ε.δε.μ²
ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ & ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΕΡΓΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Ενισχυτική Μάθηση με Προσεγγιστικές Μεθόδους:

1. Μάθηση Χρονικών Διαφορών (Temporal-Difference Learning)

2. Στοχαστικός Αλγόριθμος Q-Learning

3. Κατανεμημένη Υλοποίηση Ενισχυτικής Μάθησης

4. Αλγόριθμος Bellman-Ford, Δρομολόγηση BGP στο Internet

καθ. Βασίλης Μάγκλαρης

maglaris@netmode.ntua.gr

www.netmode.ntua.gr

Αίθουσα 02, Νέα Κτίρια ΣΗΜΜΥ

Τρίτη 10/5/2022

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Αλγόριθμος Policy Iteration (1/2) (*Επανάληψη*)

Ορισμός Q-factor

Έστω χρονοσταθερή πολιτική $\pi = \{\mu, \mu, \dots\}$ που οδηγεί σε αναμενόμενα ***costs-to-go*** $J^{\mu}(i), \forall i \in \mathcal{X}$ (καταστάσεις του ***περιβάλλοντος***) με αποφάσεις του ***agent*** $a = \mu(i) \in \mathcal{A}_i$

Για κάθε ζεύγος (i, a) στο υπό εξέταση βήμα και πολιτική για τα υπολειπόμενα βήματα $\pi = \{\mu, \mu, \dots\}$ ορίζω τους ***Q-factors*** $Q^{\mu}(i, a)$ σαν μέτρο κατάταξης εναλλακτικών άμεσων αποφάσεων $a \in \mathcal{A}_i$ του ***agent*** που θα οδηγούσαν το ***περιβάλλον*** από παρούσα κατάσταση i σε κατάσταση j με αναμενόμενα υπολειπόμενα ***costs-to-go*** $J^{\mu}(j), \forall j \in \mathcal{X}$

$$Q^{\mu}(i, a) \triangleq c(i, a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, J^{\mu}(j)$$

Μια πολιτική $\pi = \{\mu, \mu, \dots\}$ ικανοποιεί τις συνθήκες απληστίας (***greedy conditions***) σε σχέση με τα αναμενόμενα ***costs-to-go*** $J^{\mu}(j)$ στα υπολειπόμενα βήματα όταν σε κάθε βήμα και $\forall i \in \mathcal{X}$ ο ***agent*** επιλέγει $a = \mu(i)$ ώστε

$$Q^{\mu}(i, \mu(i)) = \min_{a \in \mathcal{A}_i} Q^{\mu}(i, a), \forall i \in \mathcal{X}$$

Μια πολιτική $\pi^* = \{\mu^*, \mu^*, \dots\}$ είναι βέλτιστη για όλα τα βήματα αν ικανοποιεί τις συνθήκες απληστίας (***greedy conditions***) του δυναμικού προγραμματισμού:

$$Q^{\mu^*}(i, \mu^*(i)) = \min_{a \in \mathcal{A}_i} Q^{\mu^*}(i, a)$$

Σημείωση: Όταν τα άμεσα αναμενόμενα κόστη $c(i, a)$ αντικαθίστανται από ***rewards*** $r(i, a)$, τα ***costs-to-go*** $J^{\mu}(i)$ αποκαλούνται ***Value Functions*** $V^{\mu}(i)$ και έχουμε κατ' αντιστοιχία:

$$Q^{\mu}(i, a) \triangleq r(i, a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, V^{\mu}(j) \text{ και } Q^{\mu^*}(i, \mu^*(i)) = \max_{a \in \mathcal{A}_i} Q^{\mu^*}(i, a)$$

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Αλγόριθμος Policy Iteration (2/2) (*Επανάληψη*)

Αρχιτεκτονική *Actor – Critic*

(***A.G. Barto***, ***R.S. Sutton*** & ***C.W. Anderson***, "Neuronlike adaptive elements that can solve difficult learning control problems," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, Sept. – Oct. 1983)

Επαναλήψεις $n = 0,1,2, \dots$ από δύο βήματα μέχρι $\mu_{n+1}(i) = \mu_n(i),\ J^{\mu_{n+1}}(i) = J^{\mu_n}(i)\ , \forall i$

Βήμα 1. **Policy Evaluation** (ο ***critic*** αναλύει τις αποφάσεις του ***agent***):

Με βάση την παρούσα πολιτική $\pi_n = \{\mu_n, \mu_n, \dots\}$ υπολογίζονται τα ***costs-to-go***

$J^{\mu_n}(i) = c\big(i, \mu_n(i)\big) + \gamma \sum_{j=1}^{N} p_{ij}(\mu_n(i))\, J^{\mu_n}(j)$ για $\forall i$

Για $\forall i$ και $\forall a \in \mathcal{A}_i$ υπολογίζονται τα ***Q-factors***: $Q^{\mu_n}(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, J^{\mu_n}(j)$

Βήμα 2. **Policy Improvement** (ο ***actor*** καθοδηγεί τις αποφάσεις του ***agent***):

Η πολιτική π_n βελτιώνεται σε π_{n+1} μέσω της $\mu_{n+1}(i) = \arg\min_{a \in \mathcal{A}_i} Q^{\mu_n}(i,a)$ για $i = 1,2, \dots, N$

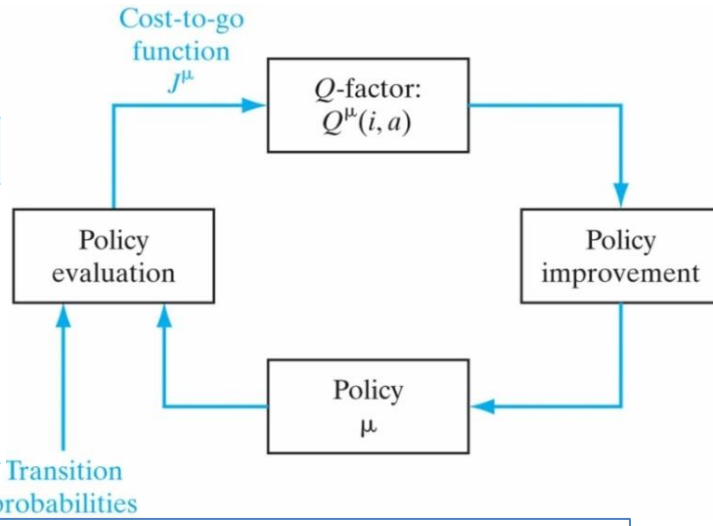
$\arg\min_x f(x)$: Η τιμή της x που οδηγεί την $f(x)$ σε ελάχιστο

TABLE 12.1 Summary of the Policy Iteration Algorithm

1. Start with an arbitrary initial policy μ_0.
2. For $n = 0, 1, 2, \dots$, compute $J^{\mu_n}(i)$ and $Q^{\mu_n}(i, a)$ for all states $i \in \mathcal{X}$ and actions $a \in \mathcal{A}_i$.
3. For each state i, compute

$$\mu_{n+1}(i) = \arg\min_{a \in \mathcal{A}_i} Q^{\mu_n}(i, a)$$

4. Repeat steps 2 and 3 until μ_{n+1} is not an improvement on μ_n, at which point the algorithm terminates with μ_n as the desired policy.



Ο αλγόριθμος συγκλίνει σε βέλτιστη πολιτική σε πεπερασμένα βήματα n λόγω πεπερασμένου πλήθους καταστάσεων N και πεπερασμένων επιλογών αποφάσεων

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Value Iteration Algorithm (*Επανάληψη*)

Εκτίμηση των Συναρτήσεων Cost-to-Go μέσω Διαδοχικών Προσεγγίσεων $J_n(i) \to J_{n+1}(i)$

- Εκκίνηση με αυθαίρετες τιμές $J_0(i) \ \ \forall i$
- Επαναλήψεις $n \to n+1$ μέχρι ***ανεκτή σύγκλιση*** (θεωρητικά $n \to \infty$) μέσω σχέσεων ***backup***:

  $J_{n+1}(i) = \min_{a \in \mathcal{A}_i} \{ c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J_n(j) \}$ για $i = 1,2,\dots,N$ (από εξισώσεις ***Bellman***)
- Τελικός υπολογισμός των βέλτιστων ***Costs-to-Go***

  $$J^*(i) = \lim_{n \to \infty} J_n(i), \ Q^*(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J^*(j)$$

  και προσδιορισμός της ***βέλτιστης πολιτικής*** $\mu^*(i) = \arg\min_{a \in \mathcal{A}_i} Q^*(i,a)$ για $i = 1,2,\dots,N$

TABLE 12.2 Summary of the Value Iteration Algorithm

1. Start with arbitrary initial value $J_0(i)$ for state $i = 1, 2, \dots, N$.
2. For $n = 0, 1, 2, \dots$, compute

$$J_{n+1}(i) = \min_{a \in \mathcal{A}_i} \left\{ c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J_n(j), \right\}, \qquad \begin{matrix} a \in \mathcal{A}_i \\ i = 1, 2, \dots, N \end{matrix}$$

Continue this computation until

$$|J_{n+1}(i) - J_n(i)| < \epsilon \qquad \text{for each state } i$$

where ϵ is a prescribed tolerance parameter. It is presumed that ϵ is sufficiently small for $J_n(i)$ to be close enough to the optimal cost-to-go function $J^*(i)$. We may then set

$$J_n(i) = J^*(i) \qquad \text{for all states } i$$

3. Compute the Q-factor

$$Q^*(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J^*(j) \qquad \begin{matrix} \text{for } a \in \mathcal{A}_i \text{ and} \\ i = 1, 2, \dots, N \end{matrix}$$

Hence, determine the optimal policy as a greedy policy for $J^*(i)$:

$$\mu^*(i) = \arg\min_{a \in \mathcal{A}_i} Q^*(i,a)$$

Ο αλγόριθμος ***Value Iteration*** αν συγκλίνει σε ικανοποιητικό χρόνο, αποφεύγει υπολογισμούς ***Q-factors*** και ενδιάμεση ανανέωση πολιτικής σε κάθε βήμα όπως ο ***Policy Iteration***

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράδειγμα Δυναμικού Προγραμματισμού: Βελτιστοποίηση Δρομολόγησης (*Επανάληψη*)

Εύρεση Δρόμων Ελάχιστου Κόστους από Κόμβο A *σε* Κόμβο J μέσω του μονο-κατευθυντικού γράφου όπως στο σχήμα με κατεύθυνση γραμμών ***Α → Δ***

Ενδεικτικό κόστος γραμμών: $A \to B: 2,\ B \to A: \infty$

$B \to F: 4,\ F \to B: \infty$

Ενδεικτικό κόστος δρόμου: Δρόμος $\{A, B, F, I, J\}$: $2 + 4 + 3 + 4 = 13$

Κατάσταση Περιβάλλοντος: Κόμβος σε παρούσα διερεύνηση $\{A, B, \dots, J\}$

Αποφάσεις Agent: Επόμενος κόμβος για διερεύνηση $\{up, down, staight\}$

Αναδρομικός Υπολογισμός *Q-Factors*:

$Q(H, down) = 3, \qquad Q(I, up) = 4$

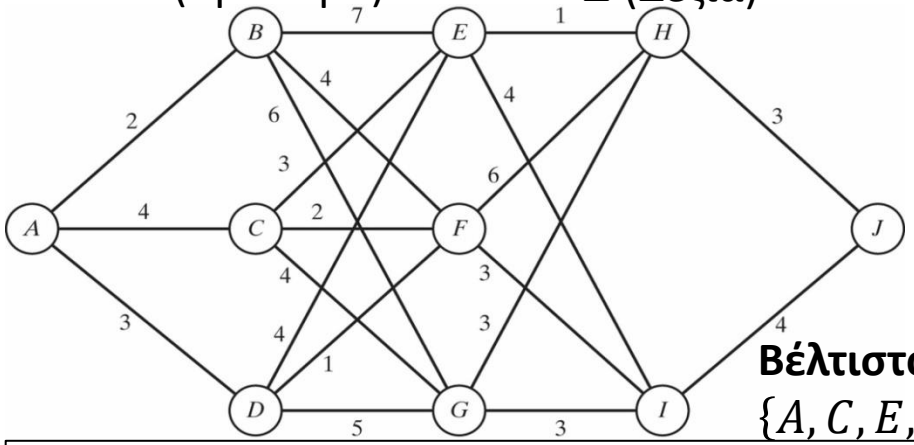
$Q(E, staight) = 1 + 3 = 4, \qquad Q(E, down) = 4 + 4 = 8$

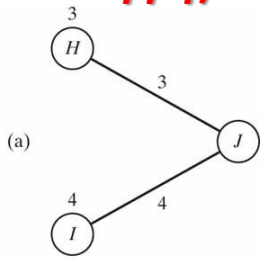
$Q(F, up) = 6 + 3 = 9, \qquad Q(F, down) = 3 + 4 = 7$

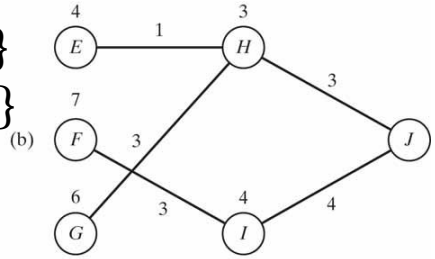
.....

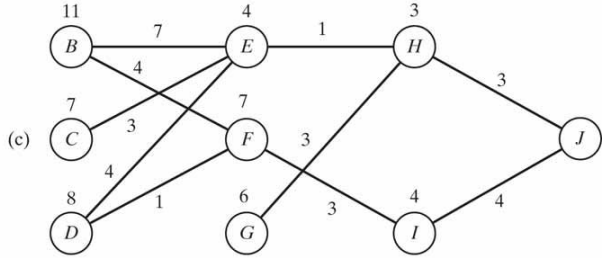
Κατεύθυνση Γραμμών

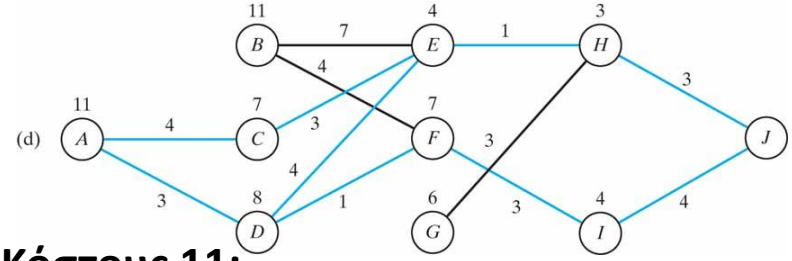
Α (Αριστερά) ⟶ ***Δ*** (Δεξιά)











Βέλτιστοι Δρόμοι Κόστους 11:

$\{A, C, E, H, J\}, \{A, D, E, H, J\}, \{A, D, F, I, J\}$

Αλγόριθμοι Δυναμικού Προγραμματισμού ***Bellman-Ford*** στηρίζουν την δρομολόγηση ***Border Gateway Protocols*** (***BGP***) ανάμεσα στα ~78,000 Αυτόνομα Συστήματα (***Autonomous Systems, AS***) στο ***Internet*** (~900,000 γνωστά δίκτυα)

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Σύνοψη Εννοιών Δυναμικού Προγραμματισμού (1/2)

Βασικές Παράμετροι Δυναμικού Προγραμματισμού - Ελαχιστοποίηση Κόστους

Άμεσο Κόστος (***Observed Cost***) βήματος μετάβασης $i \to j$ με απόφαση a: $g(i,a,j)$

Άμεσο Αναμενόμενο Κόστος (***Immediate Expected Cost***) κατάστασης i, απόφασης a:

$$c(i,a) \triangleq \sum_{j=1}^{N} p_{ij} g(i,a,j)$$

Ορισμός ***Cost-to-Go*** : $J^{\mu}(i) = c(i,\mu(i)) + \gamma \sum_{j=1}^{N} p_{ij}(\mu(i))\, J^{\mu}(j)$ για $\forall i$ και πολιτική $\mu(i)$

Βέλτιστα ***Cost-to-Go*** (***Bellman***): $J^{*}(i) = \min_{a \in \mathcal{A}_i} \left(c(i,a) + \gamma \sum_{j=1}^{N} p_{ij} J^{*}(j) \right), i = 1,2,\dots,N$

Ορισμός ***Q-Factors***: $Q^{\mu}(i,a) \triangleq c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, J^{\mu}(j)$ για $\forall i$ και $\forall a \in \mathcal{A}_i$

Βασικές Παράμετροι Δυναμικού Προγραμματισμού - Μεγιστοποίηση Οφέλους

Άμεση Ανταμοιβή (***Observed Reward***) βήματος μετάβασης $i \to j$ με απόφαση a: $R(i,a,j)$

Άμεση Αναμενόμενη Ανταμοιβή (***Immediate Expected Reward***) κατάστασης i, απόφασης a:

$$r(i,a) \triangleq \sum_{j=1}^{N} p_{ij} R(i,a,j)$$

Ορισμός ***Value Function*** : $V^{\mu}(i) = r(i,\mu(i)) + \gamma \sum_{j=1}^{N} p_{ij}(\mu(i))\, V^{\mu}(j)$ για $\forall i$ και πολιτική $\mu(i)$

Βέλτιστα ***Values*** (***Bellman***): $V^{*}(i) = \max_{a \in \mathcal{A}_i} \left(r(i,a) + \gamma \sum_{j=1}^{N} p_{ij} V^{*}(j) \right), i = 1,2,\dots,N$

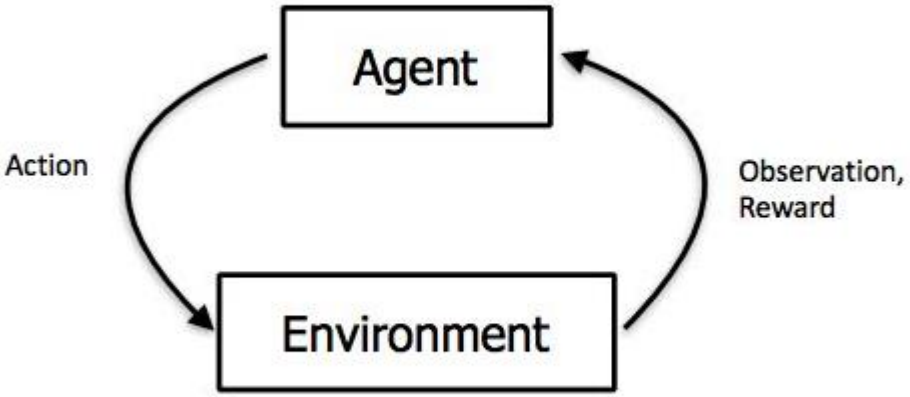
Ορισμός ***Q-Factors***: $Q^{\mu}(i,a) \triangleq r(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a)\, V^{\mu}(j)$ για $\forall i$ και $\forall a \in \mathcal{A}_i$

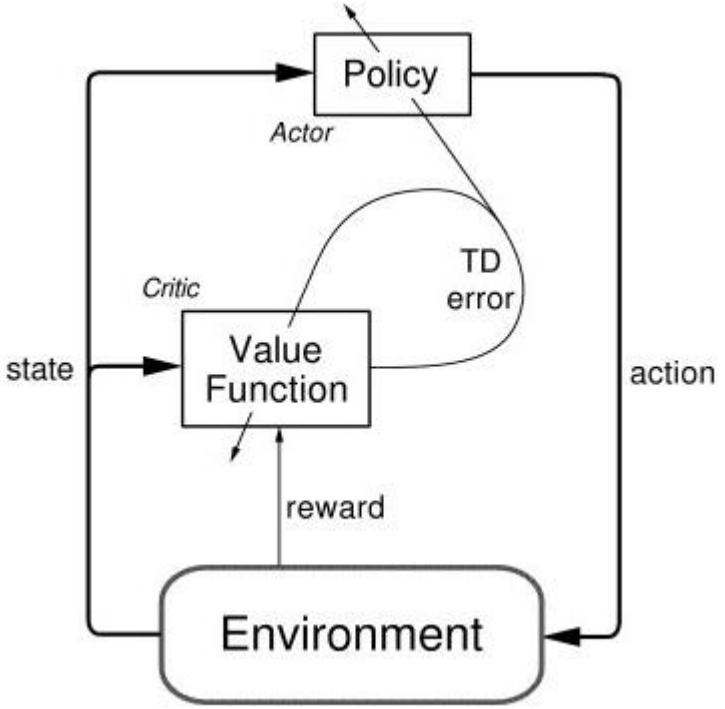
ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Σύνοψη Εννοιών Δυναμικού Προγραμματισμού (2/2)

Μοντέλο Actor – Critic

(***A.G. Barto***, ***R.S. Sutton*** & ***C.W. Anderson***, "Neuronlike adaptive elements that can solve difficult learning control problems," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, Sept. – Oct. 1983)





Policy Iteration Actor-Critic Model

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Απευθείας Προσεγγιστικές Μέθοδοι Δυναμικού Προγραμματισμού (1/3)

Οι δύο αλγόριθμοι Δυναμικού Προγραμματισμού (***Value Iteration*** & ***Policy Iteration***) προαπαιτούν γνώση των **πιθανοτήτων μεταβάσεων** $p_{ij}(a)$ και του ***άμεσα αναμενόμενου κόστους*** κατάστασης i και απόφασης a

$$c(i,a) = \sum_{j=1}^{N} p_{ij}(a) g(i,a,j)$$

εκτιμώμενου με βάση τα **γνωστά** ***observed κόστη μετάβασης*** $i \to j$ που καθορίζεται από μια πολιτική $a = \mu(i)$

$$g(i,a,j) = g(i,\mu(i),j) \triangleq g(i,j)$$

Οι απευθείας προσεγγιστικές μέθοδοι (***Direct Approximate Dynamic Programming Methods***) εκτιμούν τις πιθανότητες μετάβασης και επομένως τα αναμενόμενα κόστη καταστάσεων - αποφάσεων $c(i,a)$ πολιτικών $a = \mu(i)$

Ενσωματώνονται στους δύο βασικούς αλγόριθμους Δυναμικού Προγραμματισμού με τις εξής παραλλαγές:

- **Value Iteration** → **Temporal-Difference *TD*(0) Learning**
- **Policy Iteration** → ***Q*-Learning**

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Απευθείας Προσεγγιστικές Μέθοδοι Δυναμικού Προγραμματισμού (2/3)

Γενική Μεθοδολογία - Απαιτήσεις

- Οι προσομοιώσεις ***Monte Carlo*** δημιουργούν σενάρια πολλαπλών πιθανών τροχιών (***system trajectories***) της εξέλιξης του ***Markov Decision Process*** σε κάθε **επεισόδιο** (***episode***) από μια αρχική κατάσταση i_0 μέχρι κάποια τελική $i_n \rightarrow i_T$ (T - *Terminal* είναι το βήμα n όπου τερματίζεται το **επεισόδιο**). Η διαδικασία μάθησης συνήθως περιλαμβάνει πολλά ανεξάρτητα **επεισόδια** με διαφορετικές ***trajectories***

- Οι τιμές συναρτήσεων ***cost-to-go*** $J^{\mu}(i)$ανανεώνονται σε κάθε προσομοίωση με προσθήκη του (**γνωστού**) ***άμεσου*** (***observed***) ***κόστους μετάβασης*** $g(i,j)$ σε επισκέψεις προσομοιωμένης τροχιάς μεταβάσεων από κατάσταση i προς κατάσταση j

- Οι μέθοδοι ***Monte Carlo*** απαιτούν γνώση της δομής του περιβάλλοντος από εμπειρία (όχι από πρότερη γνώση πιθανοτήτων), διαχειρήσιμο αριθμό παρατηρήσιμων (***observable***) καταστάσεων και σημαντικό αριθμό από ***trajectories*** για καλές εκτιμήσεις

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Απευθείας Προσεγγιστικές Μέθοδοι Δυναμικού Προγραμματισμού (3/3)

Ορισμοί *on-policy*, *off-policy*

- Η ***on-policy*** σε κάθε βήμα εκτιμά με προσομοιώσεις ***Monte Carlo*** το κόστος $J^{\mu}(i)$ των καταστάσεων i μιας τροχιάς (***trajectory***) όταν ακολουθείται η υπό αξιολόγηση **συνολική** πολιτική μ. Με επαναλήψεις τροχιών που περιλαμβάνουν διορθωτικές αποφάσεις $i \rightarrow \alpha$ οδηγούνται τα $J^{\mu}(i)$ σε διαδοχικές μειώσεις: **Value Iteration → *TD*(0)-Learning**

- Η ***off-policy*** συγκρίνει εναλλακτικές αποφάσεις σε καταστάσεις του περιβάλλοντος i μιας τροχιάς (***trajectory***) και σε κάθε βήμα **επιλέγει** με απληστία αποφάσεις a με το ελάχιστο $Q(i,a)$ στην παρούσα κατάσταση i. Τα ***Cost-to-Go*** $J^{\mu}(j)$ μιας προσωρινής πολιτικής μ εκτιμώνται από προσομοιώσεις ***Monte Carlo*** των ***trajectories*** χωρίς να συμπεριλαμβάνουν βελτιώσεις $i \rightarrow \alpha$ που ίσως προκύψουν από τα ***Q-Factors***: **Policy Iteration → *Q*-Learning**

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Προσεγγιστικός Αλγόριθμος *TD*(0) Learning (1/2)

Value Iteration → Temporal-Difference *TD*(0) Learning

Εξισώσεις ***Bellman*** υπολογισμού ***costs-to-go*** από i_n στο βήμα $n < T$ με τελική κατάσταση i_T:

$$J^{\mu}(i_n) = \mathrm{E}[g(i_n, i_{n+1}) + \gamma J^{\mu}(i_{n+1})] = \mathrm{E}\left[\sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1})\right], n = 0,1, \dots, T-1$$

Με επανειλημμένες προσομοιώσεις ***Monte Carlo*** δημιουργούμε M ***trajectories*** (τροχιές) του συστήματος και προσεγγίζουμε τα ***Costs-to-Go*** $J^{\mu}(i_n)$ μέσω των συναρτήσεων $J(i_n)$. Τα $J(i_n)$ εκτιμώνται σαν ***ensemble averages*** των $c(i_n) \triangleq \sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1})$, του κόστους των M υπολειπόμενων τροχιών $\{i_n, i_{n+1}, \dots, i_T\}$ από την (i_n) προς τελική κατάσταση $i_T = 0$:

$$J^{\mu}(i_n) = \mathrm{E}[c(i_n)] \cong J(i_n) = \frac{1}{M}\sum_{M} c(i_n)$$

Τα κόστη $J(i_n)$ εκτιμώνται μέσω ***Robbins-Monro Successive Approximations*** που διορθώνουν εκτιμήσεις τιμών τους (***updates***) σε κάθε επίσκεψη της κατάστασης i_n με συντελεστή μάθησης (***learning rate***) η_n:

$$J(i_n) \coloneqq J(i_n) + \eta_n[g(i_n, i_{n+1}) + \gamma J(i_{n+1}) - J(i_n)] = J(i_n) + \eta_n d_n$$

Το σφάλμα $d_n \triangleq g(i_n, i_{n+1}) + \gamma J(i_{n+1}) - J(i_n), n = 0,1, \dots, T-1$ ονομάζεται χρονική διαφορά (***Temporal Difference***, ***TD***) στο βήμα n μίας ***trajectory*** και οδηγεί τα $J(i_n)$ προς τη ***σύγκλιση*** σε ελάχιστες τιμές με επαναλήψεις διαφορετικών ***ανεξάρτητων*** τροχιών που προκύπτουν από την εφαρμογή μιας πολιτικής μ

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Προσεγγιστικός Αλγόριθμος *TD*(0) Learning (2/2)

Value Iteration → Temporal-Difference *TD*(0) Learning

Εναλλακτικός αλγόριθμος ***update*** προκύπτει από την μακρόχρονη επαναληπτική σχέση:

$$J(i_n) := J(i_n) + \eta_n \left(\sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1}) - J(i_n) \right) = J(i_n) + \eta_n \sum_{k=0}^{T-n-1} \gamma^k d_{n+k}$$

Τα ***costs-to-go*** εκτιμώνται σαν μέσοι όροι (***ensemble averages***) σε μεγάλο αριθμό M επαναλήψεων προσομοιώσεων με πολλαπλές επισκέψεις καταστάσεων i_n στο βήμα n κάποιου ***trajectory***.

$$J^{\mu}(i_n) = \mathrm{E}\left[\sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1}) \right] = \mathrm{E}[c(i_n)] \cong J(i_n) = \frac{1}{M} \sum_{M} c(i_n)$$

όπου

$$c(i_n) \triangleq \sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1})$$

Οι συναρτήσεις $J(i_n)$υπολογίζονται με επαναλαμβανόμενες επισκέψεις της i_n σε τροχιά που παράγεται από προσομοίωση ***Monte Carlo***

$$J(i_n) := J(i_n) + \eta_n (c(i_n) - J(i_n))$$

με αρχικές συνθήκες $J(i_n) = 0$ και ***learning rate*** $\eta_n = {}^{1}/_{n}$, $n = 1,2,\dots,T$

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Προσεγγιστικός Αλγόριθμος *Q*-Learning (1/2)

Policy Iteration → *Q*-Learning

- Προσδιορισμός πολιτικής βέλτιστης συμπεριφοράς (***off-policy behavior generation***) μέσω δημιουργίας *πολλαπλών* ***trajectories*** (τροχιών) για δυνατά σενάρια αποφάσεων: ***Q*-Learning**

- Ορίζουμε $s_n \triangleq (i_n, a_n, j_n, g_n)$ στο βήμα n μιας ***trajectory*** όταν η κατάσταση του περιβάλλοντος οδηγείται σε μετάβαση $i_n \rightarrow i_{n+1} = j_n$ με απόφαση του agent a_n και observed κόστος μετάβασης $g_n = g(i_n, a_n, j_n)$

- Με βάση την καταγραφή των s_n σε προσομοιωμένες ***trajectories*** ο αλγόριθμος ***Q*-Learning** οδηγεί το σύστημα στη μάθηση βέλτιστης πολιτικής κατά προσέγγιση του ***policy iteration***

- **Προϋπόθεση**: Η i_n που προκύπτει σε μια ***trajectory*** πρέπει να είναι ***fully observable***

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Προσεγγιστικός Αλγόριθμος *Q*-Learning (2/2)

Αλγόριθμος Υπολογισμού $Q^*(i,a)$ με Successive Approximations (*Robins-Monro*)

$$Q(i,a) := (1-\eta)Q(i,a) + \eta \sum_{j=1}^{N} p_{ij}(a)\left[g(i,a,j) + \gamma \min_{b\in\mathcal{A}_j} Q(j,b)\right] \text{ για } \forall(i,a)$$

Από το όριο $Q^*(i,a)$ των επαναλήψεων προσδιορίζεται ο πίνακας βέλτιστης πολιτικής π με αντιστοίχηση

$$\mu^*(i) = \arg\min_{a\in\mathcal{A}_i} Q^*(i,a) \text{ για } i = 1,2,\dots,N$$

Στοχαστική Παραλλαγή

Έστω ότι η προσομοίωση ***Monte Carlo*** ορίζει τροχιά (***trajectory***) από αρχική κατάσταση i_0 μέχρι την i_n στο παρόν βήμα n με άπληστες επιλογές (i_n, a_n) που προσδιορίζουν τα ***costs-to-go*** $J_n(j_n)$ για $i_n \to j_n$. Ο επαναληπτικός αλγόριθμος ανανεώνει τους ***Q-factors*** από $Q_n(i,a)$ σε $Q_{n+1}(i,a)$ για προσομοίωση **επεισοδίου** $n = 0,1,\dots,T$ ως εξής:

- $Q_{n+1}(i,a) = (1-\eta_n)Q_n(i,a) + \eta_n[g(i,a,j) + \gamma J_n(j)]$ για $(i,a) = (i_n, a_n)$

  όπου $J_n(j) = \min_{b\in\mathcal{A}_j} Q_n(j,b)$ και j επόμενη κατάσταση της $i = i_n$ με τα παρόντα ***Q-factors***
- $Q_{n+1}(i,a) = Q_n(i,a)$ για όλα τα υπόλοιπα ζεύγη $(i,a) \neq (i_n, a_n)$
- Με την πρόοδο των επαναλήψεων $Q_n(i,a) \to Q^*(i,a)$, τις βέλτιστες τιμές των ***Q-factors***
- Η ***learning parameter*** η_n είναι φθίνουσα ως προς n, π.χ. $\eta_n = \alpha/(\beta+n)$ με α, β θετικά

Επειδή μια τροχιά με ***greedy*** αποφάσεις (***exploitation***) μπορεί να αγνοήσει άλλες επιλογές λόγω εκκίνησης από μια κατάσταση, μπορεί να απαιτείται προσομοίωση πολλαπλών **επεισοδίων** για επισκέψεις σε ευρύ φάσμα καταστάσεων (***exploration***). Το εύρος της αναζήτησης ενισχύεται με απόφαση ***greedy*** με πιθανότητα $(\mathbf{1}-\boldsymbol{\epsilon})$ ή άλλης με πιθανότητα $\boldsymbol{\epsilon}$

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Κατανεμημένη Υλοποίηση Ενισχυτικής Μάθησης

Μοντέλο Συνεργατικής Βελτιστοποίησης μέσω Πολλαπλών Αυτόνομων Agents

Η κατανεμημένη συνεργατική βελτιστοποίηση κωδικοποιήθηκε σαν ***Multi-Agent Reinforcement Learning – MARL*** από τον ***Michael Littman*** το 1994
https://www2.cs.duke.edu/courses/spring07/cps296.3/littman94markov.pdf

- Επέκταση του Δυναμικού Προγραμματισμού με συνεργασία (***cooperative zero-sum game***) 2+ αυτόνομων ***agents***
- Κάθε agent παίρνει αποφάσεις προς βέλτιστες πολιτικές που επηρεάζονται από τις πολιτικές των αυτόνομων συνεργατών του σύμφωνα με μοντέλο ***Markov*** (***Stochastic***) ***Game***
- Κατανεμημένη υλοποίηση αλγορίθμου ***Q-Learning*** με ***ασύγχρονα updates*** μεταξύ των ***agents***
- Ορισμός των ***Q-factors*** σαν ***minimax Q-factors*** ώστε να εξαρτώνται και από τις αποφάσεις των συνεργαζομένων ***agents***.
- Ο υπολογισμός των ***minimax Q-factors*** μπορεί να γίνεται με επαναληπτική εφαρμογή ***Linear Programs*** αλλά με σημαντική υπολογιστική επιβάρυνση. Πρακτικά και για συγκεκριμένες εφαρμογές μπορεί να είναι εξαιρετικά απλός ή να αντιμετωπιστεί με γρήγορους ευριστικούς αλγορίθμους

Εφαρμογή μεγάλης κλίμακας (**78,000** ***agents/routers***) στο ***Border Gateway Protocol*** (***BGP***) για δρομολόγηση προς τα **900,000** ***γνωστά δίκτυα*** του παγκόσμιου ***Internet***

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράδειγμα Δυναμικού Προγραμματισμού: Δρομολόγηση BGP στο Internet - RFC 4271 (1/7)

Υλικό από Παρουσιάσεις Μαθήματος Διαχείριση Δικτύων – Ευφυή Δίκτυα ΣΗΜΜΥ Ε.Μ.Π.

https://webvm.netmode.ntua.gr/courses/wp-content/uploads/2019/10/NetMan_2019_10_14.pdf

Το παγκόσμιο ***Internet*** αποτελείται (*6/2021*) από $\sim \mathbf{900,000}$ γνωστά δίκτυα τελικούς προορισμούς (π.χ. Δίκτυο ΕΜΠ, IP: 147.102.0.0/16), οργανωμένα σε $\sim \mathbf{78,000}$ Αυτόνομα Συστήματα (***Autonomous Systems, AS***) με διαχειριστική αυτονομία (π.χ. GRNET/ΕΔΙΤΕ, Autonomous System Number - ***ASN 5408***)

Η δρομολόγηση εντός Αυτόνομης Κοινότητας γίνεται με βάση κεντρικά ρυθμιζόμενα πρωτόκολλα (***Interior Gateway Protocols*** – ***IGP***, π.χ. OSPF) ενώ μεταξύ των $\mathbf{70,000}$ AS's μέσω γενικών πινάκων δρομολόγησης σε συνοριακούς δρομολογητές (***Border Gateways***, ***Border Routers***) με καταχωρήσεις για όλα τα $\sim \mathbf{880,000}$ γνωστά δίκτυα του ***Internet***

Η δημιουργία – ανανέωση των γενικών πινάκων δρομολόγησης (***σε ηλεκτρονική μνήμη των Border Gateways***) ***γίνεται με το Border Gateway Protocol – BGP*** **(RFC 4271)**

- Οι ***Border Routers*** (***Gateways***) των ***AS*** ανακοινώνουν (μέσω ***BGP signaling***) στα $\mathbf{78,000}$ ***AS***'s του ***Internet*** τα $\mathbf{900,000}$ δίκτυα – τελικούς προορισμούς τα οποία είτε ανήκουν σε αυτά ή είναι προσπελάσιμα (***reachable***) διαμέσου αυτών, με εκτιμήσεις κόστους (βάρους) βέλτιστων ***inter-AS*** δρόμων προς κάθε δίκτυο - προορισμό
- Οι ***Border Gateways*** υπολογίζουν αυτόνομα βέλτιστες διαδρομές προς όλους τους τελικούς προορισμούς με βάση τις προτιμήσεις (πολιτικές) των διαχειριστών τους, όποτε κρίνουν πως αλλαγές τοπολογίας ή πολιτικής ή επίδοσης επιβάλλουν ανανέωση δρόμων
- Ο κατανεμημένος προσδιορισμός βέλτιστης δρομολόγησης ορίζει κόστη προς τους $\mathbf{900,000}$ τελικούς προορισμούς βάση πληροφοριών ***reachability*** και μετρήσεων κόστους διασύνδεσης προς τα γειτονικά ***AS***. Βασίζεται στον Αλγόριθμο ***Bellman – Ford***

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράδειγμα Δυναμικού Προγραμματισμού: Δρομολόγηση BGP στο Internet - RFC 4271 (2/7)

Αλγόριθμος Distance Vector (Bellman – Ford)
BGP (Bellman – Ford)

- Οι συνοριακοί δρομολογητές (***Border Gateways***) κάθε Αυτόνομης Περιοχής (***AS***) εντοπίζουν τους βέλτιστους δρόμους (***shortest paths***) ενδιάμεσων και τελικού ***AS*** προς όλα τα γνωστά δίκτυα προορισμούς εκτελώντας αλγόριθμο βασισμένο στον δυναμικό προγραμματισμό (***dynamic programming***) που εισήγαγε ο ***Bellman***

- Χρειάζεται γνώση διανυσμάτων κόστους (βαρών) των άμεσων συνδέσεων (***Inter AS Interfaces***) και εκτιμήσεις κόστους (αποστάσεις, ***distance vectors***) προς όλα τα γνωστά δίκτυα προορισμούς στο ***Internet*** (**900,000+**, *6/2021*)

- Η βελτιστοποίηση βασίζεται σε κατανεμημένο αλγόριθμο ***Bellman - Ford*** που υλοποιείται μέσω σηματοδοσίας ανακοινώσεων (***BGP Announcements***) μεταξύ όλων των (**78,000+**, *6/2021*) Αυτόνομων Περιοχών (***AS***) του ***Internet*** με πληροφορίες δρομολόγησης και εκτιμήσεις κόστους

- Από τη σκοπιά του ***Reinforcement Learning*** το ***BGP*** μπορεί να θεωρηθεί κατανεμημένη επέκταση του Δυναμικού Προγραμματισμού με συνεργασία (***cooperative game***) **78,000** αυτόνομων ***Agents***

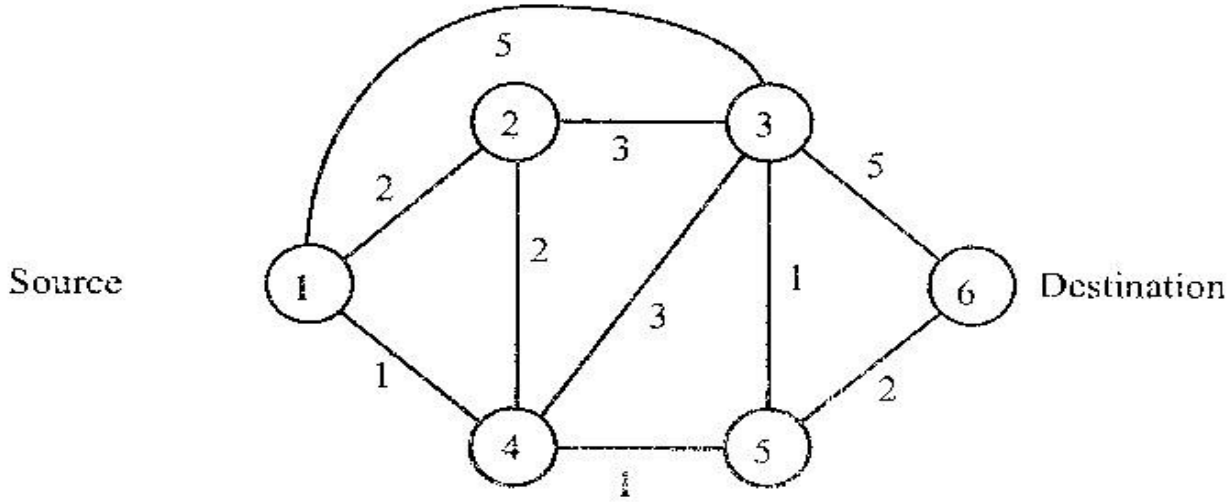
Το BGP αποτελεί κύριο παράγοντα επιτυχίας της παγκόσμιας επανάστασης του *Internet*

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράδειγμα Δυναμικού Προγραμματισμού: Δρομολόγηση BGP στο Internet - RFC 4271 (3/7)

Δίκτυο (Γράφος) Αναφοράς Παραδείγματος, $N = 6$ Κόμβων

- Οι κόμβοι του γράφου παριστούν τα διάφορα ***AS*** του ***Internet***
- Τα δίκτυα πηγής και προορισμού των χρηστών είναι ενσωματωμένα στους κόμβους (***AS***) ***Source*** – ***Destination*** του γράφου
- Τα κόστη των γραμμών του γράφου αφορούν και στις 2 κατευθύνσεις και εκτιμώνται από τους άμεσα συνδεόμενους κόμβους (***Border Gateways***) με βάση προτιμήσεις των διαχειριστών
- Στο παράδειγμα που ακολουθεί υπολογίζονται δένδρα ελαχίστων δρόμων (***shortest path trees***) από όλους τους κόμβους (***AS***) προς την ***ρίζα*** $\{6\}$
- Η επιλογή του ρόλου της ρίζας του δένδρου (πηγή ή προορισμός) έγινε αυθαίρετα. Οι αλγόριθμοι ισχύουν κατ' αναλογία για αντίστροφους ρόλους ρίζας



ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράδειγμα Δυναμικού Προγραμματισμού: Δρομολόγηση BGP στο Internet - RFC 4271 (4/7)

Υπολογισμός Δένδρου Ελάχιστων Δρόμων (Shortest Path Tree) προς $\{6\}$

Εφαρμογή Αλγορίθμου *Q*-Learning (*Off-policy*) με *Asynchronous Updates*

$\{i\}$ Κατάσταση (***State***) του γράφου, κόμβος (***AS***) $i = 1,2,\dots,N$ (στο παράδειγμα $N = 6$, μέχρι $80{,}000$ στο ***Internet***)

$P^{(n)}(i)$ Απόφαση (***Action***): Επόμενος κόμβος (***AS***) από τον $\{i\}$ προς τον $\{6\}$, ενδιάμεσος ή τελικός στην επανάληψη (***Iteration***) n

d_{ij} Κόστος (βάρος) γραμμής (i,j) στην επανάληψη n (***Transtion Cost***) ρυθμιζόμενο από την πολιτική δρομολόγησης του $\{i\}$ ή/και απευθείας μετρήσεις των αμέσων γειτόνων $\{i,j\}$. Αν $d_{ij} = c, \forall(i,j) \Rightarrow$ ***min hop routing***

$L^{(n)}(i)$ ***Labels***, ***Q-Factors*** $L^{(n)}(i) \triangleq Q(i, P^{(n)}(i))$: Εκτιμήσεις ελάχιστου κόστους από τον $\{i\}$ προς τον $\{6\}$ στην επανάληψη n (ανανεώνονται ***ασύγχρονα***, σύμφωνα με τις ***πιο πρόσφατες εκτιμήσεις*** ανάλογα με την σειρά εκτέλεσης των ανανεώσεων – ***updates***). Οι τροχιές (***trajectories***) αφορούν στις επιλογές δρόμων από τον $\{i\}$ προς τον $\{6\}$ σε κάθε επανάληψη

Περιγραφή Αλγορίθμου Bellman – Ford

- Αρχικά έχουμε $L_i^{(0)} = \infty \;\; \forall i \neq 6, \;\; L_6^{(n)} = 0 \;\; \forall n,$
- Σε κάθε διαδοχική επανάληψη (***iteration***) $n = 1,2,\dots$ και $\forall i$ ανανεώνουμε ***ασύγχρονα*** τις εκτιμήσεις ελαχίστου κόστους από την παρούσα κατάσταση προς τον προορισμό με βάση τις σχέσεις του Δυναμικού Προγραμματισμού σύμφωνα με τις πιο πρόσφατες εκτιμήσεις (***updates***) των $L_j^{(n)}$ για όλους τους άμεσους γείτονες j του i:

$$L_i^{(n+1)} = \min_j \left\{ L_j^{(n)} + d_{ij} \right\} \forall i \neq 6$$

- Αν $L_i^{(n+1)} = L_i^{(n)} \;\; \forall i$ σταματάμε τον αλγόριθμο και προσδιορίζουμε τους βέλτιστους δρόμους από όλα τα $\{i\}$ προς τον προορισμό $\{6\}$ σύμφωνα με τις αποφάσεις $P^{(n)}(i)$ σαν ***Shortest Path Tree*** με ρίζα τον $\{6\}$
- Πολυπλοκότητα αλγορίθμου: $O(N^3)$

ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράδειγμα Δυναμικού Προγραμματισμού: Δρομολόγηση BGP στο Internet - RFC 4271 (5/7)

Εκτέλεση Αλγορίθμου για Προορισμό {6}

Παράδειγμα: INITIAL LABELS: L(1)=L(2)=...=L(5)= ∞, L(6)=0

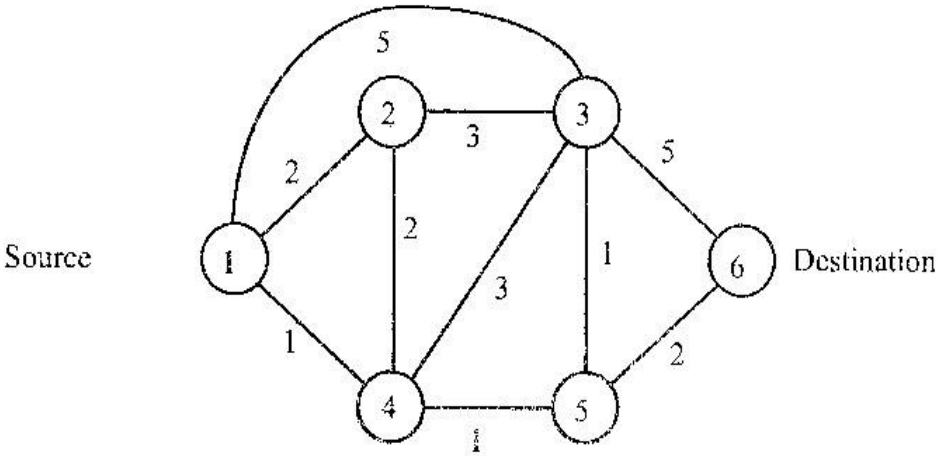
UPDATE ORDER 5,4,3,2,1

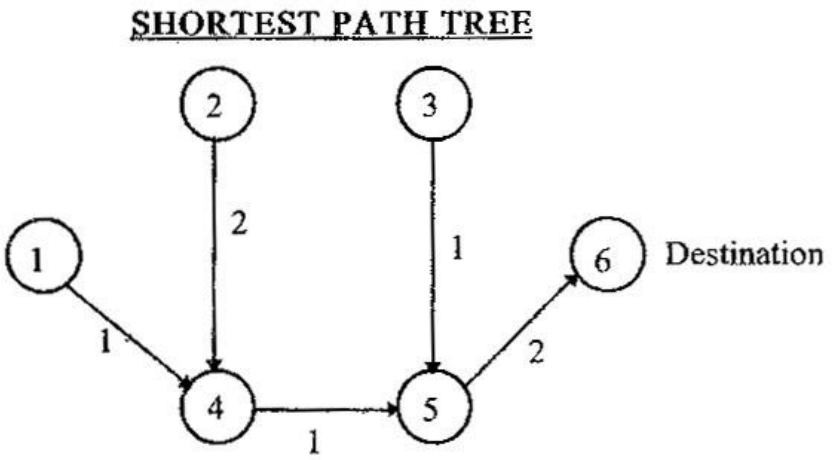
Iteration Number	Labels L(n), Current Predecessor Node P(n)									
	L(5), P(5)		L(4), P(4)		L(3), P(3)		L(2), P(2)		L(1), P(1)	
1	2	6	3	5	3	5	5	4	4	4
2	2	6	3	5	3	5	5	4	4	4

UPDATE ORDER 1,2,3,4,5

Iteration Number	Labels L(n), Current Predecessor Node P(n)									
	L(1), P(1)		L(2), P(2)		L(3), P(3)		L(4), P(4)		L(5), P(5)	
1	∞	-	∞	-	5	6	**8**	**3**	2	6
2	**9**	**4**	8	3	3	5	3	5	2	6
3	4	4	5	4	3	5	3	5	2	6
4	4	4	5	4	3	5	3	5	2	6

Η ταχύτητα σύγκλησης εξαρτάται από την σειρά ανανέωσης των Labels των κόμβων

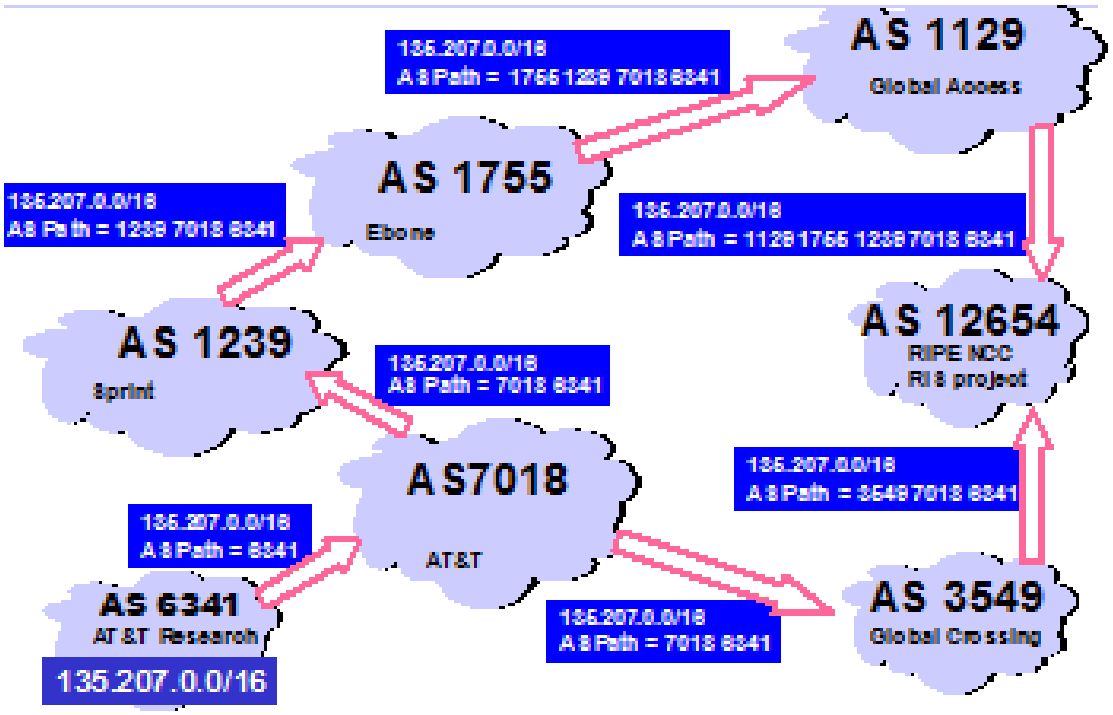




Παράδειγμα Δυναμικού Προγραμματισμού: Δρομολόγηση BGP στο Internet - RFC 4271 (6/7)

Παράδειγμα Μάθησης - Ανακοίνωσης Δικτύου 135.207.0.0/16

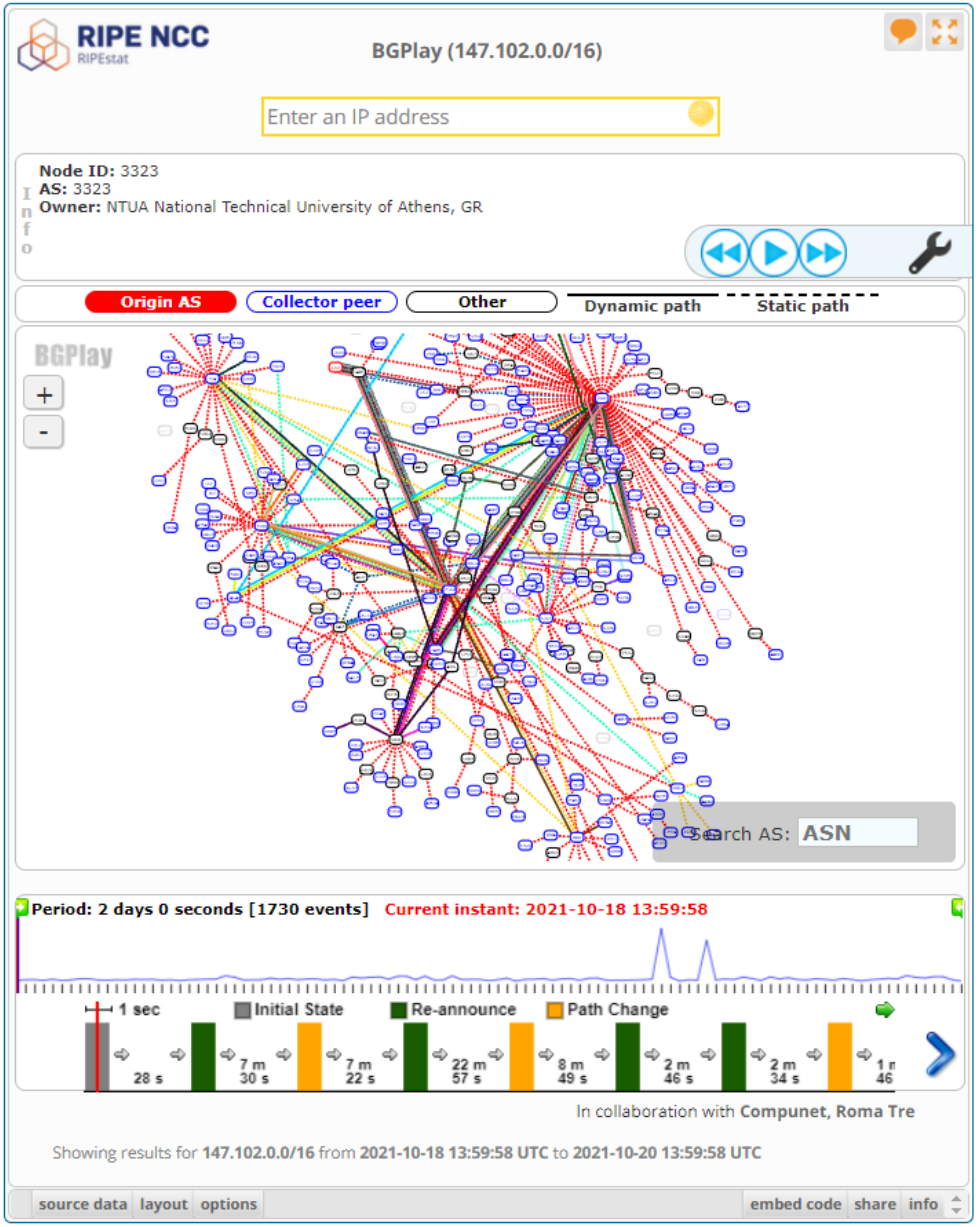
(από παρουσίαση του Timothy G. Griffin, AT&T Research, Paris 2002)



ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παράδειγμα Δυναμικού Προγραμματισμού: Δρομολόγηση BGP στο Internet - RFC 4271 (7/7)

Πραγματική Εικόνα των Δρόμων BGP (20-10-2021)



Παροχή Internet στο NTUA (ASN: 3323)

- GRNET (5408)
- GÉANT (21320)

GÉANT Internet Feeds

- COGENT (174)
- Telia (1299)
- HURRICANE US (6939)
- NORDUnet (2603)