A Structure-Preserving Pivotal Method for Affine Variational Inequalities

Youngdae Kim · Olivier Huber · Michael C. Ferris

20 janvier 2017
Variational Inequalities: $VI(C, F)$

$0 \in F(z) + \mathcal{N}_C(z)$
Variational Inequalities: $VI(C, F)$

$$0 \in F(z) + \mathcal{N}_C(z)$$

VI generalizes many problem classes
Affine Variational Inequalities: $AVI(C, q, M)$

\[ 0 \in Mz + q + N_C(z) \]
Affine Variational Inequalities: \( AVI(C, q, M) \)

\[ 0 \in Mz + q + \mathcal{N}_C(z) \]

AVI subsumes systems of linear equations, linear complementarity problems and optimality conditions for quadratic programs.
C is polyhedral $\mathcal{C} = \{ z \in \mathbb{R}^n | Az \geq b \}$
\[ C \text{ is polyhedral } C = \{ z \in \mathbb{R}^n | Az \geq b \} \]

- Affine Variational Inequalities: \( AVI(C, q, M) \)

\[ 0 \in Mz + q + \mathcal{N}_C(z) \]
- $\mathcal{C}$ is polyhedral $\mathcal{C} = \{z \in \mathbb{R}^n | Az \geq b\}$
- Affine Variational Inequalities: $AVI(\mathcal{C}, q, M)$

$$0 \in Mz + q + \mathcal{N}_\mathcal{C}(z)$$

$$\langle Mz + q, y - z \rangle \geq 0, \forall y \in \mathcal{C}$$
Introduction

- $\mathcal{C}$ is polyhedral $\mathcal{C} = \{z \in \mathbb{R}^n | A z \geq b\}$
- Affine Variational Inequalities: $AVI(\mathcal{C}, q, M)$

$$0 \in M z + q + \mathcal{N}_\mathcal{C}(z)$$

$$\langle M z + q, y - z \rangle \geq 0, \forall y \in \mathcal{C}$$

- Normal Map

$$x^* \text{ solves } 0 = M(\pi_\mathcal{C}(x)) + q + x - \pi_\mathcal{C}(x)$$
Normal manifold = \{ F_i + N_{F_i} \}

(Relative) interiors of faces $F_i$ form partition of $C$
Normal Manifold: normal mapping

\[ C = \{ z | Bz \geq b \}, \ N_C(z) = \{ B'v | v \leq 0, v_{I(z)} = 0 \} \]

\[ M \pi_C(x) + x - \pi_C(x) \]

\[ Mz + B'v \]

\[ z \in F_i \]

\[ v \leq 0, v_{I(z)} = 0 \]
Normal Manifold: Piecewise linear mapping

\[ C = \{ z | Bz \geq b \}, \quad F(z) = Mz + q \]
Path solver algorithm

Path$^1$: an implementation of a stabilized Newton method for the solution of the Mixed Complementarity Problem (MCP).

Path solver algorithm

Path\(^1\) : an implementation of a stabilized Newton method for the solution of the Mixed Complementarity Problem (MCP).

- **Start** in cell that has interior: ray start from an extreme point;

---

Path solver algorithm

Path\(^1\) : an implementation of a stabilized Newton method for the solution of the Mixed Complementarity Problem (MCP).

- **Start** in cell that has interior: ray start from an extreme point;
- **Move** towards a zero of affine map in cell: simplified according to the active constraints;

---

**Path solver algorithm**

**Path**¹: an implementation of a stabilized Newton method for the solution of the Mixed Complementarity Problem (MCP).

- **Start** in cell that has interior: ray start from an extreme point;
- **Move** towards a zero of affine map in cell: simplified according to the active constraints;
- **Pivot**: update direction when hit boundary;

---

Path solver algorithm

Path\(^1\): an implementation of a stabilized Newton method for the solution of the Mixed Complementarity Problem (MCP).

- **Start** in cell that has interior: ray start from an extreme point;
- **Move** towards a zero of affine map in cell: simplified according to the active constraints;
- **Pivot**: update direction when hit boundary;
- **Terminate**: a zero is found or secondary ray

---

**Path** solves MCP, a VI problem with box constraints:

\[ x^* \text{ solves } VI(F, C) \]
\[ \iff 0 \in F(x^*) + \mathcal{N}_C(x^*) \]
\[ \iff 0 = F(x^*) + \nabla g(x^*) \lambda, \]
\[ 0 \leq -g(x^*) \perp \lambda \geq 0 \]
\[ \iff z = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \quad G(z) = \begin{bmatrix} F(x) + \nabla g(x) \lambda \\ -g(x) \end{bmatrix}, \]
\[ z^* \text{ solves } VI(G, \mathbb{R}^n \times \mathbb{R}_+^m) \]
**PATH** solves MCP, a VI problem with box constraints:

\[
x^* \text{ solves } \text{VI}(F, C) \implies 0 \in F(x^*) + \mathcal{N}_C(x^*) \\
\iff 0 = F(x^*) + \nabla g(x^*)\lambda, \\
\quad 0 \leq -g(x^*) \perp \lambda \geq 0 \\
\iff z = \begin{bmatrix} x \\ \lambda \end{bmatrix}, \quad G(z) = \begin{bmatrix} F(x) + \nabla g(x)\lambda \\ -g(x) \end{bmatrix}, \\
z^* \text{ solves } \text{VI}(G, \mathbb{R}^n \times \mathbb{R}_+^m)
\]

The MCP transformation of an AVI is sometimes less desirable especially when the underlying polyhedral set of the original AVI is bounded.
Illustration of the LCP reformulation drawbacks, for an $M_{n \times n}$ symmetrical matrix with some negative eigenvalues and compact set $C$.

<table>
<thead>
<tr>
<th>$(m, n)$</th>
<th>PathAVI</th>
<th>PATH</th>
<th>% negative eigenvalues</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>status</td>
<td># iterations</td>
<td>status</td>
</tr>
<tr>
<td>(180,60)</td>
<td>S</td>
<td>55</td>
<td>S</td>
</tr>
<tr>
<td>(180,60)</td>
<td>S</td>
<td>45</td>
<td>S</td>
</tr>
<tr>
<td>(180,60)</td>
<td>S</td>
<td>2</td>
<td>F</td>
</tr>
<tr>
<td>(180,60)</td>
<td>S</td>
<td>1</td>
<td>F</td>
</tr>
<tr>
<td>(360,120)</td>
<td>S</td>
<td>124</td>
<td>S</td>
</tr>
<tr>
<td>(360,120)</td>
<td>S</td>
<td>55</td>
<td>S</td>
</tr>
<tr>
<td>(360,120)</td>
<td>S</td>
<td>2</td>
<td>F</td>
</tr>
<tr>
<td>(360,120)</td>
<td>S</td>
<td>1</td>
<td>F</td>
</tr>
</tbody>
</table>
Because of the MCP reformulation of the AVI, PATH does not use the fact that the constraint set is compact. It loses in robustness and efficiency.

Table 3: Performance of PATHAVI and PATH over compact sets

<table>
<thead>
<tr>
<th>Name</th>
<th>(#constrs,#vars)</th>
<th>(nnz(A),nnz(M))</th>
<th>Number of iterations</th>
<th>Elapsed time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>PATHAVI</td>
<td>PATH</td>
</tr>
<tr>
<td>CVXQP1_M</td>
<td>(500, 1000)</td>
<td>(2495, 999)</td>
<td>3119</td>
<td>fail</td>
</tr>
<tr>
<td>CVXQP2_M</td>
<td>(250, 1000)</td>
<td>(1746, 999)</td>
<td>33835</td>
<td>fail</td>
</tr>
<tr>
<td>CVXQP3_M</td>
<td>(750, 1000)</td>
<td>(3244, 999)</td>
<td>360</td>
<td>3603</td>
</tr>
<tr>
<td>CONT-050</td>
<td>(2401, 2597)</td>
<td>(14597, 6407)</td>
<td>11</td>
<td>382</td>
</tr>
<tr>
<td>CONT-100</td>
<td>(9801,10197)</td>
<td>(59197,98875)</td>
<td>3</td>
<td>fail</td>
</tr>
</tbody>
</table>
Theoretical results

- Feasibility: Under certain conditions over $M$ and $C$, it has been shown that the feasibility (or not) may be proven;

- Starting phase: to perform a ray start in the case where there is no extreme point, PathAVI finds an *implicit extreme point* which generalizes the notion of an extreme point when the underlying feasible region contains lines.
Theoretical results

- Feasibility: Under certain conditions over M and C, it has been shown that the feasibility (or not) may be proven;
- Starting phase: to perform a ray start in the case where there is no extreme point, PathAVI finds an *implicit extreme point* which generalizes the notion of an extreme point when the underlying feasible region contains lines.

**Worst-case performance comparison** - PathAVI will take fewer iterations than PATH, which solves the MCP reformulation:

- In the worst case both may visit every cell in the manifold;
- Same formulation but they look at different feasible regions: MCP deals with a larger normal manifold.
Various Friction Problems expressed as AVI, solved by PATH and PATHAVI

Fig. 3: Time comparison between PATH and PATHAVI
Results: PathAVI reduced vs. original

The problem: $C$ contains lines. Possibility of not finding an extreme point

---

The problem: $C$ contains lines. Possibility of not finding an extreme point

- **PathAVI** (reduced) reduces the problem to $AVI(\tilde{M}, \tilde{q}, \tilde{C})$ removing the lines in $C$

---

Results: PathAVI reduced vs. original

The problem: \( C \) contains lines. Possibility of not finding an extreme point

- **PathAVI** (reduced) reduces the problem to \( AVI(\tilde{M}, \tilde{q}, \tilde{C}) \)
  - removing the lines in \( C \)

- **PathAVI** (original) solves the original problem \( AVI(M, q, C) \)
  - by finding implicit extreme points to allow ray starts

---

Results: PathAVI reduced vs. original

The problem: $C$ contains lines. Possibility of not finding an extreme point

- PathAVI (reduced) reduces the problem to $AVI(\tilde{M}, \tilde{q}, \tilde{C})$ removing the lines in $C$

- PathAVI (original) solves the original problem $AVI(M, q, C)$ by finding implicit extreme points to allow ray starts

If the original AVI is sparse, there is no guarantee that the resulting reduced AVI would enjoy the same property. PathAVI (original) takes advantage of a sparse structure, whereas other methods\(^2\) often needs to perform dense linear algebra computations.

---

Application to the Friction Problem

Fig. 1: Nonzero patterns of the matrices $M$ (size: $1452 \times 1452$, nnz: 11330), $H$ (size: $1452 \times 363$, nnz: 1747) and $W := H^T M^{-1} H$ (size: $363 \times 363$, nnz: 56770).
Results: PATHAVI reduced vs. original

Application to the Friction Problem

![Histogram showing the time ratio distribution for PATHAVI (original) vs. PATHAVI (reduced).]
Discussion

Pros:
- It has strong theoretical guarantee to find solution, or reject feasibility

Cons:
- In the worst case, the algorithm can end up visiting every cells (up to $2^{(n-1)}$) and perform dense algebraic computation at every step.
Discussion

Pros:

- It has strong theoretical guarantee to find solution, or reject feasibility
- It can solve efficiently QP with bounded constraints

$$\min_z \frac{1}{2} z^T Mz - z^T q$$

s.t. $z \in C := \{z | Az \geq b\}$, bounded
Pros:

▶ It has strong theoretical guarantee to find solution, or reject feasibility
▶ It can solve efficiently QP with bounded constraints

$$\min_z \frac{1}{2} z^T M z - z^T q$$

s.t. $z \in C := \{z | A z \geq b\}$, bounded

Cons:

▶ In the worst case, the algorithm can end up visiting every cells (up to $2^{(n-1)}$) and perform dense algebraic computation at every step.
PathAVI a pivotal method for solving Affine Variable Inequalities

- It can preserve the structure of the problem allowing to run on large sparse datasets
- Many applications such as for example Friction Problems or bounded quadratic problems.