PhD, MBA Βεζέρης Δημήτριος

Επιχειρησιακή έρευνα.

ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

Lecture OR - Bellman-Ford Algorithm

OR - Αλγόριθμος Bellman-Ford

Αλγόριθμοι συντομότερης διαδρομής

Εκτός από τον αλγόριθμο Dijsktra, τα προβλήματα συντομότερης διαδρομής λύνονται με δυναμικό προγραμματισμό. Εδώ εξετάζουμε τον αλγόριθμο Bellman-Ford, μία από τις μεθοδολογίες δυναμικού προγραμματισμού.

Αλγόριθμος Bellman-Ford

Ο αλγόριθμος αυτός εφαρμόζεται σε κατευθυνόμενους γράφους με βάρη G=(V,E), ειδικά με αρνητικά βάρη που δεν εφαρμόζεται ο Dijkstra. Με δεδομένο ένα κατευθυνόμενο γράφο με βάρη G=(V,E) και μια συνάρτηση βαρών w: E→R, θέλουμε να βρούμε μονοπάτια με το ελάχιστο δυνατό βάρος.

Το βάρος w(p) ενός μονοπατιού p δίνεται ως εξής:

Αν $p = v_o \to v_1 \to \ldots \to v_k$

τότε $w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$

Πηγή εικόνας (https://www.chegg.com/homework-help/questions-and-answers/run-bellman-ford-algorithm-directed-graph-figure-using-ver-tex-z-source-pass-relax-edges-o-q25269187)

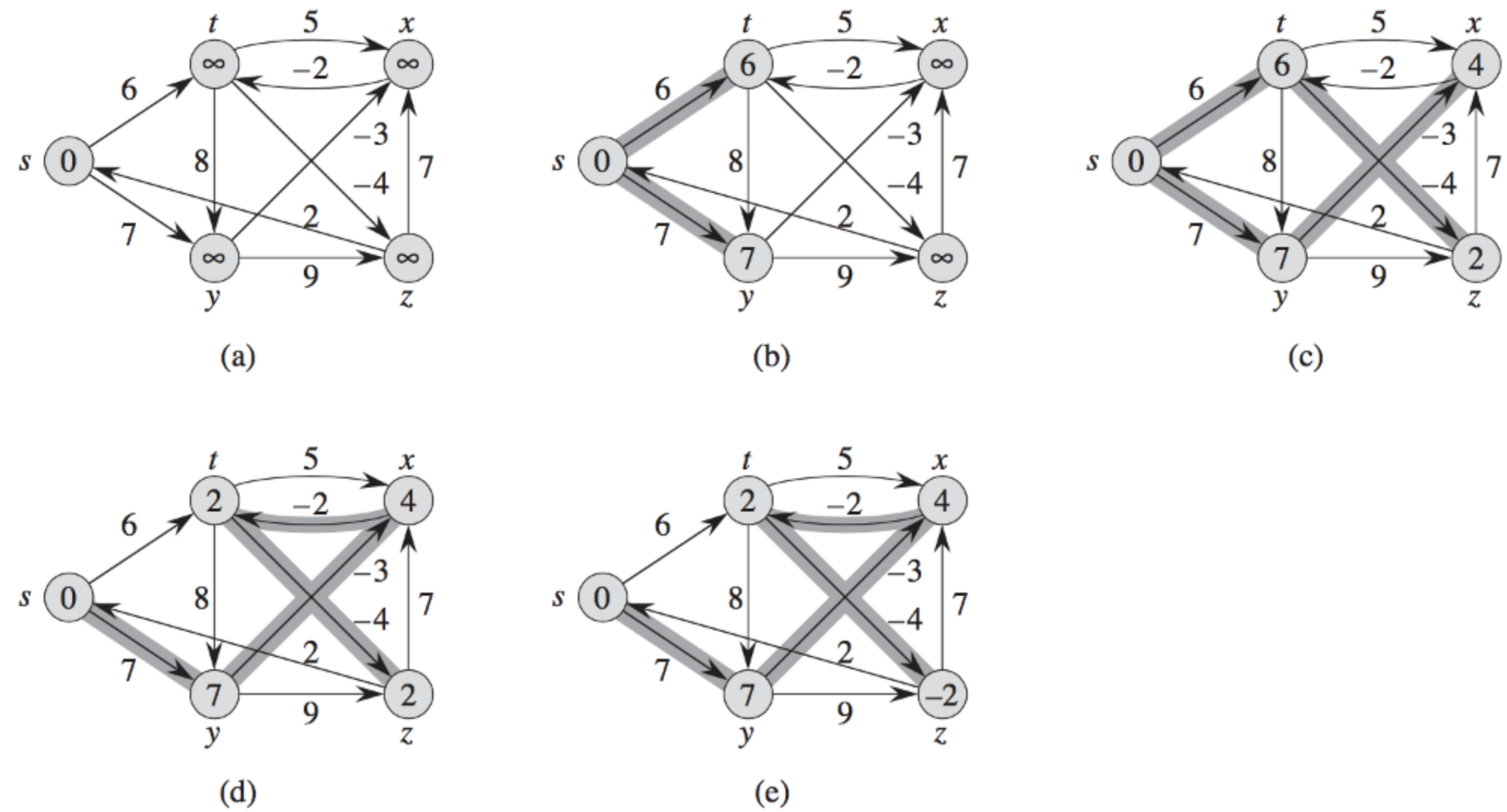


Figure 24.4 The execution of the Bellman-Ford algorithm. The source is vertex s. The d values appear within the vertices, and shaded edges indicate predecessor values: if edge (u, v) is shaded, then $v.\pi = u$. In this particular example, each pass relaxes the edges in the order $(t,x), (t,y), (t,z), (x,t), (y,x), (y,z), (z,x), (z,s), (s,t), (s,y)$. **(a)** The situation just before the first pass over the edges. **(b)–(e)** The situation after each successive pass over the edges. The d and π values in part (e) are the final values. The Bellman-Ford algorithm returns TRUE in this example.

Έτσι λοιπόν έχουμε:

Αρχικό βήμα

Βάζουμε ετικέτα στον κόμβο - πηγή [0]. Βάζουμε σε όλους τους υπόλοιπους κόμβους την ετικέτα [∞], που ορίζει στην αρχικά άπειρη την απόσταση από την πηγή (μεγαλύτερη δεν υπάρχει, συνεπώς όλες οι υπόλοιπες που μπορούν να υπολογιστούν θα είναι μικρότερες).

OR - Αλγόριθμος Bellman-Ford

Βήμα i (γενικά)

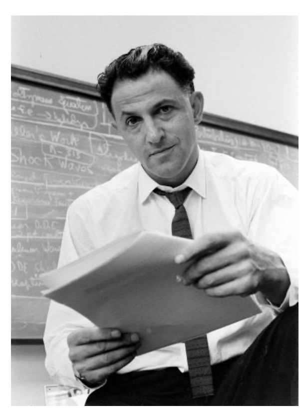
Όπως και ο αλγόριθμος της Dijkstra , ο Bellman – Ford εφαρμόζεται με χαλάρωση (relaxation), στην οποία οι εκτιμήσεις στη σωστή απόσταση αντικαθίστανται από καλύτερες μέχρι να φτάσουμε τελικά στη λύση.

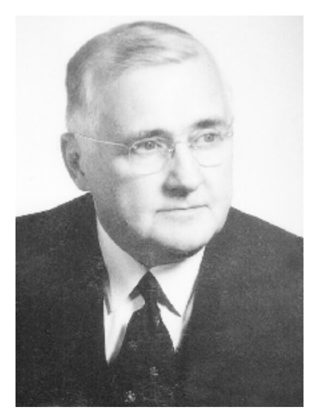
Και στους δύο αλγόριθμους, η κατά προσέγγιση απόσταση από κάθε κορυφή είναι πάντοτε υπερεκτίμηση της πραγματικής απόστασης και αντικαθίσταται από την ελάχιστη από την παλιά τιμή που βρέθηκε πρόσφατα.

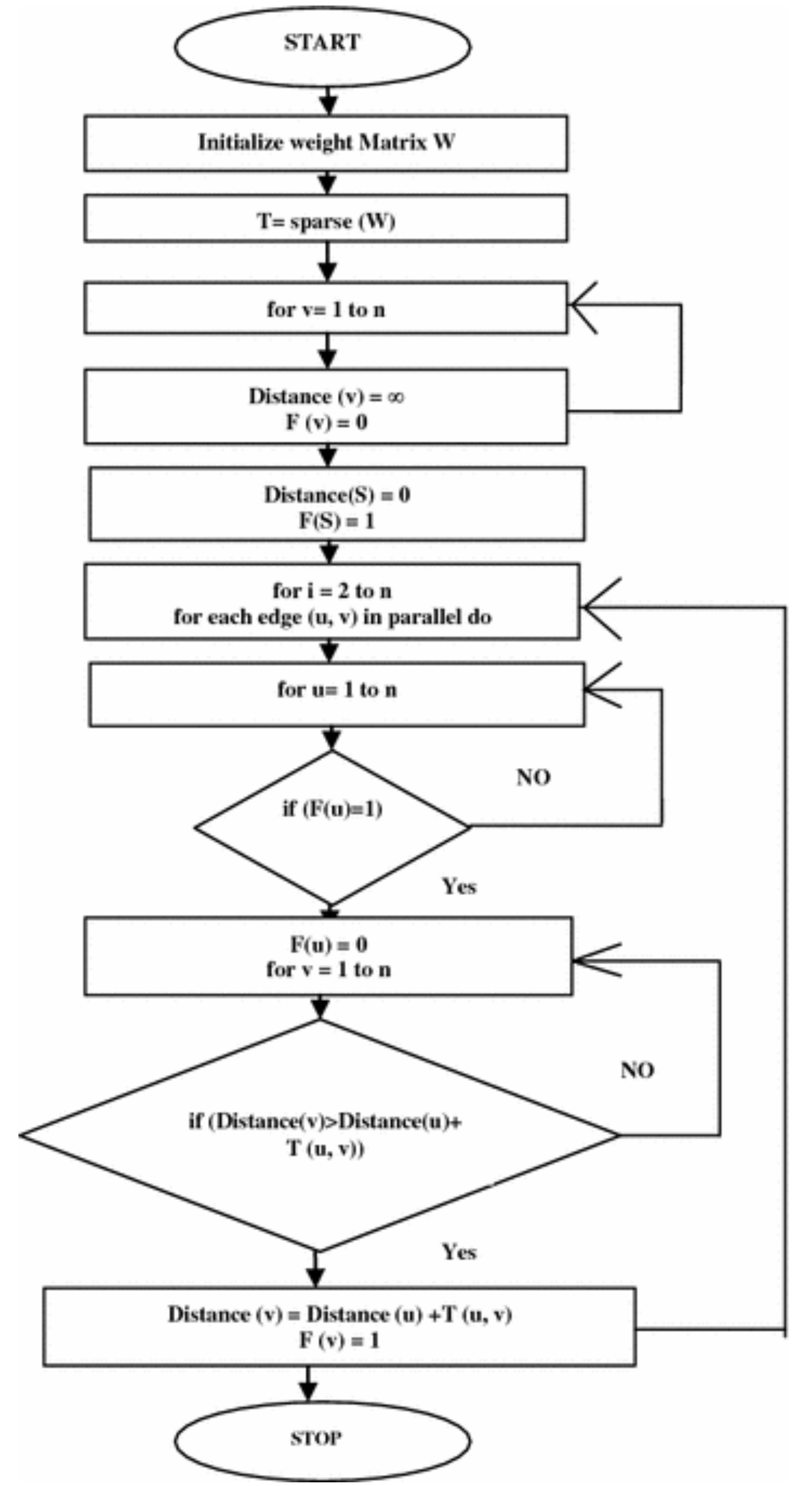
Ωστόσο, ο αλγόριθμος Dijkstra χρησιμοποιεί μια σειρά προτεραιότητας για να επιλέξει την πλησιέστερη κορυφή που δεν έχει ακόμη υποβληθεί σε εκτίμηση και εκτελεί αυτήν τη διαδικασία χαλάρωσης σε όλες τις ακμές της.

Αντίθετα, ο αλγόριθμος Bellman – Ford χαλαρώνει απλώς όλες τις άκρες και το κάνει V -1 φορές, πού V είναι ο αριθμός κορυφών στο γράφημα. Σε καθεμία από αυτές τις επαναλήψεις, αυξάνεται ο αριθμός των κορυφών με σωστά υπολογισμένες αποστάσεις, από τις οποίες προκύπτει ότι τελικά όλες οι κορυφές θα έχουν τις σωστές αποστάσεις τους.

Αυτή η μέθοδος επιτρέπει στον αλγόριθμο Bellman-Ford να εφαρμοστεί σε μια ευρύτερη κατηγορία προβλημάτων από την Dijkstra.







Εφαρμογή αλγορίθμου Bellman-Ford

Θα εφαρμόσουμε τον αλγόριθμο στο διπλανό δίκτυο. Θέλουμε να βρούμε την διαδρομή από το A στο G με το μικρότερο βάρος. Προσέξτε ότι υπάρχουν αρνητικά βάρη.

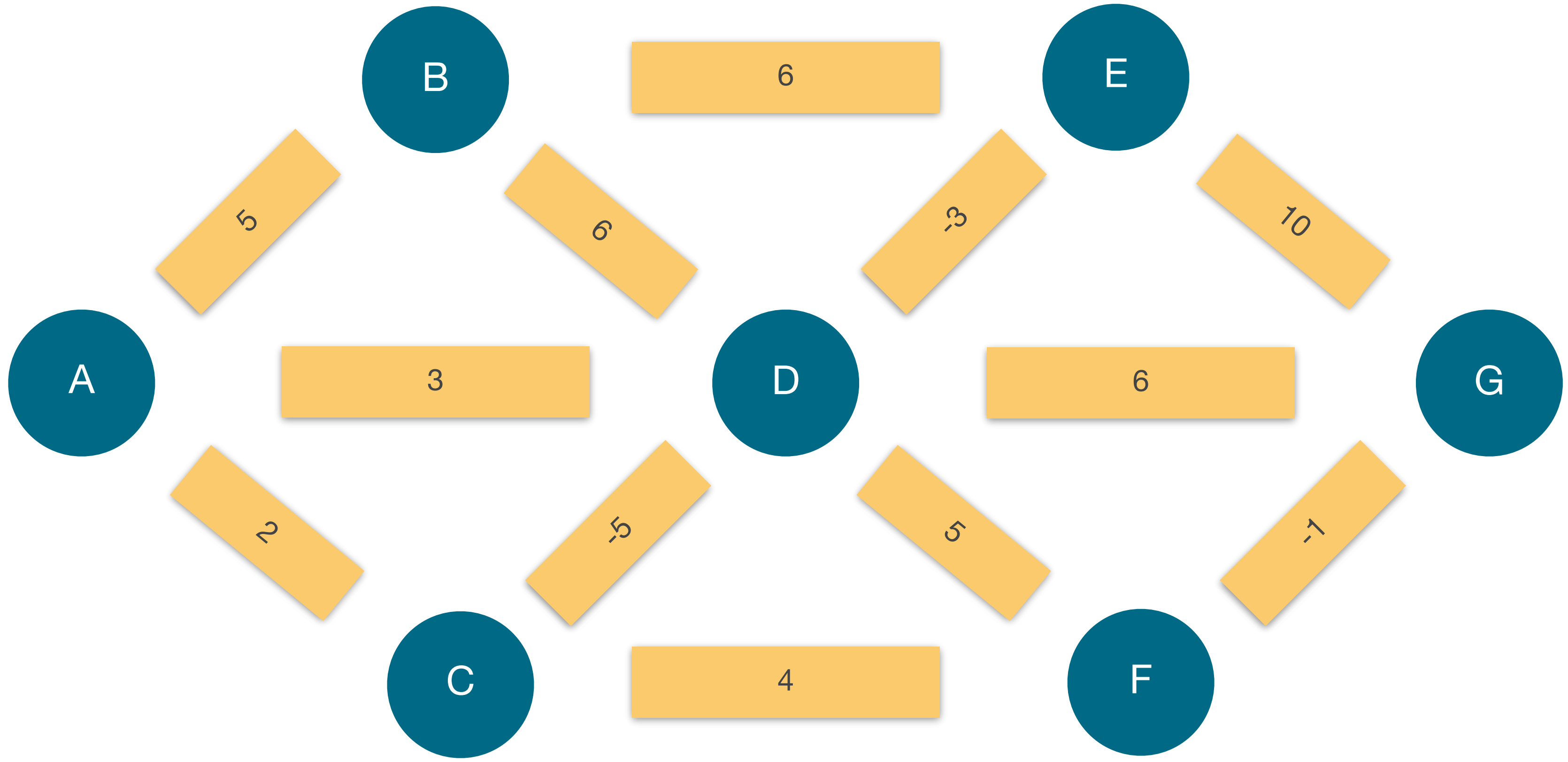
Αρχικά θα τοποθετήσουμε την ετικέτα στον κόμβο πηγή με μηδενικό βάρος.

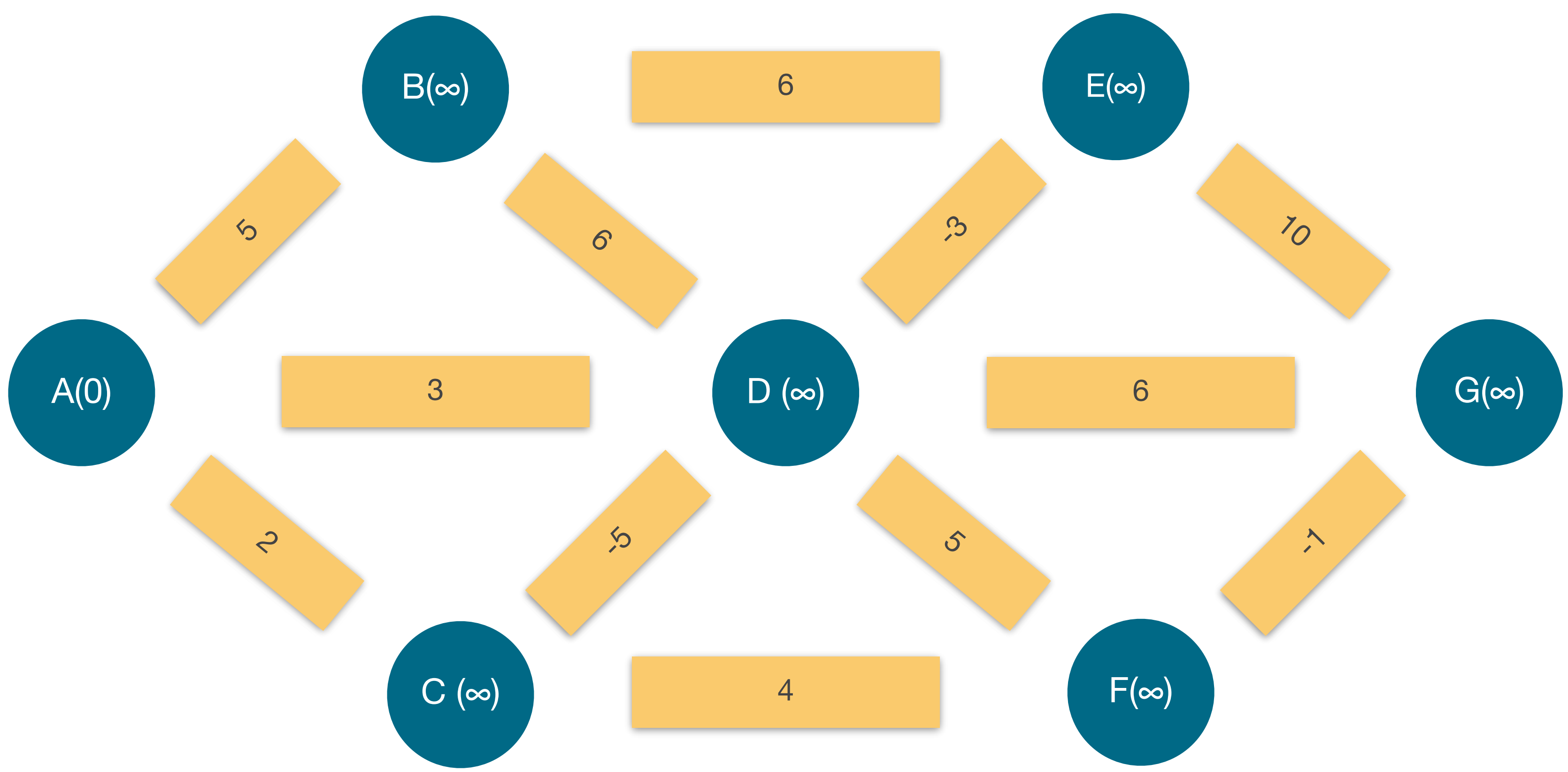
Σε όλους τους άλλους κόμβους θα τοποθετήσουμε την ετικέτα με την άπειρη απόσταση.

Υπάρχουν όπως φαίνεται A,B,C,D,E,F,G στο σύνολο V=7 κόμβοι. Αυτό σημαίνει ότι θα κάνουμε V-1 = 7-1 = 6 επαναλήψεις υπολογισμών, μέχρι να ολοκληρώσουμε την μέθοδο και να βρούμε την μικρότερη σε μήκος διαδρομή.

Ορίζοντας όλες τις πιθανές συνδέσεις έχουμε υπολογισμό στις επαναλήψεις για τα μονοπάτια:

(A,B),(A,C),(A,D),(B,E),(B,D),(C,D),(C,F),(D,E),(D,F),(D,G),(E,G),(F,G).





Εφαρμογή αλγορίθμου Bellman-Ford Βήμα 1/6

Υπολογίζοντας τα βάρη για όλους τους κλάδους έχουμε:

A,B : 0+5=5<∞ —-> **B(5)**

A,C : 0+2=2<∞ —-> **C(2)**

A,D : 0+3=3<∞ —-> **D(3)**

B,E : 5+6=11<∞ —-> **E(11)**

B,D : 5+6=11>**3** —-> **D(3)**

C,D : 2+(-5)=-3<**3** —-> **D(-3)**

C,F : 2+4=6<∞ —-> **F(6)**

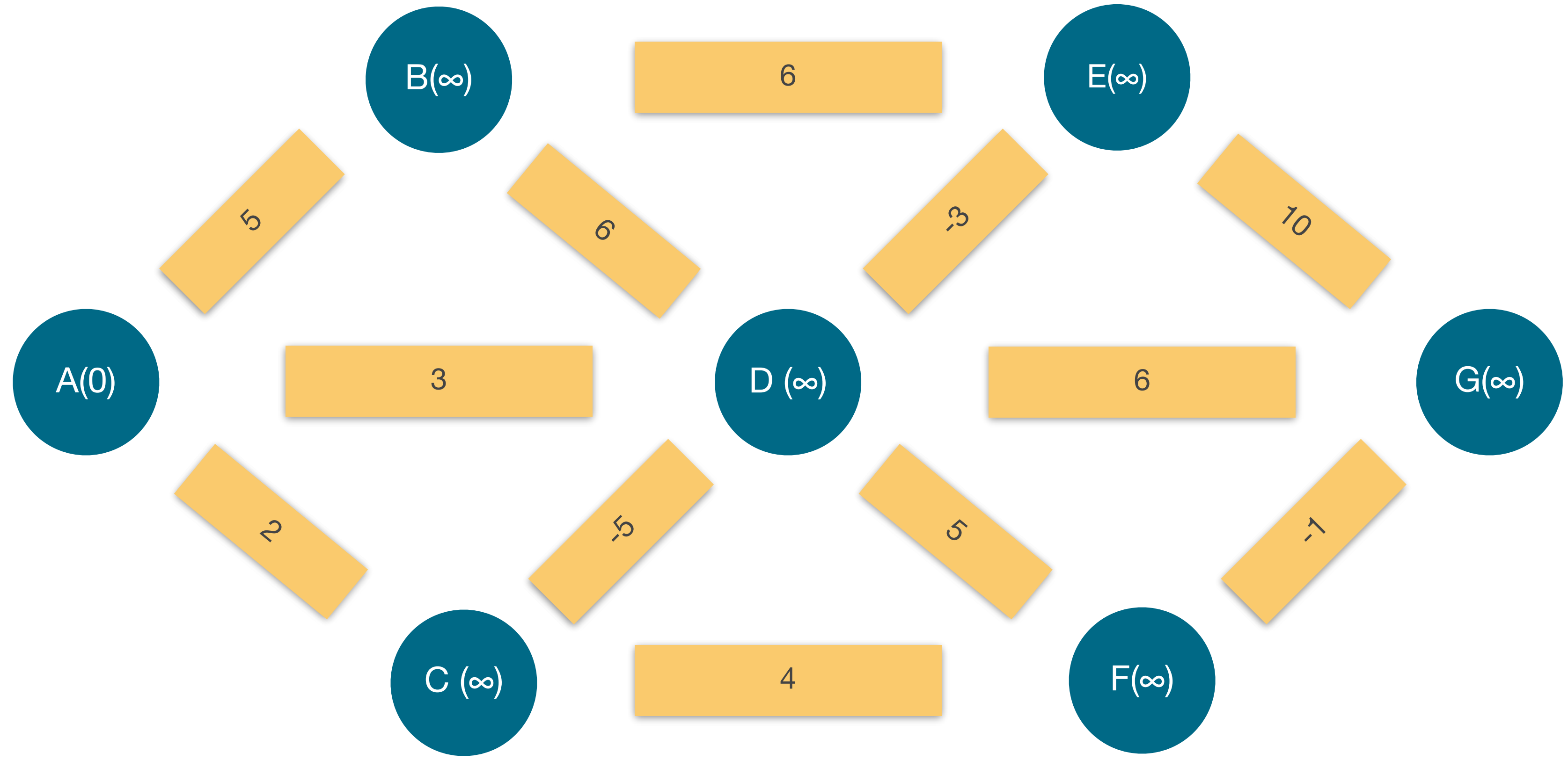
D,E : -3+(-3)=-6<**11** —-> **E(-6)**

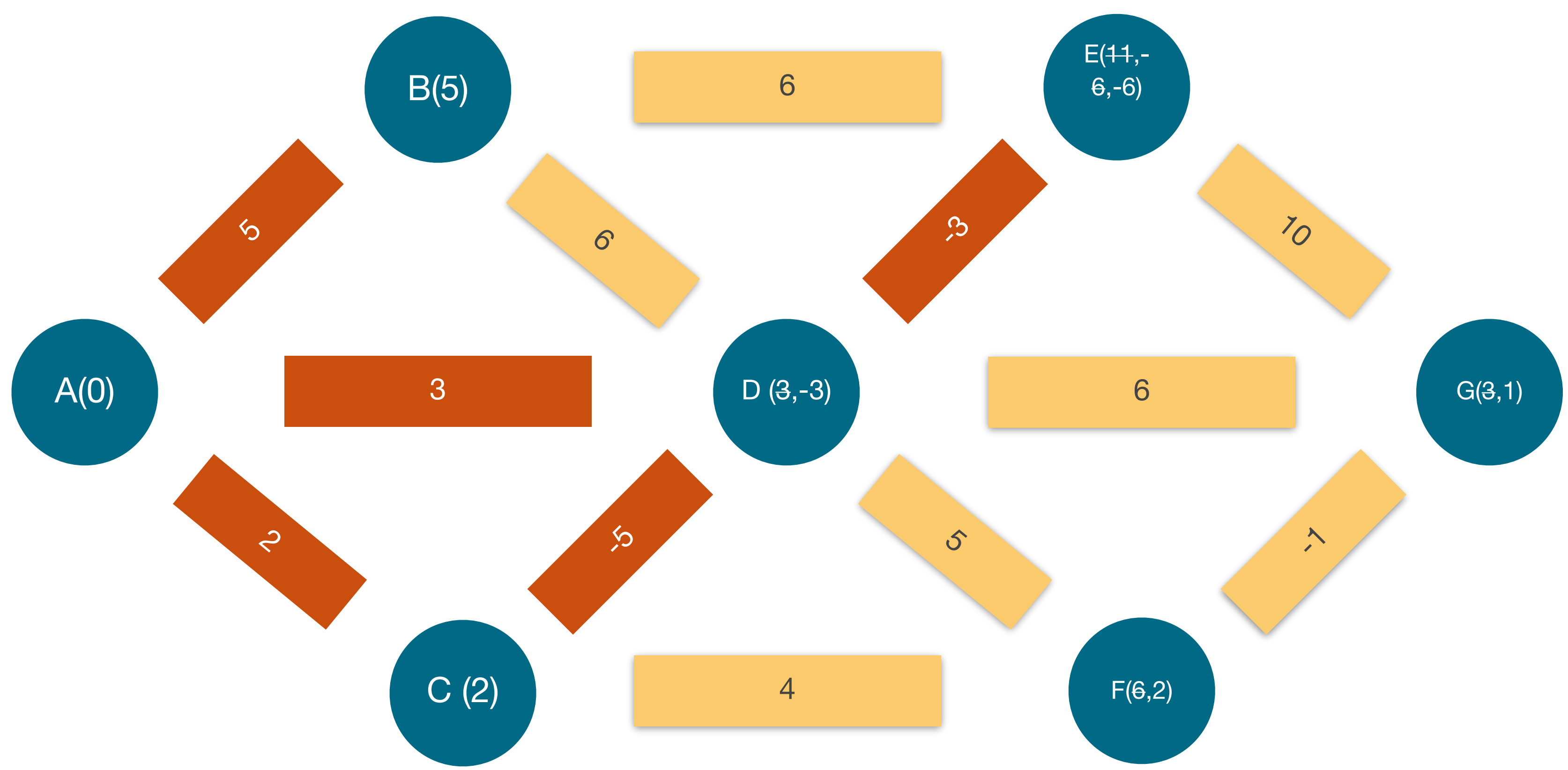
D,G : -3+6=3<∞ —-> **G(3)**

D,F : -3+5=2<**6** —-> **F(2)**

E,G : -6+10=4>**3** —-> **G(3)**

F,G : 2+(-1)=1<**3** —-> **G(1)**





Εφαρμογή αλγορίθμου Bellman-Ford Βήμα 2/6

Υπολογίζοντας τα βάρη για όλους τους κλάδους έχουμε:

A,B : 0+5=5=**5 —-> B(5)**

A,C : 0+2=2=**2 —-> C(2)**

A,D : 0+3=3>-**3 —-> D(-3)**

B,E : 5+6=11>**-6 —-> E(-6)**

B,D : 5+6=11>-**3 —-> D(-3)**

C,D : 2+(-5)=-3=-**3 —-> D(-3)**

C,F : 2+4=6>**2 —-> F(2)**

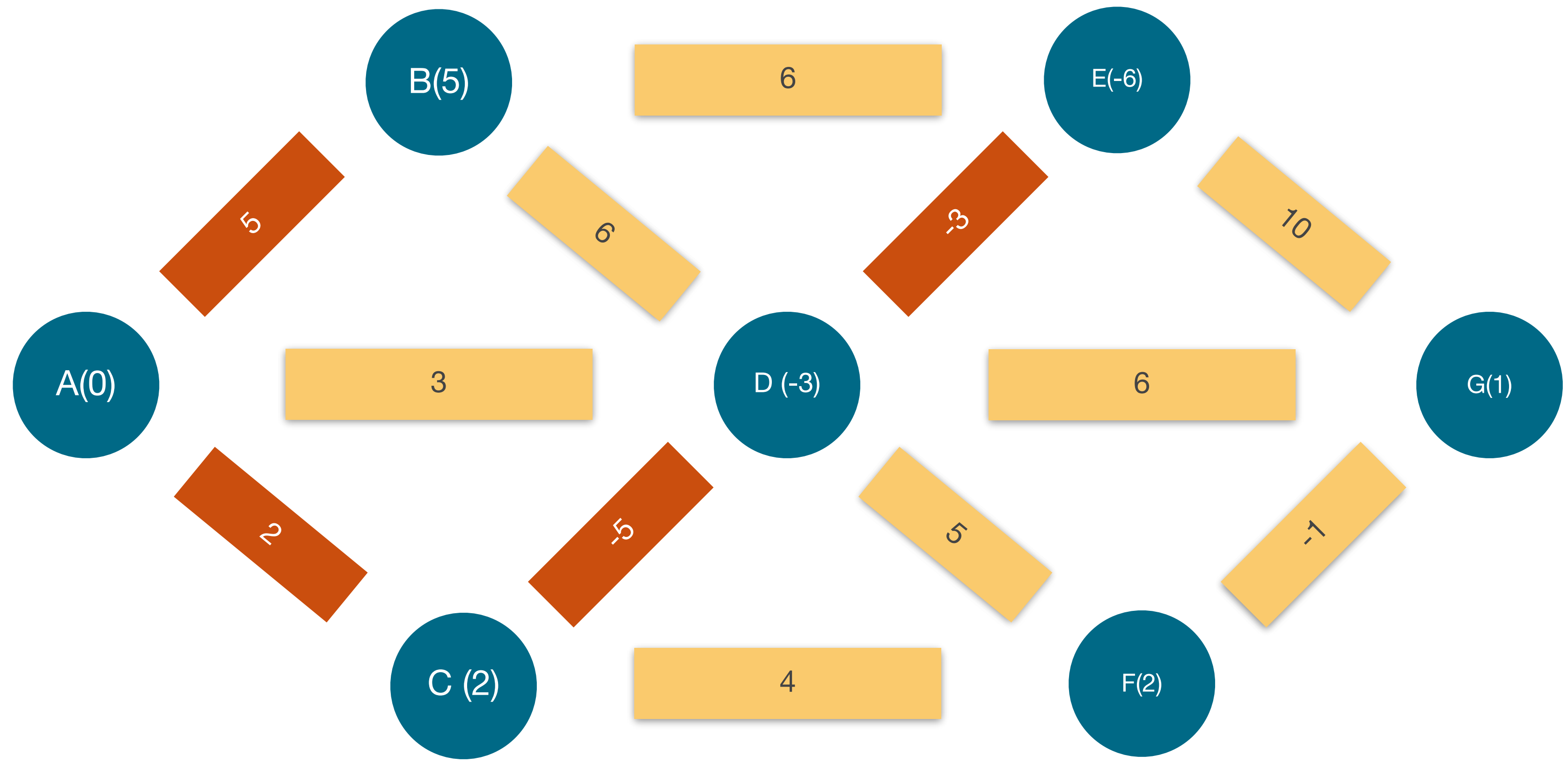
D,E : -3+(-3)=-6=**-6 —-> E(-6)**

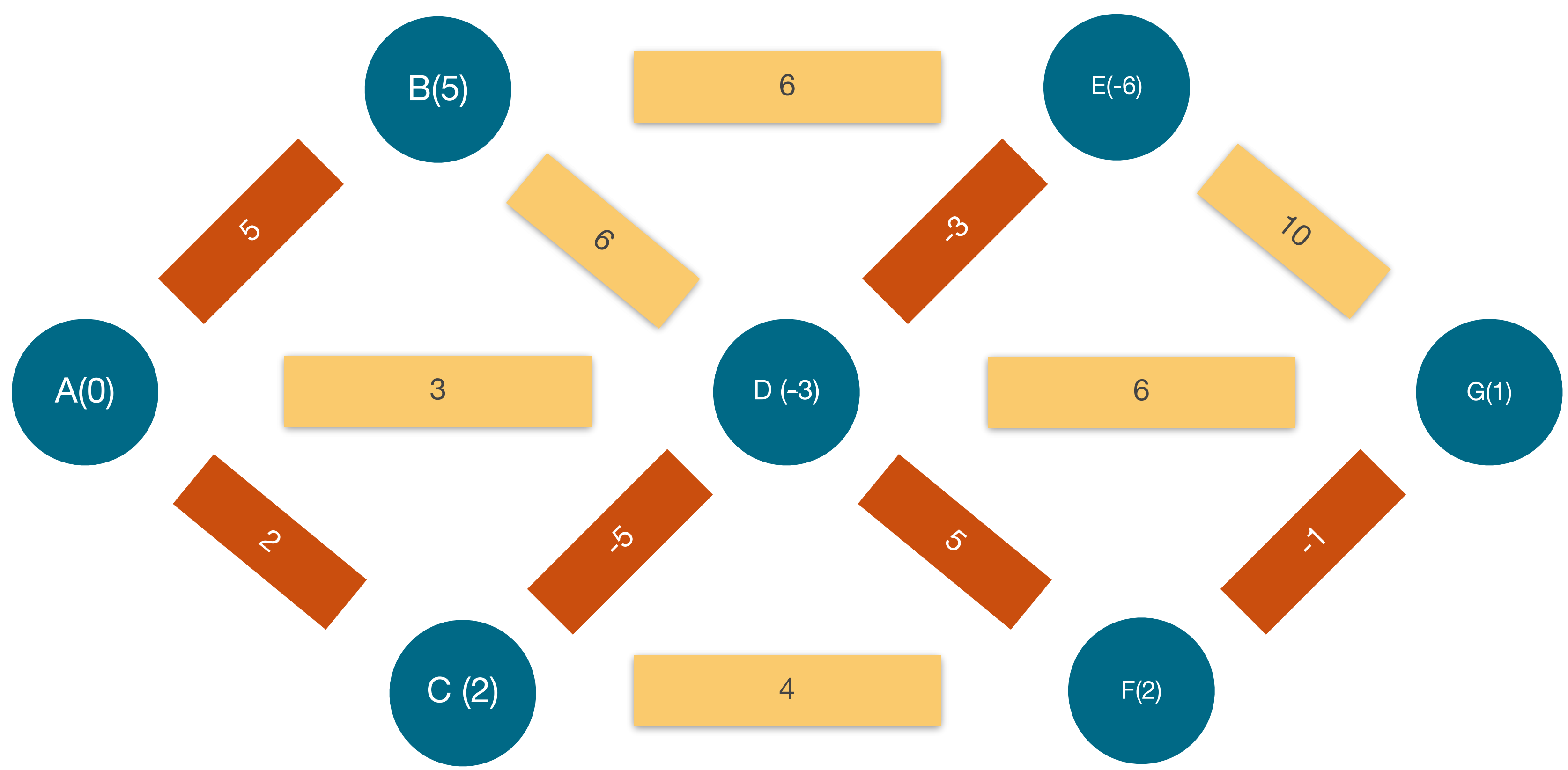
D,G : -3+6=3>**1 —-> G(1)**

D,F : -3+5=2=**2 —-> F(2)**

E,G : -6+10=4>**1 —-> G(1)**

F,G : 2+(-1)=1=**1 —-> G(1)**





Εφαρμογή αλγορίθμου Bellman-Ford Βήμα 3,4,5/6

Υπολογίζοντας τα βάρη για όλους τους κλάδους έχουμε:

A,B : 0+5=5=**5 —-> B(5)**

A,C : 0+2=2=**2 —-> C(2)**

A,D : 0+3=3>-**3 —-> D(-3)**

B,E : 5+6=11>**-6 —-> E(-6)**

B,D : 5+6=11>-**3 —-> D(-3)**

C,D : 2+(-5)=-3=-**3 —-> D(-3)**

C,F : 2+4=6>**2 —-> F(2)**

D,E : -3+(-3)=-6=**-6 —-> E(-6)**

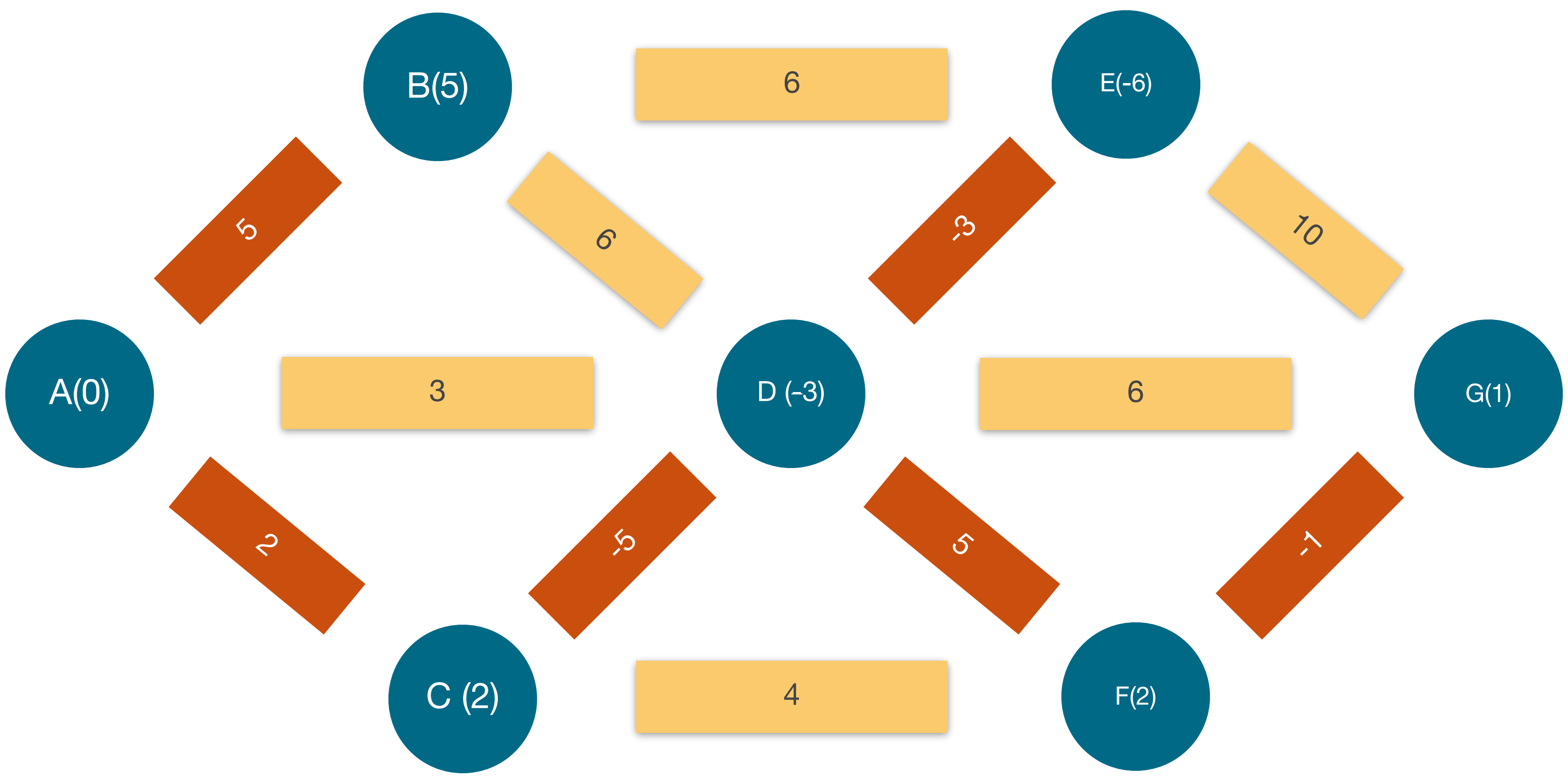
D,G : -3+6=3>**1 —-> G(1)**

D,F : -3+5=2=**2 —-> F(2)**

E,G : -6+10=4>**1 —-> G(1)**

F,G : 2+(-1)=1=**1 —-> G(1)**

Στην συγκεκριμένη περίπτωση φαίνεται ότι ο αλγόριθμος ολοκληρώθηκε με μόνο 2 επαναλήψεις και δεν φέρνει νέο αποτέλεσμα με τις επόμενες, έως τις 6.



Κατά συνέπεια μπορούμε να σταματήσουμε τον αλγόριθμο, βλέποντας τις συντομότερες διαδρομές από κόμβο σε κόμβο, με κεραμιδί χρώμα.

Εφαρμογή αλγορίθμου Bellman-Ford

Διαπιστώσεις

1. Η μικρότερη απόσταση από

    1. το A στο G είναι η απόσταση A-C-D-F-G με βάρος 1,

    2. το A στο E είναι η απόσταση A-C-D-E με βάρος -6.

2. Όλοι οι κόμβοι έχουν μόνιμες ετικέτες.

3. Όλες οι αποστάσεις από το A στους υπόλοιπους κόμβους έχουν σημανθεί με το μικρότερο βάρος.