ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΜΕ ΘΕΜΑ:

ΜΕΛΕΤΗ ΔΙΚΤΥΩΝ

ΜΕΘΟΔΟΙ ΚΑΙ ΕΦΑΡΜΟΓΕΣ

Του προπτυχιακού φοιτητή:

Δημήτρη Μ. Κατσίποδα

Α.Μ. 09111104

Επιβλέπων καθηγητής: **Ιωάννης Κολέτσος,**

Επίκουρος Καθηγητής Ε.Μ.Π.

Τριμελής Επιτροπή:

1) Κοκκίνης Βασίλειος, Επίκουρος Καθηγητής Ε.Μ.Π.

2) Κολέτσος Ιωάννης, Επίκουρος Καθηγητής Ε.Μ.Π.

3) Στεφανέας Πέτρος , Επίκουρος Καθηγητής Ε.Μ.Π.

ΑΘΗΝΑ, ΣΕΠΤΕΜΒΡΙΟΣ 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΜΕ ΘΕΜΑ:

ΜΕΛΕΤΗ ΔΙΚΤΥΩΝ

ΜΕΘΟΔΟΙ ΚΑΙ ΕΦΑΡΜΟΓΕΣ

Του προπτυχιακού φοιτητή:

Δημήτρη Μ. Κατσίποδα

Α.Μ. 09111104

Επιβλέπων καθηγητής:

Ιωάννης Κολέτσος,

Επίκουρος Καθηγητής ΕΜΠ

ΑΘΗΝΑ, ΣΕΠΤΕΜΒΡΙΟΣ 2017

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας όπου σηματοδοτεί και το τέλος των προπτυχιακών μου σπουδών στη Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών του Ε.Μ.Π. , νιώθω την ανάγκη να εκφράσω γραπτώς τις ευχαριστίες μου στα άτομα, η συμβολή και η καθοδήγηση των οποίων υπήρξε καθοριστική για την εκπόνησή της.

Κατ’ αρχάς, θα ήθελα να ευχαριστήσω τον κ. Κολέτσο Ιωάννη, Επίκουρο Καθηγητή του Ε.Μ.Π. για τη πολύτιμη και συνεχή καθοδήγηση του καθ’ όλη τη διάρκεια εκπόνησης της διπλωματικής εργασίας.

Τις ευχαριστίες μου εκφράζω στον Επίκουρο Καθηγητή κ. Κοκκίνη Βασίλειο και στον Επίκουρο Καθηγητή κ. Στεφανέα Πέτρο που δέχτηκαν να είναι μέλη της τριμελής επιτροπής αξιολόγησης της εργασίας.

Τέλος, θέλω να ευχαριστήσω τους γονείς μου Μιχάλη και Κωνσταντίνα, την αδερφή μου Έλενα, καθώς και τους φίλους μου που με υπομονή και κουράγιο πρόσφεραν την απαραίτητη ηθική συμπαράσταση για την ολοκλήρωση της διπλωματικής μου εργασίας.

Περίληψη

Η παρούσα διπλωματική εργασία είναι μια παρουσίαση και ανάλυση των προβλημάτων Δικτύων ροής , κλάδου της Επιχειρησιακής Έρευνας και των μεθόδων επίλυσής τους.

Αρχικά, στο Κεφάλαιο 1, γίνεται μία εισαγωγή στην επιστήμη της Επιχειρησιακής Έρευνας, παρουσιάζοντας τις απαρχές της, αλλά και την σημαντικότητά της στην σύγχρονη κοινωνία. Έπειτα, στο Κεφάλαιο 2 γίνεται η εισαγωγή στα Δίκτυα Ροής, αναφέροντας χρήσιμους ορισμούς και συμβολισμούς. Στη συνέχεια, ακολουθούν οι τέσσερις γνωστοί μέθοδοι βελτιστοποίησης. Στο Κεφάλαιο 3 αναλύεται το πρόβλημα της Ελάχιστης Διαδρομής (Shortest-Path Problem) αναλύοντας τον αλγόριθμο Dijkstra και τον αλγόριθμο Floyd. Στο 4^{o} Κεφάλαιο, γίνεται η αναφορά στα Προβλήματα Μεγίστης Ροής (Max Flow Problem) και αναλύεται ο αλγόριθμος Ford-Fulkerson(ή μέθοδος μονοπατιού επαύξησης) και το θεώρημα μέγιστης ροής και ελάχιστης τομής και δίνονται 2 παραδείγματα. Στο 5^{o} Κεφάλαιο, παρουσιάζονται τα Ελάχιστα Παραγόμενα Δέντρα (Minimum Spanning Trees) και αναφέρεται ο Αλγόριθμος του Prim και ο Αλγόριθμος του Kruskal καθώς και δύο παραδείγματα για τον καθένα αλγόριθμο. Στο 6^{o} Κεφάλαιο, γίνεται η ανάλυση ενός εξίσου σημαντικού προβλήματος, του Προβλήματος Ροής Ελαχίστου Κόστους (Minimum Cost Flow Problem) και παρουσιάζεται ένα παράδειγμα. Τέλος στο 7^{o} Κεφάλαιο παρουσιάζουμε μια ανάλυση περίπτωσης (case study) για μια επιχείρηση και λύνουμε ένα πρόβλημα γραμμικού προγραμματισμού.

Abstract

This dissertation is both a presentation and an analysis of network flow problems, a branch of operational research, as well as their resolution methods.

Chapter 1 is actually an introduction to the science of Operational Research presenting its starting point historically and their salience in contemporary society. Subsequently, in chapter 2, we introduce the reader to flow networks mentioning useful definitions and explaining the main symbols we use, while we also present four widely-known optimization methods. In chapter 3, the Shortest- Path problem is analyzed presenting both Dijkstra and Floyd algorithms. In chapter 4, we present Max Flow problem where Ford- Fulkerson algorithm (or enhancement- path method) is analyzed as well as max-flow and minimum incision theorems are cited giving two examples. In Chapter 5, Minimum Spanning Trees are presenting, Prim's and Kruskal's algorithm and two examples for each. Chapter 6 includes the analysis of an equally significant problem, that of Minimum Cost Flow where an example also follows. Finally, this dissertation concludes with Chapter 7 where a case-study is presented solving a linear programming problem.

Πίνακας περιεχομένων

ΚΕΦΑΛΑΙΟ 1

Επιχειρησιακή Έρευνα

1.1 Εισαγωγή στην Επιχειρησιακή Έρευνα

Η Επιχειρησιακή Έρευνα ή διαφορετικά η Ποσοτική Ανάλυση για τη Λήψη Διοικητικών Αποφάσεων είναι μια αναλυτική μέθοδος επίλυσης προβλημάτων και λήψης αποφάσεων που χρησιμοποιείται στη διοίκηση οργανισμών, μέσω ενός συνόλου τεχνικών που βασίζονται στην επιστημονική προσέγγιση. Συγκεκριμένα, αναφερόμαστε σε τεχνικές που κάνουν χρήση μαθηματικών μοντέλων , ώστε να περιγράψουν τη λειτουργία ενός συστήματος και τα οποία μπορούν να βοηθήσουν στη βελτίωση της λειτουργίας του. Θα πρέπει να υπογραμμίσουμε πως στην περίπτωση της Επιχειρησιακής Έρευνας, δεν πρόκειται απλώς για μια συλλογή μαθηματικών τεχνικών που εφαρμόζεται στο πεδίο της διοίκησης, αλλά για τον ακρογωνιαίο λίθο της συστημικής προσέγγισης για την επίλυση των προβλημάτων που αντιμετωπίζουν οι επιχειρήσεις κι οι οργανισμοί. Πρόκειται, λοιπόν, για την εφαρμογή μια συγκεκριμένης φιλοσοφίας κατά την αντιμετώπιση τους, που ενσωματώνει όλους εκείνους τους παράγοντες που θεωρούνται σημαντικοί κι επηρεάζουν τη λήψη της βέλτιστης απόφασης (reference). Έχουν δοθεί και άλλοι ορισμοί που θα παρουσιαστούν στη συνέχεια του κεφαλαίου.

Το όνομά της προέρχεται από τον αγγλικό όρο Operational research (ή Operations Research στην Αμερική). Τα τελευταία χρόνια χρησιμοποιούνται και οι όροι Διοικητική Επιστήμη (Management Science), Επιστήμη Αποφάσεων (Decision Science) και Ανάλυση Συστημάτων (Systems Analysis) .

1.2 Ιστορική αναδρομή

Από την έλευση της βιομηχανικής επανάστασης έως σήμερα παρατηρείται αξιοσημείωτη ανάπτυξη στο χώρο των επιχειρήσεων: τα μικρά μαγαζιά (των τεχνητών) που κυριαρχούσαν ως επιχειρησιακή μορφή πριν τη βιομηχανική επανάσταση έχουν μετεξελιχθεί στις σημερινές πολυεθνικές με κύκλο δισεκατομμυρίων. Αποτέλεσμα της πορείας αυτής της αλλαγής ήταν η τεράστια αύξηση του καταμερισμού της εργασίας και ο κατακερματισμός των αρμοδιοτήτων της διαχείρισης. Ένα επίσης κρίσιμο ζήτημα για τη λειτουργία των επιχειρήσεων, ήταν η εξορθολογισμένη και σωστή κατανομή των διαθέσιμων πόρων στους διάφορους τομείς, ώστε να μεγιστοποιείται η αποτελεσματικότητά τους. Στην ουσία, η ταχύτατη αυτή ανάπτυξη και η επαυξημένη ειδίκευση της εργασίας δημιούργησε τα προαναφερθέντα προβλήματα στην παραγωγική δομή τα οποία απασχολούν ακόμα και σήμερα τις επιχειρήσεις καθώς τότε διαφάνηκε ως αναγκαίος ένας καινούργιος τρόπος διαχείρισης και επίλυσης διαφόρων τέτοιων προβλημάτων. Η Επιχειρησιακή Έρευνα προσπαθεί να απαντήσει σε αυτά τα προβλήματα που δημιουργούνται.

Ωστόσο η «γέννησή» της παρατηρείται κατά την περίοδο του Β' παγκόσμιου πολέμου όπου τέθηκαν οι βάσεις για τη δημιουργία του επιστημονικού κλάδου που ονομάζεται ¨ Επιχειρησιακή Έρευνα ¨. Ο όρος Επιχειρησιακή Έρευνα (Operational Research) προκύπτει από τις συγκεκριμένες έρευνες που διεξήχθησαν για την ανάπτυξη μαθηματικών μοντέλων για στρατιωτικές επιχειρήσεις (operations) κι εν γένει για τον εντοπισμό των αποτελεσματικότερων τρόπων χρήσης και διαχείρισης των περιορισμένων στρατιωτικών πόρων. Οι τεχνικές της Επιχειρησιακής Έρευνας, ιστορικά, έχουν τις ρίζες τους στις προσπάθειες των Συμμάχων να αντιμετωπίσουν με τον πλέον αποτελεσματικό τρόπο τις δυνάμεις του Άξονα. Συγκριμένα, την περίοδο αυτή επιστρατεύτηκαν από τη Βρετανική στρατιωτική διοίκηση ένας μεγάλος αριθμός επιστημόνων διάφορων κλάδων ώστε να εφαρμόσουν μια επιστημονική προσέγγιση για να αντιμετωπίσουν με αυτό τον τρόπο προβλήματα στρατηγικής , τακτικής και οργάνωσης που προέκυπταν κατά τη διάρκεια του πολέμου. Στην πραγματικότητα, ξεκίνησαν την λήψη επιστημονικών αποφάσεων όσον αφορά τη διαχείριση του πολεμικού υλικού με τον καλύτερο δυνατό τρόπο. Ειδικότερα, αυτές οι ομάδες επιστημόνων ασχολήθηκαν με τη μελέτη προβλημάτων σχετικών με την επιχειρησιακή λειτουργία των ραντάρ, τον προσδιορισμό του βέλτιστου μεγέθους των νηοπομπών, τον εντοπισμό και βομβαρδισμό των εχθρικών υποβρυχίων, κλπ συμβάλλοντας σημαντικά στην επικράτηση των συμμαχικών δυνάμεων.

Η επιτυχία που αποδόθηκε στην Επιχειρησιακή Έρευνα στις πολεμικές επιχειρήσεις, οδήγησε στην εφαρμογή αντίστοιχης έρευνας και κατά την μετέπειτα περίοδο της οικονομικής άνθισης που ακολούθησε το Β’ Παγκόσμιο. Οι έρευνες αυτές έμελε να θέσουν τις βάσεις μιας νέας φιλοσοφίας για την αντιμετώπιση πολύπλοκων προβλημάτων διοίκησης. Έπειτα από τον τερματισμό του πολέμου, αρκετοί από τους επιστήμονες που έλαβαν μέρος στις ομάδες επιχειρησιακών ερευνών κατά τη διάρκεια αυτού, απασχολήθηκαν σε επιχειρήσεις κι οργανισμούς, όπου οδηγήθηκαν στη διαπίστωση πως τα προβλήματα που καλούνταν να αντιμετωπίσουν και να επιλύσουν ήταν επί της ουσίας ίδια με αυτά που είχαν αντιμετωπίσει κατά τη διάρκεια του Β’ Παγκοσμίου, βέβαια σε διαφορετικό πεδίο εφαρμογής. Με αυτό τον τρόπο, προκύπτουν οι πρώτες εφαρμογές της Επιχειρησιακής Έρευνας σε προβλήματα Διοίκησης Επιχειρήσεων, ενώ αναπτύσσονται ταυτόχρονα νέες μέθοδοι της Επιχειρησιακής Έρευνας που ενσωματώνονται στις ήδη υπάρχουσες τεχνικές για την επίλυση πολύπλοκων προβλημάτων του εμπορίου, της βιομηχανίας, της δημόσιας διοίκησης και των διαφόρων υπηρεσιών. Μια τέτοια μέθοδος είναι και η Simplex για την επίλυση προβλημάτων γραμμικού προγραμματισμού, που παρουσιάστηκε το 1947 από τον George Dantzig.

Η ανάπτυξή της συνεχίστηκε και τις επόμενες δεκαετίες χάρις σε τρεις ευνοϊκούς παράγοντες: τη σημαντική πρόοδο που σημειώθηκε στις τεχνικές της Επιχειρησιακής Έρευνας, την αξιοσημείωτη και ραγδαία εξέλιξη των ηλεκτρονικών υπολογιστών, καθώς τα περισσότερα προβλήματα που πρέπει να αντιμετωπίσει η Επιχειρησιακή Έρευνα είναι αδύνατο να επιλυθούν με το χέρι, την επέκταση της χρήση των προσωπικών υπολογιστών κι ιδιώς την εξέλιξη της τους σε επίπεδο υλικού και τέλος, την ανάπτυξη του τομέα του λογισμικού (software).Όλοι οι παραπάνω παράγοντες συνέβαλαν στη δυνατότητα χρήση των μεθόδων της Επιχειρησιακής Έρευνας κι από μη ειδικούς, οδηγώντας στην περαιτέρω διάδοση της επίλυσης των διοικητικών προβλημάτων με την προσέγγιση της Ποσοτικής Ανάλυσης. Στο πλαίσιο αυτό, η Επιχειρησιακή Έρευνα εισήχθη σε διάφορους τομείς όπως η βιομηχανία, οι μεταφορές, η υγεία, κλπ αποδεικνύοντας το εύρος της εφαρμογής της σε μια πληθώρα επιχειρήσεων σήμερα.

1.3 Ορισμός και χρήση της Επιχειρησιακής Έρευνας

Ο απώτερος σκοπός της είναι να βοηθήσει τη διοίκηση να καθορίσει την πολιτική και τις ενέργειές της επιστημονικά (κατά τον καλύτερο δυνατό τρόπο).

Ορισμένοι ορισμοί που έχουν δοθεί:

- Κατά την Βρετανική Εταιρία Επιχειρησιακής Έρευνας :
  Η Επιχειρησιακή Έρευνα είναι η εφαρμογή της σύγχρονης επιστήμης πάνω σε πολύπλοκα προβλήματα που ανακύπτουν στη διεύθυνση και διοίκηση μεγάλων συστημάτων, αποτελούμενων από ανθρώπους, μηχανές, υλικά και κεφάλαια στις επιχειρήσεις.

- Επιχειρησιακή έρευνα μπορεί να θεωρηθεί σύμφωνα με τον Ackoff και Sasienni: η εφαρμογή επιστημονικών μεθόδων από μικτές ομάδες σε προβλήματα που αφορούν τον έλεγχο οργανωμένων συστημάτων (αποτελούμενο από ανθρώπους και μηχανές) κατά τρόπο ώστε να παρέχουν λύσεις που εξυπηρετούν κατά τον καλύτερο δυνατό τρόπο τους σκοπούς του οργανισμού ως συνόλου.

- Σύμφωνα με τον Ελληνικό ορισμό η επιχειρησιακή έρευνα: είναι η επιστημονική προετοιμασία των αποφάσεων της διοίκησης με την επιστημονική ανάλυση των δεδομένων και την δημιουργία μαθηματικών προτύπων.

1.3 Εφαρμογές της Επιχειρησιακής Έρευνας

Στον παρακάτω πίνακα παρουσιάζονται ενδεικτικά ορισμένα πραγματικά παραδείγματα χρήσης της Επιχειρησιακής Έρευνας σε επιχειρήσεις.

ΕΠΙΧΕΙΡΗΣΗ	ΕΦΑΡΜΟΓΗ	ΕΤΟΣ	ΕΤΗΣΙΑ ΚΕΡΔΗ
Monsanto Corporation	Βελτιστοποίηση της διαδικασίας παραγωγής χημικών εταιριών ώστε να επιτευχθούν οι στόχοι παραγωγής με ελάχιστο δυνατό κόστος	1985	$2 εκατομμύρια
IBM	Ενσωμάτωση ενός εθνικού δικτύου αποθήκευσης ανταλλακτικών για τη βελτίωση των υπηρεσιών υποστήριξης πελατών	1990	$20 εκατομμύρια Και $250 εκατομμύρια σε μικρότερο απόθεμα
Yellow Freight System, Inc.	Βελτιστοποίηση του σχεδιασμού ενός εθνικού δικτύου μεταφοράς και τη δρομολόγηση των αποστολών.	1992	$17,3 εκατομμύρια
Digital Equipment Corp.	Αναδιάρθρωση της εφοδιαστικής αλυσίδας των προμηθευτών, των εγκαταστάσεων, των κέντρων διανομής, πιθανές τοποθεσίες, και τομείς της αγοράς.	1995	$800 εκατομμύρια
Hewlett-Packard	Επανασχεδιασμός του μεγέθους και της θέσης των αποθεμάτων στη γραμμή παραγωγής εκτυπωτών που συμβαδίζουν με τη ζήτηση	1998	$280 εκατομμύρια από πρόσθετα έσοδα

ΚΕΦΑΛΑΙΟ 2

Δίκτυα

2.1 Εισαγωγή

Τα δίκτυα εμφανίζονται σε πολυάριθμες συνθέσεις και σε ποικίλες μορφές. Διάφοροι τύποι δικτύων, όπως τα ηλεκτρονικά , της μεταφοράς και της επικοινωνίας είναι διάχυτα στην καθημερινότητά μας. Παράλληλα, οι αναπαραστάσεις τους χρησιμοποιούνται σε ευρεία κλίμακα για την επίλυση προβλημάτων σε αρκετά πεδία, μερικά χαρακτηριστικά παραδείγματα των οποίων είναι η παραγωγή, η διανομή, η σχεδίαση έργου, οι εγκαταστάσεις, η διαχείριση πόρων κι ο οικονομικός σχεδιασμός .

Στην πραγματικότητα, η αναπαράσταση δικτύου παρέχει βοήθεια αντιληπτικού τύπου, διευκολύνοντας την κατανόηση του προβλήματος μέσω της απεικόνισης της σχέσης μεταξύ των μερών του συστήματος που χρησιμοποιείται ουσιαστικά σε κάθε τομέα επιστημονικού, κοινωνικού κι οικονομικού εγχειρήματος.

Το αντικείμενο μελέτης της θεωρίας δικτύων είναι η μαθηματική μοντελοποίηση εφαρμογών που περιέχουν κάποιο είδος δικτύου και η επίλυσή τους μέσω αλγορίθμων. Οι ρίζες της εντοπίζονται στο αντικείμενο μελέτης του Gustav Kirchhoff καθώς και άλλων ερευνητών της ηλεκτρολογίας και της μηχανικής που πρώτοι συστηματοποίησαν και ανέλυσαν ηλεκτρικά κυκλώματα. Αυτές οι μελέτες αποτελούν τη βάση πολλών βασικών ιδεών της θεωρίας δικτύων και καθιέρωσαν τα δίκτυα ως χρήσιμα μαθηματικά αντικείμενα για την αναπαράσταση πολλών φυσικών συστημάτων. Η γρήγορη ανάπτυξη της μεθοδολογίας αλλά και της εφαρμογής των μοντέλων βελτιστοποίησης των δικτύων αποτελεί ιδιαίτερα θετική κι ενδιαφέρουσα εξέλιξη για την Επιχειρησιακή Έρευνα. Στο πλαίσιο αυτό, σημαντική συνεισφορά έχουν και μια σειρά από επιτεύγματα στο πεδίο των αλγορίθμων, στην ίδια κατεύθυνση με τις ιδέες που εξάγονται και παρέχονται από την επιστήμη της πληροφορικής σχετικά με τις βάσεις δομένων και την αποτελεσματική επεξεργασία αυτών. Ως αποτέλεσμα, πλέον έχουμε στη διάθεσή μας προς χρήση για την επίλυση σημαντικών προβλημάτων αλγορίθμους και λογισμικό που θα ήταν ακατόρθωτο να υπήρχαν πριν από δύο ή τρεις δεκαετίες. Πολλά μοντέλα βελτιστοποίησης δικτύων είναι στην πραγματικότητα ειδικοί τύποι γραμμικού προγραμματισμού, όπως τα προβλήματα μεταφοράς και τα προβλήματα εκχώρησης (ανάθεσης εργασίας).

Κάποια από τα σημαντικότερα προβλήματα της θεωρίας δικτύων είναι τα ακόλουθα:

- **Πρόβλημα συντομότερης διαδρομής-shortest path problem**

  Πρόκειται για τα προβλήματα στα οποία καλούμαστε να εντοπίσουμε την ελάχιστη διαδρομή που χρειάζεται να διασχίσουμε σε ένα δίκτυο ώστε να μεταφερθούμε από ένα σημείο σε ένα άλλο. Τέτοιο πρόβλημα για παράδειγμα είναι η εύρεση συντομότερης διαδρομής μεταξύ δύο τοποθεσιών σε ένα υπάρχον δίκτυο (π.χ. google maps).

- **Πρόβλημα μέγιστης ροής-maximum flow problem**

  Αφορά την κατηγορία προβλημάτων που ζητούν τη μέγιστη δυνατή χωρητικότητα της διαδρομής σε ένα δίκτυο από ένα σημείο σε ένα άλλο. Χαρακτηριστικό παράδειγμα είναι η σχεδίαση συστημάτων μεταφοράς και διανομής πετρελαίου και φυσικού αερίου

- **Πρόβλημα ελάχιστου κόστους ροής-minimum cost flow problem**

  Αποτελούν τα προβλήματα βελτιστοποίησης και λήψης αποφάσεων για να βρεθεί ο "φθηνότερος" δυνατός τρόπος αποστολής συγκεκριμένης ποσότητας ροής μέσω του δικτύου ροής. Για παράδειγμα μια τυπική εφαρμογή είναι αυτό του προβλήματος που αναζητούμε την καλύτερη διαδρομή παράδοσης σε ένα εργοστάσιο και μια αποθήκη όπου το δίκτυο έχει κάποια χωρητικότητα και κόστος.

2.2 Βασικές έννοιες και ορολογία

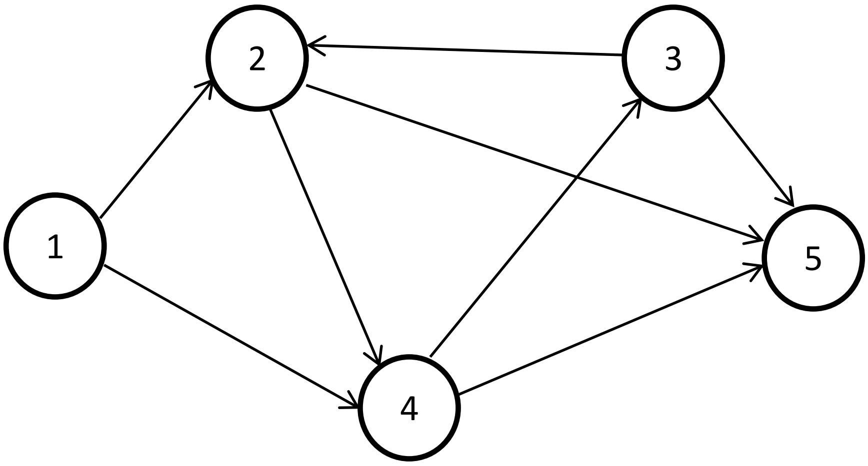
Υπάρχει μια αρκετά εκτεταμένη ορολογία για την περιγραφή των διαφόρων ειδών δικτύων και των μερών του. Στο σημείο αυτό , θα αναφέρουμε τη βασικότερη ορολογία , ώστε να περιγράψουμε μια σχετικά απλή μορφή.

Ένα **γράφημα** (graph) αποτελείται από ένα σύνολο σημείων συνδέσεων, που καλούνται κόμβοι (ή κορυφές) όπου ορισμένα ζεύγη κόμβων συνδέονται με γραμμές που ονομάζονται ακμές (ή κλάδοι ή δεσμοί). **Δίκτυο** (network) θεωρείται ένα γράφημα με κάποιου είδους ροή στις ακμές του.

Πίνακας 2.1 Παραδείγματα δικτύων

Σύστημα	Κόμβοι	Ακμές	Τιμή Ακμών	Ροή
Συγκοινωνιακά δίκτυα	Πόλεις, Διασταυρώσεις, Στάσεις , Σταθμοί επιβατών	Εθνικοί οδοί, Γραμμές τραίνων, κ.τ.λ.	Απόσταση, Χρόνος ταξιδιού ,Κόστος	Οχήματα, Μέσα μεταφοράς
Δίκτυα απορριμμάτων	Μονάδες επεξεργασίας αποβλήτων	Αγωγοί	Χρόνος και κόστος μεταφοράς	Απόβλητα , σκουπίδια
Δίκτυα υδροδότησης ή άρδευσης	Αντλιοστάσια, Σημεία κατανάλωσης νερού	Σωληνώσεις	Μήκος, Κόστος	Νερό, Υγρά
Δίκτυα ηλεκτρονικής επικοινωνίας	Υπολογιστές, εκτυπωτές	Καλώδια, Συνδέσεις ασύρματης επικοινωνίας	Μήκος καλωδίου, Κόστος	Δεδομένα

Ο συμβολισμός για την περιγραφή ενός δικτύου είναι **(N,A)** , όπου **N**: το σύνολο των κόμβων και **A**: το σύνολο των ακμών.



Σχήμα 2.2 Παράδειγμα ενός δικτύου (N,A)

Το παραπάνω δίκτυο του Σχήματος 2.2 περιγράφεται ως :

N = {1,2,3,4,5}

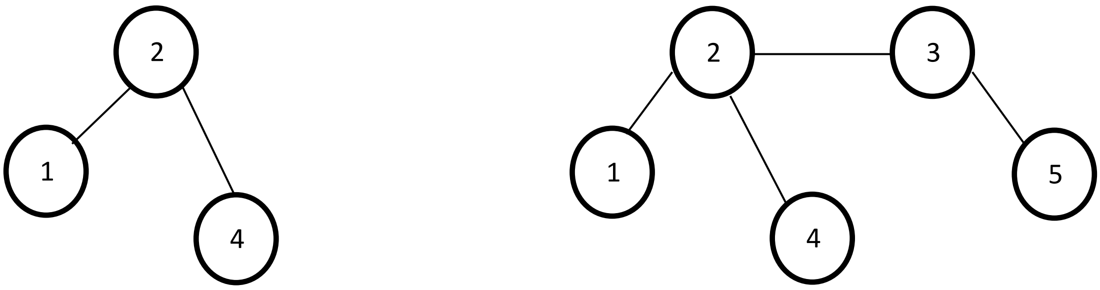
A = {(1,2),(1,4),(2,4),(2,5),(3,2),(3,5),(4,3),(4,5)}

Αν κάθε ακμή έχει μία συγκεκριμένη διεύθυνση, τότε το δίκτυο ονομάζεται **προσανατολισμένο** ή **κατευθυνόμενη** (directed ή oriented), διαφορετικά ονομάζεται **μη προσανατολισμένο** (undirected). Αν ορισμένες ακμές έχουν διεύθυνση και κάποιοι άλλες όχι, τότε το δίκτυο ονομάζεται **μεικτό** (mixed). Η **δυναμικότητα ροής** (flow capacity) μιας ακμής προς μια συγκεκριμένη κατεύθυνση είναι το μέγιστο εφικτό μέγεθος του ρυθμού ροής της ακμής προς αυτή την κατεύθυνση. Η δυναμικότητα ροής λαμβάνει οποιαδήποτε μη αρνητική ποσότητα. Ακόμη, μια ακμή είναι προσανατολισμένη αν η δυναμικότητα ροής μιας κατεύθυνσης είναι μηδέν.

Δύο κόμβοι που συνδέονται με μια ακμή καθώς και δύο ακμές που συνδέονται με έναν κόμβο ονομάζονται **γειτονικοί**. Ο **βαθμός** ενός κόμβου σε ένα μη προσανατολισμένο δίκτυο είναι ο αριθμός των ακμών των οποίων μία κορυφή είναι αυτός ο κόμβος. Σε ένα προσανατολισμένο δίκτυο, ορίζεται αντίστοιχα ο βαθμός για τον αριθμό των ακμών που καταλήγουν σε αυτό τον κόμβο και ο βαθμός για τον αριθμό των ακμών που απομακρύνονται από αυτό τον κόμβο .

Ένα **μονοπάτι** (ή **αλυσίδα**) σε ένα μη προσανατολισμένο δίκτυο είναι μία αλληλουχία γειτονικών κλάδων και ακμών. Σε ένα προσανατολισμένο δίκτυο, τα μονοπάτια έχουν και αυτά διεύθυνση. Για παράδειγμα , ένα μονοπάτι που συνδέει τα 1 και 5 στο Σχήμα 2.2 είναι τα (1,4) (4,5) . Ένα μονοπάτι είναι **απλό** αν κάθε κλάδος εμφανίζεται το πολύ μία φορά στην αλληλουχία και βασικό αν κάθε κόμβος εμφανίζεται το πολύ μία φορά στην αλληλουχία. **Κύκλος** (ή **κύκλωμα**) είναι ένα μονοπάτι του οποίου ο αρχικός και ο τελικός κόμβος συμπίπτουν. Τα (2,4) (4,3) (3,2) σχηματίζουν ένα κύκλο.

Ένα δίκτυο ονομάζεται **συνεκτικό** (connected) αν δυο οποιοιδήποτε κόμβοι του συνδέονται μέσω τουλάχιστον ενός μονοπατιού. Όταν υπάρχει μονοπάτι που να οδηγεί από κάθε κόμβο σε ένα άλλο, σε ένα προσανατολισμένο δίκτυο, τότε αυτό ονομάζεται **ισχυρά συνεκτικό**. **Δέντρο** (tree) είναι ένα συνεκτικό γράφημα χωρίς κύκλους, ενώ ένα **δέντρο κάλυμμα** είναι ένα δέντρο που συνδέει όλους τους κόμβους του δικτύου. Ένα δέντρο σε ένα δίκτυο με n κόμβους περιέχει ακριβώς (n-1) κλάδους. Ακόμη, κάθε ζεύγος κόμβων ενός δέντρου συνδέονται μέσω ενός μοναδικού μονοπατιού. Στο παρακάτω Σχήμα 2.3 παρουσιάζονται παραδείγματα ενός απλού και ενός δέντρου καλύμματος όσον αφορά το δίκτυο του Σχήματος 2.2 .



Δέντρο

Δέντρο Κάλυμμα

Σχήμα 2.3 Παραδείγματα δέντρου και δέντρου καλύμματος

Ένας κόμβος ενός δικτύου θεωρείται **πηγή** (source) αν κάθε μια από τις ακμές του έχει κατεύθυνση τέτοια ώστε η ροή να απομακρύνεται από την κορυφή. **Δέκτης** (sink) καλείται ο κόμβος όπου κάθε μια από τις ακμές του προσανατολίζεται προς αυτό.

ΚΕΦΑΛΑΙΟ 3

Το πρόβλημα της Ελάχιστης Διαδρομής (Shortest-Path Problem)

3.1 Εισαγωγή-Ορισμός

Ένας χαρακτηριστικός τύπος προβλημάτων των δικτύων ροής - ίσως το συνηθέστερο πρόβλημα που καλείται να επιλύσει κανείς σε ένα δίκτυο- είναι τα προβλήματα Ελάχιστης Διαδρομής. Ο στόχος σε αυτού του είδους τα προβλήματα είναι ο εντοπισμός της συντομότερης διαδρομής, στην ουσία της διαδρομής με το μικρότερο συνολικό μήκος ακμών (ή κόστος, χρονική διάρκεια, κίνδυνο, κλπ) από ένα σημείο εκκίνησης προς ένα τερματικό κόμβο σε ένα δίκτυο μεταφοράς. Σε πρακτικό επίπεδο, τα προβλήματα ελάχιστης διαδρομής προκύπτουν κατά κόρον όταν επιθυμούμε να στείλουμε κάποιο υλικό (πχ. ένα όχημα, ένα σύνολο δεδομένων υπολογιστή, κλπ) ανάμεσα σε δύο καθορισμένα σημεία , όσο το δυνατόν πιο γρήγορα, φτηνά κι αξιόπιστα. Με τη βοήθεια ενός αλγορίθμου εκτιμάται το μήκος (σε χρονική διάρκεια, κόστος, απόσταση, κλπ) κάθε πιθανής διαδρομής μεταξύ δύο κόμβων του δικτύου κι υποδεικνύεται η βέλτιστη.

Τα προβλήματα ελάχιστης διαδρομής παρουσιάζουν ενδιαφέρον τόσο σε ερευνητικό, όσο και σε επαγγελματικό επίπεδο για τους εξής λόγους:

1) Αποτελούν τη βάση για να μελετηθούν πιο σύνθετα μοντέλα δικτύων, καθώς πρόκειται για μια σχετικά απλή μορφή μοντέλων δικτύου, διαθέτοντας τα πιο βασικά χαρακτηριστικά του πυρήνα των δικτύων ροής.

2) Ενδεχομένως να προκύψουν αρκετά συχνά ως υπό-προβλήματα κατά την επίλυση πολλών συνδυαστικών προβλημάτων ή μοντέλων βελτιστοποίησης.

3) Η επίλυσή τους είναι σχετικά εύκολη

Αν και η επίλυσή τους δεν παρουσιάζει ιδιαίτερες δυσκολίες, παρόλα αυτά ο σχεδιασμός κι η ανάλυση των περισσότερων αποδοτικών αλγορίθμων προκειμένου να επιλυθούν απαιτεί ιδιαίτερες ικανότητες. Εξαιτίας αυτού, η εξέταση των προβλημάτων ελάχιστης διαδρομής αποτέλεσε την απαρχή ώστε να εισαγάγουμε πολλές ιδέες κλειδιά από τα δίκτυα ροής συμπεριλαμβανομένης της χρήσης έξυπνων δομών δεδομένων.

3.2 Αλγόριθμοι ελάχιστης διαδρομής

Στην ενότητα αυτή παρουσιάζονται δύο βασικοί αλγόριθμοι που χρησιμοποιούνται για την επίλυση δικτύων, είτε άκυκλων δικτύων, είτε κυκλικών (δικτύων που περιέχουν βρόγχους):

- Ο πρώτος αλγόριθμος ονομάζεται (αλγόριθμος του) *Dijkstra* και χρησιμοποιείται για τον προσδιορισμό της συντομότερης διαδρομής *ανάμεσα στον κόμβο και στην πηγή, καθώς και σε κάθε άλλο κόμβο του δικτύου.*
- Ο δεύτερος αλγόριθμος ονομάζεται (αλγόριθμος του) *Floyd* και χρησιμοποιείται για τον προσδιορισμό της συντομότερης διαδρομής ανάμεσα σε *δύο οποιουσδήποτε κόμβους του δικτύου*.

3.2.1 Αλγόριθμος Dijkstra

1. Γενικά-Εισαγωγή

Ο αλγόριθμος Dijkstra, δημιουργήθηκε από τον επιστήμονα Edsger Dijkstra το 1956 και εκδόθηκε το 1959 είναι ένας αλγόριθμος αναζήτησης γράφου ο οποίος χρησιμοποιείται για την εύρεση των συντομότερων μονοπατιών μεταξύ των κόμβων με μη αρνητικά βάρη στις ακμές του, δημιουργώντας ένα δέντρο σύντομου μονοπατιού. Αυτός ο αλγόριθμος χρησιμοποιείται συχνά ως υπορουτίνα για την δρομολόγηση σε άλλους αλγόριθμους γράφων. Για ένα συγκεκριμένο κόμβο-πηγή σε ένα γράφο ο αλγόριθμος βρίσκει το μονοπάτι με το ελάχιστο κόστος μεταξύ αυτού του κόμβου-πηγή και οποιουδήποτε άλλου κόμβου στο γράφο. Μπορεί να χρησιμοποιηθεί, επίσης, για να βρεθούν τα κόστη των συντομότερων μονοπατιών από έναν μονό κόμβο σε ένα μόνο κόμβο προορισμού σταματώντας τον αλγόριθμο με το που εντοπιστεί ο κόμβος προορισμού. Για παράδειγμα, εάν οι κόμβοι του γράφου αναπαριστούν πόλεις και τα κόστη των ακμών αναπαριστούν αποστάσεις μεταξύ δύο πόλεων που συνδέονται μέσω δρόμου, ο αλγόριθμος Dijkstra μπορεί να χρησιμοποιηθεί για να βρει τη συντομότερη διαδρομή μεταξύ μιας πόλης και όλων των υπολοίπων πόλεων.

2. Βήματα αλγορίθμου

Αρχικά ορίζουμε έναν κόμβο ως τον κόμβο αρχή (ή κόμβο-πηγή) από όπου ο αλγόριθμος ξεκινάει. Θέτουμε a_i τη συντομότερη διαδρομή από τον κόμβο 1 στον κόμβο i και ορίζουμε d_{ij} το μήκος της ακμής (i,j). Στη συνέχεια ορίζεται η ετικέτα για έναν άμεσα διαδοχικό κόμβο ως $[a_i + d_{ij} , i]$, $d_{ij} \geq 0$ Ο αλγόριθμος Dijkstra θα αρχικοποιήσει αυτές τις αποστάσεις και στη συνέχεια σε βήματα θα τις ενημερώνει. Ο αλγόριθμος χωρίζει τους κόμβους σε δύο ομάδες αυτές που ορίζει ως *μόνιμες (*permanently labeled*)* και αυτές που ορίζει ως *προσωρινές (*temporarily labeled). Η απόσταση σε κάθε μόνιμο κόμβο αντιπροσωπεύει την ελάχιστη απόσταση από την πηγή σε αυτό τον κόμβο , ενώ για ένα προσωρινό η απόσταση είναι ένα άνω όριο στην ελάχιστη απόσταση μονοπατιού σε αυτό τον κόμβο.

Βήμα 1

Για την απόσταση του κόμβου-πηγή δίνουμε μια μόνιμη τιμή 0 (μηδέν) και για την απόσταση κάθε άλλου κόμβου τιμή ίση με ∞ (άπειρο).

Βήμα 2

Δημιουργούμε ένα σύνολο το οποίο θα περιέχει τους κόμβους τους οποίους ο αλγόριθμος δεν έχει ακόμη επισκεφτεί. Στο βήμα αυτό δεν έχουμε επισκεφτεί κανέναν κόμβο άρα το σύνολο αυτό θα περιέχει όλους τους κόμβους του γράφου. Θέτουμε τον αρχικό κόμβο ως τρέχον κόμβο.

Βήμα 3

Για τον τρέχον κόμβο, υπολογίζουμε τις αποστάσεις για τους γείτονες κόμβους του που ο αλγόριθμος δεν έχει ακόμα επισκεφτεί. Συγκρίνουμε την καινούρια απόσταση με την προηγούμενη και αν είναι μικρότερη αντικαθιστούμε την παλιά με την νέα. Για παράδειγμα, αν σε έναν τρέχον κόμβο Α έχει σημειωθεί ότι η απόστασή του είναι 5 και η ακμή που συνδέει αυτόν με έναν γείτονά του Β έχει μήκος 4 τότε η απόσταση του Β μέσω του Α θα είναι 5+4=9. Εάν ο Β πριν τον υπολογισμό αυτόν είχε απόσταση μεγαλύτερη του 9 τότε η απόσταση του Β αντικαθίσταται και γίνεται 9, διαφορετικά θα παραμείνει η παλιά της τιμή.

Βήμα 4

Όταν για όλους τους γείτονες του τρέχον κόμβου έχει ενημερωθεί η απόστασή τους, σημειώνουμε τον τρέχον κόμβο ότι ο αλγόριθμος τον έχει επισκεφτεί και τον αφαιρούμε από το σύνολο των κόμβων που δεν έχουμε επισκεφτεί. Επομένως ,ένας κόμβος τον οποίο ο αλγόριθμος έχει επισκεφτεί δεν θα ελεγχθεί εκ νέου.

<u>Βήμα 5</u>

Αν ο αλγόριθμος εκτελείται προκειμένου να βρεθεί η συντομότερη διαδρομή ανάμεσα σε δύο συγκεκριμένους κόμβους, τότε τερματίζεται όταν ο κόμβος προορισμού έχει σημειωθεί ότι τον έχουμε επισκεφτεί.

Αν ο αλγόριθμος εκτελείται για την εύρεση των αποστάσεων όλων των κόμβων του γράφου από τον κόμβο πηγή, τότε ο αλγόριθμος τερματίζεται όταν η μικρότερη απόσταση ανάμεσα στους κόμβους του συνόλου των κόμβων που δεν έχουμε επισκεφτεί είναι άπειρη. Αυτό συνεπάγεται ότι δεν υπάρχει σύνδεση ανάμεσα στον κόμβο πηγή και στους εναπομείναντες κόμβους.

<u>Βήμα 6</u>

Διαλέγουμε από το σύνολο των κόμβων που δεν έχουμε επισκεφτεί τον κόμβο με την μικρότερη απόσταση και τον θέτουμε ως τρέχον κόμβο. Στη συνέχεια, επιστρέφουμε στο βήμα 3.

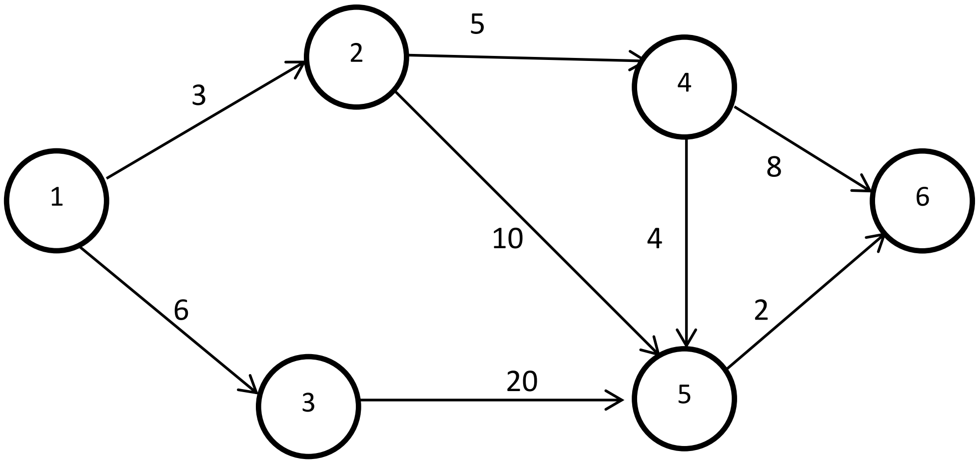
3. Πολυπλοκότητα αλγορίθμου

Η ανάλυση της πολυπλοκότητας του αλγορίθμου Dijkstra γίνεται ως εξής: Σε κάθε επανάληψη ο αλγόριθμος εκτελεί μια σειρά από ενέργειες που είναι ανάλογες του αριθμού των άλυτων κόμβων ο οποίος σε ένα δίκτυο με n κόμβους είναι αρχικά το πολύ n και μειώνεται σε κάθε επανάληψη. Με αυτό τον τρόπο η πολυπλοκότητα είναι : $O(n + n-1 + n-2 + \ldots + 1) = O(n^2)$

4.Παραδείγματα αλγορίθμου

<u>Παράδειγμα 1</u>

Το κύριο οδικό δίκτυο μιας πόλης με το κόστος (καθυστέρηση, χιλιομετρική απόσταση κ.τ.λ.) κάθε κλάδου εμφανίζονται στο επόμενο σχήμα. Αναζητούμε τη συντομότερη διαδρομή του δικτύου.



<u>Λύση:</u>

Εκκίνηση:

Αρχικοποίηση ετικετών των κόμβων.

Λίστα = {1}

	1	**2**	**3**	**4**	**5**	**6**
$a_i + d_{ij}$	0	∞	∞	∞	∞	∞
i	0	7	7	7	7	7
Κατάσταση	Μόνιμη	Προσωρινή	Προσωρινή	Προσωρινή	Προσωρινή	Προσωρινή

$1^η$ **Επανάληψη**:

Μόνιμος κόμβος =1

Λίστα = {}

	1	**2**	**3**	**4**	**5**	**6**
$a_i + d_{ij}$	0	3	6	∞	∞	∞
i	0	1	1	7	7	7
Κατάσταση	Μόνιμη	Προσωρινή	Προσωρινή	Προσωρινή	Προσωρινή	Προσωρινή

Λίστα = {2,3}

$2^η$ **Επανάληψη**:

Μόνιμος κόμβος = 2

Λίστα = {3}

	1	**2**	**3**	**4**	**5**	**6**
$a_i + d_{ij}$	0	3	6	8	13	∞
i	0	1	1	2	2	7
Κατάσταση	Μόνιμη	Μόνιμη	Προσωρινή	Προσωρινή	Προσωρινή	Προσωρινή

Λίστα = {3,4,5}

3η Επανάληψη:

Μόνιμος κόμβος = 3

Λίστα = {4,5}

	1	2	3	4	5	6
$a_i + d_{ij}$	0	3	6	8	13	∞
i	0	1	1	2	2	7
Κατάσταση	Μόνιμη	Μόνιμη	Μόνιμη	Προσωρινή	Προσωρινή	Προσωρινή

Λίστα = {4,5}

4η Επανάληψη:

Μόνιμος κόμβος = 4

Λίστα = {5,6}

	1	2	3	4	5	6
$a_i + d_{ij}$	0	3	6	8	12	16
i	0	1	1	2	4	4
Κατάσταση	Μόνιμη	Μόνιμη	Μόνιμη	Μόνιμη	Προσωρινή	Προσωρινή

Λίστα = {5,6}

5^{η} Επανάληψη:

Μόνιμος κόμβος = 5

Λίστα = {6}

	1	**2**	**3**	**4**	**5**	**6**
$a_i + d_{ij}$	0	3	6	8	12	14
i	0	1	1	2	4	5
Κατάσταση	Μόνιμη	Μόνιμη	Μόνιμη	Μόνιμη	Μόνιμη	Προσωρινή

Λίστα = {6}

6^{η} Επανάληψη:

Μόνιμος κόμβος =6

Λίστα = {} ,ΤΕΡΜΑΤΙΣΜΟΣ

	1	**2**	**3**	**4**	**5**	**6**
$a_i + d_{ij}$	0	3	6	8	12	14
i	0	1	1	2	4	5
Κατάσταση	Μόνιμη	Μόνιμη	Μόνιμη	Μόνιμη	Μόνιμη	Μόνιμη

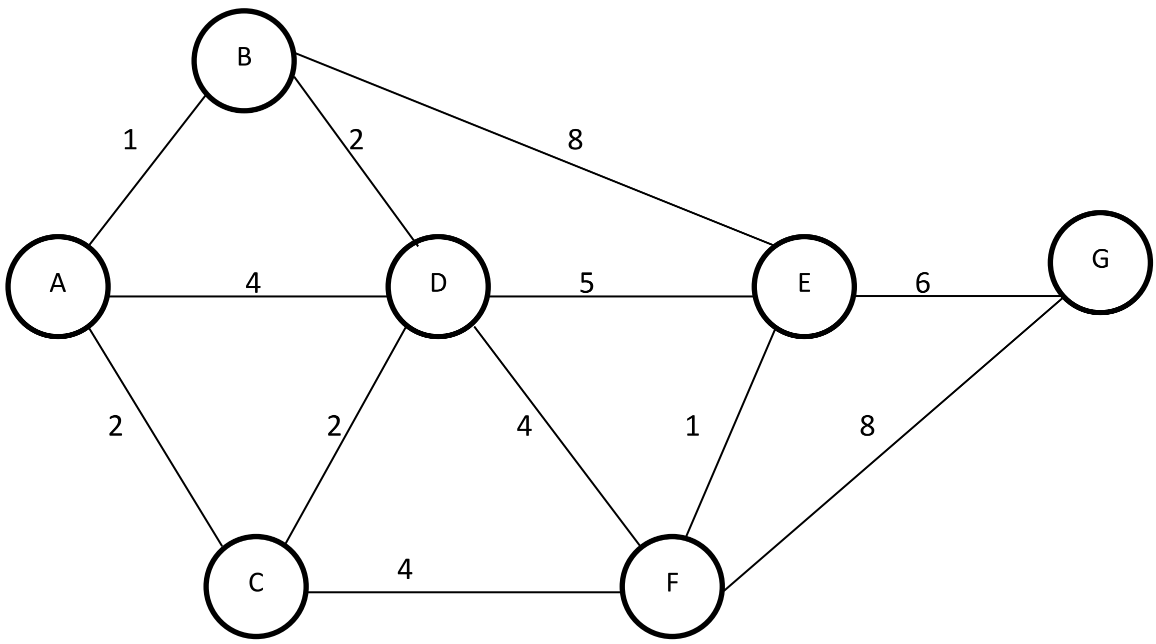
Κάνουμε χρήση του αλγορίθμου Dijkstra στο παραπάνω δίκτυο. Στην εκκίνηση η απόσταση του κόμβου 1 από την πηγή ορίζεται ως μηδέν ενώ οι υπόλοιποι ως άπειροι και ο κόμβος 1 γίνεται μόνιμος. Σε κάθε επανάληψη ένας κόμβος γίνεται μόνιμος , αυτό σημαίνει ότι η απόσταση από τον κόμβο προς την πηγή είναι καθορισμένη και στη συνέχεια συμπληρώνονται οι αποστάσεις προς του υπόλοιπους κόμβους που συνδέονται με τον μόνιμο. Αναλυτικότερα, στην 1^{η} επανάληψη το σύνολο Λίστα περιλαμβάνει μόνο τον κόμβο 1 (πηγή). Επομένως, ο

κόμβος 1 γίνεται μόνιμος και παίρνει την τιμή μηδέν συμπληρώνουμε τις αποστάσεις για τους κόμβους 2 και 3 και στις υπόλοιπες η απόσταση είναι άπειρο. Στη συνέχεια στην 2η επανάληψη ο κόμβος 2 είναι αυτός με την μικρότερη απόσταση από την πηγή επομένως αφαιρείται από τη Λίστα και η κατάστασή του μεταβάλλεται από προσωρινή σε μόνιμη. Έπειτα ενημερώνονται οι αποστάσεις προς τους κόμβους 4 και 5 που έχουν προστεθεί στη Λίστα. Στην 3η επανάληψη ο κόμβος 5 προσεγγίζεται από τον κόμβο 3 που έχει γίνει μόνιμος. Στην 4η επανάληψη προστίθεται ο κόμβος 6 στη Λίστα μαζί με τον κόμβο 5 και ενημερώνονται οι αποστάσεις. Στην 5η επανάληψη μεταβάλλεται η ετικέτα [16,4] και γίνεται [14,5] επειδή ο κόμβος 5 παρέχει μια συντομότερη διαδρομή. Ο αλγόριθμος του Dijkstra απαιτεί 6 επαναλήψεις για να κάνει μόνιμους όλους τους κόμβους στο δίκτυο.
Η συντομότερη διαδρομή ανάμεσα στον κόμβο 1 και σε οποιονδήποτε άλλο κόμβο του δικτύου προσδιορίζονται ξεκινώντας από τον επιθυμητό κόμβο προορισμού και επιστρέφοντας προς τον αρχικό κόμβο κάνοντας χρήση των μόνιμων ετικετών που έχουν συμπληρωθεί.
Για παράδειγμα η συντομότερη διαδρομή ανάμεσα στον κόμβο 1 και 6 είναι:

6 → [14,**5**] → **5** → [12,**4**] → **4** → [8,**2**] → **2** → [3,**1**] → **1**

<u>Παράδειγμα 2</u>

Αναζητούμε την ελάχιστη διαδρομή του παρακάτω δικτύου:



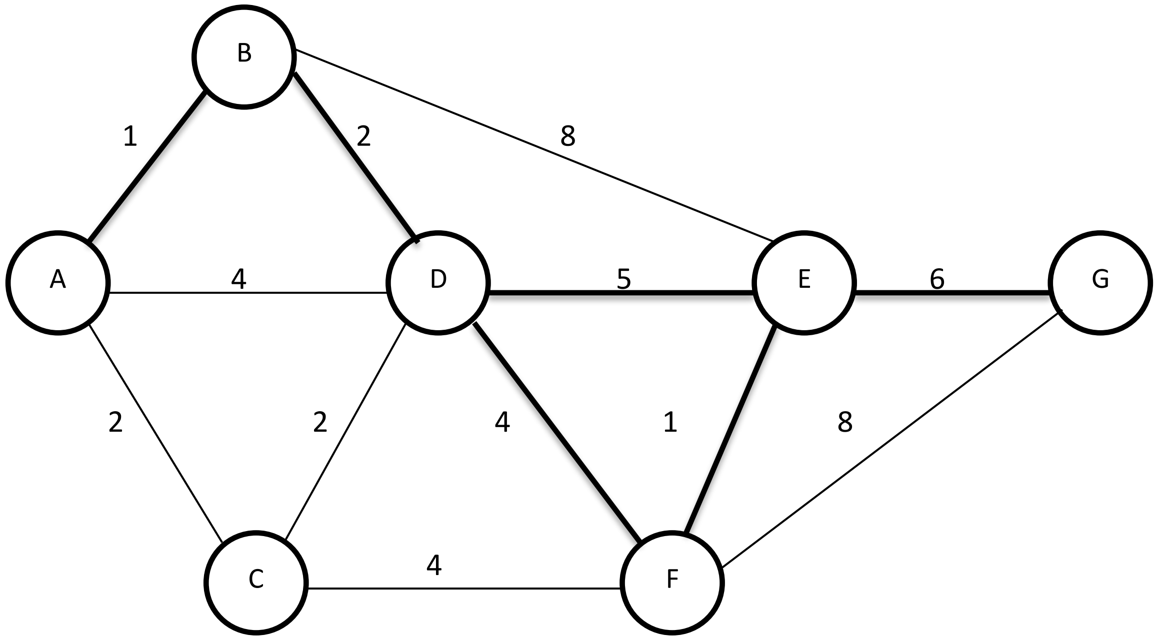
Λύση:

Εφαρμόζοντας τον αλγόριθμο Dijkstra λαμβάνουμε τα παρακάτω αποτελέσματα του πίνακα:

n	Λυμένοι κόμβοι άμεσα συνδεδεμένοι με άλυτους κόμβους	Πλησιέστερος συνδεδεμένος άλυτος κόμβος	Συνολική απόσταση	n-οστός πλησιέστερος κόμβος	Ελάχιστη απόσταση	Τελευταία Σύνδεση
1	A	B	1	B	1	AB
2,3	A B	C D	2 1+2=3	C D	2 3	AC BD
4	B C D	E F F	1+8=9 2+4=6 3+4=7	F	6	CF
5	B D F	E E E	1+8=9 3+5=8 7+1=8	E E	8 8	DE FE
6	E F	G G	8+6=14 7+8=15	G	14	EG

Στον παραπάνω πίνακα παρατηρούμε ότι η ισοβάθμηση στη δεύτερη επανάληψη μας δίνει τη δυνατότητα να προσθέσουμε δύο νέους λυμένους κόμβους, επομένως στην επόμενη επανάληψη γίνεται ο υπολογισμός του τέταρτου πλησιέστερου

κόμβου. Η ελάχιστη διαδρομή από την αφετηρία έως τον τερματισμό μπορεί να βρεθεί από την τελευταία στήλη του πίνακα και είναι η G →E→F→D→B→A ή G→E→D→B→A .
Επομένως , οι δύο εναλλακτικές λύσεις της συντομότερης διαδρομής από την αρχή μέχρι τον προορισμό είναι η A →B→D→F→E→G και A → B → D → E → G με συνολική απόσταση 14 η καθεμία.



3.2.2 Αλγόριθμος Floyd

1. Εισαγωγή

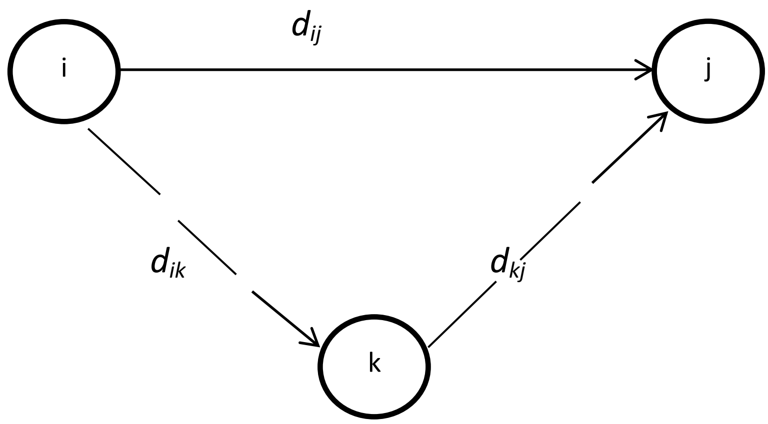
Ο αλγόριθμος Floyd είναι πιο γενικός συγκριτικά με τον αλγόριθμο του Dijkstra που παρουσιάστηκε στην προηγούμενη παράγραφο και έχει σχεδιαστεί για να υπολογίζει μια συντομότερη διαδρομή από κάθε κόμβο σε κάθε άλλο κόμβο του δικτύου. Ο αλγόριθμος απεικονίζει ένα δίκτυο (γράφημα) n κόμβων σε ένα τετραγωνικό πίνακα που αποτελείται από n γραμμές και n στήλες. Η τιμή του στοιχείου (i,j) του πίνακα , δίνει την απόσταση d_{ij} από τον κόμβο i στον κόμβο j η οποία είναι:

$d_{ij} = \begin{cases} aij\text{ , αν ο κόμβος i συνδέεται άμεσα με τον κόμβο j} \\ \infty\text{, διαφορετικά} \end{cases}$, όπου a_{ij} η τιμή του κελιού (i,j).

2. Βήματα αλγορίθμου

Η εφαρμογή του αλγορίθμου ξεκινά θεωρώντας τρεις κόμβους i,j,k (που φαίνονται και στο σχήμα 3.2) όπου γνωρίζουμε τις αποστάσεις τους ανάμεσά τους και θα υπολογίσουμε την μικρότερη απόσταση d_{ij} μέσω του κόμβου k , αν $d_{ik} + d_{kj} < d_{ij}$.

Σχήμα 3.2 : Η τριπλή λειτουργία του αλγορίθμου του Floyd



Στη συνέχεια τροποποιούμαι τη διαδρομή i→ j και την αντικαθιστούμε με την διαδρομή i →k→j . Η τριπλή λειτουργία εναλλαγής εφαρμόζεται στον πίνακα αποστάσεων χρησιμοποιώντας τα παρακάτω βήματα:

<u>ΒΗΜΑ 0</u>

Αρχικά ορίζουμε των πίνακα των αποστάσεων ως D_0 και τον πίνακα των κόμβων ως S_0. Τα διαγώνια στοιχεία των δύο πινάκων είναι φραγμένα και τα συμβολίζουμε με (-) .Θέτουμε k=1.

$D_0 =$

	1	2	…	j	…	N
1	–	d_{12}	…	d_{1j}	…	d_{1n}
2	d_{21}	–	…	d_{2j}	…	d_{2n}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
I	d_{i1}	d_{i2}	…	d_{ij}	…	d_{in}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	d_{n1}	d_{n2}	…	d_{nj}	…	–

$S_0 =$

	1	2	…	j	…	N
1	–	2	…	j	…	N
2	1	–	…	j	…	N
⋮	⋮	⋮	⋮	⋮	⋮	⋮
I	1	2	…	j	…	N
⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	1	2	…	j	…	–

<u>Γενικό Βήμα k.</u>

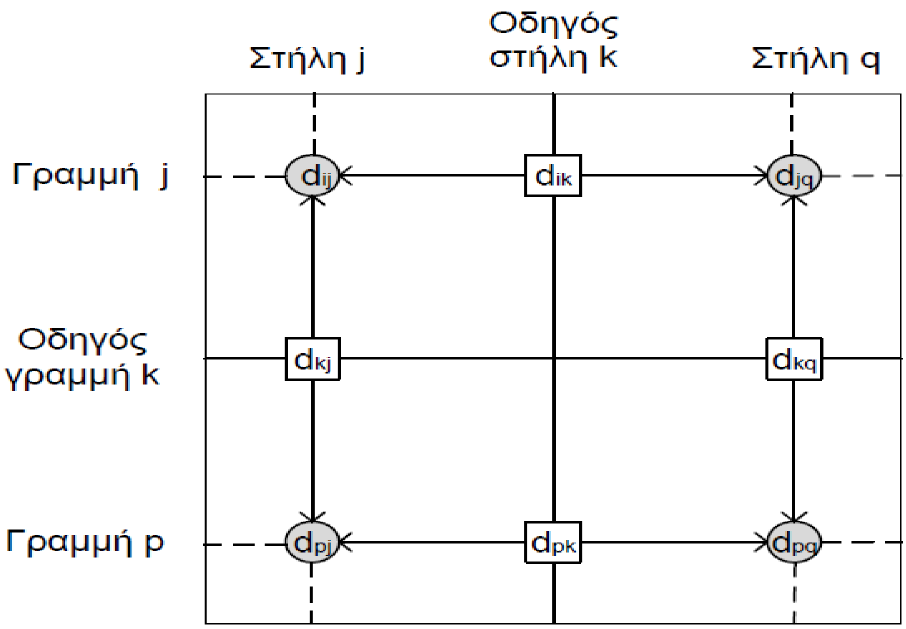
Στη συνέχεια καθορίζουμε την k γραμμή ως οδηγό γραμμή και την k στήλη ως οδηγό στήλη. Κάνουμε χρήση της τριπλής λειτουργίας σε κάθε στοιχείο d_{ij} του πίνακα D_{k-1} για όλες τις τιμές των i , j . Στην περίπτωση που ικανοποιείται η σχέση $d_{ik} + d_{kj} < d_{ij}$ (με i≠k , j≠k , i≠j) τροποποιούμαι ως εξής:

1. Δημιουργούμε τον πίνακα D_k αντικαθιστώντας το d_{ij} στον πίνακα D_{k-1} με το $d_{ik} + d_{kj}$.

2. Δημιουργούμε τον πίνακα S_k αντικαθιστώντας το s_{ij} στον πίνακα S_{k-1} με το k.Θέτουμε k=k+1.Αν k=n+1 ο αλγόριθμος σταματά σε αντίθετη περίπτωση επαναλαμβάνουμε το βήμα k.

Το γενικό βήμα k του αλγορίθμου μπορούμε να το εξηγήσουμε παρουσιάζοντας τον παρακάτω πίνακα D_{k-1} του σχήματος.. Η γραμμή k και η στήλη k ορίζουν τη γραμμή οδηγό και τη στήλη οδηγό αντίστοιχα. Η γραμμή i απεικονίζει οποιαδήποτε από τις γραμμές 1,2,…, k-1 και η γραμμή p απεικονίζει οποιαδήποτε από τις γραμμές k+1,k+2,…, n. Αντίστοιχα η στήλη j απεικονίζει οποιαδήποτε από τις στήλες 1,2,…, k-1 και η στήλη q απεικονίζει οποιαδήποτε από τις στήλες k+1,k+2,…, n. Η τριπλή λειτουργία εφαρμόζεται ως εξής: αν το άθροισμα των στοιχείων της γραμμής οδηγού και στήλης οδηγού (απεικονίζονται με τετράγωνα στο σχήμα) είναι μικρότερο από το στοιχείο της τομής της γραμμής οδηγού και στήλης οδηγού (απεικονίζεται με κύκλο) τότε αντικαθιστούμε την απόσταση της τομής με το άθροισμα της οδηγού απόστασης.

Σχήμα 3.3



Μετά από n βήματα μπορούμε να προσδιορίσουμε τη συντομότερη απόσταση ανάμεσα στους κόμβους i και j μέσω των πινάκων D_n και S_n με βάση τους παρακάτω κανόνες:

- Από τον πίνακα D_n το στοιχείο d_{ij} δηλώνει την ελάχιστη διαδρομή ανάμεσα στους κόμβους i και j.
- Από τον πίνακα S_n βρίσκουμε τον ενδιάμεσο κόμβο $k=s_{ij}$ που δίνει τη διαδρομή i →k→j . Αν s_{ik} = k και s_{kj} = j η διαδικασία διακόπτεται και όλοι οι

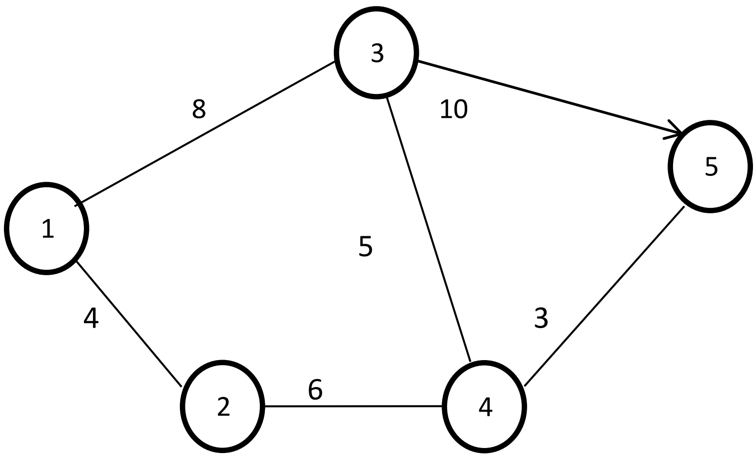
ενδιάμεσοι κόμβοι έχουν βρεθεί. Σε αντίθετη περίπτωση η διαδικασία επαναλαμβάνεται διαδοχικά ανάμεσα στους κόμβους i και k και στους k και j κόμβους

Πολυπλοκότητα αλγορίθμου:

Ο αλγόριθμος Floyd συγκρίνει όλες τις πιθανές διαδρομές μέσω του γραφήματος μεταξύ κάθε ζεύγους κορυφών. Είναι σε θέση να το κάνει αυτο σε χρόνο $0(n^3)$

3. Παράδειγμα αλγορίθμου

Αναζητούμε τις συντομότερες διαδρομές ανάμεσα σε δύο οποιουσδήποτε κόμβους για το δίκτυο του επόμενου σχήματος. Η ακμή (3,5) είναι κατευθυνόμενη ως εκ τούτου δεν επιτρέπεται η κυκλοφορία από τον κόμβο 5 προς τον κόμβο 3.Όλες οι άλλες ακμές επιτρέπουν την κυκλοφορία και προς τις δύο κατευθύνσεις. Οι αποστάσεις δίνονται στις κορυφές και είναι υπολογισμένες σε μίλια.



Επανάληψη 0

Οι πίνακες D_0 και S_0 δίνουν την αρχική αναπαράσταση του δικτύου. Ο D_0 είναι συμμετρικός πίνακας εκτός από το στοιχείο $d_{53}= \infty$ εφόσον δεν επιτρέπεται η κυκλοφορία από τον κόμβο 5 στον κόμβο 3.

D_0

	1	2	3	4	5
1	-	4	8	∞	∞
2	4	-	∞	6	∞
3	8	∞	-	5	10
4	∞	6	5	-	3
5	∞	∞	∞	3	-

S_0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Επανάληψη 1

Θέτουμε k=1. Η πρώτη γραμμή και η πρώτη στήλη του πίνακα D_0 που είναι ελαφρώς σκιαγραφημένες , αντιστοιχούν στην γραμμή οδηγός και στη στήλη οδηγός. Τα πιο έντονα στοιχεία του πίνακα d_{23} και d_{32} είναι τα μοναδικά κελιά που μπορούν να βελτιωθούν από την τριπλή λειτουργία. Οι πίνακες D_1 και S_1 προκύπτουν με τον ακόλουθο τρόπο:

1. Αντικαθιστούμε την τιμή του στοιχείου d_{23} με την τιμή $d_{21} + d_{13} = 4 + 8 =12$ και θέτουμε $s_{23} = 1$.
2. Αντικαθιστούμε την τιμή του στοιχείου d_{32} με την τιμή $d_{31} + d_{12} = 8 + 4 =12$ και θέτουμε $s_{32} = 1$.

Παρακάτω παρουσιάζεται ο πίνακας D_1 και S_1 μετά τις παραπάνω αλλαγές.

D_1

	1	2	3	4	5
1	-	4	8	∞	∞
2	4	-	**12**	6	∞
3	8	**12**	-	5	10
4	∞	6	5	-	3
5	∞	∞	∞	3	-

S_1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	**1**	4	5
3	1	**1**	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Επανάληψη 2

Θέτουμε k=2 , όπου διακρίνεται από την ελαφρά σκιασμένη γραμμή και στήλη του πίνακα D_1 . Η τριπλή λειτουργία εφαρμόζεται στα πιο έντονα κελιά των πινάκων D_1 και S_1 . Παρακάτω παρουσιάζονται οι πίνακες D_2 και S_2 μετά τις αλλαγές που πραγματοποιήθηκαν με τον ίδιο τρόπο όπως και στην επανάληψη 1.

D_2

	1	2	3	4	5
1	-	4	8	**10**	∞
2	4	-	12	6	∞
3	8	12	-	5	10
4	**10**	6	5	-	3
5	∞	∞	∞	3	-

S_2

	1	2	3	4	5
1	-	2	3	**2**	5
2	1	-	1	4	5
3	1	1	-	4	5
4	**2**	2	3	-	5
5	1	2	3	4	-

Επανάληψη 3

Θέτουμε k=3 , όπου διακρίνεται από την ελαφρά σκιασμένη γραμμή και στήλη του πίνακα D_2 . Με τον ίδιο τρόπο προκύπτουν οι πίνακες D_3 και S_3 .

D_3

	1	2	3	4	5
1	-	4	8	10	**18**
2	4	-	12	6	**22**
3	8	12	-	5	10
4	10	6	5	-	3
5	∞	∞	∞	3	-

S_3

	1	2	3	4	5
1	-	2	3	2	**3**
2	1	-	1	4	**3**
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Επανάληψη 4

Θέτουμε k=4 , όπου διακρίνεται από την ελαφρά σκιασμένη γραμμή και στήλη του πίνακα D_3 . Με τον ίδιο τρόπο προκύπτουν οι πίνακες D_4 και S_4 .

D_4

	1	2	3	4	5
1	-	4	8	10	**13**
2	4	-	**11**	6	**9**
3	8	**11**	-	5	**8**
4	10	6	5	-	3
5	**13**	**9**	**8**	3	-

S_4

	1	2	3	4	5
1	-	2	3	2	**4**
2	1	-	**4**	4	**4**
3	1	**4**	-	4	**4**
4	2	2	3	-	5
5	**4**	**4**	**4**	4	-

Επανάληψη 5

Θέτουμε k=5 , όπου διακρίνεται από την ελαφρά σκιασμένη γραμμή και στήλη του πίνακα D_4 . Στην επανάληψη 5 δεν μπορούν να γίνουν κι άλλες βελτιώσεις, επομένως οι πίνακες D_5 και S_5 παραμένουν ίδιοι με τους D_4 και S_4 .

Οι τελικοί πίνακες D_4 και S_4 περιλαμβάνει όλες τις πληροφορίες που χρειάζονται για τον προσδιορισμό της συντομότερης διαδρομής ανάμεσα σε δύο οποιουσδήποτε κόμβους του δικτύου.

Για παράδειγμα θα αναλύσουμε τη συντομότερη απόσταση από τον κόμβο 1 στον κόμβο 5 είναι η d_{15} = 13 μίλια. Για τον υπολογισμό της αντίστοιχης διαδρομής αξίζει να σημειωθεί πως ένα τμήμα (i , j) αναπαριστά μια άμεση σύνδεση αν $s_{ij} = j$, σε αντίθετη περίπτωση τα i και j συνδέονται τουλάχιστον με τη βοήθεια ενός ακόμα κόμβου. Από τον πίνακα S_4 έχουμε $s_{15} = 4 \neq 5$ επομένως η διαδρομή αρχικά δίνεται ως 1→ 4 → 5 και επειδή $s_{14} = 2 \neq 4$ το τμήμα (1,4) δεν αποτελεί μία άμεση σύνδεση , επομένως η 1 → 4 θα αντικατασταθεί με τη διαδρομή 1 → 2 → 4 και η διαδρομή που μελετούμε θα γίνει 1 → 2 → 4 → 5 . Ακόμη , επειδή $s_{12} = 2$, $s_{24} = 4$ $s_{45} = 5$ δε χρειάζεται περαιτέρω ανάλυση επομένως η συντομότερη διαδρομή είναι η 1 → 2 → 4 → 5 .

ΚΕΦΑΛΑΙΟ 4

Το πρόβλημα της Μέγιστης Ροής (Max Flow Problem)

4.1 Εισαγωγή

Το πρόβλημα της Μέγιστης Ροής σχηματοποιήθηκε για πρώτη φορά από τους T.E. Harris και F.S. Ross με σκοπό να μετατρέψουν το πολύπλοκο σιδηροδρομικό δίκτυο της ΕΕΣΔ σε απλούστερη κι άρα υπολογίσιμη μορφή. Η επόμενη σημαντική συνεισφορά έρχεται έπειτα από τους Lester R. Ford Jr και Delber R. Fulkerson, μέσω του γνωστού αλγόριθμου Ford- Fulkerson όπου δημοσιεύτηκε το 1955, καθιστώντας ευκολότερο τον τρόπο υπολογισμού.

Τα προβλήματα Μέγιστης Ροής συναντώνται συχνά σε δίκτυα μεταφοράς (transportation network) αλλά και σε άλλα πεδία όπως στις γεωργικές εργασίες, στα συστήματα ύδρευσης κλπ. Τέτοιου είδους προβλήματα αντιμετωπίζουν κατά κόρον οι τεχνικές και κατασκευαστικές εταιρείες στην προσπάθεια τους να υπολογίσουν τη μέγιστη ροή οχημάτων που διασχίζουν ένα δρόμο κατά τη λήψη αποφάσεων διαπλάτυνσης και βελτιωτικών έργων.

Ένα δίκτυο μεταφοράς – στο οποίο παρουσιάζονται τέτοια προβλήματα- είναι ένα γενικό μοντέλο με διαδρομές μεταφοράς υλικών, ρευστών, προϊόντων, αγαθών, κλπ. από ένα κόμβο- αφετηρία (το οποίο αποκαλείται πηγή/ source) σε ένα κόμβο – προορισμό (το οποίο αποκαλείται δεξαμενή/ sink). Σε ένα δίκτυο μεταφοράς, οι κλάδοι αντιπροσωπεύουν τη μέγιστη δυνατή χωρητικότητα της διαδρομής από ένα κόμβο σε έναν άλλο, την οποία δεν μπορεί να υπερβεί η μεταφερόμενη ποσότητα. Για παράδειγμα, για ρευστά, οι κλάδοι θα είναι η μέγιστη παροχή των σωληνώσεων, για μεταφορές το μέγιστο φορτίο, για συγκοινωνίες οι μέγιστες θέσεις επιβατών, κ.λ.π.

Ορισμός: Στην ουσία, τα προβλήματα Μέγιστης Ροής ανακύπτουν όταν σε ένα δίκτυο με χωρητικότητες, κι επιθυμούμε να στείλουμε όσο το δυνατόν περισσότερη ροή ανάμεσα σε δύο συγκεκριμένου κόμβους, στον κόμβο-αφετηρία (πηγή- source node) και στον κόμβο- προορισμού (δεξαμενή- sink node), χωρίς να υπερβούμε τη χωρητικότητα κάποιου τόξου.

Στο πρόβλημα Μέγιστης Ροής διαθέτουμε μια πηγή που παράγει μια ορισμένη ποσότητα κι έναν προορισμό που δέχεται ένα μέρος της παραγόμενης ποσότητας καθώς αυτή διαμοιράζεται στους υπόλοιπους ενδιάμεσους, εσωτερικούς κόμβους. Σκοπός είναι να μεγιστοποιήσουμε την ποσότητα που θα καταλήξει εν τέλει στον κόμβο- προορισμό χωρίς να παραβιάσουμε τους περιορισμούς χωρητικότητας των ακμών.

Το σκεπτικό επίλυσης των προβλημάτων Μέγιστης Ροής στηρίζεται στην απαίτηση ότι σε κάθε κόμβο κατά μήκος της διαδρομής, η ποσότητα που καταλήγει σε αυτόν δεν ξεπερνά την χωρητικότητα που μπορεί να «αντέξει» αυτός ο κόμβος.

Κάθε ακμή μπορεί να έχει αμφίδρομη διεύθυνση και φέρει θετική χωρητικότητα. Η πηγή έχει μόνο εξερχόμενες ακμές ενώ ο προορισμός μόνο εισερχόμενες ακμές. Η ροή που εξέρχεται από έναν κόμβο είναι ίση με τη ροή που εισέρχεται σε αυτόν.

Ένα κλασικό παράδειγμα που προκύπτει πρόβλημα Μέγιστης Ροής είναι όταν μια επιχείρηση προσπαθεί να μεταφέρει όλα της τα προϊόντα στο κεντρικό κατάστημα από το εργοστάσιο παραγωγής. Το πρόβλημα που προκύπτει είναι το εξής: Πως μπορεί να συνδυάσει τους διάφορους δρόμους και μέσα μεταφοράς ώστε να μεταφέρει τη μέγιστη δυνατή ποσότητα το συντομότερο δυνατόν?

<u>Συσχέτιση προβλημάτων Συντομότερης Διαδρομής και Μέγιστης Ροής</u>

Τα προβλήματα Μέγιστης Ροής και τα προβλήματα Συντομότερης Διαδρομής παρουσιάζουν ένα βαθμό συμπληρωματικότητας. Το κοινό τους χαρακτηριστικό οφείλεται στο γεγονός ότι και τα δύο συμβαίνουν κατά την πράξη καθώς εμφανίζονται ως υποπροβλήματα σε αλγόριθμους ελαχίστους κόστους ροής. Ωστόσο, η διαφορά τους έγκειται στο ότι και οι δύο αυτοί τύποι προβλημάτων ασχολούνται με διαφορετικές πτυχές του προβλήματος ελαχίστου κόστους ροής: Τα προβλήματα Ελάχιστης Διαδρομής μοντελοποιούν τα κόστη των τόξων κι όχι τις χωρητικότητες τους, ενώ τα προβλήματα Μέγιστης Ροής ακριβώς το αντίθετο. Εάν συνενώσουμε αυτά τα δύο προβλήματα, συνδυάζουμε όλα τα βασικά στοιχεία των δικτύων ροής και γι αυτό έχουν γίνει οι πυρήνες της βελτιστοποίησης των δικτύων.

4.2 Ο Αλγόριθμος Ford – Fulkerson

Ο αλγόριθμος Ford-Fulkerson (L. R. Ford Jr, D. R. Fulkerson -1956), λύνει το πρόβλημα της μέγιστης ροής σε δίκτυα ροών. Αναφέρεται και ως μέθοδος Ford-Fulkerson ή μέθοδος μονοπατιού επαύξησης λόγω χρήσης του σε πολλές διαφορετικές υλοποιήσεις. Ανήκει στους αλγορίθμους γραμμικού προγραμματισμού και αποτελεί μια πιο αποδοτική λύση από τη λύση με τη μέθοδο simplex που είναι ο βασικός αλγόριθμος που χρησιμοποιείται στον γραμμικό προγραμματισμό. Ο αλγόριθμος βασίζεται σε δύο σημαντικές έννοιες, το **υπολειπόμενο δίκτυο (residual network)** και το **επαυξάνον μονοπάτι (augmenting path)**.

Το **υπολειπόμενο** δίκτυο δείχνει τις απομένουσες χωρητικότητες των κλάδων για την ανάθεση επιπλέον ροών. Πιο αναλυτικά επεξηγείται στο επόμενο παράδειγμα.

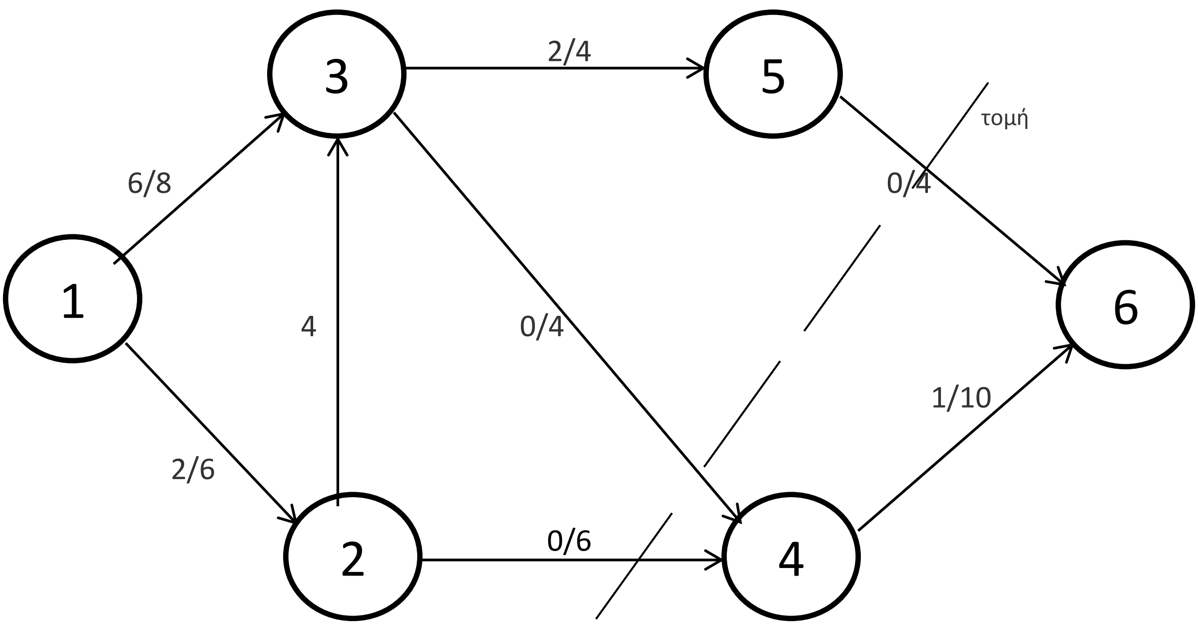
Έστω ο κλάδος 1-2 που φαίνεται στην παρακάτω εικόνα με χωρητικότητα 5.

Υποθέτουμε πως κάποια στιγμή εκχωρούμε μια ροή από τον κόμβο 1 στον κόμβο 2, οπότε η υπολειπόμενη ροή που μπορεί να εκχωρηθεί στην κατεύθυνση 1→2 είναι 5-3 = 2.Δηλαδή, η μεταβολή θα είναι η εξής:

Ο αριθμός σε έναν κλάδο δίπλα σε έναν κόμβο είναι ενδεικτικός της απομένουσα χωρητικότητα για ροή από αυτό τον κόμβο προς τον άλλο κόμβο. Συνεπώς, εξαιρουμένης της απομένουσας χωρητικότητας ίση με 2 για ροή από τον κόμβο 1 προς τον κόμβο 2, το 3 δεξιά δείχνει μία απομένουσα χωρητικότητα ίση με 3 για ροή από τον κόμβο 2 προς τον κόμβο 1 .

Ένα **επαυξάνον** μονοπάτι (το οποίο επιλέγεται τυχαία) είναι ένα προσανατολισμένο μονοπάτι από την πηγή προς το δέκτη στο υπολειπόμενο δίκτυο τέτοιο ώστε κάθε κλάδος σε αυτό το μονοπάτι να έχει αυστηρά θετική απομένουσα χωρητικότητα. Η ελάχιστη από αυτές τις απομένουσες χωρητικότητες είναι η απομένουσα χωρητικότητα του επαυξάνοντος μονοπατιού, καθώς αντιπροσωπεύει την ποσότητα ροής που μπορεί με εφικτό τρόπο να εκχωρηθεί σε ολόκληρο το μονοπάτι. Κατ΄ επέκταση, κάθε επαυξάνον μονοπάτι προσφέρει μία επιλογή για να αυξήσουμε περαιτέρω τη ροή σε ολόκληρο το αρχικό δίκτυο. Ο αλγόριθμος επιλέγει διαδοχικά κάποιο επαυξάνον μονοπάτι και προσθέτει μια ροή ίση με την απομένουσα χωρητικότητά του στο αρχικό δίκτυο. Αυτή η διαδικασία συνεχίζεται μέχρις ότου να μην υπάρχουν καθόλου επαυξάνοντα μονοπάτια, οπότε και δεν είναι δυνατόν να αυξηθεί περαιτέρω η ροή από την πηγή προς το δέκτη. Η βελτιστότητα της τελικής λύσης επισφραγίζεται από το ότι επαυξάνοντα μονοπάτια δύναται να ακυρώσουν κάποια ροή που έχει εκχωρηθεί προηγουμένως στο αρχικό δίκτυο, επομένως μία οποιαδήποτε επιλογή μονοπατιών για εκχώρηση ροής δεν μπορεί να δημιουργήσει εμπόδια στη χρήση ενός καλύτερου συνδυασμού μονοπατιών για εκχώρηση ροών.

Οι Ford & Fulkerson εισήγαγαν επίσης την έννοια της **τομής** ενός δικτύου. Ως τομή (cut) ορίζεται ένα σύνολο προσανατολισμένων κλάδων, που περιέχει τουλάχιστον έναν κλάδο από κάθε μονοπάτι από την πηγή στο δέκτη. Υπάρχουν πολλοί τρόποι να σχηματίσουμε μία τομή που κόβει στα δύο το δίκτυο. Η χωρητικότητα μιας τομής είναι το άθροισμα των χωρητικοτήτων των κλάδων που τέμνει και που έχουν κατεύθυνση από την είσοδο προς την έξοδο. Για παράδειγμα βλέπουμε τις τομές στο παρακάτω τυχαίο δίκτυο.



Δηλαδή για την τομή η χωρητικότητα είναι της τομής που περνά από τους τρεις κορεσμένους κλάδους (2,4), (3,4) & (5,6): 6+4+4 = 14.

Θεώρημα Μέγιστης Ροής και Ελάχιστης Τομής

Σε ένα δίκτυο η μέγιστη τιμή της ροής είναι ίση με την ελάχιστη από τις χωρητικότητες όλων των δυνατών τομών του δικτύου.

Δίνεται κατευθυνόμενος γράφος G=(V, E) με χωρητικότητα ακμής από τον κόμβο u στον κόμβο v ορισμένη ως c (u,v) και ροή δικτύου f (u, v). Ο αλγόριθμος υπολογίζει τη ροή από τον κόμβο-αφετηρία s μέχρι τον κόμβο-προορισμό t με τη μέγιστη τιμή.

Μετά από κάθε βήμα στον αλγόριθμο διατηρούνται τα εξής:

- Περιορισμοί χωρητικότητας : $\forall$ (u,v) ϵ E $f(u,v) \leq c(u,v)$
  Δηλαδή η ροή κατά μήκος μιας ακμής δεν μπορεί να υπερβεί την χωρητικότητά της
- Διαγώνια συμμετρία : $\forall$ (u,v) ϵ E $f(u,v) = -f(v,u)$
  Η ροή από το u στο v πρέπει να είναι αντίθετο της ροής από το v στο u.

- Διατήρηση ροής: $\forall$ u ϵ V : $u \neq s$ και $u \neq t$ => $\sum f(u,w) = 0$
  Δηλαδή, η ροή σε ένα κόμβο είναι μηδέν εκτός από την πηγή η οποία παράγει ροή (s) και την πηγή (t) που καταναλώνει ροή.

Ορίζουμε ως **υπολειπόμενο δίκτυο**, το δίκτυο με χωρητικότητα $c_f(u,v) = c(u,v) - f(u,v)$ και χωρίς ροή. Αξίζει να σημειωθεί ότι μπορεί να συμβεί μια ροή στο υπολειπόμενο δίκτυο από το u στο v παρ' όλο που δεν επιτρέπεται στο αρχικό δίκτυο αν $f(u,v) > 0$ και $c(v,u) = 0$, τότε $c_f(v,u) = c(v,u) - f(v,u) = f(u,v) > 0$.

Αλγόριθμος Ford & Fulkerson

1. Αρχικά, η ροή είναι μηδενική δηλαδή, f(u,v)=0 για κάθε (u,v) ϵ V. Σε κάθε επανάληψη, ο αλγόριθμος των Ford και Fulkerson χρησιμοποιεί την αναζήτηση κατά πλάτος (depth first search, DFS) για την εύρεση μονοπατιού επαύξησης P στο υπολειπόμενο δίκτυο G_f.
2. Όσο υπάρχει ένα τέτοιο μονοπάτι αυξάνει τη ροή f κατά μήκος αυτού του μονοπατιού με τον εξής τρόπο: Αφού πρώτα βρει την ελάχιστη υπολειπόμενη χωρητικότητα c_f (P) μεταξύ των ακμών που ανήκουν στο μονοπάτι επαύξησης P, στη συνέχεια για κάθε ακμή (u,v) ϵ P αυξάνει την τρέχουσα ροή κατά την τιμή της c_f (P) σε κάθε ακμή προς τα εμπρός (forward edge, u,v) και την μειώνει κατά το ίδιο ποσό σε κάθε ακμή προς τα πίσω (backwards edge, v,u).

3. Ο αλγόριθμος τερματίζει όταν δεν υπάρχει άλλο μονοπάτι επαύξησης και επιστρέφει την τιμή της μέγιστης ροής f.

Πολυπλοκότητα αλγορίθμου

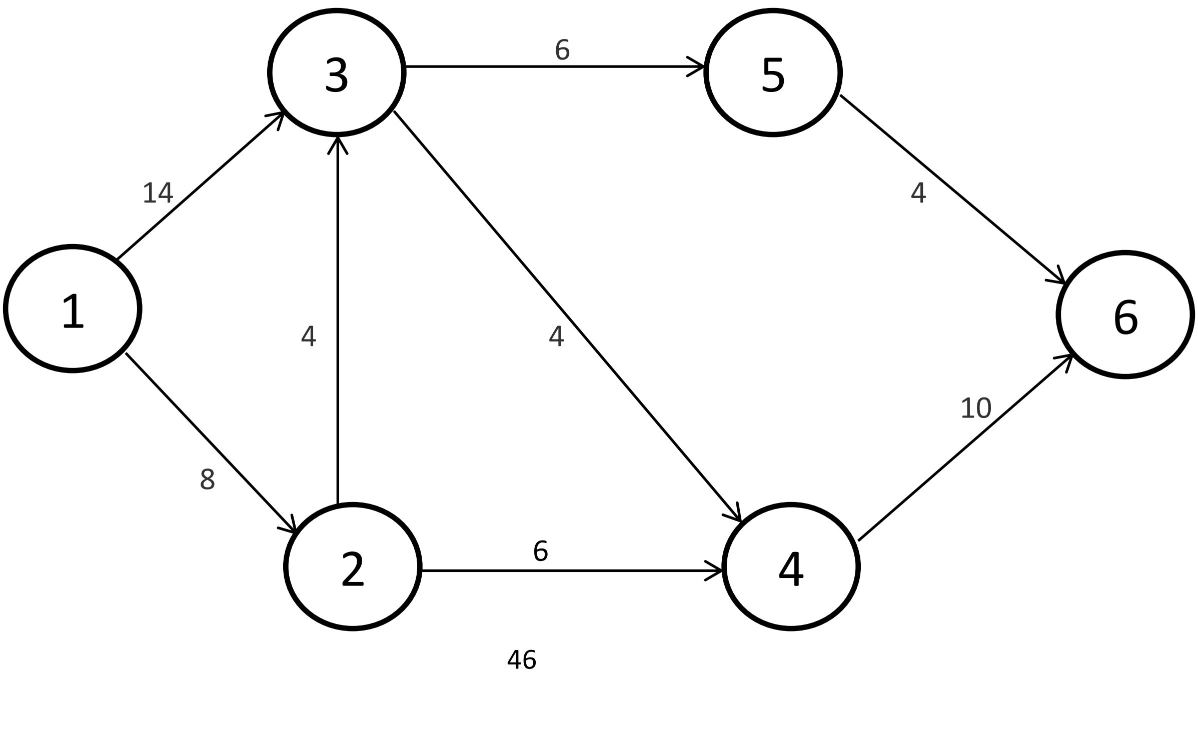
Έστω ότι όλες οι χωρητικότητες είναι ακέραιοι, οπότε η τιμή της μέγιστης ροής f* είναι ακέραιος. Τότε γίνονται το πολύ f* επαναλήψεις (η ροή αυξάνεται τουλάχιστον κατά 1 σε κάθε επανάληψη) και κάθε επανάληψη απαιτεί χρόνο O(|E|), υποθέτοντας ότι το επεκταμένο μονοπάτι επαύξησης βρίσκεται σε αναζήτηση κατά πλάτος.

Άρα ο συνολικός χρόνος εκτέλεσης είναι O (f*·|E|) .

4.3 Παραδείγματα στον αλγόριθμο Ford – Fulkerson

<u>ΠΑΡΑΔΕΙΓΜΑ 1</u>

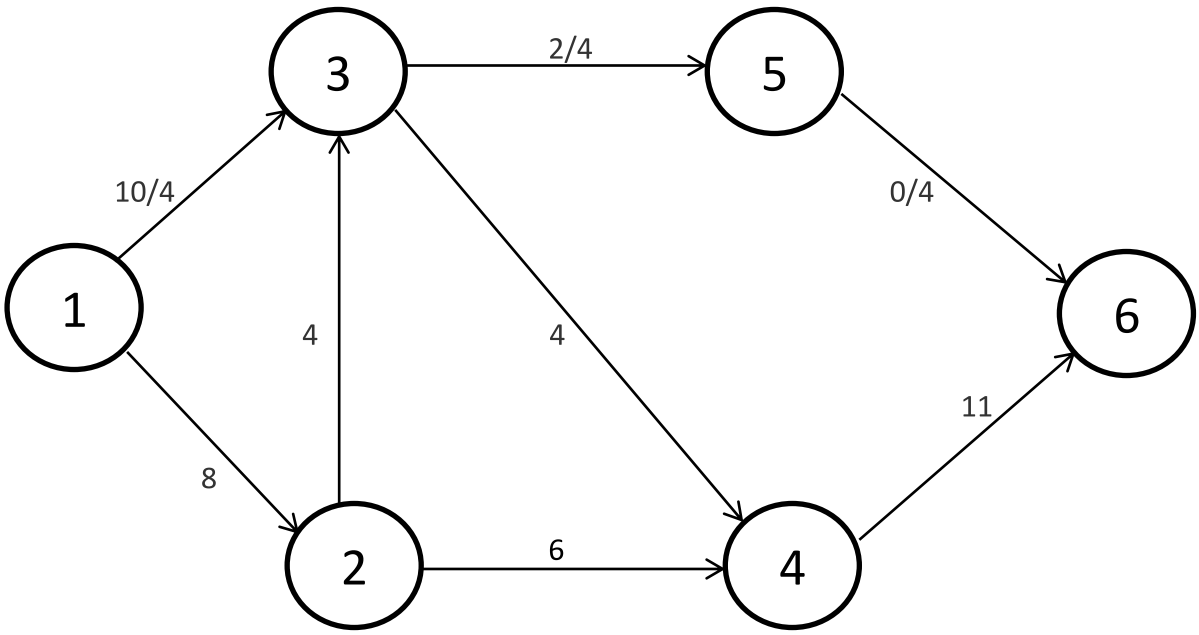
Ένα δίκτυο αγωγών χρησιμοποιείται για τη μεταφορά υγρού μεταξύ 6 δεξαμενών. Προέκυψε η ανάγκη να μεταφερθούν τα περιεχόμενα της δεξαμενής 1 στη δεξαμενή 6 όσο το δυνατόν πιο γρήγορα. Αν κάθε αγωγός έχει μια μέγιστη παροχή όπως αναγράφεται στο παρακάτω σχήμα, ποια θα είναι η μέγιστη ροή υγρού από τη δεξαμενή 1 στη δεξαμενή 6;



ΛΥΣΗ:

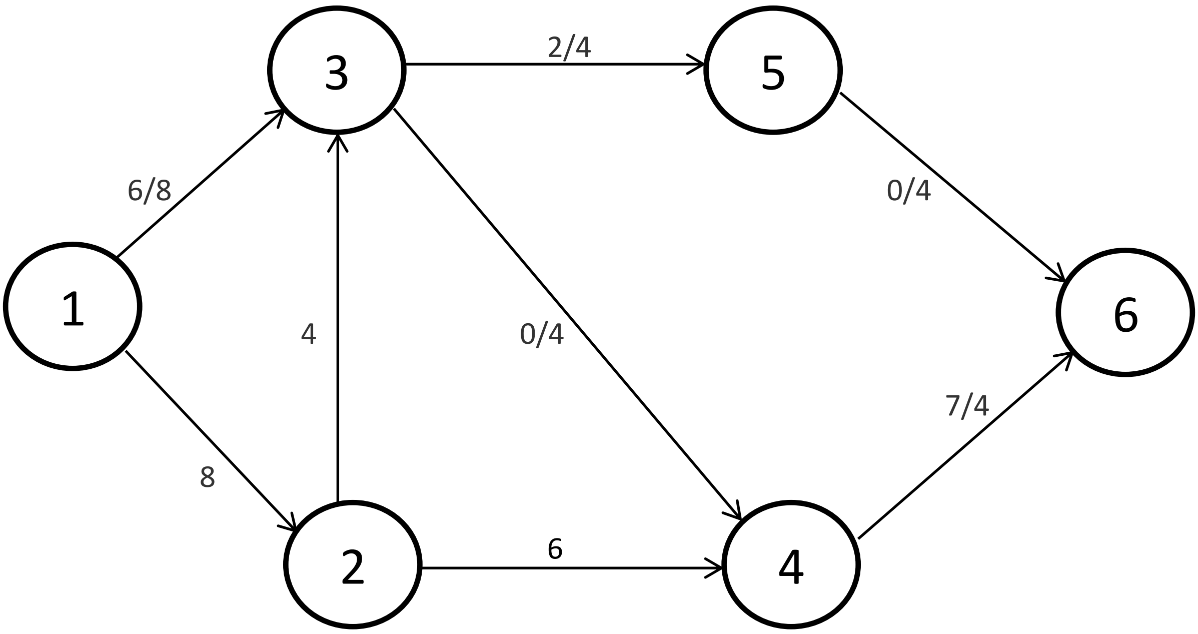
Βήμα 1

Επιλέγουμε αυθαίρετα ένα μονοπάτι με θετική ροή από την πηγή προς το δέκτη. Ένα τέτοιο μονοπάτι είναι για παράδειγμα το μονοπάτι 1-3-5-6 . Η μέγιστη ροή του μονοπατιού αυτού είναι ίση με 4 μονάδες όπως καθορίζεται από την ακμή του με την μικρότερη χωρητικότητα ροής, δηλαδή την ακμή 5-6. Έτσι, στέλνουμε 4 μονάδες μέσω του μονοπατιού αυτού από την πηγή 1 προς το δέκτη 6 και αναπροσαρμόζουμε κατάλληλα τις ροές των ακμών που συμμετέχουν.



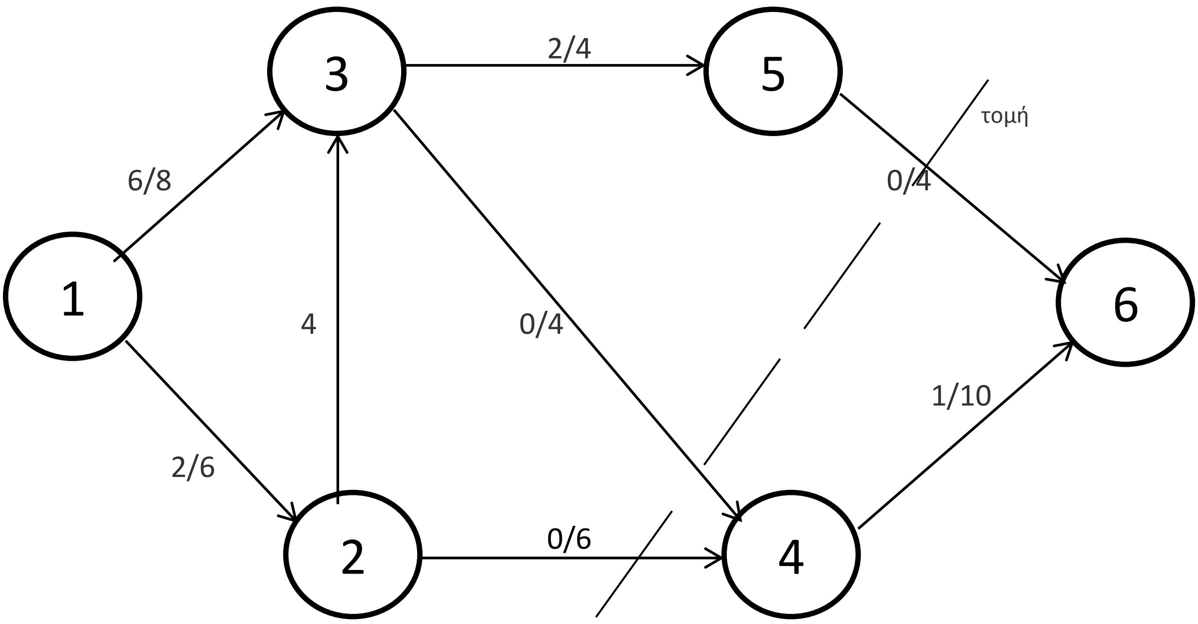
Βήμα 2

Συνεχίζουμε με το μονοπάτι (αυθαίρετα) 1-3-4-6 , που έχει θετική ροή χωρητικότητας από την πηγή προς το δέκτη. Η δυναμικότητα ροής του μονοπατιού αυτού είναι ίση με 4 μονάδες που καθορίζεται από την ακμή 3-4 που έχει την ελάχιστη χωρητικότητα ροής ανάμεσα στις ακμές του μονοπατιού. Έτσι, στέλνουμε 4 μονάδες από το μονοπάτι αυτό και αναπροσαρμόζουμε τις ροές των ακμών.



Βήμα 3

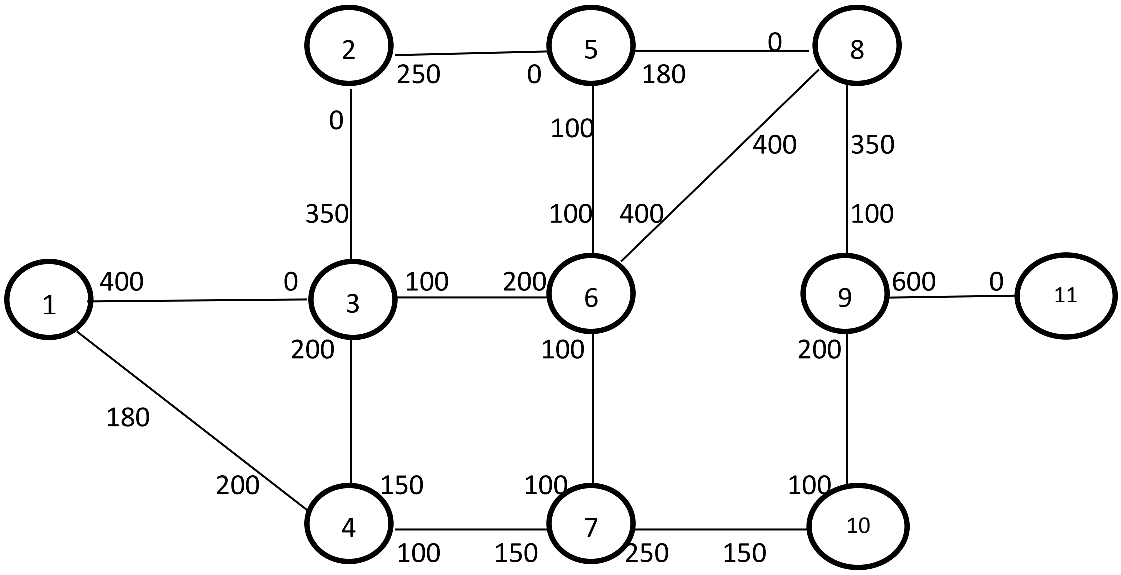
Επιλέγουμε το μονοπάτι 1-2-4-6.Η μέγιστη δυνατή ροή που μπορεί να περάσει από το συγκεκριμένο μονοπάτι είναι ίση με 6 και προκύπτει από την ακμή 2-4. Αφαιρούμε το 6 και έχουμε τα εξής:



Σε αυτό το σημείο δεν υπάρχει άλλη διαδρομή από το κόμβο 1 στο κόμβο 6 με θετική ροή. Η μέγιστη ροή είναι 10+4 = 14 και είναι ίση με την χωρητικότητα της τομής που περνά από τους τρεις κορεσμένους κλάδους (2,4), (3,4) & (5,6): 6+4+4 = 14.

ΠΑΡΑΔΕΙΓΜΑ 2

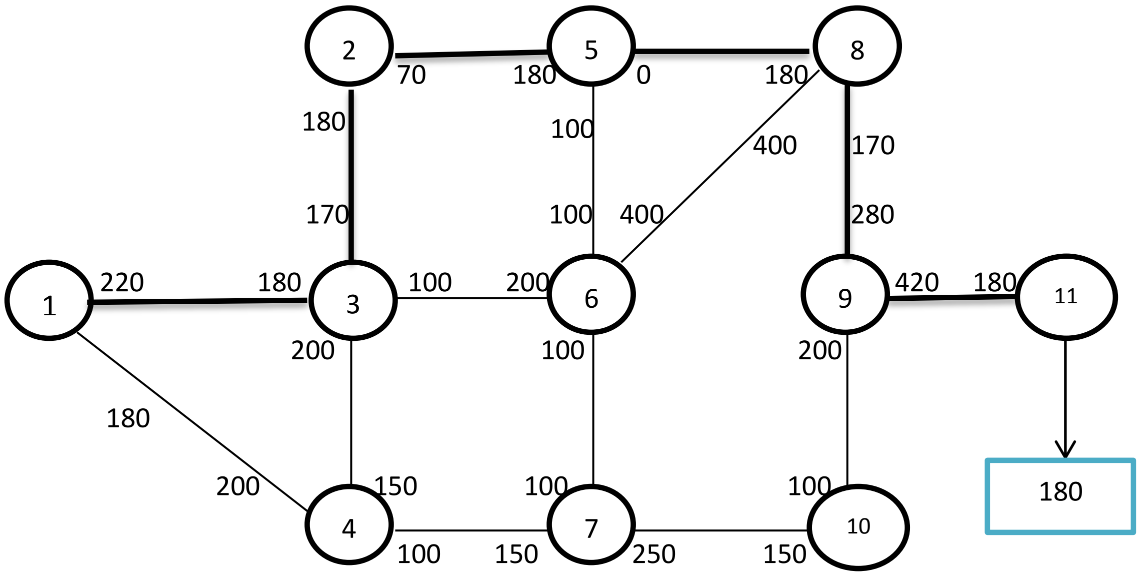
Στο επόμενο σχήμα, ο κόμβος 1 παριστάνει έναν εξυπηρετητή (server) ενός δικτύου υπολογιστών ο οποίος αποστέλλει δεδομένα προς κάποιο άλλο server, τον κόμβο 11, που παριστάνει τον τελικό αποδέκτη. Τα δεδομένα αυτά περνούν διαμέσου ενδιάμεσων servers που παρεμβάλλονται στο δίκτυο και παριστάνονται από τους υπόλοιπους κόμβους. Οι αριθμοί πάνω σε κάθε ακμή παριστάνουν τα Megabytes τα οποία μπορούν να σταλούν από τον αντίστοιχο κόμβο προς το γειτονικό του. Επιθυμούμε να υπολογιστεί τη μέγιστη ποσότητα Megabytes που είναι δυνατό να μεταφερθεί από τον κόμβο 1 στην πηγή 11.



ΛΎΣΗ:

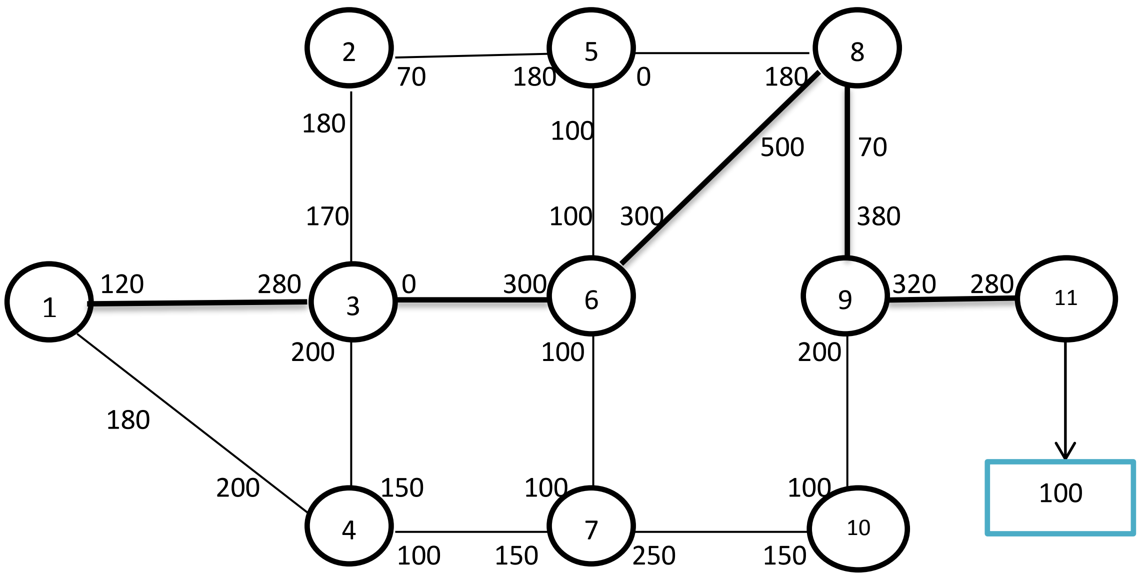
Βήμα 1

Επιλέγουμε τυχαία ένα μονοπάτι το οποίο είναι το 1-3-2-5-8-9-11 . Η μέγιστη δυνατή ροή που μπορεί να περάσει από το μονοπάτι είναι 180 Megabytes και καθορίζεται από την ακμή 5-8.Αφαιρούμε την ποσότητα αυτή από τις υπόλοιπες ακμές προς την κατεύθυνση του δέκτη και την προσθέτουμε στις ακμές που έχουν κατεύθυνση την πηγή. Στο παρακάτω σχήμα σημειώνεται το μονοπάτι με έντονες τις αντίστοιχες ακμές



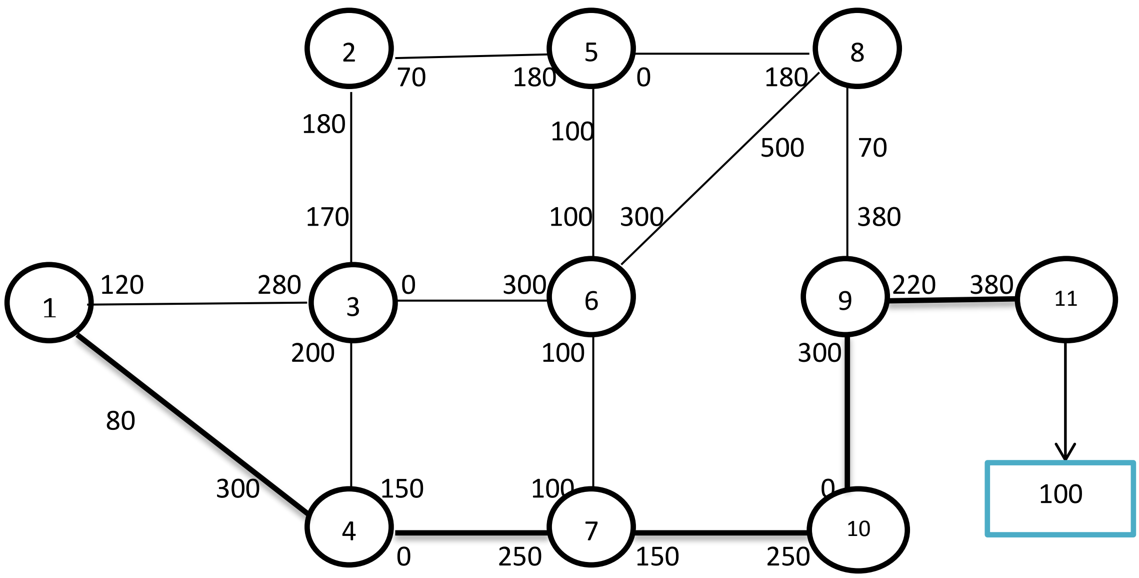
Βήμα 2

Επιλέγουμε το τυχαίο μονοπάτι 1-3-6-8-9-11. Η μέγιστη δυνατή ροη που μπορεί να περάσει από το μονοπάτι είναι 100 Megabytes και καθορίζεται από την ακμή 3-6 και σημειώνεται με βέλος στον κόμβο 11 στο επόμενο σχήμα. Ακόμη, αναπροσαρμόζουμε τις χωρητικότητες των ακμών αφαιρώντας ή προσθέτοντας την ποσότητα αυτή στις αντίστοιχες χωρητικότητες κάθε ακμής.



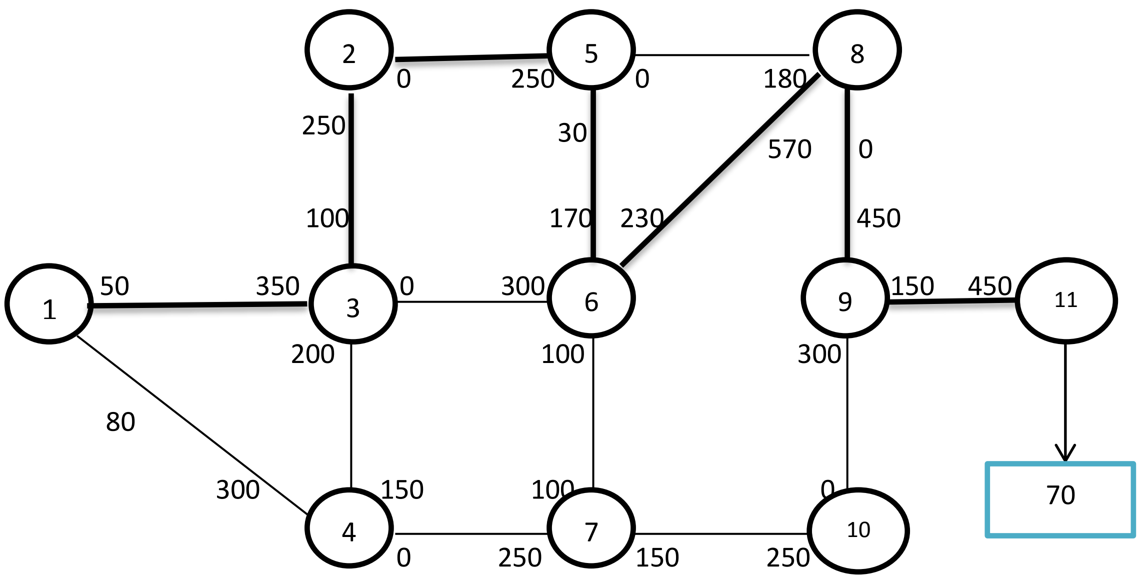
Βήμα 3

Στη συνέχεια επιλέγουμε το μονοπάτι 1-4-7-10-9-11.Η μέγιστη χωρητικότητα είναι 100 Megabytes όπως ορίζεται από τις ακμές 4-7 και 10-9. Επαναλαμβάνουμε την ίδια διαδικασία , τα αποτελέσματα της οποίας απεικονίζονται στο επόμενο σχήμα.



Βήμα 4

Επιλέγουμε το μονοπάτι 1-3-2-5-6-8-9-11 η μέγιστη χωρητικότητά του είναι 70 Megabytes όπως προκύπτει από τις ακμές 2-5 και 8-9.Στο παρακάτω σχήμα βλέπουμε τα αποτελέσματα του 4ου βήματος.



Παρατηρούμε ότι στο παραπάνω σχήμα δεν υπάρχει άλλο μονοπάτι με θετική χωρητικότητα ροής. Επομένως τελειώνει η εφαρμογή της μεθόδου μέγιστης ροής. Η συνολική μέγιστη ροή που μπορεί να περάσει διαμέσου του δικτύου από την πηγή προς τον δέκτη είναι ίση με το άθροισμα των χωρητικοτήτων των μονοπατιών που βρήκαμε και ανέρχεται σε 450 Megabytes .Στον παρακάτω πίνακα δίνουμε συνοπτικά τα αποτελέσματα της άριστης λύσης , δηλαδή τα μονοπάτια που πρέπει να χρησιμοποιηθούν και τις αντίστοιχες ροές τους , ώστε να μεγιστοποιηθεί η ροή των δεδομένων από την πηγή προς τον δέκτη.

Μονοπάτι	Ροή
1-3-2-5-8-9-11	180
1-3-6-8-9-11	100
1-4-7-10-9-11	100
1-3-2-5-6-8-9-11	70
Σύνολο	450

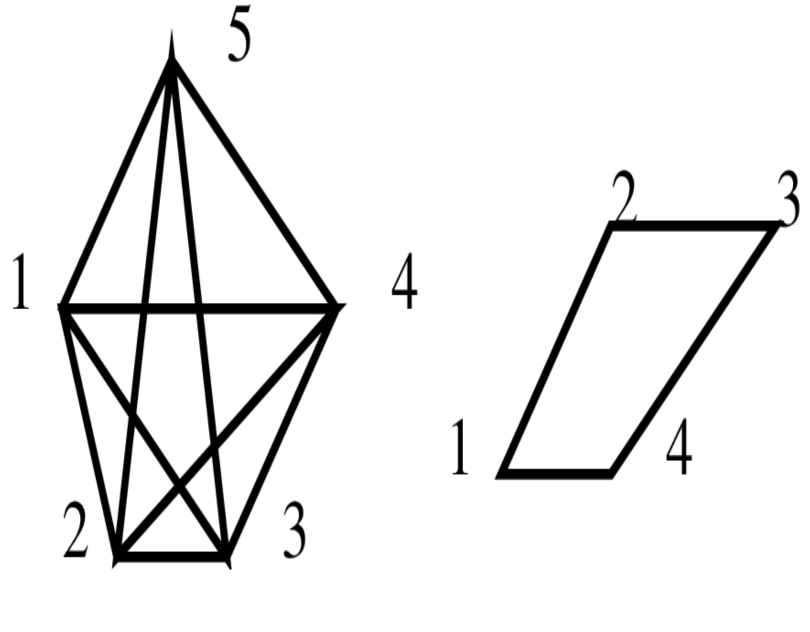
ΚΕΦΑΛΑΙΟ 5

Ελάχιστα Παραγόμενα Δέντρα (Minimum Spanning Trees)

5.1 Εισαγωγή-Ορισμοί εννοιών

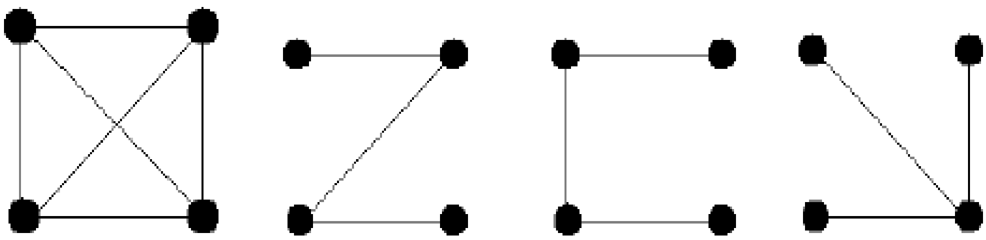
Στο κεφάλαιο αυτό παρουσιάζεται το πρόβλημα του ελάχιστου παραγόμενου δέντρου. Αρχικά ορίζουμε ως υπογράφημα του G(V,E) να είναι ένα γράφημα G'(V',E') με V'⊆ V και E'⊆E. Θυμίζουμε ότι σε ένα συνδεδεμένο, μη-κατευθυνόμενο γράφημα, ένα παραγόμενο (ζευγνύον) δέντρο του γραφήματος είναι ένα υπογράφημα που συνδέει όλους τους κόμβους του δικτύου.

Σχήμα 5.1: Το δεξί γράφημα είναι υπογράφημα του αριστερού



Ένα γράφημα ενδέχεται να περιέχει πολλά παραγόμενα δέντρα. Παράλληλα, έχουμε τη δυνατότητα να ορίσουμε ένα βάρος σε κάθε τόξο, όπου πρόκειται ουσιαστικά για ένα αριθμό που αντιπροσωπεύει το βαθμό δυσμένειας και να χρησιμοποιήσουμε αυτό τον αριθμό προκειμένου να ορίσουμε ένα βάρος σε ένα παραγόμενο δέντρο, αθροίζοντας τα βάρη των τόξων σε αυτό το παραγόμενο δέντρο. Ορίζουμε ως ελάχιστο παραγόμενο (ζευγνύον) δέντρο ή δέντρο ελαχίστου βάρους, το παραγόμενο δέντρο με βάρος μικρότερο ή ίσο από το βάρος όλων των άλλων παραγόμενων δέντρων. Με άλλα λόγια, το πρόβλημα έγκειται στο να βρεθεί το συνεκτικό δένδρο με το ελάχιστο συνολικό μήκος. Γενικά ισχύει ότι κάθε μη-κατευθυνόμενο γράφημα (που δεν είναι απαραίτητα συνδεδεμένο) εμπεριέχει ένα ελάχιστα παραγόμενο δάσος, το οποίο είναι η ένωση των ελάχιστα παραγόμενων δέντρων για τις συνεκτικές ενώσεις του.

Στην ουσία, ένα παραγόμενο δέντρο T του G είναι ένα συνδεδεμένο ακυκλικό υπογράφημα που εκτείνεται σε όλους τους κόμβους. Το ελάχιστα παραγόμενο δέντρο ενός μη-κατευθυνόμενου γραφήματος G=(N,A) με n=|N| κόμβους και m=|A| τόξα και με μήκος ή κόστος $c_{i,j}$ που συνδέεται με κάθε τόξο (i,j) ∈ A, είναι το ελάχιστο συνολικό κόστος (ή μήκος) των τόξων του και μετράται ως το σύνολο των κοστών των τόξων στο παραγόμενο δέντρο. Ακόμη, κάθε παραγόμενο δέντρο του G έχει το πολύ $n - 1$ ακμές.



Εικόνα: Ένας γράφος (αριστερά) και κάποια από τα πιθανά παραγόμενα υποδέντρα του (δεξιά).

Το πρόβλημα του ελάχιστα παραγόμενου δέντρου ανακύπτει σε διάφορες εφαρμογές, είτε ως μεμονωμένο πρόβλημα, είτε ως επιμέρους υποπρόβλημα σε πιο περίπλοκα προβλήματα. Προκειμένου να εκτιμήσουμε, αν ένα δεδομένο δέντρο είναι ένα ελάχιστο παραγόμενο δέντρο, θα πρέπει να θεωρήσουμε δύο βασικές συνθήκες βελτιστοποίησης.

- Η **πρώτη** βασίζεται στη σύγκριση του κόστους κάθε τόξου που ανήκει στο δέντρο με τα υπόλοιπα τόξα που περιέχονται στην αποκοπή που ορίζεται αν αφαιρέσουμε αυτό το τόξο από το δέντρο.

- Η **δεύτερη** συνθήκη βασίζεται στη σύγκριση του κόστους ενός τόξου που δεν ανήκει στο δέντρο με τα τόξα που ανήκουν στο δέντρο, στο μονοπάτι που συνδέει τα καταληκτικά σημεία του τόξου που δεν ανήκει στο δέντρο.

Αυτές οι δύο συνθήκες βελτιστοποίησης, της αποκοπής και του μονοπατιού, είναι εύκολες στη διατύπωση και την ανάπτυξή τους, και αποτελούν τον ακρογωνιαίο λίθο για την ανάπτυξη αλγορίθμων που επιλύουν το πρόβλημα του ελάχιστα παραγόμενου δέντρου.

Οι αλγόριθμοι που χρησιμοποιούνται για την εύρεση του ελάχιστου παραγόμενου δέντρου είναι σχετικά απλοί, αν κι η αποτελεσματική υλοποίηση τους προϋποθέτει σημαντικό βαθμό προσοχής. Στο κεφάλαιο αυτό θα αναπτύξουμε δύο τέτοιους αλγορίθμους, εκείνον του *Kruskal* κι εκείνον του *Prim*. Οι δύο αλγόριθμοι είναι "άπληστοι" υπό την έννοια ότι σε κάθε στάδιο λειτουργίας τους παίρνουν μία απόφαση που οδηγεί στην επιλογή της καλύτερης δυνατής ακμής και στην εισαγωγή της στο ελάχιστο παραγόμενο δέντρο T του γραφήματος G. Ακόμη, και οι δύο αυτοί αλγόριθμοι διατηρούν ένα δάσος, που περιέχει ήδη προεπιλεγμένα τόξα και μετά προσθέτουν ένα ή περισσότερα τόξα για να μεγαλώσουν το μέγεθος του δάσους. Για τον αλγόριθμο του Kruskal, η υποψήφια λίστα είναι όλο το δίκτυο, ενώ για τον αλγόριθμο του Prim, το δάσος είναι ένα μεμονωμένο δέντρο συν ένα σύνολο απομονωμένων κόμβων και η υποψήφια λίστα του περιέχει όλα τα τόξα ανάμεσα στο δέντρο και στους κόμβους που δεν ανήκουν στο δέντρο.

Ομοιότητες και Διαφορές με το Πρόβλημα Ελάχιστης Διαδρομής

Το πρόβλημα του ελάχιστα παραγόμενου δέντρου παρουσιάζει κάποιες ομοιότητες με το πρόβλημα της ελάχιστης διαδρομής που παρουσιάστηκε στο κεφάλαιο 3. Για την ακρίβεια, η ομοιότητα τους έγκειται στο γεγονός πως και στις δυο περιπτώσεις, έχουμε ένα μη-κατευθυνόμενο και συνδεδεμένο δίκτυο όπου η δεδομένη πληροφορία περιλαμβάνει κάποια μέτρο θετικού μήκους (απόσταση, κόστος, χρόνος κ.λπ.) που σχετίζεται με κάθε σύνδεσμο. Επίσης, και τα δύο προβλήματα περιλαμβάνουν την επιλογή ενός συνόλου συνδέσμων που έχουν το μικρότερο συνολικό μήκος μεταξύ όλων των δεσμών που ικανοποιούν μια συγκεκριμένη ιδιότητα. Για το πρόβλημα της συντομότερης διαδρομής, αυτή η ιδιότητα είναι ότι οι επιλεγμένοι σύνδεσμοι πρέπει να παρέχουν μια διαδρομή μεταξύ της προέλευσης και του προορισμού. Για το πρόβλημα του ελάχιστα παραγόμενου δέντρου, η ιδιότητα είναι ότι οι επιλεγμένοι σύνδεσμοι πρέπει να παρέχουν μια διαδρομή μεταξύ κάθε ζεύγους κόμβων.

Όσο αφορά τις διαφορές τους, μπορούμε να εντοπίσουμε τις εξής δύο:

1. Στην περίπτωση του προβλήματος του ελάχιστα παραγόμενου δέντρου, τα τόξα είναι μη κατευθυνόμενα. (Εφόσον, το δίκτυο είναι μη κατευθυνόμενο, αναφερόμαστε στο τόξο ανάμεσα στο ζεύγος των κόμβων i και j ως είτε (i, j) είτε (j, i)). Αντίθετα, στην περίπτωση του προβλήματος της ελάχιστης διαδρομής, τα δίκτυα είναι κατευθυνόμενα. Αυτή η διάκριση είναι ασήμαντη κατά μία έννοια, εφόσον θα μπορούσαμε εύκολα να αναπτύσσαμε τα προηγούμενα αποτελέσματα για τα προβλήματα ελάχιστου μονοπατιού κάνοντας χρήση μη κατευθυνόμενων γραφημάτων, καθώς επίσης και κατευθυνόμενων γραφημάτων . Από την άλλη πλευρά όμως, αυτή η διάκριση είναι σημαντική, καθώς η εύρεση ενός ελάχιστα παραγόμενου δέντρου σε ένα κατευθυνόμενο δίκτυο με όλα τα μονοπάτια κατευθυνόμενα από ένα δεδομένο κόμβο-πηγή είναι ένα πολύ πιο δύσκολο και περίπλοκο πρόβλημα από εκείνο του μη κατευθυνόμενου ελάχιστα παραγόμενου δέντρου.

2. Οι αντικειμενικές συναρτήσεις του προβλήματος ελάχιστα παραγόμενου δέντρου και του ελάχιστου μονοπατιού που ανήκει σε δέντρο είναι διαφορετικές. Αναφορικά με το πρόβλημα του ελάχιστα παραγόμενου δέντρου, μετράμε το κόστος κάθε τόξου ακριβώς μία φορά, ενώ για το πρόβλημα της ελάχιστης διαδρομής που ανήκει σε δέντρο, μετράμε το κόστος κάποιων τόξων τόσες φόρες όσες και ο αριθμός των μονοπατιών από τον κόμβο-πηγή που περνάει από αυτό το τόξο (δηλαδή, τον αριθμό των ελάχιστων μονοπατιών στο δέντρο που περιέχουν αυτό το τόξο).

Μερικές Περιπτώσεις του προβλήματος

Στο σημείο αυτό παραθέτουμε μερικές χαρακτηριστικές περιπτώσεις που ανακύπτει το πρόβλημα του ελάχιστου παραγόμενου δέντρου:

1. Κατά το σχεδιασμό τηλεπικοινωνιακών δικτύων (δίκτυα οπτικών ινών, δίκτυα υπολογιστών, τηλεφωνικά δίκτυα μισθωμένων γραμμών, δίκτυα καλωδιακής τηλεόρασης κ.λπ.) όπου πρέπει να βρεθεί κάποιο δέντρο, που να συνδέει όλους τους κόμβους μεταξύ τους με κάποιο μονοπάτι με τον πιο οικονομικό τρόπο.

2. Κατά το σχεδιασμό ενός δικτύου μεταφορών που στοχεύει στην ελαχιστοποίηση του συνολικού κόστους παροχής των διασυνδέσεων (σιδηροδρομικές γραμμές, δρόμοι κ.λπ.).

3. Κατά το σχεδιασμό ενός δικτύου γραμμών μεταφοράς ηλεκτρικής ενέργειας υψηλής τάσης.

4.Κατά το σχεδιασμό ενός δικτύου καλωδίωσης ηλεκτρικού εξοπλισμού (π.χ. ψηφιακού συστήματος ηλεκτρονικών υπολογιστών) με στόχο να ελαχιστοποιηθεί το συνολικό μήκος του σύρματος.

5. Κατά το σχεδιασμός ενός δικτύου αγωγών για τη σύνδεση διάφορων θέσεων.

5.2 Ο αλγόριθμος Prim

Ο αλγόριθμος του Prim είναι ίσως ο απλούστερος αλγόριθμος για την εύρεση ενός ελάχιστου ζευγνύοντος δένδρου σε έναν συνδεδεμένο γράφο με βάρη. Ο αλγόριθμος αυτός πρωτοαναπτύχθηκε από τον μαθηματικό Vojtěch Jarník το 1930.Στη συνέχεια το 1957 ο μαθηματικός και πληροφορικός Robert C. Prim διατύπωσε εκ νέου τον αλγόριθμο, ο οποίος και αναδημοσιεύθηκε το 1959 από τον Edsger Dijkstra. Ως εκ τούτου, ο αλγόριθμος του Prim ονομάζεται και DJP αλγόριθμος ή ο αλγόριθμος του Jarník ή ο αλγόριθμος των Prim-Jarník.

Ο αλγόριθμος που θα παρουσιάσουμε χωρίζει τους κόμβους του δικτύου σε αυτούς που έχουν συνδεθεί στο δέντρο (συνδεδεμένους) και σε αυτούς που δεν έχουν συνδεθεί ακόμα (μη συνδεδεμένοι). Στην αρχή ξεκινώντας από έναν οποιοδήποτε κόμβο, επιλέγουμε από όλους τους κλάδους που συνδέουν αυτό τον κόμβο με κάποιο άλλο, εκείνον με τη μικρότερη απόσταση-κόστος , ανεξάρτητα από την επίδραση που μπορεί να έχει αυτή μας η απόφαση σε επόμενες αποφάσεις. Σε κάθε επόμενο στάδιο προσδιορίζουμε το μη συνδεμένο κόμβο, που είναι πλησιέστερα σε οποιοδήποτε από τους συνδεμένους κόμβους και στη συνέχεια προσθέτουμε τον αντίστοιχο κλάδο στο δέντρο. Η διαδικασία αυτή επαναλαμβάνεται, όπως παρουσιάζεται και παρακάτω, μέχρι να συνδεθούν όλοι οι κόμβοι. Το δίκτυο που προκύπτει καθίσταται βέβαιο ότι είναι ένα δέντρο ελάχιστης κάλυψης. Παρακάτω παρουσιάζονται συνοπτικά τα βήματα του αλγορίθμου.

<u>Ο αλγόριθμος Prim για την εύρεση του δέντρου ελάχιστης κάλυψης</u>

Βήμα 1 Επιλέγουμε τυχαία έναν οποιονδήποτε κόμβο του δικτύου για να ξεκινήσουμε. Αυτός ο κόμβος εισέρχεται πρώτος στο σύνολο των συνδεδεμένων κόμβων.

Βήμα 2 Συνδέουμε τον προηγούμενο κόμβο με αυτόν που βρίσκεται πιο κοντά του. Αυτός ο κόμβος εισέρχεται στο σύνολο των συνδεδεμένων.

Βήμα 3 Εντοπίζουμε τον κόμβο (μη συνδεδεμένο) που είναι πιο κοντά σε κάποιον από τους συνδεδεμένους κόμβους και τον συνδέουμε και αυτόν. Σε περίπτωση ισοβάθμισης επιλέγουμε τυχαία έναν από τους ισοβαθμούντες κόμβους. Πιθανότατα σε αυτή την περίπτωση υπάρχει κι άλλη λύση.

Βήμα 4 Επαναλαμβάνουμε το βήμα 3 μέχρις ότου να συνδέσουμε όλους τους κόμβους.

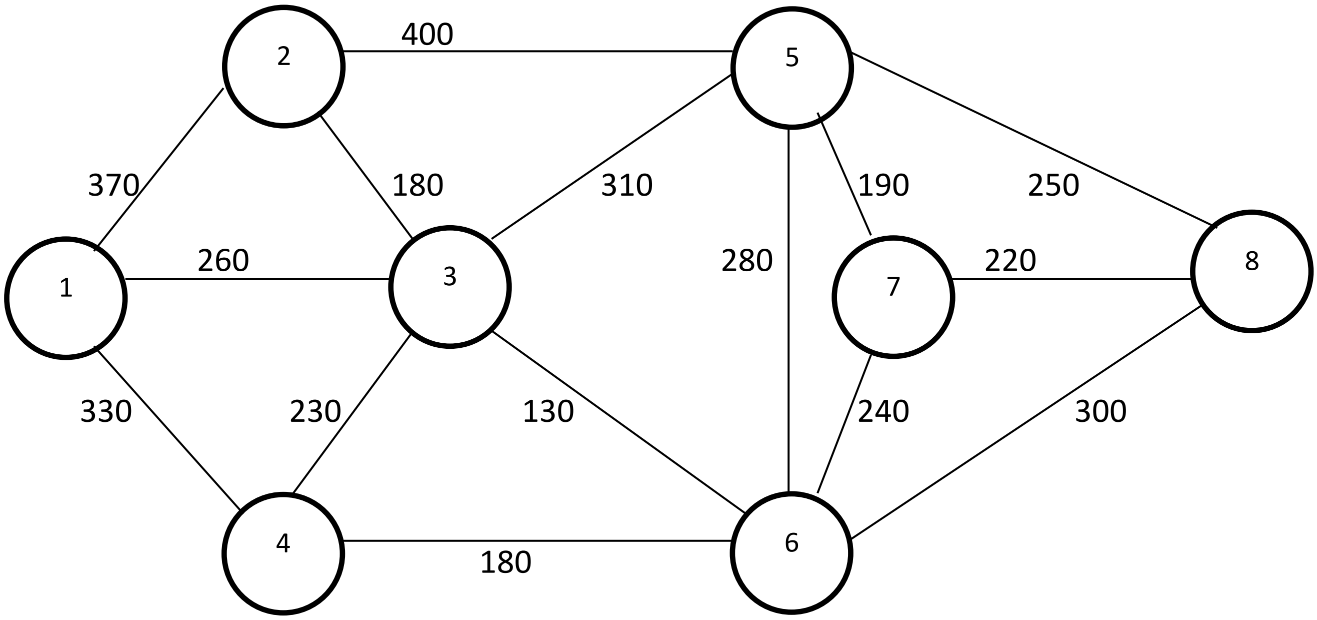
Για την πολυπλοκότητα του αλγόριθμου Prim , διαπιστώνουμε ότι εκτελούνται συνολικά n επαναλήψεις, αφού n είναι ο αριθμός των κόμβων που πρέπει να συνδεθούν συνολικά. Σε κάθε επανάληψη, πρέπει να βρούμε έναν κλάδο ελάχιστου κόστους του οποίου ο ένας κόμβος ανήκει στο υπάρχον δέντρο (είναι συνδεδεμένος) και ο άλλος δεν ανήκει. Για να βρούμε αυτό αν εξετάζουμε όλους τους κλάδους που είναι συνολικά m , τότε προκύπτει πως η πολυπλοκότητα μιας επανάληψης είναι O(m), οπότε η πολυπλοκότητα του αλγόριθμου είναι O(mn). Με κάποιες κατάλληλες βελτιώσεις, η πολυπλοκότητα του αλγόριθμου μπορεί να βελτιωθεί ακόμα περισσότερο.

5.2.1 Παραδείγματα του αλγορίθμου Prim

Παράδειγμα 1

Μια ξενοδοχειακή μονάδα μελετά την δημιουργία χώρων αναψυχής για τους πελάτες του. Στο σχήμα 5.2 περιγράφονται οι αποστάσεις σε μέτρα μεταξύ των 8 χώρων. Ζητείται να καθορίσουμε τον τρόπο με τον οποίο όλοι οι χώροι αναψυχής θα συνδεθούν μεταξύ τους,
άμεσα ή έμμεσα, με το μικρότερο δυνατό κόστος και να υπολογίσετε το κόστος αυτό.

Σχήμα 5.2



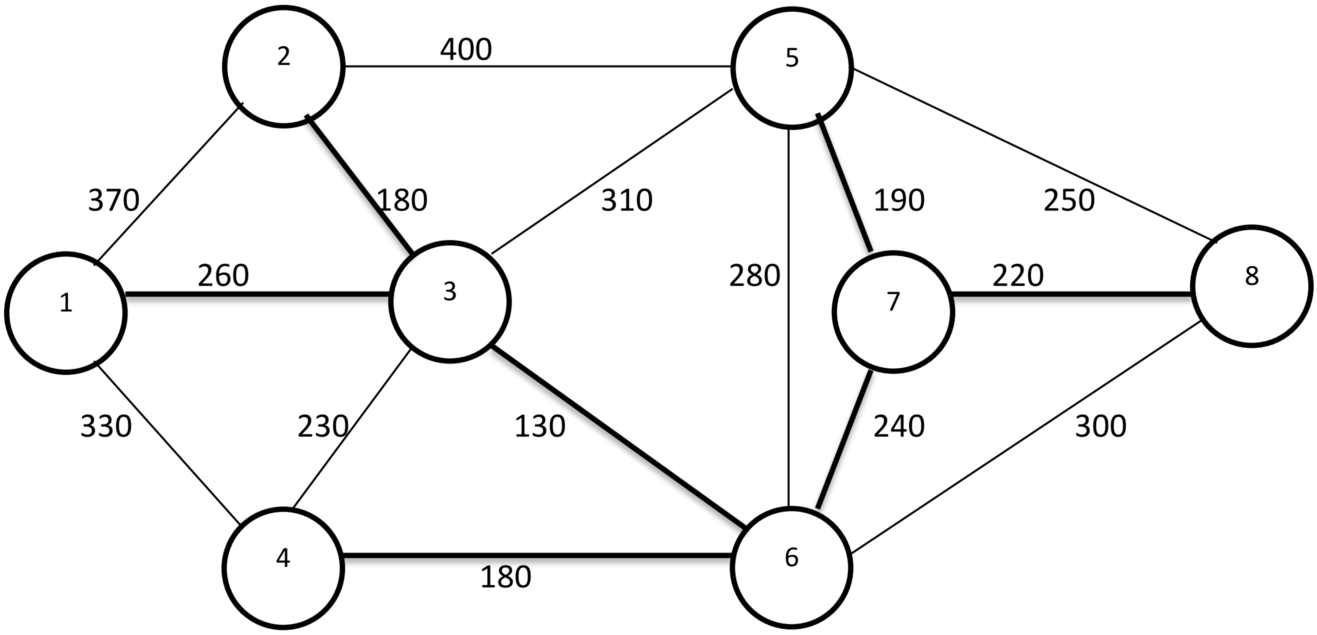
Λύση:

Στο βήμα 1 του αλγορίθμου επιλέγουμε τυχαία έναν οποιονδήποτε κόμβο του σχήματος. Έστω ότι επιλέγουμε τον κόμβο 1 για να ξεκινήσουμε. Ο κόμβος 1 είναι ο πρώτος συνδεδεμένος κόμβος και ο πιο κοντινός του είναι ο κόμβος 3 όπου απέχει 260 μέτρα με άμεση σύνδεση. Σημειώνουμε τη σύνδεσή τους με έντονη γραμμή. Το σύνολο των συνδεδεμένων κόμβων περιέχει στο σημείο αυτό τους κόμβους 1 και 3. Ο πλησιέστερος κόμβος που συνδέεται άμεσα με κάποιον από τους κόμβους 1 και 3 είναι ο κόμβος 6 με απόσταση 130 μέτρα από τον κόμβο 3. Σημειώνουμε τη σύνδεση των κόμβων 3 και 6 με έντονη γραμμή και ο κόμβος 6 εισέρχεται κι αυτός στο σύνολο των συνδεδεμένων κόμβων. Στη συνέχεια, παρατηρούμε ότι στους κόμβους 1,3 και 6 ο πιο κοντινός κόμβος είναι ο 2 που συνδέεται με τον 3 με απόσταση 180 μέτρα, ωστόσο και ο κόμβος 4 με την ακμή 6-4 απέχει επίσης 180 μέτρα. Σε αυτή την περίπτωση έχουμε ισοβάθμιση και όπως είδαμε παραπάνω μπορούμε να επιλέξουμε τυχαία έναν από τους ισοβαθμούντες κόμβους. Έστω ότι επιλέγουμε την ακμή 3-2 και σημειώνουμε με έντονη γραμμή τη σύνδεση των 2 κόμβων. Συνεχίζοντας συνδέουμε τον κόμβο 4 με τον κόμβο 6 επειδή ο 4 είναι ο πιο κοντινός κόμβος στους ήδη συνδεδεμένους κόμβους.

Έχουμε συνδέσει τους κόμβους 1,3,6,2 και 4. Ο αμέσως επόμενος κόμβος που θα συνδέσουμε είναι ο 7 με τον 6 όπου απέχουν 240 μέτρα γιατί είναι πιο κοντά στους συνδεδεμένους κόμβους που αναφέραμε. Σημειώνουμε με έντονη γραμμή την ακμή

6-7. Έπειτα, συνδέουμε τον κόμβο 5 με τον κόμβο 7 σημειώνοντας με έντονη γραμμή την ακμή 7-5 μήκους 190 μέτρων. Τέλος , συνδέουμε τον κόμβο 8 με τον 7 κόμβο και σημειώνουμε με έντονη γραμμή την ακμή 7-8 μήκους 220 μέτρων.

Σχήμα 5.3



Το συνολικό μήκος των ακμών του δέντρου που σχηματίσαμε με έντονες γραμμές στο παραπάνω σχήμα 5.3 είναι 1400 μέτρα. Το μήκος αυτό είναι το ελάχιστο συνολικό μήκος που απαιτείται ώστε να επικοινωνούν άμεσα ή έμμεσα όλοι οι κόμβοι μεταξύ τους.

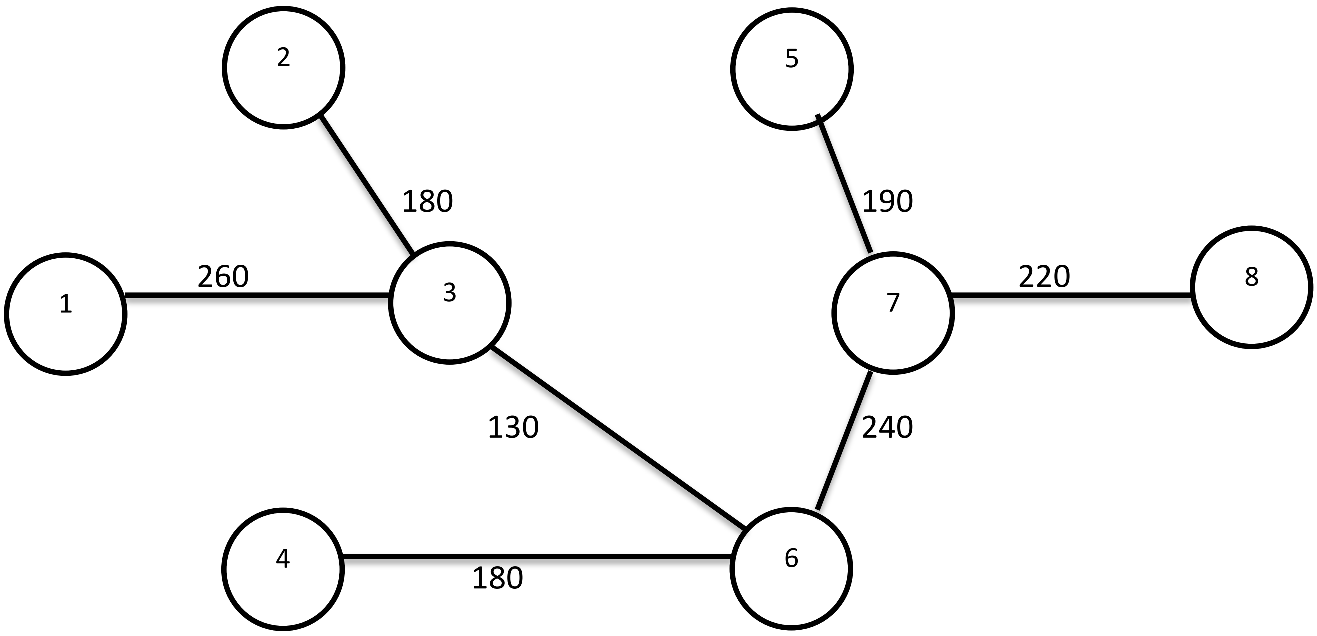
Στον επόμενο πίνακα 5.4 βλέπουμε τις ακμές που χρησιμοποιήθηκαν εφαρμόζοντας τον αλγόριθμο Prim καθώς επίσης και τα αντίστοιχα μήκη τους.

Πίνακας 5.4

Ακμή	1-3	3-6	3-2	6-4	6-7	7-5	7-8
Μήκος	260	130	180	180	240	190	220

Στο σχήμα 5.5 δίνουμε το ελάχιστο παραγόμενο (ζευγνύον) δέντρο που προκύπτει μετά την εφαρμογή του αλγορίθμου. Το υπογράφημα αυτό είναι ένα δέντρο με ακμές 7 , όσοι δηλαδή είναι οι κόμβοι του αρχικού γραφήματος μείον ένα (8-1=7) .

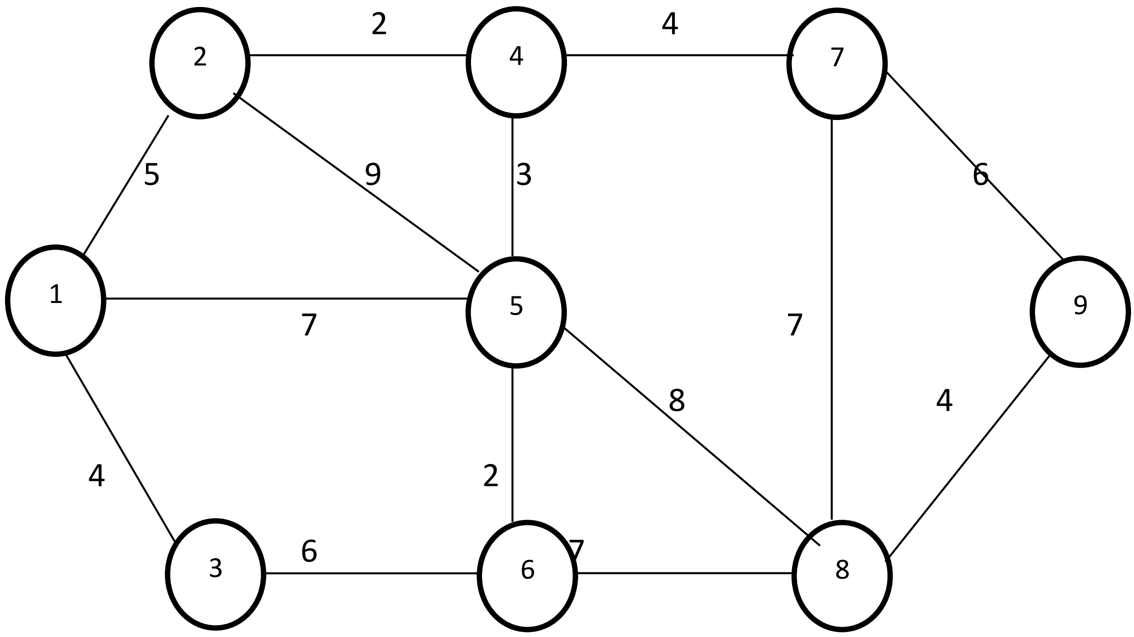
Σχήμα 5.5



Παράδειγμα 2

Μία Τράπεζα πρόκειται να συνδέσει τους υπολογιστές όλων των υποκαταστημάτων της σε δίκτυο, ώστε κάθε υποκατάστημα να μπορεί να επικοινωνήσει με οποιοδήποτε άλλο, είτε απευθείας είτε μέσω άλλων υποκαταστημάτων. Τα υποκαταστήματα της Τράπεζας και οι πιθανές συνδέσεις μεταξύ τους παριστάνονται με το ακόλουθο δίκτυο σχήμα 5.6 , στο οποίο οι τιμές των ακμών παριστάνουν μήκος καλωδίου σύνδεσης σε δεκάδες χιλιόμετρα. Το κόστος εγκατάστασης και λειτουργίας του δικτύου είναι ανάλογο του μήκους του καλωδίου που θα χρησιμοποιηθεί. Να προσδιορίσετε τον τρόπο με τον οποίο όλα τα υποκαταστήματα θα συνδεθούν μεταξύ τους, άμεσα ή έμμεσα, με το μικρότερο δυνατό κόστος και να υπολογίσετε το κόστος αυτό.

Σχήμα 5.6

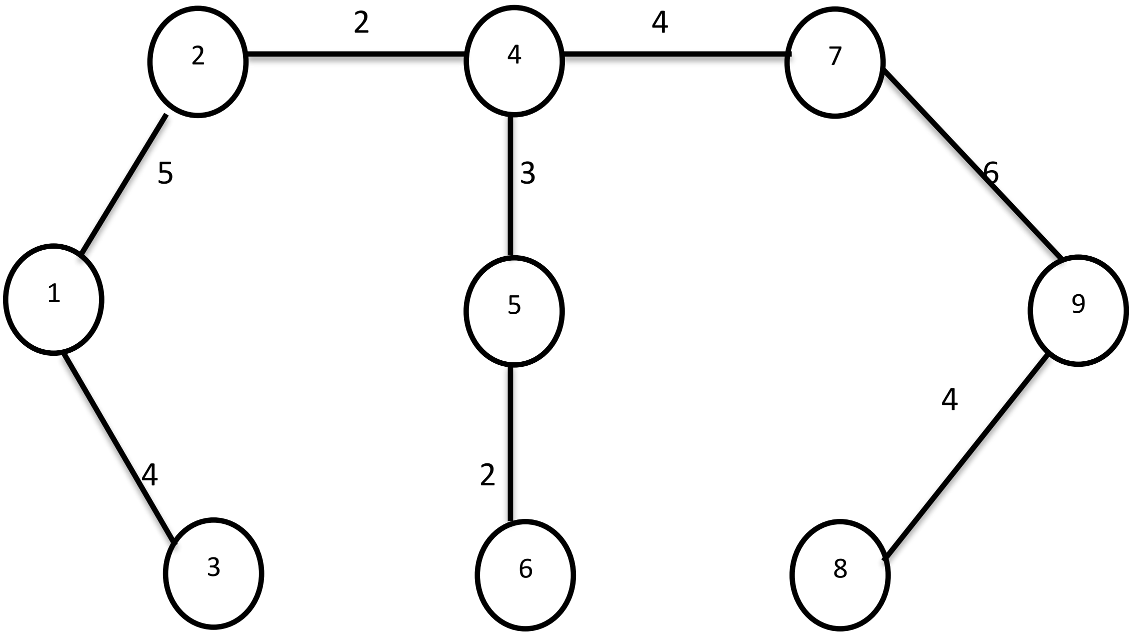


Λύση:

Ξεκινάμε αυθαίρετα από οποιοδήποτε κόμβο, έστω τον κόμβο 5. Συνδέουμε τον πλέον κοντινό του, που είναι ο κόμβος 6, μέσω της ακμής 5-6 με μήκος 2. Οι κόμβοι {5, 6} είναι συνδεδεμένοι. Ο πλησιέστερος μη συνδεδεμένος κόμβος στους {5, 6} είναι ο κόμβος 4 με την ακμή 5-4 μήκους 3. Με αυτό τον τρόπο, συνδεδεμένοι είναι τώρα οι κόμβοι {5, 6, 4}. Ο επόμενος πλησιέστερος κόμβος είναι ο κόμβος 2 με την ακμή 4-2 μήκους 2, οπότε συνδεδεμένοι είναι τώρα οι κόμβοι {5, 6, 4, 2}. Ο πλησιέστερος στους συνδεδεμένους είναι ο κόμβος 7 με την ακμή 4-7 μήκους 4. Το σύνολο των συνδεδεμένων κόμβων είναι τώρα {5, 6, 4, 2, 7}. Επόμενος συνδέεται ο κόμβος 1 με τον κόμβο 2 μέσω της ακμής 2-1 μήκους 5, οπότε το σύνολο των συνδεδεμένων κόμβων είναι το {5, 6, 4, 2, 7, 1}. Ο αμέσως επόμενος κόμβος που συνδέεται είναι ο κόμβος 3 με τον κόμβο 1 μέσω της ακμής 1-3 με μήκος 4. Το σύνολο των συνδεδεμένων κόμβων γίνεται {5, 6, 4, 2, 7, 1, 3}. Στη συνέχεια, συνδέεται ο κόμβος 9 με τον κόμβο 7 με την ακμή 7-9 μήκους 6, οπότε το σύνολο γίνεται {5, 6, 4, 2, 7, 1, 3, 9}. Τελευταίος συνδέεται ο κόμβος 8 με την ακμή 9-8 μήκους 4. Το άθροισμα των ακμών που χρησιμοποιήθηκαν είναι 30 και είναι το ελάχιστο συνολικό.

Κατά συνέπεια, το ελάχιστο ζευγνύον δέντρο φαίνεται στο ακόλουθο σχήμα 5.7 στο οποίο εμφανίζονται μόνο τις ακμές, ενώ όπως προαναφέρθηκε το ελάχιστο συνολικό μήκος καλωδίων είναι 300 χιλιόμετρα.

Σχήμα 5.7



5.3 Ο αλγόριθμος Kruskal

Ο αλγόριθμος του Kruskal είναι ένας απλός αλγόριθμος ο οποίος χρησιμοποιείται για την εύρεση ελάχιστων ζευγνυόντων δένδρων. Αυτό σημαίνει πως βρίσκει ένα υποσύνολο των ακμών του γράφου, έτσι ώστε να δημιουργήσει ένα δένδρο το οποίο περιέχει κάθε κόμβο, και το συνολικό κόστος όλων των ακμών του δένδρου ελαχιστοποιείται. Εάν ο γράφος δεν είναι συνδεδεμένος τότε βρίσκει το ελάχιστο ζευγνύον δάσος (ένα ελάχιστο ζευγνύον δένδρο για κάθε συνιστώσα του γράφου). Ο αλγόριθμος αυτός εμφανίστηκε για πρώτη φορά στο Proceedings of the American Mathematical Society, σελίδα 48–50 το 1956, και γράφτηκε από τον Joseph Kruskal.

Η βασική ιδέα του αλγορίθμου είναι ότι προσπαθεί να δημιουργήσει το ελάχιστο παραγόμενο (ζευγνύον) δέντρο σταδιακά, εξετάζοντας μία προς μία τις ακμές του γράφου κατά αύξουσα τιμή κόστους. Πιο συγκεκριμένα σε κάθε βήμα επιλέγεται η ακμή με το μικρότερο κόστος και της οποίας η προσθήκη στις ήδη επιλεγμένες ακμές δεν δημιουργεί κύκλο.

Παρακάτω παρουσιάζονται συνοπτικά τα βήματα του αλγορίθμου.

Ο αλγόριθμος Kruskal για την εύρεση του δέντρου ελάχιστης κάλυψης

Βήμα 1 Επιλέγουμε τον κλάδο με το μικρότερο κόστος (βάρος) και τον προσθέτουμε στο δέντρο.

Βήμα 2 Βρίσκουμε τον επόμενο κλάδο με το μικρότερο κόστος. Στη συνέχεια τον προσθέτουμε τον κλάδο αυτόν στο δέντρο υπό την προϋπόθεση ότι με αυτή την προσθήκη δεν προκύπτει κάποιος κύκλος.

Βήμα 3 Επαναλαμβάνουμε το βήμα 2 μέχρι να συνδεθούν όλοι οι κόμβοι.

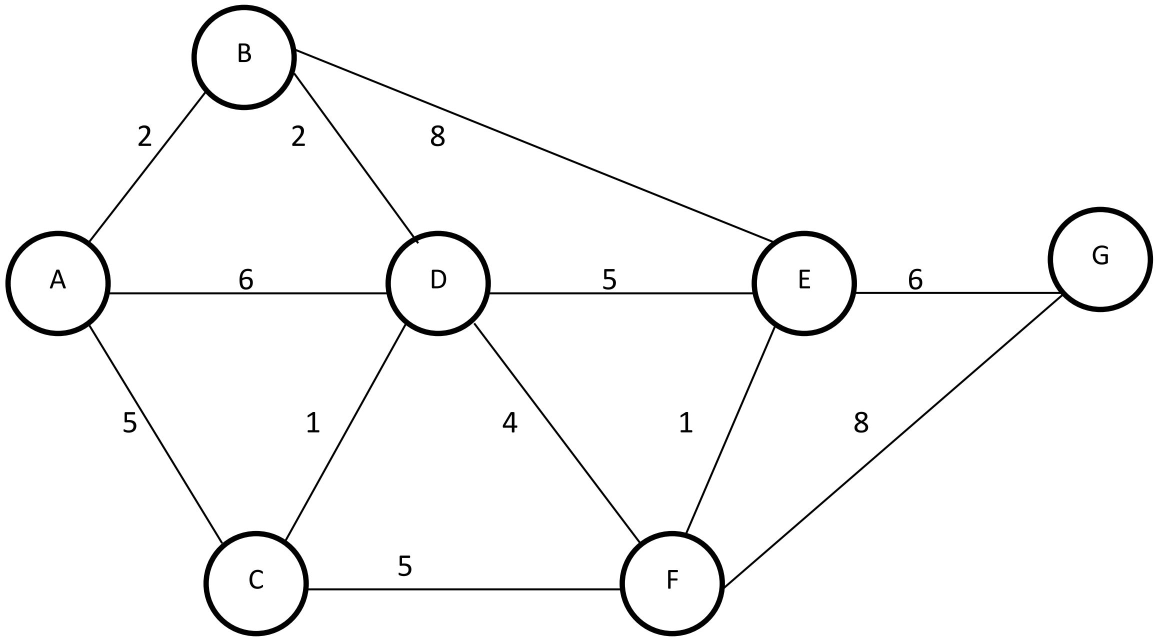
Όπως και στον προηγούμενο αλγόριθμο του Prim, έτσι και εδώ σε ενδεχόμενο ισοβάθμισης μπορούμε να επιλέξουμε αυθαίρετα ένα κόμβο και είναι εξασφαλισμένο ότι ο αλγόριθμος θα δώσει μια βέλτιστη λύση. Όμως, τέτοιες ισοβαθμίσεις είναι ένα σήμα ότι μπορεί να υπάρχουν (αλλά όχι απαραίτητα) πολλαπλές βέλτιστες λύσεις.

Για την πολυπλοκότητα του αλγόριθμο του Kruskal μπορούμε να θεωρήσουμε ότι αποτελείται από δύο φάσεις. Η πρώτη είναι η φάση ταξινόμησης των κλάδων κατά αύξον τιμή και η δεύτερη η φάση ελέγχου για κύκλους και προσθήκης των κλάδων. Η ταξινόμηση m κλάδων απαιτεί χρόνο O(mlogm). O έλεγχος και η προσθήκη (ή η απόρριψη ενός κλάδου) μπορεί να γίνει σε χρόνο O(n) με τη διαδικασία που περιγράφεται παραπάνω. Τέλος, στη χειρότερη περίπτωση εκτελούνται m επαναλήψεις, αφού αυτός είναι ο συνολικός αριθμός των κλάδων. Επομένως, η πολυπλοκότητα του αλγόριθμου είναι O(mlogm) + O(mn) = O(mn). Όπως και στην περίπτωση του αλγόριθμου Prim έτσι και στον αλγόριθμο Kruskal, με κατάλληλες βελτιώσεις, αυτή η πολυπλοκότητα μπορεί να βελτιωθεί.

5.3.1 Παραδείγματα του αλγορίθμου Kruskal

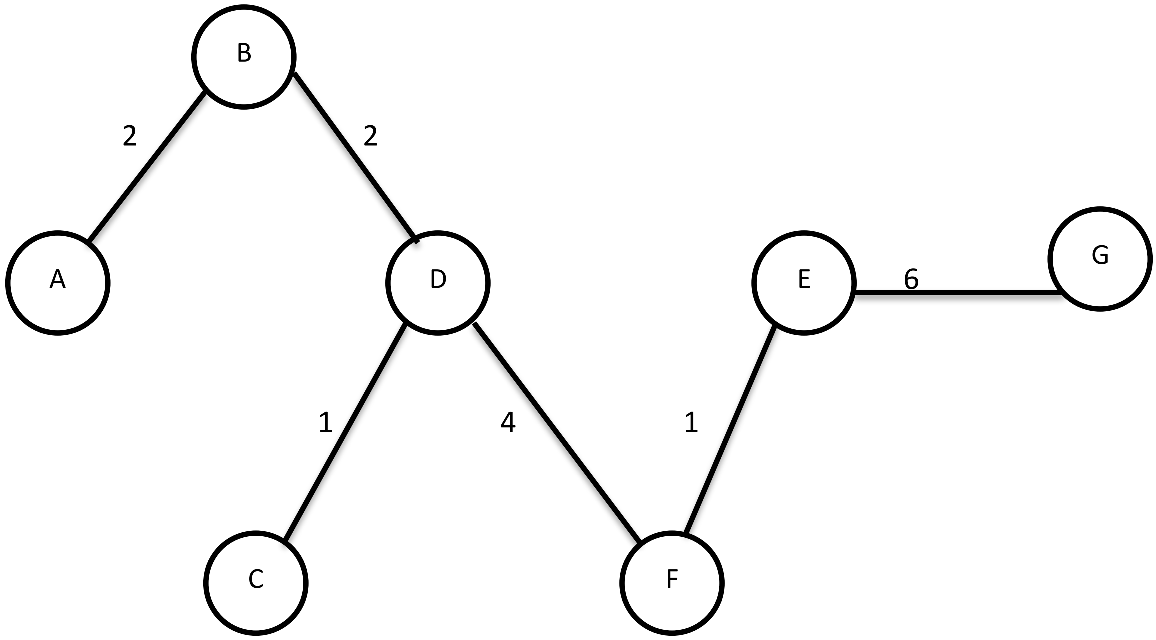
Παράδειγμα 1

Η διοίκηση ενός πάρκου θέλει να προσδιορίσει τους δρόμους κάτω από τους οποίους θα πρέπει να εγκατασταθούν οι τηλεφωνικές γραμμές, που θα συνδέουν όλους τους σταθμούς με το ελάχιστο συνολικό μήκος των γραμμών. Οι πιθανές συνδέσεις μεταξύ τους παριστάνονται με το ακόλουθο δίκτυο σχήμα 5.8 , στο οποίο οι τιμές των ακμών παριστάνουν μήκος καλωδίου σύνδεσης σε χιλιόμετρα.



<u>Λύση:</u>

Στο βήμα 1 του αλγορίθμου επειδή υπάρχουν δύο κλάδοι με ελάχιστο κόστος 1 χιλιόμετρο, ο CD και ο EF επιλέγουμε αυθαίρετα έναν εκ των δύο, έστω τον EF και τον προσθέτουμε σχηματίζοντας ένα αρχικό δέντρο. Ο επόμενος κλάδος είναι ο CD. Επειδή δε σχηματίζει κύκλο τον προσθέτουμε κι αυτόν. Στη συνέχεια υπάρχουν δύο κλάδοι που μπορούμε να επιλέξουμε με κόστος 2 χιλιόμετρα. Ο AB και ο BD και επιλέγουμε αυθαίρετα τον BD και τον προσθέτουμε , αφού αυτή η προσθήκη δεν σχηματίζει κύκλο. Στη συνέχεια, ο επόμενος κλάδος είναι ο AB με κόστος 2 χιλιόμετρα τον οποίον τον προσθέτουμε εφόσον δεν προκύπτει κύκλος. Ο επόμενος κλάδος είναι ο DF με κόστος 4 χιλιόμετρα. Επειδή δε σχηματίζεται κύκλος, προσθέτουμε και αυτόν τον κλάδο. Έπειτα, υπάρχουν τρείς κλάδοι με ελάχιστο κόστος ο AC , CF και DE. Επιλέγουμε αυθαίρετα έναν από τους τρείς κλάδους έστω τον AC. Επειδή σχηματίζεται κύκλος , ο κλάδος αυτός δεν προστίθεται. Από τους δύο κλάδους που απομένουν δηλαδή τον CF και DE επιλέγουμε αυθαίρετα έναν από τους δύο, έστω τον CF. Επειδή κι εδώ σχηματίζεται κύκλος, ούτε αυτός ο κλάδος προστίθεται. Συνεχίζοντας, ο επόμενος κλάδος είναι ο DE όπου και σε αυτή την περίπτωση δεν προστίθεται γιατί σχηματίζει κύκλο. Σε αυτό το βήμα υπάρχουν δύο εναλλακτικοί κλάδοι, ο AD και ο EG με κόστος 6 χιλιόμετρα. Επιλέγουμε αυθαίρετα έναν από τους δύο, έστω τον AD αλλά επειδή σχηματίζεται κύκλος δεν τον προσθέτουμε. Ο επόμενος κλάδος είναι ο EG και επειδή δεν σχηματίζει κύκλο προστίθεται κι αυτός. Η διαδικασία σταματά καθώς όλοι οι κόμβοι είναι τώρα συνδεδεμένοι. Η λύση που προκύπτει είναι η βέλτιστη και παριστάνεται στο σχήμα 5.9.

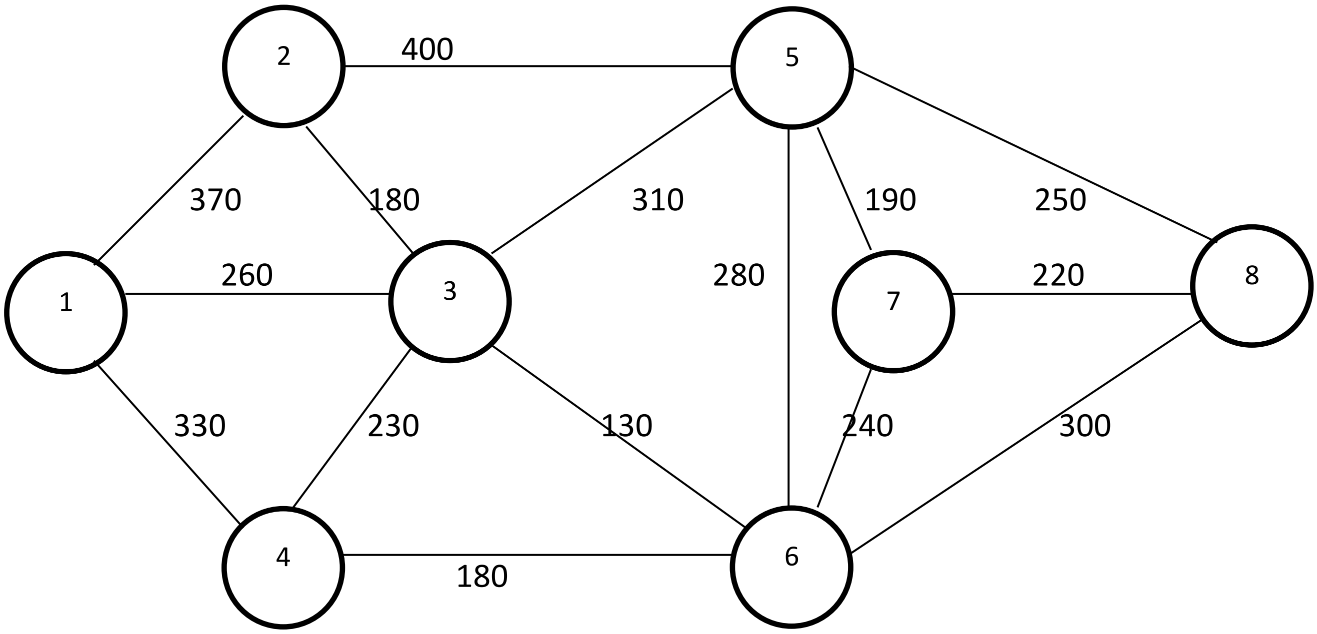


<u>Παράδειγμα 2</u>

Θα χρησιμοποιήσουμε το παράδειγμα 1 του αλγορίθμου Prim.

Μια ξενοδοχειακή μονάδα μελετά την δημιουργία χώρων αναψυχής για τους πελάτες του. Στο σχήμα 5.10 περιγράφονται οι αποστάσεις μεταξύ των 8 χώρων. Ζητείται να καθορίσουμε τον τρόπο με τον οποίο όλοι οι χώροι αναψυχής θα συνδεθούν μεταξύ τους,
άμεσα ή έμμεσα, με το μικρότερο δυνατό κόστος και να υπολογίσετε το κόστος αυτό.

Σχήμα 5.10



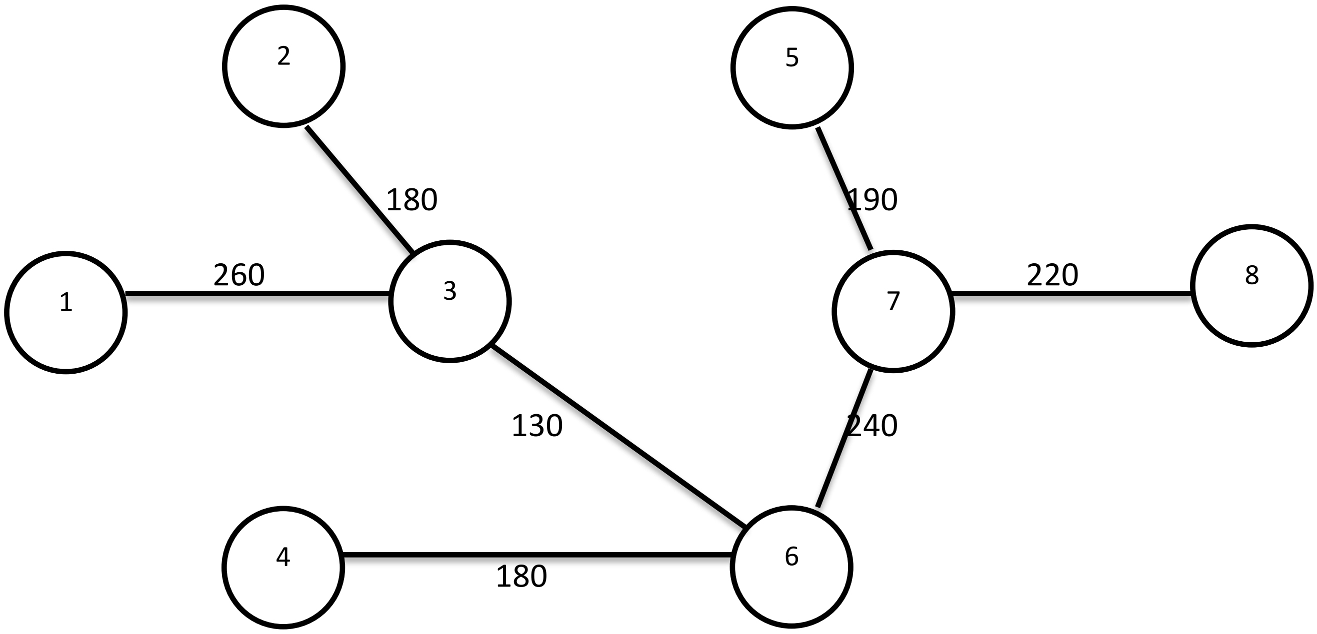
<u>Λύση:</u>

Στο βήμα 1 του αλγορίθμου ξεκινάμε από τον κλάδο 3-6 όπου έχει το ελάχιστο κόστος 130 μέτρα και προσθέτοντάς τον σχηματίζουμε ένα αρχικό δέντρο. Έπειτα ,μπορούμε να διαλέξουμε να είτε τον κλάδο 2-3 είτε τον κλάδο 4-6 με ελάχιστο κόστος 180 μέτρα. Διαλέγουμε αυθαίρετα τον 4-6 και τον προσθέτουμε κι αυτόν. Στη συνέχεια προσθέτουμε τον κλάδο 2-3 αφού αυτή η προσθήκη δεν σχηματίζει κύκλο. Συνεχίζοντας, ο επόμενος κλάδος είναι ο 5-7 με κόστος 190 μέτρα. Εφόσον δεν σχηματίζει κύκλο τον προσθέτουμε κι αυτόν. Στη συνέχεια ο επόμενος κλάδος είναι ο 7-8 με κόστος 220 μέτρα. Επειδή δε σχηματίζεται κύκλος, προσθέτουμε και αυτόν τον κλάδο. Ο επόμενος κλάδος είναι ο 3-4 ωστόσο σχηματίζει κύκλο επομένως δε τον προσθέτουμε. Ο αμέσως επόμενος κλάδος είναι ο 6-7 με κόστος 240 μέτρα. Αφού δεν σχηματίζει κύκλο τον προσθέτουμε κι αυτόν. Τέλος, ο κλάδος 1-3 είναι ο τελευταίος που θα προστεθεί με κόστος 260 μέτρα.

Το συνολικό μήκος των ακμών του δέντρου που σχηματίσαμε με έντονες γραμμές στο παρακάτω σχήμα 5.11 είναι 1400 μέτρα. Το μήκος αυτό είναι το ελάχιστο συνολικό μήκος που απαιτείται ώστε να επικοινωνούν άμεσα ή έμμεσα όλοι οι κόμβοι μεταξύ τους.

Παρατηρούμε ότι η βέλτιστη λύση που πήραμε είναι η ίδια με αυτή που πήραμε από τον προηγούμενο αλγόριθμο Prim και έχει φυσικά το ίδιο κόστος.

Σχήμα 5.11



ΚΕΦΑΛΑΙΟ 6

Πρόβλημα Ροής Ελάχιστου Κόστους (Minimum Cost Flow Problem)

6.1 Εισαγωγή

Το πρόβλημα της Ροής του Ελάχιστου Κόστους είναι ένα ακόμη πρόβλημα γραμμικού προγραμματισμού (υπενθυμίζουμε ότι ο γραμμικός προγραμματισμός είναι η πιο χαρακτηριστική τεχνική της επιχειρησιακής έρευνας όπου σχεδιάστηκε για μοντέλα με αυστηρούς γραμμικούς προγραμματισμούς) κι αποτελεί ένα ιδιαίτερα σημαντικό πρόβλημα μεταξύ των μοντέλων βελτιστοποίησης του δικτύου για δύο βασικούς λόγους. Καταρχάς, διότι περιλαμβάνει μια αρκετά ευρεία κλίμακα εφαρμογών και κατά δεύτερον, επειδή μπορεί να λυθεί πολύ αποτελεσματικά. Η ομοιότητα του με το πρόβλημα της συντομότερης διαδρομής που εξετάσαμε παραπάνω έγκειται στη θεώρηση ενός κόστους (ή απόσταση) για ροή μέσω ενός κλάδου. Παράλληλα, σε αναλογία με το πρόβλημα μέγιστης ροής, το πρόβλημα της ροής ελάχιστου κόστους, εξετάζει επίσης τη ροή μέσω ενός δικτύου με περιορισμένες χωρητικότητες κλάδων. Όπως, λοιπόν, μπορεί να αντιληφθεί κανείς, τα προβλήματα που αναλύθηκαν ήδη αποτελούν ειδικές περιπτώσεις του προβλήματος ροής ελαχίστου κόστους. Σε αντίθεση όμως με τα προβλήματα ελάχιστου μονοπατιού και μέγιστης ροής που είναι σχετικά εύκολο να κατανοήσουμε τη δομή τους και να αναπτύξουμε αλγορίθμους για την επίλυση τους, στην περίπτωση του προβλήματος της ροής του ελάχιστου κόστους δεν είναι εφικτό να αναπτύξουμε εξίσου αποδοτικούς αλγορίθμους, αν και εξακολουθούν να συμπεριλαμβάνονται στους πιο αποτελεσματικούς και γνωστούς αλγορίθμους της επιστήμης των εφαρμοσμένων μαθηματικών και της επιχειρησιακής έρευνας για την επίλυση των προβλημάτων βελτιστοποίησης μεγάλης κλίμακας.

Χαρακτηριστικά του προβλήματος ροής ελάχιστου κόστους:

1.Το δίκτυο είναι προσανατολισμένο και συνεκτικό.

2. Τουλάχιστον ένας κόμβος είναι πηγή.

3. Τουλάχιστον ένας κόμβος είναι δέκτης.

4. Όλοι οι υπόλοιποι κόμβοι είναι κόμβοι ενδιάμεσης μεταφοράς.

5. Ροή σε έναν κλάδο επιτρέπεται μόνο στην κατεύθυνση που δηλώνεται από το αντίστοιχο βέλος και η μέγιστη τιμή της ροής αυτής δίνεται από τη χωρητικότητα του κλάδου αυτού. Στην περίπτωση που επιτρέπεται ροή και στις δύο κατευθύνσεις, αυτό υποδηλώνεται από ένα ζεύγος τόξων με αντίθετες κατευθύνσεις.

6. Το δίκτυο έχει αρκετούς κλάδους με αρκετή χωρητικότητα για να είναι ικανή η ροή που παράγεται στους κόμβους-πηγές να φτάσει στους κόμβους-δέκτες.

7. Το κόστος της ροής μέσω ενός κλάδου είναι ανάλογο με την ποσότητα ροής και το μοναδιαίο τέτοιο κόστος είναι γνωστό.

8. Ο σκοπός είναι να ελαχιστοποιηθεί το συνολικό κόστος μεταφοράς της ποσότητας ροής ώστε να ικανοποιηθεί η συνολική ζήτηση. (Εναλλακτικά ζητούμενο είναι η μεγιστοποίηση του συνολικού κέρδους.)

6.2 Εφαρμογές του προβλήματος ροής ελάχιστου κόστους

Στον παρακάτω πίνακα παρουσιάζονται ενδεικτικά παραδείγματα προβλημάτων ελάχιστης ροή κόστους που προκύπτουν σε διάφορες περιπτώσεις δικτύων.

Χαρακτηριστικές Εφαρμογές	Κόμβοι Προμήθειας	Κόμβοι Μεταφόρτωσης	Κόμβοι Ζήτησης
Λειτουργία ενός δικτύου διανομής	Πηγές αγαθών	Ενδιάμεσες υποδομές αποθήκευσης	Πελάτες
Διαχείριση στερεών αποβλήτων	Πηγές στερεών αποβλήτων	Υποδομές επεξεργασίας	Χώροι υγειονομικής ταφής απορριμμάτων
Λειτουργία ενός δικτύου προμηθειών	Πωλητές	Ενδιάμεση αποθήκευση εμπορευμάτων	Υποδομές επεξεργασίας/μεταποίησης
Συντονιστικές μείξεις προϊόντων στις επιχειρήσεις	Επιχειρήσεις	Παραγωγή ενός συγκεκριμένους προϊόντος	Αγορά για ένα συγκεκριμένο προϊόν
Διαχείριση χρηματικών ροών	Πηγή χρημάτων σε μια συγκεκριμένη χρονική στιγμή	Βραχυπρόθεσμες Επενδυτικές επιλογές	Ανάγκη για χρήματα σε μια συγκεκριμένη χρονική στιγμή

6.3 Μορφοποίηση του προβλήματος

Έστω ένα G=(N,A) προσανατολισμένο (κατευθυνόμενο) και συνεκτικό δίκτυο όπου οι n κόμβοι περιλαμβάνουν τουλάχιστον έναν κόμβο πηγή και τουλάχιστον έναν δέκτη.

Οι μεταβλητές απόφασης είναι:

X_{ij} = η ροή μέσω του κλάδου $i \rightarrow j$

Και οι παράμετροι του προβλήματος είναι:

c_{ij} = το μοναδιαίο κόστος ροής μέσω του κλάδου $i \rightarrow j$,

u_{ij} (L_{ij}) = η μέγιστη (ελάχιστη) χωρητικότητα ροής του κλάδου $i \rightarrow j$,

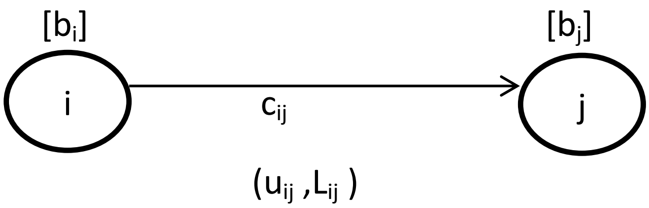
b_i = η καθαρή ροή που δημιουργείται στον κόμβο i .

Η τιμή του b_i εξαρτάται από τη φύση του κόμβου i :

$$b_i \begin{cases} > 0\ , \text{αν ο κόμβος } i \text{ είναι πηγή} \\ < 0, \text{αν ο κόμβος } i \text{ είναι δέκτης} \\ = 0, \text{αν ο κόμβος } i \text{ είναι ενδιάμεσης μεταφοράς} \end{cases}$$

Στο σχήμα 6.2 αποτυπώνονται οι παραπάνω διευκρινήσεις για τον κλάδο i→ j.

Σχήμα 6.1



Ο στόχος είναι να ελαχιστοποιηθεί το συνολικό κόστος της μεταφοράς της διαθέσιμης προμήθειας μέσω του δικτύου για να ικανοποιήσει τη δεδομένη ζήτηση. Χρησιμοποιώντας τη σύμβαση ότι τα σύνολα λαμβάνονται μόνο σε υπάρχοντα τόξα, η μορφοποίηση γραμμικού προγραμματισμού αυτού του προβλήματος είναι :

$$\text{Min } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij}$$

Υπό τους περιορισμούς:

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = b_i, \ \forall \text{ κομβο } i,$$

$$0 \le x_{ij} \le u_{ij}, \ \forall \text{ κλαδο } i \rightarrow j.$$

Το πρώτο άθροισμα στους περιορισμούς κόμβου αντιπροσωπεύει τη συνολική ροή έξω από τον κόμβο i, ενώ το δεύτερο άθροισμα αντιπροσωπεύει τη συνολική ροή προς τον κόμβο i, οπότε η διαφορά είναι η καθαρή ροή που παράγεται σε αυτόν τον κόμβο.

Σε ορισμένες εφαρμογές , είναι απαραίτητο να υπάρχει ένα κάτω όριο $L_{ij} > 0$ για τη ροή μέσω κάθε κλάδου $i \rightarrow j$. Σε αυτή την περίπτωση χρησιμοποιούμε έναν μετασχηματισμό των μεταβλητών $x'_{ij} = x_{ij} - L_{ij}$, και αντικαθιστούμε παντού στο μοντέλο όπου $x_{ij} = x'_{ij} + L_{ij}$,επομένως παίρνουμε πάλι την παραπάνω μορφοποίηση με περιορισμούς μη αρνητικότητας.
Δεν είναι βέβαιο πως το πρόβλημα θα έχει εφικτές λύσεις , το οποίο εξαρτάται κυρίως από το ποιοι κόμβοι είναι υπάρχοντες στο δίκτυο και ποιές οι χωρητικότητές τους. Ωστόσο για ένα εύλογα σχεδιασμένο δίκτυο η κύρια προϋπόθεση είναι η ακόλουθη:

<u>Ιδιότητα για ύπαρξη εφικτών λύσεων</u>: Μία απαραίτητη προϋπόθεση για να έχει ένα πρόβλημα ροής ελάχιστου κόστους εφικτές λύσεις είναι

$$\sum_{i=1}^{n} b_i = 0 .$$

Στην ουσία, η συνολική ροή που παράγεται στους κόμβους-πηγές ισούται με τη συνολική ροή που απορροφάται στους κόμβους-δέκτες.

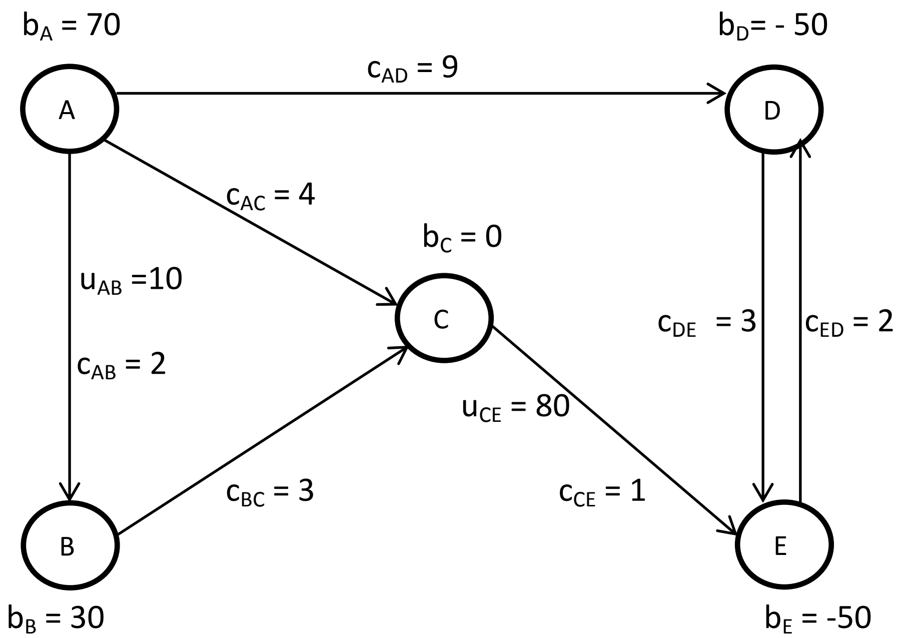
Αν οι τιμές του b_i για κάποια εφαρμογή παραβιάζουν αυτή τη συνθήκη ,η εξήγηση είναι πως είτε οι παροχές είτε οι ζητήσεις (όποιες είναι μεγαλύτερες) αντιπροσωπεύουν στην πραγματικότητα ανώτερα όρια και όχι ακριβή ποσά. Στην περίπτωση που συμβαίνει αυτό είτε προσθέτουμε έναν εικονικό κόμβο-δέκτη για να απορροφήσει την επιπλέον ροή (με κλάδους $c_{ij} = 0$ από κάθε κόμβο-πηγή προς αυτό τον κόμβο) , είτε έναν εικονικό κόμβο πηγή για να παρέχει την επιπλέον ροή (με κλάδους με κόστος $c_{ij} = 0$ από αυτό τον κόμβο προς κάθε κόμβο πηγή).
Σε πολλές εφαρμογές , τα b_i και u_{ij} έχουν ακέραιες τιμές , και η εφαρμογή τους θα απαιτεί ότι και οι ποσότητες ροής X_{ij} θα είναι ακέραιες τιμές. Αυτός ο περιορισμός εξασφαλίζεται χωρίς να χρειάζεται να επιβάλλουμε περιορισμούς ακεραιότητας στις μεταβλητές λόγω της ακόλουθης ιδιότητας:

<u>Ιδιότητα ακέραιων λύσεων:</u> Για προβλήματα ροής ελάχιστου κόστους όπου κάθε bi και uij έχουν ακέραιες τιμές, όλες οι βασικές μεταβλητές σε κάθε βασική εφικτή λύση (συμπεριλαμβανομένης και της βέλτιστης) έχουν επίσης ακέραιες τιμές.

6.4 Παράδειγμα

Έστω ότι έχουμε ένα δίκτυο υδροδοτήσεως που απεικονίζεται στο γράφημα και περιγράφει τη ροή του νερού διαμέσου αγωγών με συγκεκριμένα έξοδα για τη μεταφορά του που για τεχνικούς λόγους θεωρούμε ότι έχουν απεριόριστη χωρητικότητα. Αναζητούμε ελάχιστο κόστος ροής δικτύου.

Σχήμα 6.3



Στο σχήμα 6.3 φαίνονται οι τιμές των b_i για κάθε κόμβο, και τα κόστη c_{ij} για κάθε κλάδο.

Όταν bi > 0, σημαίνει ότι ο αντίστοιχος κόμβος είναι κόμβος πηγή, ενώ όταν bi < 0 σημαίνει ότι είναι κόμβος δέκτης. Ακόμη, όταν $b_i = 0$, ο αντίστοιχος κόμβος είναι ενδιάμεσης μεταφοράς. Όλοι εκτός από δύο κλάδους έχουν χωρητικότητες που είναι μεγαλύτερες από τη συνολική ροή που παράγεται (70 + 30 = 100), οπότε για όλους αυτούς τους κλάδους, η χωρητικότητα μπορεί να θεωρηθεί άπειρη.

Η μορφοποίηση γραμμικού προγραμματισμού αυτού του προβλήματος είναι:

Minimize $Z = 2x_{AB} + 4x_{AC} + 9x_{AD} + 3x_{BC} + x_{CE} + 3x_{DE} + 2x_{ED}$

Υπό τους περιορισμούς :

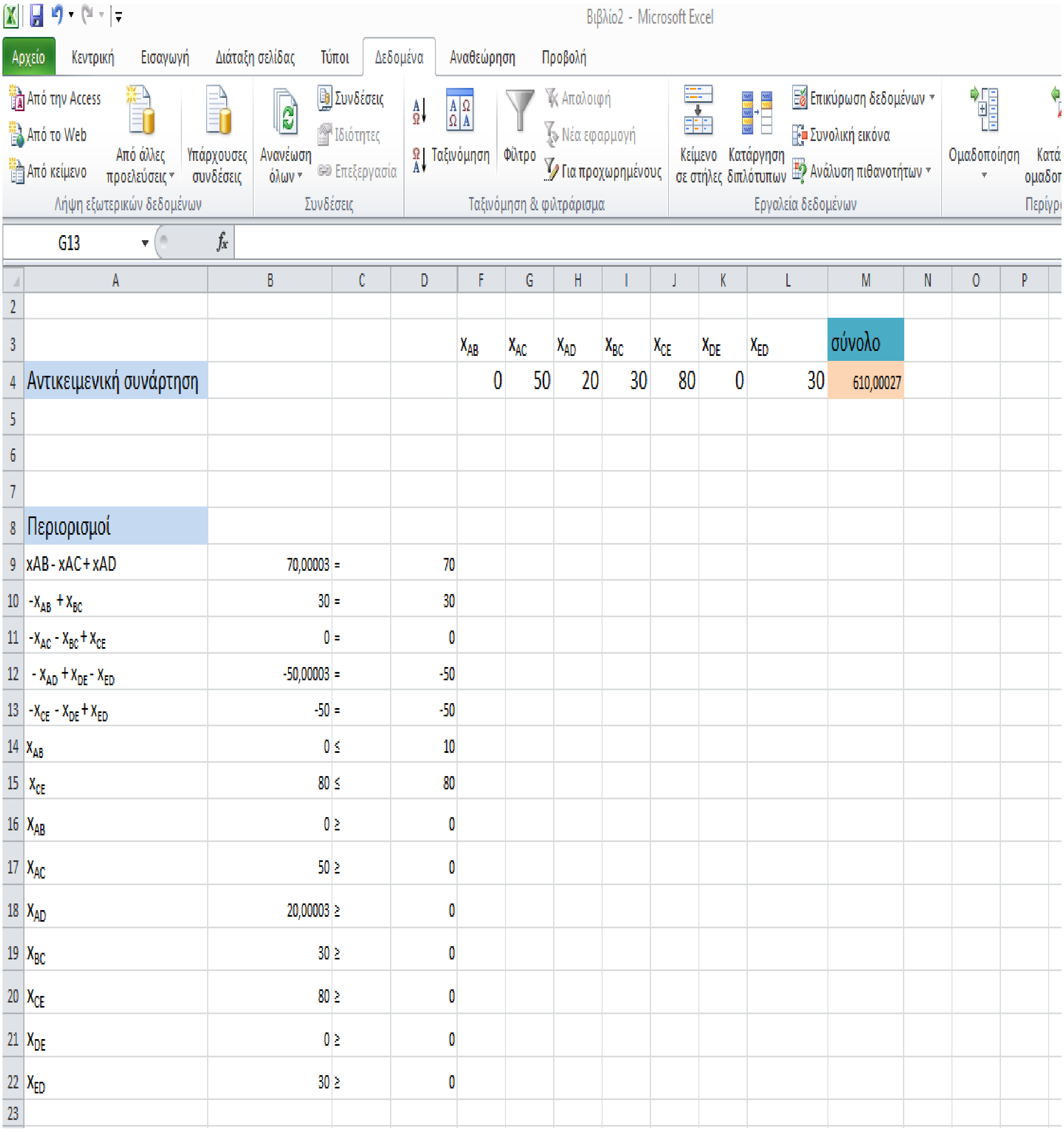
$x_{AB} + x_{AC} + x_{AD}$	$= 70$
$-x_{AB} + x_{BC}$	$= 30$
$-x_{AC} - x_{BC} + x_{CE}$	$= 0$
$-x_{AD} + x_{DE} - x_{ED}$	$= -50$
$-x_{CE} - x_{DE} + x_{ED}$	$= -50$
x_{AB}	≤ 10
x_{CE}	≤ 80
και όλα τα $x_{ij} \geq 0$	≥ 0

Διαπιστώνουμε στην παραπάνω μορφοποίηση την ειδική δομή του προβλήματος ροής ελάχιστου κόστους. Κάθε μεταβλητή εμφανίζεται στους λειτουργικούς περιορισμούς εμφανίζεται ακριβώς δύο φορές, μία με συντελεστή +1 και μία με συντελεστή -1. Αυτή η δομή εμφανίζεται σε κάθε πρόβλημα ροής ελάχιστου κόστους και είναι η αιτία που το πρόβλημα έχει την ιδιότητα ακέραιων λύσεων. Ακόμη, ένα άλλο αποτέλεσμα που προκύπτει από αυτή τη δομή είναι ότι ένας οποιοσδήποτε από τους περιορισμούς κάποιου κόμβου είναι περιττός. Αυτό συμβαίνει επειδή αν αθροίσουμε όλους τους περιορισμούς αυτούς κατά μέλη, και υποθέτοντας ότι εφικτές λύσεις υπάρχουν (δηλαδή το άθροισμα των b_i είναι 0), παίρνουμε 0 και στα δύο μέλη. Με άλλα λόγια, το αρνητικό οποιασδήποτε ισότητας είναι ίσο με το άθροισμα όλων των άλλων ισοτήτων.

Επίλυση του παραδείγματος με χρήση solver του excel

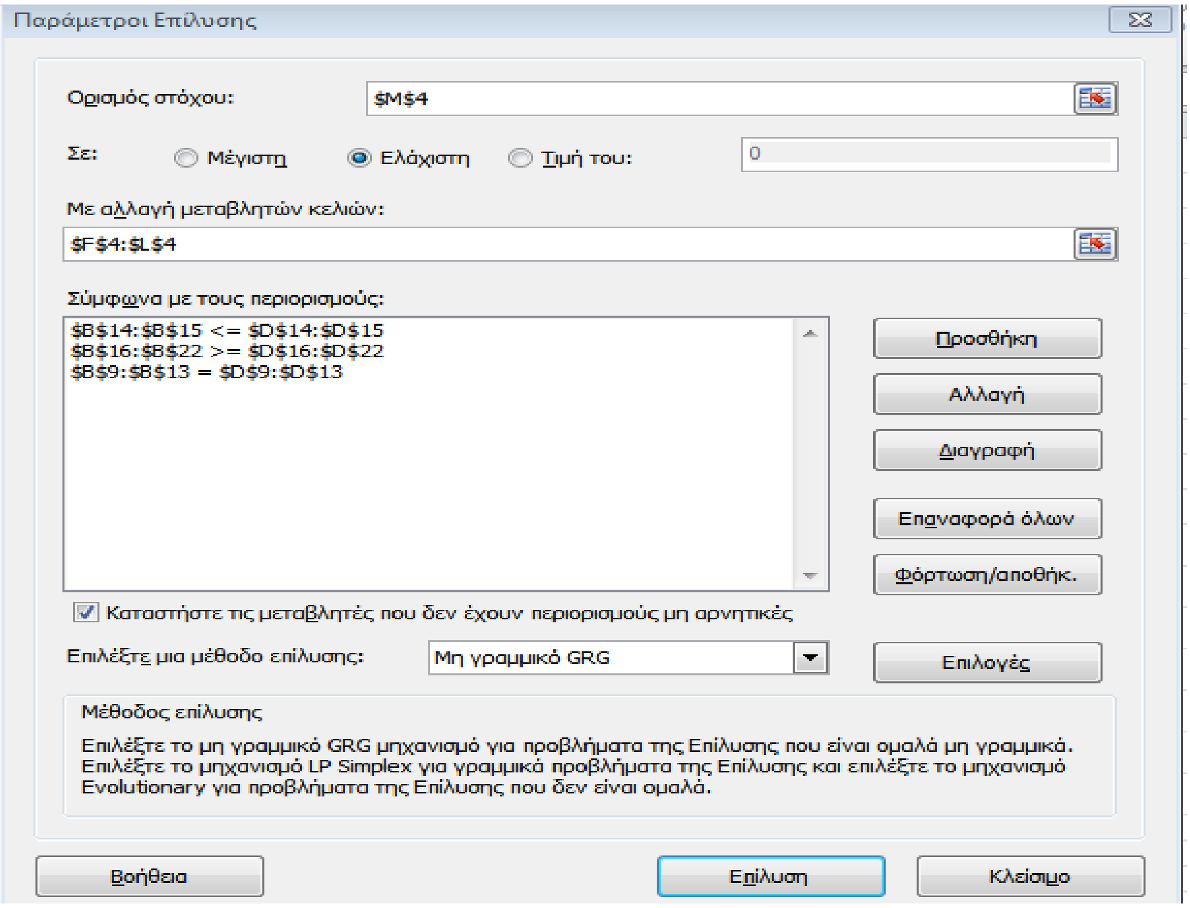
Σαν πρόβλημα γραμμικού προγραμματισμού το πρόβλημα του ελάχιστου κόστους μπορεί να λυθεί με τη μέθοδο simplex. Το excel μας δίνει τη δυνατότητα να επιλύσουμε αυτό το πρόβλημα καθώς και μεγαλύτερα προβλήματα δικτύων.
Επομένως θα πάρουμε τα εξής αποτελέσματα:

Εικόνα 1



	A	B	C	D	F	G	H	I	J	K	L	M
2												
3					X_{AB}	X_{AC}	X_{AD}	X_{BC}	X_{CE}	X_{DE}	X_{ED}	σύνολο
4	Αντικειμενική συνάρτηση				0	50	20	30	80	0	30	610,00027
5												
6												
7												
8	Περιορισμοί											
9	xAB - xAC + xAD	70,00003	=	70								
10	$-X_{AB} + X_{BC}$	30	=	30								
11	$-X_{AC} - X_{BC} + X_{CE}$	0	=	0								
12	$-X_{AD} + X_{DE} - X_{ED}$	-50,00003	=	-50								
13	$-X_{CE} - X_{DE} + X_{ED}$	-50	=	-50								
14	X_{AB}	0	≤	10								
15	X_{CE}	80	≤	80								
16	X_{AB}	0	≥	0								
17	X_{AC}	50	≥	0								
18	X_{AD}	20,00003	≥	0								
19	X_{BC}	30	≥	0								
20	X_{CE}	80	≥	0								
21	X_{DE}	0	≥	0								
22	X_{ED}	30	≥	0								
23												

Εικόνα 2



Στις εικόνες 1 και 2 διακρίνουμε ότι:
στο κελί M4 όπου έχει καταχωρηθεί η εντολή = SUMPRODUCT(2*F4+4*G4+9*H4+3*I4+J4+3*K4+2*L4) αποδίδεται η τιμή της αντικειμενικής συνάρτησης του προβλήματός μας. Για τα δεδομένα του παραδείγματος προκύπτει πως το ελάχιστο κόστος είναι 610 μονάδες. Για να έχουμε ωστόσο το παραπάνω ελάχιστο κόστος και για να ικανοποιούνται και οι περιορισμοί απαιτείται να μην στείλουμε καμία μονάδα από το A στα B ,να στείλουμε 50 μονάδες από το A στο C , 20 μονάδες από το A στο D. Ακόμη, να σταλθούν 30 μονάδες από το B στο C ,80 μονάδες από το C στο E , καμία μονάδα από το D στο E και 30 μονάδες από το E στο D.

ΚΕΦΑΛΑΙΟ 7

Ανάλυση περίπτωσης (Case Study) για την επιχείρηση «Olive-oil Kalamata»

Το πρόβλημα της Συντομότερης Διαδρομής-Μια μελέτη περίπτωσης (Shortest Path problem- A case study)

7. 1.Περιγραφή:

Στη σύγχρονη κοινωνία, το ζήτημα της συντόμευσης του χρόνου έχει αναχθεί σε υψίστης σημασίας. Ιδίως στην περίπτωση του εμπορίου και της μεταφοράς των αγαθών, ο χρόνος αποτελεί μια κρίσιμη παράμετρο που επηρεάζει σε σημαντικό βαθμό την εύρυθμη λειτουργία της επιχείρησης, τα λειτουργικά της κόστη, την ταχύτερη πρόσβαση στην αγορά, κλπ. Κατά συνέπεια, στο πλαίσιο της βελτιστοποίησης της λειτουργία της επιχείρησης, είναι λογικό να αναζητείται η συντομότερη διαδρομή προκειμένου η εταιρεία να μεταφέρει τα προϊόντα της ελαχιστοποιώντας της απόσταση από την αγορά-στόχο και κατ' επέκταση το χρόνο, το κόστος μεταφοράς, κοκ.

Στην προκείμενη περίπτωση που μελετάμε, επιχειρούμε να επιλύσουμε ένα πρόβλημα που αντιμετωπίζει η επιχείρηση «Olive-oil Kalamata» που συσκευάζει λάδι που παράγεται στο νομό Μεσσηνίας με έδρα την Καλαμάτα. Η εν λόγω εταιρεία πρόσφατα ήλθε σε συμφωνία με τοπική αλυσίδα σουπερ-μάρκετ στην Θεσσαλονίκη προκείμενου να πωλούνται τα προϊόντα της στα καταστήματα της αλυσίδας.

Στόχος : της επιχείρησης είναι να εντοπίσει τη διαδρομή εκείνη με τη μικρότερη χιλιομετρικά απόσταση, ώστε να μεταφέρει το τυποποιημένο λάδι από την Καλαμάτα στη Θεσσαλονίκη. Για το σκοπό αυτό, στη συνέχεια μέσω του excel solver επιχειρούμε να βρούμε τη συντομότερη διαδρομή προκειμένου να επιλύσουμε αυτό το πρόβλημα μεταφοράς της επιχείρησης.

7.2. Χρήση του Excel για τη διαμόρφωση και την επίλυση του προβλήματος ελάχιστης διαδρομής (shortest path problem)

Εκτός από τους αλγορίθμους Dijkstra και Floyd που αναλύσαμε στο κεφάλαιο 3 σε αυτό το κεφάλαιο θα χρησιμοποιήσουμε το excel για την επίλυση προβλημάτων ελάχιστη διαδρομής.

Δεδομένου ότι το πρόβλημα της μικρότερης διαδρομής είναι ένας ειδικός τύπος προβλήματος γραμμικού προγραμματισμού, η γενική μέθοδος simplex μπορεί επίσης να χρησιμοποιηθεί όταν οι καλύτερες επιλογές δεν είναι άμεσα διαθέσιμες. Το Excel, το οποίο βασίζεται στη γενική μέθοδο simplex, παρέχει έναν βολικό τρόπο διαμόρφωσης και επίλυσης προβλημάτων μικρής διαδρομής με δεκάδες τόξα και κόμβους.

Μία διαδρομή, από την προέλευση στον προορισμό ερμηνεύεται ως μια "ροή" 1 στην επιλεγμένη διαδρομή μέσω του δικτύου. Οι αποφάσεις που πρέπει να λαμβάνονται είναι ποιά τόξα θα πρέπει να περιλαμβάνονται στο μονοπάτι που θα διασχίζεται. Σε μια ροή ,του αποδίδεται ο αριθμός 1 σε ένα τόξο αν συμπεριλαμβάνεται, ενώ η ροή είναι 0 εάν δεν περιλαμβάνεται.

Επομένως, οι μεταβλητές απόφασης είναι:

$$X_{ij} = \begin{cases} 1\,, αν\ η\ ακμή\ i \rightarrow j\ είναι\ επιλεγμένη \\ 0, αν\ η\ ακμή\ i \rightarrow j\ δεν\ είναι\ επιλεγμένη \end{cases}$$, για κάθε ένα από τα

υπό εξέταση τόξα

Κάθε κόμβος μπορεί να θεωρηθεί ότι έχει ροή 1 που περνά διαμέσου αυτού αν είναι στην επιλεγμένη διαδρομή, αλλά δεν έχει ροή διαφορετικά. Η καθαρή ροή που παράγεται σε έναν κόμβο είναι η ροή μείον τη ροή, έτσι ώστε η καθαρή ροή είναι 1 στην αρχή, -1 στον προορισμό και 0 σε κάθε άλλο κόμβο.

Η αντικειμενική συνάρτηση του γραμμικού προβλήματος λαμβάνει τη μορφή:

Minimize Z = $\sum_{όλων\ των\ ορισμένων\ τόξων\ (i,j)} c_{ij}\, x_{ij}$

Όπου x_{ij} : η ποσότητα ροής του τόξου (i,j)

c_{ij} : το μήκος του τόξου (i,j)

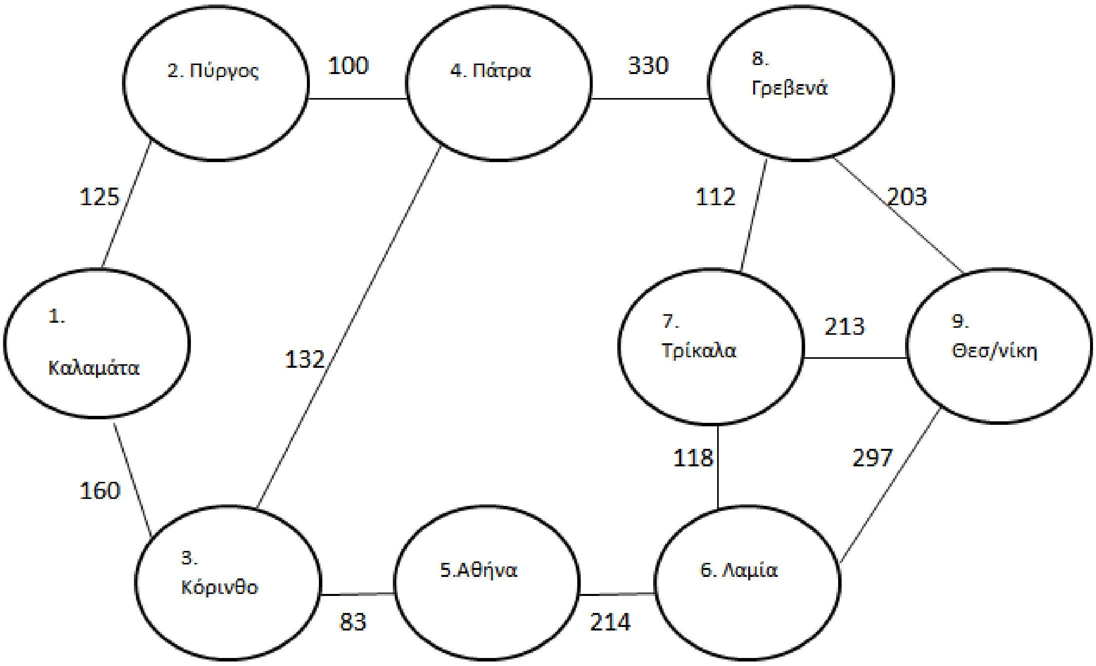
7.3 Επίλυση παραδείγματος

Η επιχείρηση επιθυμεί να μεταφέρει το εμπόρευμά της με φορτηγό από το εργοστάσιό της στην Καλαμάτα, στη Θεσσαλονίκη. Στον παρακάτω πίνακα 7.1 παρουσιάζονται οι διάφορες διαδρομές που απαιτείται ώστε να φτάσει το φορτηγό στον προορισμό του. Ο στόχος όπως περιγράψαμε και νωρίτερα είναι να βρούμε τη συντομότερη διαδρομή από την αφετηρία (Καλαμάτα) στον τελικό κόμβο (Θεσσαλονίκη).

Ο Πίνακας 1 εμφανίζει τις αποστάσεις για τις επιλεγμένες διαδρομές που αποκτήθηκαν από το Λογισμικό Google Earth.

Από	Προς	Χιλιομετρική απόσταση
Καλαμάτα	Πύργο	125
Καλαμάτα	Κόρινθο	160
Πύργο	Πάτρα	100
Πάτρα	Γρεβενά	330
Κόρινθο	Αθήνα	83
Αθήνα	Λαμία	214
Πάτρα	Κόρινθος	132
Λαμία	Τρίκαλα	118
Κόρινθος	Πάτρα	132
Τρίκαλα	Γρεβενά	112
Γρεβενά	Τρίκαλα	112
Τρίκαλα	Λαμία	118
Γρεβενά	Θεσσαλονίκη	203
Λαμία	Θεσσαλονίκη	297
Τρίκαλα	Θεσσαλονίκη	213

Σχήμα7.1



Η αντικειμενική συνάρτηση του προβλήματος είναι:

MinimizeZ=

$$125*X_{12}+160*X_{13}+100*X_{24}+132*X_{34}+83*X_{35}+132*X_{43}+214*X_{56}+118*X_{67}+118*X_{76}+330*X_{48}+112*{}_{X78}+112*{}_{X87}+297*X_{69}+203*X_{89}+213*X_{79}$$

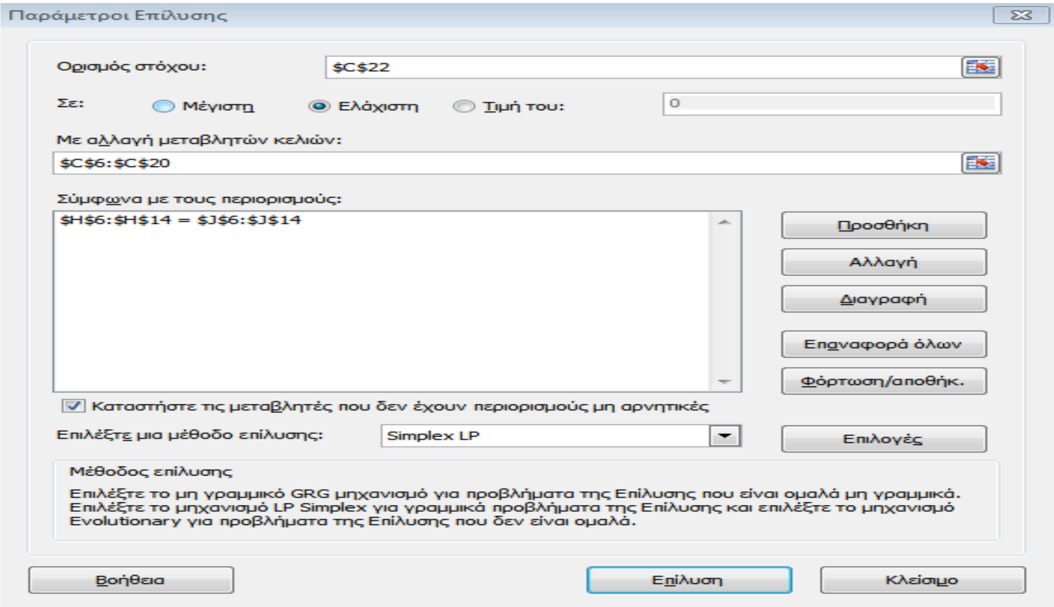
Τα αποτελέσματα και οι εντολές που χρησιμοποιήσαμε στο solver του excel παρουσιάζονται παρακάτω:

Εικόνα 1

P8

	A	B	C	D	E	F	G	H	I	J
1	**Shortest Path problem**									
2										
3										
4	From	To	Go	Distance			Κόμβος	Ροή Δικτύου		Προμήθεια/Ζήτηση
5										
6	Καλαματα	Πυργο	0	121			Καλαματα	1	=	1
7	Καλαματα	Κόρινθο	1	160			Κορινθο	0	=	0
8	Κορινθο	Αθηνα	1	83			Αθηνα	0	=	0
9	Αθηνα	Λαμια	1	214			Λαμια	0	=	0
10	Πατρα	Κορινθο	0	132			Πυργο	0	=	0
11	Πυργο	Πατρα	0	97			Πατρα	0	=	0
12	Κορινθο	Πατρα	0	132			Τρικαλα	0	=	0
13	Πατρα	Γρεβενα	0	324			Γρεβενα	0	=	0
14	Τρικαλα	Γρεβενα	0	112			Θεσ/νικη	-1	=	-1
15	Γρεβενα	Τρικαλα	0	112						
16	Λαμια	Τρικαλα	0	118						
17	Τρικαλα	Λαμια	0	118						
18	Τρικαλα	Θεσ/νικη	0	213						
19	Γρεβενα	Θεσ/νικη	0	203						
20	Λαμια	Θεσ/νικη	1	297						
21										
22		Ελάχιστη Απόσταση	754							
23										

Εικόνα 2



Στις παραπάνω εικόνες 1 και 2 παρατηρούμε:

Ότι στο κελί C22 όπου είναι η ελάχιστη απόσταση που αναζητούμε έχουμε εισάγει την εντολή:

=SUMPRODUCT(C6:C20; D$6:D$20) , και είναι η αντικειμενική συνάρτηση του προβλήματος.

Επίσης από το κελί H6 έως το κελί H14 έχουμε εισάγει τις εντολές:

=SUMIF(A6:A20;G$6;C$6:C$20)-SUMIF(B6:B20;G$6;C$6:C$20)

=SUMIF(A6:A20;G$7;C$6:C$20)-SUMIF(B6:B20;G$7;C$6:C$20)

=SUMIF(A6:A20;G$8;C$6:C$20)-SUMIF(B6:B20;G$8;C$6:C$20)

=SUMIF(A6:A20;G$9;C$6:C$20)-SUMIF(B6:B20;G$9;C$6:C$20)

=SUMIF(A6:A20;G$10;C$6:C$20)-SUMIF(B6:B20;G$10;C$6:C$20)

=SUMIF(A6:A20;G$11;C$6:C$20)-SUMIF(B6:B20;G$11;C$6:C$20)

=SUMIF(A6:A20;G$12;C$6:C$20)-SUMIF(B6:B20;G$12;C$6:C$20)

=SUMIF(A6:A20;G$13;C$6:C$20)-SUMIF(B6:B20;G$13;C$6:C$20)

=SUMIF(A6:A20;G$14;C$6:C$20)-SUMIF(B6:B20;G$14;C$6:C$20)

Ακόμη, στη στήλη C6 έως C20 σε όποιο κελί υπάρχει ο αριθμός 1 σημαίνει ότι αυτή η διαδρομή είναι η ελάχιστη που πρέπει να ακολουθήσει το φορτηγό της επιχείρησης.

Επίσης στην εικόνα 1 βλέπουμε τους περιορισμούς που έχουμε για τη λύση του προβλήματος οι οποίου βρίσκονται στην δεξιά πλευρά στη σελίδα του φύλλου του excel.

Τέλος, η συνολική ελάχιστη απόσταση που αναζητούσε η επιχείρηση «Olive-oil Kalamata» για την μεταφορά του ελαιολάδου από την Καλαμάτα στη Θεσσαλονίκη είναι **754 χιλιόμετρα** και η διαδρομή που πρέπει να ακολουθήσει το φορτηγό είναι :

Καλαμάτα→ Κόρινθο→ Αθήνα →Λαμία→ Θεσσαλονίκη.

Βιβλιογραφία

Ελληνική βιβλιογραφία

- **Κολέτσος Ιωάννης, Στογιάννης Δημήτρης**, Εισαγωγή στην Επιχειρησιακή Έρευνα, Εκδόσεις Συμεών, 2012

- **Ξηρόκωστας, Α. Δ.** Επιχειρησιακή Έρευνα, Αθήνα, Εκδόσεις Συμμετρία, 1999

- **Οικονόμου Γεωργίου, Γεωργίου Ανδρέα** , Επιχειρησιακή Έρευνα για τη λήψη Διοικητικών Αποφάσεων, 2016

- **Παπαγεωργίου, Γ.** Εισαγωγή στην Επιχειρησιακή Έρευνα – Γραμμικός Προγραμματισμός και Εφαρμογές, Ε.Μ.Π, Αθήνα, 2004

Ξένη βιβλιογραφία

- **Ahuja R. , T. Magnati and J. Orlin,** Network Flows: Theory, Algorithms and Applications,Prentice Halll, Upper Saddle River, NJ, 1993

- **Bazaraa, M. S., Jarvis, J. J., Sherali, H. D.** , Linear Programming and Network Flows, Wiley Intersienc*e, 2009*

- **Bersetkas, D**. , Linear Network Optimization: Algorithms and Codes. Cambridge, Mass.: MIT Press, 1991.

- **Hillier S. F., Lieberman G. J.** , Introduction to Operations Research, 7th Edition, Mc-Graw – Hill ,2001

- **Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin**, Network Flows: Theory, Algorithms and Applications, Prentice Hall, 1993

- **Taha, A. Handy** , Operations Research, an introduction , Ninth edition , 2010

- **Wayne L. Winston** , Operatios Research : Applications and Algorithms ,2003

Ηλεκτρονική Βιβλιογραφία

- http://en.wikipedia.org