ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Προσεγγιστικοί Αλγόριθμοι για το Πρόβλημα της Χρονοδρομολόγησης Αποστολής Δεδομένων σε Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΕΡΕΜΙΑΔΗ ΛΟΥΚΑ

Επιβλέπων : Φώτω Αφράτη
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Οκτώβριος 2009

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Προσεγγιστικοί Αλγόριθμοι για το Πρόβλημα της Χρονοδρομολόγησης Αποστολής Δεδομένων σε Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΕΡΕΜΙΑΔΗ ΛΟΥΚΑ

Επιβλέπων : Φώτω Αφράτη
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 23 η Οκτωβρίου 2009.

....	....	....
Φώτω Αφράτη Καθηγήτρια Ε.Μ.Π.	Νικόλαος Παπασπύρου Επικ. Καθηγητής Ε.Μ.Π.	Δημήτριος Φωτάκης Λέκτορας Ε.Μ.Π.

Αθήνα, Οκτώβριος 2009

....

ΙΕΡΕΜΙΑΔΗΣ ΛΟΥΚΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ένα πρόβλημα το οποίο προέκυψε στις ασύρματες επικοινωνίες, είναι αυτό της χρονοδρομολόγησης της αποστολής και λήψης δεδομένων (Preemptive Bipartite Scheduling problem-PBS). Στο πρόβλημα αυτό έχουμε ένα σύνολο από πομπούς, ένα σύνολο από δέκτες και ένα σύνολο από δεδομένα, τα οποία θέλουμε να στείλουμε από τους πομπούς στους δέκτες. Για κάθε δεδομένο γνωρίζουμε εκ των προτέρων τον χρόνο που χρειάζεται για να σταλεί. Ένας πομπός δεν μπορεί να στέλνει ταυτόχρονα δεδομένα σε δύο ή περισσότερους δέκτες, και αντίστοιχα ένας δέκτης δεν μπορεί να λαμβάνει ταυτόχρονα δεδομένα από δύο ή περισσότερους πομπούς. Προσπαθούμε να βρούμε μία συγκεκριμένη σειρά αποστολής των δεδομένων, έτσι ώστε να επιτύχουμε την αποστολή όλων των δεδομένων στον ελάχιστο δυνατό χρόνο. Η αποστολή αυτή γίνεται χωρίζοντας τα δεδομένα σε κατάλληλες ομάδες δεδομένων, αποτελούμενες από δεδομένα που μπορούν να αποσταλούν ταυτόχρονα. Ο καθορισμός του συστήματος, ώστε να αποσταλεί μία ομάδα αποστολής θεωρούμε ότι απαιτεί γνωστό εκ των προτέρων χρόνο. Η βέλτιστη λύση του προβλήματος, είναι εκείνη που ελαχιστοποιεί τον συνολικό χρόνο αποστολής των δεδομένων, δηλαδή το άθροισμα των χρόνων αποστολής και των χρόνων καθορισμού των ομάδων αποστολής. Επειδή η εύρεση της βέλτιστης λύσης συχνά είναι ιδιαίτερα δύσκολη και απαιτεί πολλές φορές περισσότερο χρόνο από εκείνον που διαθέτουμε, οδηγούμαστε στη χρήση προσεγγιστικών αλγορίθμων για την επίλυση του προβλήματος αυτού. Οι προσεγγιστικοί αλγόριθμοι βρίσκουν μία λύση, σε πολυωνυμικό χρόνο, η οποία εν γένει δεν είναι η βέλτιστη, αλλά μας εξασφαλίζουν πόσο «κοντά» στην βέλτιστη είναι, στην χειρότερη περίπτωση. Στην παρούσα εργασία, παρουσιάζουμε τρείς τέτοιους αλγορίθμους. Στη συνέχεια τους υλοποιούμε, και πραγματοποιούμε μια σειρά πειραμάτων , προκειμένου να βγάλουμε χρήσιμα συμπεράσματα για την απόδοσή τους.

Λέξεις Κλειδιά: <<προσεγγιστικοί αλγόριθμοι, χρονοδρομολόγηση αποστολής δεδομένων, Preemptive Bipartite Scheduling Problem, ελαχιστοποίηση του χρόνου αποστολής >>

Abstract

The Preemptive Bipartite Scheduling problem (PBS), has aroused in satellite communication systems. We consider, having a set of transmitters (T), a set of receivers (R), and a set of messages, that we want to be sent from (T) to (R). Preemption of the messages during the transmission is allowed. We usually, prefer describing this problem, in terms of a bipartite weighted graph. One set of nodes denotes the transmitters, the other set the receivers, and edges are communication tasks, where the weight of an edge is the time required for the specific transmission. It is implied that each transmitter and receiver can only send or receive one message at a time, but messages between different transmitters and receivers can overlap. Thus, a simultaneous transmission of messages between several transmitters and receivers is a matching of the graph. We consider, that there is a cost for the set-up of each parallel transmission step, referred as set-up delay. The objective is to find a scheduling for parallel transmissions, minimizing the total transmission time. Unfortunately, finding the optimal solution of this problem, is usually difficult, and in practice non realizable. Thus, we use approximation algorithms, for solving it. An approximation algorithm produces feasible solutions, in polynomial time, that are not, in general, the optimal, but are "close" to the optimal. In the current work, we present three approximation algorithms for the PBS problem. After implementing these algorithms, we run some test for comparison, and we present the experimental results.

Keywords: <<approximation algorithms, scheduling in switching networks, preemptive bipartite scheduling problem, minimization of transmission time>>

Πίνακας περιεχομένων

1

Εισαγωγή

1.1 Εισαγωγή-Γενική Περιγραφή

Στην παρούσα εργασία θα παρουσιάσουμε και έπειτα θα συγκρίνουμε τρεις προσεγγιστικούς αλγορίθμους, οι οποίοι επιλύουν το πρόβλημα της διμερούς χρονοδρομολόγησης σε δίκτυα αποστολής και λήψης δεδομένων. Οι προσεγγιστικοί αλγόριθμοι χρησιμοποιούνται ευρέως στην επιστήμη των υπολογιστών για την επίλυση πολύπλοκων και δυσεπίλυτων προβλημάτων. Το πρόβλημα στο οποίο αναφερόμαστε στην παρούσα εργασία, είναι ένα ρεαλιστικό πρόβλημα με εφαρμογές σε πολλούς τομείς, και συνεπώς η εύρεση αλγορίθμων που βρίσκουν λύση σε σύντομο χρονικό διάστημα, είναι ουσιώδης για τον κόσμο της τεχνολογίας. Για τον λόγο αυτό, θα προσπαθήσουμε στο πειραματικό τμήμα της εργασίας, να δημιουργήσουμε περιπτώσεις τέτοιες ώστε να καλύψουμε όλες τις πιθανές εφαρμογές και να εξάγουμε ασφαλή συμπεράσματα για την αποδοτικότητα των αλγορίθμων.

1.2 Οργάνωση κειμένου

Στο σημείο αυτό παρουσιάζουμε συνοπτικά τα θέματα τα οποία καλύπτουν τα κεφάλαια που ακολουθούν. Στο Κεφάλαιο 2 παρουσιάζονται κάποιες βασικές θεωρητικές έννοιες, με τις οποίες ο αναγνώστης πρέπει να εξοικειωθεί προκειμένου να κατανοήσει την υπόλοιπη εργασία. Στο κεφάλαιο 3 αναφέρουμε τα βασικά στοιχεία των προσεγγιστικών αλγορίθμων. Οι προσεγγιστικοί αλγόριθμοι είναι η βασική έννοια με την οποία θα ασχοληθούμε στην παρούσα εργασία. Στο κεφάλαιο 4 παρουσιάζουμε το πρόβλημα PBS, καθώς και τρείς προσεγγιστικούς αλγορίθμους που το επιλύουν. Στο κεφάλαιο 5 αναφέρουμε τα πειράματα σύγκρισης των αλγορίθμων που πραγματοποιήσαμε και παρουσιάζουμε τα αποτελέσματα των πειραμάτων αυτών. Τέλος, στο κεφάλαιο 6, καταγράφουμε τα συμπεράσματα στα οποία οδηγηθήκαμε από την όλη μελέτη που κάναμε.

2

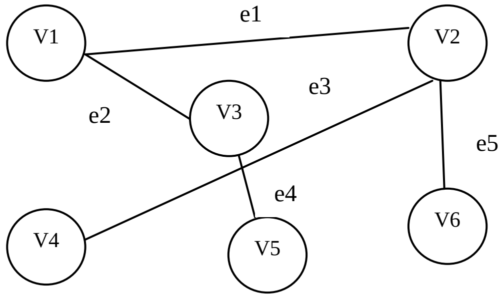
Θεωρητικό Υπόβαθρο

Στο κεφάλαιο αυτό θα περιγράψουμε χρήσιμες έννοιες-θέματα με τις οποίες, ενδεχομένως ο αναγνώστης να πρέπει να εξοικειωθεί προτού συνεχίσει στην ανάγνωση των επόμενων κεφαλαίων.

2.1 Στοιχεία Θεωρίας Γραφημάτων

2.1.1 Εισαγωγή- Ορισμοί

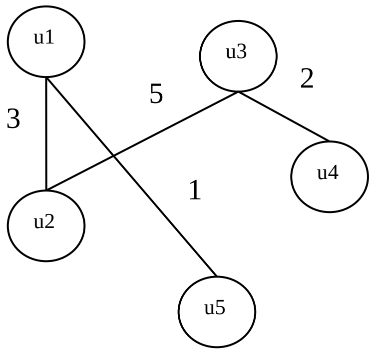
Ένας γράφος (Graph) G είναι ένα ζεύγος (V,E), όπου V είναι ένα σύνολο από κορυφές (Vertices) και E είναι ένα σύνολο από ακμές (Edges) που συνδέουν δύο κορυφές μεταξύ τους. Ένας γράφος απεικονίζεται σχηματικά με τον τρόπο που φαίνεται στο σχήμα που ακολουθεί.



Σχήμα 2.1 : απεικόνιση τυχαίου γράφου.

Ο γράφος του σχήματος έχει 6 κορυφές τις v1,v2,v3,v4,v5,v6 και 5 ακμές τις e1,e2,e3,e4,e5.

Γράφο με βάρη (weighted graph) αποκαλούμε έναν γράφο G(V,E) στον οποίο, σε κάθε ακμή (u,v) є E, όπου u, v є V, έχουμε αποδώσει μια τιμή, την οποία ονομάζουμε κόστος ή βάρος της ακμής C[u,v].



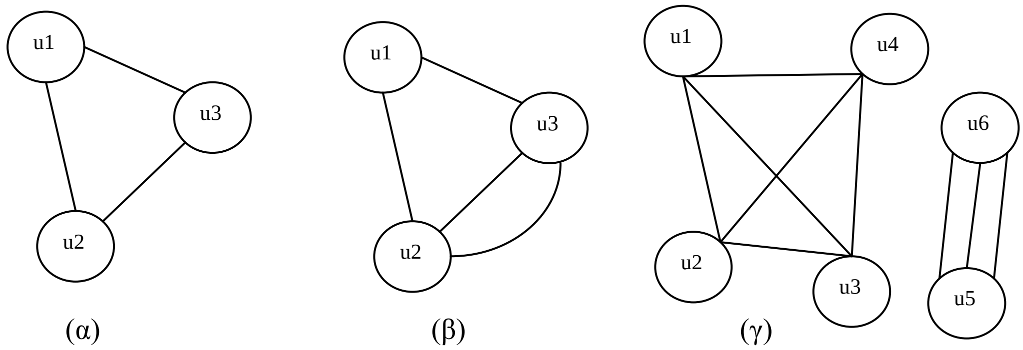
Σχήμα 2.2: απεικόνιση τυχαίου γράφου με βάρη.

Στον γράφο του σχήματος έχουμε C[u1,u2]= 3 , C[u2,u3]= 5, C[u3,u4]= 2, C[u1,u5]= 1.

2.1.2 Πλήρης Γράφος- Κανονικός Γράφος

Δύο κορυφές ενός γράφου που ενώνονται με μία ακμή ονομάζονται γείτονες ή γειτονικές κορυφές (neighbors). Ένας γράφος στον οποίο κάθε κορυφή γειτονεύει με όλες τις υπόλοιπες κορυφές ονομάζεται πλήρης γράφος (complete graph).

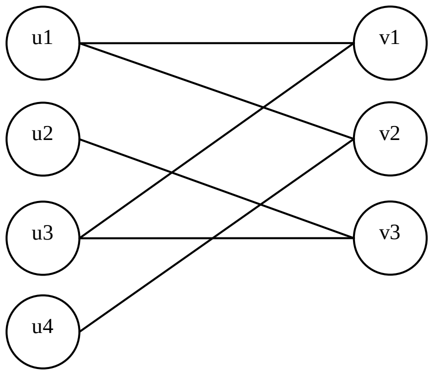
Βαθμός (degree) μιας κορυφής ονομάζεται ο αριθμός των ακμών που ενώνονται με αυτήν. Ο βαθμός μία κορυφής u σε έναν γράφο G συμβολίζεται με d(u). Ο μέγιστος βαθμός όλων των κορυφών ενός γράφου G συμβολίζεται με Δ(G), ενώ ο ελάχιστος με δ(G). Αν για κάποιο γράφο G ισχύει Δ(G) = δ(G) = κ τότε ο γράφος ονομάζεται κ-Κανονικός (κ-Regular).



Σχήμα 2.3 : (α) γράφος 2-κανονικός-πλήρης, (β) γράφος πλήρης- όχι κανονικός, (γ) γράφος 3-κανονικός- όχι πλήρης.

2.1.3 *Διμερής Γράφος*

Διμερής γράφος (bipartite graph) ονομάζεται ένας γράφος G(V,E) στον οποίο το σύνολο των κορυφών V έχει χωριστεί σε δύο σύνολα V και U, έτσι ώστε για κάθε ακμή (u,v) є E να ισχύει u є V και v є U ή u є U και v є V.

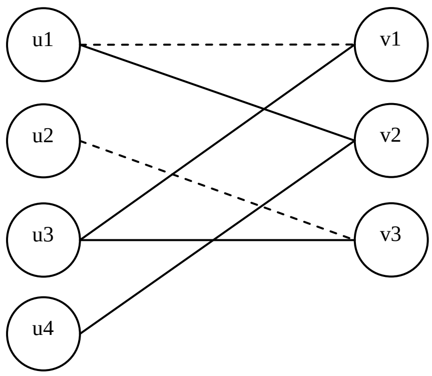


Σχήμα 2.4 : απεικόνιση διμερούς γράφου.

2.1.4 *Matchings σε διμερή γραφήματα*

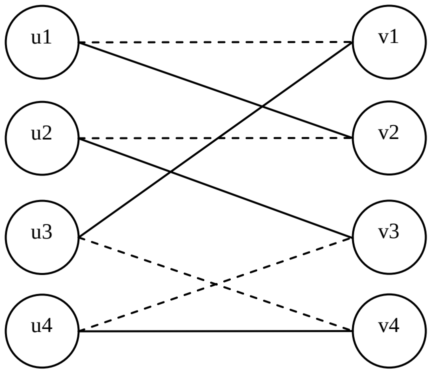
Matching σε έναν γράφο G είναι ένα σύνολο από ακμές του γράφου που δεν έχουν κοινές κορυφές. Δηλαδή για το γράφο G(V,E), ισχύει matching $M \subseteq E, \forall \left(u_i, v_j\right) \in M, (u_i, v_k) \notin M\ και\ \left(u_k, v_j\right) \notin M$.

Το μέγεθος (cardinality) ενός matching είναι το πλήθος των ακμών που περιέχει. Στα σχήματα που ακολουθούν, θα παριστάνουμε τις ακμές που ανήκουν στο matching με διακεκομμένη γραμμή.



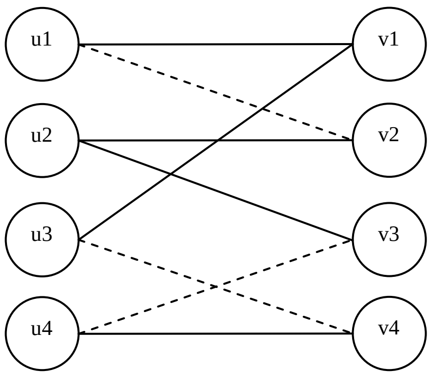
Σχήμα 2.5 : matching μεγέθους 2.

Perfect Matching είναι ένα matching M στο οποίο οι ακμές που ανήκουν στο M καλύπτουν όλες τις κορυφές του G. Προφανώς για να υπάρχει perfect matching σε έναν διμερή γράφο, θα πρέπει τα σύνολα U και V να έχουν τον ίδιο αριθμό στοιχείων.



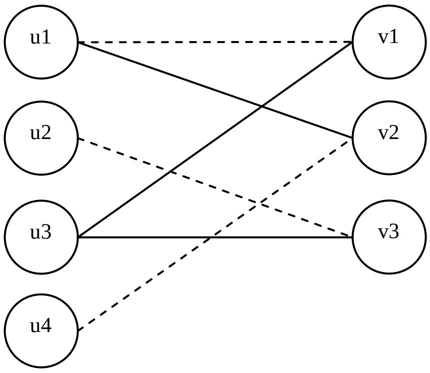
Σχήμα 2.6 : perfect matching.

Maximal Matching ονομάζουμε ένα matching στο οποίο δεν μπορούμε να προσθέσουμε καμία ακμή του γράφου G. Δηλαδή είναι ένα matching του οποίου το μέγεθος δεν μπορεί να αυξηθεί.



Σχήμα 2.7 : maximal matching.

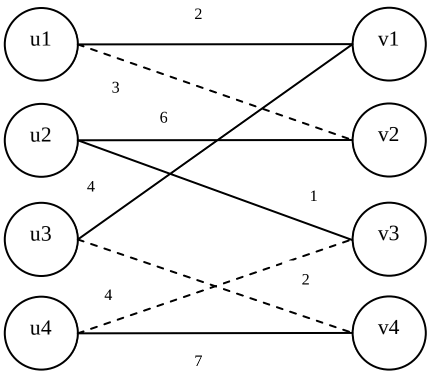
Maximum Matching ενός γράφου G ονομάζουμε το matching που έχει το μεγαλύτερο μέγεθος από όλα τα matchings του γράφου G.



Σχήμα 2.8 : maximum matching.

Το μέγεθος ενός maximum matching ενός διμερούς γραφήματος θα είναι πάντα μικρότερο ή ίσο από το πλήθος των στοιχείων (κορυφών) , του συνόλου με τα λιγότερα στοιχεία , εκ των U και V. Επίσης, προφανώς ένα maximum matching θα είναι και maximal matching, ενώ ένα perfect matching θα είναι και maximum matching.

Στην περίπτωση που έχουμε διμερή γράφο με βάρη, τότε αποκαλούμε βάρος ή κόστος του matching C[M], το άθροισμα των βαρών των ακμών που περιέχονται στο matching (M). Δηλαδή C[M] = $\sum C(e), \forall\, e \in M$.

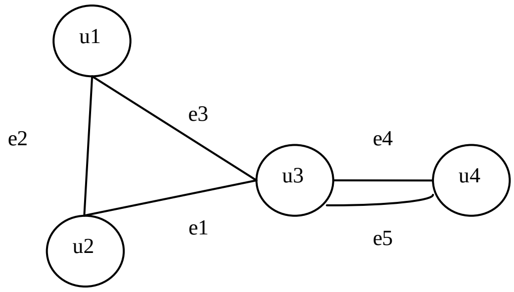


Σχήμα 2.9 : matching κόστους 9.

2.1.5 Αναπαράσταση γράφου με πίνακα

2.1.5.1 Πίνακας Πρόσπτωσης

Έστω ένας γράφος G(V,E) όπου το πλήθος των κορυφών του είναι n και το πλήθος των ακμών του m. Ο πίνακας πρόσπτωσης (incidence matrix) για τον γράφο G θα είναι ο n x m πίνακας IM(G) = [im_{ij}], όπου im_{ij} = 1 αν η ακμή j του γράφου προσπίπτει στην κορυφή i, αλλιώς im_{ij} =0.



Σχήμα 2.10 : γράφος G, n=4 και m=5.

Για τον γράφο G του σχήματος ο πίνακας πρόσπτωσης θα είναι $IM(G) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$.

2.1.5.2 Πίνακας γειτνίασης

Έστω γράφος G(V,E), όπου n το πλήθος των κορυφών του. Ο πίνακας γειτνίασης για τον γράφο G θα είναι ο nxn πίνακας A(G) = [a_{ij}] όπου $α_{ij}$ =1 αν υπάρχει ακμή στον γράφο μεταξύ των κορυφών i και j και $α_{ij}$ =0 αν οι κορυφές i και j δεν συνδέονται μέσω κάποιας ακμής στον γράφο G. Στην περίπτωση όπου δύο κορυφές μπορεί να συνδέονται με περισσότερες από μία ακμές τότε η τιμή του στοιχείου $α_{ij}$ μπορεί να δηλώνει το πλήθος των ακμών που συνδέουν τις κορυφές i και j. Για τον γράφο G του σχήματος, ο πίνακας γειτνίασης θα είναι $A(G)= \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

2.2 Στοιχεία Αλγοριθμικής Θεωρίας

2.2.1 Εισαγωγή

Στο σημείο αυτό θα παρουσιάσουμε κάποια βασικά στοιχεία από την θεωρία αλγορίθμων, η γνώση των οποίων, είναι πολύ σημαντική για την κατανόηση των εννοιών που εισάγονται στα επόμενα κεφάλαια. Ένας αλγόριθμος είναι μια διαδικασία από σαφώς ορισμένα βήματα, η οποία δέχεται ένα σύνολο παραμέτρων ως είσοδο, και επιστρέφει ένα σύνολο παραμέτρων ως έξοδο. Οι αλγόριθμοι χρησιμοποιούνται για την επίλυση απλών έως και εξαιρετικά πολύπλοκων προβλημάτων. Οι έννοιες οι οποίες μας απασχολούν κατά την σχεδίαση ενός αλγορίθμου, είναι αυτές της ορθότητας και της αποδοτικότητας. Ως ορθός, χαρακτηρίζεται ένας αλγόριθμος του οποίου η έξοδος αποτελεί λύση του προβλήματος για το οποίο έχει σχεδιαστεί, σύμφωνα με τον ορισμό του. Την αποδοτικότητα, μπορούμε να την θεωρήσουμε σαν ένα μέτρο σύγκρισης και ταξινόμησης δύο διαφορετικών αλγορίθμων.

2.2.2 Ορισμοί

Ένα μέτρο χαρακτηρισμού ενός αλγορίθμου, ως προς την αποδοτικότητά του, είναι ο χρόνος που χρειάζεται για να επιστρέψει την λύση του προβλήματος για το οποίο τον χρησιμοποιούμε. Στην περίπτωση αυτή, ως μέσο σύγκρισης, χρησιμοποιούμε τον χρόνο που απαιτείται στην χειρότερη περίπτωση (worst case), δηλαδή τον μέγιστο δυνατό χρόνο που μπορεί να χρειαστεί η εκτέλεση του αλγόριθμου.

2.2.2.1 Ασυμπτωτικοί Συμβολισμοί

Συνήθως, δεν μας ενδιαφέρει να μετρήσουμε ακριβώς το κόστος εκτέλεσης ενός αλγορίθμου, αλλά να βρούμε μόνο την τάξη μεγέθους του κόστους. Δηλαδή, μας ενδιαφέρει η ασυμπτωτική συμπεριφορά του αλγορίθμου. Για τον λόγο αυτό έχουν εισαχθεί οι ακόλουθοι συμβολισμοί. Αν f(n) είναι μία συνάρτηση πολυπλοκότητας τότε αν: $f(n) = \Theta(g(n))$ σημαίνει ότι $\exists c1, c2, n_0 > 0 : \forall n \geq n_0 \; 0 \leq c1g(n) \leq f(n) \leq c2g(n)$, αν $f(n) = O(g(n))$ σημαίνει ότι $\exists c, n_0 > 0 : \forall n \geq n_0 \,, 0 \leq f(n) \leq cg(n)$, αν $f(n) = \Omega(g(n))$, σημαίνει ότι $\exists c, n_0 > 0 : \forall n \geq n_0 \,, 0 \leq c\big(g(n)\big) \leq f(n)$, αν $f(n) = o\big(g(n)\big)$, σημαίνει ότι $\forall c, \; \exists n_0 > 0 : n \geq n_0 \,, 0 \leq f(n) \leq cg(n)$.

2.2.2.2 Ορισμός Προβλήματος-Στιγμιότυπου Προβλήματος

Με τον όρο πρόβλημα θα εννοούμε την αφηρημένη διαδικασία, για την οποία επιθυμούμε να βρούμε μία λύση, εφαρμόζοντας τα βήματα κάποιου αλγορίθμου. Η υλοποίηση-πραγμάτωση του προβλήματος, δίνοντας συγκεκριμένες τιμές στις παραμέτρους του, δημιουργεί ένα στιγμιότυπο (instance) I, του συγκεκριμένου προβλήματος. Το στιγμιότυπο αυτό είναι η είσοδος σε έναν αλγόριθμο επίλυσης του προβλήματος. Για παράδειγμα, ας θεωρήσουμε το πρόβλημα της ταξινόμησης. Σύμφωνα με αυτό το πρόβλημα, θέλουμε για μία ακολουθία n αριθμών, να βρούμε μία αναδιάταξή τους, τέτοια ώστε κάθε αριθμός να είναι μεγαλύτερος από τον αμέσως προηγούμενο του (ταξινόμηση σε αύξουσα σειρά). Αν η ακολουθία, η οποία καλούμαστε να ταξινομήσουμε είναι η {5,8,1,6}, τότε αυτή αποτελεί ένα στιγμιότυπο του προβλήματος της ταξινόμησης και ένας αλγόριθμος που επιλύει το πρόβλημα αυτό, θα δέχεται το στιγμιότυπο αυτό σαν είσοδο και θα επιστρέφει σαν έξοδο την ακολουθία {1,5,6,8}.

2.2.2.3 Κατηγοριοποίηση Προβλημάτων

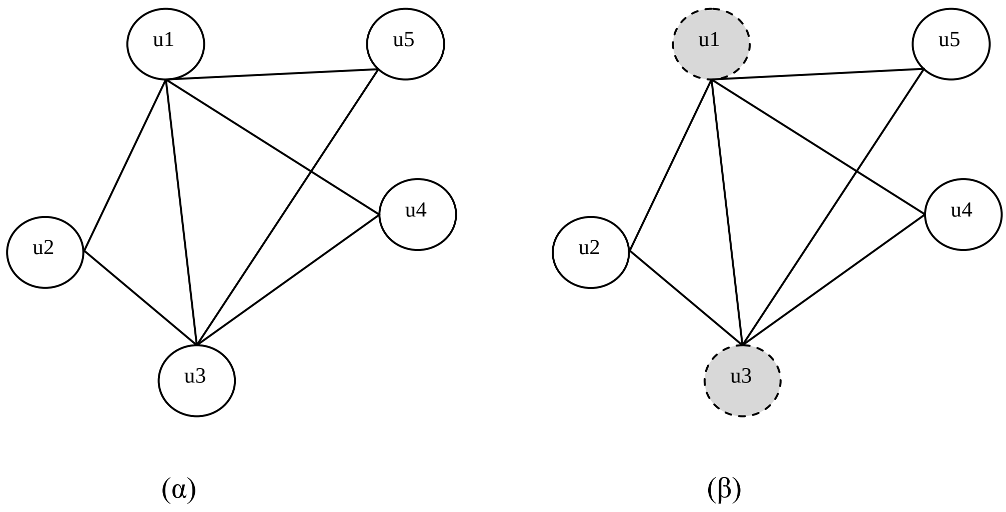
Τα προβλήματα τα οποία επιλύουμε με την χρήση αλγορίθμων, μπορούν να χωριστούν σε τρείς κατηγορίες. Στα προβλήματα απόφασης, στα προβλήματα βελτιστοποίησης και στα υπολογιστικά προβλήματα. Στα υπολογιστικά προβλήματα, το πρόβλημα που καλούμαστε να επιλύσουμε, απαιτεί την διενέργεια υπολογισμών προκειμένου να βρούμε την τιμή κάποιας παραμέτρου, που για τα συγκεκριμένα δεδομένα εισόδου, αποτελεί την λύση του προβλήματος.

Στα προβλήματα βελτιστοποίησης ζητάμε τη λύση που βελτιστοποιεί (ελαχιστοποιεί ή μεγιστοποιεί) μία αντικειμενική συνάρτηση. Για κάθε στιγμιότυπο Ι, ενός προβλήματος Π, υπάρχει ένα σύνολο εφικτών λύσεων F(I).Σε ένα πρόβλημα βελτιστοποίησης, για κάθε λύση SOL ∈ F(I), κάνουμε αντιστοίχηση μίας τιμής , μέσω μίας αντικειμενικής συνάρτησης. Έστω ότι η τιμή αυτή είναι η C(SOL). Αναζητούμε την λύση εκείνη, από όλες τις εφικτές λύσεις, για την οποία η τιμή της αντικειμενικής συνάρτησης είναι μέγιστη ή ελάχιστη. Δηλαδή θέλουμε να βρούμε την $SOL \in F(I): C(SOL) \leq C(SOL_i)$ ή $C(SOL) \geq C(SOL_i), \forall SOL_i \in F(I), 1 \leq i \leq |F(I)|$.

Στα προβλήματα απόφασης, η λύση τους είναι της μορφής «ναι»-«όχι». Δηλαδή για κάθε πιθανή είσοδο σε έναν αλγόριθμο που επιλύει ένα πρόβλημα απόφασης, αντιστοιχεί η τιμή «1» (ναι) ή «0» (όχι) στην έξοδό του. Αν θεωρήσουμε ότι Α, είναι το σύνολο όλων των εισόδων x, για τις οποίες η έξοδος του αλγορίθμου είναι «1», τότε μπορούμε να αναδιατυπώσουμε το ένα πρόβλημα απόφασης ως : $x \in A$?

Για την καλύτερη κατανόηση των δύο παραπάνω κατηγοριών προβλημάτων, θα ορίσουμε το πρόβλημα Vertex Cover (κάλυψη κορυφών) σε έναν γράφο. Στο πρόβλημα αυτό, για έναν δεδομένο γράφο G(V,E), αναζητούμε ένα σύνολο από κορυφές του, έτσι ώστε κάθε ακμή του G, να καλύπτεται τουλάχιστον από μία κορυφή. Δηλαδή αν VC είναι μία τέτοια κάλυψη, τότε θα ισχύει : $VC \subseteq V: \forall (u, v) \in E, u \in VC$ ή $v \in VC$. Ένα πρόβλημα βελτιστοποίησης είναι το ακόλουθο: για τον γράφο G(V,E) να βρεθεί η κάλυψη (VC) με το ελάχιστο πλήθος κορυφών. Ένα αντίστοιχο πρόβλημα απόφασης είναι: για τον γράφο G, υπάρχει κάλυψη (VC), που να έχει πλήθος κορυφών ίσο με n;

Έστω ότι G(V,E) είναι ο γράφος του σχήματος που ακολουθεί. Ο γράφος G αποτελεί ένα στιγμιότυπο του προβλήματος καθορισμού μιας κάλυψης κορυφών. Ένας αλγόριθμος που επιλύει το παραπάνω πρόβλημα βελτιστοποίησης, για είσοδο τον γράφο G, θα επιστρέψει την κάλυψη που φαίνεται στο ακόλουθο σχήμα με διακεκομμένη γραμμή. Δηλαδή η ελάχιστη κάλυψη είναι η VC={u1,u3}, η οποία έχει πλήθος κορυφών ίσο με 2. Αντίστοιχα ένας αλγόριθμος που επιλύει το παραπάνω πρόβλημα απόφασης, στην ερώτηση αν υπάρχει κάλυψη μεγέθους 2(ή μεγαλύτερη), θα αποφαίνεται καταφατικά.



Σχήμα 2.11: (α) γράφος G, (β) Vertex Cover μεγέθους 2

2.2.3 Στοιχεία Πολυπλοκότητας Αλγορίθμων

2.2.3.1 Αλγόριθμοι πολυωνυμικού χρόνου

Ένας αλγόριθμος, ο οποίος για είσοδο μεγέθους n, στην χειρότερη περίπτερη (worst case), έχει χρόνο τρεξίματος της τάξης του $O(n^k)$, όπου k είναι σταθερά, ονομάζεται Αλγόριθμος Πολυωνυμικού Χρόνου (polynomial time algorithm). Οι μελέτες που έχουν γίνει πάνω στο πεδίο της ανάλυσης πολυπλοκότητας αλγορίθμων, έχουν δείξει ότι υπάρχουν προβλήματα τα οποία δεν μπορούν να λυθούν από οποιονδήποτε υπολογιστή όσο χρόνο και αν διαθέτουμε (π.χ. Halting Problem). Ενώ άλλα προβλήματα, δεν μπορούν να λυθούν σε χρόνο $O(n^k)$ οποιαδήποτε σταθερά k και αν επιλέξουμε (π.χ. Traveling Salesperson Problem). Έτσι, προβλήματα τα οποία επιλύονται με αλγόριθμους πολυωνυμικού χρόνου θα τα θεωρούμε «εύκολα», ενώ τα υπόλοιπα «δύσκολα».

2.2.3.2 Κλάσεις προβλημάτων

Σύμφωνα με την πολυπλοκότητα που χαρακτηρίζει έναν αλγόριθμο, μπορούμε να τον κατατάξουμε σε διάφορες ομάδες- κλάσεις προβλημάτων. Στη συνέχεια θα παρουσιάσουμε συνοπτικά εκείνες τις κλάσεις ,με τις οποίες θα ασχοληθούμε στην παρούσα εργασία. Η κλάση P (Polynomial) περιέχει τα προβλήματα, που λύνονται σε πολυωνυμικό χρόνο. Η κλάση NP (Non-deterministic Polynomial time) περιέχει τα προβλήματα που λύνονται σε πολυωνυμικό χρόνο από έναν μη- ντετερμινιστικό αλγόριθμο. Ένας τέτοιος αλγόριθμος, είναι ένας θεωρητικός αλγόριθμος, ο οποίος δεν έχει προκαθορισμένη αλληλουχία βημάτων, αλλά σε κάθε βήμα μπορεί να επιλέξει μία από τις πιθανές ακολουθίες βημάτων που θα ακολουθήσει. Διαφορετικά, μπορούμε να πούμε ότι για τα προβλήματα αυτά, η εύρεση της λύσης τους είναι χρονοβόρα, ενώ ο έλεγχος της ορθότητας της λύσης αυτής μπορεί να γίνει σε πολυωνυμικό χρόνο. Δηλαδή αν μας δοθεί μία υποψήφια λύση ενός τέτοιου προβλήματος, μπορούμε σε πολυωνυμικό χρόνο να αποφανθούμε αν είναι λύση ή όχι.

Σαν παράδειγμα αναφέρουμε το πρόβλημα του καθορισμού κύκλου Hamilton (Hamiltonian Cycle) σε έναν γράφο. Σύμφωνα με το πρόβλημα αυτό, για έναν δεδομένο γράφο αναζητούμε να βρούμε, να υπάρχει, μία διαδρομή η οποία αρχίζει και τελειώνει στην ίδια κορυφή του γράφου, και διέρχεται ακριβώς μία φορά από κάθε κορυφή του. Το να βρούμε μία τέτοια ακολουθία ακμών είναι ιδιαίτερα δύσκολο και χρονοβόρο, όμως το να ελέγξουμε αν μία ακολουθία είναι κύκλος Hamilton ή όχι, χρειάζεται πολυωνυμικό χρόνο. Συνεπώς το πρόβλημα της εύρεσης κύκλου Hamilton σε έναν γράφο ανήκει στην κλάση προβλημάτων NP.

Ένα πρόβλημα Π το οποίο ανήκει στην κλάση P, θα ανήκει επίσης και στην NP. Αυτό προκύπτει από το ότι ο ντετερμινιστικός αλγόριθμος που επιλύει το Π σε πολυωνυμικό χρόνο, μπορεί να θεωρηθεί σαν ειδική περίπτωση ενός μη-ντετερμινιστικού αλγορίθμου. Επομένως, οποιοδήποτε πρόβλημα μπορεί να επιλυθεί σε πολυωνυμικό χρόνο από ντετερμινιστικό αλγόριθμο, μπορεί να επιλυθεί σε πολυωνυμικό χρόνο και από μη-ντετερμινιστικό αλγόριθμο. Συνεπώς ισχύει $P \subseteq NP$.

2.2.3.3 Αναγωγές Πολυωνυμικού Χρόνου

Ας υποθέσουμε ότι θέλουμε να λύσουμε το πρόβλημα Α, το οποίο είναι ένα πρόβλημα απόφασης. Επίσης ότι υπάρχει ένα διαφορετικό πρόβλημα απόφασης Β, και ότι γνωρίζουμε μία διαδικασία η οποία μετασχηματίζει ένα οποιοδήποτε στιγμιότυπο του Α, σε κάποιο στιγμιότυπο του Β, με τις εξής ιδιότητες: (α) ο μετασχηματισμός χρειάζεται πολυωνυμικό χρόνο, και (β) οι απαντήσεις των δύο προβλημάτων είναι ίδιες. Δηλαδή η απάντηση στο πρόβλημα Α είναι «ναι» αν και μόνο αν η απάντηση στο πρόβλημα Β είναι επίσης «ναι». Τότε αποκαλούμε την διαδικασία αυτή αναγωγή πολυωνυμικού χρόνου (polynomial time reduction).

Αν όλα τα προβλήματα που ανήκουν στην κλάση NP, ανάγονται πολυωνυμικά σε κάποιο πρόβλημα Π, τότε το Π χαρακτηρίζεται ως NP-Hard πρόβλημα. Αν το Π ανήκει στην κλάση NP τότε λέμε ότι το πρόβλημα Π είναι NP-Complete. Συνεπώς για να δείξουμε ότι ένα πρόβλημα είναι NP-Complete, πρέπει να δείξουμε ότι ανήκει στην κλάση NP, και ότι κάθε NP πρόβλημα ανάγεται σε αυτό, ή διαφορετικά ότι υπάρχει κάποιο NP-Complete πρόβλημα το οποίο ανάγεται σε αυτό. Αν βρεθεί αλγόριθμος πολυωνυμικού χρόνου για κάποιο NP-

Complete πρόβλημα, τότε προφανώς ο ίδιος αλγόριθμος θα επιλύει κάθε NP πρόβλημα και στην περίπτωση αυτή θα ισχύει ότι P=NP.

3

Προσεγγιστικοί Αλγόριθμοι

3.1 Εισαγωγή

Όπως γνωρίζουμε, υπάρχουν προβλήματα τα οποία είναι υπολογιστικά δυσεπίλυτα, καθώς ο χρόνος που απαιτείται από κάποιον αλγόριθμο για να επιστρέψει μία λύση, είναι ιδιαίτερα μεγάλος, καθιστώντας συχνά τον υπολογισμό λύσης πρακτικά ανέφικτο. Πολλές φορές καλούμαστε να επιλύσουμε τέτοια προβλήματα βελτιστοποίησης, έχοντας περιορισμένο χρόνο. Στις περιπτώσει εκείνες που μας ενδιαφέρει να βρούμε μία λύση του προβλήματος, σε μικρό χρονικό διάστημα, και είμαστε ελαστικοί ως προς το αν η λύση που θα βρούμε θα είναι βέλτιστη ή όχι, μπορούμε να χρησιμοποιήσουμε αλγορίθμους που βρίσκουν μία σχεδόν-βέλτιστη (near-optimal) λύση σε πολυωνυμικό χρόνο. Τέτοιοι αλγόριθμοι, που βρίσκουν για κάποιο πρόβλημα μία λύση, ή οποία εν γένει δεν είναι βέλτιστη αλλά είναι «κοντά» στην βέλτιστη , ονομάζονται προσεγγιστικοί αλγόριθμοι.

3.2 Ορισμοί-Συμβολισμοί

Στη συνέχεια θα συμβολίζουμε με Π το πρόβλημα βελτιστοποίησης, το οποίο καλούμαστε να επιλύσουμε και με Α τον αλγόριθμο τον οποίο θα χρησιμοποιήσουμε για την επίλυση του. Το Π μπορεί να είναι ένα πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης. Για κάθε στιγμιότυπο Ι του Π, θα υπάρχει ένα σύνολο από εφικτές λύσεις (feasible solutions), $F_{\Pi}(I)$. Θα υπάρχει αλγόριθμος πολυωνυμικού χρόνου, οποίος δεχόμενος μία πιθανή λύση s και το στιγμιότυπο Ι, μπορεί να αποφανθεί αν $s \in F_{\Pi}(I)$. Για κάθε εφικτή λύση $s \in F_{\Pi}(I)$, υπάρχει αντικειμενική συνάρτηση C_{Π} , υπολογίσιμη σε πολυωνυμικό χρόνο, η οποία αναθέτει έναν μη-αρνητικό ρητό αριθμό σε κάθε ζεύγος (I,s). Η τιμή $C_{\Pi}(I,s)$ που προκύπτει από την εφαρμογή του αλγόριθμου Α, είναι η λύση που παίρνουμε από τον Α, και την συμβολίζουμε με $SOL_A(I)$. Για ένα πρόβλημα ελαχιστοποίησης, η βέλτιστη λύση είναι η εφικτή λύση της οποίας η τιμή της αντικειμενικής συνάρτησης C_{Π} είναι η ελάχιστη. Στην περίπτωση που έχουμε πρόβλημα μεγιστοποίησης, αντίστοιχα είναι η εφικτή λύση με την μέγιστη τιμή αντικειμενικής συνάρτησης. Και στις δύο περιπτώσεις συμβολίζουμε την βέλτιστη λύση με $OPT_{\Pi}(I)$. Το μέγεθος του στιγμιότυπου I θα συμβολίζουμε με |I| και θα αντιστοιχεί στο πλήθος των bits που χρειαζόμαστε για την αναπαράσταση του I σε δυαδική μορφή.

3.3 *Παράγοντας Προσέγγισης*

Αναφορικά στην απόδοση των προσεγγιστικών αλγορίθμων, μελετάμε την χειρότερη περίπτωση, όχι μόνο ως προς την χρονική πολυπλοκότητα, αλλά και ως προς την βέλτιστη λύση του προβλήματος. Δηλαδή ενδιαφερόμαστε για το πόσο κοντά στη βέλτιστη λύση ($OPT_{\Pi}(I)$), είναι η λύση που μας δίνει ένας προσεγγιστικός αλγόριθμος ($SOL_A(I)$). Ένας αλγόριθμος Α, που επιλύει ένα πρόβλημα ελαχιστοποίησης Π, έχει παράγοντα-πηλίκο προσέγγισης (approximation factor-ratio) έναν πραγματικό αριθμό $\rho \geq 1$, αν για κάθε στιγμιότυπο I του Π ισχύει :

$$SOL_A(I) \leq \rho \cdot OPT_{\Pi}(I) \Rightarrow \frac{SOL_A(I)}{OPT_{\Pi}(I)} \leq \rho$$

Αντίστοιχα, για ένα πρόβλημα μεγιστοποίησης θα ισχύει $OPT_{\Pi}(I) \geq SOL_A(I) \geq \rho \cdot OPT_{\Pi}(I)$ $\Rightarrow \frac{OPT_{\Pi}(I)}{SOL_A(I)} \leq \rho$ και στις δύο περιπτώσεις χαρακτηρίζουμε τον Α σαν ρ-προσεγγιστικό αλγόριθμο.

Εναλλακτικά, μπορούμε να ορίσουμε τον παράγοντα προσέγγισης σαν $\rho = max \frac{SOL_A(I)}{OPT_{\Pi}(I)}, \forall I \in F_{\Pi}(I)$ για προβλήματα ελαχιστοποίησης και $\rho = max \frac{OPT_{\Pi}(I)}{SOL_A(I)}, \forall I \in F_{\Pi}(I)$ για προβλήματα μεγιστοποίησης. Από τον παραπάνω ορισμό, φαίνεται πως μας ενδιαφέρει η τιμή του ρ στην χειρότερη περίπτωση (worst case), δηλαδή για το στιγμιότυπο I εκείνο, όπου η τιμή $SOL_A(I)$ απέχει την μεγαλύτερη απόσταση από την τιμή της βέλτιστης λύσης για το Π ($OPT_{\Pi}(I)$). Όσο η τιμή του ρ πλησιάζει στην μονάδα, τόσο ο αλγόριθμος Α γίνεται αποδοτικότερος, αφού η λύση που μας επιστρέφει πλησιάζει την βέλτιστη λύση. Ένας αλγόριθμος που έχει ρ=1, παύει να είναι προσεγγιστικός, καθώς επιστρέφει πάντα την βέλτιστη λύση. Εν γένει ο παράγοντας ρ μπορεί να μην είναι ένας σταθερός αριθμός για κάποιον αλγόριθμο Α, αλλά να είναι συνάρτηση της εισόδου του. Αν θεωρήσουμε ότι |I|=n, τότε στην γενική περίπτωση ο παράγοντας προσέγγισης θα είναι $\rho_A(n)$.

3.4 Κάτω Φράγμα Βέλτιστης Λύσης

Κατά την σχεδίαση ή την ανάλυση ενός προσεγγιστικού αλγορίθμου είναι απαραίτητος ο υπολογισμός του παράγοντα προσέγγισης ρ. Στο σημείο αυτό προκύπτει ένα πολύ σημαντικό πρόβλημα. Όπως αναφέραμε, χρησιμοποιούμε προσεγγιστικούς αλγορίθμους για την επίλυση δύσκολων προβλημάτων, των οποίων η λύση είναι χρονοβόρα, επομένως η εύρεση της βέλτιστης λύσης, που είναι απαραίτητη για τον υπολογισμό του ρ, δεν είναι εφικτή. Για τον λόγο αυτό, ακολουθούμε την τακτική της εύρεσης ενός κάτω φράγματος για την βέλτιστη λύση, αντί για την ίδια την τιμή. Έστω ότι LB είναι η τιμή του κάτω φράγματος που έχουμε βρει για την τιμή της βέλτιστης λύσης του προβλήματος ελαχιστοποίησης Π. Τότε θα ισχύει $OPT \geq LB$ και για τον υπολογισμό της προσεγγισιμότητας του αλγορίθμου θα χρησιμοποιούμε το πηλίκο $\frac{SOL_A(I)}{LB(I)}$. Έτσι, είμαστε σε θέση να υπολογίσουμε τον παράγοντα προσέγγισης, χωρίς να γνωρίζουμε την τιμή της βέλτιστης λύσης. Στην περίπτωση που έχουμε πρόβλημα μεγιστοποίησης, αρκεί να υπολογίσουμε ένα άνω φράγμα για την βέλτιστη λύση.

3.5 Σχεδίαση-Βελτιστοποίηση Προσεγγιστικών Αλγορίθμων

Κατά την σχεδίαση ενός αλγορίθμου, σε γενικές γραμμές, μπορούμε να ακολουθήσουμε την παρακάτω στρατηγική. Πρώτα πρέπει να βρούμε ένα φράγμα LB για την βέλτιστη λύση του προβλήματος που θέλουμε να επιλύσουμε. Στη συνέχεια, εκμεταλλευόμενοι το φράγμα που υπολογίσαμε, βρίσκουμε έναν αλγόριθμο, ο οποίος μας επιστρέφει μια λύση, η τιμή της οποίας είναι ρ φορές η τιμή του φράγματος που έχουμε βρει. Αφού ελέγξουμε την ορθότητα και την αποδοτικότητά του, πρέπει να μελετήσουμε αν μπορεί να γίνει καλύτερος. Δηλαδή, κατά πόσο μπορούμε να επιτύχουμε καλύτερη προσέγγιση της βέλτιστης λύσης. Μέχρι τώρα, έχουμε δημιουργήσει έναν αλγόριθμο, έχουμε βρει κάποιο φράγμα για την βέλτιστη λύση και σύμφωνα με αυτό έχουμε υπολογίσει τον παράγοντα προσέγγισης ρ του αλγορίθμου μας.

Το πρώτο ερώτημα στο οποίο πρέπει να απαντήσουμε είναι αν πράγματι η λύση που μας δίνει ο προσεγγιστικός αλγόριθμος που κατασκευάσαμε είναι ρ φορές χειρότερη από την βέλτιστη λύση. Προκειμένου να αποδείξουμε ότι κάτι τέτοιο ισχύει, αρκεί να βρούμε ένα στιγμιότυπο I, του προβλήματός μας, για το οποίο η λύση που παίρνουμε, αν ακολουθήσουμε τα βήματα του αλγορίθμου μας, είναι ρ φορές χειρότερη από το φράγμα που έχουμε βρει.

Ένα άλλο ερώτημα στο οποίο πρέπει να απαντήσουμε είναι, μήπως θα μπορούσαμε να μεταβάλλουμε τον αλγόριθμό μας, έτσι ώστε να προσεγγίζει καλύτερα την βέλτιστη λύση. Προκειμένου να μεταβάλουμε τον αλγόριθμο, θα πρέπει να εξαλείψουμε ή να διαφοροποιήσουμε κάποιο από τα βήματά του. Για να αποδείξουμε ότι κάτι τέτοιο δεν είναι εφικτό, αρκεί να βρούμε ένα στιγμιότυπο του προβλήματος, στο οποίο όταν εφαρμόσουμε τον αλγόριθμο που έχουμε κατασκευάσει, κάθε βήμα του είναι απαραίτητο για να προκύψει βέλτιστη λύση για το πρόβλημά μας. Τέλος, θα πρέπει να ελέγξουμε, αν το φράγμα που έχουμε βρει, θα μπορούσε να γίνει καλύτερο. Δηλαδή, αν είναι δυνατόν να βρούμε ένα διαφορετικό φράγμα, το οποίο θα πλησιάζει περισσότερο στην βέλτιστη λύση από εκείνο που έχουμε ήδη βρει. Το κομμάτι αυτό συχνά είναι το δυσκολότερο, καθώς δεν υπάρχει συγκεκριμένη τακτική την οποία πρέπει να ακολουθήσουμε, αλλά η προσέγγιση που πρέπει να κάνουμε εξαρτάται από το πρόβλημα το οποίο καλούμαστε να επιλύσουμε.

3.6 Παραδείγματα

3.6.1 Προσεγγιστικός αλγόριθμος για το Vertex Cover

Το παράδειγμα που θα δούμε είναι η επίλυση του προβλήματος της εύρεσης για έναν γράφο, το Vertex Cover με το ελάχιστο μέγεθος. Έστω G(V,E) ένας μη-κατευθυνόμενος γράφος. Όπως έχουμε δει και στο προηγούμενο κεφάλαιο, ένα σύνολο VC από κορυφές του G είναι vertex cover αν $VC \subseteq V: \forall\ (u,v) \in E, u \in VC$ ή $v \in VC.$ Το πρόβλημα που καλούμαστε να επιλύσουμε, είναι ένα πρόβλημα ελαχιστοποίησης. Στιγμιότυπο του προβλήματος είναι ένας γράφος G. Το σύνολο των εφικτών λύσεων, είναι το σύνολο όλων των VC που υπάρχουν στον G. Η αντικειμενική συνάρτηση είναι το μέγεθος του VC, και η βέλτιστη λύση θα είναι το VC που ελαχιστοποιεί την τιμή της αντικειμενικής συνάρτησης, δηλαδή το VC με το ελάχιστο πλήθος.

Αρχικά, πρέπει να βρούμε ένα κάτω φράγμα για την τιμή της βέλτιστης λύσης. Αν βρούμε ένα maximal matching M για τον γράφο G, τότε το μέγεθος του matching |M| θα είναι ένα κάτω φράγμα. Αυτό ισχύει, διότι οποιοδήποτε VC στον γράφο G, θα περιέχει τουλάχιστον ένα από τα δύο άκρα (κορυφές) των ακμών που ανήκουν στο M. Αν για κάποια ακμή δεν περιέχεται κανένα από τα δύο της άκρα, τότε η ακμή αυτή δεν θα καλύπτεται. Ένας προσεγγιστικός αλγόριθμος που προκύπτει, είναι ο ακόλουθος. Για τον γράφο G, βρες ένα maximal matching M. Για κάθε ακμή (u,v)∈M, τοποθέτησε στο σύνολο VC τις κορυφές u και v.

Για τον παραπάνω αλγόριθμο, θα υπολογίσουμε τον παράγοντα προσέγγισης. Όπως δείξαμε παραπάνω, ισχύει $|M| \leq OPT$ και $|VC| = 2 \cdot |M|$, οπότε προκύπτει:

$SOL = |VC| = 2 \cdot |M| \leq 2 \cdot OPT \Rightarrow SOL \leq 2 \cdot OPT \Rightarrow \rho = 2$.Επομένως ο αλγόριθμος που παρουσιάσαμε είναι 2-προσεγγιστικός και συνεπώς στην χειρότερη περίπτωση, θα μας επιστρέψει ένα VC το οποίο θα περιέχει τις διπλάσιες κορυφές, από εκείνο της βέλτιστης λύσης. Στη συνέχεια πρέπει να βρούμε, σύμφωνα με την προηγούμενη ενότητα, ένα στιγμιότυπο στο οποίο, η λύση που παίρνουμε από την εφαρμογή του αλγορίθμου, είναι διπλάσια από την βέλτιστη. Για τον λόγο αυτό, θεωρούμε έναν πλήρη διμερή γράφο $K_{n,n}$.Οποιοδήποτε maximal matching στον γράφο αυτό, θα έχει μέγεθος n, ενώ ο αλγόριθμος θα επιστρέψει πάντα VC μεγέθους 2n. Εφόσον το βέλτιστο VC έχει μέγεθος n, η τιμή που επιστρέφει η λύση του αλγόριθμου, είναι όντως διπλάσια από την βέλτιστη.

Στην συνέχεια, θεωρούμε ότι εφαρμόζουμε τον αλγόριθμο σε έναν πλήρη γράφο, με περιττό αριθμό κορυφών K_{2n+1}. Στην περίπτωση αυτή, ένα maximal matching θα έχει μέγεθος n, και συνεπώς ο αλγόριθμος θα επιστρέψει ένα VC μεγέθους 2n. Σε έναν πλήρη γράφο με περιττό πλήθος κορυφών όμως, το ελάχιστο VC έχει μέγεθος 2n. Επομένως ο αλγόριθμος μας επιστρέφει την βέλτιστη λύση.

4

Το πρόβλημα της χρονοδρομολόγησης αποστολής δεδομένων σε δίκτυα

Στο κεφάλαιο αυτό θα περιγράψουμε το πρόβλημα με το οποίο θα ασχοληθούμε στην παρούσα εργασία και την μοντελοποίηση του. Έπειτα θα αναφέρουμε κάποιες εφαρμογές στις οποίες προκύπτει το πρόβλημα αυτό, και στη συνέχεια θα παρουσιάσουμε τρεις προσεγγιστικούς αλγορίθμους, που επιλύουν το πρόβλημα αυτό.

4.1 Το πρόβλημα

4.1.1 Περιγραφή

Θεωρούμε πως έχουμε ένα σύνολο από δεδομένα, τα οποία θέλουμε να στείλουμε από ένα αρχικό σημείο (αποστολέας) σε ένα τελικό σημείο (παραλήπτης). Θεωρούμε επίσης πως η αποστολή των δεδομένων γίνεται από ένα σύνολο πομπών και η λήψη από ένα σύνολο από δέκτες. Ένας πομπός δεν μπορεί να στέλνει ταυτόχρονα δεδομένα σε δύο ή περισσότερους δέκτες, και αντίστοιχα ένας δέκτης δεν μπορεί να λαμβάνει ταυτόχρονα δεδομένα από δύο ή περισσότερους πομπούς. Δεχόμαστε ότι γνωρίζουμε εκ των προτέρων το πλήθος των μηνυμάτων (δεδομένων) που πρέπει να σταλούν από τους πομπούς στους δέκτες. Για κάθε μήνυμα γνωρίζουμε το πομπό από τον οποίο πρέπει να σταλεί, τον δέκτη που πρέπει να το λάβει, καθώς και τον χρόνο που απαιτείται για την αποστολή αυτή. Προκειμένου να γίνει η αποστολή των δεδομένων, τα ομαδοποιούμε έτσι ώστε να γίνονται ταυτόχρονες αποστολές δεδομένων, όταν αυτό είναι εφικτό. Πριν από την αποστολή κάθε ομάδας μηνυμάτων, θεωρούμε πως απαιτείται κάποιος χρόνος ώστε να καθορίσει το σύστημα ποια μηνύματα θα αποσταλούν και συνεπώς ποιοι πομποί και δέκτες θα ενεργοποιηθούν. Το χρόνο αυτό τον ονομάζουμε καθυστέρηση-χρόνο καθορισμού ή set-up delay. Ο χρόνος που απαιτείται για την αποστολή κάθε ομάδας μηνυμάτων θα είναι το άθροισμα της καθυστέρησης καθορισμού και του χρόνου του μηνύματος με τον μεγαλύτερο χρόνο αποστολής της συγκεκριμένης ομάδας μηνυμάτων. Κατά αντιστοιχία, ο συνολικός χρόνος που χρειάζεται για την αποστολή όλων των μηνυμάτων, θα είναι ίσος με το άθροισμα των χρόνων όλων των ομάδων μηνυμάτων που στάλθηκαν. Θέλουμε να βρούμε την σειρά εκείνη, με την οποία πρέπει να γίνει η αποστολή των μηνυμάτων, ώστε να απαιτείται ο ελάχιστος συνολικός χρόνος για την αποστολή όλων των μηνυμάτων. Επομένως πρόκειται για ένα πρόβλημα βελτιστοποίησης και συγκεκριμένα ελαχιστοποίησης.

4.1.2 Μοντελοποίηση

Έστω ότι έχουμε ένα σύνολο από n πομπούς, το S: {s1,s2,…sn} και ένα σύνολο από m δέκτες, το R:{r1,r2,…,rm}, που αποτελούν το σύστημά μας. Επίσης έχουμε ένα σύνολο από k δεδομένα, το D:{d1,d2,…,dk}, τα οποία και θέλουμε να στείλουμε από τους πομπούς στους δέκτες. Στη συνέχεια ορίζουμε τρεις συναρτήσεις, προκειμένου για κάθε δεδομένο di ∈ D, να μπορούμε να ανακτήσουμε τις απαραίτητες πληροφορίες για αυτό. Συγκεκριμένα, η συνάρτηση s(di) μας επιστρέφει τον πομπό, ο οποίος πρέπει να στείλει το δεδομένο di, η συνάρτηση r(di) αντίστοιχα μας επιστρέφει τον δέκτη στον οποίο πρέπει να σταλεί το δεδομένο di, και η συνάρτηση t(di) μας επιστρέφει τον χρόνο που απαιτείται για την αποστολή του δεδομένου di. Μία ομάδα μηνυμάτων-δεδομένων ταυτόχρονης αποστολής είναι ένα σύνολο M ⊆ D τέτοιο ώστε ∀ di, dj ∈ M : s(di) ≠ s(dj) και r(di) ≠ r(dj). Το πλήθος των στοιχείων που αποτελούν μία ομάδα M θα το συμβολίζουμε με |M|. Ο χρόνος που απαιτείται για την αποστολή μιας ομάδας μηνυμάτων M, αν d είναι η καθυστέρηση καθορισμού - set up delay, θα είναι: time (M) = max {t (di)} + d, ∀ di ∈ M. Αν για την αποστολή όλων των δεδομένων, απαιτούνται συνολικά l ομάδες, τότε ο συνολικός χρόνος που απαιτείται θα είναι total time T=$\sum_{i=1}^{l} time(Mi)$.

Ως εφαρμογή των παραπάνω, παραθέτουμε το παράδειγμα που ακολουθεί. Έστω το ακόλουθο στιγμιότυπο του προβλήματος που μελετούμε: S={s1,s2,s3,s4} , R={r1,r2,r3}, D={d1,d2,d3,d4,d5} και d =1. Δηλαδή έχουμε ένα σύστημα που αποτελείται από 4 πομπούς και 3 δέκτες, στο οποίο θέλουμε να στείλουμε συνολικά 5 δεδομένα, των οποίων οι πληροφορίες φαίνονται στον παρακάτω πίνακα, ενώ η καθυστέρηση καθορισμού είναι 1 χρονική μονάδα.

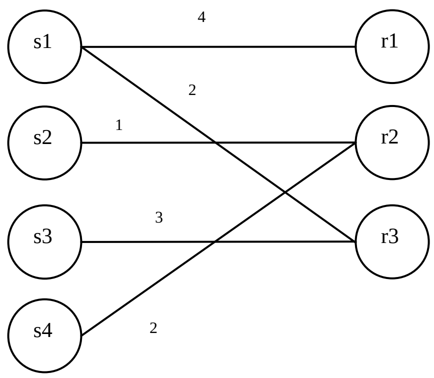
di	***s(di)***	***r(di)***	***t(di)***
1	1	1	4
2	1	3	2
3	2	2	1
4	3	3	3
5	4	2	2

Σχήμα 4.1 : πίνακας δεδομένων.

Έστω ότι για το παράδειγμά μας, επιλέγουμε τις εξής ομάδες αποστολής: M1={d1,d3,d4} και M2={d2,d5}. Υπολογίζουμε τον χρόνο αποστολής για την κάθε ομάδα: time(M1)= max{t(d1), t(d3), t(d4)} + *d* = max{4,1,3} +1= 4+1 = 5 χρονικές μονάδες. Αντίστοιχα για την ομάδα M2 έχουμε time(M2)= max{t(d2),t(d5)}+*d* = max{2,2} +1=2+1=3 χρονικές μονάδες. Συνεπώς ο συνολικός χρόνος που απαιτείται, για την συγκεκριμένη επίλυση του στιγμιότυπου που έχουμε θα είναι T= time(M1)+time(M2) = 5+3 = 8 χρονικές μονάδες.

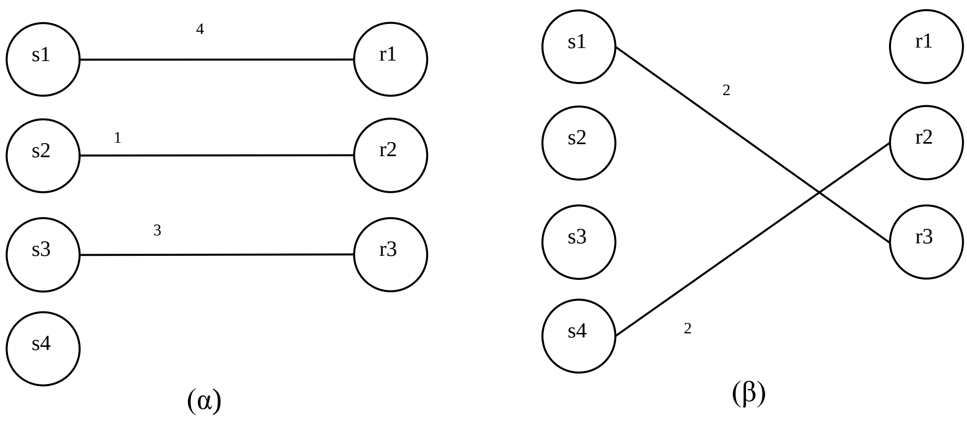
4.1.2.1 Απεικόνιση στιγμιότυπου προβλήματος με γράφο

Ο πιο εύχρηστος τρόπος απεικόνισης και παρουσίασης ενός στιγμιότυπου του προβλήματός μας, είναι κάνοντας χρήση ενός διμερούς γράφου με βάρη. Στην περίπτωση αυτή το ένα σύνολο των κορυφών (U) του διμερούς γράφου (G) αντιστοιχεί στο σύνολο των πομπών (T), ενώ το άλλο σύνολο κορυφών (V) στο σύνολο των δεκτών (R). Οι ακμές του γράφου αντιστοιχούν στα δεδομένα που θέλουμε να στείλουμε, ενώ τα βάρη στις ακμές στον χρόνο αποστολής κάθε δεδομένου αντίστοιχα. Δηλαδή αν η ακμή (u,v) ∈ E αντιστοιχεί στο δεδομένο d, τότε θα ισχύει s(d)=u, r(d)=v και t(d)= C[u,v]. Έτσι το στιγμιότυπο του παραπάνω παραδείγματος μπορεί να απεικονιστεί με τον ακόλουθο γράφο:



Σχήμα 4.2 : απεικόνιση γράφου στιγμιότυπου του προβλήματος.

Προφανώς, μία ομάδα αποστολής M για το πρόβλημά μας, θα αποτελεί ένα matching στον γράφο του προβλήματος, και ο χρόνος αποστολής της ομάδας θα είναι ίσος με το κόστος του αντίστοιχου matching (η μέγιστη τιμή των βαρών των ακμών του M), προσαυξημένη κατά την τιμή του χρόνου καθορισμού : time(M)= C[M]+d. Αν απαιτείται η αποστολή l ομάδων, ο συνολικός χρόνος αποστολής των δεδομένων θα είναι T=$\sum_{i=1}^{l} C[Mi] + l \cdot d$. Στο σχήμα που ακολουθεί φαίνονται τα matchings M1 και M2 που αντιστοιχούν στις ομάδες αποστολής που επιλέξαμε για την επίλυση του στιγμιότυπου του παραδείγματός.



Σχήμα 4.3 : (α) matching M1 κόστους 4, (β) matching M2 κόστους 2.

4.1.3 Πολυπλοκότητα του προβλήματος

Προκειμένου να βρούμε την βέλτιστη λύση στο πρόβλημα αυτό, θα πρέπει να βρούμε όλους τους πιθανούς συνδυασμούς ομάδων (matchings), που καλύπτουν στο σύνολό τους, όλα τα προς αποστολή δεδομένα. Κατόπιν να επιλέξουμε ως βέλτιστη, την λύση εκείνη στην οποία ο συνολικός χρόνος αποστολής (T) είναι ο ελάχιστος. Στην περίπτωση αυτή ο χώρος αναζήτησης πιθανών λύσεων αυξάνεται εκθετικά ως προς το μέγεθος των δεδομένων του προβλήματος, γεγονός που καθιστά συχνά ανέφικτη την εύρεση της βέλτιστης λύσης. Συγκεκριμένα το πρόβλημα αυτό έχει αποδειχθεί πως ανήκει στην κλάση πολυπλοκότητας NP. Η απόδειξη ότι το πρόβλημα ανήκει στην κλάση προβλημάτων NP-complete γίνεται με αναγωγή στο πρόβλημα σχεδιασμού χρονοπίνακα (timetable design problem) [GW85]. Προκειμένου λοιπόν να βρίσκουμε λύσεις σε εφικτό χρόνο, χρησιμοποιούμε προσεγγιστικούς αλγορίθμους που μας δίνουν λύσεις «κοντά» στην βέλτιστη. Για να βρούμε πόσο «κοντά» στην βέλτιστη, είναι η λύση που μας δίνει ένας προσεγγιστικός αλγόριθμος, και να χαρακτηρίσουμε έναν αλγόριθμο, θα πρέπει να υπολογίσουμε τον παράγοντα προσέγγισης (approximation ratio) και συνεπώς να συγκρίνουμε την λύση που μας δίνει ο αλγόριθμος με την βέλτιστη. Επειδή, όπως αναφέραμε, ο υπολογισμός της βέλτιστης λύσης είναι ανέφικτος, είναι απαραίτητο να βρούμε έναν τρόπο να υπολογίζουμε ένα κάτω φράγμα (lower bound) για την βέλτιστη λύση, δηλαδή μία τιμή από την οποία η βέλτιστη λύση θα είναι πάντα μεγαλύτερη ή ίση, και να χρησιμοποιήσουμε αυτή για τον υπολογισμό του παράγοντα προσέγγισης. Στο πρόβλημά μας ο συνολικός χρόνος της λύσης είναι το άθροισμα ανάμεσα σε δύο προσθετέους. Ο πρώτος είναι το γινόμενο της καθυστέρησης καθορισμού και του πλήθους των ομάδων αποστολής- matchings που απαιτούνται. Αν με G συμβολίζουμε τον γράφο του προβλήματός μας, τότε το πλήθος των matchings που απαιτούνται θα είναι τουλάχιστον ίσο με τον μέγιστο βαθμό του γράφου Δ(G). Ο δεύτερος είναι το άθροισμα των βαρών των matchings που απαιτούνται για την αποστολή των δεδομένων. Δεδομένου ότι ακμές με κοινή κορυφή θα ανήκουν σε διαφορετικό matching, ο προσθετέος αυτός θα έχει τιμή τουλάχιστον ίση με το συνολικό βάρος της κορυφής του G με το μεγαλύτερο βάρος, δηλαδή W(G). Επομένως το κάτω φράγμα στο οποίο καταλήξαμε και θα χρησιμοποιήσουμε για το πρόβλημά μας είναι : $d \cdot \Delta(G) + W(G)$. Στο παράδειγμα της προηγούμενης ενότητας έχουμε Δ(G)=2 και W(G)=6, οπότε για d=1 το κάτω φράγμα προκύπτει $1 \cdot 2 + 6 = 8$ χρονικές μονάδες , και επομένως η επιλογή των matchings που κάναμε στο παράδειγμα, μας δίνει την βέλτιστη λύση.

4.1.4 Εφαρμογές

Στο σημείο αυτό θα παρουσιάσουμε κάποιες τεχνολογικές εφαρμογές, στις οποίες εμφανίζεται το πρόβλημα που περιγράφουμε.

4.1.4.1 *Συστήματα Δορυφορικών Επικοινωνιών SS/TDMA*

Τα συστήματα SS/TDMA (satellite switched/time division multiple access), είναι συστήματα δορυφορικών επικοινωνιών ανάμεσα σε σταθμούς, που βρίσκονται σε διαφορετικές γεωγραφικές περιοχές[GW85],[JWB97]. Ολόκληρη η περιοχή της γης, είναι χωρισμένη σε έναν αριθμό από γεωγραφικές ζώνες. Ένας δορυφόρος, μέσω μίας ισχυρής κεραίας, μπορεί να καλύπτει πολλές τέτοιες ζώνες, και χρησιμοποιώντας έναν μεταγωγέα (switch), επιτρέπει την διασύνδεση ανάμεσα σε εισερχόμενες (uplink) και εξερχόμενες (downlink) ζεύξεις. Σύμφωνα με την συγκεκριμένη διαμόρφωση, η συνολική προς αποστολή πληροφορία, χωρίζεται και αποστέλλεται σε πλαίσια αποστολής (TDMA frames). Το κάθε πλαίσιο αντίστοιχα χωρίζεται σε χρονοσχισμές (time slots), στην κάθε μία από της οποίας ανατίθεται η αποστολή ενός συγκεκριμένου μηνύματος ή ενός τμήματος αυτού, αναλόγως το μέγεθος της σχισμής και του μηνύματος. Ένας περιορισμός που προκύπτει, είναι ότι ο μεταγωγέας δεν μπορεί να στέλνει ταυτόχρονα (στο ίδιο πλαίσιο), μηνύματα τα οποία προέρχονται από την ίδια ζώνη ή προορίζονται προς την ίδια ζώνη.

Προκειμένου να βελτιστοποιήσουμε την λειτουργία ενός τέτοιου συστήματος, πρέπει να βρούμε μία ανάθεση χρονοσχισμών σε πλαίσια TDMA , έτσι ώστε να έχουμε την αποστολή όλων των μηνυμάτων χωρίς συγκρούσεις. Ακόμα πρέπει να μεγιστοποιήσουμε την χρησιμοποίηση (utilization) του μεταγωγέα, και να ελαχιστοποιήσουμε τον συνολικό χρόνο αποστολής των δεδομένων. Χρησιμοποιώντας την ορολογία του προβλήματος μας, στον διμερή γράφο δεδομένων G((U-V),E), οι κορυφές που ανήκουν στο σύνολο U, θα αντιστοιχούν στις ζώνες αποστολής των δεδομένων, ενώ οι υπόλοιπες στις ζώνες προορισμού των δεδομένων. Οι ακμές θα αντιστοιχούν στα δεδομένα που θέλουμε να στείλουμε, ενώ το κόστος τους, σε χρονικές μονάδες αποστολής. Η διάσπαση των μηνυμάτων (preemption), πραγματοποιείται μέσω του ορισμού μήκους των χρονοσχισμών. Τέλος το κόστος καθορισμού αντιστοιχεί στον χρόνο που απαιτείται για τον καθορισμό από τον μεταγωγέα, των μηνυμάτων που θα αποσταλούν.

4.1.4.2 *Οπτικά Δίκτυα WDM*

Τα οπτικά δίκτυα WDM (wavelength division multiplexing) στις μέρες μας, αποτελούν την επικρατούσα τάση για την ανάπτυξη ευρυζωνικών δικτύων. Παρέχουν εξασφαλισμένη ποιότητα υπηρεσιών και μεγάλο διαθέσιμο εύρος ζώνης, καθιστώντας τα, απαραίτητα για την εξυπηρέτηση ενός συνόλου σύγχρονων ευρυζωνικών υπηρεσιών όπως, πολυμεσικών, τηλεματικών και γρήγορης πλοήγησης στο διαδίκτυο. Σύμφωνα με την διαμόρφωση αυτή, μπορεί να χωριστεί μία σύνδεση οπτικής ίνας, σε ένας πλήθος καναλιών, ανάλογα με την συχνότητα. Κάθε κανάλι που δημιουργείται, αντιστοιχεί σε ένα συγκεκριμένο μήκος κύματος (χρώμα).

Χρησιμοποιείται ένας πολυπλέκτης (multiplexer) στον πομπό, ώστε να πολυπλέξει διάφορα σήματα προς μετάδοση σε διαφορετικά κανάλια, και αντίστοιχα ένα αποπλέκτης (demultiplexer) στον δέκτη για να ξεχωρίσει ξανά τα λαμβανόμενα σήματα. Τα σύγχρονα συστήματα υποστηρίζουν την ταυτόχρονη αποστολή έως και 160 διαφορετικών σημάτων. Τα συστήματα αυτά χρησιμοποιούνται ιδιαίτερα και στις τηλεπικοινωνίες, καθώς οι εταιρίες τηλεπικοινωνιών έχουν την δυνατότητα να αυξήσουν την χωρητικότητα του δικτύου τους χωρίς την εγκατάσταση νέων γραμμών.

Στην εφαρμογή αυτή, όπως και στην προηγούμενη, έχουμε την δυνατότητα διάσπασης των μηνυμάτων που θέλουμε να αποστείλουμε σε μικρότερα, χρησιμοποιώντας χρονοσχισμές αντίστοιχου μεγέθους στα κανάλια. Χρησιμοποιώντας τους συμβολισμούς του προβλήματος μας, ο πίνακας ζήτησης εκπομπής του WDM, αντιστοιχεί στον πίνακα δεδομένων του προβλήματος. Στόχος μας είναι, ο υπολογισμός ενός χρονοδιαγράμματος εκπομπής των δεδομένων στις χρονοσχισμές των καναλιών, ταυτόχρονα μεγιστοποιώντας την χρησιμοποίηση του δικτύου, και ελαχιστοποιώντας τον χρόνο αποστολής.[SWW+02]

4.1.4.3 Δίκτυα Μεταγωγής Πακέτου

Στα δίκτυα μεταγωγής πακέτων (packet switching networks), έχουμε ένα πλήθος μηνυμάτων, τα οποία θέλουμε να στείλουμε, από κάποιον αρχικό κόμβο, σε κάποιο κόμβο προορισμού. Το κάθε μήνυμα, χωρίζεται σε μικρότερα τμήματα, τα οποία ονομάζονται πακέτα. Κάθε πακέτο αποστέλλεται αυτόνομα, από τα υπόλοιπα πακέτα που ανήκουν στο ίδιο μήνυμα. Ένα μήνυμα, θεωρούμε ότι έχει αποσταλεί πλήρως, όταν όλα τα πακέτα που έχουν προκύψει από την διάσπαση αυτού, έχουν φτάσει στον κόμβο-προορισμό και συνεπώς αυτός, είναι σε θέση να συγκροτήσει το αρχικό μήνυμα.

Το δίκτυο που υπάρχει ανάμεσα στους δύο ακραίους κόμβους, θεωρούμε ότι αποτελείται από μεταγωγείς (switches). Θεωρούμε ότι οι μεταγωγείς αυτοί, είναι σε θέση να στέλνουν οποιοδήποτε πακέτο φτάσει σε αυτούς στον επόμενο κόμβο, χωρίς να έχουμε απόρριψη πακέτων. Κάθε μεταγωγός έχει κάποιες ζεύξεις εισόδου, από τις οποίες δέχεται πακέτα δεδομένων, και κάποιες ζεύξεις εξόδου, όπου στέλνει πακέτα προς τον επόμενο μεταγωγό ή τον κόμβο προορισμού. Ο μεταγωγός λειτουργεί με τον περιορισμό, ότι δεν μπορεί να στέλνει πακέτα προς την ίδια ζεύξη εξόδου, τα οποία προέρχονται από διαφορετικές ζεύξεις εισόδου, και αντίστοιχα δεν μπορεί να προωθεί ταυτόχρονα, πακέτα που προέρχονται από την ίδια ζεύξη, και προορίζονται σε διαφορετική ζεύξη εξόδου.

Σκοπός, για το δίκτυο αυτό είναι να βρούμε έναν αλγόριθμο χρονοδρομολόγησης, για κάθε μεταγωγό, έτσι ώστε να μεγιστοποιείται η χρησιμοποίηση των μεταγωγών και συνεπώς του δικτύου, και συνάμα να ελαχιστοποιείται ο χρόνος που κάνει ένα πακέτο για να φτάσει από τον κόμβο αποστολής στον κόμβο προορισμού (packet delay). Τον χρόνο αυτό, καταφέρνουμε να τον μειώσουμε, ελαττώνοντας τον χρόνο που περιμένει ένα πακέτο στον μεταγωγό για αποστολή (queuing delay), καθώς ο χρόνος μετάδοσης των πακέτων σε κάθε ζεύξη, εξαρτάται από τα χαρακτηριστικά του δικτύου. Χρησιμοποιώντας τον συμβολισμό του προβλήματός μας, ένας διμερής γράφος δεδομένων, αντιστοιχεί σε κάθε μεταγωγό. Συγκεκριμένα, οι δύο ομάδες κορυφών αντιστοιχούν στις ζεύξεις εισόδου και εξόδου αντίστοιχα, ενώ οι ακμές αντιστοιχούν στις εσωτερικές συνδέσεις του μεταγωγού, ώστε να προωθήσει τα πακέτα από τις ζεύξεις εισόδου, στις ζεύξεις εξόδου. Τα κόστη στις ακμές, αντιστοιχούν στον χρόνο που απαιτείται για την προώθηση κάθε πακέτου, και είναι ανάλογα με το μέγεθός του. [PHI97]

4.1.4.4 Μεταφορά δεδομένων σε συστήματα πολυεπεξεργαστών

Θεωρούμε ένα υπολογιστικό σύστημα, κατανεμημένο ή μη, στο οποίο έχουμε έναν αριθμό από επεξεργαστές, και έναν αριθμό από υποσυστήματα εισόδου-εξόδου (I/O subsystems). Λόγω της ραγδαίας αύξησης της τεχνολογίας τις τελευταίες δεκαετίες, έχει επιτευχθεί η συνεχής αύξηση της ταχύτητας των επεξεργαστών που κατασκευάζονται, χωρίς όμως η αύξηση της ταχύτητας των μονάδων εισόδου-εξόδου να είναι αντίστοιχη. Αποτέλεσμα αυτού, είναι η δημιουργία συμφόρησης (bottleneck) στις εργασίες I/O σε πολλές εφαρμογές, όπως πολυμεσικές, επιστημονικών υπολογισμών και βάσεις δεδομένων. Λύση στο πρόβλημα αυτό, μπορεί να δώσει η κατάλληλη χρονοδρομολόγηση της ανταλλαγής δεδομένων ανάμεσα σε επεξεργαστές και μονάδες I/O [DJT94].

Συγκεκριμένα, η μοντελοποίηση του προβλήματος μπορεί να είναι η ακόλουθη: θεωρούμε έναν διμερή γράφο με βάρη G(U-V,E) , κατά αντιστοιχία με τον γράφο δεδομένων του αρχικού μας προβλήματος. Στον γράφο αυτό, η μία ομάδα κορυφών αντιστοιχεί στους επεξεργαστές του συστήματος, και η άλλη στις μονάδες I/O, οι οποίες ας θεωρήσουμε ότι είναι δίσκοι. Κάθε ακμή του G, αντιστοιχεί σε κάποια λειτουργία I/O που πρέπει να γίνει ανάμεσα στις μονάδες που ορίζουν τα άκρα της. Το κόστος της κάθε ακμής, αντιστοιχεί στον χρόνο που απαιτείται την αντίστοιχη λειτουργία. Η μεταφορά δεδομένων πάντα ξεκινάει από τους επεξεργαστές, όμως η ροή, μπορεί να είναι αμφίδρομη(π.χ. read-write). Δεδομένης της σύγχρονης αρχιτεκτονικής των επεξεργαστών και εν γένει των υπολογιστικών συστημάτων, κάθε αποστολή έχει συγκεκριμένο μέγεθος δεδομένων, ίσο με το block της μνήμης. Μεγαλύτερα μηνύματα, διασπώνται (preemption) στα όρια του block. Τέλος, θεωρούμε ότι ανάμεσα στους επεξεργαστές και τους δίσκους δεδομένων, δεν υπάρχει κάποια κοινή μνήμη.

Όλοι οι επεξεργαστές, μπορούν να στέλνουν δεδομένα ταυτόχρονα, και όλοι οι δίσκοι μπορούν να λαμβάνουν δεδομένα ταυτόχρονα και αντίστροφα. Ο περιορισμός που υπάρχει, είναι ότι κάθε μονάδα (επεξεργαστής ή δίσκος) δεν μπορεί να πραγματοποιεί περισσότερες από μία συναλλαγή, την ίδια χρονική στιγμή. Στόχος είναι, η εύρεση ενός αλγόριθμου χρονοδρομολόγησης για την μεταφορά των δεδομένων στον ελάχιστο δυνατό χρόνο. Δηλαδή, η εύρεση ομάδων μεταφοράς δεδομένων (matching στον γράφο G), που μπορούν να πραγματοποιηθούν ταυτόχρονα, όπου ο συνολικός χρόνος μεταφοράς όλων των δεδομένων να είναι ο ελάχιστος. Δεδομένου, ότι η χρόνος που απαιτείται για την χρονοδρομολόγηση, πρέπει να είναι σημαντικά μικρότερος από τον χρόνο αποστολής των δεδομένων, είναι εμφανής η ανάγκη για την εύρεση αλγορίθμων που δίνουν λύση σε πολυωνυμικό χρόνο.

4.2 Οι αλγόριθμοι

Στο σημείο αυτό θα παρουσιάσουμε και θα αναλύσουμε τρεις προσεγγιστικούς αλγορίθμους που επιλύουν το πρόβλημα.

4.2.1 Ο αλγόριθμος G-W

4.2.1.1 Εισαγωγή

Ο αλγόριθμος αυτός παρουσιάστηκε από τους Inder S. Gopal και C. K. Wong το 1985 [GW85]. Ο αλγόριθμος G-W σκοπό έχει να επιλύσει το πρόβλημα στην εφαρμογή του σε ένα σύστημα SS/TDMA. Για τον λόγο αυτό προκύπτει μία νέα παράμετρος εισόδου του αλγορίθμου, που είναι ο αριθμός των transponders του συστήματος. Τον αριθμό αυτόν θα τον συμβολίζουμε με K και θα μας υποδεικνύει τον μέγιστο αριθμό ταυτόχρονων αποστολών που μπορούμε να έχουμε στο σύστημά μας. Προφανώς, στην περίπτωση που ο αριθμός K είναι μεγαλύτερος ή ίσος από την μικρότερη από τις διαστάσεις του πίνακα του προβλήματος, τότε δεν επηρεάζει τον αριθμό των ταυτόχρονων αποστολών που μπορούμε να έχουμε. Οι σχεδιαστές του αλγορίθμου χρησιμοποιούν την ιδέα, πως μπορούμε να πάρουμε μία «καλή» λύση του προβλήματος, δηλαδή μία λύση που θα είναι κοντά στην βέλτιστη, αν στείλουμε όλα τα δεδομένα που έχουμε, με τον μικρότερο αριθμό αποστολών. Δηλαδή αν ελαχιστοποιήσουμε το πλήθος των ομάδων αποστολής. Αυτό ακριβώς επιτυγχάνει ο αλγόριθμος G-W.

4.2.1.2 Είσοδος του αλγορίθμου- ορισμοί

Σαν είσοδο, αρχικά, πρέπει να έχουμε τα δεδομένα που πρέπει να στείλουμε, και τις απαραίτητες πληροφορίες για αυτά. Τα στοιχεία αυτά τα δεχόμαστε συγκεντρωμένα στον πίνακα δεδομένων του προβλήματός μας, που αντιστοιχεί στον διμερή γράφο απεικόνισης του στιγμιότυπου που επιθυμούμε να επιλύσουμε. Τον πίνακα αυτόν θα συμβολίζουμε με D και θα είναι εν γένει ένας mxn πίνακας. Τα στοιχεία του, θεωρούμε ότι είναι μη αρνητικοί, ακέραιοι αριθμοί, αφού αντιπροσωπεύουν τον χρόνο αποστολής κάποιου δεδομένου. Αναφέραμε πως K θα είναι ο αριθμός των transponders που έχουμε. Έτσι, το πλήθος των ταυτόχρονων εκπομπών, δηλαδή το πλήθος των στοιχείων που θα περιέχει μια ομάδα αποστολής-matching M, δεν θα μπορεί να είναι μεγαλύτερο από το μικρότερο εκ των K,m,n. Επομένως θα ισχύει: $\forall\, M : |M| \le \min\{K, m, n\}$.

Θα συμβολίζουμε με B τον αριθμό των αποστολών που απαιτούνται, δηλαδή το πλήθος των matchings που θα παίρνουμε στον γράφο. Αν για τον πίνακα D, με r_i συμβολίζουμε το πλήθος των μη-μηδενικών στοιχείων που υπάρχουν στην γραμμή i, με c_j το πλήθος των μη-μηδενικών στοιχείων που υπάρχουν στην στήλη j, και με T το πλήθος των μη-μηδενικών στοιχείων που έχει συνολικά ο πίνακας D, τότε για την τιμή του B θα ισχύει: $\forall\, i, j : 1 \le i \le m\, , 1 \le j \le n \;\; B \ge \max\left\{\max(r_i)\, , \max(c_j)\, , \left\lceil \frac{T}{K} \right\rceil\right\}$. Με το τύπο αυτό, καταφέρνουμε να υπολογίσουμε ένα κάτω φράγμα για το πλήθος των αποστολών που πρέπει να κάνουμε.

4.2.1.3 Τα βήματα του αλγορίθμου

Αρχικά κατασκευάζουμε τον διμερή γράφο G_D = (U,V,E) που αντιστοιχεί στον πίνακα D του προβλήματος που καλούμαστε να επιλύσουμε. Προσθέτουμε τις κορυφές u_{m+1}, u_{m+2}, …,u_{m+n-K} στο σύνολο U, και τις κορυφές v_{m+1}, v_{m+2}, …,v_{m+n-K} στο σύνολο V. Υπολογίζουμε τον ελάχιστο δυνατό αριθμό από matchings B που πρέπει να πάρουμε, χρησιμοποιώντας τον τύπο του κάτω φράγματος της προηγούμενης ενότητας.

Στη συνέχεια προσθέτουμε ακμές, με τρόπο τέτοιο, ώστε κάθε κορυφή να έχει βαθμό ακριβώς ίσο με B. Συγκεκριμένα, για κάθε κορυφή $u_i \in \{ u_{m+1}, u_{m+2}, \ldots, u_{m+n-K} \}$ όπου $d(u_i) < B$, βρίσκουμε μία κορυφή $v_j \in \{ v_1, v_2, \ldots, v_n \}$ όπου $d(v_j) < B$, και προσθέτουμε την ακμή (u_i,v_j) στο σύνολο E των ακμών του G_D. Επαναλαμβάνουμε την παραπάνω διαδικασία, μέχρις ότου κάθε ακμή στο σύνολο $\{ u_{m+1}, u_{m+2}, \ldots, u_{m+n-K} \}$ να έχει βαθμό ίσο με B. Έπειτα, αντίστοιχα για κάθε κορυφή $v_j \in \{ v_{n+1}, v_{n+2}, \ldots, v_{m+n-K} \}$ όπου $d(v_j) < B$, βρίσκουμε μία κορυφή $u_i \in \{ u_1, u_2, \ldots, v_m \}$ όπου $d(v_j) < B$, και προσθέτουμε την ακμή (u_i,v_j) στο σύνολο E των ακμών του G_D. Επαναλαμβάνουμε την παραπάνω διαδικασία, μέχρις ότου κάθε ακμή στο σύνολο $\{ v_{n+1}, v_{n+2}, \ldots, v_{m+n-K} \}$ να έχει βαθμό ίσο με B. Τέλος για κάθε κορυφή $u_i \in \{ u_1, u_2, \ldots, u_m\}$ όπου $d(u_i) < B$, βρίσκουμε μία κορυφή $v_j \in \{ v_1, v_2, \ldots, v_n \}$ όπου $d(v_j) < B$, και προσθέτουμε την ακμή (u_i,v_j) στο σύνολο E των ακμών του G_D. Επαναλαμβάνουμε την παραπάνω διαδικασία, μέχρις ότου κάθε ακμή στο σύνολο $\{ u_1, u_2, \ldots, u_m\}$ να έχει βαθμό ίσο με B.

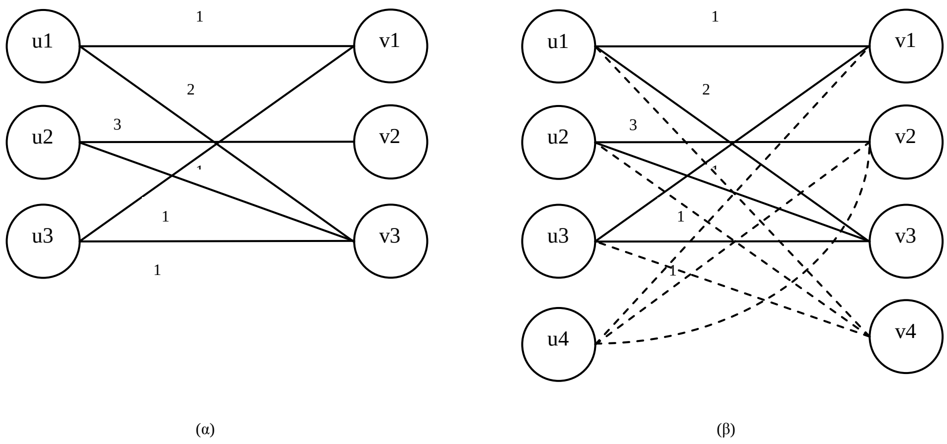
Αφού προσθέσαμε τις κορυφές και τις ακμές στον γράφο G_D, θεωρούμε πως οι νεοεισαχθείσες ακμές έχουν βάρος ίσο με μηδέν. Ταξινομούμε όλες τις ακμές που ανήκουν στο σύνολο E σε αύξουσα σειρά, ως προς την τιμή του βάρους τους. Έτσι προκύπτει η εξής διάταξη των ακμών $e_1,e_2,\ldots,e_{|E|}$.

Δημιουργούμε δύο σύνολα, το σύνολο των ακμών P και το σύνολο του matching Q, καθώς και δύο μεταβλητές i και j. Αρχικοποιούμε i,j=1 και $P \leftarrow \{e_i\}$, $Q \leftarrow \{e_i\}$. Θέτουμε i=i+1 και $P \leftarrow P \cup e_i$. Στο σύνολο Q τοποθετούμε τις ακμές που αποτελούν το τρέχον matching, έτσι αν η ακμή e_i μπορεί να προστεθεί στο Q, την προσθέτουμε, αλλιώς επαναλαμβάνουμε το βήμα του αλγορίθμου. Επαναλαμβάνουμε το βήμα έως ότου |Q|= m+n-K, όπου πλέον έχουμε σχηματίσει το matching το οποίο πρέπει να επιλέξουμε.

Στο τελευταίο βήμα του αλγορίθμου δημιουργούμε τον πίνακα αποστολής. Ο πίνακας αποστολής S(j) θα είναι ένας mxn πίνακας, που θα σχηματιστεί από τα μη-μηδενικά στοιχεία που ανήκουν στο σύνολο Q. Θέτουμε $P \leftarrow P-Q$, αναζητούμε matching μέγιστου μεγέθους στο σύνολο P και, αν αυτό υπάρχει, θέτουμε το Q ίσο με αυτό. Αν j<B τότε θέτουμε j=j+1 και επαναλαμβάνουμε τα δύο τελευταία βήματα, αλλιώς σταματάμε καθώς έχουμε καλύψει όλα τα στοιχεία του αρχικού πίνακα.

4.2.1.4 Παράδειγμα Εφαρμογής του Αλγορίθμου

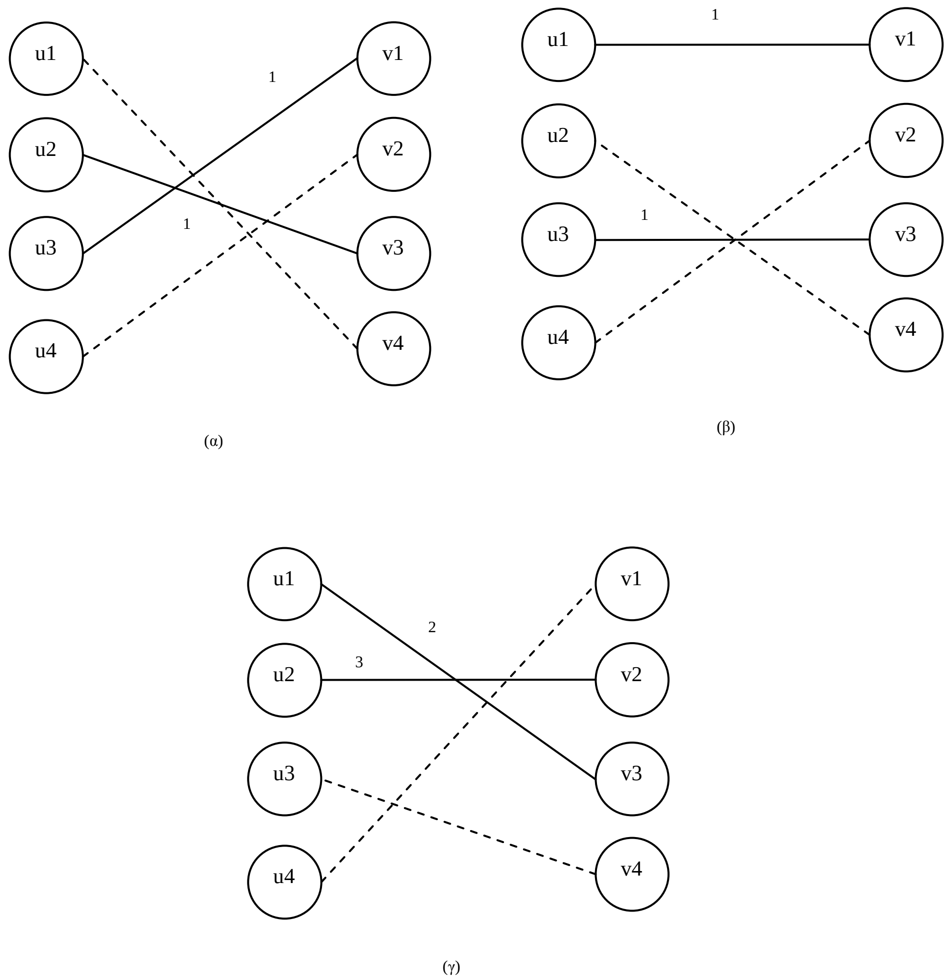
Έστω ότι έχουμε να επιλύσουμε το ακόλουθο πρόβλημα: $D=\begin{bmatrix}1 & 0 & 2\\ 0 & 3 & 1\\ 1 & 0 & 1\end{bmatrix}$, K=2, set-up delay *d*=1. Προφανώς θα ισχύει m=n=3. Επίσης $\max\{r_i\}=2$, $\max\{c_j\}=3$ και T=6 ⇒ T/K = 3, επομένως το κάτω φράγμα για το B θα είναι max{2,3,3}=3. Κατασκευάζουμε τον διμερή γράφο G_D και προσθέτουμε τις κορυφές και τις ακμές που χρειάζονται, όπως ορίζει ο αλγόριθμος.



Σχήμα 4.4: (α) γράφος G_D του προβλήματος , (β) επεκταμένος γράφος G_D'

Στο παραπάνω σχήμα βλέπουμε την απεικόνιση του γράφου G_D που αντιστοιχεί στα δεδομένα του πίνακα D, και τον γράφο που προκύπτει μετά την προσθήκη νέων κορυφών και ακμών σύμφωνα με τον αλγόριθμο. Στο παράδειγμά μας m+n-K=3+3-2=4 επομένως προσθέσαμε τις κορυφές u4 και v4. Προκειμένου να αποκτήσουν όλες οι κορυφές βαθμό ίσο με B=3, προσθέτουμε τις ακμές που φαίνονται στο σχήμα με διακεκομμένη γραμμή και έχουν βάρος μηδέν.

Στη συνέχεια δημιουργούμε μια διάταξη των ακμών του γράφου G_D', σε αύξουσα σειρά ως προς τα βάρη τους. Στη διάταξη αυτή, τα πρώτα στοιχεία θα αντιστοιχούν στις ακμές που προσθέσαμε στον αρχικό γράφο, οι οποίες έχουν βάρος ίσο με μηδέν. Έπειτα δημιουργούμε τα matchings που αντιστοιχούν στις ομάδες αποστολής μηνυμάτων. Σύμφωνα με τον αλγόριθμο, επιδιώκουμε κάθε matching να έχει μέγιστο βαθμό- πληθικότητα, δηλαδή για το παράδειγμά μας, να αποτελείται από τέσσερεις ακμές. Τα τρία matchings προκύπτουν από την εφαρμογή του αλγορίθμου, απεικονίζονται ,κατά σειρά, στο ακόλουθο σχήμα. Με διακεκομμένη γραμμή εμφανίζονται και πάλι οι ακμές μηδενικού κόστους.



Σχήμα 4.5 : Matchings που αντιστοιχούν στις τρείς αποστολές δεδομένων του προβλήματος μας, (α) M_1, (β) M_2, (γ)M_3.

Από τα matchings που επιλέξαμε, προκύπτουν οι πίνακες αποστολής S(j), 1≤j≤3. Συγκεκριμένα, έχουμε τους ακόλουθους πίνακες : $S(1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$, $S(2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $S(3) = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. Προφανώς, τα μηδενικά στοιχεία των πινάκων, δεν απαιτούν την μετάδοση κάποιων δεδομένων. Ως επιβεβαίωση πως έχουν αποσταλεί όλα τα δεδομένα, θα πρέπει από το άθροισμα των πινάκων αποστολής να προκύπτει ο αρχικός πίνακας δεδομένων D. Πράγματι $S(1) + S(2) + S(3) = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 1 \\ 1 & 0 & 1 \end{bmatrix} = D$. Ο χρόνος αποστολής για κάθε πίνακα θα είναι ίσος με το μεγαλύτερο στοιχείο του. Έτσι οι χρόνοι αποστολής για τους πίνακες S(1),S(2),S(3) θα είναι αντίστοιχα 1,1 και 3 χρονικές μονάδες. Δεδομένου ότι έχουμε τρείς αποστολές και ότι d=1 ο συνολικός χρόνος αποστολής θα είναι : $3 \cdot 1 + (1 + 1 + 3) = 3 + 5 = 8$ χρονικές μονάδες.

4.2.1.5 Παράγοντας προσέγγισης

Ο αλγόριθμος G-W βρίσκει μία λύση, η οποία χρησιμοποιεί το μικρότερο δυνατό πλήθος από ομάδες αποστολής, όμως ο συνολικός χρόνος αποστολής, που προκύπτει από την εφαρμογή του αλγορίθμου, δεν είναι εν γένει ο μικρότερος δυνατός, και γι' αυτό άλλωστε είναι ένας προσεγγιστικός αλγόριθμος. Επίσης, δεν μας δίνει κάποια «εγγύηση» για το πόσο κοντά στην βέλτιστη, θα είναι η λύση που δίνει ο αλγόριθμος, επιτρέποντάς μας να υπολογίσουμε τον παράγοντα προσέγγισης (ρ) του. Παρακάτω θα αποδείξουμε, πως ο παράγοντας προσέγγισης του αλγορίθμου G-W δεν είναι φραγμένος, και κατά συνέπεια η λύση του αλγορίθμου, μπορεί να είναι απείρως μεγαλύτερη από την βέλτιστη λύση.

Έστω ότι καλούμαστε να επιλύσουμε, χρησιμοποιώντας τον αλγόριθμο G-W, το στιγμιότυπο ενός προβλήματος, το οποίο έχει τον εξής πίνακα δεδομένων : $D = \begin{bmatrix} a & 2a & ra \\ 3a & ra & 2a \\ ra & 2a & a \end{bmatrix}$, όπου α είναι ένας θετικός ακέραιος, και r ακέραιος με r>3. Μετά την εφαρμογή του αλγορίθμου G-W θα προκύψουν οι ακόλουθοι πίνακες αποστολής:

$S(1) = \begin{bmatrix} a & 0 & 0 \\ 0 & ra & 0 \\ 0 & 0 & a \end{bmatrix}$, $S(2) = \begin{bmatrix} 0 & 2a & 0 \\ 0 & 0 & 2a \\ ra & 0 & 0 \end{bmatrix}$, $S(3) = \begin{bmatrix} 0 & 0 & ra \\ 3a & 0 & 0 \\ 0 & 2a & 0 \end{bmatrix}$.

Επομένως ο συνολικός χρόνος αποστολής που προκύπτει από την λύση αυτή θα είναι $SOL = 3 \cdot d + 3 \cdot ra$. Το κάτω φράγμα του χρόνου αποστολής θα είναι $\Delta(G) \cdot d + W(G) = 3 \cdot d + ra + 3a + 2a$. Μία άλλη λύση του στιγμιότυπου θα ήταν, να είχαμε χρησιμοποιήσει τους ακόλουθους πίνακες αποστολής: $S(1) = \begin{bmatrix} 0 & 0 & ra \\ 0 & ra & 0 \\ ra & 0 & 0 \end{bmatrix}$, $S(2) = \begin{bmatrix} a & 0 & 0 \\ 0 & 0 & 2a \\ 0 & 2a & 0 \end{bmatrix}$, $S(3) = \begin{bmatrix} 0 & 2a & 0 \\ 3a & 0 & 0 \\ 0 & 0 & a \end{bmatrix}$, σύμφωνα με τους οποίους, ο συνολικός χρόνος είναι $T = 3 \cdot d + ra + 2a + 3a$ και είναι βέλτιστος, αφού είναι ίσος με το κάτω φράγμα. Συνεπώς $OPT = 3 \cdot d + ra + 3a + 2a$. Ο παράγοντας προσέγγισης θα είναι :

$$\rho = \frac{SOL}{OPT} = \frac{3d + 3ra}{3d + ra + 3a + 2a}$$

Έστω πως η τιμή α συνδέεται με τον χρόνο καθορισμού d μέσω της σχέσης $\alpha = k \cdot d$, όπου k είναι ένας θετικός πραγματικός αριθμός. Στην περίπτωση αυτή θα έχουμε :

$$\rho = \frac{3d + 3rkd}{3d + rkd + 3kd + 2kd} = \frac{3rk + 3}{rk + 5k + 3} = \frac{3r + \frac{3}{k}}{r + 5 + \frac{3}{k}}$$

Όπου για $k \to \infty : \rho = \frac{0}{r+5} \Rightarrow \rho \to \infty$.

4.2.1.6 Σχόλια

Σύμφωνα με το αποτέλεσμα της προηγούμενης ενότητας, βλέπουμε πως όσο μεγαλώνει η τιμή του k τόσο μεγαλώνει και ο παράγοντας προσέγγισης, και συνεπώς τόσο η λύση που μας δίνει ο αλγόριθμος, απομακρύνεται από την βέλτιστη. Ο παράγοντας k συνδέει την τιμή της μεταβλητής α, και επομένως τις τιμές του πίνακα δεδομένων D, με τον χρόνο καθορισμού *d*. Συγκεκριμένα, όσο μεγαλώνει η τιμή του k, τόσο μεγαλύτερες είναι οι τιμές του πίνακα σε σχέση με τον χρόνο *d*. Συνεπώς, στον υπολογισμό στου συνολικού χρόνου αποστολής των δεδομένων, όσο μεγαλύτερη τιμή έχει το k, μεγαλύτερο ρόλο έχουν οι χρόνοι αποστολής των δεδομένων και ο χρόνος καθορισμού μπορεί να θεωρηθεί αμελητέος. Έτσι ο αλγόριθμος G-W, οποίος σκοπό έχει να ελαχιστοποιήσει το πλήθος των αποστολών, ώστε να μειωθεί η συνεισφορά στον συνολικό χρόνο του χρόνου καθορισμού, δίνει λύσεις «μακριά» από την βέλτιστη.

Συμπερασματικά, μπορούμε να πούμε, πως η εφαρμογή του αλγορίθμου G-W ενδείκνυται, στις περιπτώσεις εκείνες όπου ο χρόνος καθορισμού είναι αρκετά μεγάλος, σε σχέση με τον χρόνο αποστολής των δεδομένων. Στην περίπτωση αυτή, ακόμα και μία επιπλέον αποστολή, από τις ελάχιστες δυνατές, μπορεί να επιβαρύνει χρονικά το αποτέλεσμα που θα πάρουμε, ώστε να απομακρυνθούμε αισθητά από τον χρόνο αποστολής της βέλτιστης λύσης.

4.2.2 Ο αλγόριθμος CXP

4.2.2.1 Εισαγωγή

Ο αλγόριθμος CXP αναπτύχθηκε από τους P. Crescenzi, D. Xiaotie και C. Papadimitrioiu το 2001. Από την συγκεκριμένη εργασία θα παρουσιάσουμε δύο αλγορίθμους. Ο πρώτος, είναι ένας απλός , σε υλοποίηση, προσεγγιστικός αλγόριθμος, ο οποίος μας δίνει λύσεις με παράγοντα προσέγγισης ίσο με 2. Επειδή όμως ο αλγόριθμος αυτός, είναι ψεύδο-πολυωνυμικός, στη συνέχεια προτείνεται ένας πιο εξελιγμένος αλγόριθμος, και πιο περίπλοκος, ο οποίος είναι επίσης 2-προσεγγιστικός, και μας δίνει λύσεις σε πολυωνυμικό χρόνο.

4.2.2.2 Είσοδος του αλγορίθμου

Σαν είσοδο, όπως και σε όλους τους αλγόριθμους που θα δούμε, πρέπει να έχουμε τα δεδομένα που πρέπει να στείλουμε, και τις απαραίτητες πληροφορίες για αυτά. Τα στοιχεία αυτά τα δεχόμαστε συγκεντρωμένα στον πίνακα δεδομένων του προβλήματός μας, που αντιστοιχεί στον διμερή γράφο απεικόνισης του στιγμιότυπου που επιθυμούμε να επιλύσουμε. Τον διμερή αυτό γράφο θα συμβολίζουμε με G. Τον πίνακα δεδομένων θα συμβολίζουμε με D και θα είναι εν γένει ένας mxn πίνακας. Τα στοιχεία του, θεωρούμε ότι είναι μη αρνητικοί, ρητοί αριθμοί, σε αντίθεση με τον αλγόριθμο G-W που δέχεται σαν είσοδο μόνο ακέραιες τιμές στον χρόνο αποστολής κάποιου δεδομένου. Στον αλγόριθμο CXP δεχόμαστε πως ο χρόνος καθορισμού (set-up delay) διαρκεί μία χρονική μονάδα, δηλαδή έχουμε d=1. Θεωρούμε δηλαδή, πως οι χρόνοι αποστολής των δεδομένων είναι κανονικοποιημένοι ως προς τον χρόνο καθορισμού. Συνεπώς, το κάτω φράγμα για την βέλτιστη λύση θα είναι Δ(G)+W(G). Τέλος, στους αλγόριθμους που θα δούμε, θεωρούμε ότι επιτρέπεται ο διάσπαση των δεδομένων (preemption) και η τμηματική τους αποστολή. Δηλαδή ότι ένας πομπός μπορεί να στείλει ένα τμήμα ενός δεδομένου, το οποίο αντίστοιχα θα απαιτεί χρόνο, ένα ποσοστό του συνολικού χρόνου αποστολής του δεδομένου. Φυσικά, για να ολοκληρωθεί η αποστολή των δεδομένων, πρέπει όλα τα δεδομένα να έχουν σταλεί στο σύνολό τους.

4.2.2.3 Ο αλγόριθμος CXP1

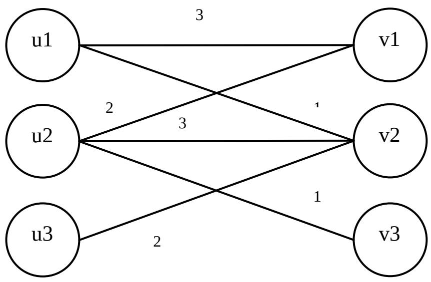
Στο σημείο αυτό, παρουσιάζουμε τα βήματα του πρώτου, από τους δύο αλγορίθμους που θα δούμε. Πρώτα στρογγυλοποιούμε προς τα πάνω, τις τιμές των ακμών του γράφου G.Δηλαδή $\forall\, e\ \in E, w(e) = \lceil w(e) \rceil$. Στη συνέχεια, χωρίζουμε κάθε ακμή e σε w(e) ακμές μοναδιαίου κόστους, και ονομάζουμε τον γράφο που προκύπτει H. Βρίσκουμε για τον γράφο H, ακριβώς Δ(H) matchings, το καθένα με κόστος ίσο με την μονάδα. Τα matchings που βρίσκουμε, αντιστοιχούν στις ομάδες αποστολής των δεδομένων, για το πρόβλημά μας.

4.2.2.4 Παράγοντας προσέγγισης

Έστω ότι s είναι το πλήθος των αποστολών που απαιτούνται από τον αλγόριθμο, για την αποστολή όλων των δεδομένων. Η λύση εν γένει θα απαιτεί $s + \sum w(Mi)$ χρονικές μονάδες, όπου ο δεύτερος όρος αντιστοιχεί στο άθροισμα των βαρών των matchings που επιλέγουμε. Ισχύει όμως s=Δ(H) και $\sum w(Mi) \leq$ Δ(H), επομένως SOL= $s + \sum w(Mi) \leq \Delta(H) + \Delta(H) = 2\Delta(H) \leq 2(\Delta(G) + W(G))$ όμως OPT=Δ(G)+W(G) ⇒ SOL≤2OPT ⇒ ο αλγόριθμος έχει παράγοντα προσέγγισης 2.

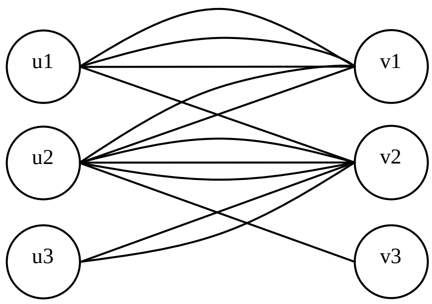
4.2.2.5 Παράδειγμα Εφαρμογής του Αλγορίθμου CXP1

Έστω ότι θέλουμε να χρησιμοποιήσουμε τον αλγόριθμο CXP1, για να επιλύσουμε ένα στιγμιότυπο ενός προβλήματος, το οποίο έχει ως πίνακα δεδομένων τον $D = \begin{bmatrix} 3 & 1 & 0 \\ 2 & 3 & 1 \\ 0 & 2 & 0 \end{bmatrix}$. Ο γράφος G που προκύπτει από τον πίνακα D, θα είναι αυτός που φαίνεται στο ακόλουθο σχήμα.



Σχήμα 4.6: γράφος G του προβλήματος.

Από τον γράφο G, βλέπουμε ότι Δ(G)=3 και W(G)=6, επομένως το κάτω φράγμα για την βέλτιστη λύση του στιγμιότυπου που έχουμε θα είναι Δ(G)+W(G)=3+6=9 χρονικές μονάδες. Στη συνέχεια, σύμφωνα με τον αλγόριθμο, δημιουργούμε τον γράφο H, διασπώντας κάθε ακμή e του G, σε w(e) ακμές μοναδιαίου κόστους.



Σχήμα 4.7: παραγόμενος γράφος H του προβλήματος.

Για τον γράφο H που προκύπτει, ισχύει Δ(H)=6 και W(H)=6. Επομένως η λύση που μας δίνει ο αλγόριθμος, θα αποτελείται από 6 matchings-αποστολές, όπου το καθένα θα απαιτεί μία χρονική μονάδα. Πράγματι, έχουμε $\begin{bmatrix} 3 & 1 & 0 \\ 2 & 3 & 1 \\ 0 & 2 & 0 \end{bmatrix} = 3 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 2 \cdot \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ και επομένως η λύση θα απαιτεί 6+(3+2+1)=12 χρονικές μονάδες.

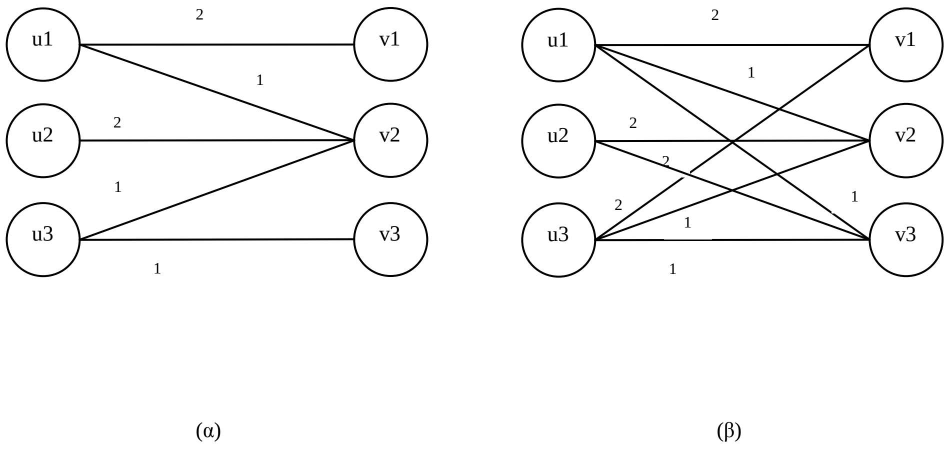
4.2.2.6 *Ο αλγόριθμος CXP2*

Ο αλγόριθμος CXP1 είναι ψεύδο-πολυωνυμικός, καθώς ο χρόνος που χρειάζεται για επιστρέψει την λύση στο πρόβλημα, εξαρτάται από τις τιμές των βαρών, που έχουν οι ακμές του γράφου G. Για τον λόγο αυτό, παρουσιάζουμε τον αλγόριθμο CXP2, ο οποίος στηρίζεται στον προηγούμενο αλγόριθμο, είναι επίσης 2-προσεγγιστικός, αλλά είναι πολυωνυμικός.

Σύμφωνα με τον αλγόριθμο, στρογγυλοποιούμε προς τα πάνω τις τιμές των βαρών των ακμών του γράφου G. Αν ονομάσουμε (G,w΄) τον νέο γράφο που προκύπτει, υπολογίζουμε την τιμή του W(G,w΄). Αυξάνουμε τις τιμές των βαρών των ακμών του γράφου (G,w΄), με τρόπο τέτοιο ώστε, κάθε κορυφή να έχει βάρος ίσο με W(G,w΄), και ονομάζουμε τον γράφο που προκύπτει G΄. Συνεπώς για τον γράφο G΄, θα ισχύει $\forall u \in U, v \in V: w(u) = w(v) = W(G^{'}) = W(G, w')$. Έπειτα, βρίσκουμε perfect matching M, στον γράφο G΄. Επειδή ο G΄ είναι κανονικός, ως προς τα βάρη των κορυφών του, θα υπάρχει perfect matching. Έστω w(M) ότι είναι η ελάχιστη τιμή των βαρών, των ακμών που μετέχουν στο M, δηλαδή $w(M) = \min\{w(e): e \in M\}$. Τότε, για κάθε ακμή e ∈ M, μειώνουμε το βάρος της, κατά w(M). Στον γράφο G΄, μετά την μείωση των κατάλληλων βαρών, αφαιρούμε τις ακμές εκείνες που έχουν μηδενικό βάρος. Ο γράφος εξακολουθεί να είναι κανονικός ως προς τα βάρη των κορυφών του, αφού ο αριθμός των ακμών του, έχει μειωθεί τουλάχιστον κατά ένα. Επαναλαμβάνουμε την παραπάνω διαδικασία, το πολύ |E| φορές, μέχρις ότου αφαιρεθούν όλες οι ακμές του γράφου G΄.

4.2.2.7 *Παράδειγμα Εφαρμογής του Αλγορίθμου CXP2*

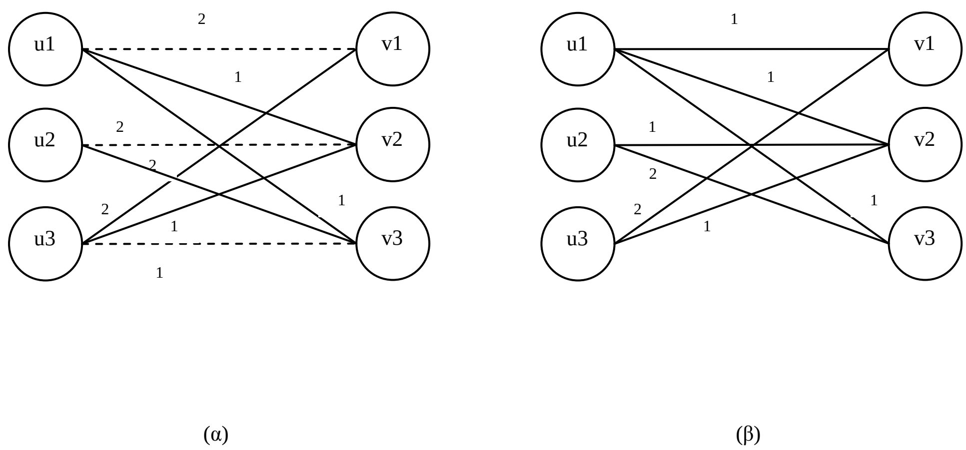
Έστω ότι καλούμαστε να επιλύσουμε το στιγμιότυπο, που έχει πίνακα δεδομένων τον $D = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$, χρησιμοποιώντας το αλγόριθμο CXP2. Δημιουργούμε τον γράφο G που αντιστοιχεί στα δεδομένα του προβλήματος. Για τον G παρατηρούμε πως W(G)=4 και Δ(G)=3, επομένως το κάτω φράγμα για την βέλτιστη λύση θα είναι 3+4=7 χρονικές μονάδες. Προσθέτουμε στον G ακμές, και βάρη στις ακμές του ώστε να προκύψει ο γράφος G΄, στον οποίο κάθε κορυφή έχει βάρος ίσο με W(G)=4.



Σχήμα 4.8 : (α) γράφος G, (β) παραγόμενος γράφος G΄

Ο πίνακας δεδομένων που αντιστοιχεί στον γράφο G΄ θα είναι ο $D' = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 2 \\ 2 & 1 & 1 \end{bmatrix}$.

Παρατηρούμε πως για τον πίνακα D πρέπει πάντα, το άθροισμα των τιμών των στοιχείων μίας οποιασδήποτε στήλης, να είναι ίσο με το άθροισμα των τιμών των στοιχείων μίας οποιασδήποτε γραμμής και ίσο με W(G). Στη συνέχεια, έστω ότι επιλέγουμε το perfect matching M_1={(u1,v1),(u2,v2),(u3,v3)}, με w(M_1)= min{w(u1,v1),w(u2,v2),w(u3,v3)}= min{2,2,1}=1. Επομένως, στον γράφο G΄ θα πρέπει να μειώσουμε τα κόστη των ακμών που ανήκουν στο M_1 κατά ένα και να αφαιρέσουμε την ακμή (u3,v3).



Σχήμα 4.9 : (α) matching M_1, (β) ο παραγόμενος γράφος G΄ που προκύπτει

Στο παραπάνω σχήμα, οι ακμές που ανήκουν στο M_1 εμφανίζονται με διακεκομμένη γραμμή. Ο γράφος G΄ που προκύπτει μετά την μείωση των βαρών και την αφαίρεση της ακμής, είναι επίσης κανονικός ως προς τον βαθμό των κορυφών του. Η διαφορά είναι ότι ο βαθμός αυτός έχει μειωθεί κατά $w(M_1)$, δηλαδή κατά ένα σε αυτήν την περίπτωση, και επομένως πλέον W(G΄)=3. Ο νέος πίνακας δεδομένων που προκύπτει είναι $D' = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 2 & 1 & 0 \end{bmatrix}$. Συνεχίζουμε την παραπάνω διαδικασία έως ότου έχουν σταλεί όλα τα δεδομένα και συνεπώς έχουν μηδενιστεί όλα τα στοιχεία του πίνακα D΄. Συνοπτικά, μπορούμε να παρουσιάσουμε όλη την διαδικασία για την επίλυση του προβλήματός μας, χρησιμοποιώντας την αναπαράσταση των πινάκων αποστολής, ως εξής : $\begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 2 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} +$ $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$. Όπως βλέπουμε, η αποστολή όλων των δεδομένων, σύμφωνα με τον αλγόριθμο CXP2, απαιτεί την δημιουργία τεσσάρων ομάδων αποστολής δεδομένων και επομένως θα απαιτεί 4+4=8 χρονικές μονάδες.

4.2.3 Ο αλγόριθμος A-PBS(α)

4.2.3.1 Εισαγωγή-Είσοδος του αλγορίθμου

Ο αλγόριθμος A-PBS(α) παρουσιάστηκε από τους F. Afrati, T. Aslanidis, E. Bambis και I. Milis το 2005. Είναι ο πρώτος προσεγγιστικός αλγόριθμος για το πρόβλημά μας, που έχει παράγοντα προσέγγισης μικρότερο από 2. Σαν είσοδο ο αλγόριθμος δέχεται τον διμερή γράφο G=$(V \cup U, E, w)$ που αντιστοιχεί στα δεδομένα του προβλήματος. Με w συμβολίζουμε το σύνολο των βαρών των ακμών του γράφου, που αντιπροσωπεύουν τους χρόνους αποστολής των δεδομένων. Ακόμη δέχεται έναν ακέραιο α, και την τιμή του χρόνου καθορισμού *d*. Τέλος, για την εφαρμογή του αλγορίθμου, θεωρούμε ότι επιτρέπεται ο διάσπαση των δεδομένων (preemption), και η τμηματική τους αποστολή. Δηλαδή ότι ένας πομπός μπορεί να στείλει ένα τμήμα ενός δεδομένου, το οποίο αντίστοιχα θα απαιτεί χρόνο, ένα ποσοστό του συνολικού χρόνου αποστολής του δεδομένου. Φυσικά, για να ολοκληρωθεί η αποστολή των δεδομένων, πρέπει όλα τα δεδομένα να έχουν σταλεί στο σύνολό τους.

4.2.3.2 Τα βήματα του αλγορίθμου

Τα βήματα που ακολουθούμε για την εκτέλεση του αλγόριθμου A-PBS(α) είναι τα εξής:

Αρχικά, στρογγυλοποιούμε προς τα πάνω τις τιμές των βαρών των ακμών του γράφου G, έτσι ώστε κάθε βάρος να είναι πολλαπλάσιο της τιμής του ακεραίου α. Ονομάζουμε τον νέο γράφο που προκύπτει μετά την ανάθεση των νέων τιμών στα βάρη, G΄=$(V \cup U, E, w')$. Δηλαδή για κάθε ακμή e του G΄ θα ισχύει: $w^{'}(e) = k \cdot w(e), k \in Z$.

Στη συνέχεια χωρίζουμε κάθε ακμή e του G΄ σε $w^{'}(e)/\alpha$ ακμές, όπου η κάθε μία έχει κόστος ίσο με α. Ονομάζουμε τον παραγόμενο γράφο G_α ,και υπολογίζουμε την τιμή του $W(G_\alpha)$

Βρίσκουμε στον γράφο G_α, ακριβώς $\frac{W(G_\alpha)}{\alpha}$ matchings , που αντιστοιχούν στις αποστολές που χρειάζεται να γίνουν, σύμφωνα με τον αλγόριθμο, ώστε να έχουμε στείλει όλα τα δεδομένα. Κάθε matching M που επιλέγουμε, έχει κόστος αποστολής W(M)=α. Επομένως ο συνολικός χρόνος αποστολής που προκύπτει από την εφαρμογή του αλγορίθμου A-PBS(α), θα είναι ίσος με $\frac{W(G_\alpha)}{\alpha} \cdot \alpha + \frac{W(G_\alpha)}{\alpha} \cdot d$.

4.2.3.3 Παράγοντας προσέγγισης

Όπως αναφέραμε παραπάνω, θα έχουμε ακριβώς $\frac{W(G_\alpha)}{\alpha}$ αποστολές, η κάθε μία κόστους α. Έτσι το συνολικό κόστος θα είναι:

$$\frac{W(G_\alpha)}{\alpha} \cdot \alpha + \frac{W(G_\alpha)}{\alpha} \cdot d \quad (1)$$

Κατά την δημιουργία του γράφου G΄, οι τιμές των βαρών στις ακμές του G , έχουν αυξηθεί το πολύ κατά α-1, προκειμένου να είναι πολλαπλάσιες της τιμής του α. Επομένως η μέγιστη τιμή των βαρών των κορυφών του γράφου G_α ($W(G_\alpha)$), θα είναι αυξημένη σε σχέση με την αντίστοιχη τιμή του γράφου G, το πολύ κατά $(\alpha - 1) \cdot \Delta(G)$. Συνεπώς θα ισχύει $W(G_\alpha) \leq W(G) + (a-1) \cdot \Delta(G)$, και χρησιμοποιώντας την σχέση (1), συμπεραίνουμε πως ο συνολικός χρόνος αποστολής των δεδομένων φράσσεται από την παράσταση :

$$W(G) + (a-1) \cdot \Delta(G) + \frac{W(G)+(a-1)\cdot\Delta(G)}{\alpha} \cdot d = \frac{d+a}{a} \cdot \big(W(G) + (a-1) \cdot \Delta(G)\big) \quad (2)$$

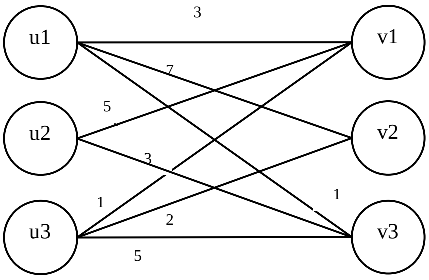
Αν δεχθούμε ότι ισχύει $a = d + 1$, τότε από την σχέση (2), προκύπτει ότι το συνολικό κόστος της λύσης που παίρνουμε από την εφαρμογή του αλγορίθμου A-PBS(d+1) είναι το πολύ ίσο με:

$$\frac{2 \cdot d + 1}{d + 1} \cdot \big(W(G) + d \cdot \Delta(G)\big) \quad (3)$$

Δεδομένου πως, το κάτω φράγμα για το κόστος της βέλτιστης λύσης είναι $W(G) + d \cdot \Delta(G)$, από την σχέση (3) προκύπτει ότι ο παράγοντας προσέγγισης για τον αλγόριθμο A-PBS(d+1) είναι ίσος με $\frac{2d+1}{d+1} = 2 - \frac{1}{d+1}$ δηλαδή μικρότερος από 2.

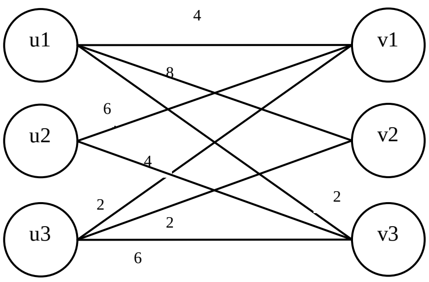
4.2.3.4 *Παράδειγμα εφαρμογής*

Υποθέτουμε πως καλούμαστε να επιλύσουμε, χρησιμοποιώντας τον αλγόριθμο A-PBS(α), το στιγμιότυπο ενός προβλήματος, το οποίο έχει πίνακα δεδομένων τον $A = \begin{bmatrix} 3 & 7 & 1 \\ 5 & 0 & 3 \\ 1 & 2 & 5 \end{bmatrix}$ και ο χρόνος καθορισμού είναι d=1. Για την επίλυσή του, θα εφαρμόσουμε τα βήματα του αλγόριθμου A-PBS(2). Ο γράφος G που αντιστοιχεί στα δεδομένα που έχουμε, φαίνεται στο σχήμα που ακολουθεί.

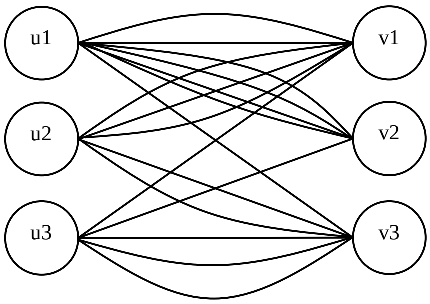


Σχήμα 4.10: γράφος δεδομένων G.

Στη συνέχεια αυξάνουμε τις τιμές των βαρών των ακμών, όπου αυτό απαιτείται, ώστε όλες οι τιμές να είναι πολλαπλάσιες της τιμής του α=2. Στον γράφο που προκύπτει (G΄) χωρίζουμε κάθε ακμή e σε w(e)/2 ακμές βάρους 2, δημιουργώντας έτσι τον γράφο $G_α$.



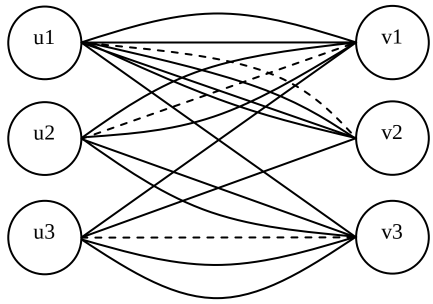
(α)



(β)

Σχήμα 4.11: παραγόμενοι γράφοι (α) G΄, (β) $G_α$.

Ο πίνακας δεομένων που αντιστοιχεί στους γράφους G΄ και $G_α$ είναι ο $A' = \begin{bmatrix} 4 & 8 & 2 \\ 6 & 0 & 4 \\ 2 & 2 & 6 \end{bmatrix}$.
Υπολογίζουμε την τιμή του $W(G_α)$, όπου στη συγκεκριμένη περίπτωση είναι 4+8+2=14 και επομένως θα βρούμε $W(G_α)/α$ = 14/2= 7 matchings. Κάθε ένα από τα matchings θα αντιστοιχεί σε μία ομάδα αποστολής, και θα απαιτεί 2 χρονικές μονάδες για την αποστολή αυτή. Έστω ότι επιλέγουμε από τον γράφο $G_α$ το matching M_1={(u1,v2),(u2,v1),(u3,v3)}, το οποίο μπορούμε να δούμε στο ακόλουθο σχήμα.



Σχήμα 4.12: matching M_1.

Ο πίνακας αποστολής που αντιστοιχεί στο matching M_1 θα είναι ο $\begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix}$. Συνεχίζουμε την παραπάνω διαδικασία μέχρις ότου να αποσταλούν όλα τα δεδομένα που έχουμε. Ολοκληρωμένη η διαδικασία, παρουσιάζεται συνοπτικά χρησιμοποιώντας τον συμβολισμό των πινάκων αποστολής ως εξής:

$A' = \begin{bmatrix} 4 & 8 & 2 \\ 6 & 0 & 4 \\ 2 & 2 & 6 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} +$
$\begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. Η λύση που βρίσκουμε, έχει κόστος $7 \cdot 1 + 7 \cdot 2 = 7 + 14 = 21$ χρονικές μονάδες, ενώ η βέλτιστη λύση έχει κόστος $W(G)+d\cdot\Delta(G)=11+3\cdot1=14$ χρονικές μονάδες.

4.2.3.5 Σχόλια

Ο αλγόριθμος που παρουσιάσαμε, είναι ψεύδο-πολυωνυμικός καθώς το πλήθος των επαναλήψεων που θα χρειαστούν, για να επιστρέψει την λύση, εξαρτάται από τις τιμές των βαρών στον γράφο, δηλαδή από τους χρόνους αποστολής των δεδομένων. Για τον λόγο αυτό, δημιουργούμε μία παραλλαγή του αλγορίθμου A-PBS(α), η οποία μας επιστρέφει την λύση σε πολυωνυμικό χρόνο. Σύμφωνα με αυτήν, δεν χωρίζουμε τις ακμές του γράφου G΄, αλλά βρίσκουμε matchings σε αυτόν το γράφο. Σε κάθε matching M, βρίσκουμε το ελάχιστο βάρος των ακμών που ανήκουν στο M, και αφαιρούμε την τιμή αυτή από τα βάρη των ακμών που ανήκουν στο M. Επαναλαμβάνουμε, έως ότου έχουν μηδενιστεί όλες οι ακμές του γράφου G΄, δηλαδή έχουν σταλεί όλα τα δεδομένα.

Στο παραπάνω παράδειγμα, αν εφαρμόσουμε τον πολυωνυμικό αλγόριθμο A-PBS(2) προκύπτει η εξής λύση: $A' = \begin{bmatrix} 4 & 8 & 2 \\ 6 & 0 & 4 \\ 2 & 2 & 6 \end{bmatrix} = \begin{bmatrix} 0 & 6 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 6 \end{bmatrix} + \begin{bmatrix} 4 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix}$ η οποία έχει συνολικό κόστος $4 \cdot 1 + (6 + 4 + 2 + 2) = 4 + 14 = 18$ χρονικές μονάδες.

5

Πειράματα Σύγκρισης Αλγορίθμων

Στο κεφάλαιο αυτό, αφού έχουμε υλοποιήσει τους τρεις προσεγγιστικούς αλγορίθμους που έχουμε δει, θα τρέξουμε μία σειρά πειραμάτων, προκειμένου να εξάγουμε κάποια συμπεράσματα για την αποδοτικότητά τους. Η υλοποίηση των αλγορίθμων, έχει γίνει με τέτοιο τρόπο, ώστε η είσοδος και στους τρεις αλγορίθμους να είναι κοινή. Συγκεκριμένα, η είσοδος είναι ένας πίνακας από ακέραιες, μη-αρνητικές τιμές, που αντιστοιχεί στον πίνακα δεδομένων του προβλήματος, και η τιμή της παραμέτρου d (set-up delay). Στον αλγόριθμο G-W, έχουμε θεωρήσει, πως ο αριθμός των transponders k , είναι μεγαλύτερος από την διάσταση του πίνακα δεδομένων, και έτσι η τιμή του δεν επηρεάζει την εφαρμογή του αλγορίθμου και συνεπώς δεν χρειάζεται να συμπεριλάβουμε την τιμή του, στις εισόδους του.

5.1 Πείραμα Σύγκρισης

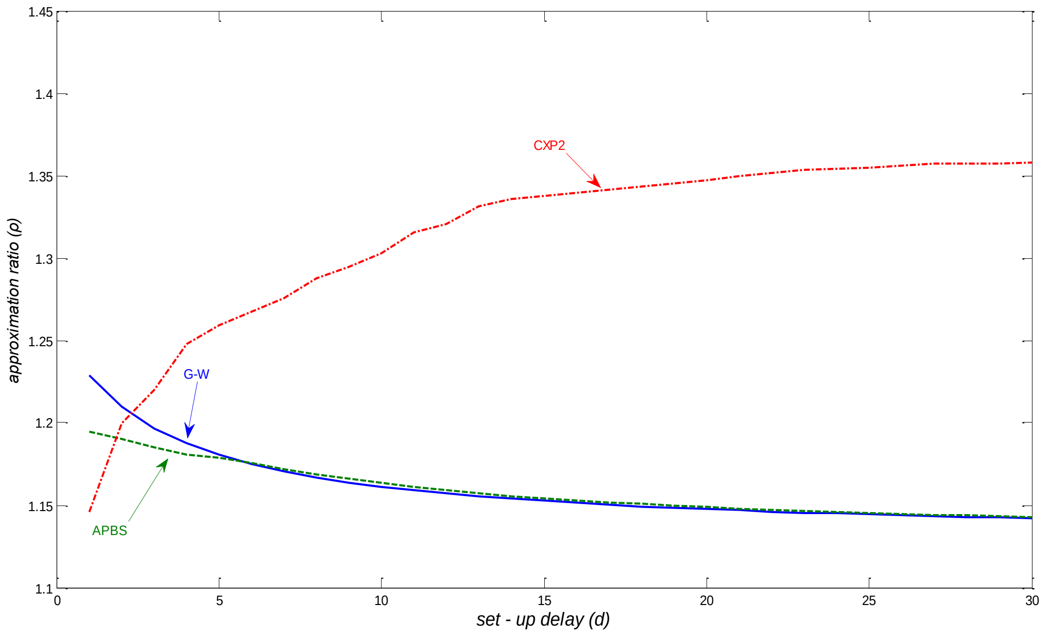
Αρχικά, δημιουργήσαμε μία δομή από 150 πίνακες, οι τιμές των οποίων είναι ακέραιες, μη-αρνητικές , προκειμένου να τους χρησιμοποιήσουμε ως πίνακες δεδομένων στην είσοδο των αλγορίθμων μας. Οι τιμές των στοιχείων κάθε πίνακα είναι τυχαίες και ομοιόμορφα κατανεμημένες στο διάστημα [0-10]. Οι πίνακες έχουν διάσταση 3x3, 4x4 και 5x5, συγκεκριμένα η δομή αποτελείται από 50 πίνακες της κάθε διάστασης από τις παραπάνω. Θεωρούμε, πως ο χρόνος καθορισμού (set-up delay) είναι ίσος με μία χρονική μονάδα.

Στη συνέχεια, για κάθε πίνακα, υπολογίζουμε την βέλτιστη λύση για το πρόβλημά μας, δηλαδή τον ελάχιστο χρόνο που απαιτείται για την αποστολή όλων των δεδομένων. Ο υπολογισμός αυτός γίνεται μέσω εξαντλητικής αναζήτησης στο χώρο λύσεων, χρησιμοποιώντας την τεχνική του κλαδέματος (pruning). Έπειτα, τρέχουμε τον κάθε αλγόριθμο 150 φορές, κάθε φορά με είσοδο έναν διαφορετικό πίνακα δεδομένων από την δομή που δημιουργήσαμε. Στο τέλος κάθε εκτέλεσης του αλγορίθμου για κάθε πίνακα, υπολογίζουμε τον παράγοντα προσέγγισης (approximation ratio), διαιρώντας τις χρονικές μονάδες που μας επέστρεψε η εφαρμογή του αλγορίθμου, προς την αντίστοιχη τιμή της βέλτιστης λύσης που έχουμε υπολογίσει. Αφού υπολογίσουμε την τιμή του παράγοντα προσέγγισης και για τους 150 πίνακες, υπολογίζουμε και αποθηκεύουμε την μέση τιμή της παραμέτρου αυτής, προκειμένου να έχουμε μία γενική εικόνα για την προσεγγισιμότητα του κάθε αλγορίθμου.

Αφού ολοκληρώσουμε την παραπάνω διαδικασία και για τους τρεις αλγορίθμους, αυξάνουμε κατά μία χρονική μονάδα την τιμή της καθυστέρησης καθορισμού (d), και υπολογίζουμε εκ νέου την μέση τιμή του παράγοντα προσέγγισης για κάθε αλγόριθμο. Επαναλαμβάνουμε τα παραπάνω βήματα, έως ότου ο χρόνος καθορισμού λάβει και την τιμή των 30 χρονικών μονάδων. Με την διαδικασία αυτή, έχουμε υπολογίσει τον παράγοντα προσέγγισης, για τιμές του χρόνου καθορισμού μικρότερες, του ίδιου μεγέθους έως και πολύ μεγαλύτερες από τον μέσο χρόνο αποστολής ενός δεδομένου.

5.2 Γραφήματα – Αποτελέσματα

Χρησιμοποιώντας την μέση τιμή του παράγοντα προσέγγισης για κάθε αλγόριθμο, για κάθε χρόνο καθορισμού (d), δημιουργούμε του βασικό γράφημα σύγκρισης των τριών αλγορίθμων. Στο γράφημα που ακολουθεί, στον οριζόντιο άξονα βλέπουμε την τιμή του χρόνου καθορισμού σε χρονικές μονάδες, ενώ στον κατακόρυφο άξονα την μέση τιμή του παράγοντα προσέγγισης όπως προκύπτει από την εφαρμογή των αλγορίθμων στους πίνακες δεδομένων. Οι τρείς καμπύλες που βλέπουμε, αντιστοιχούν στην προσεγγισιμότητα των τριών αλγορίθμων σε σχέση με τον χρόνο καθορισμού.

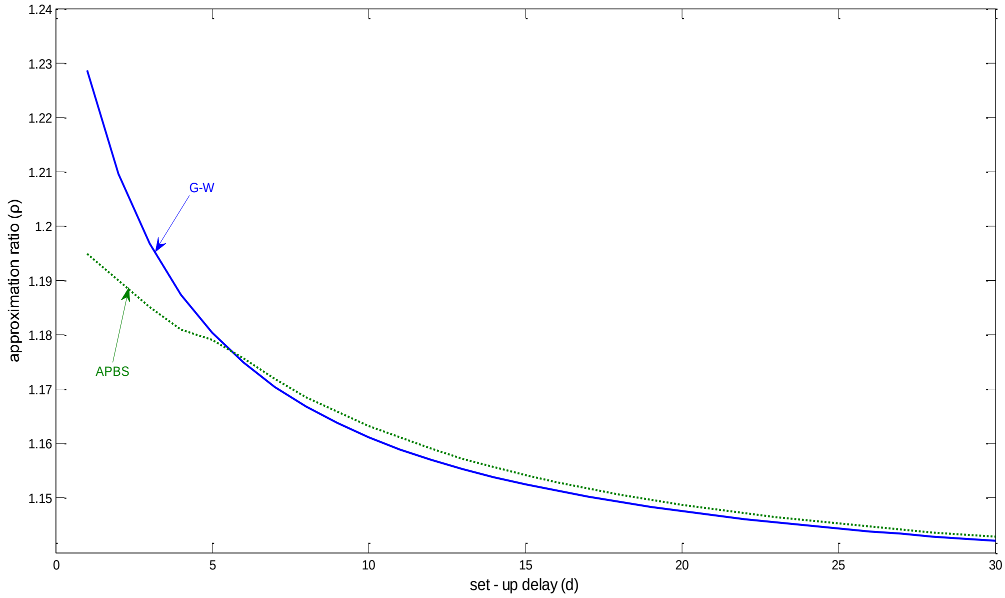


Σχήμα 5.1 : Γράφημα σύγκρισης παράγοντα προσέγγισης των τριών αλγορίθμων, σε σχέση με τον χρόνο καθορισμού.

Παρατηρώντας το γράφημα αυτό, βλέπουμε πως ο αλγόριθμος CXP δίνει λύσεις πολύ κοντά στην βέλτιστη όταν η τιμή της καθυστέρησης καθορισμού είναι πολύ μικρή, σε σχέση με τον μέσο χρόνο αποστολής ενός δεδομένου. Στη συνέχεια όμως, και καθώς η τιμή του d αυξάνεται, οι λύσεις που παίρνουμε από την εφαρμογή του CXP απομακρύνονται από την βέλτιστη λύση. Το γεγονός αυτό είναι λογικό, καθώς ο αλγόριθμος CXP έχει σχεδιαστεί για τιμή του d ίση με την μονάδα. Όταν η τιμή αυτή είναι μεγαλύτερη, έχουμε χρονικές απώλειες κατά την κανονικοποίηση των τιμών των στοιχείων του πίνακα δεδομένων, οι οποίες αυξάνονται με την αύξηση του d.

Για τον αλγόριθμο G-W, παρατηρούμε πως καθώς η τιμή του d αυξάνεται τόσο παίρνουμε λύσεις πιο κοντά στην βέλτιστη. Κάτι τέτοιο είναι αναμενόμενο, καθώς ο αλγόριθμος αυτός επιτυγχάνει να ελαχιστοποιήσει το πλήθος των αποστολών που γίνονται, προκειμένου να σταλούν όλα τα δεδομένα. Έτσι, για μεγάλες τιμές του d, σε σχέση με τους χρόνους αποστολής των δεδομένων, επιστρέφει λύσεις πολύ κοντά στη βέλτιστη.

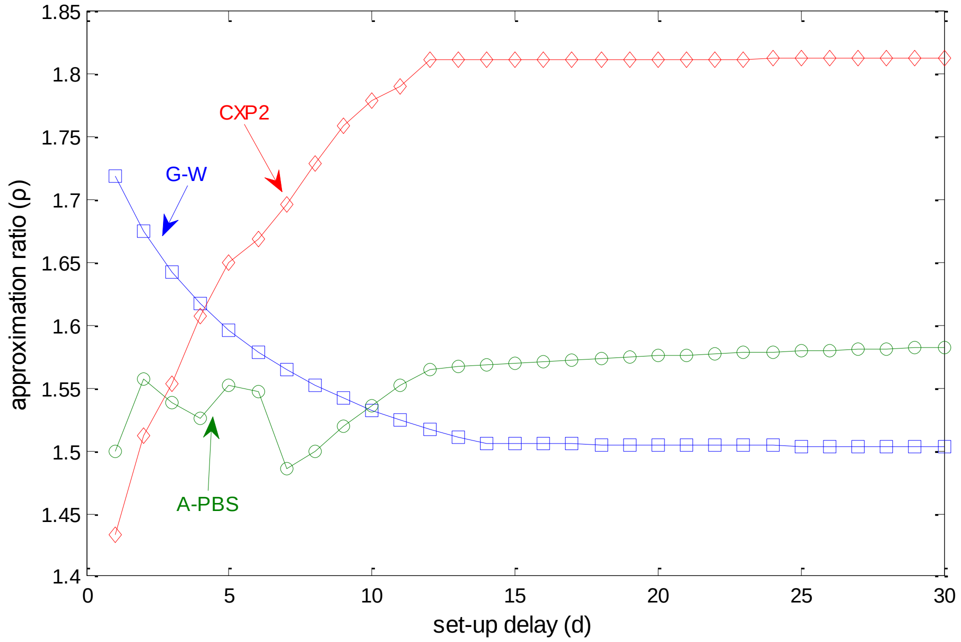
Ο αλγόριθμος A-PBS, παρατηρούμε ότι έχει τις μικρότερες διακυμάνσεις, ως προς τον παράγοντα προσέγγισης των λύσεων που επιστρέφει. Έχει καλύτερη απόδοση από τον αλγόριθμο CXP, όταν η τιμή του d είναι μεγαλύτερη της μονάδας, δηλαδή, για το συγκεκριμένο πείραμα, όταν είναι μεγαλύτερη από το 1/10 του μέγιστου χρόνου αποστολής κάποιου δεδομένου. Για να μπορέσουμε να δούμε με μεγαλύτερη λεπτομέρεια, την καμπύλη του αλγορίθμου A-PBS σε σχέση με τον G-W, παρουσιάζουμε το ακόλουθο γράφημα.



Σχήμα 5.2 : γράφημα σύγκρισης αλγορίθμων A-PBS και G-W

Όπως παρατηρούμε από το παραπάνω γράφημα, η εφαρμογή του αλγόριθμου A-PBS ενδείκνυται για στιγμιότυπα του προβλήματος, όπου η τιμή του χρόνου καθορισμού (d) είναι ίση ή μικρότερη από τον χρόνο αποστολής ενός δεδομένου. Για μεγαλύτερες τιμές του χρόνου d, αν και ο αλγόριθμος G-W υπερέχει, ο A-PBS εξακολουθεί να επιστρέφει λύσεις, που είναι πολύ κοντά στην βέλτιστη.

Στη συνέχεια, επαναλαμβάνουμε το παραπάνω πείραμα, με την διαφοροποίηση ότι για κάθε αλγόριθμο, αποθηκεύουμε την μέγιστη τιμή του παράγοντα προσέγγισης. Έτσι, βρίσκουμε πόσο απέχει η λύση που μας επιστρέφει ο κάθε αλγόριθμος, από την βέλτιστη, στην χειρότερη περίπτωση (worst case).Επαναλαμβάνουμε την παραπάνω διαδικασία, για τιμές του χρόνου καθορισμού από 1 έως 30, και από τις τιμές του παράγοντα προσέγγισης που προκύπτουν για κάθε αλγόριθμο, σχεδιάζουμε το νέο διάγραμμα σύγκρισης. Στο νέο αυτό διάγραμμα, το οποίο είναι περισσότερο συμβατό με τους θεωρητικούς ορισμούς του παράγοντα προσέγγισης και των προσεγγιστικών αλγορίθμων, και πάλι η τιμές του χρόνου καθορισμού εμφανίζονται στον οριζόντιο άξονα, ενώ οι τιμές του παράγοντα προσέγγισης και για τους τρείς αλγορίθμους στον κατακόρυφο.



Σχήμα 5.3 : γράφημα σύγκρισης αλγορίθμων ως προς την χειρότερη περίπτωση

Σύμφωνα με το παραπάνω γράφημα, βλέπουμε ότι και αναφορικά στην «χειρότερη» λύση που επιστρέφουν οι τρείς αλγόριθμοι, προκύπτουν παρόμοιες καμπύλες με το προηγούμενο γράφημα, και συνεπώς ισχύουν οι παρατηρήσεις που έχουμε επισημάνει.

5.3 Συμπεράσματα Πειράματος

Από το πείραμα που πραγματοποιήσαμε, μπορούμε να καταλήξουμε σε κάποια συμπεράσματα. Αρχικά, μπορούμε να πούμε πως και οι τρεις αλγόριθμοι επιστρέφουν λύσεις οι οποίες είναι κοντά στην βέλτιστη. Βεβαίως, το πόσο κοντά στη βέλτιστη πρέπει να είναι οι λύσεις αυτές, για να χαρακτηρίσουμε ως αποδοτικό έναν αλγόριθμο, είναι άμεσα συνδεδεμένο με την εφαρμογή του προβλήματος την οποία καλούμαστε να επιλύσουμε. Αν στην εφαρμογή αυτή, μπορούμε να θεωρήσουμε αμελητέο τον χρόνο καθορισμού σε σχέση με τον χρόνο αποστολής ενός δεδομένου, τότε θα πρέπει να επιλέξουμε τον αλγόριθμο CXP ή τον A-PBS για την επίλυση του. Αντίθετα, αν ο χρόνος καθορισμού είναι μεγαλύτερος από τους χρόνους αποστολής των δεδομένων, τότε προφανώς θα πρέπει να επιλέξουμε τον αλγόριθμο G-W. Στην περίπτωση όπου καλούμαστε να εφαρμόσουμε έναν από τους τρείς αλγορίθμους, για μία πληθώρα διαφορετικών σχέσεων χρόνων αποστολής- καθορισμού, τότε η καταλληλότερη επιλογή θα είναι αυτή του A-PBS, καθιστώντας τον, σαν έναν γενικό αλγόριθμο επίλυσης του προβλήματός μας.

6

Επίλογος

Στο σημείο αυτό θα συνοψίσουμε το περιεχόμενο της διπλωματικής εργασίας. Θα παρουσιάσουμε κάποια συγκεντρωτικά συμπεράσματα, στα οποία καταλήξαμε από την μελέτη και τα πειράματα της εργασίας, και θα προτείνουμε μελλοντικές μελέτες- εργασίες, οι οποίες μπορούν να βασιστούν σε αυτήν.

6.1 Σύνοψη και συμπεράσματα

Στην παρούσα εργασία παρουσιάσαμε τρεις προσεγγιστικούς αλγορίθμους, για την επίλυση του προβλήματος της χρονοδρομολόγησης αποστολής δεδομένων σε δίκτυα. Το πρόβλημα αυτό, είναι ένα ρεαλιστικό πρόβλημα για το οποίο, όπως φαίνεται και από τις εφαρμογές του που παρουσιάσαμε, η εύρεση αλγορίθμων που μας επιστρέφουν λύσεις, κοντά στην βέλτιστη και σε σύντομο χρονικό διάστημα είναι μεγάλης τεχνολογικής και επιστημονικής σημασίας.

Αφού υλοποιήσαμε τους τρεις αλγορίθμους και τρέξαμε κάποια πειράματα σε αυτούς, καταλήξαμε σε μία σειρά συμπερασμάτων, χρήσιμα για την επιλογή του κατάλληλου αλγορίθμου για την επίλυση του προβλήματος μας. Συγκεκριμένα, διαπιστώσαμε πως ο καθοριστικότερος παράγοντας για την επιλογή αυτή, είναι η σχέση ανάμεσα στον χρόνο καθορισμού (set-up delay-d) και του χρόνου αποστολής των δεδομένων. Συγκεντρωτικά, μπορούμε να πούμε πως, σε εφαρμογές όπου ο χρόνος καθορισμού είναι μεγαλύτερος από τον χρόνο αποστολής των δεδομένων, ο καταλληλότερος αλγόριθμος είναι ο G-W. Σε εφαρμογές όπου ο χρόνος καθορισμού είναι πολύ μικρότερος από τον χρόνο αποστολής και συνεπώς μπορεί να αγνοηθεί, η καλύτερη επιλογή είναι αυτή του αλγορίθμου CXP ή του A-PBS, ενώ για την περίπτωση όπου ο χρόνος καθορισμού είναι περίπου ίσος με τον χρόνο αποστολής, ή όπου θέλουμε να επιλέξουμε έναν αλγόριθμο για την επίλυση στιγμιότυπων με διαφορετικές σχέσεις χρόνου καθορισμού-αποστολής, ο καταλληλότερος αλγόριθμος είναι ο A-PBS.

6.2 Μελλοντικές επεκτάσεις

Για την εξαγωγή ασφαλέστερων και πιο συγκεκριμένων συμπερασμάτων, σκόπιμη θεωρούμε την διεξαγωγή πειραμάτων, στα οποία τα δεδομένα εισόδου των αλγορίθμων να είναι τιμές από πραγματικές εφαρμογές του προβλήματος. Βασιζόμενοι στο γεγονός ότι, η σχέση ανάμεσα στον χρόνο καθορισμού και τον χρόνο αποστολής των δεδομένων, επηρεάζει σημαντικά την προσέγγιση της βέλτιστης λύσης, ένας συνδυαστικός αλγόριθμος, που θα στηρίζεται στο παραπάνω κριτήριο, ενδεχομένως να έχει καλύτερα πειραματικά αποτελέσματα από τους τρεις αλγορίθμους που εξετάσαμε. Τέλος, λόγω της λειτουργίας των εφαρμογών του προβλήματος, η δημιουργία αλγορίθμων επίλυσης του προβλήματος, που θα επιτρέπουν την αλλαγή των τιμών των δεδομένων εισόδου τους, κατά την διάρκεια της επεξεργασίας τους (On-line αλγόριθμοι), θεωρείται απαραίτητη.

7

Βιβλιογραφία

[AAB+05] F.Afrati, T.Aslanidis, E.Bampis, I.Milis. Scheduling in switching networks with set-up delay. Journal of Combinatorial Optimization, vol. 9, no. 1, pp. 49-57, 2005.

[CXP01] P.Crescenzi, D.Xiaotie, C.H.Papadimitriou. On approximating a scheduling problem, Journal of Combinatorial Optimization, vol. 5, p. 287-297, 2001.

[CJP+06] J.Cohen, E.Jeannot, N.Padoy, F.Wagner. Messages scheduling for parallel data redistribution between clusters. IEEE, Trans. On Parallel and Distributed Systems, vol. 17, no. 10, p.1163-1175, 2006.

[CLR+01] T.H.Cormen, C.E.Lieserson, R.LRivest, C.Stein. Introduction to Algorithms, second edition. MIT Press- McGraw Hill, 2001.

[DJT94] D.Durand, R.Jain, D.Tseytlin. Distributed scheduling algorithms to improve the performance of parallel data transfers. ACM, Computer Architecture News, vol. 22, is. 4, p. 35-40, 1994.

[DPV06] S.Dasgupta, C.H.Papadimitriou, U.V.Vazirani. Algorithms, McGraw-Hill, 2006.

[GW85] I.S.Gopal, C.K.Wong. Minimizing the number of switchings in an SS/TDMA system. IEEE, Trans. On Communications, vol. 33, p. 497-501, 1985.

[JWB97] R.Jain, J.Werth, J.C.Browne. A note on scheduling problems arising in satellite communications. Journal of the Operational Research Society, vol. 48, no. 1, p. 100-103, 1997.

[PHI97] I.R.Philp. Scheduling real-time messages in packet- switched networks, Ph.D thesis, Comp. Science, Univ. of Illinois at Urbana Champaign, 1997.

[SWW+02] K.M.Sivalingam, J.Wang, X.Wu, M.Mishra. An internal-based scheduling algorithm for optical WDM star networks. Journal of Photonic Network Communications, vol. 4, no. 1, p.73-87, 2002M.

[Vaz01] V.Vazirani. Approximation Algorithms. Springer Verlag Berlin Heidelberg, New York, 2001.

[Wes01] D.B.West. Introduction to Graph Theory, second edition. Prentice Hall, 2001.