Numerical Algorithms **35:** 175–184, 2004.

Some inexact hybrid proximal augmented Lagrangian algorithms

Carlos Humes Jr. [a,*], Paulo J.S. Silva [a,**] and Benar F. Svaiter [b,***]

[a] *Department of Computer Science, University of São Paulo, Brazil*
E-mail: {humes,rsilva}@ime.usp.br
[b] *Instituto de Matemática Pura e Aplicada, Brazil*
E-mail: benar@impa.br

Received 30 December 2001; accepted 29 October 2002

In this work, Solodov–Svaiter's hybrid projection-proximal and extragradient-proximal methods [16,17] are used to derive two algorithms to find a Karush–Kuhn–Tucker pair of a convex programming problem. These algorithms are variations of the proximal augmented Lagrangian. As a main feature, both algorithms allow for a fixed relative accuracy of the solution of the unconstrained subproblems. We also show that the convergence is Q-linear under strong second order assumptions. Preliminary computational experiments are also presented.

Keywords: augmented Lagrangian, proximal methods, convex programming

AMS subject classification: 90C25

1. Introduction

The proximal point algorithm [9–11,13] and its connection to augmented Lagrangian algorithms for nonlinear programming were established in [14]. In this article, Rockafellar showed that the augmented Lagrangian method introduced by Hestenes and Powell [5,12] can be viewed as the proximal point algorithm applied to solve the dual of a nonlinear programming problem. He also introduced the *proximal augmented Lagrangian*, based on the the proximal algorithm used to find a saddle point of the Lagrangian function, which corresponds to a Karush–Kuhn–Tucker (KKT) pair of the nonlinear problem. The convergence of the algorithm was proved assuming summable relative errors in the unconstrained minimization needed at each step. Moreover, under extra second order conditions, the rate of convergence was shown to be Q-linear.

In this work, based on Solodov–Svaiter's hybrid methods [16,17], we derive two variations of Rockafellar's proximal augmented Lagrangian that share the convergence

* Partially supported by PRONEX-Optimization.
** Partially supported by FAPESP Grant 96/09939-0 and PRONEX-Optimization.
*** Supported by CNPq Grant 301200/93-9(RN) and by PRONEX-Optimization.

properties, but that require only constant relative accuracy. Some preliminary computational results are presented.

2. The algorithm

2.1. Definitions and notation

Consider the problem

$$\begin{array}{l} \min\ f(x) \\ \text{s.t.}\ \ g(x) \leqslant 0, \quad x \in \mathbb{R}^n, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to (-\infty, \infty]$ and $g(\cdot)$ has m components $g_i : \mathbb{R}^n \to (-\infty, \infty]$, $i = 1, \ldots, m$. We assume throughout this paper that $f(\cdot)$ and each component of $g(\cdot)$ are convex and lower semi-continuous, i.e.

$$\liminf_{x \to \bar{x}} f(x) \geqslant f(\bar{x}).$$

We also suppose that the relative interior of their effective domains intersect. Moreover, we assume that (1) and its dual have solutions.

The extended Lagrangian function of (1) is:

$$l(x, y) := \begin{cases} f(x) + \displaystyle\sum_{j=1}^{m} y_j g_j(x), & \text{if } y \in \mathbb{R}^m_+ \\ -\infty, & \text{otherwise.} \end{cases}$$

Saddle points of the Lagrangian are the KKT pairs for problem (1) and therefore the pairs of solutions of (1) and its dual.

Associated to the Lagrangian, $l(\cdot, \cdot)$, we have the maximal monotone operator $T_l : (x, y) \mapsto \{(u, v) \mid (u, -v) \in \partial l(x, y)\}$, i.e.:

$$T_l(x, y) \doteq \begin{bmatrix} \partial f(x) + \displaystyle\sum_{j=1}^{m} y_j \partial g_j(x) \\ -g(x) + \mathbf{N}_{\mathbb{R}^m_+}(y) \end{bmatrix},$$

where

$$\mathbf{N}_{\mathbb{R}^m_+}(y)_j = \begin{cases} 0, & \text{if } y_j > 0, \\ (-\infty, 0], & \text{if } y_j = 0, \\ \emptyset, & \text{if } y_j < 0. \end{cases}$$

Due to the convexity assumptions, it is clear that the saddle points of $l(\cdot, \cdot)$ correspond to the zeroes of $T_l(\cdot, \cdot)$.

In [14], Rockafellar introduces the proximal augmented Lagrangian algorithm to find a zero of $T_l(\cdot,\cdot)$. In this method one computes a sequence $\{x^k, y^k\}$, where x^{k+1} is an approximate unconstrained minimizer of the function

$$\phi_k(x) \doteq f(x) + P\big(g(x), y^k, c^k\big) + \frac{1}{2c^k}\big\|x - x^k\big\|^2, \tag{2}$$

$$P(w, y, c) \doteq \frac{1}{2c}\sum_{j=1}^{m}\big[(y_j + cw_j)_+^2 - (y_j)^2\big].^1 \tag{3}$$

Then, one takes $y^{k+1} \doteq \nabla_w P(g(x^{k+1}), y^k, c^k)$.[2]

An approximate solution to the minimization of $\phi_k(\cdot)$ is considered acceptable if

$$\operatorname{dist}\big(0, \partial\phi_k\big(x^{k+1}\big)\big) \leqslant \frac{\varepsilon^k}{c^k}, \qquad \sum_{i=0}^{\infty}\varepsilon^k < \infty; \tag{4}$$

or

$$\operatorname{dist}\big(0, \partial\phi_k\big(x^{k+1}\big)\big) \leqslant \left(\frac{\delta^k}{c^k}\right)\big\|\big(x^{k+1}, y^{k+1}\big) - \big(x^k, y^k\big)\big\|, \qquad \sum_{i=0}^{\infty}\delta^k < \infty. \tag{5}$$

This error tolerance comes directly from the error bounds demanded by the proximal point algorithm in [13]. In particular, an exact solution, x^{k+1}, of the minimization of $\phi_k(\cdot)$ and the corresponding y^{k+1} $(=\nabla_w P(g(x^{k+1}), y^k, c^k))$ form the solution of the exact proximal step.

Rockafellar used the criterion (4) to ensure convergence and (5) to prove Q-linear convergence rate.

2.2. *The hybrid algorithms*

In [16,17], Solodov and Svaiter introduced two variations of the proximal point algorithm to compute zeroes of maximal monotone operators. Their main feature is a less stringent acceptance criterion. To achieve this, a step is done on the direction of an image of the maximal monotone operator calculated at an approximate solution of the proximal step.

If we apply these algorithms to find a zero of $T_l(\cdot,\cdot)$ we have:

1. Initialization: Let $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m_+$ and $\sigma \in [0, 1)$.
2. Iteration: Given (x^k, y^k) and $c^k > 0$.

---

[1] As usual, a_+ denotes $\max\{0, a\}$. We shall use a_- for the corresponding minimum operation and we also allow these operations to be applied componentwise to vectors.

[2] Observe that $P(w, y, c)$ is Lipschitz continuously differentiable with respect to w, but it is not twice differentiable at the points at $(-1/c)y_i$.

(a) Inner step: Find $(\tilde{x}^k, \tilde{y}^k)$ and $\tilde{v}^k \in T_l(\tilde{x}^k, \tilde{y}^k)$, such that

$$\left\| \tilde{v}^k + \frac{1}{c^k}\big((\tilde{x}^k, \tilde{y}^k) - (x^k, y^k)\big) \right\| \leqslant \sigma \left(\frac{1}{c^k}\right) \big\| (\tilde{x}^k, \tilde{y}^k) - (x^k, y^k) \big\|. \tag{6}$$

(b) Extragradient step: If $\tilde{v}^k = 0$, or $(\tilde{x}^k, \tilde{y}^k) = (x^k, y^k)$, Stop.
Otherwise, make a step in the direction $\tilde{v}^k$. The step size must be one of the following, which characterizes each method:

- *Projection-proximal method*[3]

$$\big(x^{k+1}, y^{k+1}\big) \doteq \big(x^k, y^k\big) - \frac{\langle \tilde{v}^k, (x^k, y^k) - (\tilde{x}^k, \tilde{y}^k)\rangle}{\|\tilde{v}^k\|^2} \tilde{v}^k. \tag{7}$$

- *Extragradient-proximal method*

$$\big(x^{k+1}, y^{k+1}\big) \doteq \big(x^k, y^k\big) - c^k \tilde{v}^k. \tag{8}$$

Recall, from last section, that the exact minimization of $\phi_k(\cdot)$, gives a solution of the proximal step, i.e. a pair $(\tilde{x}^k, \tilde{y}^k)$ that has a $\tilde{v}^k \in T_l(\tilde{x}^k, \tilde{y}^k)$ such that:

$$\tilde{v}^k + \frac{1}{c^k}\big((\tilde{x}^k, \tilde{y}^k) - (x^k, y^k)\big) = 0.$$

Then, it is natural to use an approximate solution of this minimization problem to perform the inner step above. The main difficulty here is how to compute a good element in $T_l(\cdot, \cdot)$ that permits to test the inner acceptance criterion (6), and to perform the extragradient step described above. This will be handled by the next two simple results:

Lemma 1. Let $\tilde{x} \in \mathbb{R}^n$ and $y \in \mathbb{R}^m_+$ and $c > 0$. Define

$$\tilde{y} \doteq \nabla_w P\big(g(\tilde{x}), y, c\big) \quad \text{and} \quad \tilde{\nu} \doteq \big(y + cg(\tilde{x})\big)_-.$$

Then

$$\tilde{\nu} \in N_{\mathbb{R}^m_+}(\tilde{y}),$$

and, for any $\tilde{\gamma} \in \partial_x l(\tilde{x}, \tilde{y})$,

$$\tilde{v} \doteq \begin{bmatrix} \tilde{\gamma} \\ -g(\tilde{x}) + \frac{1}{c}\tilde{\nu} \end{bmatrix} \in T_l(\tilde{x}, \tilde{y}).$$

Proof. The result follows trivially, observing that $\nabla_w P(g(\tilde{x}), y, c) = (y + cg(\tilde{x}))_+$. □

In the last lemma, the definition of $\tilde{y}$ comes directly from the proximal augmented Lagrangian step described in section 2.1. Moreover, we have carefully selected a special

[3] In the projection-proximal method we could use a weaker stopping criterion:

$$\|\tilde{v}^k + 1/c^k((\tilde{x}^k, \tilde{y}^k) - (x^k, y^k))\| \leqslant \sigma \max\{\|\tilde{v}^k\|, 1/c^k\|(\tilde{x}^k, \tilde{y}^k) - (x^k, y^k)\|\}.$$

For more details see [7,16].

element of $N_{\mathbb{R}^m_+}(\tilde{y})$ to define $\tilde{v}$. This choice simplifies the test of the inner acceptance criterion (6).

Proposition 1. Let $\tilde{x}^k \in \mathbb{R}^n$ and $y^k \in \mathbb{R}^m_+$ and $c^k > 0$. Defining $\tilde{y}^k$, $\tilde{v}^k$, $\tilde{\gamma}^k$ and $\tilde{v}^k$ as above it follows that:

$$r^k \doteq \tilde{v}^k + \frac{1}{c^k}\big((\tilde{x}^k, \tilde{y}^k) - (x, y)\big) \in \begin{bmatrix} \partial\phi_k(\tilde{x}^k) \\ 0 \end{bmatrix}. \tag{9}$$

Proof. Clearly, $\partial\phi_k(\tilde{x}^k) = \partial_x l(\tilde{x}, \tilde{y}) + 1/c^k(\tilde{x}^k - x^k)$.

On the other hand,

$$\begin{aligned} &-g(\tilde{x}^k) + \frac{1}{c^k}\tilde{v}^k + \frac{1}{c^k}(\tilde{y}^k - y^k) \\ &= -g(\tilde{x}^k) + \frac{1}{c^k}\big[(y^k + c^k g(\tilde{x}^k))_- + (y^k + c^k g(\tilde{x}^k))_+ - y^k\big] \\ &= -g(\tilde{x}^k) + \frac{1}{c^k}(y^k + c^k g(\tilde{x}^k) - y^k) = 0. \end{aligned}$$

This completes the proof. □

Now, using the above definitions, it is easy to present an implementable form of the hybrid algorithms. Since both algorithms are very similar we will focus the following presentation on the hybrid extragradient-proximal algorithm, which is simpler and performed slightly better in our experiments.[4]

1. Initialization: Let $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m_+$ and $\sigma \in [0, 1)$.
2. Iteration: Given (x^k, y^k) and $c^k > 0$, define $P(\cdot, y^k, c^k)$ as in equation (3). Define also

$$\varphi_k(x) \doteq f(x) + P\big(g(x), y^k, c^k\big),$$
$$\phi_k(x) \doteq \varphi_k(x) + \frac{1}{2c^k}\|x - x^k\|^2.$$

   (a) Inner optimization: Find $\tilde{x}^k$ approximate solution of the unconstrained minimization of $\phi(\cdot)$ such that

$$\|\nabla\phi_k(\tilde{x}^k)\| \leqslant \sigma\left(\frac{1}{c^k}\right)\|\tilde{z}^k\|, \tag{10}$$

   where

$$\tilde{v}^k \doteq \begin{bmatrix} \nabla\varphi_k(\tilde{x}^k) \\ \frac{1}{c^k}(y^k - \tilde{y}^k) \end{bmatrix}, \qquad \tilde{z}^k \doteq \begin{bmatrix} \tilde{x}^k - x^k \\ \tilde{y}^k - y^k \end{bmatrix}$$

[4] Moreover, to simplify the presentation we will assume that the objective function and the constraints are differentiable. From the previous results, the adaptation to non-differentiable case should be straightforward.

with

$$\tilde{y}^k \doteq \nabla_w P\big(g(\tilde{x}^k), y^k, c^k\big).$$

(b) Extragradient step: If $\tilde{v}^k = 0$, or $\tilde{z}^k = 0$, Stop.
Otherwise,

$$\big(x^{k+1}, y^{k+1}\big) \doteq \big(x^k, y^k\big) - c^k \tilde{v}^k.$$

Note that, with our particular choices for $\tilde{y}^k$ and $\tilde{v}^k$, the acceptance criterion (6) becomes the very simple formula (10).

The main convergence theorems are:

Theorem 1. If the problem (1) and its dual have solutions and the sequence of penalization parameters, $\{c^k\}$, is bounded away from zero; then the sequence generated by the inexact hybrid extragradient-proximal augmented Lagrangian converges to a pair of solutions of these problems (a KKT pair).

Proof. This is a corollary of the convergence of the hybrid extragradient-proximal point [17, theorem 3.1]. □

Theorem 2. Under the assumptions of theorem 1, if $T_l^{-1}(\cdot,\cdot)$ is Lipschitz continuous at the origin then the convergence rate is at least Q-linear.

Proof. The result follows from the convergence rate of the hybrid extragradient-proximal algorithm [17, theorem 3.2]. □

Note that the Lipschitz continuity of $T_l^{-1}(\cdot,\cdot)$ can be guaranteed under strong second order conditions, as shown in [14, pp. 102, 103].

It is important to stress that, to the authors' knowledge, this is the *first* convergence result of an optimization method similar to the augmented Lagrangian algorithm that does not require increasing relative accuracy, i.e. the σ is held constant during the whole process. All the convergence results, so far, asked the relative accuracy to decrease to zero [1,2,14].

3. Computational experiments

In this section, we present some preliminary computational results to demonstrate the applicability of the above algorithm. We also compare it to two different algorithms:

1. The ordinary Proximal augmented Lagrangian method, with the stringent error acceptance criterion (4), presented in the beginning of section 2.[5]
2. The ordinary augmented Lagrangian with errors, as presented in [1, chapter 5], usually implemented in practice. We remind the reader that in [14], Rockafellar showed

[5] The method described in [14], that does not depend on the extragradient step.

that this algorithm can be seen as the proximal point method applied to the dual of (1). We present the algorithm for the sake of completeness:

(a) Initialization: Let $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m_+$ and $\{\varepsilon^k\}$ a sequence converging to zero.

(b) Iteration: Given (x^k, y^k) and $c^k > 0$, define $P(\cdot, y^k, c^k)$ as in equation (3), and define $\varphi_k(\cdot)$ as in the hybrid algorithm:

$$\varphi_k(x) \doteq f(x) + P\big(g(x), y^k, c^k\big).$$

(i) Inner optimization: Find an approximate solution of the unconstrained optimization of $\varphi_k(\cdot)$, x^{k+1}, such that

$$\big\|\nabla\varphi_k\big(x^{k+1}\big)\big\| \leqslant \frac{\varepsilon^k}{c^k}\big\|\nabla_w P\big(g\big(x^{k+1}\big), y^k, c^k\big) - y^k\big\|.$$

(ii) Multiplier update: Define $y^{k+1} \doteq \nabla_w P(g(x^{k+1}), y^k, c^k)$.

3.1. Implementation details

We have implemented the methods in Fortran 90. Since they are very similar the results should not be influenced by implementation details. To solve the unconstrained optimizations problems, we used the LBFGS-B code from Byrd et al. [3], which is freely available at the OTC site.[6]

Moreover, we did not try to fine-tune the parameters of the algorithms to achieve better performance in *each* problem. The parameters were chosen to be robust, i.e. to guarantee convergence in all the problems tested. Some parameters that must be described are:

1. *Initialization*. The initial primal–dual pair was chosen randomly in $[-2, 2] \times [0, 2]$. These values are in the order of magnitude of the solutions.

2. *Stopping criterion*. Since the solutions to all problems are known, we decided to use as stopping criterion ε-optimality and ε-feasibility. Formally, let x^k be an approximate solution and f^* be the optimal value. The point x^k is accepted if:

$$\big|f(x) - f^*\big| \leqslant \max\big(\varepsilon_1, \varepsilon_2|f^*|\big);$$
$$g_i(x) \leqslant \varepsilon_3, \quad \forall i = 1, \ldots, m.$$

   Here $\varepsilon_1 = 5.0\text{E}{-5}$ and $\varepsilon_2 = \varepsilon_3 = 1.0\text{E}{-4}$.

3. *Update of the penalty parameter* c^k. We have decided to keep c^k fixed. Otherwise, a slower method could force c^k to increase faster, hiding its deficiency.

[6] The source code is freely available at the site `http://www.ece.northwestern.edu/OTC/OTCsoftware.htm`.

4. *Stopping criterion for the inner step in the hybrid method*. Instead of using the original acceptance criterion given in (10), we decided to use a simpler threshold,

$$\|\nabla\phi_k(\tilde{x}^k)\| \leqslant \sigma\frac{1}{c^k}\|\tilde{x}^k - x^k\|,$$

that is faster to compute. The chosen σ was 0.9. If we had used the original acceptance test, smaller values for σ would be better.

5. *The error control sequence* $\{\varepsilon^k\}$. For the ordinary augmented Lagrangian, we have used

$$\varepsilon^k \doteq \frac{1}{1+k/5},$$

which is based on the harmonic sequence. This sequence was chosen because it goes slowly to zero and worked well in our tests.
For the proximal augmented Lagrangian, we have used the same sequence squared.[7]

The code was run on a PC class computer based on the AMD K6-2 300 MHz CPU and with 128 MB of main memory. Linux was the operating system. The compiler used was the Intel Fortran Compiler, version 5.0.1. Finally, each problem was solved one thousand times to minimize start up effects and to randomize the starting point.

3.2. The tests problems

The following convex test problems were used:

- From Hock and Schittkowski collection [6,15]: problems 21, 28, 35, 51, 76, 215, 218, 224, 262, 268, 284, 315, 384;
- From Lasdon [8]: problems 1 and 3.

These are all small scale problems, with up to 15 variables and 10 constraints that are clearly convex.

3.3. Computational results

Table 1 presents the processing time in seconds used to solve each test problem a thousand times. We also show the number of unconstrained minimizations used by each method to find an approximate solution to the constrained problem.[8] This value will be used to better explain the behavior of the methods.

The results confirm that the looser acceptance criterion followed by the extra-gradient step has a positive effect on the computational time. Actually, considering the mean behavior in all problems, the hybrid version of the proximal augmented Lagrangian used only 87% of the time used by the version with summable errors, without increasing the number of unconstrained minimizations.[9]

---

[7] The error sequence of the proximal augmented Lagrangian method must be summable.

[8] The column #min.

[9] This mean behavior is the geometric mean of the ratios of the times in both methods.

Table 1
Performance results.

Problem	Aug. Lagrangian		Prox. Lagrangian		Hybrid	
	Time	#min	Time	#min	Time	#min
S21	1.11	4065	2.02	9186	1.94	9260
S215	2.48	16049	4.78	24106	3.71	24653
S218	1.41	4723	2.76	15523	3.46	17004
S224	9.07	34757	8.86	28169	7.80	28785
S262	8.38	9796	13.70	33515	19.20	48263
S268	18.80	3819	16.60	6875	17.30	8164
S28	4.68	3505	5.06	10925	3.78	8181
S284	11.80	8079	15.60	22222	10.80	14677
S315	1.82	6795	3.25	12365	2.65	12403
S35	2.56	7440	4.55	11881	3.79	12113
S384	43.80	38737	71.80	37598	38.40	37611
S51	6.16	15991	8.14	15017	6.63	15462
S76	6.49	14248	12.80	24164	9.83	27405
Lasdon1	2.61	12451	2.50	10743	2.36	11397
Lasdon3	14.40	29682	18.60	28650	16.10	28706

On the other hand, when compared to the augmented Lagrangian without a primal regularization, the Hybrid method is still slower. This seems to be a consequence of an increase in the number of unconstrained minimizations required by the methods using the primal regularization. Actually, the mean time used by the hybrid method is 25% bigger than the one used by the ordinary augmented Lagrangian method and the number of unconstrained minimizations increased 67%. Hence, although less work is done at each minimization due to the new acceptance criterion, the higher number of unconstrained minimizations is still a bottleneck.

This last observation raises the question of whether it is possible to use the hybrid algorithm to solve directly the dual problem, deriving an augmented Lagrangian method with a better error criterion. Unfortunately, this is not a straightforward extension of the ideas presented in this paper. The error criterion and the extragradient step would need an element in the subgradient of the dual function, which requires a *full* unconstrained minimization to compute. This extension should be the subject of further investigation.

References

[1] D. Bertsekas, *Constrained Optimization and Lagrange Multipliers* (Academic Press, New York, 1982).

[2] D. Bertsekas, *Nonlinear Programming* (Athena Scientific, 1995).

[3] R.H. Byrd, P. Lu, J. Nocedal and C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. 16 (1995) 1190–1208.

[4] D.E. Dennis Jr. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (SIAM, Philadephia, PA, 1996).

[5] M.D. Hestenes, Multiplier and gradient methods, J. Optim. Theory Appl. 4 (1969) 303–320.

[6] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes* (Springer, Berlin, 1981).

[7] C. Humes Jr., P.J.S. Silva and B. Svaiter, An inexact projection-proximal augmented Lagrangian algorithm, in: *Proc. of the 36th Annual Allerton Conf. on Communication, Control, and Computing*, 1998, pp. 450–459.

[8] L.S. Lasdon, An efficient algorithm for minimizing barrier and penalty functions, Math. Programming 2 (1972) 65–106.

[9] B. Lemaire, The proximal algorithm, in: *International Series of Numerical Mathematics*, ed. J.P. Penot (Birkhäuser, Basel, 1989) pp. 73–87.

[10] B. Martinet, Determination approché d'un point fixe d'une application pseudo-contractante, C. R. Acad. Sci. Paris Sér. A/B 274 (1972) A163–A165.

[11] J. Moreau, Proximité et dualité dans un espace hilbertien, Bull. Soc. Math. France 93 (1965) 273–299.

[12] M.J.D. Powell, A method for nonlinear constraints in minimization problems, in: *Optimization*, ed. R. Fletcher (Academic Press, New York, 1969) pp. 283–298.

[13] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim. 14 (1976) 887–898.

[14] R.T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, Math. Oper. Res. 1 (1976) 97–116.

[15] K. Schittkowski, *More Test Examples for Nonlinear Programming Codes* (Springer, Berlin, 1987).

[16] M. Solodov and B. Svaiter, A hybrid projection-proximal point algorithm, J. Convex Anal. 6 (1999) 59–70.

[17] M. Solodov and B. Svaiter, A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator, Set-Valued Analysis 7 (1999) 323–345.