO. A descent proximal level bundle method for convex nondifferentiable optimization, Operation Research Letters. 1995;17:121-126.

[15] Kiwiel KC. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities, Mathematical Programming. 1995;69:89-109.

[16] Lemaréchal C, Nemirovskii A, Nesterov Yu. New variants of bundle methods, Mathematical Programming. 1995;69:111-147.

[17] Mifflin, R. and Sagastizábal, C.: A UV-algorithm for convex minimization, Mathematical Programming, Ser. B. 2005;104:583-608.

[18] Hiriart-Urruty J, Lemaréchal, C. Convex analysis and minimization algorithms, Springer Verlag, Germany, Berlin; 1993.

[19] Fukushima M. A descent algorithm for nonsmooth convex optimization, Mathematical Programming. 1984;30;163-175.

[20] Correa R, Lemaréchal C. Convergence of some algorithms for convex minimization, Mathematical Programming. 1993;62:261-275.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=615&id=6&aid=5612