Προηγμένα Θέματα Αλγορίθμων - 1η σειρά γραπτών ασκήσεων

> Εαρινό εξάμηνο 2021 - 2022
>
> Ον/μο : **Νίκος Μπέλλος**
> AM : **el18183**

DIRECTORY

Άσκηση 1 (Gradient Descent)

1. Να δείξετε για $\eta = \frac{B}{G\sqrt{T}}$ οτι ισχύει η σχέση για το $Regret_T$:

$Regret_T = \sum_{t=1}^{T} f_t(x_t) - \underset{x \in S}{min} \sum_{t=1}^{T} f_x(t) \leq \sum_{t=1}^{T} u_t(x_t - x^*)$

, όπου $x^* = min\{x : x \in S\}$

$u_t(x_t - x^*) = \frac{1}{2\eta}(||x_t - x^*||^2 - ||x_{t+1} - x^*||^2) + \frac{\eta}{2}||u_t||^2$

$\Rightarrow Regret_T \leq \frac{1}{2\eta}(||x_1 - x^*||^2) + \frac{\eta}{2}\sum_{t=1}^{T}||u_t||^2$

$\leq \frac{B^2}{2\eta} + \frac{\eta T G^2}{2} = BG\sqrt{T}$

2. Επιλέγουμε και πάλι βήμα $\eta_t = \frac{B}{G\sqrt{t}}$, διότι $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}$

3. Επιλέγουμε βήμα $\eta_t = \frac{1}{(a \cdot t)}$

   $\Rightarrow Regret_T \leq \sum_{t=1}^{T}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - a)||x_t - x^*||^2 + \sum_{t=1}^{T}\eta_t||u_t||^2$

   $\leq 0 + \frac{G^2 lnT}{\alpha}$

Άσκηση 2 (Simplex algorithm)

a) Simplex Algorithm

Initial Matrix

b	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	**-10**	**57**	**9**	**24**	**0**	**0**	**0**
0	0.5	-5.5	-2.5	9	1	0	0
0	0.5	-1.5	-0.5	1	0	1	0
1	1	0	0	0	0	0	1

🟠 columns : βασικές μεταβλητές

1. **Pivoting με επιλογή μεταβλητής ελαχίστου ανηγμένου κόστους (greedy)**

Chosen variable : X1

R1 = R1 + 20*R2
R2 = R2 + R2

R3 = R3 - R2
R4 = R4 - 2*R2

b	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	**0**	**-53**	**-41**	**204**	**20**	**0**	**0**
0	1	-11	-5	18	2	0	0
0	0	4	2	-8	-1	1	0
1	0	11	5	-18	-2	0	1

Chosen variable : X2

R1 = R1 + 53/4*R3
R2 = R2 + 11/4*R3
R3 = R3 - 3/4*R3
R4 = R4 - 11/4*R3

b	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	**0**	**0**	**-14.5**	**98**	**6.75**	**13.25**	**0**
0	1	0	0.5	-4	-0.75	2.75	0
0	0	1	0.5	-2	-0.25	0.25	0
1	0	0	-0.5	4	0.75	-2.75	1

Chosen variable : X3

R1 = R1 + 29*R2
R2 = R2 + R2
R3 = R3 - R2
R4 = R4 + R2

b	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	**20**	**0**	**0**	**-18**	**-15**	**93**	**0**
0	2	0	1	-8	-1.5	5.5	0
0	-1	1	0	2	0.5	-2.5	0
1	1	0	0	0	0	0	1

Chosen variable : X4

R1 = R1 + 9*R3
R2 = R2 + 4*R3
R3 = R3 - 0.5*R3
R4 = R4

b	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	20	9	0	0	-10.5	70.5	**0**
0	-2	4	1	0	0.5	-4.5	0
0	-0.5	0.5	0	1	0.25	-1.25	0
1	1	0	0	0	0	0	1

Chosen variable : X5

R1 = R1 + 21*R2
R2 = R2 + R2
R3 = R3 - 0.5*R2
R4 = R4

b	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	-22	93	21	0	0	-24	**0**
0	-4	8	2	0	1	-9	0
0	0.5	-1.5	-0.5	1	0	1	0
1	1	0	0	0	0	0	1

→ Παρατηρούμε ότι στο επόμενο βήμα θα διαλέξουμε για βασική μεταβλητή την X6 και θα βγάλουμε από τις βασικές μεταβλητές την X4 καταλήγοντας έτσι και πάλι στην ίδια βάση με την αρχή, δηλαδή την $\{x_5, x_6, x_7\}$. Άρα έχουμε κάνει κύκλο επιστρέφοντας στην ιδια κορυφή.

2. **Pivoting με επιλογή μεταβλητής ελαχίστου δείκτη**

📌 Παρατηρούμε ότι στην εφαρμογή του simplex με αυτό το κανόνα τα βήματα ειναι πανομοιότυπα με αυτά του προηγούμενου κανόνα, καθώς κάθε φορά θα διαλέγαμε την ίδια μεταβλητή, εκτός από το τελευταίο βήμα.

Επομένως συνεχίζουμε την εφαρμογή του simplex από το τελευταίο βήμα της 2.a.1 επιλέγοντας αυτή τη φορά το X1 αντί για το X6

Chosen variable : X1

R1 = R1 + 44*R3
R2 = R2 + 8*R3
R3 = R3 + R3
R4 = R4 - 2*R3

b	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	0	27	-1	44	0	20	**0**
0	0	-4	-2	8	1	-1	0
0	1	-3	-1	2	0	2	0
1	0	3	1	-2	0	-2	1

Chosen variable : X3

R1 = R1 + R4
R2 = R2 + 2*R4
R3 = R3 + R4
R4 = R4

b	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	0	30	0	42	0	18	1
0	0	2	0	4	1	-5	1
0	1	0	0	0	0	0	0
1	0	3	1	-2	0	-2	1

🏁 Όλα τα ανηγμένα κόστη είναι θετικά, άρα ο simplex βρήκε τη βέλτιστη λύση για αυτό το γραμμικό πρόβλημα που είναι η :

$[x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [0, 0, 1, 0, 0, 0, 0]$

---

b) Dual Problem | Complementary Slackness

Complementary Slackness συνθήκες :

1. $y_1 \cdot (0.5x_1 - 5.5x_2 - 25x_3 + 9x_4 + x_5) = 0$
2. $y_2 \cdot (0.5x_1 - 1.5x_2 - 0.5x_3 + x_4 + x_6) = 0$
3. $y_3 \cdot (x_1 + x_7 - 1) = 0$

Άσκηση 4 (Load Balancing)

$\sum_j x_{ij} = 1 \;, \forall \; task \; i$

$\sum_i x_{ij} p_{ij} \leq L^*, \; \forall \; machine \; j$

Άσκηση 5 (Vertex Cover)

Σκοπός είναι να αποδείξουμε ότι ο αλγόριθμος αυτός ειναι προσεγγιστικός με λόγο ρ=2.

Επεξήγηση αλγορίθμου :
Ο συγκεκριμένος αλγόριθμος για κάθε ακμή $e = (u, v)$, θέτει το βάρος της ακμής e ισο με το ελάχιστο των δυο βαρών των κορυφών, $\delta = min\{w(u), w(v)\}$ και μειώνει κάθε κορυφή κάτα δ. Επομένως, κάθε φορά τουλάχιστον μία κορυφή θα αποκτήσει μηδενικό βάρος. Αυτή η κορυφή θα μπεί και στο σύνολο C.

Πιο αναλυτικά :

1. Λόγω του 'while C no vertex cover' καταλαβαίνουμε ότι ο αλγόριθμος δεν θα σταματήσει μέχρι να βρεί κάποιο vertex cover. Πιο συγκεκριμένα, θα σταματήσει όταν δεν θα υπάρχουν πλέον ακάλυπτες ακμές που να μην έχουν κάποιο βάρος c(e)

2. Για κάθε κορυφή u που είναι μέρος του συνόλου C ισχύει ότι το αρχικό της βάρος $w(u)$ θα ειναι διαμοιρασμένο στις ακμές με τις οποίες συνδέεται, καθώς σε κάθε βήμα αφαιρείται η τιμή δ από την υπολοιπόμενη τιμή $t(u)$ και αυτή τείθεται ίση με $c(e)$. Δηλαδή :

   $(1)\ \forall u \in C : w(u) = \sum_{u \in e} c(e)$

   Στη χειρότερη περίπτωση όλες οι κορυφές θα βρίσκονται στο σύνολο C και άρα τη τιμή c(e) κάθε ακμής θα τη μετράμε δύο φορές, σύμφωνα με το τύπο (1). Προκύπτει :

   $||C|| \leq ||V|| \leq 2||E||$
   $\Rightarrow \sum_{u \in C} w(u) = \sum_{u \in C} \sum_{u \in e} c(e) \leq \sum_{u \in V} \sum_{u \in e} c(e) \leq 2 \sum_{e \in E} c(e)$

3. Διαισθητικά, λαμβάνοντας υπόψην το ερώτημα 2 μπορούμε να βρούμε το κάτω φράγμα δεδομένου ότι κάθε ακμή έχει μόνο 1 κορυφή στο σύνολο C και τότε δεν θα διπλομετράμε τη τιμή $c(e)$.

   Χρησιμοποιώντας duality για να αποδείξουμε το παραπάνω :

   Για κάθε ακμή θα πρέπει να συμπεριλάβουμε τουλάχιστον ένα άκρο της στο σύνολο C , ώστε να πετύχουμε το vertex cover. Χρησιμοποιόντας LP duality προκύπτουν τα παρακάτω :

   Min problem : Έχουμε ένα σύνολο κορυφών και θέλουμε από αυτές που θα πάρουμε να έχουν όσο το δυνατό μικρότερο άθροισμα βαρών (για κάθε ακμή πέρνουμε τουλάχιστον μία κορυφή)

   $min :\ \sum\ w^T x$

   $\forall (u, v) \in E,\ x_u + x_v \geq 1\ \ , x_i \in \{0, 1\}$

   Max problem : Θέλουμε να μεγιστοποιήσουμε όσο γίνεται το άθροισμα των βαρών στις ακμές

   $max :\ 1 \cdot y$

   $\forall u \in V,\ \sum_{e \in E, u \in e} y_e \leq w(u)$

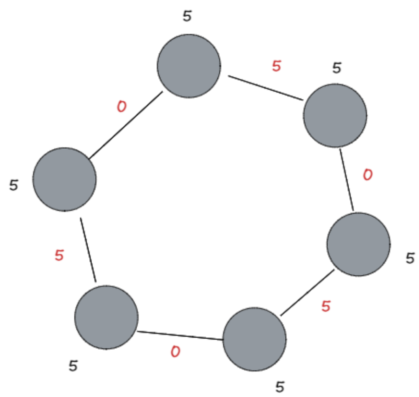
   $\overset{y_e = c(e)}{\Rightarrow} \forall u \in V,\ \sum_{e \in E, u \in e} c(e) \leq w(u)$

   Πόρισμα :

   Έστω ότι έχουμε μία λύση x^* του original LP. Επειδή η λύση αυτή δεν είναι χειρότερη από αυτή του min problem και επειδή κάθε λύση του δϋικού είναι μικρότερη από όλες τις λύσεις του min τότε θα ισχύει :

   $\sum_{e \in E} c(e) \leq x^*$

<u>**ΠΑΡΑΔΕΙΓΜΑ ΓΡΑΦΟΥ ΓΙΑ ρ=2**</u>



Για ένα γράφο όπου είναι κύκλος άρτιου μήκους και όλες οι κορυφές έχουν το ίδιο βάρος μπορούμε να δούμε στο παραπάνω διάγραμμα ότι αποτελεί ένα tight example για το λόγο ρ=2 του παραπάνω προσεγγιστικού αλγορίθμου

Άσκηση 6 (Weighted Set Cover)

Σύντομη επεξήγηση του αλγορίθμου :

Ο αλγόριθμος ξεκινάει με ένα ακάλυπτο Set C από n σημεία με στόχο να τα καλύψει όλα. Σε κάθε βήμα ταξινομεί τα διαθέσιμα Sets ως προς τον λόγο $cost(S_i)/|S - C|$ το οποίο αντιπροσωπεύει το κόστος κάθε επιμέρους σημείου που θα προσθέσουμε στο C αν επιλέξουμε το συγκεκριμένο Set. Από αυτή τη ταξινόμηση διαλέγει το Set με το μικρότερο λόγο.

📌 **Παρατήρηση** : Σε κάθε βήμα το κάθε set θα έχει το πολύ όσα σημεία είχε και προηγουμένως. Επομένως ο λόγος που είδαμε προηγουμένως μπορεί μόνο να αυξάνει (καθώς μειώνεται ο παρονομαστής) και άρα σε κάθε βήμα ο καλύτερος λόγος που μπορούμε να διαλέξουμε θα είναι μεγαλύτερος από αυτό του προηγούμενου βήματος. Καθώς αυτός ο λόγος αναπαριστά την αξία των επιμέρους στοιχείων, τα στοιχειά που θα βάζουμε σε κάθε βήμα θα μας "κοστίζουν" περισσότερο από αυτά που βάλαμε στο προηγούμενο.

Όταν διαλέγουμε ένα σύνολο, τοποθετούμε ένα κόστος c_e σε κάθε στοιχείο αυτού του συνόλου. Αν το σύνολο έχει $|S_i|$ στοιχεία, το καλύτερο κόστος που μπορεί να πάρει κάθε στοιχείο είναι το $cost(S_i)/|S_i|$, ενώ το χειρότερο είναι $cost(S_i)/1$. Εμείς υποθέτουμε ότι όταν διαλέγουμε ένα στοιχείο i από το 1 μέχρι το $|S_i|$ τότε τα πρώτα i δεν ανήκουν στο σύνολο C και θα πάρουν τη τιμή $cost(S_i)/i$, ενώ τα υπόλοιπα $|S_i| - i$ ανήκουν σε κάποιο άλλο σύνολο το οποίο έχουμε βάλει ήδη στο S και καθώς το έχουμε διαλέξει πιο πριν θα έχει και μικρότερη τιμή άνα στοιχείο.

Προκύπτει η παρακάτω ανισότητα :

$$\sum_{e \in S_i} c_e \leq \frac{cost(S_i)}{|S_i|} + \frac{cost(S_i)}{|S_i|-1} + \frac{cost(S_i)}{|S_i|-2} + ... + \frac{cost(S_i)}{1} = cost(S_i) \cdot H_{|S_i|}$$

Γενικέυοντας για όλα τα S_i και παίρνοντας ως άνω φράγμα των $H_{|S_i|}$ το $H_{|S_{max}|}$ προκύπτει :

$$\sum_{e \in S} c_e \leq \sum_{S_i \in OPT} \sum_{e \in S_i} c_e \leq \sum_{S_i \in OPT} cost(S_i) \cdot H_{|S_i|} \leq \sum_{S_i \in OPT} cost(S_i) \cdot H_{|S_{max}|}$$
$$= OPT \cdot H_{|S_{max}|}$$

Άσκηση 7 (Cardinality Set Cover, Weighed Set Cover)

α) **Στόχος προβλήματος (Cardinality Set Cover) :**

Έχοντας ένα σύνολο από σημεία $U = \{u_1, u_2..., u_n\}$ και ένα σύνολο από υποσύνολα του U, $S = \{S_1, S_2, ..., S_m\}$ θέλουμε να καλύψουμε το U με όσο το δυνατόν λιγότερα στοιχεία του S

* Ταυχόχρονα, γνωρίζουμε ότι κάθε σημείο στο U μπορεί να ανήκει το πολύ σε f στοιχεία του S. (Επομένως δεν μπορούμε να καλύψουμε πάω από f φορές κάποιο σημείο)

Λύση :

Για να προσπαθήσουμε να σχεδιάσουμε κάποιον απλό f-προσεγγιστικό αλγόριθμο για το συγκεκριμένο πρόβλημα θα κατατρέξουμε στο πρόβλημα του Cardinality Vertex Cover. Εκεί ουσιαστικά διαλέγουμε κορυφές αντί για Sets και όπως στο παρόν πρόβλημα μία κορυφή στο U μπορεί να αντιστοιχηθεί με το πολύ f από τα αντικείμενα που θέλουμε να ελαχιστοποιήσουμε, έτσι στο πρόβλημα του Vertex Cover κάθε ακμή αντιστοιχείται με το πολύ 2 κορυφές (οι οποίες είναι το αντικείμενο ελαχιστοποίησης).

Στο Vertex Cover η λύση είναι να βρούμε ένα Maximal Matching για τις κορυφές τους επιλέγοντας τυχαία κάθε φορά μία ακμή και προσθέτωντας και τις δύο κορυφές της ακμής στο Set μας.

Αναλογικά, η λύση μας για το Cardinality Set Cover θα αποτελεί την εφαρμογή του Maximal Matching αλλά για σημεία αυτή τη φορά. Πιο αναλυτικά, θα διαλέγουμε κάθε φορά ένα τυχαίο σημείο και θα προσθέτουμε στο τελικό Set μας όλα τα υποσύνολα στα οποία ανήκει το σημείο αυτό (το πολύ f). Έπειτα, θα διαλέγουμε τυχαία ένα άλλο σημείο με το περιορισμό να μην έχει κανένα κοινό Set με όσα σημεία έχουμε διαλέξει μέχρι τώρα και θα επαναλαμβάνουμε την ίδια διαδικασία. Ξέρουμε ότι ο αλγόριθμός μας θα επιστρέψει ένα Set Cover διότι θα σταματήσει μόνο όταν δεν υπάρχουν άλλα σημεία τα οποία να μην ανήκουν σε κανένα από τα επιλεγμένα Sets S_i

$\rightarrow \forall i, j \in SOL \; : S_i \neq S_j, \quad i \in S_i, j \in S_j$

Απόδειξη ορθότητας :

Κατά τη διαδικασία του Maximal Matching παρατηρήσαμε ότι όλα τα στοιχεία που διαλέξαμε ήταν ασυσχέτιστα μεταξύ τους. Άρα αν διαλέξαμε συνολικά K σημεία τότε αυτός ήταν και ο ελάχιστος αριθμός από υποσύνολα του S που θα μπορούσαμε να διαλέξουμε.
Δηλαδή $OPT = K$

Όμως για κάθε σημείο που διαλέγουμε συμπεριλαμβάνουμε στη λύση μας μέχρι το πολύ άλλα f-1 στοιχεία του S. Γιαυτό η λύση μας δεν μπορεί να ξεπερνά το λόγο προσέγγισης f
Δηλαδή $SOL \leq f \cdot K = f \cdot OPT$

Tight Example

Το πιο εύκολα αντιληπτό παράδειγμα που μπορούμε να δώσουμε είναι να αντικαταστήσουμε τα σημεία του U με ακμές και τα στοιχεία του S με κορυφές ενός γράφου. Εκεί γνωρίζουμε ότι το f αντιστοιχεί στο 2 και ταυτόχρονα δεν μπορούμε να επιλέξουμε περισσοτερες από $2 \cdot OPT$ κορυφές, διότι δεν υπάρχουν. Με πιο απλά λόγια το tight example είναι ένας πλήρης γράφος στον οποίο εφαρμόζουμε το min vertex cover. Στη χειρότερη με max maching κάθε ακμή θα έχει και τις δύο κορυφές της να ανήκουν στη λύση μας και άρα ο αλγόριθμός μας φράσεταιαπό λόγο προσέγγισης 2 (δηλαδή f στη γενική περίπτωση).

β) **Στόχος προβλήματος (Weighted Set Cover) :**

Ίδιος με αυτό του προβλήματος α με τη παραλαγή ότι εδώ κάθε Set εχει ένα βάρος W_i και θέλουμε να ελαχιστοποιήσουμε το άθροισμα βαρών των συνόλων S_i που θα διαέξουμε και όχι τη πληθικότητά τους.

Λύση :

Επειδή το Cardinality Set Cover αποτελεί υποπερίπτωση του Weighted Set Cover δεν μπορούμε να εφαρμόσουμε τον παραπάνω αλγόριθμο του ερωτήματος α, διότι πριν κάθε Set είχε σταθερό κόστος 1 για να μπεί στη λύση μας και έτσι ήταν εύκολα συσχετίσιμο με τη ποσότητα f που αντιστοιχούσε στο πόσα Sets περιέχουν το ίδιο σημείο.

Έτσι, καταφέγουμε στη μεθοδολογία που είδαμε και στην άσκηση 6. Την επιλογή δηλαδή με βάση το συντελεστή $cost(S_i)/|S_i|$.

Ξεκινάμε αρχικά με ολόκληρο το σετ S. Μετράμε το λόγο $C_{S_i} = cost(S_i)/|S_i|$ για κάθε σύνολο και διαλέγουμε το μικρότερο από αυτά : $W_j = \min_{S_i \in S}\{C_{S_i}\}$, όπου j το αντίστοιχο βήμα στο οποίο εφαρμόζουμε αυτό τον αλγόριθμο. → S_j :
$selected\ set\ in\ step\ j, W_j : weight/point\ in\ selected\ set.$

Το σύνολο αυτο το προσθέτουμε στη λύση μας (έστω S') και το αφαιρούμε από το αρχικό σύνολο S, καθώς επίσης αφαιρούμε και τα σημεία από όλα τα υπόλοιπα Sets που έμειναν.

Η προσθήκη αυτή μας κοστίζει W_j για κάθε νέο σημείο που βάλαμε στο S' και θα ισχύει ότι

$(1)\ W_j \cdot |S_j| \geq cost(S_j)$

. Δηλαδή, επειδή σε κάθε βήμα αφαιρούμε σημεία από τα διάφορα Sets η επιμέρους τιμή των σημείων στο επόμενο Set που θα διαλέξουμε θα έχει αυξηθεί.

Απόδειξη ορθότητας :

Κάθε σημείο μπορεί να συμμετάσχει σε f το πολύ σύνολα →

$\sum_{s \in S} |s| \leq f \cdot |U| \Rightarrow c \cdot \sum_{s \in S} |s| \leq f \cdot c \cdot |U| \Rightarrow cost(S) \leq f \cdot c \cdot |U|$

Επίσης, από την (1) ξέρουμε ότι το συνολικό κόστος της λύσης μας S' θα είναι πιο ακριβό από το να δίναμε την ελάχιστη επιμέρους τιμή σε όλα τα στοιχεία του U. Δηλαδή :

$$\sum_{S_j \in S'} W_j \cdot |S_j| \geq \sum_{S_j \in S'} cost(S_j)$$
$$\Rightarrow cost(S') \geq cost(OPT) \geq c \cdot |U|$$
$$,\ c = min\{S_i/|S_i|\}$$

Προκύπτει ότι $cost(S) \leq f \cdot c \cdot |U| \leq f \cdot cost(S') \Rightarrow cost(S) \leq f \cdot cost(S')$

→ Το S' μπορεί να αναπαριστά τη βέλτιστη λυση και το S να αναπαριστά τη δική μας λύση η οποία θα είναι να πάρουμε όλα τα πιθανά σύνολα. Ακόμα και τότε που είναι η χειρότερη περίπτωση το συνολικό κόστος της λύσης μας δεν θα είναι μεγαλύτερο από το OPT*f

$cost(S') \leq f \cdot cost(OPT)$

Αρα ο αλγόριθμος είναι f-προσεγγιστικός

Άσκηση 8 (Metric TSP)

α) Επεξήγηση & απόδειξη των 2 αλγορίθμων

Αλγόριθμος Α

1. Φτιάξε το MST του γράφου
2. Διπλασίασε τις ακμές (ώστε όλοι οι κόμβοι να έχουν άρτιο βαθμό)
3. Αφαίρεση του s-t path
4. Βρες Hamilton path s-t

$\rightarrow SOL_1 \leq 2OPT_{s,t} - c_{s,t}$

📌 Το $c_{s,t}$ αντιπροσωπεύει το μήκος του μονοπατιού που συνδέει το s και το t στο MST

Αλγόριθμος Β

1. Φτιάξε το MST του γράφου
2. Βρές perfect matching στους κόμβους περιττού βαθμού
3. Ένωσε το MST με το Perfect matching
4. Βρες Hamilton path s-t

$\rightarrow SOL_2 \leq (3OPT_{s,t} + c_{s,t})/2$

📌 Το $c_{s,t}$ αντιπροσωπεύει το βάρος της ακμής που συνδέει το s και το t

Συνδυασμός των δύο αλγορίθμων :

$\rightarrow min\{SOL_1, SOL_2\} \leq \frac{5}{3}OPT_{s,t}$

Παρατηρούμε ότι το $c_{s,t}$ παίζει καθοριστικό ρόλο στο ποιά από τις δύο λύσεις θα διαλέξουμε. Η πρώτη λύση έχει ώς άνω φράγμα το $2OPT$ και μειώνεται ανάλογα με τη τιμή του $c_{s,t}$, ενώ η δεύτερη έχει ως κάτω φράγμα το $\frac{3}{2}OPT$ και αυξάνεται ανάλογα με το $c_{s,t}$.

Διακρίνουμε 2 περιπτώσεις :

1. $SOL_1 \leq SOL_2 \Rightarrow 2OPT_{s,t} - c_{s,t} \leq (3OPT_{s,t} + c_{s,t})/2 \Rightarrow c_{s,t} \geq \frac{OPT}{3}$
   $\overset{c_{s,t}=OPT/3}{\Rightarrow} SOL = SOL_1 \leq 2OPT_{s,t} - \frac{OPT}{3} = \frac{5}{3}OPT$
2. $SOL_1 \geq SOL_2 \Rightarrow 2OPT_{s,t} - c_{s,t} \geq (3OPT_{s,t} + c_{s,t})/2 \Rightarrow c_{s,t} \leq \frac{OPT}{3}$
   $\overset{c_{s,t}=OPT/3}{\Rightarrow} SOL = SOL_2 \leq \frac{3}{2}OPT + \frac{1}{6}OPT = \frac{5}{3}OPT$

   Επομένως σε κάθε περίπτωση προκύπτει ο λόγος $\frac{5}{3}OPT$

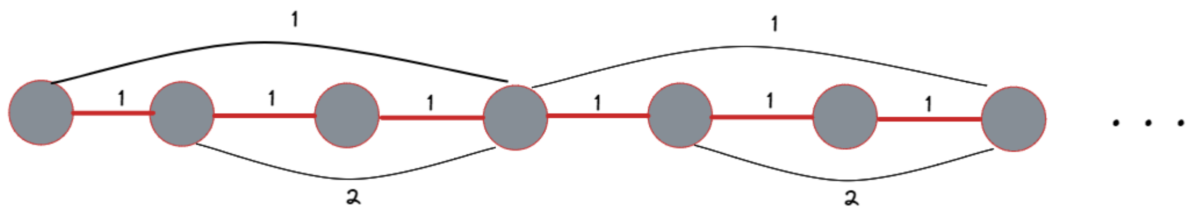
β) **Tight example**

**Στα παρακάτω παραδείγματα θα υποθέσουμε ότι οι γράφοι είναι πλήρης*

Θεωρούμε το παρακάτω διάγραμμα σαν ένα tight example το οποίο στη χειρότερη περίπτωση θα παρουσιάζει λόγο προσέγγισης ίσο με $\rho = 5/3$

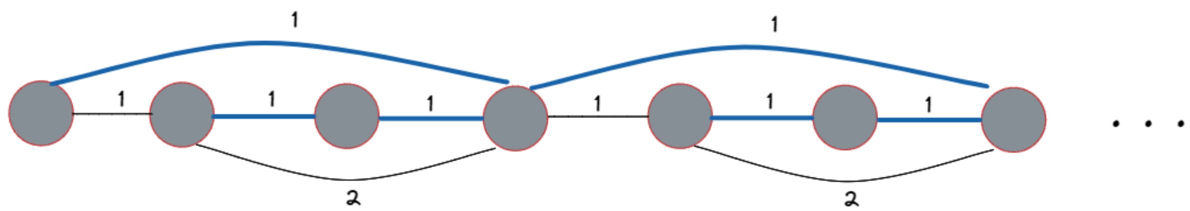
Έστω ότι έχουμ n κόμβους. Τότε το OPT κόστος θα είναι n-1 για το hamilton path.

Optimum Path



Και στους δύο αλγορίθμους το πρώτο βήμα είναι να κατασκευάσουμε το MST του γράφου το οποίο φαίνεται στο παρακάτω διάγραμμα

MST (κόστος n)



Αλγόριθμος Α

Διπλασιάζοντας τις ακμές παίρνουμε συνολικό κόστος SOL=2*cost(MST)=2*n

Στη χειρότερη περίπτωση το μονοπάτι s-t που θα επιλέξουμε να αφαιρέσουμε από το γράφημα θα είναι αυτό με το μικρότερο συνολικό κόστος, δηλαδή η ακμή που ενώνει τους κόμβους άνα 3.

Σε αυτή τη περίπτωση το $c_{s,t}$ θα ισούται με n/3=OPT/3. Και επομένως η λύση θα ειναι :

$SOL = 2 * cost(MST) - n/3 = 2 * OPT - OPT/3 = 5/3OPT$

Αλγόριθμος Β

Μετά το MST παίρνουμε όλες τις κορυφές που έχουν περιττό βαθμό και εκτελούμε min cost perfecto matching.

Για κάθε 3άδα λοιπόν, προστίθεται μία ακόμα ακμή μήκους 2 και η τελική λύση είναι ίση με

$SOL = OPT + 2/3OPT = 5/3OPT$