



# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΕΡΓΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

**Ενισχυτική Μάθηση με Προσεγγιστικές Μεθόδους:**

**1. Χρονικές Διαφορές, 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](mailto:maglaris@netmode.ntua.gr)

[www.netmode.ntua.gr](http://www.netmode.ntua.gr)

Αίθουσα 002, Νέα Κτίρια ΣΗΜΜΥ

Τρίτη 16/4/2024

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Σύνοψη Εννοιών Δυναμικού Προγραμματισμού (1/2)

*D. P. Bertsekas & J. Tsitsiklis*, "Neuro-Dynamic Programming," Athena MA 1996

*R. S. Sutton & A. G. Barto*, "Reinforcement Learning," MIT Press 2018

### Βασικές Παράμετροι Δυναμικού Προγραμματισμού - Ελαχιστοποίηση Κόστους

**Άμεσο Κόστος (Observed Cost)** βήματος μετάβασης  $i \rightarrow j$  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)**:  $J^*(i) = \min_{a \in \mathcal{A}_i} (c(i, a) + \gamma \sum_{j=1}^N p_{ij} J^*(j))$ ,  $i = 1, 2, \dots, N$

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  $i \rightarrow j$  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  $\mu(i)$

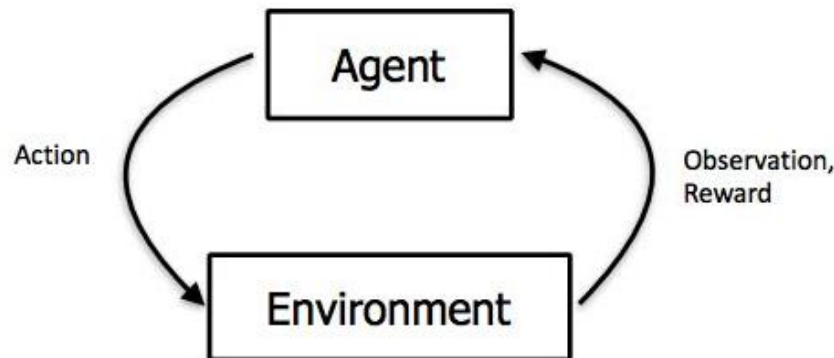
Βέλτιστα **Values (Bellman)**:  $V^*(i) = \max_{a \in \mathcal{A}_i} (r(i, a) + \gamma \sum_{j=1}^N p_{ij} V^*(j))$ ,  $i = 1, 2, \dots, N$

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  $\forall a \in \mathcal{A}_i$

### Model-based & Model-free Reinforcement Learning

[https://www.is.uni-freiburg.de/ressourcen/business-analytics/13\\_reinforcementlearning.pdf](https://www.is.uni-freiburg.de/ressourcen/business-analytics/13_reinforcementlearning.pdf)

- Αλγόριθμοι **Model-based** βασίζονται σε γνώση μοντέλων **Markov Decision Processes** εξέλιξης του περιβάλλοντος, δηλαδή πιθανοτήτων μετάβασης  $p_{ij}(a)$  και εφαρμογή επαναληπτικών μεθόδων **Policy** ή **Value Iteration**
- Αλγόριθμοι **Model-free** βασίζονται σε απευθείας μετρήσεις ή εκτιμήσεις σειρές εναλλαγής καταστάσεων του περιβάλλοντος (π.χ. με προσομοιώσεις **Monte Carlo** τροχιών καταστάσεων) και αναζήτηση βέλτιστων πολιτικών παρέμβασης του **Agent** με βάση την αποκτούμενη γνώση του κατά τη διαδικασία μάθησης, χωρίς πρότερη γνώση παραμέτρων του δυναμικού μοντέλου του περιβάλλοντος



# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Απευθείας Προσεγγιστικές Μέθοδοι Δυναμικού Προγραμματισμού (1/3)

- Οι δύο **Model-based** 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**  $i \rightarrow j$  που καθορίζονται από μια πολιτική  $a = \mu(i)$

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

- Οι απευθείας **Model-free** προσεγγιστικές μέθοδοι (**Direct Approximate Dynamic Programming Methods**) ορίζουν σε κάθε βήμα την επόμενη μετάβαση  $i \rightarrow j$  με απόφαση  $a = \mu(i)$  και με γνωστό κόστος  $g(i, a, j)$  και εκτιμούν τα αναμενόμενα κόστη καταστάσεων - αποφάσεων  $c(i, a)$  πολιτικών  $a = \mu(i)$  σαν μέσες τιμές ανεξαρτήτων τροχιών (**trajectories**) που επισκέπτονται την κατάσταση  $i$  κατά τη διαδικασία μάθησης

Αντιστοιχούν στους δύο βασικούς αλγόριθμους Δυναμικού Προγραμματισμού με τις εξής **Model-free** παραλλαγές:

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

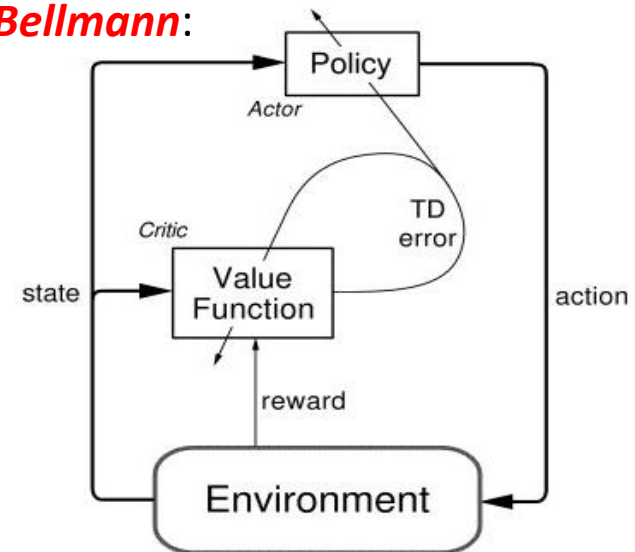
Οι απευθείας προσεγγιστικές μέθοδοι θεωρούνται οι κύριες υλοποιήσεις της Ενισχυτικής Μάθησης: **Reinforcement Learning**  $\cong$  **Direct Approximations of Dynamic Programming**

### Ορισμοί *on-policy*, *off-policy*

- Η *on-policy* σε κάθε βήμα εκτιμά (με μετρήσεις ή προσομοιώσεις *Monte Carlo*) τα *Cost-to-Go*  $J(i)$  των καταστάσεων  $i$ . Με επισκέψεις καταστάσεων κατά μήκος μιας τροχιάς (*trajectory*) οδηγούμενες από αποφάσεις  $i \rightarrow a$ , ανανεώνονται οι συναρτήσεις  $J(i)$  και οδηγούνται προς σύγκλιση σύμφωνα με τις εξισώσεις του *Bellman*:

Value Iteration  $\rightarrow$  TD(0)-Learning

Actor-Critic TD-Learning Model: *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



- Η *off-policy* συγκρίνει εναλλακτικές αποφάσεις  $a$  σε καταστάσεις του περιβάλλοντος  $i$  μιας τροχιάς (*trajectory*) με εκτιμήσεις των  $Q(i, a)$  ώστε σε επόμενο βήμα (επανάληψη) ο agent να **επιλέξει** με απληστία αποφάσεις με το ελάχιστο  $Q(i, a)$  για την κατάσταση  $i$ . Το *Cost-to-Go*  $J^\mu(i)$  μιας προσωρινής πολιτικής  $\mu$  εκτιμώνται με μετρήσεις ή προσομοιώσεις *Monte Carlo* των *trajectories* που προέκυψαν με την παρούσα πολιτική, χωρίς να συμπεριλαμβάνονται άπληστες αποφάσεις  $i \rightarrow a$  που θα προκύψουν από τα *Q-Factors* στο επόμενο βήμα:

Policy Iteration  $\rightarrow$  Q-Learning

### Γενική Μεθοδολογία Προσομοίωσης Τροχιών Monte Carlo

- Οι προσομοιώσεις **Monte Carlo** δημιουργούν σενάρια πολλαπλών πιθανών τροχιών (**system trajectories**) της εξέλιξης του **Markov Decision Process** από μια αρχική κατάσταση  $i_0$  μέχρι κάποια τελική  $i_n \rightarrow i_T$  (*T - Terminal*: Βήμα τερματισμού τροχιάς)
- Για περιπτώσεις πεπερασμένου ορίζοντα  $\{0, 1, \dots, K\}$  ορίζεται και ο όρος **επεισόδιο** για υλοποίηση πλήρους τροχιάς  $i_0 \rightarrow i_K$  **κατά μήκος όλου του ορίζοντα**. Οι τροχιές ορίζονται για υποσύνολα της εξέλιξης του περιβάλλοντος, άρα η καταγραφή τους ισχύει και για σενάρια απείρου ορίζοντα  $K \rightarrow \infty$
- Η διαδικασία μάθησης περιλαμβάνει δειγματοληπτική καταγραφή (**Monte Carlo Sampling**) πολλών ανεξάρτητων **trajectories** που δημιουργούνται σε κάθε επίσκεψη στη κατάσταση  $i_n$  με εξερεύνηση (**exploration**) εναλλακτικών επόμενης κατάστασης  $i_{n+1}$ . Οι τροχιές μπορεί να αγνοούν δυσπρόσιτες ή/και σπανίως επισκέψιμες καταστάσεις
- Οι τιμές συναρτήσεων **cost-to-go**  $J^\mu(i)$  ανανεώνονται σε κάθε προσομοίωση με προσθήκη του (εκ των προτέρων **γνωστού** από τη διατύπωση του προβλήματος) **άμεσου** (**observed**) **κόστους μετάβασης**  $g(i, j)$  σε επισκέψεις προσομοιωμένης τροχιάς μεταβάσεων του περιβάλλοντος από κατάσταση  $i$  προς κατάσταση  $j$
- Οι μέθοδοι **Monte Carlo** απαιτούν γνώση της δομής του περιβάλλοντος από εμπειρία (όχι από γνώση πιθανοτήτων μετάβασης), διαχειρήσιμο αριθμό παρατηρήσιμων (**observable**) καταστάσεων και σημαντικό αριθμό από **trajectories** για καλές εκτιμήσεις

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Προσεγγιστικός Αλγόριθμος $TD(0)$ Learning (1/2)

### Value Iteration $\rightarrow$ Temporal-Difference $TD(0)$ Learning

- Με επανειλημμένες προσομοιώσεις **Monte Carlo** δημιουργούμε  $M$  **trajectories** (τροχιές) του περιβάλλοντος με κάποια πολιτική  $\mu$  και προσεγγίζουμε τα **Costs-to-Go**  $J^\mu(i_n)$  με βάση τις εξισώσεις του **Bellman** από  $i_n$ ,  $n < T$ , προς τελική κατάσταση  $i_T = 0$ :

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

- Τα  $J^\mu(i_n)$  εκτιμώνται σαν **ensemble averages** του υπολειπόμενου κόστους  $J(i_n)$  κατά μήκος  $M$  τροχιών  $\{i_n, i_{n+1}, \dots, i_T\}$ . Το κόστος μιας τροχιάς είναι  $c(i_n) \triangleq \sum_{k=0}^{T-n-1} \gamma^k g(i_{n+k}, i_{n+k+1})$  και για  $M$  ανεξάρτητες τροχιές ο μέσος όρος  $J(i_n) = E[c(i_n)]$  εκτιμάται σαν

$$J(i_n) = E[c(i_n)] \cong \frac{1}{M} \sum_M c(i_n)$$

- Τα κόστη  $J(i_n)$  συγκλίνουν μέσω **Robbins-Monro Successive Approximations**, επαναλήψεων που διορθώνουν εκτιμήσεις τιμών τους (**updates**) σε κάθε επίσκεψη της κατάστασης  $i_n$  κατά την εξέλιξη μιας τροχιάς  $i_n \rightarrow i_{n+1}$  με συντελεστή μάθησης (**learning rate**)  $\eta_n$ :

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

- 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**. Οδηγεί τα  $J(i_n)$  προς σύγκλιση ελαχιστοποιώντας το σφάλμα  $d_n$  με επαναλήψεις **ανεξάρτητων** τροχιών που προκύπτουν από την εφαρμογή μιας πολιτικής  $\mu$

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Προσεγγιστικός Αλγόριθμος $TD(0)$ Learning (2/2)

Value Iteration  $\rightarrow$  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  $n$  κάποιας **trajectory**

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

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 παρατηρήσιμες (**observable**) τροχιές  $T$  βημάτων που παράγονται από προσομοιώσεις **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 $\rightarrow$ Q-Learning

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  $\pi$  με αντιστοίχιση

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

- Προσδιορισμός πολιτικής βέλτιστης συμπεριφοράς (**off-policy behavior generation**) μέσω επεξεργασίας 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 εναλλακτικές **trajectories** ο αλγόριθμος **Q-Learning** οδηγεί το σύστημα στη μάθηση βέλτιστης πολιτικής σαν υλοποίηση του **policy iteration**
- **Προϋπόθεση:** Η  $i_n$  που προκύπτει σε μια **trajectory** πρέπει να είναι **fully observable**

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

### Στοχαστική Model-free Παραλλαγή με Προσομοίωση Τροχιών Monte Carlo

- Για να αποφύγουμε γνώση των  $p_{ij}(a)$  ορίζουμε τροχιά (**trajectory**) από αρχική κατάσταση  $i_0$  μέχρι την  $i_n$  με προσομοίωση **Monte Carlo** εξέλιξης της συμπεριφοράς του περιβάλλοντος και εφαρμογή κάποιας **behavior policy**
- Στο παρόν βήμα  $n$  ο **agent** εκτιμά τους **Q-factors** και τα αναμενόμενα **costs-to-go**  $J_n(j)$  για καταστάσεις  $j$  που προκύπτουν με **greedy estimation policy** (μη ταυτιζόμενη με την **behavior policy** που δημιούργησε την τροχιά της κατάστασης του περιβάλλοντος)
- Η μέση τιμή (**average**) στον υπολογισμό των **Q-factors** αντί της μέσης προβλεπόμενης κατάστασης που απαιτεί γνώση των  $p_{ij}(a)$  προσεγγίζεται από την θεώρηση μιας τροχιάς και σε συνέχεια υπολογισμό μέσω  $n$  όρων πολλαπλών ανεξάρτητων τροχιών
- Για ζεύγος  $(i, a) = (i_n, a_n)$  επαναληπτικός αλγόριθμος υπολογίζει τα  $Q_{n+1}(i, a)$  από  $Q_n(i, a)$  ως εξής:
  - $Q_{n+1}(i, a) = (1 - \eta_n)Q_n(i, a) + \eta_n[g(i, a, j) + \gamma J_n(j)]$
  - $J_n(j) = \min_{b \in \mathcal{A}_j} Q_n(j, b)$  όπου  $j = i_{n+1}$  η **επόμενη** κατάσταση της  $i = i_n$
  - $Q_{n+1}(i, a) = Q_n(i, a)$  για όλα τα υπόλοιπα ζεύγη  $(i, a) \neq (i_n, a_n)$
  - Με την πρόοδο των επαναλήψεων  $Q_n(i, a) \rightarrow Q^*(i, a)$
  - Η **learning parameter**  $\eta_n$  είναι φθίνουσα ως προς  $n$ , π.χ.  $\eta_n = \alpha / (\beta + n)$   $\alpha, \beta > 0$

Επειδή τροχιές με **greedy** αποφάσεις (**exploitation**) μπορεί να αγνοήσουν επιλογές καταστάσεων λόγω εκκίνησης από μια κατάσταση, απαιτείται προσομοίωση πολλαπλών **τροχιών** για επισκέψεις σε ευρύ φάσμα καταστάσεων (**exploration**). Το εύρος αναζήτησης ενισχύεται με απόφαση **greedy** με πιθανότητα  $(1 - \epsilon)$  ή άλλης με πιθανότητα  $\epsilon$

# 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 (~73, 800 **agents/routers**) στο **Border Gateway Protocol (BGP)** για δρομολόγηση προς τα ~1, 200, 000 **γνωστά δίκτυα** του παγκόσμιου **Internet**

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

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

[https://helios.ntua.gr/pluginfile.php/247068/mod\\_page/content/4/NetMan Internet ISP IXP Routing 2023 10 30.pdf](https://helios.ntua.gr/pluginfile.php/247068/mod_page/content/4/NetMan%20Internet%20ISP%20IXP%20Routing%202023%2010%2030.pdf)

Το παγκόσμιο *Internet* αποτελείται (7/2023) από ~1, 200, 000 γνωστά δίκτυα τελικούς προορισμούς (π.χ. Δίκτυο ΕΜΠ, IP: 147.102.0.0/16), οργανωμένα σε ~73, 800 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 73, 800 AS's μέσω γενικών πινάκων δρομολόγησης σε συνοριακούς δρομολογητές (*Border Gateways, Border Routers*) με καταχωρήσεις για όλα τα ~1, 200, 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 73, 800 *AS's* του *Internet* τα 1, 200, 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 1,200, 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)

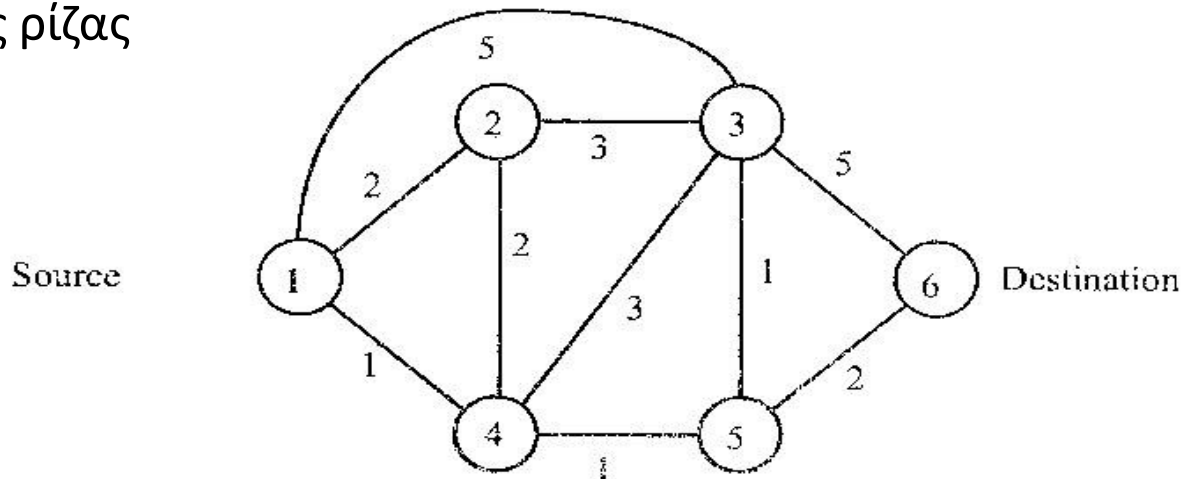
### Αλγόριθμος Distance Vector (Bellman – Ford) BGP (Bellman – Ford)

- 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** (~1, 200, 000 - 7/2023)
- Η βελτιστοποίηση βασίζεται σε κατανεμημένο αλγόριθμο **Bellman - Ford** που υλοποιείται μέσω σηματοδοσίας ανακοινώσεων (**BGP Announcements**) μεταξύ όλων των (~73, 800 - 10/2023) 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**) 73, 800 αυτόνομων **Agents**

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

### 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



### 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  $N = 6$ , πάνω από 83,000 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  $n$  (**Transition 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  $i$ :

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

- 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:  $L(1)=L(2)=\dots=L(5)=\infty, L(6)=0$

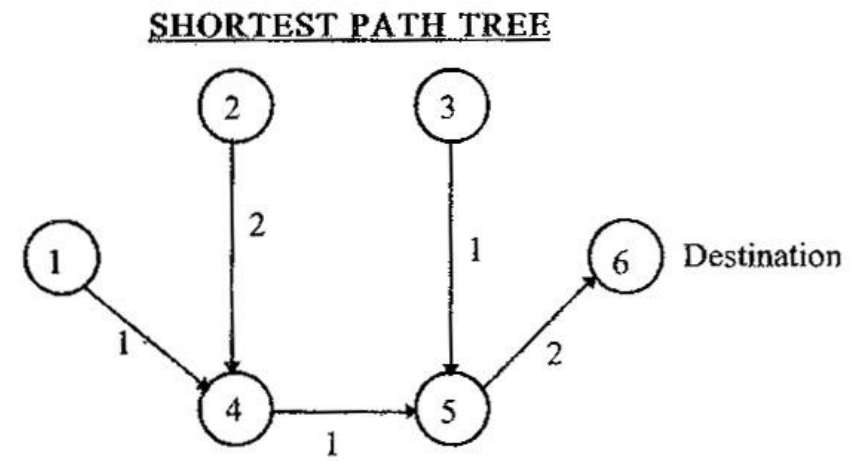
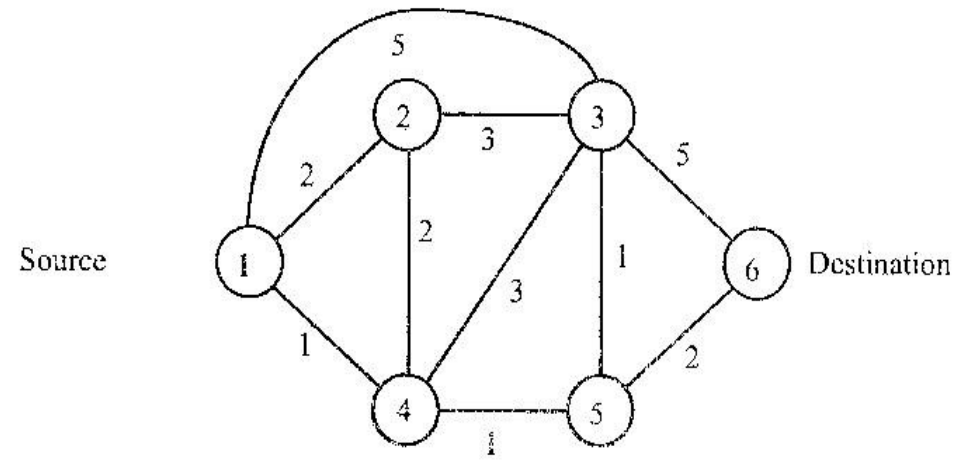
#### UPDATE ORDER 5,4,3,2,1

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

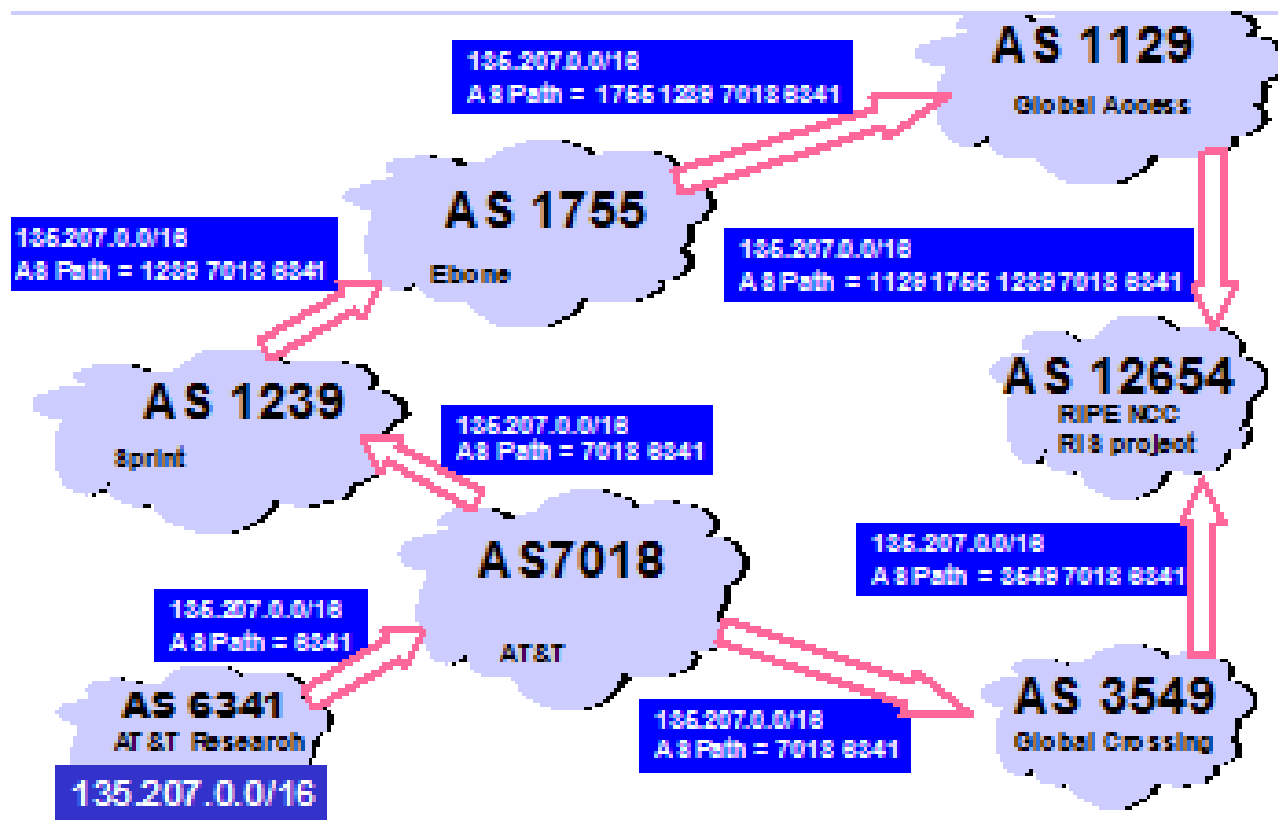
Iteration Number	Labels L(n), Current Predecessor Node P(n)				
	L(1), P(1)	L(2), P(2)	L(3), P(3)	L(4), P(4)	L(5), P(5)
1	$\infty$ -	$\infty$ -	5 6	8 3	2 6
2	9 4	8 3	3 5	3 5	2 6
3	4 4	5 4	3 5	3 5	2 6
4	4 4	5 4	3 5	3 5	2 6

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





Παράδειγμα Μάθησης - Ανακοίνωσης Δικτύου 135.207.0.0/16  
(από παρουσίαση του *Timothy G. Griffin, AT&T Research, Paris 2002*)

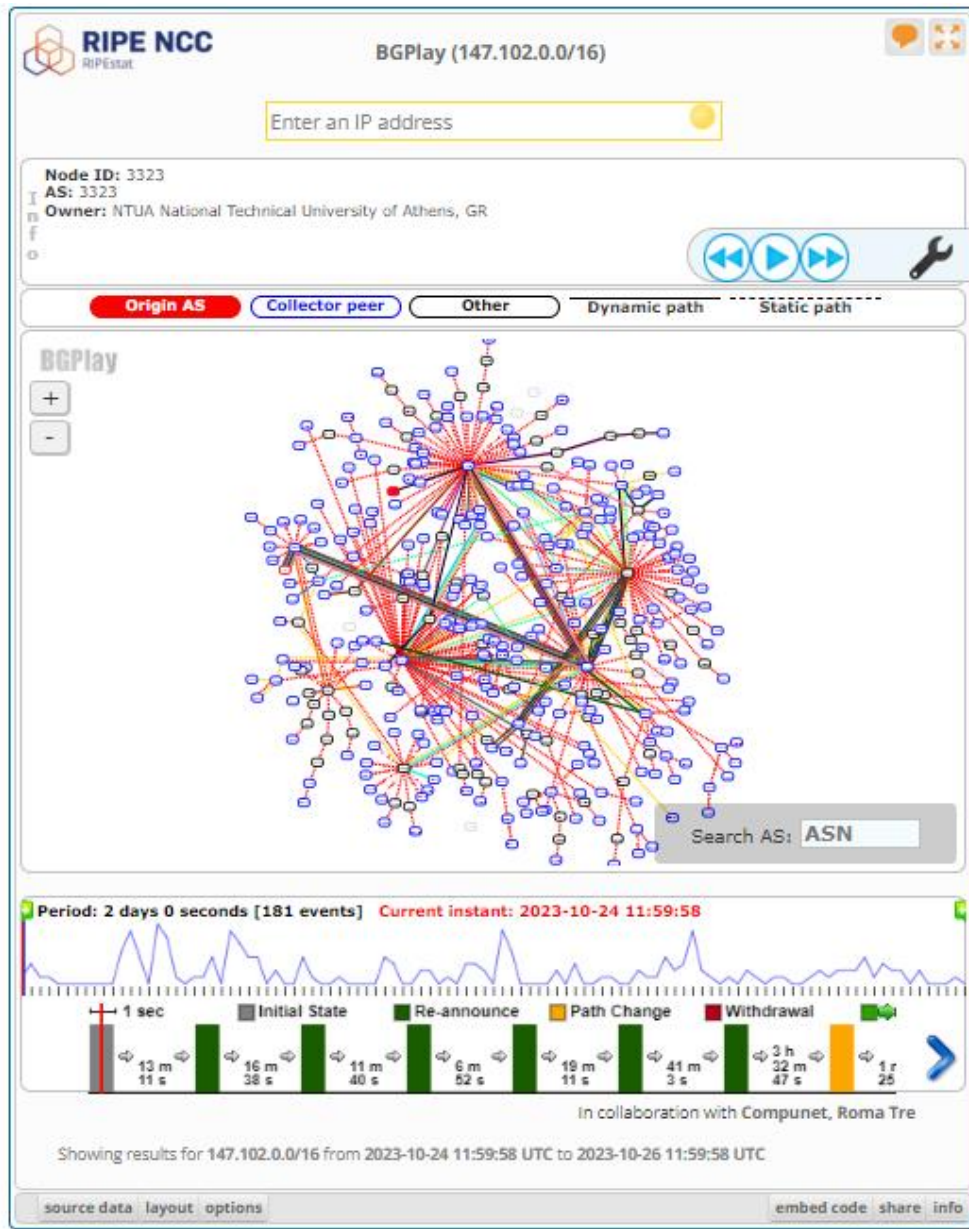


# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

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

**Εικόνα των Δρόμων BGP προς το ntu.gr – 147.102.0.0/16 (26/10/2023)**

<https://stat.ripe.net/widget/bgplay>



**NTUA (AS: 3323)**  
**GRNET (AS: 5408)**  
**ΓÉANT (AS: 21320)**

**ΓÉANT Internet Feeds**

- **LEVEL3 (AS: 3356)**
- **COGENT 174 (AS: 174)**
- **NORDUnet (AS: 2603)**