TECNICAS DE DESCOMPOSICION-AGREGACION EN EL TRATAMIENTO DE LA INECUACION BILATERA DE ISAACS

Silvia C. Di Marco

Departamento de Matemática. Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario, Avda. Pellegrini 250, 2000 Rosario, Argentina

RESUMEN

Presentamos en este trabajo técnicas especiales para resolver numéricamente la inecuación bilátera de Isaacs asociada a juegos con stopping times. Introducimos dos algoritmos iterativos probando que convergen en un número finito de pasos. El cálculo numérico en sí es simplificado a través del uso de la estructura de la cadena de Markov asociada al problema.

ABSTRACT

We study in this paper some numerical techniques to solve the Isaacs's inequality associated to games with stopping times. We have developed two ad-hoc algorithms that converge in a finite number of steps. The task of numerical computation in itself is simplified through the use of the special structure of the Markov chain associated to the problem.

1. INTRODUCCION

Consideramos en este trabajo la resolución del sistema (1) de desigualdades biláteras resultantes de la discretización de la inecuación de Isaacs asociada a problemas de juegos diferenciales de suma nula con tiempos de detención (ver [4], [5], [6]).

$$(Mw)(j) = \max\Big(\min\big((Bw+f)(j),\ \Psi_2(j)\big),\Psi_1(j)\Big) \qquad (1)$$

Nuestro objetivo es diseñar un algoritmo rápido basado en el reconocimiento de la estructura de la cadena de Markov asociada al problema y en el uso de su descomposición jerárquica en componentes recurrentes y transitorias.

Diseñamos en particular, un algoritmo de agregación de los estados de la cadena que permite subdividir en una forma maximal los sistemas lineales (que siempre es necesario resolver en el algoritmo iterativo para la solución del problema original), de forma tal que no existe una forma más simple de solucionar los mencionados sistemas lineales.

La elección sucesiva de estrategias sub-óptimas que definen el algoritmo iterativo principal de resolución está basada en el uso del *principio de máximo discreto* (ver [1], [2]) verificado por el tipo de discretizaciones utilizadas para aproximar el problema continuo original. Se utiliza el mismo para realizar simplificaciones iterativas sobre la estructura jerárquica determinada anteriormente.

Para los algoritmos diseñados, se prueba finalmente que se obtiene la solución del sistema (1) al cabo de un número finito de resoluciones lineales de sistemas descompuestos en subsistemas minimales.

Se analizan la metodología resultante de partir de diferentes condiciones iniciales y las consecuencias de las mismas.

2. ELEMENTOS DEL PROBLEMA

$B \in \mathfrak{M}_{N \times N}$, tal que

$$
\begin{aligned}
&\text{i) } B(i,j) \geq 0 \qquad \forall\, i,j = 1,\ldots,N, \\
&\text{ii) } \sum_j B(i,j) \leq \gamma \quad \forall\, i,j = 1,\ldots,N, \quad \gamma < 1
\end{aligned}
\tag{2}
$$

Sean $\Psi_1(j)$, $\Psi_2(j)$, $j = 1,\ldots,N$, tal que

$$\text{iii) } \Psi_2(j) - \Psi_1(j) \geq \alpha > 0 \qquad \forall\, j = 1,\ldots,N, \tag{3}$$

Sea $f(j) \in \mathbb{R}$, $j = 1,\ldots,N$

Sea $M: \mathbb{R}^N \to \mathbb{R}^N$ un operador definido como en (1)

$$w \to Mw = P_{[\Psi_1,\Psi_2]}\left(Bw + f \right) \tag{4}$$

El problema consiste en encontrar el punto fijo del operador M, que existe por ser éste contractivo (ver [6]). Esto es:

$$\textit{Buscar } \hat{w} \;/\; \hat{w} = M\hat{w}$$

3. ALGORITMO ACELERADO 1

3.1 Fundamentación del algoritmo acelerado 1

De (4) se deduce que:

$$\hat{w}(i) = \begin{cases} \Psi_1(i) & \forall\, i \,/\, (B\hat{w} + f)(i) \leq \Psi_1(i) \\ \Psi_2(i) & \forall\, i \,/\, (B\hat{w} + f)(i) \geq \Psi_2(i) \\ (B\hat{w} + f)(i) & \forall\, i \,/\, \Psi_1(i) < (B\hat{w} + f)(i) < \Psi_2(i) \end{cases} \tag{5}$$

Llamaremos:

$$S_1 = \{ i \,/\, \hat{w}(i) = \Psi_1(i) \}$$
$$S_2 = \{ i \,/\, \hat{w}(i) = \Psi_2(i) \}$$
$$C = \{ i \,/\, \Psi_1(i) < \hat{w}(i) < \Psi_2(i) \}$$

Consideramos como condición inicial del algoritmo, el vector

$$w^0 = (I - B)^{-1} f \tag{6}$$

que siempre existe, por ser $I - B$ una matriz regular.

Notemos que w^0 sería la solución del problema en el caso que $\Psi_1 = -\infty$ y $\Psi_2 = +\infty$.

Sea

$$\hat{S}_2^0 = \left\{ j \,/\, w^0(j) - \Psi_2(j) = \left(\max_i \left(w^0(i) - \Psi_2(i) \right) \right)^+ \right\} \tag{7}$$

$$\hat{S}_1^0 = \left\{ j \,/\, w^0(j) - \Psi_1(j) = -\left(\min_i \left(w^0(i) - \Psi_1(i) \right) \right)^- \right\} \tag{8}$$

Lema 1.

$$\hat{S}_2^0 \subseteq S_2 \text{ y } \hat{S}_1^0 \subseteq S_1$$

Demostración: Sea $\hat{S}_2^0 \neq \emptyset$, llamemos $\Delta = \left(\max_i \left(w^0(i) - \Psi_2(i) \right) \right)^+$ y del mismo modo al vector de dimensión N cuyas componentes valen Δ.

$$w^0 - \Delta \leq \Psi_2 \qquad \forall\, j = 1,\ldots,N$$

$$w^0 - \Delta = \Psi_2 \qquad \forall\, i \in \hat{S}_2^0 \tag{9}$$

$$B(w^0 - \Delta) + f = w^0 - B\Delta \geq w^0 - \Delta$$

pues $B\Delta \leq \Delta$ por (2), luego

$$P_{[\Psi_1,\Psi_2]}\left(B(w^0 - \Delta) + f \right) \geq P_{[\Psi_1,\Psi_2]}\left(w^0 - \Delta \right) \geq w^0 - \Delta$$

$$M(w^{o} - \Delta) \geq (w^{o} - \Delta) \tag{10}$$

por monotonía del operador M. Por inducción, se obtiene de (10)

$$M^{m}(w^{o} - \Delta) \geq M(w^{o} - \Delta) \geq (w^{o} - \Delta) \tag{11}$$

pero

$$M^{m}(w^{o} - \Delta)(i) \rightarrow \bar{w}(i), \text{ cuando } m \rightarrow \infty,$$

luego

$$\bar{w}(i) \geq M^{m}(w^{o} - \Delta)(i) \geq M(w^{o} - \Delta)(i) \geq (w^{o} - \Delta)(i) = \Psi_2(i) \quad \forall\, i \in \hat{S}_2^0$$

lo que implica $i \in S_2$. Con una demostración análoga obtenemos $\hat{S}_1^0 \subseteq S_1$

□

De este lema deducimos que para $\hat{S}_1^0 \cup \hat{S}_2^0$ conocemos la solución, de modo que a dichos nodos se la asignaremos redefiniendo las filas de la matriz B y las componentes de f.

Definiendo inductivamente:

$$B^{0} = B,\ f^{0} = f \text{ y } \forall\, n \in \mathbb{N}_0$$

$$w^{n}(i) = (I - B^{n})^{-1} f^{n}(i) \tag{12}$$

$$\hat{S}_2^n = \left\{ j \,/\, w^{n}(j) - \Psi_2(j) = \left(\max_i (w^{n}(i) - \Psi_2(i)) \right)^{+} \right\} \tag{13}$$

$$\hat{S}_1^n = \left\{ j \,/\, w^{n}(j) - \Psi_1(j) = - \left(\min_i (w^{n}(i) - \Psi_1(i)) \right)^{-} \right\} \tag{14}$$

$$S_2^n = \bigcup_{m=0}^{n} \hat{S}_2^m \tag{15}$$

$$S_1^n = \bigcup_{m=0}^{n} \hat{S}_1^m \tag{16}$$

$$C^{n} = \complement (S_1^n \cup S_2^n) \tag{17}$$

$$B^{n+1}(i,j) = \begin{cases} 0 & \forall\, i \in \hat{S}_1^n \cup \hat{S}_2^n \\ B^{n}(i,j) & \forall\, i \in C^{n} \end{cases} \tag{18}$$

$$f^{n+1}(i) = \begin{cases} \Psi_1(i) & \forall\, i \in \hat{S}_1^n \\ \Psi_2(i) & \forall\, i \in \hat{S}_2^n \\ f^{n}(i) & \forall\, i \in C^{n} \end{cases} \tag{19}$$

$$(M_n w)(j)=\max\Big(\min\big((B^n w+f^n)(j),\ \Psi_2(j)\big),\Psi_1(j)\Big) \tag{20}$$

Es posible probar por inducción que

$$\bullet\ w^n(i) = \begin{cases} \Psi_1(i) & \forall\ i \in S_1^{n-1} \\ \Psi_2(i) & \forall\ i \in S_2^{n-1} \end{cases} \tag{21}$$

$$\bullet\ \begin{cases} S_2^n \subseteq S_2 & \forall\ n \in \mathbb{N} \quad (22) \\ S_1^n \subseteq S_1 & \forall\ n \in \mathbb{N} \quad (23) \end{cases}$$

$$\bullet\qquad \hat{w} = M_n\ \hat{w} \tag{24}$$

Lema 2. *Es válida la siguiente alternativa:*

$$\hat{S}_1^n \cup \hat{S}_2^n \neq \emptyset \quad o \quad w^n=\hat{w}$$

Demostración: Si $\hat{S}_1^n \cup \hat{S}_2^n = \emptyset$, ello implica que si $i \notin \left(S_1^{n-1} \cup S_2^{n-1}\right)$ se verifica que

$$w^n(i) = (I - B^n)^{-1} f^n\ (i) \ \in [\ \Psi_1(i),\Psi_2(i)\ ]$$

luego

$$M_n w^n(i) = P_{[\Psi_1,\Psi_2]}\left(\ B^n w^n + f^n\ \right)(i) = P_{[\Psi_1,\Psi_2]}\ (w^n)(i) = w^n(i) \tag{25}$$

de (24), (25) tenemos: $\qquad w^n = \hat{w}$

□

3.2 Presentación del algoritmo acelerado 1

ALGORITMO

PASO 0: $n=0$

PASO 1: Calcular w^n como en (12)
Construir $\hat{S}_1^n$ y $\hat{S}_2^n$ como en (13) y (14)

PASO 2: Si $\hat{S}_1^n \cup \hat{S}_2^n = \emptyset$, entonces parar.
Si no, construir $B^n(i,j)$ y $f^n(i)$ $\forall$ i,j como en (18) y (19)
hacer $n = n+1$
ir a PASO 1.

3.3 Convergercia del algoritmo

Teorema 1: ***El algoritmo finaliza en un número finito de iteraciones no mayor que N***

Demostración: Mientras $\hat{S}_1^n \cup \hat{S}_2^n \neq \emptyset$, se tiene que $card(C^n) < card(C^{n-1})$, por lo que se arriba a un indice $\hat{n}$ tal que $card(C^{\hat{n}}) = card(C^{\hat{n}-1})$, lo que implica $\hat{S}_1^n \cup \hat{S}_2^n = \emptyset$, que por el teorema de la alternativa indica que $w^{\hat{n}} = \bar{w}$.

□

4. ALGORITMO ACELERADO 2

4.1 Fundamentación del algoritmo acelerado 2

Consideraremos como condición inicial para el nuevo algoritmo, el vector $w^0 = \Psi_2$.
Sean

$$C^1 = \{ i : (B\Psi_2 + f)(i) < \Psi_2(i) \} \tag{26}$$

$$S_2^1 = \{ i : (B\Psi_2 + f)(i) \geq \Psi_2(i) \} \tag{27}$$

Lema 3.

$$C^1 \subseteq C \cup S_1 \quad \text{o sea,} \quad S_2 \subseteq S_2^1 \tag{28}$$

Demostración: Sea $i \in C^1$, de (3), (26) se tiene

$$M\,\Psi_2(i) = P_{[\Psi_1,\Psi_2]}\Big(B\Psi_2 + f \Big)(i) < P_{[\Psi_1,\Psi_2]}(\Psi_2)(i) = \Psi_2(i),$$

Mientras que, por ser Ψ_2 una supersolución vale que:

$$M\Psi_2(j) \leq \Psi_2(j), \ \forall j$$

Inductivamente vale:

$$M^n\Psi_2(i) \leq M^{n-1}\Psi_2(i) < \Psi_2(i), \ \forall \, n \in \mathbb{N}$$

Pero, $M^n\Psi_2(i) \to \bar{w}(i)$, cuando $n \to \infty$. Por lo tanto, $\bar{w}(i) < \Psi_2(i)$

□

Del lema anterior se deduce que el total de nodos se reparte en dos conjuntos disjuntos: C^1 y S_2^1. C^1 abarca nodos donde la solución no puede ser Ψ_2 y S_2^1 constituye una aproximación por exceso de S_2.
En S_2^1 asignamos transitoriamente el valor Ψ_2. Para C^1 calculamos la solución del sistema lineal w^n, definido como en (12), y construimos iterativamente $\hat{S}_1^n$, como en (14).

Cuando para algún $\hat{n}$ resulte $\hat{S}_1^{\hat{n}} = \emptyset$, se tiene que $w^{\hat{n}}(i) = \Psi_1(i)\ \forall\ i \in S_1^{\hat{n}}$; de aquí surge que para estos nodos ya se tiene la solución definitiva, por lo que en base a ella se computa nuevamente el vector $(Bw^{\hat{n}}+f)$ y a partir de él, el algoritmo determina cuales nuevos nodos abandonan S_2^1.

4.2 Presentación del algoritmo acelerado 2

ALGORITMO

PASO 0: $n=1$, $C^1 = \emptyset$, $S_2^1 = \{ 1, ..., N \}$, $B^1 = O_{NxN}$, $f^1 = \Psi_2$

$w^1 = \Psi_2$

PASO 1: Construir $\hat{C}^n = \{ i \in S_2^1 : (Bw^n + f)(i) < \Psi_2(i) \}$

Si $\hat{C}^n = \emptyset$, entonces parar.

Si no, hacer:

$$C^{n+1} = C^n \cup \hat{C}^n$$
$$S_2^{n+1} = S_2^n \setminus \hat{C}^n$$

$$B^{n+1}(i,j) = \begin{cases} B(i,j) & \forall\ i \in \hat{C}^n \\ B^n(i,j) & \forall\ i \notin \hat{C}^n \end{cases}$$

$$f^{n+1}(i) = \begin{cases} f(i) & \forall\ i \in \hat{C}^n \\ f^n(i) & \forall\ i \notin \hat{C}^n \end{cases}$$

$n = n+1$ e ir al PASO 2.

PASO 2: Calcular $w^n = (I - B^n)^{-1} f^n$

Construir $\hat{S}_1^n = \left\{ j \ / \ w^n(j) - \Psi_1(j) = -\left(\min_i (w^n(i) - \Psi_1(i)) \right)^- \right\}$

Si $\hat{S}_1^n = \emptyset$, ir a PASO 1.

Si $\hat{S}_1^n \neq \emptyset$, hacer:

$$C^{n+1} = C^n \setminus \hat{S}_1^n$$
$$S_1^{n+1} = S_1^n \cup \hat{S}_1^n$$

$$B^{n+1}(i,j) = \begin{cases} 0 & \forall\ i \in \hat{S}_1^n \\ B^n(i,j) & \forall\ i \in C^{n+1} \end{cases}$$

$$f^{n+1}(i) = \begin{cases} \Psi_1(i) & \forall\ i \in \hat{S}_1^n \\ f^n(i) & \forall\ i \in C^{n+1} \end{cases}$$

$n = n+1$ y reiniciar el PASO 2.

4.3.Convergencia del algoritmo

Teorema 2: *Si para algún $\hat{n} \in \mathbb{N}$, $\dot{C}^{\hat{n}} = \emptyset$, entonces $w^{\hat{n}} = \hat{w}$.*

Demostración: Si $\dot{C}^{\hat{n}} = \emptyset$, se tiene que

$$Mw^{\hat{n}}(i) = w^{\hat{n}}(i) = \Psi_2(i) \qquad \forall\, i \in S_2^{\hat{n}}$$

En el caso $\hat{n} > 1$, pueden existir $i \in S_1^{\hat{n}}$, para ellos sabemos por construcción de $\hat{S}_1^{n}$ que

$$\left(Bw^{n} + f \right)(i) \leq \Psi_1(i)$$

lo que implica $Mw^{\hat{n}}(i) = w^{\hat{n}}(i) = \Psi_1(i)$

También, para $i \in C^{\hat{n}} \setminus S_1^{\hat{n}}$ sabemos por construcción de $\hat{S}_1^{n}$ que

$$(Bw^{\hat{n}} + f)(i) = w^{\hat{n}}(i), \;\; y \quad \Psi_1(i) < (Bw^{\hat{n}} + f)(i) < \Psi_2(i)$$

lo que implica $Mw^{\hat{n}}(i) = w^{\hat{n}}(i)$

□

Teorema 3: *El algoritmo finaliza en un número finito de pasos.*

Demostración: Por lo visto en Teorema 1, el PASO 2 finaliza en un número de iteraciones no mayor que $card(C^{n})$.

En PASO 1, si $\dot{C}^{n} \neq \emptyset$, existe al menos un nodo que abandona S_2^{n}, luego el algoritmo entra a PASO 2, desde el PASO 1, un número de veces no mayor que card(S_2^1).

Definiendo $\mu =$ card(S_2^1), se tiene que el número de iteraciones es no mayor que:

$$\sum_{n=1}^{\mu} card(C^{n})$$

□

5. RESOLUCION SIMPLIFICADA DE LOS SISTEMAS LINEALES ASOCIADOS

Esta simplificación está basada en el análisis de la cadena de Markov (ver [7]), asociada al problema de la siguiente forma:

- La cadena tiene N nodos o estados "i"
- Las probabilidades de transición P(i,j) se definen por

$$P(i,j) = \delta_{i,j} \;\; si \;\; \sum_{j=1}^{N} B(i,j) = 0$$

$$P(i,j) = B(i,j) \,/ \sum_{j=1}^{N} B(i,j) \;\; si \;\; \sum_{j=1}^{N} B(i,j) > 0$$

Introducimos la siguiente relación de equivalencia en el conjunto de nodos:

$i \sim j$ si $i = j$ o $\exists$ una cadena de índices $i_0, \ldots, i_p$, tal que $i_0 = i$, $i_p = i$ y $\exists\, i_l$ tal que $i_l = j$.

Dado que $\sim$ es una relación de equivalencia, la cadena quedará dividida en $\tilde{N}$ clases disjuntas, denotadas con C^r, $r = 1, \ldots, \tilde{N}$; siendo $n^r = \mathrm{card}(C^r)$.

Es posible asociar al conjunto de clases C^r un grafo de estructura arborescente. Para ello introducimos en $\{ C^r : r = 1, \ldots, \tilde{N} \}$ un orden parcial que permite subdividir este conjunto en $\bar{\nu}$ clusters jerárquicos, designados con S^{ν}.

Especificamente definimos:

- $C^r \in S^1$ si no existe $\hat{r} \neq r$ tal que $P(i,j) > 0$, $i \in C^r$, $j \in C^{\hat{r}}$
- $C^r \in S^{\nu}$, $\nu > 1$, si existe $\hat{r} \neq r$ tal que
$$P(i,j) > 0,\ i \in C^r,\ j \in C^{\hat{r}},\ C^{\hat{r}} \in S^{\nu-1}$$
- $m^{\nu} = \mathrm{card}(S^{\nu})$

Denotaremos con $C^{r^{\nu}_q}$ al q-ésimo elemento del cluster S^{ν}.

Definiendo un reordenamiento de indices $I(\cdot)$: $N \rightarrow N$, tal que existe un reagrupamiento de los números de 1 a N, en $\tilde{N}$ clases contiguas $T_{\nu q}$, $\nu = 1, \bar{\nu}$, $q = 1, m^{\nu}$

- $\mathrm{card}(T_{\nu q}) = n^{r^{\nu}_q}$
- $\{ I(j) / j \in T_{\nu q} \} = \{ i / i \in C^{r^{\nu}_q} \}$
- $\forall\ q, \nu, \exists\ i_0(\nu,q) /\ T_{\nu q} = \{ i_0, i_0 + 1, \ldots, i_0 + n^{r^{\nu}_q} - 1 \}$
- $\nu > \hat{\nu} \Rightarrow i > j$, $\forall\ i \in T_{\nu q}$, $\forall\ j \in T_{\hat{\nu}\hat{q}}$ $\forall\ q = 1, m^{\nu}$, $\forall\ \hat{q} = 1, m^{\hat{\nu}}$

Bajo este reordenamiento, la correspondiente matriz asociada B tiene una estructura block-triangular inferior con lo cual el sistema $Bw + f$ puede ser reducido a resolver $\tilde{N}$ subsistemas de dimensión n^r.

La determinación de las clases C^r se realiza por un procedimiento de agregación iterativa(ver descripción detallada en [3]).

6. EJEMPLOS DE APLICACION

6.1 Ejemplo de aplicación del algoritmo 1

Para un ejemplo cuyos datos se generaron aleatoriamente, (N=13), se muestra en la siguiente tabla la evolución (por columnas) de los conjuntos de aproximación de S_1, S_2, C, respectivamente, dados por la convención:

$$0 \rightarrow C,\ 1 \rightarrow S_2,\ -1 \rightarrow S_1$$

TABLA 1

0	1	1	1	1	1	1
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	-1	-1
0	0	1	1	1	1	1
0	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	-1	-1	-1	-1
0	0	0	0	-1	-1	-1
0	0	0	0	0	0	0
0	0	0	0	0	1	1
0	0	0	1	1	1	1

6.2 Ejemplo de aplicación del algoritmo 2

Para un ejemplo cuyos datos se generaron aleatoriamente, $(N=13)$, se muestra en la siguiente tabla la evolución (por columnas) de los conjuntos de aproximación de S_1, S_2, C, respectivamente, dados por la convención:

$$0 \rightarrow C, \quad 1 \rightarrow S_2, \quad -1 \rightarrow S_1$$

TABLA 2

1	-1	-1	-1	-1	-1	-1
1	1	1	1	1	1	1
1	0	0	0	0	0	0
1	1	1	1	1	1	1
1	0	-1	-1	-1	-1	-1
1	0	0	-1	-1	-1	-1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	0	0	-1	-1	-1

REFERENCIAS

[1] Ciarlet P.G.: Discrete maximum principle for finite-difference operators. *Aequations Math.*, Vol. 4, pp. 338-352, 1970.

[2] Ciarlet P.G., Raviart P. A.: Maximum principle and uniform convergence for the finite element method. *Computer Methods in Applied Mechanics and Engineering*, Vol. 2, pp. 17-31, 1973.

[3] Di Marco S.: Métodos especiales para la resolución numérica de la inecuación bilátera de Isaacs, trabajo en preparación.

[4] Friedman A.: *Differential Games* (Wiley-Interscience, New York, 1971).

[5] Garroni M.G., Vivaldi M.A.: Bilateral inequalities and implicit unilateral system of the non-variational type. *Manuscripta Mathematica*, 33, pp. 177-215, 1980.

[6] González R., Tidball M. : Fast solution of discrete Isaacs's inequalities, *Rapport de Recherche* N°1167, INRIA, Rocquencourt, 1990.

[7] Ross S. M.: *Applied Probability Models with Optimization Applications* (Holden-Day, San Francisco, 1970).