PATH VI: a pathsearch method for variational inequalities

Michael C. Ferris
(Joint with Youngdae Kim and Todd Munson)

University of Wisconsin, Madison
Funded by DOE-MACS Grant with Argonne National Laboratory

Computational Contact Mechanics: Advances and Frontiers in Modeling Contact February 17, 2014
VI: $-F(z) \in N_C(z)$

Many applications where $F$ is not the derivative of some $f$
Variational Inequality Formulation

- $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$
- Ideally: $C \subseteq \mathbb{R}^n$ – constraint set
- Often: $C \subseteq \mathbb{R}^n$ – simple bounds

$$0 \in F(z) + N_C(z)$$

- VI generalizes many optimization problems: LP, MCP, and LCP
  - For Nonlinear Equations: $F(z) = 0$ set $C \equiv \mathbb{R}^n$
  - For NCP: $0 \leq F(z), z \geq 0$ and $z^T F(z) = 0$ set $C \equiv \mathbb{R}_+^n$
  - For LCP, set $F(z) = Mz + q$ and $C \equiv \mathbb{R}_+^n$.
  - For MCP (rectangular VI), set $C \equiv [l, u]^n$.
  - Example: convex optimization first-order optimality condition:

$$\min_{z \in C} f(z) \iff -\nabla f(z) \in N_C(z) \iff 0 \in \nabla f(z) + N_C(z)$$

- For LP, set $F(z) \equiv \nabla f(z) = p$ and $C = \{z \mid Az = a, Hz \leq h\}$. 
AVI over polyhedral convex set

An affine function

\[ F : \mathbb{R}^n \to \mathbb{R}^n, \; F(z) = Mz + q, \; M \in \mathbb{R}^{n \times n}, \; q \in \mathbb{R}^n \]

A polyhedral convex set

\[ C = \{ z \in \mathbb{R}^n \mid Az(\geq, =, \leq)a, \; l \leq z \leq u \}, \; A \in \mathbb{R}^{m \times n} \]

Find a point \( z^* \in C \) satisfying

\[ \langle F(z^*), y - z^* \rangle \geq 0, \; \forall y \in C \]

(\( \Leftrightarrow \)) \[ \langle -F(z^*), y - z^* \rangle \leq 0, \; \forall y \in C \]

(\( \Leftrightarrow \)) \[ -F(z^*) \in N_C(z^*) \]

where

\[ N_C(z^*) = \{ v \mid \langle v, y - z^* \rangle \leq 0, \forall y \in C \} \]
Variational inequalities (current state)

- Find \( z \in C \) such that
  \[
  0 \in F(z) + N_C(z)
  \]

- model vi / F, g ;
  empinfo: vi F z g

- Convert problem into complementarity problem by introducing multipliers on representation of e.g. \( C = \{ z \in [l, u] : g(z) \leq 0 \} \)
  \[
  \begin{bmatrix}
  F(z) - \nabla g(z) \lambda \\
  g(z)
  \end{bmatrix} + N_{[l,u] \times \mathbb{R}^m}
  \]

- \( C \) polyhedral (e.g. \( C = \{ z \in [l, u] : Az \leq a \} \) and \( F(z) = Mz + q \)
  \[
  \begin{bmatrix}
  M & -A^T \\
  A & 0
  \end{bmatrix}
  \begin{bmatrix}
  z \\
  \lambda
  \end{bmatrix}
  +
  \begin{bmatrix}
  q \\
  -a
  \end{bmatrix}
  + N_{[l,u] \times \mathbb{R}^m}
  \]
Theorem

Suppose \( C \) is a polyhedral convex set and \( M \) is an \( L \)-matrix with respect to \( \text{rec}C \) which is invertible on the lineality space of \( C \). Then exactly one of the following occurs:

- \( \text{PATHAVI} \) solves (AVI)
- the following system has no solution

\[
Mz + q \in (\text{rec}C)^D, \quad z \in C.
\] (1)

Corollary

If \( M \) is copositive–plus with respect to \( \text{rec}C \), then exactly one of the following occurs:

- \( \text{PATHAVI} \) solves (AVI)
- (1) has no solution

Note also that if \( C \) is compact, then any matrix \( M \) is an \( L \)-matrix with respect to \( \text{rec}C \). So always solved.
Experimental results: AVI vs MCP

PATH is a solver for MCP (mixed complementarity problem).

- Run PathAVI over AVI formulation.
- Run PATH over AVI in MCP form (poorer theory as $\text{rec}C$ larger).
- Data generation
  - $M$ is an $n \times n$ symmetric positive definite/indefinite matrix.
  - $A$ has $m$ randomly generated bounded inequality constraints.

<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>
Complementarity Problems via Graphs

\[ \mathcal{T} = \mathcal{N}_{\mathbb{R}^+} = (\mathbb{R}_+ \times \{0\}) \cup (\{0\} \times \mathbb{R}_-) \]

\[ -y \in \mathcal{T}(\lambda) \iff (\lambda, -y) \in \mathcal{T} \iff 0 \leq \lambda \perp y \geq 0 \]

By approximating (smoothing) graph can generate interior point algorithms for example \( y\lambda = \epsilon, y, \lambda > 0 \)

\[ -F(z) \in \mathcal{N}_{\mathbb{R}^+}^{n}(z) \iff (z, -F(z)) \in \mathcal{T}^n \iff 0 \leq z \perp F(z) \geq 0 \]
Complementarity Systems (DVI)

\[ \frac{dx}{dt}(t) = f(x(t), \lambda(t)) \]

\[ y(t) = h(x(t), \lambda(t)) \]

\[ 0 \leq y(t) \perp \lambda(t) \geq 0 \]
Complementarity Systems (DVI)

\[
\frac{dx}{dt}(t) = f(x(t), \lambda(t))
\]

\[y(t) = h(x(t), \lambda(t))\]

\[0 \leq y(t) \perp \lambda(t) \geq 0\]
Complementarity Systems (DVI)

\[
\frac{dx}{dt}(t) = f(x(t), \lambda(t))
\]

\[
y(t) = h(x(t), \lambda(t))
\]

\[
(\lambda(t), -y(t)) \in \mathcal{T}
\]
Operators and Graphs \((\mathcal{C} = [-1, 1], \mathcal{T} = N_{\mathcal{C}})\)

\[
z_i = -1, -F_i(z) \leq 0 \text{ or } z_i \in (-1, 1), -F_i(z) = 0 \text{ or } z_i = 1, -F_i(z) \geq 0
\]

\(P_\mathcal{T}(y)\) is the projection of \(y\) onto \([-1, 1]\)
Generalized Equations

- Suppose $\mathcal{T}$ is a maximal monotone operator

$$0 \in F(z) + \mathcal{T}(z) \quad (GE)$$

- Define $P_\mathcal{T} = (I + \mathcal{T})^{-1}$

- If $\mathcal{T}$ is polyhedral (graph of $\mathcal{T}$ is a finite union of convex polyhedral sets) then $P_\mathcal{T}$ is piecewise affine (continuous, single-valued, non-expansive)

$$0 \in F(z) + \mathcal{T}(z) \iff z \in F(z) + I(z) + \mathcal{T}(z)$$

$$\iff z - F(z) \in (I + \mathcal{T})(z) \iff P_\mathcal{T}(z - F(z)) = z$$

Use in fixed point iterations (cf projected gradient methods)
Normal Map

- Suppose $\mathcal{T}$ is a maximal monotone operator
  \[ 0 \in F(z) + \mathcal{T}(z) \quad (GE) \]

- Define $P_\mathcal{T} = (I + \mathcal{T})^{-1}$

\[ 0 \in F(z) + \mathcal{T}(z) \iff z \in F(z) + I(z) + \mathcal{T}(z) \]
\[ \iff z - F(z) = x \text{ and } x \in (I + \mathcal{T})(z) \]
\[ \iff z - F(z) = x \text{ and } P_\mathcal{T}(x) = z \]
\[ \iff P_\mathcal{T}(x) - F(P_\mathcal{T}(x)) = x \]
\[ \iff 0 = F(P_\mathcal{T}(x)) + x - P_\mathcal{T}(x) \]

This is the so-called Normal Map Equation
Key idea of algorithm $\mathcal{T} = \mathcal{N}_C$

Homotopy: Easy solution for $\mu$ large, drive $\mu \to 0$.

$$\mu r = F(\pi_C(x(\mu))) + x(\mu) - \pi_C(x(\mu))$$

Define $z(\mu) = \pi_C(x(\mu))$, then

$$\mu r = F(z(\mu)) + x(\mu) - z(\mu)$$

$$x - z \in N_C(z)$$

$$N_C(z) = \{-A^T u - w + v\}$$

such that

$$Az(\geq, =, \leq)a \perp u(\geq, \text{free}, \leq)0$$

$$0 \leq w \perp z - l \geq 0$$

$$0 \leq v \perp u - z \geq 0$$
Ray start and complementary pivoting

Solve the normal map by

1. Computing an extreme point \( z_e \in C \) by solving Phase I.
2. Introducing a ray with a covering vector \( r \) in the interior of the normal cone at \( z_e \).
3. Setting up an initial basis for complementary pivoting using the result of Phase I.
4. Doing complementary pivoting until the multiplier on \( r \) becomes zero.

\[
-(Mz + q) + \mu r = -A^T u - w + v \\
A z(\geq, =, \leq) a \perp u(\geq, \text{free}, \leq) 0 \\
0 \leq w \perp z - l \geq 0 \\
0 \leq v \perp u - z \geq 0 \\
\mu \geq 0
\]
Example (complementary pivoting)

\[ M = \begin{bmatrix} 2 & -3 \\ -2 & 2 \end{bmatrix} \quad q = \begin{bmatrix} -2 \\ -5 \end{bmatrix} \]

\[ r = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \]

\[ x_0 = (-3, 0), z_0 = (0, 0) \]

\[ r = (-1, -1) \]

\[ N_C(z_0) \]

\[ x_1 = z_1 = (0, 0.75) \]

\[ x_2 \]

\[ z_2 = (3.5, 5.125) \]

\[ x_3 = (5, 9), z_3 = (4, 5) \]

\[ 5r \]

\[ C \]

\[ x_3 \]

\[ x_2 \]

\[ z_2 \]

\[ N_C(z_0) \]

\[ r = (-1, -1) \]
Implementation

1. Solve Phase I over $C$ using CPLEX.

$$\begin{align*}
\text{minimize} & \quad 0^T z \\
\text{subject to} & \quad Az = a \\
& \quad l \leq z \leq u
\end{align*}$$

- We have included slack and artificial variables.
- Thus, rank $A = m$.

2. Do complementary pivoting (Lemke’s method) until a feasible solution or a secondary ray is found.
Large scale implementation: Computing an extreme point

No extreme point exists when \( C \) has a non-zero lineality space

\[
\text{lin} C = \ker \begin{bmatrix} A \\ H \end{bmatrix} \neq \{0\}
\]

(\( H \) encodes bounds.) In that case, we compute a boundary point of \( C \).

- Computing a boundary point of \( C \)
  - Zero out \( \text{lin} C \) and compute an extreme point over reduced space.

\[
\text{lin} C = \quad \text{lin} C' = \{0\}
\]
Solving Phase I

If feasible region of $\mathcal{C}$ is not empty, then CPLEX comes with a basis triple $(B, N_l, N_u)$ with $B = A_B$ nonsingular such that

- $B = (B_1, \ldots, B_m) \subseteq \{1, \ldots, n\}$: indices of basic variables
- $N = \{1, \ldots, n\} \setminus B$: indices of nonbasic variables
- $N_l \cap N_u = \emptyset$, $N_l \cup N_u = \{j \notin B : x_j \text{ neither fixed nor free}\}$, $l_j > -\infty$ for $j \in N_l$ and $u_j < +\infty$ for $j \in N_u$
- $N_{fr} = \{j \in N : z_j \text{ free}\}$ and $N_{fx} = \{j \in N : z_j \text{ fixed}\}$.
- Note that $z_{N_l} = l_{N_l}$, $z_{N_u} = u_{N_u}$, $z_{N_{fr}} = 0$, $z_{N_{fx}} = l_{N_{fx}} = u_{N_{fx}}$, and $z_B = B^{-1}(b - A_N z_N)$. 
Phase I result interpretation (when \( \exists \) an extreme point)

If \( N_{fr} = \emptyset \), then \( \text{lin } C = \emptyset \) and Phase I gives us an extreme point.

- \( z \in C \) is an extreme point if \( z = \alpha \bar{z} + (1 - \alpha)\hat{z} \) for \( 0 < \alpha < 1 \) and \( \bar{z}, \hat{z} \in C \) implies that \( z = \bar{z} = \hat{z} \).

- \( z \in C \) is a BFS if \( \{A_j : l_j < z_j < u_j\} \) are linearly independent.

- \( z \in C \) is a BFS if and only if it is an extreme point.

- \( N_{fr} = \emptyset \) implies \( z \) is a BFS, hence an extreme point of \( C \).

- Existence of an extreme point implies that \( \text{lin } C = \emptyset \).
Phase I result interpretation (when \( \exists \) extreme points)

If \( N_{fr} \neq \emptyset \), then \( \text{lin } C \neq \emptyset \) and Phase I gives us a boundary point.

- Define \( z = (\bar{z}, \hat{z}) \) where \( \hat{z} = z_{N_{fr}} \). Fix \( \hat{z} = 0 \).
- Then we have a solution to the following Phase I.

\[
\begin{align*}
\text{minimize} \quad & 0^T z \\
\text{subject to} \quad & Az = a \\
& l \leq z \leq u \\
& \hat{z} = 0
\end{align*}
\]

- \( \bar{z} \) is a BFS in the reduced space of \( C \) where \( \hat{z} = 0 \), thus an extreme point in that space.
Initial basis setup for starting Lemke’s method

From Phase I, we have a nonsingular $B$

$$B_{\text{Phase I}} = \begin{bmatrix} A_{AB} & 0 \\ A_{IB} & -I_I \end{bmatrix}$$

where

- $\mathcal{A}$: the set of indices of active constraints
- $\mathcal{I}$: the set of indices of inactive constraints

So that $A_{AB}$ is nonsingular.
Initial basis setup for starting Lemke’s method

We need to solve a system of equations using complementary pivoting.

\[(Mz + q) - \mu r = A^T u + w - v\]

\[Az - s = a\]

\[0 \leq s \perp u \geq 0\]

\[0 \leq w \perp z - l \geq 0\]

\[0 \leq v \perp u - z \geq 0\]

\[r \in N_C(z_{Phase I})\]

If \(N_{fr} = \emptyset\),

\[B_{Lemke} = \begin{bmatrix}
M_{BB} & -A_{AB}^T & 0 & 0 & 0 \\
M_{LB} & -A_{AL}^T & -I_L & 0 & 0 \\
M_{UB} & -A_{AU}^T & 0 & I_U & 0 \\
A_{AB} & 0 & 0 & 0 & 0 \\
A_{\bar{A}B} & 0 & 0 & 0 & -I_{\bar{A}}
\end{bmatrix}, \quad \text{Bvars} = \begin{bmatrix}
z_B \\
u_A \\
w_L \\
v_U \\
s_{\bar{A}}
\end{bmatrix}\]
Initial basis setup for starting Lemke’s method

If $N_{fr} \neq \emptyset$,

$$
\mathbf{B}_{\text{Lemke}} = \begin{bmatrix}
    M_{BB} & M_{BF} & -A_{AB}^T & 0 & 0 & 0 \\
    M_{LB} & M_{LF} & -A_{AL}^T & -I_L & 0 & 0 \\
    M_{UB} & M_{UF} & -A_{AU}^T & 0 & I_U & 0 \\
    A_{AB} & A_{AF} & 0 & 0 & 0 & 0 \\
    A_{\bar{A}B} & A_{\bar{A}F} & 0 & 0 & 0 & -I_{\bar{A}} \\
\end{bmatrix}, \quad \mathbf{Bvars} = \begin{bmatrix}
    z_B \\
    z_F \\
    u_A \\
    w_L \\
    v_U \\
    s_{\bar{A}} \\
\end{bmatrix}
$$

If $M$ is invertible in the lineality space of $\mathcal{C}$, then the above matrix is invertible.
Initial pivoting

Solve

\[
\begin{bmatrix}
M_{BB} & -A^T_{AB} & 0 & 0 & 0 \\
M_{LB} & -A^T_{AL} & -I_L & 0 & 0 \\
M_{UB} & -A^T_{AU} & 0 & I_U & 0 \\
A_{AB} & 0 & 0 & 0 & 0 \\
A_{AB} & 0 & 0 & 0 & -I_{\bar{A}}
\end{bmatrix}
\begin{bmatrix}
z_B \\
u_A \\
w_L \\
v_U \\
s_{\bar{A}}
\end{bmatrix}
= \begin{bmatrix}
-q_B - M_{BL} z_L - M_{BU} z_U \\
-q_L - M_{LL} z_L - M_{LU} z_U \\
-q_U - M_{UL} z_L - M_{UU} z_U \\
b_A - A_{AL} z_L - A_{AU} z_U \\
b_{\bar{A}} - A_{\bar{A}L} z_L - A_{\bar{A}U} z_U
\end{bmatrix}
\]

Note that $z_B$ and $s_{\bar{A}}$ are feasible due to Phase I.

If any of $u_A$, $w_L$, or $v_U$ is infeasible, then make $r$ basic by increasing $\mu$ so that all of them become feasible.

\[
\sum_{i \in A} \begin{bmatrix}
-A^T_{iB} \\
-A^T_{iL} \\
-A^T_{iU}
\end{bmatrix} + \sum_{i \in L} \begin{bmatrix}
0 \\
-I_i \\
0
\end{bmatrix} + \sum_{i \in U} \begin{bmatrix}
0 \\
0 \\
I_i
\end{bmatrix} \in N_C(z_{\text{Phase I}})
\]
Experimental results (LPs)

Some promising results:

<table>
<thead>
<tr>
<th>Data set</th>
<th># iterations (Lemke)</th>
<th>Total elapsed time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PathAVI</td>
<td>PATH</td>
</tr>
<tr>
<td>25fv47</td>
<td>3938</td>
<td>3202</td>
</tr>
<tr>
<td>bnl1</td>
<td>592</td>
<td>3230</td>
</tr>
<tr>
<td>pilotnov</td>
<td>3046</td>
<td>&gt; 10,000</td>
</tr>
<tr>
<td>scfxm3</td>
<td>988</td>
<td>4129</td>
</tr>
<tr>
<td>wood1p</td>
<td>336</td>
<td>1325</td>
</tr>
<tr>
<td>woodw</td>
<td>1292</td>
<td>9878</td>
</tr>
</tbody>
</table>

Table: Solving LP (linear programming) problems using PathAVI and PATH (netlib data sets)
Experimental results (symmetric psd QPs)

<table>
<thead>
<tr>
<th>Data set</th>
<th># iterations (Lemke)</th>
<th>Total elapsed time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PathAVI</td>
<td>PATH</td>
</tr>
<tr>
<td>cvxqp1_M</td>
<td>340</td>
<td>1063</td>
</tr>
<tr>
<td>dualc8</td>
<td>4</td>
<td>39</td>
</tr>
<tr>
<td>qscagr25</td>
<td>240</td>
<td>868</td>
</tr>
<tr>
<td>qscfxm3</td>
<td>1072</td>
<td>2021</td>
</tr>
<tr>
<td>qship12l</td>
<td>1399</td>
<td>3246</td>
</tr>
<tr>
<td>cont-101</td>
<td>99</td>
<td>750</td>
</tr>
</tbody>
</table>

Table: Solving QP (quadratic programming) problems using PathAVI and PATH, $Q$ is symmetric and PSD

QP problems were taken from “I. Maros, Cs. Meszaros: A Repository of Convex Quadratic Programming Problems, Optimization Methods and Software, 1999”
Experimental results (unsymmetric pd $M$)

<table>
<thead>
<tr>
<th>Data set</th>
<th># iterations (Lemke)</th>
<th>Total elapsed time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PathAVI</td>
<td>PATH</td>
</tr>
<tr>
<td>bnl1</td>
<td>657</td>
<td>&gt; 10,000</td>
</tr>
<tr>
<td>capri</td>
<td>296</td>
<td>571</td>
</tr>
<tr>
<td>fit1d</td>
<td>1346</td>
<td>1839</td>
</tr>
<tr>
<td>scsd8</td>
<td>1414</td>
<td>2155</td>
</tr>
<tr>
<td>scfxm3</td>
<td>823</td>
<td>2262</td>
</tr>
<tr>
<td>wood1p</td>
<td>413</td>
<td>915</td>
</tr>
</tbody>
</table>

Table: Solving AVI problems using PathAVI and PATH, $M$ is unsymmetric PD

- $M$ was randomly generated using MATLAB.
Conclusions

- Treat feasible set $\mathcal{C}$ and $\mathcal{N}_\mathcal{C}$ explicitly leads to stronger theory
- Ensure feasibility $\mathcal{C} \neq \emptyset$, and $F$ only evaluated over $\mathcal{C}$
- Works when $\nabla F$ is not symmetric
- Can implement theory in large scale setting and get robustness (avoid rank deficiency in initial basis, high accuracy)
- Faster
- Available (subroutine or within GAMS/EMP) - requires CPLEX
- Embed AVI solver in a Newton Method for VI
  - Preprocessing incorporated
  - Each Newton step solves an AVI
  - Hot start critical
  - Nonmonotone pathsearch, watchdogging (another talk)
Splitting Methods

- Suppose $\mathcal{T}$ is a maximal monotone operator
  \[ 0 \in F(z) + \mathcal{T}(z) \quad (GE) \]

- Can devise Newton methods (e.g. SQP) that treat $F$ via calculus and $\mathcal{T}$ via convex analysis

- Alternatively, can split $F(z) = A(z) + B(z)$ (and possibly $\mathcal{T}$ also) so we solve (GE) by solving a sequence of problems involving just
  \[ \mathcal{T}_1(z) = A(z) \text{ and } \mathcal{T}_2(z) = B(z) + \mathcal{T}(z) \]

  where each of these is “simpler”

- Forward-Backward splitting:
  \[ z^{k+1} = (I + c_k T_2)^{-1} (I - c_k T_1) z^k, \]
Normal manifold = \( \{ F_i + N_{F_i} \} \)

(Relative) interiors of faces \( F_i \)
form partition of \( C \)
\[ C = \{ z | Bz \geq b \}, \quad N_C(z) = \{ B'v | v \leq 0, v_{\mathcal{I}}(z) = 0 \} \]
$C = \{ z \mid Bz \geq b \}$, $N_C(z) = \{ B'v \mid v \leq 0, v_{\mathcal{I}(z)} = 0 \}$

$\begin{bmatrix} B'_1 & B'_2 \end{bmatrix}
\begin{align*}
Mz + B'v \\
z \in F_i \\
v \leq 0, v_{\mathcal{I}(z)} = 0
\end{align*}$
\[ C = \{ z | Bz \geq b \}, \quad F(z) = Mz + q \]
Cao/Ferris Path (Eaves)

- Start in cell that has interior (face is an extreme point)
- Move towards a zero of affine map in cell
- Update direction when hit boundary (pivot)
- Solves or determines infeasible if $M$ is copositive-plus on $\text{rec}(C)$
- Solves 2-person bimatrix games, 3-person games too, but these are nonlinear

But algorithm has exponential complexity (von Stengel et al)