

APPROXIMATE DYNAMIC PROGRAMMING

LECTURE 2

LECTURE OUTLINE

- Review of discounted problem theory
- Review of shorthand notation
- Algorithms for discounted DP
- Value iteration
- Policy iteration
- Optimistic policy iteration
- Q-factors and Q-learning
- A more abstract view of DP
- Extensions of discounted DP
- Value and policy iteration
- Asynchronous algorithms

DISCOUNTED PROBLEMS/BOUNDED COST

- Stationary system with arbitrary state space

$$x_{k+1} = f(x_k, u_k, w_k), \quad k = 0, 1, \dots$$

- Cost of a policy $\pi = \{\mu_0, \mu_1, \dots\}$

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} E_{w_k} \left\{ \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k), w_k) \right\}$$

with $\alpha < 1$, and for some M , we have $|g(x, u, w)| \leq M$ for all (x, u, w)

- **Shorthand notation for DP mappings** (operate on functions of state to produce other functions)

$$(TJ)(x) = \min_{u \in U(x)} E_w \left\{ g(x, u, w) + \alpha J(f(x, u, w)) \right\}, \quad \forall x$$

TJ is the optimal cost function for the one-stage problem with stage cost g and terminal cost αJ .

- For any stationary policy μ

$$(T_\mu J)(x) = E_w \left\{ g(x, \mu(x), w) + \alpha J(f(x, \mu(x), w)) \right\}, \quad \forall x$$

“SHORTHAND” THEORY – A SUMMARY

- **Cost function expressions** [with $J_0(x) \equiv 0$]

$$J_\pi(x) = \lim_{k \rightarrow \infty} (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_k} J_0)(x), \quad J_\mu(x) = \lim_{k \rightarrow \infty} (T_\mu^k J_0)(x)$$

- **Bellman's equation:** $J^* = T J^*$, $J_\mu = T_\mu J_\mu$ or

$$J^*(x) = \min_{u \in U(x)} E_w \{ g(x, u, w) + \alpha J^*(f(x, u, w)) \}, \quad \forall x$$

$$J_\mu(x) = E_w \{ g(x, \mu(x), w) + \alpha J_\mu(f(x, \mu(x), w)) \}, \quad \forall x$$

- **Optimality condition:**

$$\mu: \text{optimal} \quad \Leftrightarrow \quad T_\mu J^* = T J^*$$

i.e.,

$$\mu(x) \in \arg \min_{u \in U(x)} E_w \{ g(x, u, w) + \alpha J^*(f(x, u, w)) \}, \quad \forall x$$

- **Value iteration:** For any (bounded) J

$$J^*(x) = \lim_{k \rightarrow \infty} (T^k J)(x), \quad \forall x$$

MAJOR PROPERTIES

- **Monotonicity property:** For any functions J and J' on the state space X such that $J(x) \leq J'(x)$ for all $x \in X$, and any μ

$$(TJ)(x) \leq (TJ')(x), \quad (T_\mu J)(x) \leq (T_\mu J')(x), \quad \forall x \in X.$$

- **Contraction property:** For any bounded functions J and J' , and any μ ,

$$\max_x |(TJ)(x) - (TJ')(x)| \leq \alpha \max_x |J(x) - J'(x)|,$$

$$\max_x |(T_\mu J)(x) - (T_\mu J')(x)| \leq \alpha \max_x |J(x) - J'(x)|.$$

- **Compact Contraction Notation:**

$$\|TJ - TJ'\| \leq \alpha \|J - J'\|, \quad \|T_\mu J - T_\mu J'\| \leq \alpha \|J - J'\|,$$

where for any bounded function J , we denote by $\|J\|$ the sup-norm

$$\|J\| = \max_x |J(x)|.$$

THE TWO MAIN ALGORITHMS: VI AND PI

- **Value iteration:** For any (bounded) J

$$J^*(x) = \lim_{k \rightarrow \infty} (T^k J)(x), \quad \forall x$$

- **Policy iteration:** Given μ^k
 - **Policy evaluation:** Find J_{μ^k} by solving

$$J_{\mu^k}(x) = E_w \left\{ g(x, \mu(x), w) + \alpha J_{\mu^k}(f(x, \mu^k(x), w)) \right\}, \quad \forall x$$

$$\text{or } J_{\mu^k} = T_{\mu^k} J_{\mu^k}$$

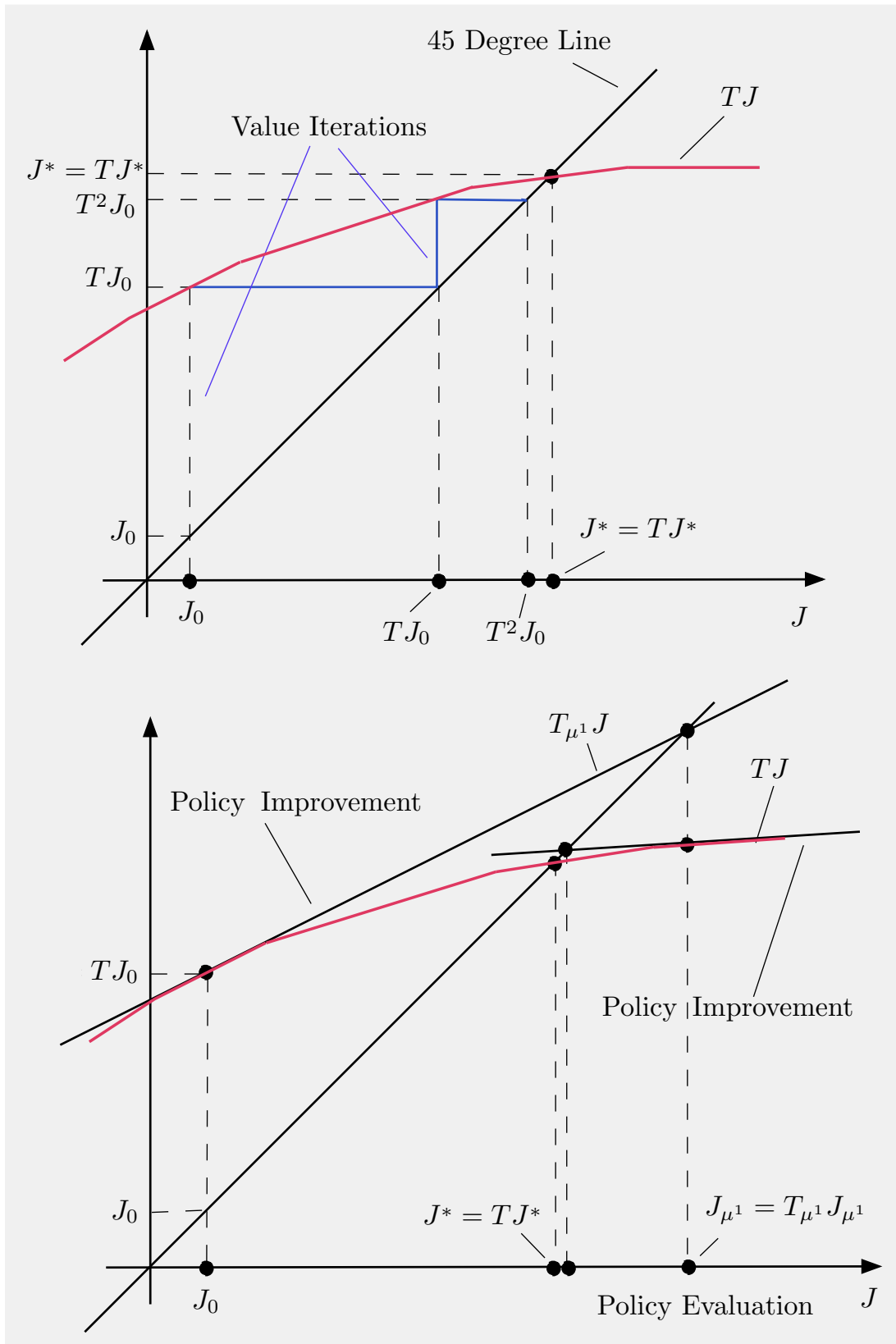
- **Policy improvement:** Let μ^{k+1} be such that

$$\mu^{k+1}(x) \in \arg \min_{u \in U(x)} E_w \left\{ g(x, u, w) + \alpha J_{\mu^k}(f(x, u, w)) \right\}, \quad \forall x$$

$$\text{or } T_{\mu^{k+1}} J_{\mu^k} = T J_{\mu^k}$$

- For finite state space **policy evaluation is equivalent to solving a linear system of equations**
- Dimension of the system is equal to the number of states.
- **For large problems, exact PI is out of the question (even though it terminates finitely)**

INTERPRETATION OF VI AND PI



JUSTIFICATION OF POLICY ITERATION

- We can show that $J_{\mu^{k+1}} \leq J_{\mu^k}$ for all k
- **Proof:** For given k , we have

$$T_{\mu^{k+1}} J_{\mu^k} = T J_{\mu^k} \leq T_{\mu^k} J_{\mu^k} = J_{\mu^k}$$

Using the monotonicity property of DP,

$$J_{\mu^k} \geq T_{\mu^{k+1}} J_{\mu^k} \geq T_{\mu^{k+1}}^2 J_{\mu^k} \geq \dots \geq \lim_{N \rightarrow \infty} T_{\mu^{k+1}}^N J_{\mu^k}$$

- Since

$$\lim_{N \rightarrow \infty} T_{\mu^{k+1}}^N J_{\mu^k} = J_{\mu^{k+1}}$$

we have $J_{\mu^k} \geq J_{\mu^{k+1}}$.

- If $J_{\mu^k} = J_{\mu^{k+1}}$, all above inequalities hold as equations, so J_{μ^k} solves Bellman's equation. Hence $J_{\mu^k} = J^*$

- Thus at iteration k either the algorithm generates a strictly improved policy or it finds an optimal policy

- For a finite spaces MDP, there are finitely many stationary policies, so the algorithm terminates with an optimal policy

APPROXIMATE PI

- Suppose that the policy evaluation is approximate,

$$\|J_k - J_{\mu^k}\| \leq \delta, \quad k = 0, 1, \dots$$

and policy improvement is approximate,

$$\|T_{\mu^{k+1}} J_k - T J_k\| \leq \epsilon, \quad k = 0, 1, \dots$$

where δ and ϵ are some positive scalars.

- **Error Bound I:** The sequence $\{\mu^k\}$ generated by approximate policy iteration satisfies

$$\limsup_{k \rightarrow \infty} \|J_{\mu^k} - J^*\| \leq \frac{\epsilon + 2\alpha\delta}{(1 - \alpha)^2}$$

- **Typical practical behavior:** The method makes steady progress up to a point and then the iterates J_{μ^k} oscillate within a neighborhood of J^* .
- **Error Bound II:** If in addition the sequence $\{\mu^k\}$ “terminates” at $\bar{\mu}$ (i.e., keeps generating $\bar{\mu}$)

$$\|J_{\bar{\mu}} - J^*\| \leq \frac{\epsilon + 2\alpha\delta}{1 - \alpha}$$

OPTIMISTIC POLICY ITERATION

- **Optimistic PI (more efficient):** This is PI, where policy evaluation is done approximately, with a finite number of VI
- So we approximate the policy evaluation

$$J_\mu \approx T_\mu^m J$$

for some number $m \in [1, \infty)$ and initial J

- **Shorthand definition:** For some integers m_k

$$T_{\mu^k} J_k = T J_k, \quad J_{k+1} = T_{\mu^k}^{m_k} J_k, \quad k = 0, 1, \dots$$

- If $m_k \equiv 1$ it becomes VI
- If $m_k = \infty$ it becomes PI
- Can be shown to converge (in an infinite number of iterations)
- **Typically works faster than VI and PI** (for large problems)

Q-LEARNING I

- We can write Bellman's equation as

$$J^*(x) = \min_{u \in U(x)} Q^*(x, u), \quad \forall x,$$

where Q^* is the unique solution of

$$Q^*(x, u) = E \left\{ g(x, u, w) + \alpha \min_{v \in U(\bar{x})} Q^*(\bar{x}, v) \right\}$$

with $\bar{x} = f(x, u, w)$

- $Q^*(x, u)$ is called the **optimal Q-factor** of (x, u)
- We can equivalently write the VI method as

$$J_{k+1}(x) = \min_{u \in U(x)} Q_{k+1}(x, u), \quad \forall x,$$

where Q_{k+1} is generated by

$$Q_{k+1}(x, u) = E \left\{ g(x, u, w) + \alpha \min_{v \in U(\bar{x})} Q_k(\bar{x}, v) \right\}$$

with $\bar{x} = f(x, u, w)$

Q-LEARNING II

- Q-factors are no different than costs
- They satisfy a Bellman equation $Q = FQ$ where

$$(FQ)(x, u) = E \left\{ g(x, u, w) + \alpha \min_{v \in U(\bar{x})} Q(\bar{x}, v) \right\}$$

where $\bar{x} = f(x, u, w)$

- VI and PI for Q-factors are mathematically equivalent to VI and PI for costs
- They require equal amount of computation ... they just need more storage
- Having optimal Q-factors is convenient when implementing an optimal policy on-line by

$$\mu^*(x) = \min_{u \in U(x)} Q^*(x, u)$$

- Once $Q^*(x, u)$ are known, the model [g and $E\{\cdot\}$] is not needed. **Model-free operation.**
- Later we will see how stochastic/sampling methods can be used to calculate (approximations of) $Q^*(x, u)$ using a simulator of the system (no model needed)

A MORE GENERAL/ABSTRACT VIEW OF DP

- Let Y be a **real vector space with a norm** $\|\cdot\|$
- A function $F : Y \mapsto Y$ is said to be a **contraction mapping** if for some $\rho \in (0, 1)$, we have

$$\|Fy - Fz\| \leq \rho\|y - z\|, \quad \text{for all } y, z \in Y.$$

ρ is called the **modulus of contraction** of F .

- **Important example:** Let X be a set (e.g., state space in DP), $v : X \mapsto \mathfrak{R}$ be a positive-valued function. Let $B(X)$ be the set of all functions $J : X \mapsto \mathfrak{R}$ such that $J(x)/v(x)$ is bounded over x .

- We define a norm on $B(X)$, called the **weighted sup-norm**, by

$$\|J\| = \max_{x \in X} \frac{|J(x)|}{v(x)}.$$

- **Important special case:** The discounted problem mappings T and T_μ [for $v(x) \equiv 1$, $\rho = \alpha$].

A DP-LIKE CONTRACTION MAPPING

- Let $X = \{1, 2, \dots\}$, and let $F : B(X) \mapsto B(X)$ be a **linear** mapping of the form

$$(FJ)(i) = b_i + \sum_{j \in X} a_{ij} J(j), \quad \forall i = 1, 2, \dots$$

where b_i and a_{ij} are some scalars. Then F is a contraction with modulus ρ if and only if

$$\frac{\sum_{j \in X} |a_{ij}| v(j)}{v(i)} \leq \rho, \quad \forall i = 1, 2, \dots$$

- Let $F : B(X) \mapsto B(X)$ be a mapping of the form

$$(FJ)(i) = \min_{\mu \in M} (F_\mu J)(i), \quad \forall i = 1, 2, \dots$$

where M is parameter set, and for each $\mu \in M$, F_μ is a contraction mapping from $B(X)$ to $B(X)$ with modulus ρ . Then F is a contraction mapping with modulus ρ .

- **Allows extensions of main DP results from bounded cost to unbounded cost.**

CONTRACTION MAPPING FIXED-POINT TH.

- **Contraction Mapping Fixed-Point Theorem:** If $F : B(X) \mapsto B(X)$ is a contraction with modulus $\rho \in (0, 1)$, then there exists a unique $J^* \in B(X)$ such that

$$J^* = FJ^*.$$

Furthermore, if J is any function in $B(X)$, then $\{F^k J\}$ converges to J^* and we have

$$\|F^k J - J^*\| \leq \rho^k \|J - J^*\|, \quad k = 1, 2, \dots$$

- This is a special case of a general result for contraction mappings $F : Y \mapsto Y$ over normed vector spaces Y that are *complete*: every sequence $\{y_k\}$ that is Cauchy (satisfies $\|y_m - y_n\| \rightarrow 0$ as $m, n \rightarrow \infty$) converges.
- The space $B(X)$ is complete (see the text for a proof).

ABSTRACT FORMS OF DP

- We consider an abstract form of DP based on monotonicity and contraction
- **Abstract Mapping:** Denote $R(X)$: set of real-valued functions $J : X \mapsto \mathfrak{R}$, and let $H : X \times U \times R(X) \mapsto \mathfrak{R}$ be a given mapping. We consider the mapping

$$(TJ)(x) = \min_{u \in U(x)} H(x, u, J), \quad \forall x \in X.$$

- We assume that $(TJ)(x) > -\infty$ for all $x \in X$, so T maps $R(X)$ into $R(X)$.
- **Abstract Policies:** Let \mathcal{M} be the set of “policies”, i.e., functions μ such that $\mu(x) \in U(x)$ for all $x \in X$.
- For each $\mu \in \mathcal{M}$, we consider the mapping $T_\mu : R(X) \mapsto R(X)$ defined by

$$(T_\mu J)(x) = H(x, \mu(x), J), \quad \forall x \in X.$$

- Find a function $J^* \in R(X)$ such that

$$J^*(x) = \min_{u \in U(x)} H(x, u, J^*), \quad \forall x \in X$$

EXAMPLES

- **Discounted problems** (and stochastic shortest paths-SSP for $\alpha = 1$)

$$H(x, u, J) = E\{g(x, u, w) + \alpha J(f(x, u, w))\}$$

- **Discounted “discrete-state continuous-time” Semi-Markov Problems** (e.g., queueing)

$$H(x, u, J) = G(x, u) + \sum_{y=1}^n m_{xy}(u)J(y)$$

where m_{xy} are “discounted” transition probabilities, defined by the distribution of transition times

- **Minimax Problems/Games**

$$H(x, u, J) = \max_{w \in W(x, u)} [g(x, u, w) + \alpha J(f(x, u, w))]$$

- **Shortest Path Problems**

$$H(x, u, J) = \begin{cases} a_{xu} + J(u) & \text{if } u \neq d, \\ a_{xd} & \text{if } u = d \end{cases}$$

where d is the destination. There are **stochastic and minimax versions** of this problem

ASSUMPTIONS

- **Monotonicity:** If $J, J' \in R(X)$ and $J \leq J'$,

$$H(x, u, J) \leq H(x, u, J'), \quad \forall x \in X, u \in U(x)$$

- We can show all the standard analytical and computational results of discounted DP if monotonicity **and** the following assumption holds:

- **Contraction:**

- For every $J \in B(X)$, the functions $T_\mu J$ and TJ belong to $B(X)$
- For some $\alpha \in (0, 1)$, and all μ and $J, J' \in B(X)$, we have

$$\|T_\mu J - T_\mu J'\| \leq \alpha \|J - J'\|$$

- With just the monotonicity assumption (as in undiscounted problems) we can still show various forms of the basic results under appropriate assumptions

- A weaker substitute for contraction assumption is **semicontractiveness**: (roughly) for some μ , T_μ is a contraction and for others it is not; also the “noncontractive” μ are not optimal

RESULTS USING CONTRACTION

- **Proposition 1:** The mappings T_μ and T are weighted sup-norm contraction mappings with modulus α over $B(X)$, and have unique fixed points in $B(X)$, denoted J_μ and J^* , respectively (cf. **Bellman's equation**).

Proof: From the contraction property of H .

- **Proposition 2:** For any $J \in B(X)$ and $\mu \in \mathcal{M}$,

$$\lim_{k \rightarrow \infty} T_\mu^k J = J_\mu, \quad \lim_{k \rightarrow \infty} T^k J = J^*$$

(cf. **convergence of value iteration**).

Proof: From the contraction property of T_μ and T .

- **Proposition 3:** We have $T_\mu J^* = T J^*$ if and only if $J_\mu = J^*$ (cf. **optimality condition**).

Proof: $T_\mu J^* = T J^*$, then $T_\mu J^* = J^*$, implying $J^* = J_\mu$. Conversely, if $J_\mu = J^*$, then $T_\mu J^* = T_\mu J_\mu = J_\mu = J^* = T J^*$.

RESULTS USING MON. AND CONTRACTION

- **Optimality of fixed point:**

$$J^*(x) = \min_{\mu \in \mathcal{M}} J_\mu(x), \quad \forall x \in X$$

- **Existence of a nearly optimal policy:** For every $\epsilon > 0$, there exists $\mu_\epsilon \in \mathcal{M}$ such that

$$J^*(x) \leq J_{\mu_\epsilon}(x) \leq J^*(x) + \epsilon, \quad \forall x \in X$$

- **Nonstationary policies:** Consider the set Π of all sequences $\pi = \{\mu_0, \mu_1, \dots\}$ with $\mu_k \in \mathcal{M}$ for all k , and define

$$J_\pi(x) = \liminf_{k \rightarrow \infty} (T_{\mu_0} T_{\mu_1} \cdots T_{\mu_k} J)(x), \quad \forall x \in X,$$

with J being any function (the choice of J does not matter)

- We have

$$J^*(x) = \min_{\pi \in \Pi} J_\pi(x), \quad \forall x \in X$$

THE TWO MAIN ALGORITHMS: VI AND PI

- **Value iteration:** For any (bounded) J

$$J^*(x) = \lim_{k \rightarrow \infty} (T^k J)(x), \quad \forall x$$

- **Policy iteration:** Given μ^k
 - **Policy evaluation:** Find J_{μ^k} by solving

$$J_{\mu^k} = T_{\mu^k} J_{\mu^k}$$

- **Policy improvement:** Find μ^{k+1} such that

$$T_{\mu^{k+1}} J_{\mu^k} = T J_{\mu^k}$$

- **Optimistic PI:** This is PI, where policy evaluation is carried out by a finite number of VI
 - Shorthand definition: For some integers m_k

$$T_{\mu^k} J_k = T J_k, \quad J_{k+1} = T_{\mu^k}^{m_k} J_k, \quad k = 0, 1, \dots$$

- If $m_k \equiv 1$ it becomes VI
- If $m_k = \infty$ it becomes PI
- For intermediate values of m_k , it is generally more efficient than either VI or PI

ASYNCHRONOUS ALGORITHMS

- Motivation for asynchronous algorithms
 - Faster convergence
 - Parallel and distributed computation
 - Simulation-based implementations
- **General framework:** Partition X into disjoint nonempty subsets X_1, \dots, X_m , and use separate processor ℓ updating $J(x)$ for $x \in X_\ell$
- Let J be partitioned as

$$J = (J_1, \dots, J_m),$$

where J_ℓ is the restriction of J on the set X_ℓ .

- **Synchronous algorithm:**

$$J_\ell^{t+1}(x) = T(J_1^t, \dots, J_m^t)(x), \quad x \in X_\ell, \ell = 1, \dots, m$$

- **Asynchronous algorithm:** For some subsets of times \mathcal{R}_ℓ ,

$$J_\ell^{t+1}(x) = \begin{cases} T(J_1^{\tau_{\ell 1}(t)}, \dots, J_m^{\tau_{\ell m}(t)})(x) & \text{if } t \in \mathcal{R}_\ell, \\ J_\ell^t(x) & \text{if } t \notin \mathcal{R}_\ell \end{cases}$$

where $t - \tau_{\ell j}(t)$ are communication “delays”

ONE-STATE-AT-A-TIME ITERATIONS

- **Important special case:** Assume n “states”, a separate processor for each state, and no delays
- Generate a sequence of states $\{x^0, x^1, \dots\}$, generated in some way, possibly by simulation (each state is generated infinitely often)
- **Asynchronous VI:**

$$J_\ell^{t+1} = \begin{cases} T(J_1^t, \dots, J_n^t)(\ell) & \text{if } \ell = x^t, \\ J_\ell^t & \text{if } \ell \neq x^t, \end{cases}$$

where $T(J_1^t, \dots, J_n^t)(\ell)$ denotes the ℓ -th component of the vector

$$T(J_1^t, \dots, J_n^t) = TJ^t,$$

- The special case where

$$\{x^0, x^1, \dots\} = \{1, \dots, n, 1, \dots, n, 1, \dots\}$$

is the **Gauss-Seidel method**

- We can show that $J^t \rightarrow J^*$ under the contraction assumption

ASYNCHRONOUS CONV. THEOREM I

- Assume that for all $\ell, j = 1, \dots, m$, \mathcal{R}_ℓ is infinite and $\lim_{t \rightarrow \infty} \tau_{\ell j}(t) = \infty$
- **Proposition:** Let T have a unique fixed point J^* , and assume that there is a sequence of nonempty subsets $\{S(k)\} \subset R(X)$ with $S(k+1) \subset S(k)$ for all k , and with the following properties:

- (1) **Synchronous Convergence Condition:** Every sequence $\{J^k\}$ with $J^k \in S(k)$ for each k , converges pointwise to J^* . Moreover, we have

$$TJ \in S(k+1), \quad \forall J \in S(k), \quad k = 0, 1, \dots$$

- (2) **Box Condition:** For all k , $S(k)$ is a Cartesian product of the form

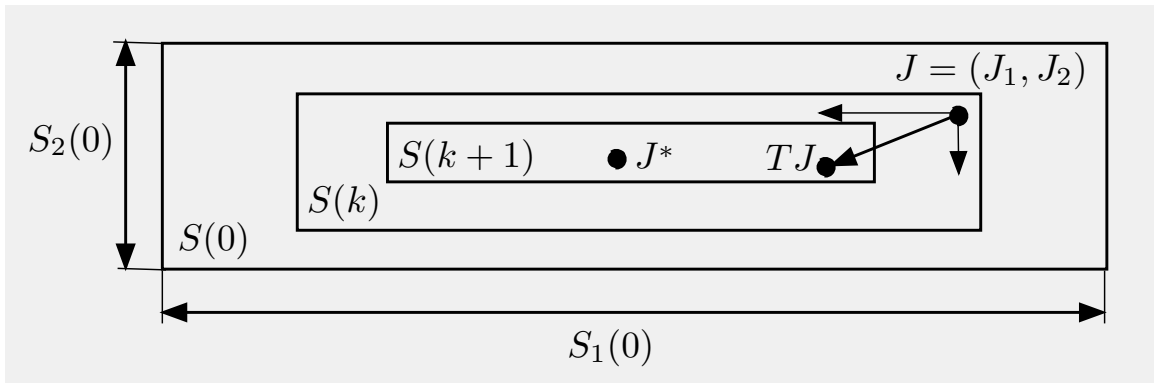
$$S(k) = S_1(k) \times \cdots \times S_m(k),$$

where $S_\ell(k)$ is a set of real-valued functions on X_ℓ , $\ell = 1, \dots, m$.

Then for every $J \in S(0)$, the sequence $\{J^t\}$ generated by the asynchronous algorithm converges pointwise to J^* .

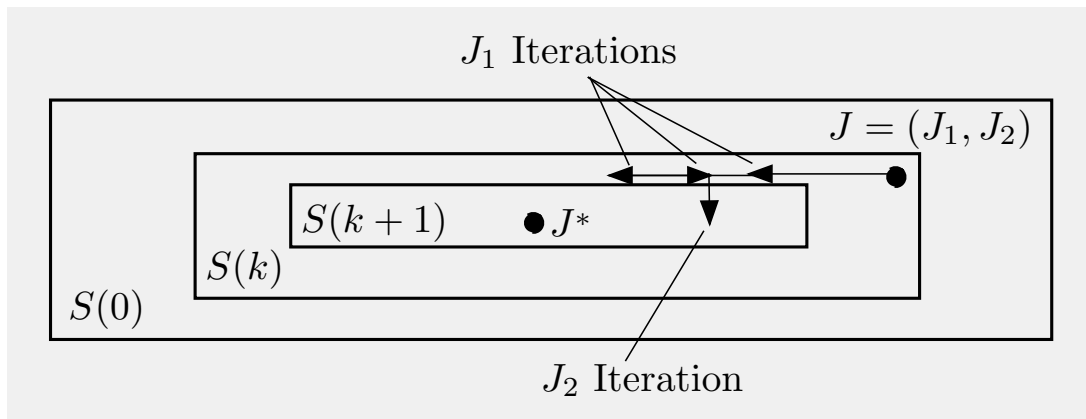
ASYNCHRONOUS CONV. THEOREM II

- Interpretation of assumptions:



A synchronous iteration from any J in $S(k)$ moves into $S(k + 1)$ (component-by-component)

- Convergence mechanism:



Key: “Independent” component-wise improvement. An asynchronous component iteration from any J in $S(k)$ moves into the corresponding component portion of $S(k + 1)$