A taxonomy of descent algorithms for nonlinear programs and variational inequalities

Michael Patriksson*

Abstract.

We classify descent algorithms for nonlinear programs and variational inequalities, based on their characterizations as cost approximation algorithms; an eight-tuple of parameters describes the problem and the choices of cost approximating mappings and merit functions. The taxonomy is illustrated on classical algorithms and is utilized to interrelate known algorithm frameworks.

Keywords. Taxonomy, classification, nonlinear programs, variational inequalities, descent algorithms, cost approximation.

1 Introduction and motivation

The development of numerical algorithms in nonlinear programming has been dramatic during the past few decades. More recently, finite-dimensional complementarity and variational inequality problems (which constitute generalizations of nonlinear programs) have attracted a lot of attention in the modelling of problems in the sciences; already, there is a large number of algorithms available for their solution (see [9] for an overview of applications and algorithms for complementarity and variational inequality problems). These developments have taken place in a wide variety of application areas, including engineering, numerical analysis, and the economical and transportation sciences. Partly because of the diversity of the areas of application, the essentially same algorithm may exist under several different names and descriptions; further, because of the different natures of the problems being solved and an emphasis on different features of the algorithm, it may have been established under slightly different assumptions on the problem data and the parameters of the algorithm.

Although a significant amount of cross-fertilization is taking place, the terminological confusion contributes to the difficulty of transferring developments in one area to another; the importance of efficient numerical nonlinear programming algorithms in applications serves as a main motive for consolidating the field.

Motivated by a desire to determine the relationships between existing iterative algorithms and to establish the most general convergence results possible, the author has developed a unified framework of algorithms, the class of *cost approximation* algorithms. The unification provides the means to summarize the existing knowledge, and a tool for investigating relationships and performing analyses and interpretations; moreover, this structured presentation of the algorithm class facilitates the construction and exploration of new algorithms, specialized to certain problem structures. It also underlines the limitations of descent algorithms. A summary of the

*Department of Mathematics, Linköping Institute of Technology, S-581 83 Linköping, Sweden; presently at Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195-4350

results obtained can be found in the author's doctoral dissertation [24].

Previously, taxonomies have been introduced for computer architectures [10], for models in queuing theory [14] and automata theory [1], for sequencing and scheduling problems [5], and in the biological sciences [11], among other areas, as a means to structure the existing knowledge, consolidate the field and to facilitate further developments. As far as the author knows, however, no proposals have been made to provide a systematic description of algorithms for continuous optimization problems or variational inequalities; this paper serves as a first attempt in this direction.

A cost approximation algorithm is characterized by very few choices of algorithm parameters; therefore, an algorithm can be interpreted and two algorithms related to each other relatively easily, and it is also possible to compare two algorithms' requirements for guaranteeing convergence with the use of the framework. Based on the cost approximation concept, we will in this paper provide a classification of descent algorithms for nonlinear programs and variational inequalities.

In the next section, we introduce the problem and the cost approximation algorithm. The taxonomy is presented in Section 3. In Section 4 we use the proposed taxonomy to relate algorithm frameworks for the solution of [VIP] to each other and to the class of cost approximation methods. In Section 5 we provide some examples of instances of the general algorithm class, to give an idea of its generality. We conclude with some examples of the usefulness of the taxonomy and possible extensions.

2 The cost approximation algorithm

Let X be a nonempty, closed and convex subset of $\Re^n$, $F : X \mapsto \Re^n$ be continuous on X, and $u : \Re^n \mapsto \Re \cup \{+\infty\}$ be lower semicontinuous, proper and convex. Consider the variational inequality problem of finding an $x^* \in X$ such that

[VIP]

$$F(x^*)^{\mathrm{T}}(x - x^*) + u(x) - u(x^*) \geq 0, \qquad \forall x \in X.$$

We let Ω denote the set of solutions to [VIP].

Patriksson [27, 24] presents and analyzes a general algorithm framework for the solution of [VIP]. One iteration of the basic algorithm consists of two main steps, which can briefly be described as follows.

(1) Given a feasible point, a search direction is defined through the (possibly inexact) solution of an approximation of the original problem, in which the mapping F is approximated by a monotone mapping.

(2) The direction defined by the solution to the above described subproblem is a feasible direction of descent with respect to a merit function whose (local) minima coincide with the set of solutions to [VIP]. A step length is taken in the direction obtained, possibly through a (inexact) line search, so that the new point receives a reduced value of the merit function.

Characteristic features of an algorithm included in the framework are hence that it generates a sequence of feasible solutions to the problem, with a decreasing value of a merit function, and that the descent directions are given through the solution of subproblems, defined by certain approximations of the original problem defining map.

The cost approximation algorithm is derived as follows. Let $x \in X$. Introduce a continuous and monotone *cost approximating mapping* $\Phi : X \mapsto \Re^n$. If the cost mapping F is replaced by Φ, the error made in the approximation obviously is $F - \Phi$. This error is taken into account by adding to Φ the fixed error term $F(x) - \Phi(x)$. Thus, we arrive at a variational inequality subproblem, in which a point $y(x) \in X$ is sought such that

[VIP$_\Phi$]

$$[\Phi(y(x)) + F(x) - \Phi(x)]^{\mathrm{T}}(y - y(x)) + u(y) - u(y(x)) \geq 0, \qquad \forall y \in X.$$

If Φ is chosen as the gradient mapping of a differentiable convex function $\varphi : X \mapsto \Re$, the subproblem [VIP$_\Phi$] amounts to solving the convex problem

$$\psi(x) = \max_{y \in X} \{u(x) - u(y) + \varphi(x) - \varphi(y) + [F(x) - \nabla\varphi(x)]^{\mathrm{T}}(x - y)\}, \tag{1}$$

and the value $\psi(x)$ (which is nonnegative, and zero if and only if $x \in \Omega$) is a measure of the distance from any $x \in X$ to the solution set of [VIP]. Thus, by applying the cost approximation concept, [VIP] is solved as a sequence of convex optimization problems.

In general, we can not expect the original problem to be solved by $y(x)$. A new iteration point is defined through a line search in the direction of $d = y(x) - x$ with respect to a merit function, $\psi : X \mapsto \Re \cup \{+\infty\}$, whose (local) minima coincide with the set of solutions to [VIP]. When the problem [VIP] corresponds to a mathematical program, in the sense that the mapping F is the gradient of a function f, it is natural to choose $\psi = f + u$ as merit function. If this is not the case, another merit function must be identified [e.g., (1)].

At the new point, the original cost mapping is again approximated—perhaps with another mapping Φ—and the algorithm proceeds until some stopping criterion is fulfilled.

We summarize the different steps of the general algorithm below.

A sequence $\{\Phi^k\}$ and $\{\psi^k\}$ of continuous and monotone cost approximating mappings and merit functions, respectively, is assumed to be given. (Note, however, that they may also be constructed adaptively, given x^k, instead of being chosen a priori.)

Step 0 (*Initialization*): Choose an initial point $x^0 \in X$, and let $k = 0$.

Step 1 (*Search direction generation*): Find a $y^k \in X$ such that

[VIP$^k_{\Phi^k}$]

$$[\Phi^k(y^k) + F(x^k) - \Phi^k(x^k)]^{\mathrm{T}}(y - y^k) + u(y) - u(y^k) \geq 0, \qquad \forall y \in X.$$

The resulting search direction is $d^k = y^k - x^k$.

Step 2 (*Stopping criterion*): If x^k solves [VIP$^k_{\Phi^k}$] $\rightarrow$ Stop (x^k solves [VIP]). Otherwise, continue.

Step 3 (*Line search*): Find a step length, t_k, which solves the one-dimensional problem

$$t_k \in \arg\min \{\psi^k(x^k + td^k) \mid t \geq 0; \quad x^k + td^k \in X\}.$$

Step 4 (*Update*): Let $x^{k+1} = x^k + t_k d^k$, and $k := k + 1$.

Step 5 (*Stopping criterion*): If x^k is acceptable as a solution $\rightarrow$ Stop. Otherwise, go to Step 1.

Although both the subproblem solution (Step 1) and the line search (Step 3) are, in the above description, performed exactly, it is in [24, 25] shown that for some instances of [VIP] they can both be done approximately, while still ensuring convergence of the algorithm.

Many well known algorithms for [VIP] can be identified as special cases from the class of cost approximation methods, by appropriately choosing $\{\Phi^k\}$ and $\{\psi^k\}$. [A less general form of cost approximating mapping is where the sequence $\{\Phi^k\}$ is defined by an a priori defined continuous cost approximating mapping on $X \times X$; then, at iteration k, $\Phi^k = \Phi(\cdot, x^k)$. The least general form is where the cost approximating mapping is an iteration independent function on X, i.e., a mapping of the form $\Phi(\cdot)$.] Algorithms included in the framework are (quasi-)Newton methods, SQP methods, the Frank–Wolfe algorithm, steepest descent methods in unconstrained optimization, subgradient optimization algorithms for nondifferentiable programs, gradient projection methods, descent algorithms for saddle point problems, and Jacobi and Gauss–Seidel type decomposition methods for problems defined over Cartesian product feasible sets; some of these algorithms are further discussed in Section 5; other examples, and detailed analyses of the above framework, are found in [24, 25, 26, 27].

For some instances of [VIP], solved by iterative algorithms in the above given class, line searches are considered too expensive computationally, and are replaced by relaxation steps (often predetermined) in the direction of d^k. (Algorithms in which $t_k \equiv 1$ is chosen are usually called *successive approximation algorithms.*) The convergence of such algorithms often relies on a monotone decrease in the Euclidean distance to the set of solutions to [VIP]. Such algorithms can also be included in the above algorithm framework, by letting

$$\psi(x) = \inf_{y \in \Omega} \|x - y\|^2, \tag{2}$$

and by showing that there exists a range of step lengths in the direction of d^k which yields a sufficient decrease in ψ.

3 A taxonomy of descent algorithms

To describe an instance of the problem [VIP], we introduce the three-tuple $[X, F, u]$ describing the feasible set and the two problem-defining mappings.

Remark 1 An instance of [VIP] is determined uniquely by this tuple up to the addition of an arbitrary differentiable, convex function to u and subtraction by its gradient mapping from F; moreover, any convex constraint may be added to the description of the feasible set X, or through its indicator function to the convex function u.

The subproblem $[\mathrm{VIP}^k_{\Phi^k}]$ is characterized by the mapping Φ^k. We also introduce the notation M, S and U meaning that, in an existing convergence analysis, the cost approximating mapping is assumed to be monotone, strictly monotone, and uniformly (or strongly) monotone on X, respectively. (Recall that the mapping F is monotone on X if $[F(x) - F(y)]^{\mathrm{T}}(x - y) \geq 0$ for all x and y in X, strictly monotone if the inequality is strict whenever $x \neq y$, and uniformly monotone if the right-hand-side of the inequality may be replaced by $m_F\|x - y\|^2$ for some positive constant m_F.) Further, we let E and A indicate, respectively, whether the subproblem is assumed to be solved exactly or if it may be solved approximately only. Consequently, the subproblem $[\mathrm{VIP}^k_{\Phi^k}]$ is described by the three-tuple $[\Phi^k, MSU, EA]$.

Remark 2 The subproblem $[\mathrm{VIP}^k_{\Phi^k}]$ is uniquely determined by the mapping Φ^k only up to the addition of a fixed cost to Φ^k. Also note that for a particular instance of a cost approximation algorithm, the monotonicity requirements on Φ^k may implicitly impose conditions on the mapping F (cf. Newton's method).

The line search step in iteration k is described by the choice of a merit function ψ^k, and the accuracy by which the line search is made. In a convergence analysis, the line search is made exactly (E), approximately (A), or it is shown that a sufficient decrease in ψ^k is obtained from a relaxation step (R). If a contractive argument is used (i.e., the merit function used is artificial, as in the case of (2), and can not be evaluated and used in a line search), we replace ψ^k by C; in this case, the step length rule will always be R.

To conclude, an instance of the cost approximation algorithm is described by the eight-tuple

$$([X, F, u];\ [\Phi^k, MSU, EA];\ [\psi^k C, EAR]). \tag{3}$$

4 Relations between algorithm frameworks

In Table 1, we provide a list of several existing algorithm frameworks for the solution of [VIP]. Each entry corresponds to an algorithm class, for which there are known convergence results. For each entry, the left column contains a reference to the article(s) in which the algorithm class is proposed and established convergent. The right column contains the class' description as a cost approximation algorithm in the taxonomy (3); these descriptions are derived from analyses made in [27, 22, 23, 24, 13, 25, 28]. In the table, γ_k is a positive parameter. The reader should note that in addition to the assumptions stated in Section 1 on the parameters in the eight-tuple (3), some technical assumptions are necessary in order to establish convergence of the algorithms given in Table 1.

5 Instances of cost approximation algorithms

In Table 2, we provide a number of examples of algorithms included in the framework of cost approximation, as described in terms of the eight-tuple (3). For each entry, the left column contains the most common name under which the algorithm is known, and in the right column a reference to a work where a convergence analysis is made. In the table, '$\cdot$' denotes the identity mapping, γ_k a positive parameter, $\mathcal{C}$ an index set, and B_k a positive semidefinite matrix of appropriate dimension. The descriptions are derived from relations between these algorithms and the cost approximation algorithm investigated in [22, 23, 24, 25].

6 Remarks

An immediate result of the classification is that we find, from Table 1, that most of the algorithm frameworks that have been developed for the solution of [VIP] and its special cases have been established both with stronger assumptions than necessary and with weaker results than possible and that have been obtained elsewhere. (The monotonicity requirements on F are, according to the analysis made in [24, 25, 26, 27], only implicit, in the sense that the requirements are given through the dependency on F of the cost approximation mapping corresponding to a certain algorithm instance; in the analysis made for many of the algorithm frameworks given in Table 1 the mapping F is, however, a priori supplied with a monotonicity property which is not always necessary.) We may also note the strong relationship between the monotonicity properties of the cost approximating mappings and the freedom of choosing step lengths in Step 3.

The simplified description of the class of cost approximation algorithms which the classification provides provokes interesting questions for future research, and serves as a source of hypotheses; is it, for example, possible to establish the convergence of an algorithm of the form

$$([X, F, u];\ [\Phi^k, U, A];\ [\psi, A]),$$

i.e., a descent algorithm with a general cost approximation mapping, that is, a mapping which is allowed to be discontinuous in the iterates (and therefore may adapt more to the conditions at x^k), while both the direction-finding and line search problems are solved approximately only?

The taxonomy may be useful as a tool for identifying which algorithms may be applied; in a practical situation where a problem with known properties is to be solved, the taxonomy can be used to choose among available software. The taxonomy has in [29] been utilized in a unified presentation of algorithms for the solution of traffic equilibrium problems.

The taxonomy is constructed based on the given problem formulation [VIP]. It should then be observed that a problem can often be transformed between several equivalent forms. Consequently, the list of algorithms given in Table 2 can be expanded with, for example, SQP and multiplier methods (where [VIP] represents primal-dual optimality conditions), reduced gradient algorithms (where the problem is represented in terms of basic and non-basic variables), and interior or exterior penalty methods, to name but a few.

The cost approximation algorithm may be extended in various ways. For example, the function u may be allowed to be linearized, and the subdifferential mapping ∂u may be replaced by a more general point-to-set mapping; thus, subgradient optimization algorithms and descent algorithms for generalized variational inequalities may be analyzed within the cost approximation framework.

Also the taxonomy may be extended. By including the weakest monotonicity properties of F possible in existing analyses into the classification, we would observe that all descent algorithms for the most general form of the problem [VIP] require F to be at least monotone; the removal of this requirement is an interesting topic for future research. The taxonomy could also be extended to more precisely describe the structure and properties of the problem, such as the linearity or nonlinearity of constraints and the separability and differentiability properties of F.

The future utilization of the proposed classification is, of course, conditional of the agreement that the description of descent algorithms using the cost approximation concept is a natural and sufficiently general one. We hope that this paper may provoke future discussions on profitable ways of presenting descent algorithms.

References

[1] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] G. Cohen, *Optimization by decomposition and coordination: a unified approach*, IEEE Transactions on Automatic Control, AC-23 (1978), pp. 222–232.

[3] G. Cohen, *Auxiliary problem principle and decomposition of optimization problems*, Journal of Optimization Theory and Applications, 32 (1980), pp. 277–305.

[4] G. Cohen, *Auxiliary problem principle extended to variational inequalities*, Journal of Optimization Theory and Applications, 59 (1988), pp. 325–333.

[5] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling*, Addison-Wesley, Reading, MA, 1967.

[6] S. Dafermos, *An iterative scheme for variational inequalities*, Mathematical Programming, 26 (1983), pp. 40–47.

[7] J.-P. Dussault, J. A. Ferland, and B. Lemaire, *Convex quadratic programming with one constraint and bounded variables*, Mathematical Programming, 36 (1986), pp. 90–104.

[8] A. A. Goldstein, *Convex programming in Hilbert space*, Bulletin of the American Mathematical Society, 70 (1964), pp. 709–710.

[9] P. T. Harker and J.-S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications*, Mathematical Programming, 48 (1990), pp. 161–220.

[10] R. W. Hockney and C. R. Jesshope, *Parallel Computers 2: Architecture, Programming and Algorithms*, Adam Hilger, Bristol, 1988.

[11] N. Jardine and R. Sibson, *Mathematical Taxonomy*, John Wiley & Sons, London, 1971.

[12] T. Larsson and A. Migdalas, *An algorithm for nonlinear programs over Cartesian product sets*, Optimization, 21 (1990), pp. 535–542.

[13] T. Larsson and M. Patriksson, *A class of gap functions for variational inequalities*, Mathematical Programming, 64 (1994), pp. 53–79.

[14] A. Lee, *Applied Queueing Theory*, Macmillan, Toronto, 1966.

[15] E. S. Levitin and B. T. Polyak, *Constrained minimization methods*, USSR Computational Mathematics and Mathematical Physics, 6 (1966), pp. 1–50.

[16] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, second ed., 1984.

[17] P. Marcotte and D. Zhu, *Global convergence of descent processes for solving non strictly monotone variational inequalities*, Computational Optimization and Applications, 4 (1995), pp. 127–138.

[18] A. Migdalas, *Mathematical programming techniques for analysis and design of communication and transportation networks*, Ph.D. dissertation, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1988.

[19] A. Migdalas, *A regularization of the Frank–Wolfe method and unification of certain nonlinear programming methods*, Mathematical Programming, 65 (1994), pp. 331–345.

[20] M. A. Noor, *General algorithm for variational inequalities*, Journal of Optimization Theory and Applications, 73 (1992), pp. 409–413.

[21] J.-S. Pang and D. Chan, *Iterative methods for variational and complementarity problems*, Mathematical Programming, 24 (1982), pp. 284–313.

[22] M. Patriksson, *Partial linearization methods in nonlinear programming*, Journal of Optimization Theory and Applications, 78 (1993), pp. 227–246.

[23] M. Patriksson, *A unified description of iterative algorithms for traffic equilibria*, European Journal of Operational Research, 71 (1993), pp. 154–176.

[24] M. Patriksson, *A unified framework of descent algorithms for nonlinear programs and variational inequalities*, Ph.D. dissertation, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1993.

[25] M. Patriksson, *Cost approximation: a unified framework of descent algorithms for nonlinear programs*, Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195-4350, 1994.

[26] M. Patriksson, *Decomposition methods for differentiable optimization problems on Cartesian product sets*, Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195-4350, 1994.

[27] M. Patriksson, *A descent algorithm for a class of variational inequalities*, Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195-4350, 1994.

[28] M. Patriksson, *On the convergence of descent methods for monotone variational inequalities*, Operations Research Letters, 16 (1994), pp. 265–269.

[29] M. Patriksson, *The Traffic Assignment Problem: Models and Methods*, VSP, Utrecht, The Netherlands, 1994.

[30] B. N. Pshenichny and Yu. M. Danilin, *Numerical Methods in Extremal Problems*, MIR Publishers, Moscow, 1978.

[31] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of Operations Research, 1 (1976), pp. 97–116.

[32] P. Tseng, *Decomposition algorithm for convex differentiable minimization*, Journal of Optimization Theory and Applications, 70 (1991), pp. 109–135.

[33] J. H. Wu, M. Florian, and P. Marcotte, *A general descent framework for the monotone variational inequality problem*, Mathematical Programming, 61 (1993), pp. 281–300.

[34] D. L. Zhu and P. Marcotte, *Co-coercivity and its role in the convergence of iterative schemes for solving variational inequalities*, Publication CRT-926, Centre de Recherche sur les Transports, Université de Montréal, Montréal, Canada, 1993.

[35] D. L. Zhu and P. Marcotte, *Modified descent methods for solving the monotone variational inequality problem*, Operations Research Letters, 14 (1993), pp. 111–120.

[36] D. L. Zhu and P. Marcotte, *An extended descent framework for variational inequalities*, Journal of Optimization Theory and Applications, 80 (1994), pp. 349–366.

Reference	Description
[18, 12]	$([X, \nabla f, 0];\ [\nabla\varphi(\cdot), S, E];\ [f, E])$
[7]	$([X, \nabla f, 0];\ [\nabla\varphi(\cdot), S, E];\ [f, A])$
[19]	$([X, \nabla f, 0];\ [\nabla_x\varphi(\cdot, x^k), S, EA];\ [f, E])$
[22]	$([X, \nabla f, 0];\ [\nabla_x\varphi(\cdot, x^k), M, E];\ [f, EA])$
[23]	$([X, \nabla f, 0];\ [\nabla_x\varphi(\cdot, x^k), M, A];\ [f, E])$
[25]	$([X, \nabla f, 0];\ [\nabla_x\varphi(\cdot, x^k), M, A];\ [f, A])$
[24]	$([X, \nabla f, 0];\ [\nabla\varphi^k, U, E];\ [f, EAR])$
[32]	$([X, \nabla f, 0];\ [\Phi(\cdot, x^k), SU, E];\ [f, A])$
[25]	$([X, \nabla f, 0];\ [\Phi(\cdot, x^k), S, E];\ [f, A])$
[25]	$([X, \nabla f, 0];\ [\Phi^k, U, E];\ [f, EAR])$
[22]	$([X, \nabla f, u];\ [\nabla\varphi(\cdot, x^k), M, E];\ [f+u, E])$
[25]	$([X, \nabla f, u];\ [\nabla\varphi(\cdot, x^k), M, A];\ [f+u, E])$
[22]	$([X, \nabla f, u];\ [\nabla\varphi(\cdot, x^k), S, E];\ [f+u, A])$
[2, 3]	$([X, \nabla f, u];\ [\nabla\varphi^k, U, E];\ [f+u, R])$
[24]	$([X, \nabla f, u];\ [\nabla\varphi^k, U, E];\ [f+u, EAR])$
[25]	$([X, \nabla f, u];\ [\Phi^k, U, E];\ [f+u, EAR])$
[6]	$([X, F, 0];\ [\nabla\varphi(\cdot, x^k), S, E];\ [C, R])$
[20]	$([X, F, 0];\ [\Phi(\cdot), U, E];\ [C, R])$
[13]	$([X, F, 0];\ [\nabla\varphi(\cdot), S, E];\ [\psi, E])$
[13]	$([X, F, 0];\ [\nabla\varphi(\cdot), U, E];\ [\psi, A])$
[13]	$([X, F, 0];\ [\nabla\varphi(\cdot), U, A];\ [\psi, E])$
[24]	$([X, F, 0];\ [\nabla\varphi(\cdot), U, E];\ [\psi, EAR])$
[33]	$([X, F, 0];\ [\nabla\varphi(\cdot, x^k), U, E];\ [\psi, EA])$
[33]	$([X, F, 0];\ [\nabla\varphi(\cdot, x^k), U, A];\ [\psi, E])$
[35]	$([X, F, 0];\ [(1/\gamma_k)\nabla\varphi(\cdot, x^k), S, E];\ [\psi^k, A])$
[36]	$([X, F, 0];\ [\Phi(\cdot, x^k), U, E];\ [\psi, A])$
[17]	$([X, F, 0];\ [(1/\gamma_k)\Phi(\cdot, x^k), U, E];\ [\psi^k, A])$
[4]	$([X, F, u];\ [\nabla\varphi(\cdot), U, E];\ [C, R])$
[34]	$([X, F, u];\ [\nabla\varphi(\cdot), U, EA];\ [C, R])$
[27]	$([X, F, u];\ [\nabla\varphi(\cdot), SU, E];\ [\psi, E])$

Table 1: Algorithm frameworks for the solution of [VIP]

Algorithm	Description	Reference
Steepest descent	$([\Re^n, \nabla f, 0];\ [\cdot, U, EA];\ [f, EAR])$	[16]
Quasi-Newton	$([\Re^n, \nabla f, 0];\ [B_k \cdot, U, EA];\ [f, EAR])$	[16]
Frank–Wolfe	$([X, \nabla f, 0];\ [0, EA];\ [f, M, EA])$	[30]
Gradient projection	$([X, \nabla f, 0];\ [\gamma_k \cdot, U, EA];\ [f, EAR])$	[8, 15]
Constrained Newton	$([X, \nabla f, 0];\ [\nabla^2 f(x^k)\cdot, MU, EA];\ [f, EAR])$	[30]
Proximal point	$([X, \nabla f, u];\ [\nabla f + \gamma_k \cdot, U, EA];\ [f + u, EAR])$	[31]
Linearization methods	$([X, F, 0];\ [B_k \cdot, U, E];\ [C, R])$	[21]
Jacobi	$([\prod_{i \in \mathcal{C}} X_i, F, 0];\ [(F_i(x^k_{i_-}, \cdot, x^k_{i_+}))_i, S, E];\ [C, R])$	[21]

Table 2: Some instances