Sisteme și algoritmi distribuiți
Curs 11

Sisteme liniare

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

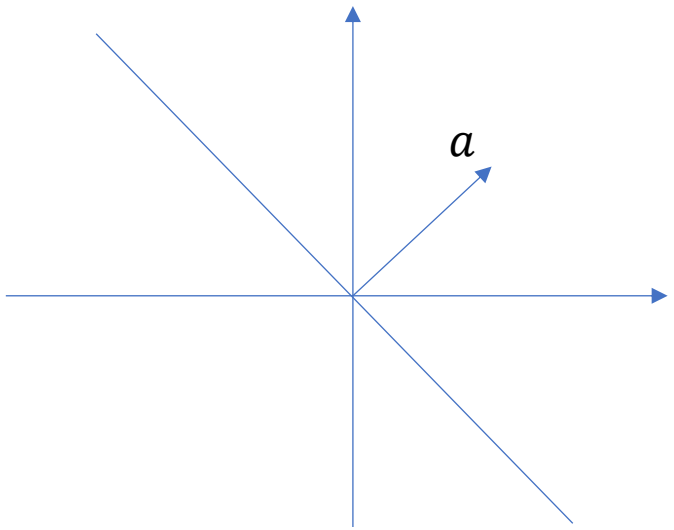
$$Ax = b$$

Hiperplan

În R^n, un hiperplan reprezintă un subspațiu liniar de dimensiune $n-1$ parametrizat de $a \in R^n, b \in R$
$H = \{x \in R^n : a^T x = b\}$.

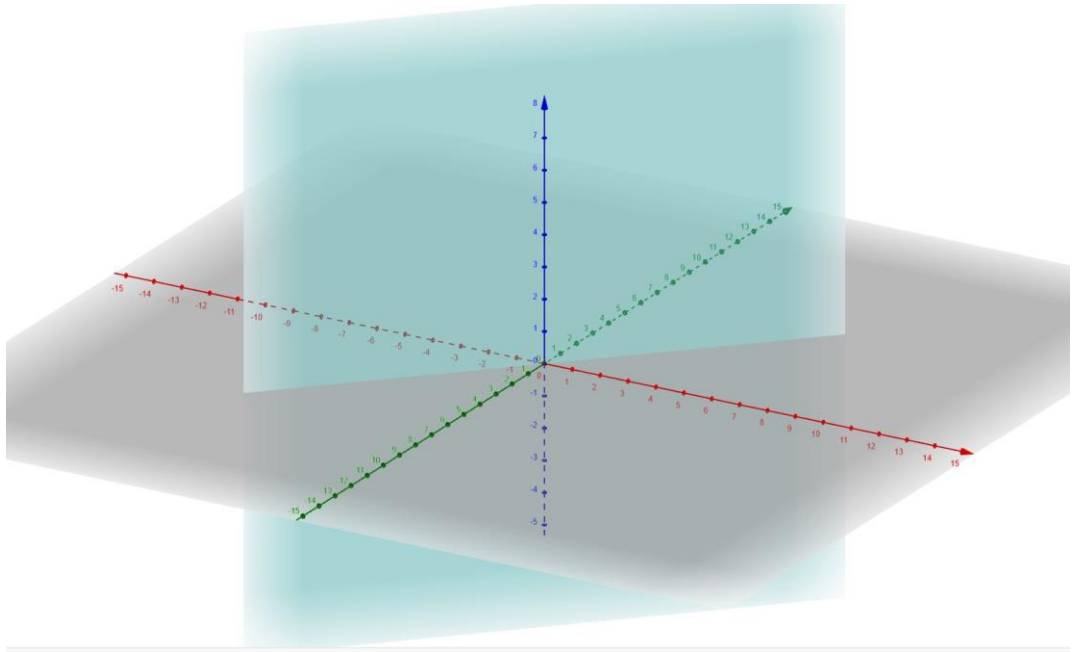
Exemple:

- în R^2 : $H = \{x \in R^2 : x_1 + x_2 = 0\}$



Exemplu R^3:

$$H = \{x \in R^3 : x_1 - x_2 = 0\}$$



Algoritmul Proiecțiilor Alternative

Rezolvarea sistemului

$$Ax = b, \qquad A \in R^{m \times n}, b \in R^m$$

este echivalentă cu determinarea

$$x \in H_1 \cap H_2 \cap \cdots \cap H_m,$$

unde $H_i = \{x: a_{(i)}^T x = b_i\}$. Schema generală APA selectează la iterația t un hiperplan $H_{i(t)}$ și actualizează iterația:

$$x_{t+1} := \pi_{H_{i(t)}}(x_t)$$

Alegerea $H_{i(t)}$ poate urma regula *ciclică*, *aleatoare*, *„greedy”* etc.

Algoritmul Proiecțiilor Alternative

Formatul distribuit al problemei găsirii $x \in H_1 \cap H_2 \cap \cdots \cap H_m$ presupune determinarea z $\in H_i = \{x: a_{(i)}^T x = b_i\}$ o sarcină individuală asociată nodului P_i.

Deci presupunem că P_i poate rezolva propria sarcină locală (i.e. determinarea unui punct din H_i, respectiv proiecția ortogonală pe H_i).

Starea $x_i(t) \in R^n$ reprezintă estimarea locală a soluției la pasul t. Dacă starea nodului P_i se află pe hiperplanul H_i, i.e. $x_i \in H_i$, atunci fiecare P_i va satisface ecuația $a_{(i)}^T x_i = b_i$, dar nu avem o soluție a întregului sistem.

Algoritmul de Consens Proiectat (ACP)

Algoritmul de Consens Proiectat compune pasul de consens cu cel de proiecție ortogonală:

$$x_i(t+1) = \pi_{H_i}\left(\sum_{j \in \mathcal{N}_i^- \cup i} w_{ij} x_j(t)\right) \qquad \forall i,$$

unde ponderile $w_{ij} \geq 0, \sum_{j \in \mathcal{N}_i^- \cup i} w_{ij} = 1$ (medie).

Dacă $H_i = R^n$, atunci ACP se reduce la Algoritmul Flooding de medie (din cursul 6):

$$x_i(t+1) = w_{ii} x_i(t) + \sum_{j \in \mathcal{N}_i^-} w_{ij} x_j(t) \qquad \forall i.$$

Algoritmul de Consens Proiectat (ACP)

Algoritmul de Consens Proiectat compune pasul de consens cu cel de proiecție ortogonală:

$$x_i(t+1) = \pi_{H_i}\left(\sum_{j \in \mathcal{N}_i^- \cup i} w_{ij} x_j(t)\right) \qquad \forall i.$$

- La fiecare iterație P_i își menține starea pe H_i (păstrează activă ecuația $a_{(i)}^T x = b_i$) prin operația de proiecție.
- P_i se apropie de consens prin operația de medie $w_i^T x(t)$
- Este suficient consensul pentru a garanta rezolvarea sistemului.

Algoritmul de Consens Proiectat (ACP)

Algoritmul de Consens Proiectat compune pasul de consens cu cel de proiecție ortogonală:

$$x_i(t+1) = \pi_{H_i}\left(\sum_{j \in \mathcal{N}_i^- \cup i} w_{ij} x_j(t)\right) \qquad \forall i$$

Teorema. Fie matricea ponderilor W dublu stohastică. Presupunem că există constanta $\eta > 0$ astfel încât toate ponderile $w_{ij} > 0$ satisfac $w_{ij} \geq \eta$ ($w_{ii} \geq \eta$). Dacă sistemul $Ax = b$ are soluție, atunci șirul generat de ACP atinge consensul asimptotic (într-una dintre soluțiile sistemului).

Probleme

1. Fie sistemul $\begin{cases} x_1 + x_2 - x_3 = 1 \\ x_1 - x_2 + 2x_3 = 2 \\ 2x_1 + 3x_2 = 5 \end{cases}$. Rezolvă Algoritmul Proiecțiilor Alternative (serial) acest sistem liniar? Stabiliți un punct inițial x(0) și scrieți primele 3 iterații APA serial, cu regula de alegere ale hiperplanelor ciclică/dinamică.

Alegeri posibile:

- Ciclică: $i(0) = 1, i(1) = 2, \dots, i(m-1) = m$
- Aleatoare: $randint(m)$
- Dinamică („greedy”): $i(t) = argmax_i \, |a_i^T x(t) - b_i|$

Probleme

1. APA serial converge doar dacă există o soluție a sistemului $Ax = b$. Deci pentru a determina convergența asimptotică este necesar calculul unei soluții a sistemului $\begin{cases} x_1 + x_2 - x_3 = 1 \\ x_1 - x_2 + 2x_3 = 2 \\ 2x_1 + 3x_2 = 5 \end{cases}$.

Alegere $x(0) = [0; 0; 0]^T$. Varianta ciclică:

$$x(1) = \pi_{H_1}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}\right) = \begin{bmatrix}0\\0\\0\end{bmatrix} + \frac{1}{3}\begin{bmatrix}1\\1\\-1\end{bmatrix} = \begin{bmatrix}1/3\\1/3\\-1/3\end{bmatrix}$$

$$x(2) = \pi_{H_2}\left(\begin{bmatrix}1/3\\1/3\\-1/3\end{bmatrix}\right) = \begin{bmatrix}1/3\\1/3\\-1/3\end{bmatrix} + \frac{4}{9}\begin{bmatrix}1\\-1\\2\end{bmatrix} = \begin{bmatrix}7/9\\-1/9\\5/9\end{bmatrix}$$

$$x(3) = \pi_{H_3}\left(\begin{bmatrix}7/9\\-1/9\\5/9\end{bmatrix}\right) = \frac{1}{9}\begin{bmatrix}7\\-1\\5\end{bmatrix} + \frac{34}{117}\begin{bmatrix}2\\3\\0\end{bmatrix} = \begin{bmatrix}159/117\\21/117\\5/9\end{bmatrix}$$

Costul unei iterații?

Probleme

1.(continuare) Alegere $x(0) = [0; 0; 0]^T$. Varianta dinamică:

$$|a_1^T x(0) - b_1| = 1, |a_2^T x(0) - b_2| = 2, |a_3^T x(0) - b_3| = 5$$

$$x(1) = \pi_{H_3}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}\right) = \begin{bmatrix}0\\0\\0\end{bmatrix} + \frac{1}{5}\begin{bmatrix}2\\3\\0\end{bmatrix} = \begin{bmatrix}2/5\\3/5\\0\end{bmatrix}$$

$$|a_1^T x(1) - b_1| = 0, |a_2^T x(1) - b_2| = \frac{11}{5}, |a_3^T x(1) - b_3| = 0$$

$$x(2) = \pi_{H_2}\left(\begin{bmatrix}2/5\\3/5\\0\end{bmatrix}\right) = \begin{bmatrix}2/5\\3/5\\0\end{bmatrix} + \frac{11}{30}\begin{bmatrix}1\\-1\\2\end{bmatrix} = \begin{bmatrix}23/30\\7/30\\22/30\end{bmatrix}$$

$$|a_1^T x(2) - b_1| = \frac{11}{15}, |a_2^T x(2) - b_2| = 0, |a_3^T x(2) - b_3| = \frac{83}{30}$$

$$x(3) = \pi_{H_3}\left(\begin{bmatrix}23/30\\7/30\\22/30\end{bmatrix}\right) = \frac{1}{30}\begin{bmatrix}23\\7\\22\end{bmatrix} + \frac{83}{390}\begin{bmatrix}2\\3\\0\end{bmatrix} = \begin{bmatrix}465/390\\340/390\\22/30\end{bmatrix}$$

Costul unei iterații?

Probleme

2. Fie sistemul $\begin{cases} x_1 + x_2 - x_3 = 1 \\ x_1 - x_2 + 2x_3 = 2 \\ 2x_1 + 3x_2 = 5 \end{cases}$. Fiecare hiperplan a fost distribuit către nodul de același index pe o topologie de tip inel cu $n = 3$ noduri.

a) Pentru ponderi uniforme și stări inițiale la alegere, calculați iterațiile ACP

$$x_i(t+1) = \pi_{H_i}\left(\sum_{j \in \mathcal{N}_i^- \cup i} w_{ij} x_j(t)\right) \qquad \forall i.$$

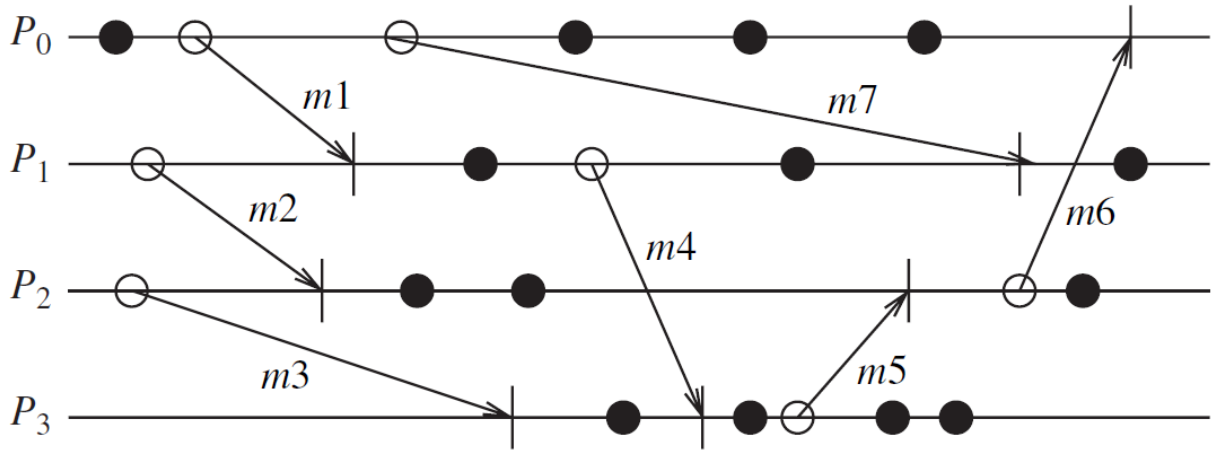
pentru $t = 1$ și $t = 2$.

b) Se va atinge consensul asimptotic? Dacă da, care este punctul de consens asimptotic $x(\infty)$?

Sisteme asincrone

Sistem distribuit asincron

1. Nu există sincronie între procese; nu avem o limită pe decalajul între ceasurile locale ale proceselor.
2. În acest fel: procesoarele rulează la viteze diferite, comunică diferit mesajele. **Întârzierile mesajelor pot fi (ne)limitate!**
3. Durata execuției unui pas nu este explicit limitată.



Sistem distribuit asincron

Teorema imposibilitate. Într-un sistem distribuit asincron este imposibil de atins consensul (distribuit) chiar și sub un singur defect de tip *crash*.

Idea demonstrației: în cazul unui potențial defect de tip crash, nu este posibil să se distingă între un proces defect și unul corect cu întârzieri în comunicație.

Consecință. Toate problemele care se pot reduce la una de consens, sunt imposibil de rezolvat sub un singur defect de tip *crash*, *e.g.* alegere lider, calcul distribuit de funcții, difuzare sigură etc.

Algoritmi asincroni - formalizare

Sub contextul asincron, este posibil ca P_i să realizeze actualizarea locală x_i folosind valori „*vechi*” ale componentelor lui x:

$$x_i(t+1) = f_i\left(x_1\left(\tau_1^i(t)\right), x_2\left(\tau_2^i(t)\right), \ldots, x_n\left(\tau_n^i(t)\right)\right),$$

- Timpii $\tau_j^i(t)$ reprezintă momentele timpului fizic în care se produc actualizările. Ele satisfac $0 \leq \tau_j^i(t) \leq t, \quad \forall t \in T$.
- T^i = setul momentelor de timp la care este actualizat x_i (dupa referința unui ceas global care masoară timpul real); e.g. $T^1 = \{0,2,5,6, \ldots.\}$
- Dacă t $\notin$ T^i atunci $x_i(t+1) = x_i(t), \quad \forall t \notin T^i$.

Algoritmi asincroni - formalizare

Sub contextul asincron, este posibil ca P_i să realizeze actualizarea locală x_i folosind valori „*vechi*” ale componentelor lui x:

$$x_i(t+1) = f_i\left(x_1\left(\tau_1^i(t)\right), x_2\left(\tau_2^i(t)\right), \ldots, x_n\left(\tau_n^i(t)\right)\right),$$

- Nu este nevoie ca procesoarele să cunoasca timpii $\tau_j^i(t)$ (sau mulțimea T^i) deoarece nu sunt folosiți în iterație.
- Diferența $t - \tau_j^i(t)$ reprezintă întârzierea de comunicație. Dacă $t - \tau_j^i(t) = 0$ la orice moment t, atunci execuția este sincronă.

Algoritmi asincroni - formalizare

$$x_i(t+1) = f_i\left(x_1\left(\tau_1^i(t)\right), x_2\left(\tau_2^i(t)\right), \ldots, x_n\left(\tau_n^i(t)\right)\right),$$

În general presupunem că fiecare P_i stochează o *vedere proprie* stării globale $x^i(t) = (x_1^i(t), x_2^i(t), \ldots, x_n^i(t))$, pe baza căreia actualizează $x_i^i(t)$ la $t \in T^i$ prin relația:

$$x_i^i(t+1) = f_i(x^i(t))$$

Algoritmi asincroni - formalizare

În general presupunem că fiecare P_i stochează o *vedere proprie* stării globale $x^i(t) = (x_1^i(t), x_2^i(t), \ldots, x_n^i(t))$, pe baza căreia actualizează $x_i^i(t)$ la $t \in T^i$ prin relația:

$$x_i^i(t+1) = f_i\left(x^i(t)\right).$$

Ocazional, P_i transmite pe $x_i^i(t+1)$ celorlalte procesoare. Când P_j primește pe noul x_i^i, îl memorează în locul x_i^j.

Algoritmi asincroni - formalizare

În general presupunem că fiecare P_i stochează o *vedere proprie* stării globale $x^i(t) = (x_1^i(t), x_2^i(t), \ldots, x_n^i(t))$, pe baza căreia actualizează $x_i^i(t)$ la $t \in T^i$ prin relația:

$$x_i^i(t+1) = f_i\left(x^i(t)\right).$$

Scop final: vectorul distribuit al stării globale

$$x(t) = (x_1^1(t), x_2^2(t), \ldots, x_n^n(t))$$

converge asimptotic către soluția x^* a problemei.

Exemplu

Determinați x astfel încât

$$x_1 = \frac{1}{2}x_1 + \frac{1}{2}x_2$$
$$x_2 = \frac{1}{2}x_1 + \frac{1}{2}x_2$$

- Mulțimea soluțiilor satisface: $x_1 = x_2$.
- Sincron, iterația $x(t+1) = Ax(t)$, ajunge după 1 pas la optim:

$$x(1) = \begin{bmatrix} \frac{1}{2}(x_1(0) + x_2(0)) \\ \frac{1}{2}(x_1(0) + x_2(0)) \end{bmatrix}$$

Algoritmi asincroni - exemplu

Scenariu:

- Avem 2 procesoare P_1, P_2
- Comunică la anumite momente $\{\tau_1, \tau_2, \dots\}$
- Transmiterea/folosirea informatiei comunicate se face instantaneu.

Algoritmi asincroni - exemplu

Scenariu:

- Avem 2 procesoare P_1, P_2
- Comunică la anumite momente $\{\tau_1, \tau_2, \dots\}$
- Transmiterea/folosirea informatiei comunicate se face instantaneu.

$$P_1\colon\ x_1(t+1) = \frac{x_1(t)}{2} + \frac{x_2(\tau_k)}{2}, \qquad \tau_k \le t < \tau_{k+1}$$

$$P_2\colon\ x_2(t+1) = \frac{x_1(\tau_k)}{2} + \frac{x_2(t)}{2}, \qquad \tau_k \le t < \tau_{k+1}$$

Algoritmi asincroni - exemplu

$$P_1\colon\ x_1(t+1) = \frac{x_1(t)}{2} + \frac{x_2(\tau_k)}{2}, \qquad \tau_k \le t < \tau_{k+1}$$

Între momentele τ_k și τ_{k+1}, P_1 menține $x_2(\tau_k)$ constant și execută iterația de mai sus de $\tau_{k+1} - \tau_k$ ori:

$$P_1\colon\ x_1(\tau_k + 1) = \frac{1}{2}x_1(\tau_k) + \frac{1}{2}x_2(\tau_k)$$

Algoritmi asincroni - exemplu

$$P_1: x_1(t+1) = \frac{x_1(t)}{2} + \frac{x_2(\tau_k)}{2}, \qquad \tau_k \le t < \tau_{k+1}$$

Între momentele τ_k și τ_{k+1}, P_1 menține $x_2(\tau_k)$ constant și execută iterația de mai sus de $\tau_{k+1} - \tau_k$ ori:

$$\begin{aligned} x_1(\tau_k + 2) &= \frac{1}{2} x_1(\tau_k + 1) + \frac{1}{2} x_2(\tau_k) \\ &= \tfrac{1}{2}\left[\tfrac{1}{2} x_1(\tau_k) + \tfrac{1}{2} x_2(\tau_k)\right] + \tfrac{1}{2} x_2(\tau_k) \\ &= \boldsymbol{\frac{1}{4} x_1(\tau_k) + \left[\frac{1}{2} + \frac{1}{4}\right] x_2(\tau_k)} \end{aligned}$$

Algoritmi asincroni - exemplu

$$P_1\colon\ x_1(t+1) = \frac{x_1(t)}{2} + \frac{x_2(\tau_k)}{2}, \qquad \tau_k \le t < \tau_{k+1}$$

Între momentele τ_k și τ_{k+1}, P_1 menține $x_2(\tau_k)$ constant și execută iterația de mai sus de $\tau_{k+1} - \tau_k$ ori:

$$x_1(\tau_k + i) = \left(\frac{1}{2}\right)^i x_1(\tau_k) + \left[\frac{1}{2} + \frac{1}{4} + \cdots + \left(\frac{1}{2}\right)^i\right] x_2(\tau_k)$$

Algoritmi asincroni - exemplu

$$P_1\colon\ x_1(t+1) = \frac{x_1(t)}{2} + \frac{x_2(\tau_k)}{2}, \qquad \tau_k \le t < \tau_{k+1}$$

Între momentele τ_k și τ_{k+1}, P_1 menține $x_2(\tau_k)$ constant și execută iterația de mai sus de $\tau_{k+1} - \tau_k$ ori:

$$x_1(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_1(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_2(\tau_k)$$

Algoritmi asincroni - exemplu

$$P_1\colon\ x_1(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_1(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_2(\tau_k)$$

$$P_2\colon\ x_2(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_2(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_1(\tau_k)$$

Algoritmi asincroni - exemplu

$$P_1\colon x_1(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_1(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_2(\tau_k)$$

$$P_2\colon x_2(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_2(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_1(\tau_k)$$

Notăm $\epsilon_k = 2\left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}$:

$$\begin{aligned} |x_2(\tau_{k+1}) - x_1(\tau_{k+1})| &\leq (1-\epsilon_k)|x_2(\tau_k) - x_1(\tau_k)| \\ &\leq \textstyle\prod_i (1-\epsilon_i)\,|x_2(0) - x_1(0)| \end{aligned}$$

Algoritmi asincroni - exemplu

$$P_1\colon x_1(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_1(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_2(\tau_k)$$

$$P_2\colon x_2(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_2(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_1(\tau_k)$$

Notăm $\epsilon_k = 2\left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}$:

$$\begin{aligned} |x_2(\tau_{k+1}) - x_1(\tau_{k+1})| &\leq (1-\epsilon_k)|x_2(\tau_k) - x_1(\tau_k)| \\ &\leq \textstyle\prod_i (1-\epsilon_i)\,|x_2(0) - x_1(0)| \end{aligned}$$

Condiție de convergență:

$$\lim_{k\to\infty} \prod_{i=0}^{k} (1-\epsilon_i) = 0.$$

Convergența asincronă

Procesorul i:

$$x_i(t+1) = f_i\left(x_1\left(\tau_1^i(t)\right), \cdots, x_n\left(\tau_n^i(t)\right)\right)$$

unde $\tau_j^i(t)$ sunt momente de timp a.i.: $0 \leq \tau_j^i(t) \leq t, \quad \forall t \in T^i$.

Procesorul i stochează vectorul $\mathrm{x^i(t)} = (x_1^i(t), \cdots, x_n^i(t))$ și calculează $x_i(t+1) = f_i(x^i(t))$ la momentele $t \in T^i$.

În general, întârzierea de comunicație este nemărginită: $\mathbf{t - \tau_j^i(t) \geq 0}$

Convergența asincronă

Ipoteza de asincronism total. Mulțimile T^i sunt infinite și, pentru $\{t_k\} \subset T^i, t_k \to \infty$, avem $\lim_{k\to\infty} \tau_j^i(t_k) = \infty$ pentru orice j.

- Fiecare componentă este actualizată de un număr infinit de ori.
- Informația „veche” este eliminată din sistem, i.e. pentru orice t_1 există t_2 astfel încât:

$$\tau_j^i(t) \geq t_1, \qquad \forall t \geq t_2.$$

Convergența asincronă

Ipoteze de convergență asincronă. Există șirul de mulțimi $\{X(t)\}_{t\geq 0}$ *astfel încât*:

$$\cdots \subset X(t+1) \subset X(t) \subset \cdots \subset X$$

și care satisfac condițiile:

1) (Convergența sincronă). $\mathrm{F}(x) \in X(t+1)$, pentru oricare $t, x \in X(t)$. Mai mult, pentru orice șir $\{y^t\}_{t\geq 0}$ care satisface $y^t \in X(t)$, toate punctele limită ale y^t sunt puncte fixe ale F.
2) (Condiția de separabilitate). $X(t) = X_1(t) \times \cdots \times X_n(t)$

Convergența asincronă

Teoremă (Convergența asincronă). Sub ipotezele precedente, dacă $x(0) \in X(0)$ atunci toate punctele limită ale $\{x(t)\}$ sunt **puncte fixe** ale lui F, i.e.

$$x^* = F(x^*) = \begin{bmatrix} f_1(x^*) \\ \vdots \\ f_n(x^*) \end{bmatrix}.$$

- Se asigură consens asimptotic la $x(\infty) := x^*$.
- Demonstrația folosește argumentul inducției.
- Ipotezele sunt suficiente, nu și necesare.
- Întârzierile pot fi nemărginite!

Aplicație în caz liniar

Fie operatorul liniar:

$$F(x) \coloneqq Hx - q.$$

Dacă operatorul F este *contracție* cu parametru $\alpha \in (0,1)$ în raport cu norma $\|\cdot\|_\infty$

$$\|F(x) - F(y)\|_\infty \leq \alpha \|x - y\|_\infty, \qquad \forall x, y,$$

și definim

$$X(k) = \{x : \|x - x^*\|_\infty \leq \alpha^k \|x(0) - x^*\|_\infty\}$$

unde x^* punct fix și α parametrul de contracție, atunci condițiile ipotezelor precedente *sunt satisfăcute*, și convergența asincronă este asigurată!

Aplicație în caz liniar

Operatorul liniar $F(x) \coloneqq Hx - q$ este *contracție* cu parametru $\alpha \in (0,1)$ în raport cu norma $\|\cdot\|_\infty$ dacă H are raza spectrală subunitară

$$\mathrm{Max}_i \, |\lambda_i(H)| < \alpha < 1.$$

- Condiția se verifică calculând spectrul matricii H.
- Ipoteza este restrictivă pentru algoritmii de medie, unde tipic matricea H este stohastică (raza spectrală egală cu 1).

Sistem liniar pătratic

$$Ax = b, A \in R^{n \times n}$$

Este echivalent cu

$$x_1 = -\frac{1}{A_{11}}(A_{12}x_2 + \cdots + A_{1n}x_n - b_1)$$

$$x_2 = -\frac{1}{A_{22}}(A_{21}x_1 + \cdots + A_{1n}x_n - b_2)$$

...

$$x_n = -\frac{1}{A_{nn}}(A_{n1}x_1 + \cdots + A_{nn-1}x_{n-1} - b_n)$$

Algoritmul Jacobi sincron

Ideea algoritmului Jacobi sincron:

$$P_i\colon\ x_i(t+1) = -\frac{1}{A_{ii}}\left(\sum_{j\neq i} A_{ij}x_j(t) - b_i\right)$$

$$x(t+1) = D^{-1}\big(b - Rx(t)\big)$$

- Pentru calcularea $x_i(t+1), P_i$ așteaptă $x_j(t)$ de la P_j, unde $j = 1, \dots, p, j \neq i$
- Este necesar ca fiecare procesor să stocheze $x(t)$!

Algoritmi asincroni

Algoritm Jacobi asincron*:*

$$x_i(t+1) = -\frac{1}{A_{ii}}\left(\sum_{j\neq i} A_{ij}x_j\left(\tau_j^i(t)\right) - b_i\right)$$

Sub ipotezele precedente, alg. Jacobi converge la $x^* = -D^{-1}Rx^* + D^{-1}b$.

În acest caz, ipotezele de convergență se reduc la:

- $D^{-1}R$ contracție în raport cu $\|\cdot\|_\infty$
- $raza\ spectrala$ a matricii $|D^{-1}R|$ sa fie subunitara, i.e. $\rho(|D^{-1}R|) < 1$.

Ipotezele sunt foarte restrictive pentru algoritmii de medie (consens).

Algoritmi partial asincroni

Cunoaștem una dintre limitările contextului asincron, când $\mathrm{t} - \tau_j^i(t)$ este posibil nemărginit!

Soluție: algoritmi parțial asincroni, în care întârzierile mesajelor $\mathbf{t} - \boldsymbol{\tau_j^i(t)}$ sunt mărginite!

Echivalent: există o margine de timp pe comunicarea inter-proces!

Algoritmi partial asincroni - formalizare

Procesorul P_i execută:

$$x_i(t+1) = f_i\left(x_1\left(\tau_1^i(t)\right), x_2\left(\tau_2^i(t)\right), \ldots, x_n\left(\tau_n^i(t)\right)\right),$$

Ipoteza de asincronism parţial. Pentru oricare i și t ≥ 0, cel puțin un element din $\{t, t+1, \ldots, t+B-1\}$ se află în T^i. Mai mult,

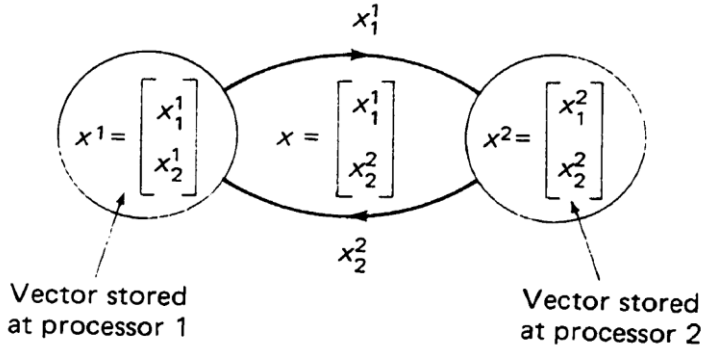
a) $t - B \leq \tau_j^i(t) \leq t, \quad \forall i, j, t \geq 0, t \in T^i$

b) $\tau_i^i(t) = t$

Exemplu – parțial asincronie

Determinați x astfel încât

$$x_1 = \frac{1}{2}x_1 + \frac{1}{2}x_2$$
$$x_2 = \frac{1}{2}x_1 + \frac{1}{2}x_2$$



Avem P_1 si P_2:

P_1 stocheaza $x^1 = (x_1^1, x_2^1)$ si actualizeaza x_1^1 la momentele $t \in T^1$

P_2 stocheaza $x^2 = (x_1^2, x_2^2)$ si actualizeaza x_2^2 la momentele $t \in T^2$

Algoritmi asincroni - exemplu

$$P_1\colon x_1(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_1(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_2(\tau_k)$$

$$P_2\colon x_2(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_2(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_1(\tau_k)$$

Notăm $\epsilon_k = 2\left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}$:

$$\begin{aligned} |x_2(\tau_{k+1}) - x_1(\tau_{k+1})| &\leq (1-\epsilon_k)|x_2(\tau_k) - x_1(\tau_k)| \\ &\leq \textstyle\prod_i (1-\epsilon_i)\,|x_2(0) - x_1(0)| \end{aligned}$$

Condiție de convergență:

$$\lim_{k\to\infty} \prod_{i=0}^{k} (1-\epsilon_i) = 0.$$

Convergenta- exemplu

$$P_1\colon x_1(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_1(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_2(\tau_k)$$

$$P_2\colon x_2(\tau_{k+1}) = \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k} x_2(\tau_k) + \left(1 - \left(\frac{1}{2}\right)^{\tau_{k+1}-\tau_k}\right) x_1(\tau_k)$$

Condiție pentru convergență: $\lim_{k\to\infty} \prod_{i=0}^{k}(1-\epsilon_i) = 0$.

Dacă $\tau_{k+1} - \tau_k \leq B$ atunci $\epsilon_k > \left(\frac{1}{2}\right)^B$

$$\Rightarrow \prod_i (1-\epsilon_i) \leq \prod_i \left(1 - \left(\frac{1}{2}\right)^B\right) \to 0$$

Ipoteza de parțial asincronie este suficientă pentru rezolvarea problemei de consens (medie).

Referințe

[1] Nedic, Angelia, Asuman Ozdaglar, and Pablo A. Parrilo. "Constrained consensus and optimization in multi-agent networks." *IEEE Transactions on Automatic Control* 55.4 (2010): 922-938.

[2] Bertsekas, Dimitri, and John Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.