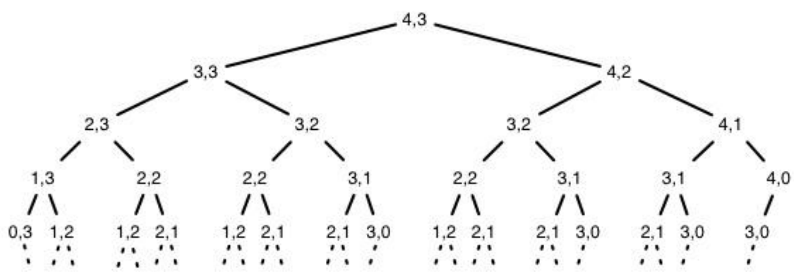
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Δ.Π.Μ.Σ.: ΕΦΑΡΜΟΣΜΕΝΕΣ ΜΑΘΗΜΑΤΙΚΕΣ ΕΠΙΣΤΗΜΕΣ

Δυναμικός Προγραμματισμός



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΧΡΥΣΟΥΛΑΣ ΧΑΤΖΗ

Α.Μ.: 09414030

Επιβλέπων: Ιωάννης Κολέτσος
Επίκουρος Καθηγητής Ε.Μ.Π

Τριμελής επιτροπή:

Βασίλειος Κοκκίνης	Ιωάννης Κολέτσος	Ευάγγελος Τυχόπουλος
Επίκουρος Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π

Αθήνα 2016

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Δ.Π.Μ.Σ.: ΕΦΑΡΜΟΣΜΕΝΕΣ ΜΑΘΗΜΑΤΙΚΕΣ ΕΠΙΣΤΗΜΕΣ

Δυναμικός Προγραμματισμός

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΧΡΥΣΟΥΛΑΣ ΧΑΤΖΗ
Α.Μ.: 09414030

Επιβλέπων : Ιωάννης Κολέτσος
Επίκουρος Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την **4η Μαρτίου 2016**

....	....	....
Βασίλειος Κοκκίνης Επίκουρος Καθηγητής Ε.Μ.Π.	Ιωάννης Κολέτσος Επίκουρος Καθηγητής Ε.Μ.Π.	Ευάγγελος Τυχόπουλος Επίκουρος Καθηγητής Ε.Μ.Π.

Η ολοκλήρωση της διπλωματικής μου εργασίας "Δυναμικός Προγραμματισμός" χρηματοδοτήθηκε από το Ι.Κ.Υ. στο πλαίσιο του προγράμματος χορήγησης υποτροφιών για μεταπτυχιακές σπουδές πρώτου κύκλου (μάστερ) στην Ελλάδα με ένταξη στην αγορά εργασίας, ακαδ. έτους 2014-15.

Ευχαριστίες

Η παρούσα εργασία αποτελεί διπλωματική εργασία στα πλαίσια του μεταπτυχιακού προγράμματος "Εφαρμοσμένων Μαθηματικών Επιστημών" της σχολής Εφαρμοσμένων Μαθηματικών & Φυσικών Επιστημών του Ε.Μ.Π.

Πριν την παρουσίαση του κυρίως μέρους της εργασίας μου, νιώθω την υποχρέωση να εκφράσω την ευγνωμοσύνη μου σε ορισμένους ανθρώπους που βοήθησαν ο καθένας με τον τρόπο του στην πραγματοποίησή της.

Καταρχάς, ευχαριστώ θερμά τον επιβλέποντα καθηγητή της διπλωματικής εργασίας επίκουρο καθηγητή Ε.Μ.Π. κ. Κολέτσο Ιωάννη που με εισήγαγε στον κόσμο της Επιχειρησιακής Έρευνας και μου έδωσε την ευκαιρία να γνωρίσω ίσως το πιο ενδιαφέρον κομμάτι της, το Δυναμικό Προγραμματισμό, που αποτέλεσε το αντικείμενο έρευνάς μου στην εργασία αυτή.

Τις ευχαριστίες μου εκφράζω και σε όλους τους καθηγητές του μεταπτυχιακού προγράμματος για τις πολύτιμες γνώσεις που μου προσέφεραν, τόσο στη βασική μου ροή της Στατιστικής, όσο και της Ανάλυσης και των Υπολογιστικών Μαθηματικών.

Θα ήθελα να ευχαριστήσω τους γονείς μου Θωμά και Μαριάνα, καθώς και τον αδερφό μου Αντώνη για την αμέριστη συμπαράσταση και αγάπη τους όλα αυτά τα χρόνια, κατά τη διάρκεια των προπτυχιακών και μεταπτυχιακών σπουδών μου.

Τέλος, τις θερμότερες και ιδιαίτερες ευχαριστίες μου θα ήθελα να απευθύνω στον Ανδρίκο Χρήστο που στάθηκε σημαντικός αρωγός σε όλη μου την προσπάθεια και κάθε φάση της μεταπτυχιακής μου πορείας και συνέβαλε τα μέγιστα στην ολοκλήρωση της μεταπτυχιακής μου εργασίας.

Περίληψη

Ο Δυναμικός Προγραμματισμός αποτελεί έναν από τους σημαντικότερους κλάδους της Επιχειρησιακής Έρευνας και μία σημαντική τεχνική για την ανάλυση θεωρητικών και πρακτικών προβλημάτων.

Η ιστορία του ξεκινά στα μέσα του 19ου αιώνα, όταν και τον ανακάλυψε ο Αμερικανός μαθηματικός Richard Bellman, με σκοπό να περιγράψει τη διαδικασία επίλυσης προβλημάτων διαδοχικών αλληλοεξαρτώμενων αποφάσεων, όπου η κάθε απόφαση επηρεάζει τις επόμενες αποφάσεις. Η λέξη "δυναμικός" προέκυψε από το γεγονός ότι η φύση της διαδικασίας του προγραμματισμού είναι χρονικά μεταβαλλόμενη και πολυσταδιακή, δηλαδή συμβαίνει σε πολλά διαδοχικά στάδια. Την περίοδο εκείνη, εμφανίστηκαν περίπλοκα προβλήματα με πολλές ανεξάρτητες μεταβλητές, στα οποία οι ήδη υπάρχουσες μέθοδοι του Γραμμικού και Ακέραιου Προγραμματισμού παρουσίαζαν κάποια δυσλειτουργία ως προς την επίλυση αυτών, αντίθετα με το Δυναμικό Προγραμματισμό που έδωσε ακριβείς και αξιόπιστες απαντήσεις.

Στόχος της παρούσας εργασίας είναι να αναπτύξει, αναλύσει και αναδείξει τη χρησιμότητα αυτής της σημαντικής μεθόδου, καθώς και να παρουσιάσει στον αναγνώστη τη μεθοδολογία της αυτής τεχνικής μέσα από μία πληθώρα εφαρμογών. Εφαρμογές που μπορεί να προέρχονται από τον επιχειρηματικό, βιομηχανικό ή τραπεζικό κλάδο, αλλά και την καθημερινή ζωή.

Η εργασία αποτελείται από τρία βασικά κεφάλαια. Στο πρώτο, γίνεται μία εισαγωγή στην επιχειρησιακή έρευνα και τον τρόπο που αυτή συνδέεται με το Δυναμικό Προγραμματισμό. Στο δεύτερο, ξεκινάει το κυρίως μέρος του Δυναμικού Προγραμματισμού με μία ιστορική αναδρομή αρχικά, κάποιες βασικές αρχές, το θεωρητικό υπόβαθρο και μεθοδολογία του, καθώς και την κατηγοριοποίησή του σε Ντετερμινιστικό και Πιθανοθεωρητικό (ή Στοχαστικό). Στο τρίτο και τελευταίο

κεφάλαιο, ακολουθεί διεξοδική παρουσίαση και ανάλυση μερικών από τα πιο διάσημα προβλήματα που χρησιμοποιούν τη συγκεκριμένη μέθοδο για την επίλυσή τους.

ΠΕΡΙΕΧΟΜΕΝΑ

1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ

Από αρχαιοτάτων χρόνων εντοπίζεται συχνά η συνεργασία μεταξύ επιστημόνων και στρατιωτικών αξιωματούχων με ένα και μοναδικό κοινό στόχο: **Τη λήψη της βέλτιστης απόφασης στη μάχη**. Συγκεκριμένα, πολλοί ειδικοί υποστηρίζουν ότι η Επιχειρησιακή Έρευνα (Operational and Operations Research) εμφανίζεται για πρώτη φορά τον 3ο π.Χ. αιώνα, κατά το δεύτερο Καρχηδονιακό Πόλεμο (Punic War) υπό τη μορφή αναλύσεων και λύσεων που ο Αρχιμήδης όρισε, στοχεύοντας στην οχύρωση της πόλης των Συρακουσών που πολιορκήθηκε από τους Ρωμαίους. Ο καταπέλτης, καθώς και ένα σύστημα κατόπτρων ικανών να καίνε τα πλοία των πολιορκητών, ήταν δύο αξιομνημόνευτες εφευρέσεις του, όπως προέκυψαν στα πλαίσια της αυτής προσπάθειας για βέλτιστη οχύρωση της πόλης.

Το 1503, ο Leonardo DaVinci λαμβάνει μέρος ως μηχανικός στον πόλεμο κατά της Prisa, καταστρώνοντας τεχνικές βομβαρδισμών, κατασκευής πλοίων, θωρακισμένων οχημάτων, κανονιών, καταπελτών αλλά και άλλων πολεμικών μηχανών. Ο F.W. Lancaster ορίζει το Lancaster’s Square Law, ένα μαθηματικό μοντέλο για την πρόβλεψη της έκβασης στρατιωτικών συρράξεων, βασισμένο σε ένα σύστημα διαφορικών εξισώσεων. Ο Thomas Edison χρησιμοποιεί την επιχειρησιακή έρευνα προκειμένου να συνεισφέρει στον πόλεμο κατά των υποβρυχίων, αναπτύσσοντας ιδέες, όπως οι ασπίδες κατά των τορπιλών.

Η *Επιχειρησιακή Έρευνα, λοιπόν,* αποτελεί επιστημονικό κλάδο που στοχεύει στην τεκμηριωμένη λήψη αποφάσεων, μέσω της εφαρμογής προηγμένων αναλυτικών μεθόδων και βέλτιστων, ως προς τα δεδομένα του εκάστοτε προβλήματος, λύσεων.

Μοχλεύοντας και επεκτείνοντας άλλες μαθηματικές τεχνικές, όπως τη στατιστική ανάλυση, τη μαθηματική μοντελοποίηση και βελτιστοποίηση, ο σπουδαίος αυτός κλάδος κατορθώνει να προσεγγίζει, ή ακόμα και να εντοπίζει βέλτιστες λύσεις σε πολύπλοκα προβλήματα λήψης αποφάσεων.

Από μαθηματικής απόψεως, κατά τον 17^{ο} και 18^{ο} αιώνα, οι Newton, Leibnitz, Bernoulli και Lagrange, εργάσθηκαν για την εξαγωγή μεγίστων και ελαχίστων, περιορισμένων από προκαθορισμένες συναρτήσεις. Οι απαρχές του γραμμικού προγραμματισμού σκιαγραφούνται εκείνη την περίοδο από τον Γάλλο Μαθηματικό Jean Batiste Joseph Fourier, ενώ προς το τέλος του 18^{ου} αιώνα, ο Gaspar Monge όρισε τις απαρχές της Γραφικής Μεθόδου (Graphical Method) χάρη στην ανάπτυξη της Περιγραφικής Γεωμετρίας (Descriptive Geometry) [1][2].

Ο Janos Von Neumann δημοσίευσε τη δουλειά του “Theory of Games”, η οποία εισήγαγε τους κλασικούς μαθηματικούς στην ιδέα του **Γραμμικού Προγραμματισμού**. Γνωστή και ως Γραμμική Βελτιστοποίηση, πρόκειται για μια μέθοδο που στοχεύει στην επίτευξη του καλύτερου δυνατού αποτελέσματος (όπως μέγιστου κέρδους ή ελάχιστου κόστους) σε ένα μαθηματικό μοντέλο του οποίου οι περιορισμοί αναπαρίστανται από γραμμικούς συσχετισμούς. Αργότερα, το 1947, εντόπισε την ομοιότητα του Γραμμικού Προγραμματισμού και της θεωρίας πινάκων που είχε ο ίδιος ορίσει. [3][4][5][6]

Βέβαια, η μαθηματική θεωρία, που είναι γνωστή ως **Γραμμικός Προγραμματισμός**, είχε ήδη αναπτυχθεί από το 1939, από το Σοβιετικό μαθηματικό και οικονομολόγο Leonid Kantorovic σε συνεργασία με τον Ολλανδό μαθηματικό Tjalling Koopmans. Ο δε **Kantorovic** θεωρείται πατέρας του Γραμμικού Προγραμματισμού και ήταν ο νικητής του βραβείου Stalin το 1949 και του βραβείου Nobel (μαζί με τον Koopmans) στα οικονομικά το 1975. Οι δύο μαθηματικοί μελέτησαν ξεχωριστά για πρώτη φορά το Πρόβλημα της Μεταφοράς, το οποίο αργότερα έμεινε γνωστό ως Πρόβλημα των Koopmans-Kantorovic. Για τη λύση του, χρησιμοποίησαν γεωμετρικές μεθόδους που προκύπτουν από το θεώρημα κυρτότητας του Minkowski.

Στα τέλη της δεκαετίας του ’30, ο George Joseph Stigler παρουσίασε ένα συγκεκριμένο πρόβλημα -γνωστό ως το πρόβλημα της βέλτιστης ειδικής δίαιτας ή απλά πρόβλημα της δίαιτας- το οποίο προέκυψε από την ανησυχία του στρατού των Ηνωμένων Πολιτειών να διασφαλίσει τα τρόφιμα για τα στελέχη του στην χαμηλότερη τιμή. Λύθηκε με μια ευρετική μέθοδο που η λύση της δεν απείχε πολύ από αυτή που θα προσέφερε χρόνια αργότερα η μέθοδος Simplex [7][8] (βλ. παράρτημα 4.1).

Γίνεται σαφές, λοιπόν, ότι η Επιχειρησιακή Έρευνα επικεντρώνεται σε πρακτικές εφαρμογές που χρήζουν τεκμηριωμένης βέλτιστης λύσης, καθώς επίσης και σε έννοιες αλληλεπίδρασης ανθρώπου - μηχανής. Πιστεύεται ότι ο Charles Babbage είναι ο πατέρας της Επιχειρησιακής Έρευνας, λόγω της εμπεριστατωμένης έρευνάς του σχετικά με το κόστος μεταφοράς και της διαλογής της αλληλογραφίας, που πραγματοποιήθηκε για λογαριασμό του Uniform Penny Post στην Αγγλία το 1840.

Ωστόσο, η Επιχειρησιακή Έρευνα θα περιμένει αρκετά, ώστε να αναγνωριστεί ως αυτόνομη επιστήμη. Κατά τον δεύτερο Παγκόσμιο Πόλεμο, η γερμανική Πολεμική Αεροπορία Luftwaffe υπέβαλε τους Βρετανούς σε μια σκληρή αεροπορική επιδρομή, δεδομένου ότι πρώτοι είχαν μειονέκτημα ισχύος στον εναέριο χώρο. Η βρετανική κυβέρνηση, για να προασπίσει τη χώρα της, συγκάλεσε επιστήμονες διαφορετικών ειδικοτήτων, οι οποίοι με τη σειρά τους επιζητούσαν τη βελτιστοποίηση της αεράμυνας της χώρας. Η λύση στο πρόβλημα δόθηκε από την τοποθέτηση των διαθέσιμων κεραιών με τρόπο τέτοιο, ώστε να επιτυγχάνεται η βέλτιστη κατανομή των σημάτων τους. Η μεγιστοποίηση του οφέλους των διαθέσιμων ραντάρ υπήρξε αρκετή, ώστε να διπλασιάσει την αποτελεσματικότητα του εναέριου αμυντικού συστήματος της χώρας! Η δραματική αυτή βελτίωση οδήγησε στη σύσταση επιπλέον παρόμοιων ομάδων στην Αγγλία, με στόχο τη μεγιστοποίηση της απόδοσης των διαθέσιμων πόρων και υποδομών του Αγγλικού κράτους.

Το ακριβές ανάλογο, όμως, αναπτύσσεται και από την αντίπερα όχθη του Ατλαντικού. Στις Η.Π.Α. τίθεται σε εφαρμογή για πρώτη φορά το σχέδιο SCOOP (Scientific Computation Of Optimum Programs), όταν αναμειγνύονται στον Πόλεμο το 1940, στο οποίο μάλιστα εργαζόταν ο George Bernard Dantzig, ο οποίος το 1947 διατύπωσε το γενικό πρόβλημα του γραμμικού προγραμματισμού. Λίγο αργότερα, θα διατυπώσει τον Αλγόριθμο **Simplex** για συστηματική επίλυση του προβλήματος. Η δυσκολία, όμως, της μεθόδου αυτής ήταν ότι στη χειρότερη περίπτωση απαιτούσε εκθετικό χρόνο!

Κατά τη διάρκεια του Ψυχρού Πολέμου (ανάμεσα στο δυτικό μπλοκ με κυρίαρχη δύναμη τις ΗΠΑ και το ανατολικό μπλοκ με την ΕΣΣΔ), η πρώην Σοβιετική Ένωση, εξαιρούμενη του σχεδίου Μάρσαλ, ήθελε να ελέγχει τις επίγειες επικοινωνίες, συμπεριλαμβανομένων των ποτάμιων διαδρομών από το Βερολίνο. Προκειμένου να αποφευχθεί η παράδοση της πόλης στη γερμανική κομμουνιστική ζώνη και ο κατακερματισμός της, η Αγγλία και οι ΗΠΑ αποφάσισαν τον από κοινού εφοδιασμό της πόλης, είτε μέσω της “συνοδείας” (που μπορούσε να είναι αφορμή για νέες συγκρούσεις), είτε μέσω αερογέφυρας, “σπάζοντας” ή αποφεύγοντας σε κάθε περίπτωση το μπλοκάρισμα από το Βερολίνο. Η δεύτερη λύση που επιλέχθηκε, έμελλε να ξεκινήσει από την αερογέφυρα Luftbrücke στις 25 Ιουνίου του 1948. Αυτή υπάχθηκε στα προβλήματα, πάνω στα οποία εργάστηκε η SCOOP το Δεκέμβρη του ίδιου έτους, π.χ. μπορούσε να μεταφέρει 4500 τόνους ημερησίως και μετά από εντατικές μελέτες βασισμένες στην Επιχειρησιακή Έρευνα, βελτιστοποίησε την προμήθεια για να πάρει (το σχεδόν διπλάσιο!) 8000-9000 τόνους ημερησίως το Μάρτη του 1949. Η ποσότητα αυτή ήταν η ίδια και για τα επίγεια μέσα, γι’ αυτό η Σοβιετική αποφάσισε να αναστείλει το μπλοκάρισμα στις 12 Μαΐου 1949.

Μετά το 2^{o} Παγκόσμιο Πόλεμο, το πρόβλημα της παραγγελίας των πόρων των ΗΠΑ (όπως ενέργεια, εξοπλισμός και όλα τα είδη των προμηθειών) επιλύθηκε με μοντέλα βελτιστοποίησης και κυρίως με την παρέμβαση του **Γραμμικού Προγραμματισμού**, όπου βέβαια, ο χρόνος επίλυσης των προβλημάτων έχει μειωθεί σημαντικά, λόγω της παράλληλης ανάπτυξης Επιχειρησιακής Έρευνας, υπολογιστών και τεχνικών υπολογισμού.

Αξίζει να αναφέρουμε ότι το πρώτο αποτέλεσμα των τεχνικών αυτών προέκυψε το 1952, όταν ένας υπολογιστής SEAC χρησιμοποιήθηκε στο Εθνικό Γραφείο Προτύπων (National Bureau of Standards), με στόχο την εξεύρεση λύσης σε προβλήματα επιχειρησιακού χαρακτήρα. Το γεγονός αυτό (ο εντοπισμός, δηλαδή, της λύσης), ήταν τόσο ενθαρρυντικό, που αμέσως χρησιμοποιήθηκε για κάθε είδους στρατιωτικό πρόβλημα, ξεκινώντας με τον καθορισμό του βέλτιστου ύψους που θα έπρεπε να πετούν τα αεροπλάνα για να εντοπίζουν τα υποβρύχια του εχθρού ή το βάθος στο οποίο θα πρέπει να στείλουν τα φορτία για να φτάσουν τα εχθρικά υποβρύχια και να προκαλέσουν τις μεγαλύτερες δυνατές απώλειες μέχρι την τεχνικοοικονομική διαχείριση για τον εφοδιασμό και εξοπλισμό, παραδείγματα που μεταφράστηκαν σε πέντε φορές αύξηση της αποτελεσματικότητας της Πολεμικής Αεροπορίας.

Το ενδιαφέρον για την Επιχειρησιακή Έρευνα και η ανάπτυξή της κορυφώθηκε κατά τη διάρκεια της δεκαετίας ’50-’60 κι αυτό, λόγω της εφαρμογής της στο χώρο της βιομηχανίας και του εμπορίου. Αυτό μπορούμε να το αντιληφθούμε καλύτερα αν αναλογιστούμε την περίπτωση της πόλης της Μόσχας: το πρόβλημα της βέλτιστης μεταφοράς άμμου για την κατασκευή των έργων διαπαιδαγώγησης της πόλης που είχε δέκα σημεία προέλευσης και 230 προορισμούς. Για την επίλυση του προβλήματος αυτού, χρησιμοποιήθηκε ένας υπολογιστής Strena επί δέκα ημέρες το μήνα Ιούνιο του 1958 και η λύση αυτή συνέβαλε στη μείωση των δαπανών σε ποσοστό 11% σε σχέση με το αρχικό κόστος.

Προφανώς, η Επιχειρησιακή Έρευνα επικαλύπτεται από άλλους κλάδους, όπως η διοίκηση επιχειρήσεων, η βιομηχανική μηχανική, η ψυχολογία κ.ά. Στοχεύει να επιλύσει σύνθετα προβλήματα μέσω του προσδιορισμού μέγιστου (ελάχιστου) κέρδους/παραγωγής (κόστους/ρίσκου) σε προβλήματα του πραγματικού κόσμου. Έτσι, εκκινώντας από τη βελτιστοποίηση στρατιωτικών επιχειρήσεων (εξ' ου και το Επιχειρησιακή!), η Επιχειρησιακή Έρευνα έχει καταφέρει να αναπτύξει τεχνικές σε πληθώρα βιομηχανιών στις μέρες μας.

Η Επιχειρησιακή Έρευνα περικλείει ένα ευρύ φάσμα τεχνικών επίλυσης προβλημάτων και μεθόδων που εφαρμόζονται με στόχο τη βέλτιστη λήψη αποφάσεων και την αύξηση της αποδοτικότητας. Η μαθηματική βελτιστοποίηση, η προσομοίωση, η θεωρία ουρών αναμονής, τα στοχαστικά μοντέλα, η θεωρία Markov, η ανάλυση αποφάσεων και τα νευρωνικά δίκτυα είναι κάποια ενδεικτικά πεδία εφαρμογής της Επιχειρησιακής Έρευνας. Όλες αυτές οι τεχνικές, βέβαια, περιλαμβάνουν την κατασκευή ενός μαθηματικού μοντέλου που θα περιγράφει πλήρως το πρόβλημα και λόγω της φύσης των προβλημάτων, η Επιχειρησιακή Έρευνα συνδέεται άμεσα με την Επιστήμη των Υπολογιστών (υπολογιστική ανάλυση δεδομένων) και τη Στατιστική (Computer Science & Analytics).

Προηγουμένως, τα προβλήματα αυτά παρουσιάστηκαν ως στρατηγική Εταιριών Έρευνας ή Ανάλυσης, οι οποίες, ωστόσο, δεν έχουν τόσο αποτελεσματικές μεθόδους, όπως αυτές που αναπτύχθηκαν κατά τη διάρκεια του 2^{ου} Παγκοσμίου Πολέμου (π.χ. η μέθοδος Simplex). Και παρότι πολλοί θα φαντάζονταν ότι υπάρχουν ποικίλες εφαρμογές της Επιχειρησιακής Έρευνας στον πόλεμο, αυτό δεν ισχύει. Εφαρμόζεται, αντίθετα, σε προβλήματα όπως η διατροφή της κτηνοτροφίας, η καταμερισμός των καλλιεργήσιμων εκτάσεων στον τομέα της γεωργίας, η μεταφορά αγαθών, η τοποθεσία, η κατανομή του προσωπικού, τα δίκτυα, τα προβλήματα ουρών αναμονής, τα γραφήματα.

Κάποια απλά παραδείγματα που μπορούμε να αναφέρουμε εδώ, είναι αυτά που σχετίζονται με τη βέλτιστη κατανομή πόρων και τις βέλτιστες μεταφορές. Για παράδειγμα, η επιλογή τροφών για σίτιση των εργατών μιας εταιρίας με ελάχιστο κόστος, η ελάχιστη διαδρομή μεταφοράς κιβωτίων νερού από τα εργοστάσια εμφιάλωσης σε κεντρικές αποθήκες, ή λίγο πιο σύνθετα, ο προγραμματισμός σε πραγματικό χρόνο στον πύργο ελέγχου ενός αεροδρομίου, με σκοπό την ελαχιστοποίηση του χρόνου αναμονής των αεροσκαφών και την παροχή μέγιστης ασφάλειας [9].

2 ΔΥΝΑΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Ο Δυναμικός Προγραμματισμός υφίσταται σαν έννοια από την εξαετία 1949-1955, όταν ο Αμερικάνος μαθηματικός Richard Bellman ξεκίνησε να ασχολείται με προβλήματα λήψης αποφάσεων σε πολλά στάδια και τους τρόπους επίλυσής τους. Μολονότι πέθανε το Μάρτιο του 1984, η ιστορία του μπορεί να διατυπωθεί με τα δικά του λόγια, αφού πρόλαβε και άφησε πίσω του την αυτοβιογραφία του με τίτλο: “**Το μάτι του τυφώνα**”. Το καλοκαίρι του 1949, όντας καθηγητής της Αναλυτικής Θεωρίας Αριθμών του Πανεπιστημίου Stanford, ασκούσε και χρέη συμβούλου στην εταιρία RAND. Κατά τη διάρκεια του 2ου Παγκοσμίου Πολέμου, το ενδιαφέρον του επικεντρώθηκε στον τομέα των εφαρμοσμένων μαθηματικών, που αργότερα έγινε γνωστός ως Επιχειρησιακή Έρευνα.

Στο σημείο αυτό αξίζει να εξηγήσουμε πώς μια σειρά από συμπτώσεις υπήρξε καθοριστική για τη σύσταση της επωνυμίας “Δυναμικός Προγραμματισμός”.

Το 1950, στα πλαίσια της συνεργασίας του με τη RAND, προτείνεται στον Bellman να ερευνήσει τις διαδικασίες λήψης πολλαπλών αποφάσεων, πρόταση που βέβαια αποδέχθηκε. Το πρώτο του μέλημα ήταν, λοιπόν, να βρει ένα όνομα για τις διαδικασίες αυτές.

Η δεκαετία του ’50 ήταν μια δύσκολη περίοδος για τη μαθηματική έρευνα. Τότε, διατελούσε Γραμματέας Άμυνας της Ουάσινγκτον, ένας πολιτικός ονόματι Wilson, ο οποίος είχε μια παθολογική φοβία, την έρευνα, και δη τη μαθηματική. Η RAND απασχολούμενη από την Πολεμική Αεροπορία, αναπόφευκτα είχε διευθυντή τον Wilson. Συνεπώς, ο Bellman έχοντας ως επικεφαλής τον Wilson θέλησε να αποκρύψει το γεγονός ότι στην εταιρία ασχολούνταν με μαθηματικά και συνέχισε να αναζητά όνομα για την εργασία του.

Αρχικά, το ενδιαφέρον του επικεντρώθηκε στη σχεδίαση, στη λήψη αποφάσεων και στη σκέψη. Η λέξη “σχεδίαση”, όμως, φάνταζε ακατάλληλη για να περιγράψει όλα όσα επιθυμούσε κι έτσι, αποφάσισε να χρησιμοποιήσει τη λέξη “προγραμματισμός”. Εν συνεχεία, διέκρινε ότι αυτός ο προγραμματισμός ήταν δυναμικός, πολυσταδιακός και χρονικά μεταβαλλόμενος. Προφανώς, η λέξη “δυναμικός” ήχησε στα αυτιά του ως η πλέον κατάλληλη και τη διατήρησε. Έτσι, λοιπόν, καθιερώθηκε ο όρος “**Δυναμικός Προγραμματισμός**”.

Ο Bellman και δύο ακόμη συνεργάτες του, οι Hal Shapiro και Ted Harris, έκαναν μία ολόκληρη μελέτη πάνω σε μία εξίσωση (functional equation), την οποία δε δημοσίευσαν ποτέ. Ο ίδιος μελέτησε μία ιδιαίτερα ενδιαφέρουσα εξίσωση, η οποία κατέληγε σε μια πολυεπίπεδη ανάλυση, τη gold mining equation. Η λύση της δινόταν ευκολότερα όχι από μία άγνωστη συνάρτηση, αλλά από μία πράξη ή μία απόφαση, κάτι που κίνησε την περιέργεια του Bellman, που αντίκριζε για πρώτη φορά το φαινόμενο αυτό. Προσπάθησε, λοιπόν, με τους συνεργάτες του να λύσουν τη γενική περίπτωση του προβλήματος που έμοιαζε μεν με κοινό πρόβλημα Θεωρίας Αριθμών, αλλά πέραν των γνώσεών τους. Κατάφεραν να αναπτύξουν τη συνεχή εκδοχή, η οποία κατέδειξε το λόγο που το αυθεντικό πρόβλημα ήταν τόσο δύσκολο.

2.1 ΑΡΧΗ ΤΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Ο Bellman αποφάσισε να ερευνήσει τρία πεδία: το δυναμικό προγραμματισμό, τη θεωρία ελέγχου και τις διαδικασίες χρονικής υστέρησης. Πρωταρχικό του μέλημα στο Δυναμικό Προγραμματισμό ήταν να τον θέσει σε μια αυστηρή βάση. Παρατήρησε ότι χρησιμοποιούσε συνεχώς την ίδια τεχνική για να εξάγει τον τύπο της συνάρτησης. Την τεχνική αυτή ονόμασε “**Αρχή της βελτιστοποίησης**”.

Δουλεύοντας συστηματικά πάνω στη Θεωρία Ελέγχου και έχοντας ήδη κάποια εμπειρία προβλημάτων Οικονομικών και Επιχειρησιακής Έρευνας ανέπτυξε ως "εργαλείο" του το ***λογισμό των μεταβολών*** **(calculus of variables).** Συνειδητοποίησε ότι απλά προβλήματα χρειάζονταν έναν ευφυή χειρισμό, καθότι δεν υπήρχε καμία σταθερότητα. Έστω και μια μικρή αλλαγή στο πρόβλημα μπορούσε να επιφέρει μια ουσιαστική αλλαγή στη λύση. Γρήγορα, όμως, διαπίστωσε ότι αυτό το εργαλείο δεν ήταν αρκετό για την εξεύρεση της λύσης.

Ξοδεύοντας περισσότερο χρόνο και προσπάθεια στις εξισώσεις του Δυναμικού Προγραμματισμού, ήταν σε θέση να λύσει ορισμένες από αυτές και να προσδιορίσει τις ιδιότητες των συναρτήσεων και την πολιτική που έπρεπε να ακολουθηθεί για άλλες. Ανέπτυξε κάποιες καινούργιες θεωρίες, όπως τις Μαρκοβιανές διαδικασίες λήψης αποφάσεων και μπορούσε να ερμηνεύσει εκ νέου μια παλιά θεωρία, όπως το λογισμό μεταβολών. Με αυτή τη θεωρία λυνόταν ένας μεγάλος αριθμός μαθηματικών μοντέλων του Δυναμικού Προγραμματισμού, ωστόσο, δεν ήταν πάντοτε δεδομένη η εύρεση της λύσης, λόγω των περιορισμών ή της γραμμικότητας. Η λύση, άλλωστε, δεν είναι απλά συναρτήσεις χρονικά εξαρτώμενες ή αριθμοί, αλλά ένας κανόνας που λέει σε αυτόν που λαμβάνει την απόφαση πώς να πράξει: είναι δηλαδή μία πολιτική.

Αφού, όπως ειπώθηκε και παραπάνω, κάθε αλλαγή στο πρόβλημα επέφερε και αλλαγή στη λύση, αυτό σήμαινε ότι για να βρεθεί αριθμητική λύση με αποτελεσματικό τρόπο, ήταν απαραίτητα και κάποια άλλα εργαλεία. Και παρά την απροθυμία του Bellman να ασχοληθεί σοβαρά με το λογισμό μεταβολών, το ερώτημα που παρέμενε ακόμα ήταν πώς θα μπορούσε κανείς να καταλήξει σε αριθμητική λύση ενός προβλήματος βελτιστοποίησης και αν οι όποιες μέθοδοι υπήρχαν ήταν αξιόπιστες.

Το θέμα ήταν ότι δεν ήθελε να καταπιαστεί με αυτό το ερώτημα και σίγουρα δεν είχε σκεφτεί να εφαρμόσει το Δυναμικό Προγραμματισμό για τον έλεγχο διαδικασιών **ντετερμινιστικού** τύπου. Μάλλον περισσότερο είχε αναπτύξει τη θεωρία σαν εργαλείο για **στοχαστικές** διαδικασίες λήψης αποφάσεων. Συμπέρανε ότι υπάρχει μία πολύ ενδιαφέρουσα αλληλεπίδραση μεταξύ δυναμικού προγραμματισμού και θεωρίας ελέγχου. Έτσι, μπορούσε να χειριστεί σε κάποια έκταση διαδικασίες ντετερμινιστικού τύπου και στοχαστικές διαδικασίες στα οικονομικά και την επιχειρησιακή έρευνα.

Στη συνέχεια, παρατήρησε ότι η χρήση του λογισμού μεταβολών στη θεωρία ελέγχου προϋποθέτει, έστω και σιωπηρά, ότι έχουμε αιτία και αποτέλεσμα υπό έλεγχο, ότι γνωρίζουμε τόσο το σκοπό, όσο και τη διάρκεια της διαδικασίας ελέγχου. Υπονοείται η υπόθεση ότι κανείς ξέρει τί να παρατηρήσει και ότι οι μεταβλητές κατάστασης μπορούν να μετρηθούν με αυθαίρετη ακρίβεια. Στον πραγματικό κόσμο, καμία από αυτές τις υποθέσεις δεν είναι ομοιόμορφα έγκυρη. Συχνά, οι άνθρωποι αναρωτιούνται γιατί τα μαθηματικά και οι υπολογιστές δε χρησιμοποιούνται για το χειρισμό ουσιαστικών προβλημάτων για την κοινωνία. Η απάντηση στο ερώτημά τους είναι ότι δε γνωρίζουμε πώς να περιγράψουμε τα πολύπλοκα συστήματα της κοινωνίας που περιλαμβάνουν και τους ανθρώπους, δεν καταλαβαίνουμε αίτιο και αποτέλεσμα, δηλαδή, τις συνέπειες των αποφάσεων. Δεν καταλαβαίνουμε καν πώς να κάνουμε τους στόχους μας ακριβέστερους. Καμία από τις απαιτήσεις της κλασικής επιστήμης δεν πληρείται και τελικά, αναπτύσσεται σταδιακά μια νέα μεθοδολογία για την αντιμετώπιση αυτών των ασαφών προβλημάτων, ο δρόμος της οποίας είναι αρκετά δύσκολος.

Η πολυπλοκότητα που παρουσιάζει ο πραγματικός κόσμος ήταν ένας καλός λόγος για να επιστρέψει ο Bellman στη Θεωρία Αριθμών, η οποία, όμως, μάλλον δεν ήταν η καλύτερη προσέγγιση για τόσο δύσκολα προβλήματα από πλευράς απόδοσης. Γι' αυτό και αποφάσισε να στραφεί σε ντετερμινιστικές διαδικασίες ελέγχου και να τις τροποποιήσει ανά στάδιο, ώστε να αποκτήσει θεωρίες για την αντιμετώπιση βασικών αβεβαιοτήτων με έναν πιο εκλεπτυσμένο τρόπο.

Είναι σκόπιμο να αναφέρουμε εδώ, ότι οι πιθανότητες που παρουσιάζουν “καλή συμπεριφορά” (well-behaved) αντιμετωπίζονται με τη βοήθεια της κλασικής Θεωρίας Πιθανοτήτων που οδηγεί στη σύγχρονη θεωρία στοχαστικών διαδικασιών ελέγχου, όπου η αβεβαιότητα εκπροσωπείται από τυχαίες μεταβλητές με γνωστές κατανομές πιθανότητας και όπου στόχος είναι η μεγιστοποίηση των αναμενόμενων τιμών.

2.2 ΚΛΑΣΙΚΟΣ ΟΡΙΣΜΟΣ

Ο Δυναμικός Προγραμματισμός είναι μία γενική προσέγγιση κατασκευής μιας ακολουθίας αλληλένδετων αποφάσεων με βέλτιστο τρόπο. Τα γενικά χαρακτηριστικά μπορούν να περιγραφούν εύκολα, οι λεπτομέρειες, όμως, έχουν άμεση εξάρτηση από τις εκάστοτε πρακτικές εφαρμογές.

Η μέθοδος είναι <u>επαναληπτική</u> και μπορούμε να την παρομοιάσουμε με μία υπολογιστική ρουτίνα, συνοδευόμενη από μία στοίβα. Η στοίβα αποτελεί τη δομή στην οποία αποθηκεύεται συνεχώς πληροφορία, καθώς η ρουτίνα εκτελούμενη καλεί αναδρομικά τον εαυτό της. Με την ολοκλήρωσή της, η λύση κείτεται στις υπο-δομές της στοίβας, απ’ όπου και μπορεί να εξαχθεί το συμπέρασμά μας.

Ακολουθεί μία επισκόπηση της μεθόδου που μόλις περιγράψαμε:

1) Έχουμε το αρχικό πρόβλημα και απομονώνουμε ένα μικρό “κομμάτι” αυτού. Βρίσκουμε μία βέλτιστη λύση γι’ αυτό το κομμάτι.

2) Επεκτείνουμε ελαφρώς το κομμάτι αυτό και βρίσκουμε τη βέλτιστη λύση για το νέο πρόβλημα, κάνοντας χρήση της προηγούμενης βέλτιστης λύσης (βήμα 1).

3) Συνεχίζουμε με το βήμα 2, μέχρις ότου η υπό εξέταση περιοχή έχει επεκταθεί τόσο, ώστε το τρέχον πρόβλημα να περικλείει το αρχικό πρόβλημα. Όταν το πρόβλημα λυθεί, θα έχουμε φτάσει στις συνθήκες τερματισμού.

4) Τελικά, η λύση ολόκληρου του προβλήματος εντοπίζεται με τη σύνθεση των βέλτιστων λύσεων των επιμέρους μικρών κομματιών.

Ας φανταστούμε, λοιπόν, ένα μυρμήγκι που ταξιδεύει στους λαβύρινθους της φωλιάς του. Πώς άραγε μπορεί να μεταβεί από το σημείο Α στο σημείο Β; Αυτό είναι ένα πρόβλημα αμιγώς δυναμικού προγραμματισμού που η φύση λύνει με τους δικούς της μοναδικούς μηχανισμούς. Για τον άνθρωπο, όμως, ένα ανάλογο αυτού του προβλήματος είναι η δρομολόγηση της βασικής μονάδας επικοινωνίας του διαδικτύου, του πακέτου IP. Πώς, ξεκινώντας ένα πακέτο IP από τον υπολογιστή μας, φτάνει άραγε στον διαδικτυακό του προορισμό; Τις απαρχές της λύσης αυτού του προβλήματος έθεσε το 1956 ο Ολλανδός μαθηματικός Edsger W. Dijkstra. Ο ομώνυμος αλγόριθμος στοχεύει στην εύρεση των συντομότερων διαδρομών μεταξύ των κόμβων ενός γράφου. Απαντάει, δηλαδή, στο ερώτημα του μυρμηγκιού και του πακέτου IP που θέσαμε παραπάνω.

Επιστρέφουμε πάλι στα 4 βήματα. Το μικρό κομμάτι του προβλήματος που απαντάμε σε κάθε στάδιο μάς καθορίζει τον πλησιέστερο στον αρχικό κόμβο. Επεκτείνουμε ελαφρώς αυτό το κομμάτι σε κάθε στάδιο, προσαρτώντας όλους τους έως τώρα μη κατειλημμένους (unsolved) κόμβους και βέλη που συνδέονται άμεσα με έναν κατειλημμένο κόμβο. Τα κριτήρια τερματισμού επαληθεύονται όταν ο επόμενος εγγύτερος κόμβος είναι ο προορισμός μας. Ανακτούμε τη λύση με προς-τα-πίσω αναδρομή (backward induction) στα τόξα (του συνόλου των τόξων) από τον κόμβο-προορισμό στον αρχικό κόμβο.

Ο ***Δυναμικός Προγραμματισμός*** *(Dynamic Programming)* είναι γενικά μία μέθοδος που λύνει προβλήματα βελτιστοποίησης, τα οποία συμμετέχουν στη λήψη μιας ακολουθίας αποφάσεων, προσδιορίζοντας, για κάθε απόφαση, υπο-προβλήματα που μπορούν να λυθούν με τέτοιο τρόπο, ώστε μία βέλτιστη λύση του πραγματικού προβλήματος να μπορεί να βρεθεί από βέλτιστες λύσεις των υπο-προβλημάτων.

Είναι δηλαδή μια προσέγγιση βελτιστοποίησης που μετατρέπει ένα σύνθετο πρόβλημα σε μια ακολουθία απλούστερων προβλημάτων. Το βασικό του χαρακτηριστικό είναι η πολυσταδιακή φύση της διαδικασίας βελτιστοποίησης [14][16].

Αυτή η μέθοδος βασίζεται στην ‘**Αρχή βελτιστοποίσης του Bellman[1]’ (1950)**, η οποία διατυπώνεται ως ακολούθως:

«Μία βέλτιστη πολιτική (στρατηγική) έχει την ιδιότητα ότι οποιεσδήποτε κι αν είναι η αρχική κατάσταση και η αρχική απόφαση, οι υπόλοιπες αποφάσεις πρέπει να συνιστούν μια βέλτιστη πολιτική (στρατηγική), όσον αφορά την κατάσταση που προκύπτει από την πρώτη απόφαση».

Ή πιο συνοπτικά, η παραπάνω αρχή υποστηρίζει ότι **«βέλτιστες πολιτικές έχουν βέλτιστες υπο-πολιτικές»**. Η αρχή της βελτιστοποίησης είναι επίσης γνωστή ως **‘βέλτιστη υπο-πολιτική’** στη βιβλιογραφία.

Η εγκυρότητα της αρχής αυτής προκύπτει από το γεγονός ότι αν μία πολιτική δεν έχει βέλτιστη υπο-πολιτική, τότε πιθανή αντικατάσταση της υπο-πολιτικής αυτής από μία βέλτιστη υπο-πολιτική θα μπορούσε να βελτιώσει την πραγματική πολιτική. Βέβαια

[1] Ο Richard Bellman πρωτοπόρησε εισάγοντας τη συστηματική μελέτη του Δυναμικού Προγραμματισμού το 1950.

[2] Στη Θεωρία Υπολογιστικής Πολυπλοκότητας, μια κατηγορία πολυπλοκότητας είναι ένα σύνολο από προβλήματα που σχετίζονται με βάση την πολυπλοκότητα τους. Η κατηγορία NP είναι ένα σύνολο προβλημάτων απόφασης των οποίων οι λύσεις μπορούν να καθοριστούν από μια μη-ντετερμινιστική μηχανή Turing σε πολυωνυμικό χρόνο.

[3] Αν και συνήθως ταυτίζουν τη μέθοδο “διαίρει και βασίλευε” με τη μέθοδο του Δυναμικού Προγραμματισμού, υπάρχει μια σημαντικότατη διαφορά. Η πρώτη διαιρεί το πρόβλημα σε ανεξάρτητα υπο-προβλήματα τα οποία

αυτό μπορεί να οδηγήσει σε ατέρμονη αναδρομή, οπότε και να καθορίσει ένα πρόβλημα ως NP-πλήρες[2]!

Εν κατακλείδι, ο Δυναμικός Προγραμματισμός είναι μία εξαιρετικά χρήσιμη μαθηματική τεχνική για τη λήψη μιας ακολουθίας αλληλένδετων αποφάσεων. Παρέχει δε μια συστηματική διαδικασία για τον προσδιορισμό του βέλτιστου συνδυασμού αποφάσεων. Η αλληλεξάρτηση αυτή μπορεί να προκύψει είτε επειδή οι αποφάσεις παρουσιάζουν κάποια χρονική διαδοχή (όπως στην περίπτωση αναζήτησης του συντομότερου μονοπατιού), είτε επειδή συνδέονται με κοινούς περιορισμούς (όπως στην περίπτωση κατανομής πόρων σε ανταγωνιστικές δραστηριότητες π.χ. πρόβλημα σακιδίου, συμπίεση εικόνας).

Σε αντίθεση με το *Γραμμικό Προγραμματισμό (Linear Programming)*, εδώ δεν υπάρχει ένας 'στάνταρ' (καθορισμένος στα ελληνικά) μαθηματικός τύπος για "το" πρόβλημα δυναμικού προγραμματισμού. Αντίθετα, ο Δυναμικός Προγραμματισμός είναι (ένα σχέδιο προσέγγισης και αντιμετώπισης ενός σύνθετου προβλήματος) ένας γενικός τύπος της λύσης του προβλήματος και οι συγκεκριμένες εξισώσεις και αναδρομικές σχέσεις που χρησιμοποιούνται, αλλάζουν άρδην από πρόβλημα σε πρόβλημα, συνεπώς, πρέπει να κατασκευάζονται με τρόπο τέτοιο, ώστε να προσαρμόζονται στην εκάστοτε περίπτωση.

Επομένως, απαιτείται κάποια εφευρετικότητα και διορατικότητα αναφορικά με τη γενική δομή των προβλημάτων Δυναμικού Προγραμματισμού, ώστε να είμαστε σε θέση να αναγνωρίζουμε πότε και πώς ένα πρόβλημα μπορεί να επιλυθεί με διαδικασίες αυτής της μεθόδου [10].

[2] Στη Θεωρία Υπολογιστικής Πολυπλοκότητας, μια κατηγορία πολυπλοκότητας είναι ένα σύνολο από προβλήματα που σχετίζονται με βάση την πολυπλοκότητα τους. Η κατηγορία NP είναι ένα σύνολο προβλημάτων απόφασης των οποίων οι λύσεις μπορούν να καθοριστούν από μια μη-ντετερμινιστική μηχανή Turing σε πολυωνυμικό χρόνο.

Ωστόσο, ο Δυναμικός Προγραμματισμός πέρα από μία μέθοδος μαθηματικής βελτιστοποίησης, είναι και μία προγραμματιστική μέθοδος, η οποία εφαρμόζεται τόσο σε διακριτά, όσο και σε συνεχή πεδία ορισμού [11][14].

Μερικές από τις πιο δημοφιλείς εφαρμογές που χρησιμοποιούν το Δυναμικό Προγραμματισμό είναι οι παρακάτω:

- Το Πρόβλημα του Σακιδίου (Knapsack)
- Το Πρόβλημα της Ελάχιστης Διαδρομής (Shortest Path Problem)
- Θεωρία Ουρών (Queuing Theory), Θεωρία Ελέγχου (Control Theory), Βιοπληροφορική (Bioinformatics), Επιχειρησιακή Έρευνα (Operations Research)
- Η Μεγαλύτερη κοινή υπακολουθία (Longest common subsequence)
- Το Πρόβλημα του Αλυσιδωτού Πολλαπλασιασμού Πινάκων (Matrix Chain Multiplication Problem)

κ.ά.

Στην παρούσα μελέτη, θα μας απασχολήσει η μέθοδος επίλυσης τέτοιων προβλημάτων, η οποία βασίζεται στον εντοπισμό κατάλληλου αναδρομικού τύπου που θα συνδέει όλες τις επιμέρους αποφάσεις (τεχνική “διαίρει και βασίλευε[3]”!), ώστε η σύνθεσή τους να μας δίνει στο τέλος τη ζητούμενη απόφαση.

2.3 ΚΟΣΤΟΣ

Για να καλυφθούν όλες οι εκδοχές που προκύπτουν από τη διασύνδεση των επιμέρους προβλημάτων (αποφάσεων), η λύση των υπο-προβλημάτων γίνεται

[3] Αν και συνήθως ταυτίζουν τη μέθοδο “διαίρει και βασίλευε” με τη μέθοδο του Δυναμικού Προγραμματισμού, υπάρχει μια σημαντικότατη διαφορά. Η πρώτη διαιρεί το πρόβλημα σε <u>ανεξάρτητα</u> υπο-προβλήματα τα οποία λύνει αναδρομικά και έπειτα συνδυάζει αυτές τις λύσεις για να λύσει το αρχικό πρόβλημα. Αντίθετα, ο Δυναμικός Προγραμματισμός εφαρμόζεται όταν τα υπο-προβλήματα <u>επικαλύπτονται</u> και έχουν κοινά υπο-προβλήματα.

παραμετρικά, δηλαδή για όλες τις τιμές των εκάστοτε παραμέτρων. Αυτός είναι και ο λόγος που αυξάνεται το υπολογιστικό κόστος της μεθόδου, αν και σε κάθε περίπτωση είναι σίγουρα αρκετά μικρότερο από τη μέθοδο της *ολικής απαρίθμησης*, στην οποία περιλαμβάνονται όλες οι δυνατές λύσεις ενός προβλήματος (βλ. ορισμό παραγράφου 3.1).

Από την άλλη πλευρά, η μέθοδος του Δυναμικού Προγραμματισμού παρουσιάζει μεγαλύτερη ευελιξία από τις παραπάνω μεθόδους, με αρκετά υψηλό, ωστόσο, υπολογιστικό κόστος. Γι' αυτό, συνήθως, πρώτα εξετάζεται αν τα προβλήματα μπορούν να επιλυθούν με τις μεθόδους του Γραμμικού ή του Ακέραιου Προγραμματισμού και στην περίπτωση που αυτό δεν καθίσταται δυνατό, τότε χρησιμοποιείται ο Δυναμικός Προγραμματισμός.

2.4 Η ΟΡΟΛΟΓΙΑ ΤΟΥ ΔΥΝΑΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η δυσκολία του Δυναμικού Προγραμματισμού είναι η ανεύρεση του κατάλληλου μοντέλου που θα μας βοηθήσει να περιγράψουμε μία συγκεκριμένη κατάσταση. Προσπαθούμε συνήθως να επιλύσουμε σύνθετα προβλήματα μέσα από τη λύση υπο-προβλημάτων, να δημιουργήσουμε αναδρομικές σχέσεις κλπ. Εδώ οι μεταβλητές απόφασης μπορεί να είναι ακέραιες ή συνεχείς, οι αντικειμενικές συναρτήσεις και οι περιορισμοί να είναι γραμμικοί ή μη-γραμμικοί και τα δεδομένα μας τυχαίες ή ντετερμινιστικές μεταβλητές με γνωστές συναρτήσεις κατανομής πιθανότητας. Ας δούμε, λοιπόν, πώς εφαρμόζονται τα βασικά του συστατικά του μοντέλου του Δυναμικού Προγραμματισμού με έναν περιορισμό:

maximize $z = \sum_{i=1}^{n} c_i(\mathrm{x}_i)$

υπό τους περιορισμούς: $\sum_{i=1}^{n} a_i x_i \leq b$

$$x_i \geq 0 \text{, ακέραιοι, } i = 1,\ldots,n$$

όπου: b και a_i, $i = 1,\ldots,n$ είναι θετικοί πραγματικοί αριθμοί.

Στόχος είναι να μεγιστοποιήσουμε το συνολικό κέρδος, δεδομένου ότι η συνάρτηση $f_i(x_i)$, $i = 1,\ldots,n$, ορισμένη για $x_i \in 1,2,\ldots,\lfloor b/a_i \rfloor$, δείχνει το κέρδος του i για x_i μονάδες [10][11][12][13][14][20]

Στάδιο (i): το αρχικό μας πρόβλημα χωρίζεται σε n στάδια, όπου αρχικό είναι το στάδιο n και τελικό το στάδιο 1 . Ο δείκτης i μας δίνει το στάδιο στο οποίο βρισκόμαστε κάποια στιγμή, $i = 1,\ldots,n$.

Κατάσταση (s_i): είναι η πιθανή συνθήκη στην οποία μπορεί να βρεθεί το σύστημα σε ένα συγκεκριμένο στάδιο του προβλήματος. Κάθε στάδιο μπορεί να περιλαμβάνει περισσότερες από μία καταστάσεις.

Μεταβλητή απόφασης (x_i): για κάθε στάδιο του προβλήματος υπάρχει μία μεταβλητή απόφασης ή ένα σύνολο μεταβλητών απόφασης.

Συνάρτηση κόστους ($c_i(x_i)$): για την απόφαση x_i, η συνάρτηση αυτή μας δίνει την τιμή στο στάδιο i.

Συνάρτηση βέλτιστης τιμής ($f_i(s_i)$): είναι συνολικά η καλύτερη (μέγιστη/ ελάχιστη) συνάρτηση τιμών από το στάδιο i στο στάδιο n, δεδομένου ότι στο στάδιο i η κατάσταση είναι η s_i.

Βέλτιστη πολιτική απόφασης ($p_i(s_i) = x_i^*$): πρόκειται για τη βέλτιστη σε κάποιο στάδιο απόφαση που εξαρτάται από την κατάσταση. Η διαδικασία επίλυσης προβλημάτων με τη μέθοδο του Δυναμικού Προγραμματισμού μας εξασφαλίζει ότι σε κάθε στάδιο θα βρούμε μία βέλτιστη απόφαση για όλες τις καταστάσεις.

Συνάρτηση μετασχηματισμού ($t_i(s_i, x_i)$): η συνάρτηση αυτή μας δείχνει πώς αλλάζει η κατάσταση στο επόμενο στάδιο, βάσει της τρέχουσας κατάστασης, σταδίου και απόφασης.

π.χ. $s_{i+1} = t_i(s_i, x_i) = s_i - a_i x_i$

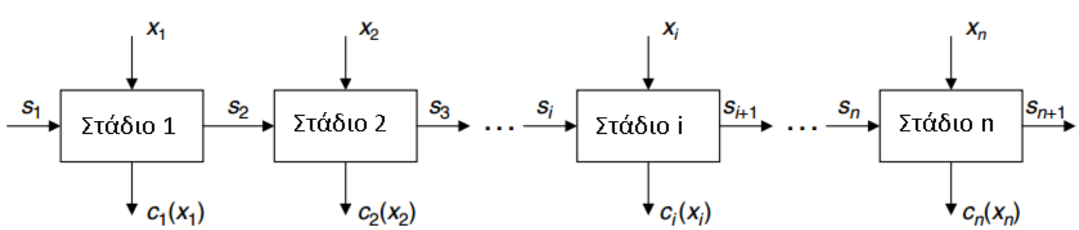
Αναδρομική σχέση: είναι μία σχέση που μας δείχνει τον τρόπο που συνδέεται η βέλτιστη απόφαση στο στάδιο i με τη βέλτιστη απόφαση στο στάδιο $i+1$ (δεδομένου, βέβαια, ότι η τελευταία είναι ήδη γνωστή).

π.χ. $f_i(s_i) = \max\limits_{x_i = 0,1,\ldots,\lfloor s_i/a_i \rfloor} c_i(\mathrm{x}_i) + f_{i+1}(s_i - a_i x_i)$, $s_i = 0,\ldots,b$, $i = 1,\ldots,n$

Συνοριακές συνθήκες: είναι οι αρχικές συνθήκες στο στάδιο n και κάποιες προφανείς συνθήκες που προκύπτουν από τη βέλτιστη συνάρτηση τιμών.

π.χ. $f_n(s_n) = \max\limits_{x_n = 0,1,\ldots,\lfloor s_n/a_n \rfloor} c_n(x_n)$, $s_n = 0,\ldots,b$

Η επεξήγηση του μοντέλου του Δυναμικού Προγραμματισμού που αναλύσαμε παραπάνω, παριστάνεται γραφικά ως ακολούθως:



Σχήμα 1: Το μοντέλο του Δυναμικού Προγραμματισμού

Στο σχήμα 1, κάθε ορθογώνιο αντιστοιχεί σε μία κατάσταση. Η κατάσταση σε ένα συγκεκριμένο στάδιο εξαρτάται από την προηγούμενη απόφαση. Με την οπισθοδρομική αναδρομική σχέση που θα χρησιμοποιήσουμε στο πρόβλημα του network (βλ. παράγραφο 3.1), ξεκινώντας από το στάδιο n, μετακινούμαστε προς τα πίσω και σταματάμε στο στάδιο 1. Όμοια, αν έχουμε προς-τα-εμπρός αναδρομή, χρησιμοποιούμε την αντίθετη διαδρομή, δηλαδή από το στάδιο 1 στο στάδιο n. Σε πολλά προβλήματα, δεν παίζει ρόλο ποια από τις δύο μεθόδους θα χρησιμοποιήσουμε. Σε άλλες, ωστόσο, ανάλογα και με τις συνοριακές συνθήκες, η μία μέθοδος μπορεί να είναι πιο αποδοτική από την άλλη.

Παρατηρούμε δύο βασικά πλεονεκτήματα που μας προσφέρει η μέθοδος αυτή:

1. μετατρέπει το αρχικό πρόβλημα με n μεταβλητές απόφασης σε n υπο-προβλήματα μιας μεταβλητής και
2. βρίσκει το καθολικό βέλτιστο (μέγιστο ή ελάχιστο) αντί για το τοπικό, σε αντίθεση με τις υπόλοιπες μεθόδους βελτιστοποίησης.

Το κλειδί για την επίλυση προβλημάτων Δυναμικού Προγραμματισμού είναι η *διάσταση* του χώρου καταστάσεων, δηλαδή, αν το μοντέλο περιέχει αρκετές μεταβλητές κατάστασης, τότε, μπορεί να εμφανιστούν δυσκολίες στην αποθήκευση των πληροφοριών και στο χρόνο υπολογισμού.

Με τη βοήθεια του Δυναμικού Προγραμματισμού θα λύσουμε το πρόβλημα ελάχιστης καθυστέρησης, εξετάζοντας διαδοχικά από πόσα στάδια ή διασταυρώσεις πρέπει να περάσουμε. Έτσι, κάθε σημείο στο οποίο λαμβάνουμε αποφάσεις ονομάζεται ***στάδιο*** της διαδικασίας λήψης αποφάσεων, ενώ η συσσωρευμένη γνώση που έχουμε σχετικά με το πρόβλημα, ώστε να είμαστε σε θέση να λάβουμε αποφάσεις ονομάζεται ***κατάσταση*** της διαδικασίας λήψης αποφάσεων. Στο πρόβλημα του δικτύου, σε κάθε στάδιο θα πρέπει απλά να γνωρίζουμε την παρούσα διασταύρωση (και όχι το πώς καταλήξαμε σε αυτήν, αφού όπως αναφέραμε και προηγουμένως μας ενδιαφέρει μόνο η τελευταία κατάσταση και όχι το παρελθόν της μεθόδου), ώστε να μπορούμε να λάβουμε τις μετέπειτα αποφάσεις. Η διασταύρωση αυτή αποτελεί την κατάστασή μας για το συγκεκριμένο στάδιο.

Έχοντας, λοιπόν, ορίσει αυτές τις βασικές έννοιες, η λύση μας στο πρόβλημα ελάχιστης καθυστέρησης του network δίνεται από την *αρχή της βελτιστοποίησης*:

"*Κάθε βέλτιστη πολιτική έχει την ιδιότητα ότι, ανεξάρτητα από την τρέχουσα κατάσταση και απόφαση, οι υπόλοιπες αποφάσεις πρέπει να συνιστούν μια βέλτιστη πολιτική σε σχέση με την κατάσταση που προκύπτει από τη σημερινή απόφαση*."

2.5 ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΠΡΟΒΛΗΜΑΤΩΝ ΤΟΥ ΔΥΝΑΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

2.5.1 Τα στάδια

Το βασικότερο χαρακτηριστικό των προβλημάτων Δυναμικού Προγραμματισμού είναι η διάρθρωσή τους σε πολλαπλά *στάδια*, τα οποία επιλύονται διαδοχικά ένα στάδιο σε κάθε βήμα (δηλαδή σε κάθε στάδιο απαιτείται μία πολιτική απόφασης), π.χ. το ***πρόβλημα δικτύου*** (network problem, παρ. 3.1) χωρίζεται σε 6 στάδια και η πολιτική απόφασης σε καθένα από αυτά είναι ποια διαδρομή θα ακολουθηθεί στο επόμενο στάδιο. Παρότι σε κάθε στάδιο το πρόβλημα λύνεται ως ένα απλό πρόβλημα βελτιστοποίησης, αποδεικνύεται πρακτικά ότι η λύση αυτή αναδεικνύει τα χαρακτηριστικά του προβλήματος στο επόμενο στάδιο της ακολουθίας.

Κάποιες φορές τα στάδια αναπαριστούν διαφορετικές χρονικές περιόδους, άλλες είναι χρονικά ανεξάρτητα. Στο συγκεκριμένο παράδειγμα, το πρόβλημα επιλογής της ελάχιστης διαδρομής για έναν κάτοικο από το σπίτι του έως το χώρο στάθμευσης (πάρκινγκ) του κέντρου της πόλης προσεγγίστηκε ως ένα πρόβλημα Δυναμικού Προγραμματισμού. Καθώς η πολιτική απόφασης σε κάθε στάδιο υπονοεί τη συσχέτιση δύο διαδοχικών καταστάσεων, εδώ η μεταβλητή απόφασης είναι το αν ο άνθρωπος θα επιλέξει να προχωρήσει πάνω ή κάτω σε κάθε διασταύρωση, ενώ τα στάδια αναπαριστούν τον αριθμό των πιθανών διασταυρώσεων. Τέτοια προβλήματα που προσεγγίζονται ως δυναμικά, με στάδια χρονικά ανεξάρτητα, είναι σχετικά δύσκολα στην επίλυσή τους.

2.5.2 Οι καταστάσεις

Κάθε στάδιο του προβλήματος βελτιστοποίησης συνδέεται με μία κατάσταση. Οι *καταστάσεις* συλλέγουν τις απαραίτητες πληροφορίες, ώστε να εκτιμήσουν τις επιπτώσεις της τρέχουσας πολιτικής απόφασης στις κατοπινές ενέργειες. Είναι, εν γένει, κάποιες πιθανές συνθήκες, στις οποίες μπορεί να βρεθεί το σύστημα σε κάθε στάδιο του προβλήματος, ενώ το πλήθος τους μπορεί να είναι είτε πεπερασμένο είτε άπειρο.

Το γεγονός αυτό καθιστά το ρόλο των καταστάσεων ιδιαίτερα σημαντικό, ίσως τον πιο σημαντικό στα μοντέλα δυναμικού προγραμματισμού. Ωστόσο, δεν υπάρχει κάποιος κανόνας ή μεθοδολογία για τον προσδιορισμό τους. Αυτό επαφίεται στην εμπειρία, εφευρετικότητα και “ενόραση” του εκάστοτε μελετητή του προβλήματος. Αξίζει, όμως, για δική του βοήθεια να λαμβάνει υπόψη τις ακόλουθες παραμέτρους:

1) Οι καταστάσεις πρέπει να περιέχουν ικανή πληροφορία για τη λήψη μελλοντικών αποφάσεων, έχοντας πάντα κατά νου και το “παρελθόν” του μοντέλου.

2) Πρέπει να προτιμάται σχετικά μικρός αριθμός μεταβλητών κατάστασης, καθώς το υπολογιστικό κόστος είναι ιδιαίτερα υψηλό σε περιπτώσεις μοντέλων με άνω των δύο μεταβλητές.

2.5.3 Αναδρομική Βελτιστοποίηση

Το χαρακτηριστικό αυτό του Δυναμικού Προγραμματισμού υπονοεί μια αναδρομική διαδικασία βελτιστοποίησης που βασίζεται στη λύση ενός προβλήματος Ν σταδίων: λύνοντας, δηλαδή, αρχικά ένα μονοσταδιακό πρόβλημα και διαδοχικά συμπεριλαμβάνοντας ένα στάδιο κάθε φορά και λύνοντας μονοσταδιακά προβλήματα μέχρι να βρεθεί η καθολική βέλτιστη λύση. Η διαδικασία αυτή μπορεί να βασιστεί είτε στην προς-τα-πίσω επαγωγή είτε στην προς-τα-εμπρός επαγωγή.

Στη μεν ***προς-τα-πίσω*** *ή* ***οπισθοδρομική επαγωγή*** *(backward induction),* ως αρχικό στάδιο του προβλήματος λαμβάνουμε το τελικό στάδιο και μετακινούμενοι προς τα πίσω ανά ένα στάδιο κάθε φορά βρίσκουμε τη βέλτιστη πολιτική για εκείνο το στάδιο, έως ότου συμπεριληφθούν όλα τα στάδια (δηλαδή μέχρι να βρούμε τη βέλτιστη πολιτική για το αρχικό/πρώτο στάδιο). Στη δε ***προς-τα-εμπρός επαγωγή*** *(forward induction),* επιλέγουμε το πρώτο στάδιο του προβλήματος ως αρχικό και μετακινούμενοι προς τα εμπρός επιλύουμε -όμοια με την προηγούμενη μέθοδο- προβλήματα ενός σταδίου κάθε φορά, έως ότου συμπεριλάβουμε όλα τα στάδια.[4]

Βάση αυτής της αναδρομικής διαδικασίας βελτιστοποίησης αποτελεί η γνωστή πλέον "αρχή της βελτιστοποίησης".

2.6 ΜΙΑ ΓΕΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΤΟΥ ΔΥΝΑΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Στην παράγραφο αυτή θα ακολουθήσει μία σύνοψη σε γενική μορφή όλων όσων ειπώθηκαν έως τώρα, ώστε να είμαστε σε θέση να αντιμετωπίσουμε οποιοδήποτε πρόβλημα δυναμικού προγραμματισμού προκύψει. Έστω μια διαδικασία λήψης πολλαπλών αποφάσεων με κόστος

[4] Σε ορισμένα προβλήματα μπορεί να εφαρμοστεί μόνο μία από τις δύο επαγωγικές μεθόδους, με την προς-τα-πίσω επαγωγή να εφαρμόζεται κατά κύριο λόγο σε προβλήματα με αβεβαιότητα.

$$f_n(d_n, s_n) \quad \textbf{(5)}$$

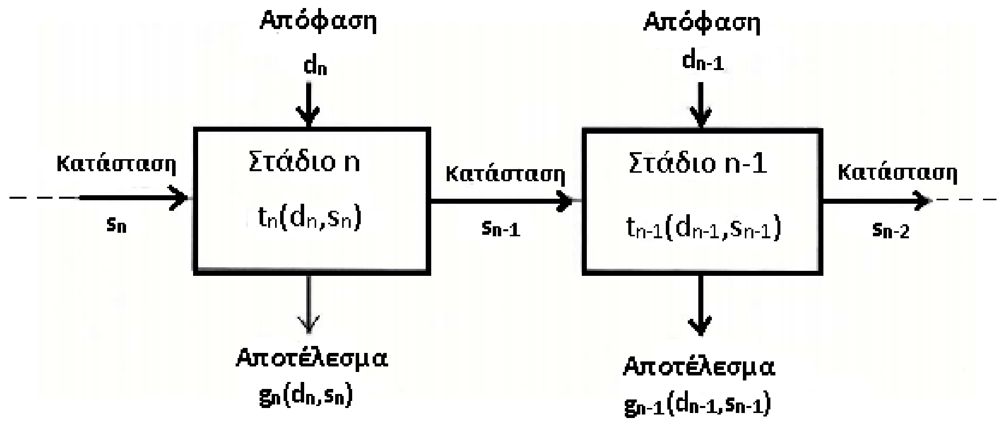
για ένα συγκεκριμένο στάδιο, όπου: d_n είναι μία πιθανή απόφαση που μπορεί να ληφθεί μέσα από ένα σύνολο δυνατών αποφάσεων D_n [5], ενώ η s_n είναι μία κατάσταση της διαδικασίας αυτής όταν απομένουν ακόμη n στάδια από τα N στάδια συνολικά. Η επόμενη κατάσταση εξαρτάται μόνο από την παρούσα κατάσταση (δε χρειάζεται να έχουμε γνώση των προηγούμενων καταστάσεων (Μαρκοβιανή ιδιότητα!)), καθώς και από την παρούσα απόφαση που θα ληφθεί. Έτσι, για δεδομένη κατάσταση s_n με n στάδια να απομένουν, η αναδρομική σχέση για $n-1$ στάδια μπορεί να γραφεί ως:

$$s_{n-1} = t_n(d_n, s_n) \quad \textbf{(6)}$$

όπου: d_n είναι η απόφαση που λαμβάνεται για την τρέχουσα κατάσταση και στάδιο.

Το Σχήμα 2 περιγράφει ακριβώς τούτο: δεδομένης μίας κατάστασης s_n, με n στάδια να απομένουν, η απόφαση d_n που θα λάβουμε θέλουμε να μεγιστοποιεί συνολικά τα αποτελέσματα που θα πάρουμε στα υπόλοιπα στάδια. Όπως φαίνεται και από το σχήμα, η απόφαση d_n στο στάδιο s_n μας δίνει αποτέλεσμα $g_n(d_n, s_n)$ και καταλήγει σε μια καινούρια κατάσταση s_{n-1} με $n-1$ στάδια να απομένουν, η οποία δίνεται από τη σχέση (6). Ας προσέξουμε εδώ ότι τα αποτελέσματα που προκύπτουν σε κάθε στάδιο είναι ανεξάρτητα μεταξύ τους [17].

[5] Κανονικά, επειδή το σύνολο D_n σε ένα στάδιο εξαρτάται από την κατάσταση στο στάδιο s_n θα μπορούσαμε να το συμβολίσουμε και ως $D_n(s_n)$, αλλά χάριν ευκολίας αφήνουμε το συμβολισμό D_n στην παρούσα εργασία.



Σχήμα 2: Διαδικασία αποφάσεων πολλών σταδίων

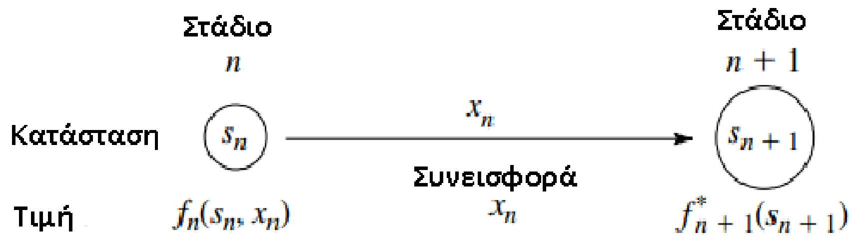
Το πρόβλημα της καταγραφής αποθεμάτων, που περιγράφεται στην παράγραφο 3.2 που ακολουθεί, εξηγεί της έννοιες αυτές.

2.7 ΚΑΤΗΓΟΡΙΕΣ ΠΡΟΒΛΗΜΑΤΩΝ ΔΥΝΑΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Τα προβλήματα δυναμικού προγραμματισμού διακρίνονται σε δύο μεγάλες κατηγορίες: το Ντετερμινιστικό και τον Πιθανοθεωρητικό Δυναμικό προγραμματισμό.

Στα *ντετερμινιστικά* προβλήματα, η κατάσταση του επόμενου σταδίου καθορίζεται πλήρως από την κατάσταση και την πολιτική απόφασης του τρέχοντος σταδίου, ενώ στα *πιθανοθεωρητικά* υπάρχει μία κατανομή πιθανότητας όσον αφορά την κατάσταση του επόμενου σταδίου.

Ο ***Ντετερμινιστικός Δυναμικός Προγραμματισμός*** (Deterministic Dynamic Programming) μπορεί να περιγραφεί από το διάγραμμα της εικόνας 1.

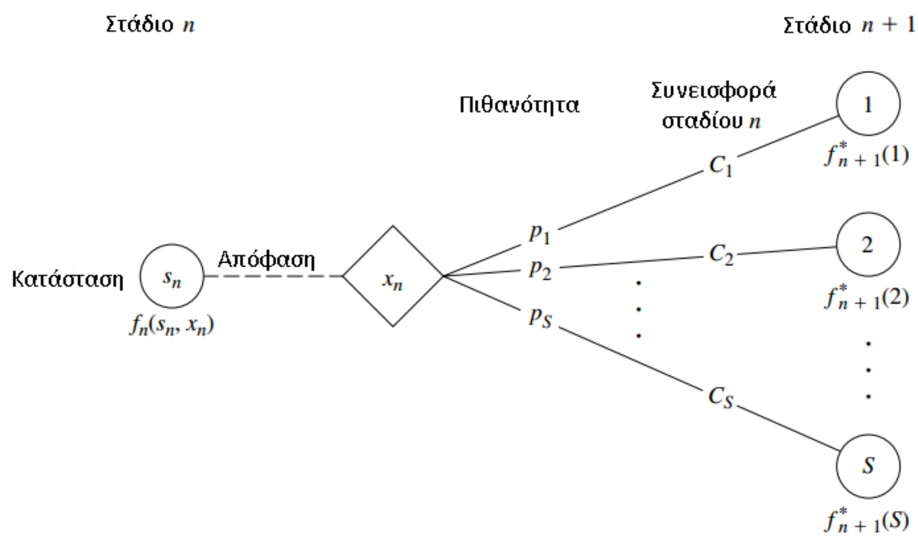


Εικόνα 1: Η δομή του Ντετερμινιστικού Δυναμικού Προγραμματισμού

Εδώ, λοιπόν, φαίνεται ότι στο στάδιο n, η διαδικασία θα βρίσκεται σε κάποια κατάσταση s_n, ενώ μετά τη λήψη της απόφασης x_n μετακινούμαστε σε μια κατάσταση s_{n+1} στο στάδιο $n+1$. Η συνεισφορά στην αντικειμενική συνάρτηση εφαρμόζοντας μια βέλτιστη πολιτική είναι $f^*_{n+1}(s_{n+1})$. Η απόφαση x_n επίσης συνεισφέρει στην αντικειμενική συνάρτηση. Συνδυασμός αυτών των δύο στοιχείων μας δίνει τη συνεισφορά $f_n(s_n, x_n)$ των σταδίων n στην αντικειμενική συνάρτηση και αν βελτιστοποιήσουμε ως προς x λαμβάνουμε $f^*_n = f_n(s_n, x^*_n)$. Αφού βρούμε τα x^*_n και $f^*_n(s_n)$, μπορούμε να εφαρμόσουμε την προς-τα-πίσω επαγωγή και να μετακινηθούμε στο προηγούμενο στάδιο.

Ένας πρώτος τρόπος κατηγοριοποίησης του ντετερμινιστικού Δυναμικού Προγραμματισμού είναι ο *τύπος της αντικειμενικής συνάρτησης*, π.χ. θα μπορούσε να ήταν η ελαχιστοποίηση της συνολικής συνεισφοράς των σταδίων (βλ. πρόβλημα δικτύου) ή η μεγιστοποίηση αυτής της ποσότητας. Ένας άλλος τρόπος κατηγοριοποίησης αφορά τη φύση του *συνόλου των καταστάσεων για τα αντίστοιχα στάδια*, δηλαδή οι καταστάσεις s_n θα μπορούσαν να αντιπροσωπεύονται από μία διακριτή μεταβλητή κατάστασης (όπως π.χ. στο πρόβλημα δικτύου), ή από μία συνεχή μεταβλητή, ή ακόμα από ένα διάνυσμα.

Στον ***Πιθανοθεωρητικό*** ή ***Στοχαστικό Δυναμικό Προγραμματισμό***, αντίθετα από τον ντετερμινιστικό, η κατάσταση στο επόμενο στάδιο δεν προσδιορίζεται πλήρως από την κατάσταση και την πολιτική απόφασης του τρέχοντος σταδίου. Εδώ υπάρχει μία κατανομή πιθανότητας για το ποια θα είναι η επόμενη κατάσταση, η οποία προσδιορίζεται πλήρως από την κατάσταση και την πολιτική απόφασης του τρέχοντος σταδίου. Η δομή του στοχαστικού δυναμικού προγραμματισμού μπορεί να περιγραφεί από το διάγραμμα της εικόνας 2.



Εικόνα 2: Η δομή του Στοχαστικού Δυναμικού Προγραμματισμού

Συμβολισμός:

S : ο αριθμός των δυνατών καταστάσεων στο στάδιο $n+1$ (οι καταστάσεις παριστάνονται με κύκλο (στα δεξιά με $1, 2, \ldots, S$))

p_i : η πιθανότητα με την οποία το σύστημα μεταβαίνει στην κατάσταση i για $i = 1, 2, \ldots, S$

s_n : η κατάσταση στο στάδιο n

x_n : η απόφαση στο στάδιο n (η απόφαση παριστάνεται με ρόμβο)

C_i : η συνεισφορά του σταδίου n στην αντικειμενική συνάρτηση, αν το σύστημα μεταβεί στην κατάσταση i

Μερικές φορές, προκειμένου να συμπεριληφθούν όλες οι καταστάσεις και αποφάσεις, επεκτείνουμε το διάγραμμα και δημιουργούμε το λεγόμενο ***δέντρο αποφάσεων*** *(decision tree).*

Στην περίπτωση του πιθανοθεωρητικού δυναμικού προγραμματισμού, η σχέση μεταξύ $f_n(s_n, x_n)$ και $f^*_{n+1}(s_{n+1})$ είναι σαφώς πιο περίπλοκη απ' ό,τι στην περίπτωση του ντετερμινιστικού δυναμικού προγραμματισμού και εξαρτάται από τον τύπο της καθολικής αντικειμενικής συνάρτησης. Αν π.χ. θέλουμε να ελαχιστοποιήσουμε το αναμενόμενο άθροισμα των συνεισφορών των σταδίων, τότε, η $f_n(s_n, x_n)$ αντιπροσωπεύει το αναμενόμενο άθροισμα του σταδίου n, όπου s_n και x_n η κατάσταση και η απόφαση στο στάδιο αυτό. Επομένως, έχουμε:

$$f_n(s_n, x_n) = \sum_{i=1}^{S} p_i \; C_i + f^*_{n+1}(i) \; ,$$

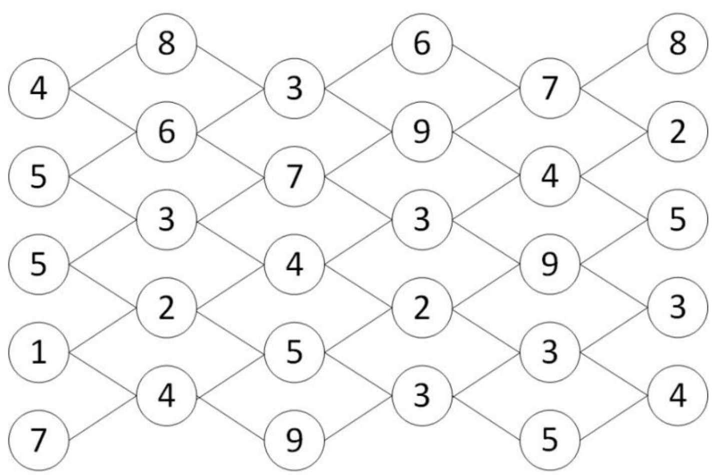
με $f^*_{n+1}(i) = \min_{x_{n+1}} f_{n+1}(i, x_{n+1})$, όπου η ελαχιστοποίηση γίνεται για τις δυνατές τιμές του x_{n+1} [10].

3 ΔΙΑΣΗΜΑ ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ ΔΥΝΑΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

3.1 ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΔΙΚΤΥΟΥ

Το πρόβλημα του δικτύου θα μας βοηθήσει να γνωρίσουμε τη μέθοδο του Δυναμικού Προγραμματισμού και να προσεγγίσουμε τη λύση του που θα προκύψει από τις επιμέρους λύσεις των προβλημάτων σε κάθε στάδιο.

Ουσιαστικά, πρόκειται για έναν οδικό χάρτη, στον οποίο παρουσιάζονται όλες οι πιθανές διαδρομές που συνδέουν τις κατοικίες και τις θέσεις στάθμευσης (πάρκινγκ) του κέντρου της πόλης κάποιων κατοίκων. Στην εικόνα 3, φαίνονται οι διαδρομές που μπορούν να ακολουθήσουν οι κάτοικοι, με τα τόξα να αντιστοιχούν σε δρόμους και τους κόμβους να αντιστοιχούν σε διασταυρώσεις, ενώ οι αριθμοί μέσα στους κόμβους δείχνουν την καθυστέρηση (σε λεπτά) που υφίσταται οποιοσδήποτε κάτοικος σε καθέναν από αυτούς.



Εικόνα 3: Οδικό δίκτυο με διασταυρώσεις και καθυστερήσεις

Το δίκτυο έχει σχεδιαστεί με τέτοιο τρόπο, ώστε οποιοσδήποτε από αυτούς τους ανθρώπους να διασχίζει υποχρεωτικά πέντε διαφορετικούς δρόμους για να φτάσει από την οικία του στο κέντρο. Επιπλέον, ο συνολικός χρόνος που δαπανεί στις διασταυρώσεις είναι ανεξάρτητος από την πορεία την οποία έχει ακολουθήσει, διότι προφανώς υπάρχει πάντα κάποια καθυστέρηση σε αυτές.

Το πρόβλημα που ανακύπτει εδώ, είναι η εύρεση της ελάχιστης συνολικής καθυστέρησης που μπορεί να υποστεί ένας κάτοικος της πόλης αυτής, μετακινούμενος από το σπίτι του στο εμπορικό κέντρο. Θα υιοθετήσουμε την παρακάτω μέθοδο με χρήση πινάκων που είναι ιδιαίτερα βοηθητική.

Στον πίνακα 1 παρουσιάζεται ακριβώς η εικόνα 3 του δικτύου, αλλά σε μορφή συμπαγούς πίνακα, ώστε να είναι πιο εύκολη η προσέγγιση του προβλήματος μέσω του Δυναμικού Προγραμματισμού. Εδώ, το κάθε κελί του πίνακα αντιστοιχεί σε μία διασταύρωση του δικτύου. Μετακινούμενος, λοιπόν, κάθε κάτοικος από την οικία του έως το κέντρο, είναι σαφές ότι η διαδρομή του θα ξεκινάει από αριστερά προς τα δεξιά κατά μήκος του διαγράμματος και η κίνησή του σε κάθε στάδιο θα γίνεται μονάχα σε κάποιο γειτονικό κελί της επόμενης στήλης στα δεξιά. Συνεπώς, όταν θα λέμε ότι απομένουν ακόμα n στάδια να συμβούν, θα εννοούμε τον υπολειπόμενο αριθμό διασταυρώσεων που πρέπει να συναντήσει ο μετακινούμενος κάτοικος, ώστε να φτάσει στον προορισμό του, χωρίς φυσικά να συνυπολογίζουμε τη διασταύρωση στην οποία βρίσκεται εκείνη τη χρονική στιγμή.

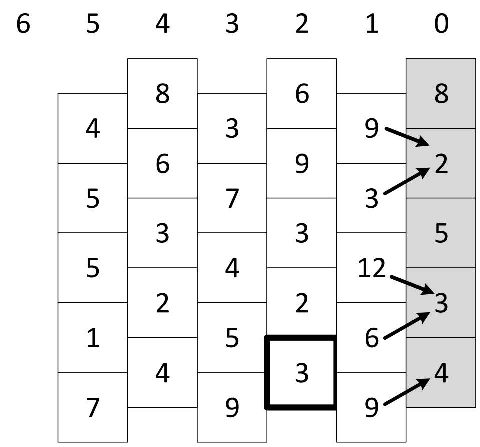
Ένας απλοϊκός τρόπος επίλυσης του ανωτέρω προβλήματος είναι η μέθοδος της ***ολικής απαρίθμησης***, δηλαδή να απαριθμήσουμε όλα τα δυνατά μονοπάτια του διαγράμματος, επιλέγοντας σε κάθε βήμα αυτό που προσφέρει τη μικρότερη καθυστέρηση στη διασταύρωση και τελικά αυτό με την ελάχιστη συνολική καθυστέρηση. Καταλαβαίνουμε αμέσως ότι αυτό απαιτεί υπερβολικό χρόνο και υπολογιστικό κόστος, συνεπώς πρέπει να αποφεύγουμε αυτή τη μέθοδο. Εδώ ακριβώς εμφανίζεται ο Δυναμικός Προγραμματισμός ως μία μέθοδος ***μερικής απαρίθμησης***, ο οποίος μειώνει δραστικά τον αριθμό των υπολογισμών χωρίς να εξετάζει άμεσα όλες τις δυνατές λύσεις, “χτίζοντας” σταδιακά τη βέλτιστη λύση μετακινούμενος από τη μία μεριά του διαγράμματος στην άλλη.

Έστω, για μεγαλύτερη ευκολία, ότι μετακινούμαστε οπισθοδρομικά στο διάγραμμα, δηλαδή ξεκινάμε από τη δεξιά μεριά με κατεύθυνση προς την αριστερή (βλ. προηγούμενη ενότητα 2.5.3: backward induction). Στον πίνακα 1 που παριστάνεται το καθολικό πρόβλημα, φαίνονται όλες οι πιθανές επιλογές που έχει ένας κάτοικος στο κελί κάθε στήλης, όσο μετακινείται από δεξιά προς τα αριστερά. Στην τελευταία στήλη, προφανώς, δεν υπάρχει καμία εναπομείνασα απόφαση να ληφθεί, άρα απλά υπολογίζεται η καθυστέρηση που υφίσταται ο κάτοικος σε εκείνη τη διασταύρωση. Οι αριθμοί 0,1,…,6 πάνω από τις στήλες του πίνακα υπονοούν τις εναπομείνασες διασταυρώσεις.

6	5	4	3	2	1	0
	4	8	3	6	**7**	8
	5	6	7	9	1	2
	5	3	4	3	9	5
	1	2	5	2	3	3
	7	4	9	3	5	4

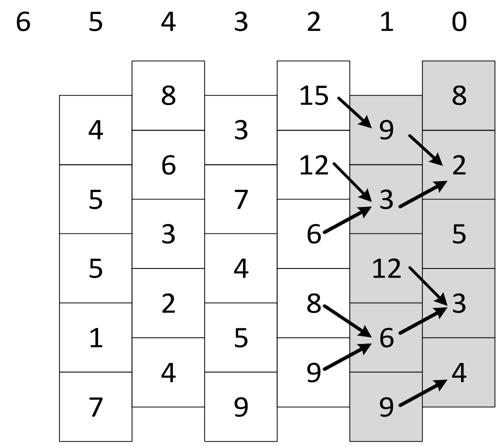
Πίνακας 1: Αναπαράσταση του οδικού δικτύου

Αρχικά αποφασίζουμε για ένα στάδιο ή μία διασταύρωση που θέλουμε να μετακινηθούμε, πάντα από δεξιά προς αριστερά. Έστω, λοιπόν, ότι βρισκόμαστε στο πρώτο κελί της δεύτερης από δεξιά στήλης με το έντονο περίγραμμα. Η καθυστέρηση που υπάρχει σε αυτό τον κόμβο είναι επτά λεπτά, συν την καθυστέρηση των οκτώ ή δύο λεπτών στην προηγούμενη διασταύρωση, ανάλογα με το αν μετακινηθήκαμε κάτω ή πάνω. Άρα, η μικρότερη δυνατή καθυστέρηση που μπορούμε να έχουμε είναι συνολικά 7+2=9 λεπτά. Με τον ίδιο τρόπο, μπορούμε να υπολογίσουμε την ελάχιστη συνολική καθυστέρηση που θα έχουμε σε κάθε κελί της στήλης αυτής (στο πρώτο στάδιο). Οι υπολογισμοί μας δίνουν τον πίνακα 2, με τα βέλη να παριστάνουν τη βέλτιστη σε κάθε μας βήμα απόφαση, δηλαδή τη μετακίνησή μας (πάνω ή κάτω) στην επόμενη διασταύρωση ή στάδιο.



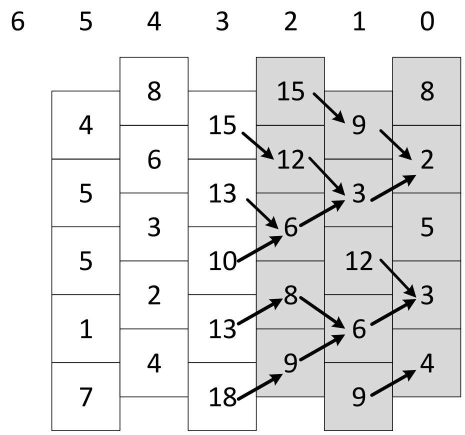
Πίνακας 2: Αποφάσεις και καθυστερήσεις για μία διασταύρωση

Έχοντας διαμορφώσει τον πίνακα 2, παρατηρούμε ότι κάθε κελί της προτελευταίας από δεξιά στήλης, μας δίνει τη συνολική καθυστέρηση των δύο τελευταίων στηλών. Το σημαντικό εδώ είναι ότι αρκεί να γνωρίζουμε <u>μόνο</u> την τελευταία κατάσταση για να λάβουμε την επόμενη προς τα αριστερά απόφαση, π.χ. στη διασταύρωση επτά του πρώτου σταδίου (του πίνακα 1) η μικρότερη καθυστέρηση που μπορούμε να πετύχουμε αν προσμετρήσουμε και την καθυστέρηση της τρέχουσας διασταύρωσης είναι εννέα λεπτά. Επομένως, ο κανόνας μας λέει ότι ***"για κάθε επόμενη απόφαση (από τη δεύτερη στήλη κι έπειτα) από τα δεξιά προς τα αριστερά, η προηγούμενη στήλη μπορεί να αγνοηθεί".***

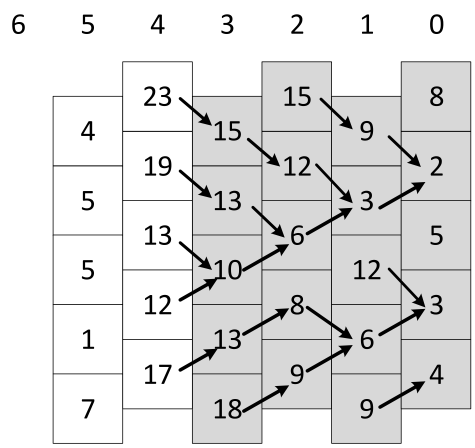


Πίνακας 3: Αποφάσεις και καθυστερήσεις για δύο διασταυρώσεις

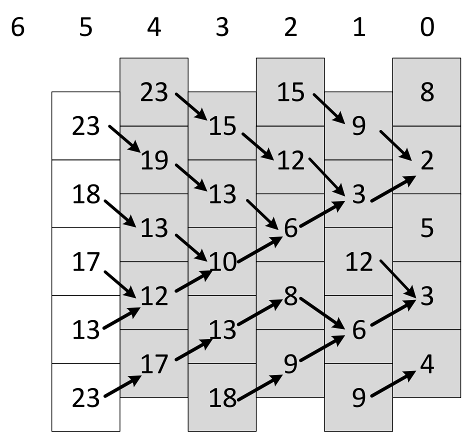
Ας μετακινηθούμε τώρα στο επόμενο στάδιο και ας υπολογίσουμε τις συνολικές καθυστερήσεις κάθε κελιού της τρίτης από δεξιά στήλης, έχοντας περάσει από δύο διασταυρώσεις/στάδια. Αν π.χ. βρισκόμαστε στη διασταύρωση 3 της τρίτης στήλης του δεύτερου πίνακα με έντονο περίγραμμα, η συνολική καθυστέρηση που έχουμε είναι τα τρία λεπτά της συγκεκριμένης διασταύρωσης συν έξι ή εννέα λεπτά, ανάλογα αν μετακινούμαστε πάνω ή κάτω. Αφού εμείς έχουμε ως στόχο να ελαχιστοποιήσουμε τη συνολική μας καθυστέρηση έως εκείνη τη στιγμή, θα μετακινηθούμε προς τα πάνω, έχοντας τελικά καθυστερήσει 3+6=9 λεπτά. Ο υπολογισμός αυτού, καθώς και των υπόλοιπων κελιών της τρίτης στήλης, φαίνεται πλέον στον πίνακα 3, όπου παριστάνονται οι ελάχιστες καθυστερήσεις κάθε διασταύρωσης ενώ έχουμε διανύσει δύο στάδια. Έχοντας αφομοιώσει τη μέθοδο αυτή, μπορούμε κατά παρόμοιο τρόπο να προχωρήσουμε ακόμα μία στήλη προς τα πίσω και να υπολογίσουμε τις βέλτιστες καθυστερήσεις και αποφάσεις για τρία στάδια, στη συνέχεια για τέσσερα κ.ο.κ. και να κατασκευάσουμε τους αντίστοιχους πίνακες.



Πίνακας 4: Αποφάσεις και καθυστερήσεις για τρεις διασταυρώσεις



Πίνακας 5: Αποφάσεις και καθυστερήσεις για τέσσερις διασταυρώσεις



Πίνακας 6: Αποφάσεις και καθυστερήσεις για πέντε διασταυρώσεις

Με την ολοκλήρωση της κατασκευής και του έκτου πίνακα βλέπουμε ότι έχουμε φτάσει στη βέλτιστη λύση μας. Για να εντοπίσουμε την ελάχιστη δυνατή καθυστέρηση που μπορεί να πετύχει ένας κάτοικος, αρκεί να κοιτάξουμε την τελευταία στήλη αυτού του πίνακα. Η απάντηση ασφαλώς είναι 13 λεπτά. Η διαδρομή που θα πρέπει να ακολουθήσει, λαμβάνοντας πάντα τη βέλτιστη απόφαση, όπως μας καταδεικνύουν τα βέλη του εν λόγω διαγράμματος, είναι να ξεκινήσει στη δεύτερη διασταύρωση από τον κόμβο 12 της τέταρτης στήλης και με συνεχή κίνηση προς τα πάνω, δηλαδή περνώντας διαδοχικά από τους κόμβους 10, 6 και 3, να φτάσει τελικά στον κόμβο 2 της τελευταίας στήλης.

Το πιθανό πρόβλημα, βέβαια, που παρουσιάζεται εδώ είναι το σημείο εκκίνησης των κατοίκων. Για το λόγο αυτό, θεωρούμε ότι ο κάθε κάτοικος δε μπορεί να επιλέξει αυθαίρετα την αρχική διασταύρωση στην οποία θα μεταβεί, αντιθέτως υποθέτουμε ότι η οικία του γειτονεύει μόνο με μία εκ των αριστερότερων διασταυρώσεων, στην οποία και θα μετακινηθεί. Άρα, ο κάθε κάτοικος έχει πλέον σταθερό σημείο εκκίνησης και η δυσκολία μας έχει ξεπεραστεί. Η θεώρησή μας αυτή προϋποθέτει ότι οι κάτοικοι δεν ενδιαφέρονται διόλου για το σημείο στάθμευσης που θα παρκάρουν. Τελικά, καταλήγουμε να βρούμε τις διαδρομές ελάχιστης καθυστέρησης από το κέντρο της πόλης προς όλα τα σπίτια των κατοίκων.

Ας δούμε τώρα τι έχουμε καταφέρει. Όχι μόνο λύσαμε το πρόβλημα ελάχιστης καθυστέρησης και βρήκαμε την ενδεδειγμένη διαδρομή για έναν συγκεκριμένο κάτοικο, αλλά μπορέσαμε να ενσωματώσουμε το συγκεκριμένο πρόβλημα στο γενικό πρόβλημα της εύρεσης διαδρομών ελάχιστης καθυστέρησης από το σύνολο των κατοικιών στο σύνολο των χώρων στάθμευσης.

Παρατηρώντας πάλι την τελευταία στήλη του πίνακα 6, προσέχουμε ότι δύο διασταυρώσεις, η οκτώ και η πέντε, δεν είναι προσβάσιμες από κανέναν κάτοικο. Αυτό συμβαίνει διότι κατά την διαδικασία λύσης του προβλήματός μας, στόχος είναι να βρεθούν οι διαδρομές ελάχιστης καθυστέρησης από κάθε σπίτι σε οποιοδήποτε σημείο στάθμευσης και όχι προς κάθε σημείο στάθμευσης (δε μας ενδιαφέρει δηλαδή να καλυφθούν όλες οι θέσεις πάρκινγκ). Για παράδειγμα, ο κάτοικος που υφίσταται συνολική καθυστέρηση 18 λεπτών, έχοντας ακολουθήσει τη βέλτιστη διαδρομή 18 $\rightarrow$ 13 $\rightarrow$ 10 $\rightarrow$ 6 $\rightarrow$ 3 $\rightarrow$ 2, ενδεχομένως θα παρκάρει είτε στη θέση 2 (δεύτερο κελί της τελευταίας στήλης), είτε σε μία από τις δύο γειτονικές της θέσεις σε περίπτωση που αυτή είναι κατειλημμένη [11][15][16].

Στο πρόβλημά μας, ευρισκόμενοι σε μια συγκεκριμένη διασταύρωση με ακόμα n διασταυρώσεις να απομένουν να περάσουμε, η συνάρτηση βέλτιστης τιμής $f_i(s_i)$ υπονοεί την ελάχιστη συνολική καθυστέρηση της τρέχουσας και των διαδοχικών διασταυρώσεων. Η τιμή της $f_i(s_i)$ σε κάθε στάδιο δίνεται από την κατάλληλη στήλη του πίνακα 6. Επίσης, σε κάθε στάδιο, η απόφαση σε μια συγκεκριμένη κατάσταση καθορίζεται από την έως τότε συνολική ελάχιστη καθυστέρηση. Αριθμώντας σε κάθε στάδιο τις καταστάσεις ως $s_n = 1$ την τελευταία (από κάτω) διασταύρωση και $s_n = 6$ την πρώτη (από πάνω) διασταύρωση, η αναδρομική σχέση που λαμβάνουμε είναι:

$$f_n(s_n) = \min \; t_n(\mathrm{s}_n) + f_{n-1}(s_{n-1}) \qquad \textbf{(1)}$$

υπό τον περιορισμό: $s_{n-1} = \begin{cases} s_n + 1, & \text{αν μετακινηθούμε προς τα πάνω με n άρτιο} \\ s_n - 1, & \text{αν μετακινηθούμε προς τα κάτω με n περιττό} \\ \mathrm{s}_n, & \text{διαφορετικά} \end{cases}$,

όπου: $t_n(s_n)$ είναι ο χρόνος καθυστέρησης στη διασταύρωση s_n στο στάδιο n.

Τα κελιά των στηλών του πίνακα 6 προσδιορίζονται χρησιμοποιώντας την εξίσωση

$$f_0(s_0) = t_0(s_0) \; , \; s_0 = 1, 2, ..., 6 \qquad \textbf{(2)}$$

(ξεκινώντας από δεξιά) και εφαρμόζοντας διαδοχικά την εξίσωση (1). Οι βέλτιστες αποφάσεις δίνονται από τα προσανατολισμένα στον πίνακα 6 βέλη.

Όλη αυτή η ανάλυση έγινε χρησιμοποιώντας την προς-τα-πίσω επαγωγή. Εάν τώρα δοκιμάσουμε να κάνουμε χρήση της προς-τα-εμπρός επαγωγής, δηλαδή ξεκινάμε από τα αριστερά και μετακινούμαστε εμπρός ένα στάδιο τη φορά, οι υπολογισμοί θα είναι ανάλογοι, αλλά η ερμηνεία θα διαφέρει ελαφρώς. Για παράδειγμα, εδώ ορίζουμε ως βέλτιστη συνάρτηση τιμής την ελάχιστη καθυστέρηση της τρέχουσας και των ήδη προσπελασμένων διασταυρώσεων, δεδομένου ότι βρισκόμαστε σε μια συγκεκριμένη διασταύρωση και μας απομένουν άλλες n να περάσουμε. Η αναδρομική σχέση στην περίπτωση αυτή είναι:

$$g_{n-1}(s_{n-1}) = \min \; g_n(\mathrm{s}_n) + t_{n-1}(s_{n-1}) \qquad \textbf{(3)}$$

υπό τον περιορισμό: $s_{n-1} = \begin{cases} s_n + 1, & \text{αν μετακινηθούμε προς τα πάνω με n άρτιο} \\ s_n - 1, & \text{αν μετακινηθούμε προς τα κάτω με n περιττό} \\ \mathrm{s}_n, & \text{διαφορετικά} \end{cases}$,

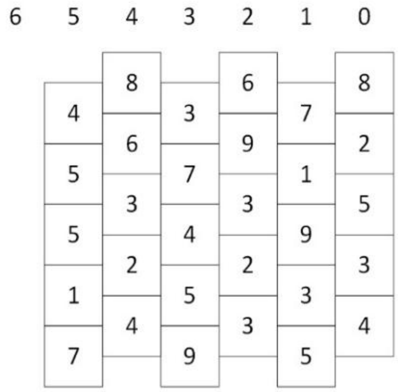
όπου τα στάδια υπονοούν τον αριθμό των διασταυρώσεων/σταδίων που έχουμε ήδη περάσει (και όχι τον αριθμό των σταδίων που πρέπει να περάσουμε). Οι αντίστοιχοι υπολογισμοί της προηγούμενης μεθόδου πραγματοποιούνται τώρα μέσω της σχέσης

$$g_5(\mathrm{s}_5) = t_5(\mathrm{s}_5), \; \mathrm{s}_5 = 1, 2, \ldots, 6 \qquad \textbf{(4)}$$

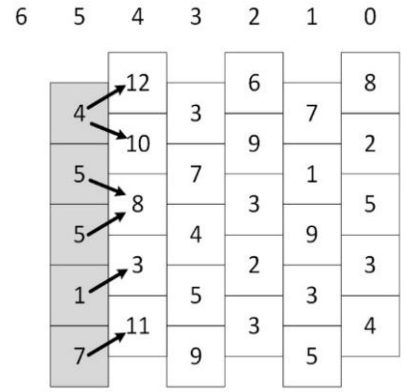
και εφαρμόζοντας διαδοχικά τη σχέση (3).

Σχηματίζουμε τους έξι πίνακες αποφάσεων και καθυστερήσεων, όπως και στη backward induction. Οι στήλες του πίνακα 6’ δίνουν τη συνάρτηση βέλτιστης τιμής του προβλήματος ελάχιστης καθυστέρησης σε κάθε στάδιο και τις ελάχιστες καθυστερήσεις από ένα συγκεκριμένο σημείο στάθμευσης προς το σύνολο των σπιτιών των κατοίκων. Αξίζει να σημειώσουμε ότι αυτή η μέθοδος βρίσκει τη διαδρομή ελάχιστης καθυστέρησης από το σύνολο των θέσεων πάρκινγκ σε *ένα* από τα σπίτια των κατοίκων. Στην πραγματικότητα, βρίσκει τη διαδρομή ελάχιστης καθυστέρησης σε ένα συγκεκριμένο σημείο εκκίνησης αν και μόνο μπορούμε να φτάσουμε σε αυτό από ένα σημείο στάθμευσης με προς-τα-πίσω ακολουθία βελών του πίνακα 6’.

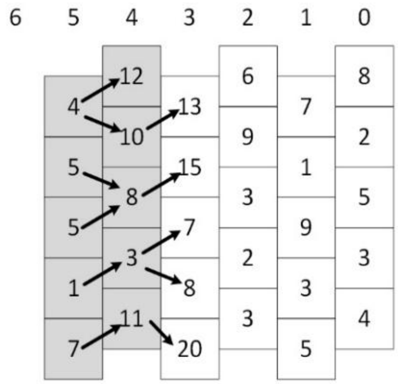
Επιλέγοντας το μονοπάτι ελάχιστης καθυστέρησης των 13 λεπτών στον πίνακα 6’ και ακολουθώντας τη ροή των βελών προς τα πίσω, παρατηρούμε ότι οδηγεί στο προτελευταίο κελί της πρώτης στήλης, δηλαδή ταυτίζεται με το μονοπάτι ελάχιστης καθυστέρησης της backward induction! Συμπερασματικά, η προς-τα-εμπρός αναδρομή βρίσκει τη διαδρομή ελάχιστης καθυστέρησης από κάθε θέση στάθμευσης στο σύνολο των σπιτιών, αντίθετα με την προς-τα-πίσω αναδρομή που βρίσκει τη διαδρομή από κάθε σπίτι στο σύνολο των θέσεων στάθμευσης.



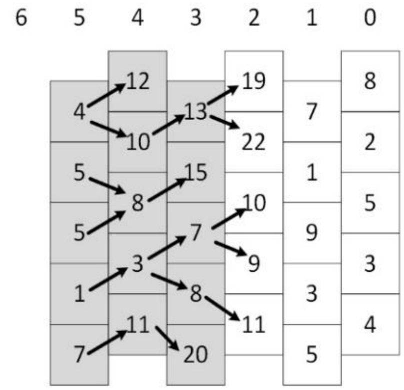
Πίνακας 1'



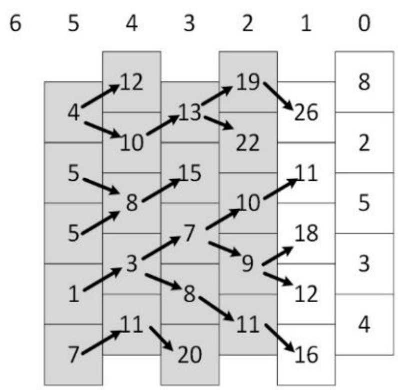
Πίνακας 2'



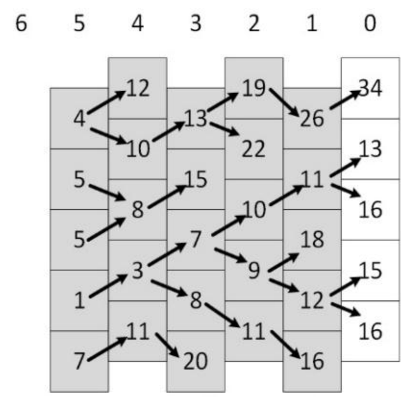
Πίνακας 3'



Πίνακας 4'



Πίνακας 5'



Πίνακας 6'

3.2 Το πρόβλημα της απογραφής αποθεμάτων (The inventory problem)

Στην παράγραφο αυτή θα ασχοληθούμε με το πρόβλημα των αποθεμάτων και το πώς ο Δυναμικός Προγραμματισμός χρησιμοποιείται για την επίλυσή του.

Τα χαρακτηριστικά του προβλήματος αποθεμάτων είναι τα ακόλουθα:

- Ο χρόνος χωρίζεται σε T περιόδους, με την παρούσα περίοδο να τη συμβολίζουμε ως περίοδο 1, την επόμενη ως περίοδο 2, κ.ο.κ και προφανώς, τελευταία θα είναι η περίοδος T. Στην αρχή της περιόδου 1, η ζήτηση σε κάθε περίοδο είναι γνωστή.

- Στην αρχή κάθε περιόδου, η επιχείρηση πρέπει να αποφασίσει πόσες μονάδες θα πρέπει να παραχθούν. Σημειώνεται ότι η παραγωγική ικανότητα κατά τη διάρκεια κάθε περιόδου είναι περιορισμένη.

- Η ζήτηση σε κάθε περίοδο πρέπει να καλύπτεται εγκαίρως τόσο από την παραγωγή, όσο και από την αποθήκη. Καθόλη τη διάρκεια της παραγωγής σε κάθε περίοδο, υπάρχει ένα σταθερό κόστος παραγωγής και προκύπτει και ένα μεταβλητό ανά μονάδα κόστος.

- Η επιχείρηση έχει περιορισμένο χώρο αποθήκευσης. Το ανά μονάδα κόστος αποθήκευσης εμφανίζεται κατά την απογραφή στο τέλος κάθε περιόδου.

- Στόχος της επιχείρησης είναι η ελαχιστοποίηση του συνολικού κόστους παραγωγής που ικανοποιεί τη ζήτηση κάθε περιόδου $1, 2, \ldots, T$.

Στο μοντέλο αυτό, η επιχείρηση επανεξετάζει την απογραφή στο τέλος κάθε περιόδου (συνήθως στο τέλος κάθε μήνα) και εν συνεχεία, αποφασίζει για την παραγωγή. Ένα τέτοιο μοντέλο καλείται ***μοντέλο περιοδικής επανεξέτασης***. Στον αντίποδα, υπάρχει το ***μοντέλο συνεχούς επανεξέτασης***, στο οποίο η επιχείρηση γνωρίζει την κατάσταση των αποθεμάτων και μπορεί να προβεί σε καινούρια παραγγελία ή να ξεκινήσει την παραγωγή οποιαδήποτε χρονική στιγμή.

Ο Δυναμικός Προγραμματισμός χρησιμοποιείται στην περίπτωση αυτή για την κατασκευή του χρονοδιαγράμματος για την παραγωγή που ελαχιστοποιεί το συνολικό κόστος, όπως αυτό προκύπτει από το πρόβλημα της απογραφής των αποθεμάτων[6].

Ακολουθεί η μαθηματική περιγραφή του προβλήματος σε γενική μορφή και αμέσως μετά ένα παράδειγμα που θα βοηθήσει στην καλύτερη κατανόηση όλων όσων περιγράφονται εδώ.

3.2.1 Το πρόβλημα στη γενική του μορφή

Ας υποθέσουμε ότι έχουμε μία επιχείρηση εμπορίας και πώλησης προϊόντων, όπου η κατάσταση s_n παριστάνεται από το μέγεθος των αποθεμάτων (Inventory level) I_n, με n μήνες να απομένουν, βάσει του σχεδιασμού που πραγματοποιήσαμε στην αρχή του έτους. Η απόφαση d_n παριστάνεται από το μέγεθος της παραγγελίας O_n που θα κάνει η επιχείρηση αυτό το μήνα. Έτσι, η αναδρομική σχέση που παίρνουμε για $n-1$ μήνες είναι:

$$I_{n-1} = I_n + O_n - R_n = t_n(I_n, O_n)$$

όπου: R_n είναι η απαίτηση της ζήτησης του προϊόντος της εταιρίας αυτό το μήνα.

Στόχος μας εδώ είναι να ελαχιστοποιήσουμε το κόστος μεταφοράς του εμπορεύματος και τη συνολική παραγγελία.

[6] Αν εξαιρέσουμε το κόστος εγκατάστασης των μονάδων που παράγονται, το πρόβλημα αυτό μπορεί να επιλυθεί και με Γραμμικό Προγραμματισμό.

Αντίθετα, για το πρόβλημα στη γενική του μορφή, το ζητούμενο είναι να μεγιστοποιήσουμε το άθροισμα των αποτελεσμάτων $g_i(d_i, s_i)$ σε κάθε βήμα, με μοναδικές περιοριστικές συνθήκες η απόφαση d_n κάθε σταδίου να ανήκει στο σύνολο των δυνατών αποφάσεων D_n και η αναδρομική σχέση για τη μετάβαση από τη μία κατάσταση στην άλλη να δίνεται από τη σχέση (6). Συνεπώς, το πρόβλημα βελτιστοποίησης υπονοεί την εξεύρεση των βέλτιστων αποφάσεων $d_n, d_{n-1}, \ldots$ (για δεδομένη κατάσταση s_n με n στάδια να απομένουν) για την επίλυση του συστήματος

$$f_n(s_n) = \max \; g_n(d_n, s_n) + g_{n-1}(d_{n-1}, s_{n-1}) + \ldots + g_0(d_0, s_0) \qquad \textbf{(7)}$$

υπό τους περιορισμούς: $\begin{cases} s_{k-1} = t_k(d_k, s_k), & \text{k=1,2,...,n} \\ d_k \in D_k, & \text{k=0,1,...,n} \end{cases}$.

Προφανώς, χρησιμοποιούμε την *προς-τα-πίσω επαγωγή*.

Επαναλαμβάνουμε ότι η $f_n(s_n)$ είναι η βέλτιστη τιμή που μπορεί να προκύψει απ’ όλες τις ακόλουθες αποφάσεις, δεδομένης της κατάστασης s_n με n στάδια να απομένουν. Επειδή, όμως, η τιμή της $g_n(s_n)$ που επιστρέφει εξαρτάται μόνο από την απόφαση d_n και καμία από τις επόμενες διαδοχικές αποφάσεις, μπορούμε κάλλιστα να γράψουμε την παραπάνω σχέση ως εξής:

$$f_n(s_n) = \max \; g_n(d_n, s_n) + \max \; g_{n-1}(d_{n-1}, s_{n-1}) + \ldots + g_0(d_0, s_0) \qquad \textbf{(8)}$$

υπό τους περιορισμούς:

$$\begin{cases} s_{m-1} = t_m(d_m, s_m) \\ d_m \in D_m \end{cases} \; \& \; \left. \begin{matrix} s_{k-1} = t_k(d_k, s_k) \\ d_k \in D_k \end{matrix} \right\}, \quad \text{για} \quad \begin{matrix} k = 1, 2, \ldots, m-1 \\ k = 0, 1, \ldots, m-1 \end{matrix},$$

δηλαδή, πρώτα παίρνουμε το μέγιστο για τη $g_n(s_n)$ για κάθε δυνατή απόφαση d_n κι έπειτα επιλέγουμε d_n τέτοια, ώστε να μεγιστοποιήσουμε την παράσταση $g_{n-1}(d_{n-1}, s_{n-1}) + ... + g_0(d_0, s_0)$.

Η μετάβαση από την κατάσταση n στην κατάσταση $n-1$ μπορεί να φανεί ακόμα καλύτερα, αν παρατηρήσουμε ότι το δεύτερο μέλος της σχέσης (8) ισούται με τη βέλτιστη τιμή της $f_{n-1}(s_{n-1})$, με $n-1$ στάδια να απομένουν (ουσιαστικά δεν κάνουμε τίποτα άλλο παρά μια αντικατάσταση του $n-1$ αντί για n στη σχέση (7)). Επομένως, αυτό που λαμβάνουμε πλέον ως μία πιο επίσημη μορφή της “αρχής της βελτιστοποίησης” είναι η παρακάτω αναδρομική σχέση:

$$f_n(s_n) = \max \; g_n(d_n, s_n) + f_{n-1}(s_{n-1}) \qquad \textbf{(9)}$$

υπό τους περιορισμούς: $\begin{cases} s_{n-1} = t_n(d_n, s_n) \\ d_n \in D_n \end{cases}$

ή ισοδύναμα η:

$$f_n(s_n) = \max \; g_n(d_n, s_n) + f_{n-1} \;\; t_n(d_n, s_n) \qquad \textbf{(10)}$$

υπό τον περιορισμό: $d_n \in D_n$,

όπου υποδηλώνεται ότι, δεδομένης της κατάστασης s_{n-1} που προκύπτει από την τρέχουσα απόφαση, όλες οι επόμενες αποφάσεις πρέπει να είναι βέλτιστες.

Προσέχουμε ότι για το στάδιο 0 του προβλήματος δεν υπάρχουν περαιτέρω στάδια της διαδικασίας προς ανάλυση μετά το τελευταίο, άρα δε μπορούμε να γράψουμε αναδρομική σχέση γι’ αυτό, όπως κάνουμε π.χ. για το $f_n(s_n)$. Γράφουμε απλά

$$f_0(s_0) = \max f_0(d_0, s_0) \quad \textbf{(11)}$$

υπό τον περιορισμό: $d_0 \in D_0$.

Τις περισσότερες φορές δεν υπάρχει μηδενικό στάδιο στο πρόβλημα, αφού το $f_0(s_0)$ είναι μηδέν για όλα τα τελευταία στάδια. Αυτό συνέβη και στο παράδειγμα του δικτύου, όπου ψάχνοντας για τη διαδρομή ελάχιστης καθυστέρησης μέσα από διαδοχικές διασταυρώσεις, το μηδενικό στάδιο ήταν η καθυστέρηση στη διασταύρωση που αντιστοιχεί σε κάθε τελική κατάσταση.

Ως εδώ δουλέψαμε με την προς-τα-πίσω επαγωγή. Εάν τώρα χρησιμοποιήσουμε την *προς-τα-εμπρός επαγωγή*, η $f_n(s_n)$ θα ορίζεται ως η βέλτιστη τιμή όλων των προηγούμενων αποφάσεων, δεδομένου ότι βρισκόμαστε στην κατάσταση s_n με n στάδια να απομένουν. Συνεπώς, έχουμε:

$$f_{n-1}(s_{n-1}) = \max \quad f_n(s_n) + g_n(d_n, s_n) \qquad \textbf{(12)}$$

υπό τους περιορισμούς: $\begin{cases} s_{n-1} = t_n(d_n, s_n) \\ d_n \in D_n \end{cases}$,

όπου συνήθως θέτουμε $f_n(s_n) = 0$ ή βρίσκουμε μια τιμή σε κάποια αρχική συνθήκη, ώστε να αρχικοποιήσουμε τους υπολογισμούς μας.

Ωστόσο, αυτό που έχουμε αμελήσει να αναφέρουμε μέχρι στιγμής είναι το τί ακριβώς συμβαίνει με τις αποφάσεις που λαμβάνουμε σε κάθε στάδιο. Για παράδειγμα, ανήκουν σε ένα σύνολο διακριτό, συνεχές ή ένωση αυτών των δύο; Η απάντηση είναι απλή. Το μόνο απαραίτητο στοιχείο είναι σε κάθε στάδιο η αναδρομική μας σχέση να είναι επιλύσιμη για κάθε βέλτιστη λύση, καθώς και να προσδιορίζεται συνολικά η βέλτιστη λύση στο καθολικό πρόβλημα, όπως αυτή που καταλήξαμε με δύο τρόπους (backward και forward επαγωγή) στο πρόβλημα του δικτύου. Τέτοια προβλήματα βελτιστοποίησης που μπορούν να προσδιοριστούν σε κάθε στάδιο μπορούν να οδηγήσουν στην εφαρμογή ποικίλων τεχνικών, όπως του γραμμικού προγραμματισμού, του ακέραιου προγραμματισμού, της θεωρίας δικτύων κ.ο.κ., ανάλογα πάντα και με τη συνάρτηση μετασχηματισμού, τους περιορισμούς και τη συνάρτηση που θέλουμε να βελτιστοποιήσουμε.

Σημείωση: Σε όλα τα παραπάνω θεωρήσαμε ότι ο αριθμός των καταστάσεων σε κάθε στάδιο ήταν πεπερασμένος, ενώ στην πραγματικότητα οφείλαμε να λύνουμε τις αναδρομικές σχέσεις του προβλήματος για κάθε πιθανή κατάσταση του συστήματος σε κάθε στάδιο. Αν, για παράδειγμα, το σύνολο των δυνατών καταστάσεων ήταν συνεχές και είχαμε απειρία καταστάσεων σε κάθε στάδιο, τότε θα μπορούσαμε να μετατρέψουμε τον αριθμό των καταστάσεων σε πεπερασμένο προσεγγίζοντας το διακριτό σύνολο των δυνατών καταστάσεων κι εφαρμόζοντας έπειτα τις ίδιες διαδικασίες. Επισημαίνουμε ότι οι σχέσεις (9), (10) και (12) ισχύουν ανεξάρτητα από τον αριθμό των καταστάσεων.

3.2.2 Εφαρμογή προβλήματος αποθεμάτων

Έστω μία εταιρεία της οποίας η ζήτηση για το προϊόν της τους επόμενους τέσσερις μήνες είναι γνωστή κι έστω ότι ζητείται 1 μονάδα τον $1^{ο}$ μήνα, 3 μονάδες το $2^{ο}$ μήνα, 2 μονάδες τον $3^{ο}$ μήνα και 4 μονάδες τον $4^{ο}$ μήνα. Στην αρχή κάθε μήνα, η εταιρεία πρέπει να αποφασίζει πόσες μονάδες θα παράγονται τον τρέχοντα μήνα. Το κόστος εγκατάστασης για ένα μήνα παραγωγής είναι 3 χρηματικές μονάδες, ενώ το μεταβλητό κόστος για κάθε μονάδα που παράγεται είναι 1 χρηματική μονάδα. Στο τέλος κάθε μήνα, υπάρχει ένα κόστος αποθήκευσης 0,5 χρηματικές μονάδες για κάθε προϊόν που παραμένει στην αποθήκη. Λόγω περιορισμένης παραγωγικής ικανότητας, κάθε μήνα μπορούν να παραχθούν το πολύ 5 μονάδες, ενώ λόγω μεγέθους της αποθήκης, μπορούν να αποθηκευτούν το πολύ 4 μονάδες. Το ζητούμενο είναι να κατασκευαστεί από την εταιρία ένα σχεδιάγραμμα παραγωγής, το οποίο να ικανοποιεί τις εγκαίρως τη ζήτηση κάθε μήνα και ταυτόχρονα να ελαχιστοποιεί το συνολικό κόστος παραγωγής και αποθήκευσης τους τέσσερις αυτούς μήνες. (Υποθέτουμε ότι δεν υπάρχει καμία διαθέσιμη μονάδα στην αρχή κάθε μήνα.)

Λύση

Όπως σε κάθε πρόβλημα Δυναμικού Προγραμματισμού, έτσι κι εδώ πρέπει να προσδιορίσουμε τα στάδια, τις καταστάσεις, τους περιορισμούς και τις αποφάσεις. Για να προσδιορίσουμε τα στάδια, πρέπει να σκεφτούμε αρχικά ότι όταν απομείνει ένα μόνο στάδιο, το πρόβλημα θα μπορεί να λυθεί. Αν είμαστε π.χ. στην αρχή του 4^{ου} μήνα, η εταιρεία μπορεί με ελάχιστο κόστος να ικανοποιήσει τη ζήτηση, παράγοντας απλώς τις μονάδες που χρειάζονται, ώστε: (η παραγωγή του 4^{ου} μήνα) + (η απογραφή στο τέλος του 3^{ου} μήνα) = (η ζήτηση του 4^{ου} μήνα). Έτσι, είναι πράγματι εύκολο να λυθεί το πρόβλημα όταν απομείνει μόνο ένας μήνας. Ορίζουμε, λοιπόν, το στάδιο να αντιπροσωπεύει το χρόνο[7].

Σε κάθε στάδιο (εδώ: μήνα), η εταιρεία πρέπει να αποφασίσει πόσες μονάδες θα παράγει. Για την απόφασή της αυτή, αρκεί να γνωρίζει το αποθεματικό της επίπεδο στην αρχή του τρέχοντος μήνα ή στο τέλος του προηγούμενου. Ορίζουμε, λοιπόν, την κατάσταση σε κάθε στάδιο να είναι το αρχικό επίπεδο αποθεμάτων. Για να εξασφαλίσουμε ότι πληρούνται έγκαιρα οι απαιτήσεις ζήτησης, θέτουμε τον περιορισμό ότι το αποθεματικό στο τέλος κάθε μήνα είναι μη-αρνητικό.

Ορίζουμε, επίσης, τα βασικά μεγέθη που θα χρειαστούμε στην αναδρομική σχέση ως ακολούθως:

[7] Στα περισσότερα προβλήματα Δυναμικού Προγραμματισμού, τα στάδια έχουν να κάνουν με χρόνο.

$f_t(i)$: το ελάχιστο κόστος για να ικανοποιηθεί η ζήτηση τους μήνες $t, t+1, \ldots, 4$ αν στην αρχή του μήνα t είναι διαθέσιμες i μονάδες

$c(x)$: το κόστος παραγωγής x μονάδων κατά τη διάρκεια μιας περιόδου

$x_t(i)$: το επίπεδο παραγωγής κατά το μήνα t που ελαχιστοποιεί το συνολικό κόστος τους μήνες $t, t+1, \ldots, 4$, αν στην αρχή του μήνα t είναι διαθέσιμες i μονάδες

Θέτουμε $c(0) = 0, \text{ για x>0}$ και $c(\mathrm{x}) = \mathrm{x+3}$.

Οι δυνατές καταστάσεις σε κάθε περίοδο είναι $0,1,2,3,4$, διότι υπάρχει περιορισμένος αποθηκευτικός χώρος και η ζήτηση πρέπει να ικανοποιείται έγκαιρα. Η μέθοδος που θα χρησιμοποιήσουμε είναι η προς-τα-πίσω επαγωγή. Αρχικά, θα βρούμε τα $f_4(i), \text{ για } i = 1,2,3,4$, με βάση αυτά στο επόμενο στάδιο θα βρούμε τα $f_3(i), \text{ για } i = 1,2,3,4$ και μετά τα $f_2(i), \text{ για } i = 1,2,3,4$, ώσπου να βρούμε το $f_1(0)$, οπότε και θα είμαστε σε θέση να εκτιμήσουμε το βέλτιστο επίπεδο παραγωγής για κάθε μήνα.

Στάδιο 4: Κατά τη διάρκεια του 4^{ου} μήνα, η εταιρεία θα παράγει τόσες μονάδες, όσες ακριβώς χρειάζονται για να ικανοποιήσει τη ζήτηση των 4 μονάδων του μήνα αυτού. Έχουμε:

Κατάσταση	$c(x)$	Βέλτιστη λύση	
	$x+3$	$f_4(i)$	$x_4^*(i)$
0	4+3=7	7	4
1	3+3=6	6	3
2	2+3=5	5	2
3	1+3=4	4	1
4	0	0	0

Πίνακας 7

Στάδιο 3: Κάνοντας ένα βήμα πιο πίσω, θέλουμε να προσδιορίσουμε τα $f_3(i)$, για $i=1,2,3,4$. Το $f_3(i)$ είναι το ελάχιστο κόστος που προέκυψε τον 3$^{\text{ο}}$ και 4$^{\text{ο}}$ μήνα, αν το αποθεματικό επίπεδο στην αρχή του 3$^{\text{ου}}$ είναι i. Για κάθε δυνατό επίπεδο παραγωγής τον 3$^{\text{ο}}$ μήνα, το συνολικό κόστος κατά τη διάρκεια του 3$^{\text{ου}}$ και 4$^{\text{ου}}$ μήνα είναι:

$$\frac{1}{2}(i+x-2)+c(x)+\mathrm{f}_4(i+x-2) \quad \textbf{(13)},$$

διότι, αν κατά τον 3$^{\text{ο}}$ μήνα παραχθούν 3 μονάδες, τότε τα αποθέματα στο τέλος του μήνα αυτού θα είναι $i+x-2$, το κόστος διατήρησης θα είναι $\frac{1}{2}(i+x-2)$ και το κόστος παραγωγής $c(x)$. Στην αρχή του 4$^{\text{ου}}$ μήνα θα διαθέτουμε $i+x-2$ μονάδες. Άρα, βάσει της Αρχής της βελτιστοποίησης που χρησιμοποιούμε, το κόστος για τον 4^o μήνα θα είναι $f_4(i+x-2)$. Εμείς θέλουμε το επίπεδο παραγωγής τον 3^o μήνα να είναι τέτοιο που να ελαχιστοποιεί τη σχέση (13), άρα

$$f_3(i) = \min_x \left\{ \frac{1}{2}(i+x-2) + c(x) + \mathrm{f}_4(i+x-2) \right\} \quad \textbf{(14)},$$

όπου: το x πρέπει να είναι ένα εκ των $\{0,1,2,3,4,5\}$ και να ισχύει $0 \le i+x-2 \le 4$, που σημαίνει ότι η ζήτηση του τρέχοντος μήνα είναι μη-αρνητική ($i+x-2 \ge 0$) και η χωρητικότητα της αποθήκης να μην ξεπερνάει τις 4 μονάδες ($i+x-2 \le 4$).

Έτσι, έχουμε τον παρακάτω πίνακα:

	$\frac{1}{2}(i+x-2)+c(x)+\mathrm{f}_4(i+x-2)$						Βέλτιστη λύση	
Κατάσταση	$x=0$	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$	$f_3(\mathrm{x})$	$x_3^*(i)$
0	-	-	12	12,5	13	13,5	12	2
1	-	11	11,5	12	12,5	10	10	5
2	7	10,5	11	11,5	9	-	7	0
3	6,5	10	10,5	8	-	-	6,5	0
4	6	9,5	7	-	-	-	6	0

Πίνακας 8

Στάδιο 2: Υπολογίζουμε τώρα το ελάχιστο κόστος $f_2(i)$ του 2ου, 3ου και 4ου μήνα, δεδομένου ότι στην αρχή του 2ου μήνα υπάρχουν i μονάδες στην αποθήκη. Έστω x η παραγωγή του 2ου μήνα. Εφόσον η ζήτηση του 2ου μήνα είναι 3 μονάδες, το κόστος διατήρησης στο τέλος αυτού του μήνα είναι $\frac{1}{2}(i+x-3)$, ενώ το συνολικό του κόστος

	$\frac{1}{2}(i+x-3)+c(x)+\mathrm{f}_3(i+x-3)$						Βέλτιστη λύση	
Κατάσταση	$x=0$	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$	$f_2(\mathrm{x})$	$x_2^*(i)$
0	-	-	-	18	17,5	16	16	5
1	-	-	17	16,5	15	16	15	4
2	-	16	15,5	14	15	16	14	3
3	12	14,5	13	14	15	-	12	0
4	10,5	12	13	14	-	-	10,5	0

Πίνακας 9

$\frac{1}{2}(i+x-3)+c(x)$. Τον 3ο και 4ο μήνα ακολουθούμε βέλτιστη πολιτική. Στην αρχή του 3ου μήνα, το αποθεματικό είναι $i+x-3$ και προκύπτει ένα κόστος $f_3(i+x-3)$. Τότε, έχουμε

$$f_2(i) = \min_x \left\{ \frac{1}{2}(i+x-3) + c(x) + \mathrm{f}_3(i+x-3) \right\} \quad \textbf{(15)},$$

όπου: το x είναι ένα εκ των $\{0,1,2,3,4,5\}$ και ισχύει $0 \le i+x-3 \le 4$.

Ο πίνακας 9 δίνει τις τιμές των $f_2(i)$, για $i = 1,2,3,4$.

Κατάσταση	$\frac{1}{2}(i+x-1)+c(x)+\mathrm{f}_2(i+x-1)$						Βέλτιστη λύση	
	$x=0$	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$	$f_1(\mathrm{x})$	$x_1^*(i)$
0	-	20	20,5	21	20,5	20,5	20	1
1	16	19,5	20	19,5	19,5	19,5	16	0
2	16,5	19	18,5	18,5	-	-	16,5	0
3	15	17,5	17,5	-	-	-	15	0
4	13,5	16,5	-	-	-	-	13,5	0

Πίνακας 10

Στάδιο 1: Η $f_1(i)$ υπολογίζεται μέσω της αναδρομικής σχέσης:

$$f_2(i) = \min_x \left\{ \frac{1}{2}(i+x-1) + c(x) + \mathrm{f}_2(i+x-1) \right\} \quad \textbf{(16)},$$

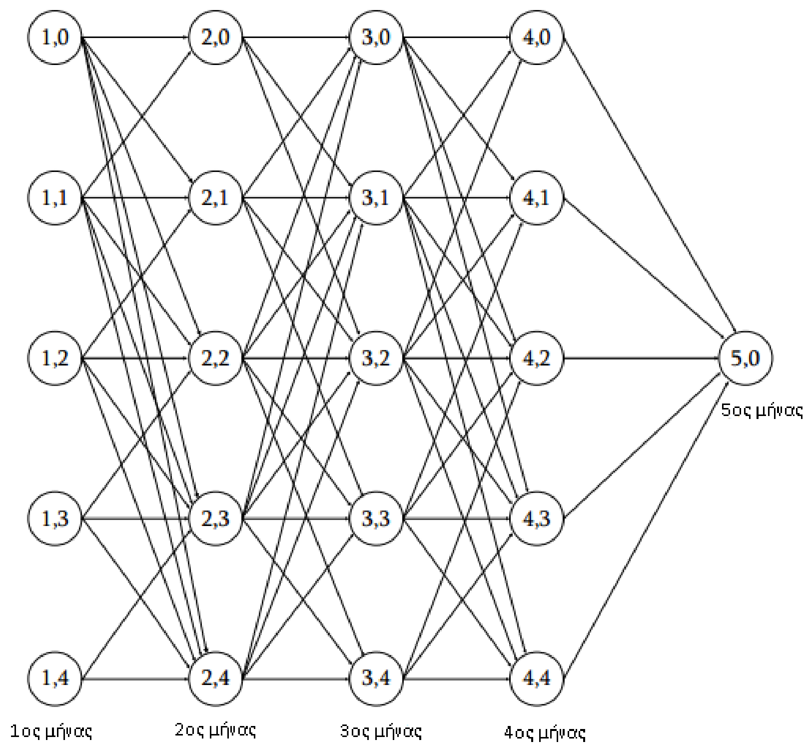
όπου: το x είναι ένα εκ των $\{0,1,2,3,4,5\}$ και ισχύει $0 \le i+x-1 \le 4$. Εφόσον στην απογραφή του 1^{ου} μήνα υπάρχουν 0 μονάδες, λαμβάνουμε τον πίνακα 10.

Εφόσον αναλύσαμε και το τελευταίο στάδιο, είμαστε πλέον σε θέση να κατασκευάσουμε το πλάνο παραγωγής που ελαχιστοποιεί το συνολικό κόστος και ικανοποιεί τη ζήτηση και για τους τέσσερις μήνες. Όπως είπαμε και στο στάδιο 1, η αρχική απογραφή είναι 0 μονάδες, άρα το ελάχιστο κόστος για τους τέσσερις μήνες θα είναι 20 χρηματικές μονάδες (βλ. πίνακα 10). Για να πετύχουμε $f_1(0)$, πρέπει να παράγουμε $x_1(0)=1$ μονάδα τον 1^{ο} μήνα. Το απόθεμα στην αρχή του 2^{ου} μήνα, τότε, θα είναι 0 μονάδες και άρα, το 2^{ο} μήνα πρέπει να παράγουμε $x_2(0)=5$ μονάδες. Στην αρχή του 3^{ου} μήνα η απογραφή θα είναι 2 μονάδες, άρα, το μήνα αυτό πρέπει να παράγουμε $x_3(0)=5$ μονάδες. Τον 4^{ο} μήνα ξεκινάμε με 0 μονάδες διαθέσιμες, άρα, πρέπει να παράγουμε $x_4(0)=4$ μονάδες.

Τελικά, το βέλτιστο σχεδιάγραμμα παραγωγής περιλαμβάνει ένα συνολικό κόστος 20 χρηματικών μονάδων και παράγει 1 μονάδα τον 1^{ο} μήνα, 5 μονάδες το 2 μήνα, 0 μονάδες τον 3^{ο} μήνα και 4 μονάδες τον 4^{ο} μήνα.

3.2.3 Αναπαράσταση του προβλήματος σε δίκτυο

Θα παρουσιάσουμε τώρα μια ισοδύναμη με αυτή του παραδείγματος λύση για το πρόβλημα της απογραφής αποθεμάτων και θέλουμε να βρούμε την ελάχιστη απόσταση που συνδέει τους κόμβους (1,0) και (5,0). Κάθε κόμβος αντιπροσωπεύει ένα στάδιο και κάθε στήλη με κόμβους όλες τις πιθανές καταστάσεις ενός συγκεκριμένου σταδίου. Αν βρισκόμαστε π.χ. στον κόμβο (3,2), τότε είμαστε στην αρχή του 3^{ου} μήνα, όπου η αρχική απογραφή είναι 2 μονάδες.



Εικόνα 4: Αναπαράσταση του παραδείγματος 2 σε δίκτυο

Κάθε τόξο παριστάνει τον τρόπο με τον οποίο μια απόφαση μετατρέπει την κατάσταση του τρέχοντος σταδίου στην κατάσταση του επόμενου σταδίου, π.χ. το τόξο που συνδέει τους κόμβους (3,1) και (4,3) δείχνει ότι παράγονται 4 μονάδες τον 3^{ο} μήνα, διότι ο 4^{ος} μήνας στο ξεκίνημά του έχει 0 μονάδες διαθέσιμες και αν έχουν παραχθεί 4 μονάδες τον 3^{ο} μήνα, τότε (από τη σχέση που γράψαμε νωρίτερα στο στάδιο 3) ο 4^{ος} μήνας ξεκινάει με 1+4-2=3 μονάδες. Το μήκος του τόξου είναι το άθροισμα του κόστους παραγωγής και του κόστους αποθήκευσης δεδομένης της τρέχουσας κατάστασης και της απόφασης που αφορά το συγκεκριμένο τόξο.

Σημείωση: Δε συνδέονται απαραίτητα όλοι οι κόμβοι γειτονικών σταδίων μεταξύ τους, π.χ. ο κόμβος (2,4) δε συνδέεται με τον κόμβο (3,0), αφού ξεκινώντας το 2^{ο} μήνα με 4 μονάδες σημαίνει ότι στην αρχή του 3^{ου} μήνα θα χρειαζόμασταν τουλάχιστον 4-3=1 μονάδα. Επίσης, όλοι οι κόμβοι του 4^{ου} μήνα συνδέονται με τον κόμβο (5,0), διότι δεν επιθυμούμε να έχουμε στο τέλος του 4^{ου} μήνα θετικό απόθεμα (δεν αποτελεί βέλτιστη λύση!).

Αν ξαναγυρίσουμε στο παράδειγμά μας, βλέπουμε ότι ελάχιστη διαδρομή που συνδέει τους κόμβους (1,0) και (5,0) αντιστοιχεί στο ελάχιστο κόστος παραγωγής. Η ροή της βέλτιστης διαδρομής, λοιπόν, είναι $(1,0) \to (2,0) \to (3,2) \to (4,0) \to (5,0)$, όπου ο κόμβος (2,0)=(2,0+1-1), ο (3,2)=(3,0+5-3), ο (4,0)=(4,0+2-2) και ο (5,0)=(5,0+4-4).

3.3 Το Προβλημα της Συντομοτερης Διαδρομης σε ενα Δικτυο (Shortest Path problem in a network)

Στο πρόβλημα αυτό αναζητούμε τη συντομότερη διαδρομή σε ένα συνεκτικό δίκτυο, μεταξύ ενός σημείου εκκίνησης (αρχική κατάσταση) και ενός τελικού σημείου (τελική κατάσταση), με τις αποστάσεις μεταξύ των σημείων του δικτύου να είναι γνωστές.

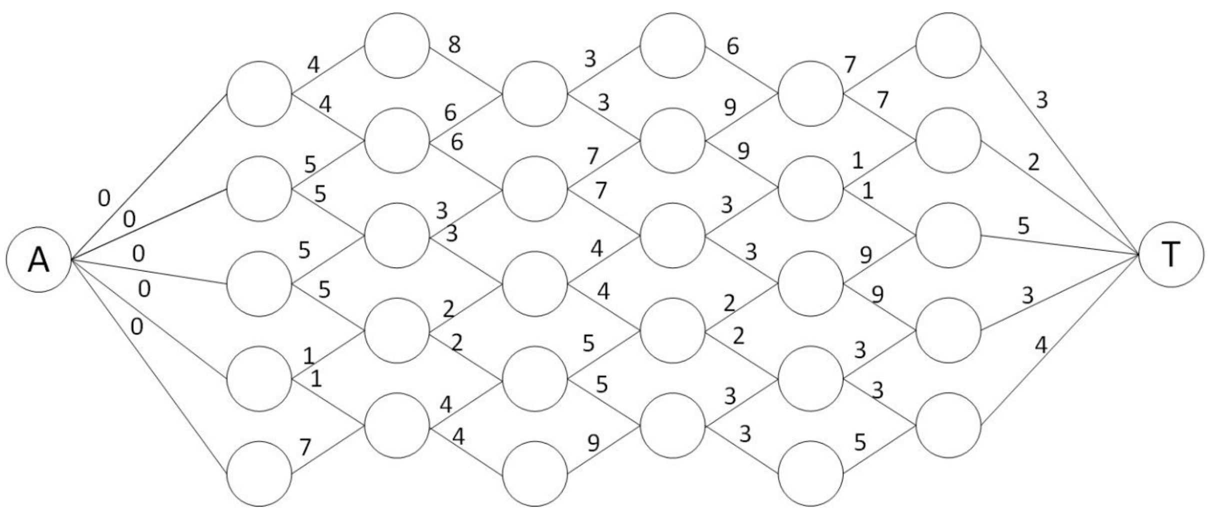
Η εικόνα 1 του πρώτου παραδείγματος (network problem) που μετατρέπεται τώρα στην εικόνα 5 με το οδικό δίκτυο με τις διασταυρώσεις και τις καθυστερήσεις τους (όπου Α: αφετηρία και Τ: τερματισμός), ουσιαστικά παρουσιάζει ένα δίκτυο ελάχιστης διαδρομής για ένα πρόβλημα ελάχιστης καθυστέρησης. Εκεί, ως σημείο τερματισμού λαμβάνουμε το σύνολο των θέσεων στάθμευσης στο κέντρο της πόλης, που όπως προείπαμε είναι αδιάφορο για τους κατοίκους σε ποια από αυτές θα παρκάρει ο καθένας τους. Ως αφετηρία λαμβάνουμε το σύνολο των σπιτιών τους. Έτσι, χρησιμοποιώντας τη μέθοδο του δυναμικού προγραμματισμού, διακρίνουμε δύο περιπτώσεις:

1) αν έχουμε *προς-τα-πίσω επαγωγή* (backward induction), βρίσκουμε τη συντομότερη διαδρομή από το τέλος προς την αρχή, ή καλύτερα τη συντομότερη

διαδρομή από την τελική διασταύρωση προς όλες τις διασταυρώσεις του δικτύου, δηλαδή προς όλες τις κατοικίες και

2) αν έχουμε *προς-τα εμπρός επαγωγή* (forward induction), βρίσκουμε αντίθετα το συντομότερο μονοπάτι από την αρχή προς το τέλος, δηλαδή το συντομότερο μονοπάτι για το σπίτι ενός και μόνο κατοίκου.

Παρόλο που, όπως αποδείξαμε και στο παράδειγμα του δικτύου, η συντομότερη διαδρομή είναι η ίδια και για τις δύο περιπτώσεις, η προς-τα-εμπρός επαγωγή τελικά δε λύνει το πρόβλημα ελάχιστης καθυστέρησης για όλους τους κατοίκους, διότι εδώ οι κάτοικοι σαφώς ενδιαφέρονται για το σπίτι στο οποίο θα φτάσουν.



Εικόνα 5: Το δίκτυο συντομότερης διαδρομής για το πρόβλημα ελάχιστης καθυστέρησης

Τονίζουμε ότι ***η μέθοδος του δυναμικού προγραμματισμού και η μέθοδος της ελάχιστης διαδρομής είναι παρόμοιες***. Θα ακολουθήσει παράδειγμα στη συνέχεια που δείχνει ακριβώς αυτή τη μεταξύ τους συσχέτιση. Για την ώρα ας κρατήσουμε αυτό το γενικό κανόνα:

" Όσο καλύτερη είναι η δομή του δικτύου, τόσο πιο αποδοτικός είναι ο αλγόριθμος που θα αναπτύξουμε." [17][20]

Ας εξηγήσουμε λίγο περισσότερο τον κανόνα αυτό μέσα από δύο ορισμούς για τα δίκτυα. Έχουμε, λοιπόν, δύο κατηγορίες δικτύων:

α) τα ***μη-κυκλικά (απεριοδικά) δίκτυα***: είναι τα δίκτυα ή οι γράφοι που δεν περιέχουν προσανατολισμένους βρόχους, δηλαδή είναι αδύνατο ξεκινώντας από οποιονδήποτε κόμβο και ακολουθώντας τα προσανατολισμένα βέλη τους να επιστρέψουμε στον ίδιο αρχικό κόμβο.

Η μέθοδος έχει ως εξής: διατάσσουμε τους κόμβους με τέτοιο τρόπο ώστε, αν το δίκτυο περιέχει το τόξο $i-j$, τότε $i>j$. Ξεκινάμε από τον τελικό κόμβο (ο οποίος θεωρούμε ότι περιλαμβάνει μόνο τόξα που καταλήγουν σε αυτόν και όχι τόξα που φεύγουν από αυτόν) ονομάζοντάς τον “μηδενικό κόμβο” και στη συνέχεια αγνοούμε τον κόμβο αυτό και τα τόξα που έχουν κατεύθυνση προς αυτόν και ονοματίζουμε κάθε κόμβο στον οποίο μόνο καταλήγουν τόξα. Αυτό το πλεονέκτημα ύπαρξης τέτοιων κόμβων μας το παρέχει το γεγονός ότι το δίκτυο είναι ακυκλικό. Η διαδικασία συνεχίζεται μέχρις ότου ονοματιστούν όλοι οι κόμβοι.

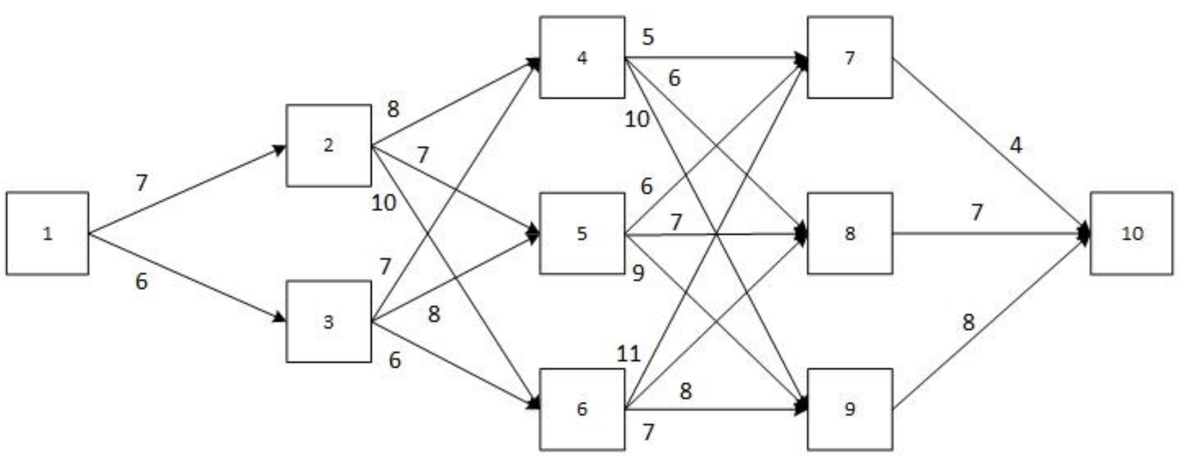
Άρα, με αυτόν ακριβώς τον τρόπο έγινε η μετάβαση από το δίκτυο της εικόνας 5 στην εικόνα 1 στο πρόβλημα συντομότερης διαδρομής της ελάχιστης καθυστέρησης στο δίκτυο. Γενικά, για ένα πρόβλημα συντομότερης διαδρομής, αν δε θέλουμε να βρούμε τα συντομότερα μονοπάτια από οποιονδήποτε κόμβο στον τελικό κόμβο, αλλά μονάχα το μικρότερο μονοπάτι στον αρχικό κόμβο, τότε μπορούμε να τερματίσουμε τη διαδικασία αμέσως μόλις ονοματιστεί ο αρχικός κόμβος. Έτσι, ξεκινώντας από τον αρχικό κόμβο, μπορούμε να βρούμε τις συντομότερες διαδρομές από τον κόμβο αυτό προς κάθε άλλο κόμβο, ονοματίζοντας τους κόμβους, εφαρμόζοντας απολύτως αντίστροφα τη διαδικασία.

β) τα ***δίκτυα χωρίς αρνητικούς βρόχους***: είναι τα δίκτυα που μπορεί να περιέχουν βρόχους με την απόσταση γύρω από οποιονδήποτε βρόχο (π.χ. το άθροισμα των μηκών των τόξων του) να είναι απαραιτήτως μη αρνητική.

Αν σε ένα γενικό πρόβλημα θέλουμε να βρούμε την ελάχιστη διαδρομή μεταξύ δύο κόμβων, έστω 1 και N, τότε αναπτύσσουμε έναν αλγόριθμο δυναμικού προγραμματισμού βασισμένο στην αρχή της βελτιστοποίησης που λειτουργεί ως εξής: κάθε μονοπάτι από τον κόμβο 1 σε έναν κόμβο j φτάνει στον κόμβο j από τον κόμβο i μέσω του τόξου $i-j$, περνώντας το πολύ από n ενδιάμεσους κόμβους κι αυτό συμβαίνει αφού προηγουμένως εντοπίσει το μικρότερο μονοπάτι, περνώντας το πολύ από $n-1$ κόμβους από τον κόμβο j στον κόμβο i.

3.3.1 *Εύρεση συντομότερης διαδρομής (παράδειγμα κλειστού γράφου)*

Έστω το οδικό δίκτυο της εικόνας 6 και ένας ταξιδιώτης που επιθυμεί να ταξιδέψει από την πόλη 1 στην πόλη 10. Αναγκαστικά θα περάσει από κάποιες ενδιάμεσες πόλεις, σκοπός του, όμως, είναι να διανύσει συνολικά τη συντομότερη δυνατή διαδρομή.



Εικόνα 6: Οδικό δίκτυο παραδείγματος

Τα αριθμημένα τετράγωνα της εικόνας παριστάνουν τις 10 διαφορετικές πόλεις, ενώ ο αριθμός πάνω σε κάθε βέλος είναι η απόσταση μεταξύ των δύο διαδοχικών πόλεων που συνδέονται με αυτό. Προφανώς, αν σκεφτούμε απλοϊκά, υπάρχουν πολλές διαδρομές μεταξύ των οποίων μπορεί να επιλέξει ο ταξιδιώτης να κινηθεί, ώστε να επιτύχει την ελάχιστη απόσταση συνολικά. Αυτή, όπως έχουμε πει και σε προηγούμενη παράγραφο, είναι η μέθοδος της συνολικής απαρίθμησης την οποία γενικά πρέπει να αποφεύγουμε, καθώς απαιτεί πολύ χρόνο και κόστος υπολογισμών (π.χ. εδώ θα χρειαστεί να εξεταστούν άμεσα $1\times2\times3\times3=18$ εναλλακτικές διαδρομές!).

Μία τακτική που επίσης θα μας βοηθήσει μελλοντικά είναι να “απαλλαγούμε” από την ιδέα ότι σε κάθε στάδιο πρέπει να επιλέγουμε την πιο κοντινή πόλη του επόμενου σταδίου, καθώς αυτό δε μας εξασφαλίζει πάντοτε τη συντομότερη διαδρομή συνολικά. Αν, λοιπόν, ο ταξιδιώτης κινούνταν με βάση αυτό το λανθασμένο σκεπτικό, θα επέλεγε τη διαδρομή $1\to3\to6\to9\to10$ με συνολική απόσταση 27 (=6+6+7+8). Ωστόσο, ακόμα και μία αλλαγή σε ένα στάδιο μπορεί να μας ωφελήσει το ίδιο ή και περισσότερο σε επόμενα στάδια [10].

Για τους λόγους αυτούς, θα χρησιμοποιήσουμε την “έξυπνη” μέθοδο που μας παρέχει ο δυναμικός προγραμματισμός, τη *μέθοδο της μερικής απαρίθμησης*. Χωρίζουμε, λοιπόν, το πρόβλημα σε 4 υποπροβλήματα, τα οποία εδώ ονομάζουμε ***στάδια*** και στο σχήμα τα έχουμε συμβολίσει με $n=1,2,3,4$. Σε καθένα από αυτά, ο ταξιδιώτης καλείται να λάβει μία ***απόφαση*** που αφορά την επόμενη πόλη στην οποία θα βρεθεί. Τέλος, κάθε στάδιο περιλαμβάνει έναν συγκεκριμένο αριθμό ***καταστάσεων,*** στην περίπτωσή μας πόλεων που μπορεί να βρεθεί ο ταξιδιώτης. Δηλαδή, στο στάδιο $n=1$ αντιστοιχεί μόνο η πόλη 1, στο στάδιο $n=2$ αντιστοιχούν οι πόλεις 2 και 3, στο στάδιο $n=3$ αντιστοιχούν οι πόλεις 4, 5 και 6 και στο στάδιο $n=4$ οι πόλεις 7, 8 και 9.

Γενικά, όταν λέμε ότι ο ταξιδιώτης βρίσκεται σε μια κατάσταση i, θα εννοούμε ότι βρίσκεται στην πόλη i.

Προσοχή, δεν υπάρχει στάδιο $n=5$ στο οποίο να αντιστοιχεί η πόλη 10, διότι στο στάδιο αυτό δεν έχει να πάρει καμία επιπλέον απόφαση ο ταξιδιώτης.

Θα προχωρήσουμε στη διαδικασία επίλυσης του προβλήματος συμβολίζοντας με

n : το στάδιο του προβλήματος (για $n=1,2,3,4$),

s_n: τις καταστάσεις των σταδίων n (δηλαδή την πόλη που βρίσκεται ο ταξιδιώτης σε κάθε στάδιο n),

x_n: τις μεταβλητές απόφασης του σταδίου n (δηλαδή το στάδιο $n+1$ στο οποίο θα μετακινηθεί ο ταξιδιώτης αμέσως μετά το στάδιο n),

$d_{s_n x_n}$: η απόσταση ανάμεσα στην κατάσταση s_n και τη μεταβλητή απόφασης x_n,

$f_n(s_n, x_n)$: η συνολικά ελάχιστη απόσταση των υπόλοιπων σταδίων (δηλαδή όταν ο ταξιδιώτης αποφασίζει να κινηθεί στην πόλη x_n (στο στάδιο n) ευρισκόμενος στην κατάσταση s_n),

x_n^*: την τιμή της x_n που ελαχιστοποιεί την $f_n(s_n, x_n)$ και

$f_n^*(s_n)$: την ελάχιστη τιμή της $f_n(s_n, x_n)$ (δηλαδή $f_n^*(s_n) = f_n(s_n, x_n^*)$).

Η επίλυση του προβλήματος θα γίνει με τη μέθοδο της προς-τα-πίσω επαγωγής, άρα θα βρούμε αρχικά την $f_4^*(s_4)$, μετά τις $f_3^*(s_3)$, $f_2^*(s_2)$ και τέλος την $f_1^*(s_1)$ που θα είναι και η λύση του προβλήματός μας.

Ξεκινάμε να εφαρμόζουμε τη μέθοδο που προαναφέραμε, με τον ταξιδιώτη να βρίσκεται στο στάδιο $n=4$, άρα σε μία από τις τρεις καταστάσεις (πόλεις) 7, 8, 9 και να πρέπει να πάρει μία απόφαση για να φτάσει στον επιθυμητό προορισμό του που είναι η κατάσταση (πόλη) 10. Στο tableau 1 παρουσιάζεται η λύση του προβλήματος για το στάδιο αυτό.

Λύση προβλήματος ενός σταδίου ($n=4$)

s_4 \ x_4	$f_4(s_4,x_4)=\mathrm{d}_{s_4x_4}$	$f_4^*(s_4)$	x_4^*
	10		
7	4	4	10
8	7	7	10
9	8	8	10

Tableau 1

Στο στάδιο $n=4$, λοιπόν, ο ταξιδιώτης μπορεί να βρεθεί είτε στην κατάσταση $s_4=7$, είτε στην $s_4=8$, είτε στην $s_4=9$, έχοντας ως μοναδική δυνατή επιλογή τη μετάβαση στην πόλη 10. Άρα, αν π.χ. βρεθεί στην κατάσταση $s_4=7$, θα είναι $f_4(s_4,x_4)=\mathrm{d}_{s_4x_4}=4$. Αξίζει να σημειώσουμε ότι μόνο στο στάδιο $n=4$(που είναι το τελευταίο στάδιο) συμπίπτουν τα $f_n(s_n,x_n)$ και $d_{s_nx_n}$, με το s_n να είναι 7, 8 ή 9. Βλέπουμε, λοιπόν, ότι για $f_4^*(s_4)=4$ πετυχαίνουμε την ελαχιστοποίηση της $f_4(s_4,x_4)$, άρα η βέλτιστη απόφαση είναι η $x_4^*=10$, δηλαδή ο ταξιδιώτης μεταβαίνει από την πόλη 7 στην πόλη 10 με ελάχιστη απόσταση ίση με 4. Αντίστοιχα αποτελέσματα παίρνουμε για $s_4=8$ ή $s_4=9$. Αν βρεθεί στην κατάσταση $s_4=8$, είναι $f_4^*(s_4)=7$ με βέλτιστη απόφαση τη $x_4^*=10$, δηλαδή μεταβαίνει από την πόλη 8 στην πόλη 10 με ελάχιστη απόσταση 7. Τέλος, αν βρεθεί στην κατάσταση $s_4=9$, είναι $f_4^*(s_4)=8$ με βέλτιστη απόφαση τη $x_4^*=10$, δηλαδή μεταβαίνει από την πόλη 9 στην πόλη 10 με ελάχιστη απόσταση ίση με 8.

Προχωράμε ένα στάδιο πιο πίσω, όπου απομένουν στον ταξιδιώτη δύο στάδια ($n=3$) μέχρι να φτάσει στον τελικό του προορισμό. Εδώ έχουμε πάλι τρεις πιθανές καταστάσεις, την $s_3=4$, την $s_3=5$ ή την $s_3=6$. Η λύση του προβλήματος στο στάδιο αυτό φαίνεται στο tableau 2. Αν ο ταξιδιώτης βρίσκεται στην κατάσταση $s_3=4$, μπορεί να αποφασίσει να μεταβεί είτε στην πόλη 7 (άρα $x_3=7$ με απόσταση $d_{47}=5$), είτε στην πόλη 8 (άρα $x_3=8$ με απόσταση $d_{48}=6$), είτε στην πόλη 9 (άρα $x_3=9$ με απόσταση $d_{49}=10$). Έστω ότι αποφασίζει να μεταβεί στην πόλη 7. Τότε, εκτός από την απόσταση $d_{47}=5$ που υπάρχει μεταξύ των δύο πόλεων, θα πρέπει να προσθέσουμε και την ελάχιστη απόσταση της πόλης 7 μέχρι τον προορισμό του που είναι $f_4^*(7)=4$, άρα συνολικά θα διανύσει απόσταση $d_{47}+f_4^*(7)=5+4=9$. Ομοίως, αν αποφασίσει να μεταβεί στην πόλη 8, θα διανύσει συνολική απόσταση $d_{48}+f_4^*(8)=6+7=13$ και αν μεταβεί στην πόλη 9 θα διανύσει συνολική απόσταση $d_{49}+f_4^*(9)=10+8=18$ μέχρι τον τελικό του προορισμό. Επομένως, η βέλτιστη απόφαση που μπορεί να πάρει ο ταξιδιώτης όντας στην πόλη 4 είναι η μετάβασή του στην πόλη 7, αφού έτσι επιτυγχάνει την ελάχιστη απόσταση.

Με τον ίδιο τρόπο βρίσκουμε και τις υπόλοιπες συνολικές αποστάσεις, αν ο ταξιδιώτης βρίσκεται στην κατάσταση $s_3=5$ ή στην κατάσταση $s_3=6$. Ουσιαστικά, ο τύπος που μας δίνει τις αποστάσεις αυτές είναι ο

$$f_3(s_3,x_3)=\mathrm{d}_{s_3x_3}+f_4^*(\mathrm{x}_3)$$

και βάσει αυτού συμπληρώνουμε τα αντίστοιχα κελιά του δεύτερου tableau. Έχουμε:

$$f_3(4,7)=\mathrm{d}_{47}+f_4^*(7)=5+4=9$$

$$f_3(4,8)=\mathrm{d}_{48}+f_4^*(8)=6+7=13$$

$$f_3(4,9)=\mathrm{d}_{49}+f_4^*(9)=10+8=18$$

$$f_3(5,7)=\mathrm{d}_{57}+f_4^*(7)=6+4=10$$

$$f_3(5,8) = \mathrm{d}_{58} + f_4^*(8) = 7 + 7 = 14$$

$$f_3(5,9) = \mathrm{d}_{59} + f_4^*(9) = 9 + 8 = 17$$

$$f_3(6,7) = \mathrm{d}_{67} + f_4^*(7) = 11 + 4 = 15$$

$$f_3(6,8) = \mathrm{d}_{68} + f_4^*(8) = 8 + 7 = 15$$

$$f_3(6,9) = \mathrm{d}_{69} + f_4^*(9) = 7 + 8 = 15$$

Λύση προβλήματος δύο σταδίων ($n = 3$)

s_3 \ x_3	$f_3(s_3, x_3) = \mathrm{d}_{s_3 x_3} + f_4^*(\mathrm{x}_3)$			$f_3^*(s_3)$	x_3^*
	7	8	9		
4	9	13	18	9	7
5	10	14	17	10	7
6	15	15	15	15	7 ή 8 ή 9

Tableau 2

Μπορούμε, επομένως, να πούμε ότι αν βρεθεί στην πόλη 4 θα μεταβεί στην πόλη 10 που είναι και ο τερματικός σταθμός με ελάχιστη συνολική απόσταση 9, έχοντας επιλέξει προηγουμένως να περάσει από την πόλη 7. Αν βρεθεί στην πόλη 5, επιλέγει και πάλι την πόλη 7 με ελάχιστη συνολική απόσταση 10, ενώ αν βρεθεί στην πόλη 6 επιλέγει ή την πόλη 7, ή την πόλη 8, ή την πόλη 9 με ελάχιστη συνολική απόσταση 15.

Προχωράμε ένα στάδιο πιο πίσω, όπου απομένουν ακόμα τρία στάδια ($n = 2$) για να φτάσει ο ταξιδιώτης στον τελικό του προορισμό. Έστω ότι βρίσκεται στην κατάσταση $s_2 = 2$, άρα, μπορεί να επιλέξει μεταξύ των πόλεων 4 (με απόσταση $d_{24} = 8$), 5 (με απόσταση $d_{25} = 7$) και 6 (με απόσταση $d_{26} = 10$). Αν τώρα επιλέξει να μεταβεί στην πόλη 4, εκτός από τη μεταξύ τους απόσταση d_{24}, θα πρέπει να συνυπολογίσουμε και την απόσταση της πόλης 4 από τον τελικό προορισμό (πόλη 10). Η απόσταση αυτή, όπως υπολογίσαμε στο tableau 2 είναι $f_3^*(4) = 9$, επομένως, η ελάχιστη συνολική απόσταση μεταξύ των πόλεων 2 και 10 με ενδιάμεσο σταθμό την πόλη 4 είναι $f_2(2,4) = \mathrm{d}_{24} + f_3^*(4) = 8 + 9 = 17$. Αν αντί της πόλης 4 επιλέξει την πόλη 5, η ελάχιστη συνολική απόσταση μεταξύ των πόλεων 2 και 10 θα είναι $f_2(2,5) = \mathrm{d}_{25} + f_3^*(5) = 7 + 10 = 17$, ενώ αν επιλέξει την πόλη 6 θα έχει ελάχιστη συνολική απόσταση $f_2(2,6) = \mathrm{d}_{26} + f_3^*(6) = 10 + 15 = 25$. Όπως και νωρίτερα, ο τύπος που μας δίνει τις αποστάσεις αυτές είναι ο

$$f_2(s_2, x_2) = \mathrm{d}_{s_2 x_2} + f_3^*(\mathrm{x}_2)$$

και βάσει αυτού συμπληρώνουμε τα αντίστοιχα κελιά του τρίτου tableau.

Έχουμε:

$$f_2(2,4) = \mathrm{d}_{24} + f_3^*(4) = 8 + 9 = 17$$

$$f_2(2,5) = \mathrm{d}_{25} + f_3^*(5) = 7 + 10 = 17$$

$$f_2(2,6) = \mathrm{d}_{26} + f_3^*(6) = 10 + 15 = 25$$

$$f_2(3,4) = \mathrm{d}_{34} + f_3^*(4) = 7 + 9 = 16$$

$$f_2(3,5) = \mathrm{d}_{35} + f_3^*(5) = 8 + 10 = 18$$

$$f_2(3,6) = \mathrm{d}_{36} + f_3^*(6) = 6 + 15 = 21$$

Λύση προβλήματος τριών σταδίων ($n = 2$)

x_2 / s_2	$f_2(s_2, x_2) = \mathrm{d}_{s_2 x_2} + f_3^*(\mathrm{x}_2)$			$f_2^*(s_2)$	x_2^*
	4	5	6		
2	17	17	25	17	4 ή 5
3	16	18	21	16	4

Tableau 3

Συμπεραίνουμε ότι, αν ο ταξιδιώτης βρεθεί στην πόλη 2 θα μεταβεί στην πόλη 10 που είναι και ο τερματικός σταθμός με ελάχιστη συνολική απόσταση 17, έχοντας επιλέξει προηγουμένως να περάσει είτε από την πόλη 4, είτε από την πόλη 5. Αν βρεθεί στην πόλη 3, επιλέγει την πόλη 4 με ελάχιστη συνολική απόσταση 16.

Προχωράμε ένα στάδιο ακόμα οπισθοδρομικά, όπου απομένουν τέσσερα στάδια ($n=1$) για να φτάσει ο ταξιδιώτης στον προορισμό του. Στην ουσία, βρίσκεται στην αφετηρία του, την πόλη 1 και έχει να λύσει ολόκληρο το πρόβλημα. Έχει, λοιπόν, να επιλέξει μεταξύ των πόλεων 2 και 3, με αντίστοιχες αποστάσεις από την πόλη 1 τις $d_{12}=7$ και $d_{13}=6$. Ο τύπος που μας δίνει την ελάχιστη συνολική απόσταση των δύο αυτών πόλεων μέχρι την τελική πόλη είναι ο

$$f_1(s_1,x_1)=\mathrm{d}_{s_1x_1}+f_2^*(\mathrm{x}_1)$$

και βάσει αυτού συμπληρώνουμε τα αντίστοιχα κελιά του τέταρτου tableau. Έχουμε:

$$f_1(1,2)=\mathrm{d}_{12}+f_2^*(2)=7+17=24$$
$$f_1(1,3)=\mathrm{d}_{13}+f_2^*(3)=6+16=22$$

Λύση προβλήματος τεσσάρων σταδίων ($n=1$)

s_1 \ x_1	$f_1(s_1,x_1)=\mathrm{d}_{s_1x_1}+f_2^*(\mathrm{x}_1)$		$f_1^*(s_1)$	x_1^*
	2	3		
1	24	22	22	3

Tableau 4

Αυτό που συμπεραίνουμε συνολικά για το πρόβλημα είναι ότι ο ταξιδιώτης, με αφετηρία την πόλη 1 στο πρόβλημα τεσσάρων σταδίων, επιλέγει αρχικά να μετακινηθεί στην πόλη 3, εφόσον $x_1^*=3$. Έπειτα, στο πρόβλημα τριών σταδίων μεταβαίνει στην πόλη 4, αφού $x_2^*=4$. Επόμενος σταθμός του είναι η πόλη 7 στο πρόβλημα τριών σταδίων, αφού $x_3^*=7$ και τέλος, όπως είναι αναμενόμενο $x_4^*=10$ και μεταβαίνει στον προορισμό του που είναι η πόλη 10. Επομένως, η συντομότερη διαδρομή που κάνει από την πόλη 1 έως την πόλη 10 είναι η $1 \to 3 \to 4 \to 7 \to 10$ με συνολική ελάχιστη απόσταση $f_1^*(1) = 22$.

3.4 ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΣΑΚΙΔΙΟΥ (THE KNAPSACK PROBLEM)

Το πρόβλημα του σακιδίου περιγράφει την κατάσταση ενός οδοιπόρου που πρέπει να αποφασίσει τον τρόπο με τον οποίο θα αποθηκεύσει διάφορα αντικείμενα στο σακίδιό του κατά το βέλτιστο, όμως, τρόπο[8]. Το πρόβλημα μπορεί να αντιμετωπισθεί και ως ένα γενικό πρόβλημα κατανομής πόρων, όπου ένας περιορισμένος αριθμός πόρων αντιστοιχεί σε ένα σύνολο εναλλακτικών λύσεων, με στόχο τη μεγιστοποίηση της συνολικής απόδοσης.

Έστω ότι το σακίδιο έχει χωρητικότητα W για τα n συνολικά αντικείμενα που θα τοποθετηθούν σε αυτό, με m_i να είναι ο αριθμός των μονάδων του i αντικειμένου, ενώ έχουν βάρος w_i και κέρδος p_i έκαστο. Για την περιγραφή του προβλήματος θα χρησιμοποιήσουμε πάλι την προς-τα-πίσω αναδρομή, οπότε η μοντελοποίηση θα είναι η εξής:

$$\max \ z = p_1 m_1 + p_2 m_2 + ... + p_n m_n$$

υπό τους περιορισμούς: $\begin{cases} w_1 m_1 + w_2 m_2 + ... + w_n m_n = \sum_{i=1}^{n} w_i m_i \le W \\ m_1, m_2, ..., m_n \ge 0 \text{ και ακέραιοι} \end{cases}$,

[8] Ένα πρόβλημα της ίδιας κατηγορίας είναι και το "***πρόβλημα του βέλτιστου φορτίου***", όπου σε ένα μέσο μεταφοράς περιορισμένης χωρητικότητας πρέπει να φορτωθούν αντικείμενα με το βέλτιστο τρόπο, δηλαδή με το φορτίο μέγιστης αξίας.

όπου: $w_i m_i$ είναι το βάρος του κάθε είδους, $\sum_{i=1}^{n} w_i m_i$ το συνολικό βάρος των αντικειμένων, $p_i m_i$ η αξία του κάθε είδους και $\sum_{i=1}^{n} p_i m_i$ η συνολική αξία των αντικειμένων.

Για να λύσουμε το πρόβλημα με τη μέθοδο της οπισθοδρομικής επαγωγής, θα ορίσουμε την

$$f_i(x_i) = \{\text{η μέγιστη απόδοση της κατάστασης } x_i \text{ για τα στάδια } i, i+1, \ldots, n\}$$

ως βέλτιστη συνάρτηση, όπου x_i : το συνολικό βάρος στο στάδιο $i, i+1, \ldots, n$.

Έστω ότι το διαθέσιμο βάρος φόρτωσης για τα αντικείμενα $i, i+1, \ldots, n$ είναι w ($w = 0,1,\ldots,W$) και αποφασίζουμε να τοποθετήσουμε m_i μονάδες του αντικειμένου i, με αναμενόμενη απόδοση $p_i m_i$. Αφού, λοιπόν, το διαθέσιμο βάρος είναι W και το βάρος κάθε μονάδας του αντικειμένου είναι w_i, συνεπάγεται ότι ο μεγαλύτερος ακέραιος αριθμός αντικειμένων που μπορούν να τοποθετηθούν στο σακίδιο είναι $\left[\frac{W}{w_i}\right]$, άρα το $m_i = 0,1,2,\ldots,\left[\frac{W}{w_i}\right]$.

Θα παρουσιάσουμε την αναδρομική σχέση ως μία διαδικασία με δύο στάδια.

Στάδιο 1: Η $f_i(x_i)$ συναρτήσει της $f_i(x_{i+1})$ γράφεται ως εξής:

$$f_i(x_i) = \max_{m_i = 0,1,\ldots,\left[\frac{W}{w_i}\right]} p_i m_i + f_{i+1}(x_{i+1}) \ , \ i = 1,2,\ldots,n$$

με $f_{n+1}(x_{n+1}) = 0$.

Στάδιο 2: Έχουμε ότι στο στάδιο i το βάρος του αντικειμένου είναι $x_i - x_{i+1} = w_i m_i \Rightarrow x_{i+1} = x_i - w_i m_i$. Άρα, αντικαθιστώντας το x_{i+1} στην προηγούμενη σχέση λαμβάνουμε:

$$f_i(x_i) = \max_{\substack{m_i = 0,1,\dots,\left[\frac{W}{w_i}\right] \\ x_i \le W}} p_i m_i + f_{i+1}(x_i - w_i m_i) \ , \ i = 1,2,\dots,n\,.$$

3.4.1 Εφαρμογή

Έστω μία αντιπροσωπία αγροτικών μηχανημάτων που θέλει να μεταφέρει τριών ειδών εμπορεύματα (1,2,3) από την κεντρική της αποθήκη στο κατάστημα. Το συνολικό βάρος που μπορεί να φορτωθεί το φορτηγό που θα τα μεταφέρει είναι $W = 4$ μονάδες βάρους, ενώ το βάρος w_i και η αξία p_i ανά είδος εμπορεύματος δίνονται στον ακόλουθο πίνακα:

Αντικείμενο i	Βάρος w_i	Κέρδος p_i
1	2	29
2	3	39
3	1	12

Πίνακας 11

Πόσα αντικείμενα θα πρέπει άραγε να φορτωθούν στο φορτηγό, ώστε να μεγιστοποιήσουμε τη συνολική απόδοση;

Λύση

Στάδιο 3: Εφόσον χρησιμοποιούμε την προς-τα πίσω αναδρομή, θα υπολογίσουμε πρώτα την κατάσταση x_3 (δηλαδή το βάρος του αντικειμένου 3), μετά την κατάσταση x_2 (βάρος του αντικειμένου 2) και τέλος την κατάσταση x_1 (βάρος του αντικειμένου 1). Το βάρος που θα διατεθεί για το στάδιο 3 (αντικείμενο 3), μπορεί να πάρει τις τιμές

0,1,…,4, αφού $W = 4$. Προφανώς, οι καταστάσεις $x_3 = 0$ και $x_3 = 4$ αποτελούν τις ακραίες συνθήκες, όπου δε φορτώνεται καθόλου κανένα αντικείμενο 3 ή το αντικείμενο 3 καταλαμβάνει όλη τη χωρητικότητα του φορτηγού, αντίστοιχα. Οι υπόλοιπες τιμές του x_3 αντιστοιχούν σε μια μικρή έως μέτρια καταγραφή χωρητικότητας του φορτηγού με το αντικείμενο αυτό.

Από τον πίνακα 7, έχουμε ότι $w_3 = 12$, άρα το μέγιστο φορτίο που μπορεί να υπάρξει με αντικείμενο 3 είναι $\left[\frac{4}{1}\right] = 4$, άρα οι δυνατές τιμές που μπορεί να πάρει το m_3 είναι 0,1,2,3,4, με εναλλακτικές λύσεις του m_3 μόνο αν ισχύει $w_3 m_3 \le x_3$. Η επόμενη εξίσωση θα δώσει τη δυνατότητα μιας σύγκρισης των εναλλακτικών (εφικτών) λύσεων για όλες τις τιμές του x_3 στο στάδιο αυτό:

$f_3(x_3) = \max_{m_3}\{12m_3\}$, όπου: $\max\{m_3\} = \left[\frac{4}{1}\right] = 4$

Είναι: $f_3(0) = \max_{m_3}\{0\}, \text{ για } m_3 = 0$

$$f_3(1) = \max_{m_3} \begin{cases} 0, \text{ για } m_3 = 0 \\ 12, \text{ για } m_3 = 1 \end{cases}$$

$$f_3(2) = \max_{m_3} \begin{cases} 0, \text{ για } m_3 = 0 \\ 12, \text{ για } m_3 = 1 \\ 24, \text{ για } m_3 = 2 \end{cases}$$

$$f_3(3) = \max_{m_3} \begin{cases} 0, \text{ για } \mathrm{m}_3 = 0 \\ 12, \text{ για } \mathrm{m}_3 = 1 \\ 24, \text{ για } \mathrm{m}_3 = 2 \\ 36, \text{ για } \mathrm{m}_3 = 3 \end{cases}$$

$$f_3(4) = \max_{m_3} \begin{cases} 0, \text{ για } \mathrm{m}_3 = 0 \\ 12, \text{ για } \mathrm{m}_3 = 1 \\ 24, \text{ για } \mathrm{m}_3 = 2 \\ 36, \text{ για } \mathrm{m}_3 = 3 \\ 48, \text{ για } \mathrm{m}_3 = 4 \end{cases}$$

Οι τιμές που μόλις βρήκαμε μπορούν να παρουσιαστούν πιο συγκεντρωμένα στον πίνακα 12.

x_3	$12\,m_3$					Βέλτιστη λύση	
	$m_3 = 0$	$m_3 = 1$	$m_3 = 2$	$m_3 = 3$	$m_3 = 4$	$f_3(x_3)$	m_3^*
0	0	-	-	-	-	0	0
1	0	12	-	-	-	12	1
2	0	12	24	-	-	24	2
3	0	12	24	36	-	36	3
4	0	12	24	36	48	48	4

Πίνακας 12

Στάδιο 2: Πηγαίνοντας ένα βήμα πιο πίσω για να βρούμε την κατάσταση x_2, έχουμε την εξής αναδρομική σχέση:

$$f_2(x_2) = \max_{m_2}\{39m_2 + f_3(x_2 - 3m_2)\}, \text{ όπου } \max\{m_2\} = \left[\frac{4}{3}\right] = 1$$

Έχουμε:

$$f_2(0) = \max_{m_2} \begin{cases} 0 + f_3(0) = 0 + 0 = 0 \text{ , για } m_2 = 0 \\ \qquad - \qquad\qquad \text{ , για } m_2 = 1 \end{cases}$$

$$f_2(1) = \max_{m_2} \begin{cases} 0 + f_3(1) = 0 + 12 = 12 \text{ , για } m_2 = 0 \\ \qquad - \qquad\qquad \text{ , για } m_2 = 1 \end{cases}$$

$$f_2(2) = \max_{m_2} \begin{cases} 0 + f_3(2) = 0 + 24 = 24 \text{ , για } m_2 = 0 \\ \qquad - \qquad\qquad \text{ , για } m_2 = 1 \end{cases}$$

$$f_2(3) = \max_{m_2} \begin{cases} 0 + f_3(3) = 0 + 36 = 36 \text{ , για } m_2 = 0 \\ 39 + f_3(0) = 39 + 0 = 39 \text{ , για } m_2 = 1 \end{cases}$$

$$f_2(4) = \max_{m_2} \begin{cases} 0 + f_3(4) = 0 + 48 = 48 \text{ , για } m_2 = 0 \\ 39 + f_3(1) = 39 + 12 = 51 \text{ , για } m_2 = 1 \end{cases}$$

Ο πίνακάς μας γι' αυτή την περίπτωση είναι ο ακόλουθος:

x_2	$39m_2 + f_3(x_2 - 3m_2)$		Βέλτιστη λύση	
	$m_2 = 0$	$m_2 = 1$	$f_2(x_2)$	m_2^*
0	0+0=0	-	0	0
1	0+12=12	-	12	0
2	0+24=24	-	24	0
3	0+36=36	39+0=39	39	1
4	0+48=48	39+12=51	51	1

Πίνακας 13

Στάδιο 1: Στο στάδιο αυτό βρίσκουμε την κατάσταση x_1 πηγαίνοντας ακόμα ένα βήμα πιο πίσω. Η αναδρομική σχέση έχει ως εξής:

$$f_1(x_1) = \max_{m_1}\{29m_1 + f_2(x_1 - 2m_1)\}, \text{ όπου } \max\{m_1\} = \left[\frac{4}{2}\right] = 2$$

Έχουμε:

$$f_1(0) = \max_{m_1} \begin{cases} 0 + f_2(0) = 0 + 0 = 0 \text{ , για } m_1 = 0 \\ \qquad - \qquad\qquad \text{ , για } m_1 = 1 \\ \qquad - \qquad\qquad \text{ , για } m_1 = 2 \end{cases}$$

$$f_1(1) = \max_{m_1} \begin{cases} 0 + f_2(1) = 0 + 12 = 12 \text{ , για } m_1 = 0 \\ \qquad - \qquad\qquad \text{ , για } m_1 = 1 \\ \qquad - \qquad\qquad \text{ , για } m_1 = 2 \end{cases}$$

$$f_1(2) = \max_{m_1} \begin{cases} 0 + f_2(2) = 0 + 24 = 24 \ , \text{ για } m_1 = 0 \\ 29 + f_2(0) = 29 + 0 = 29 \ , \text{ για } m_1 = 1 \\ - \ , \text{ για } m_1 = 2 \end{cases}$$

$$f_1(3) = \max_{m_1} \begin{cases} 0 + f_2(3) = 0 + 39 = 39 \ , \text{ για } m_1 = 0 \\ 29 + f_2(1) = 29 + 12 = 41 \ , \text{ για } m_1 = 1 \\ - \ , \text{ για } m_1 = 2 \end{cases}$$

$$f_1(4) = \max_{m_1} \begin{cases} 0 + f_2(4) = 0 + 51 = 51 \ , \text{ για } m_1 = 0 \\ 29 + f_2(2) = 29 + 24 = 53 \ , \text{ για } m_1 = 1 \\ 58 + f_2(0) = 58 + 0 = 58 \ , \text{ για } m_1 = 2 \end{cases}$$

Ο πίνακας τώρα διαμορφώνεται ως εξής:

x_1	$29m_1 + f_2(x_1 - 2m_1)$			Βέλτιστη λύση	
	$m_1 = 0$	$m_1 = 1$	$m_1 = 2$	$f_1(x_1)$	m_1^*
0	0+0=0	-	-	0	0
1	0+12=12	-	-	12	0
2	0+24=24	29+0=29	-	29	1
3	0+39=39	29+12=41	-	41	1
4	0+51=51	29+24=53	58+0=58	58	2

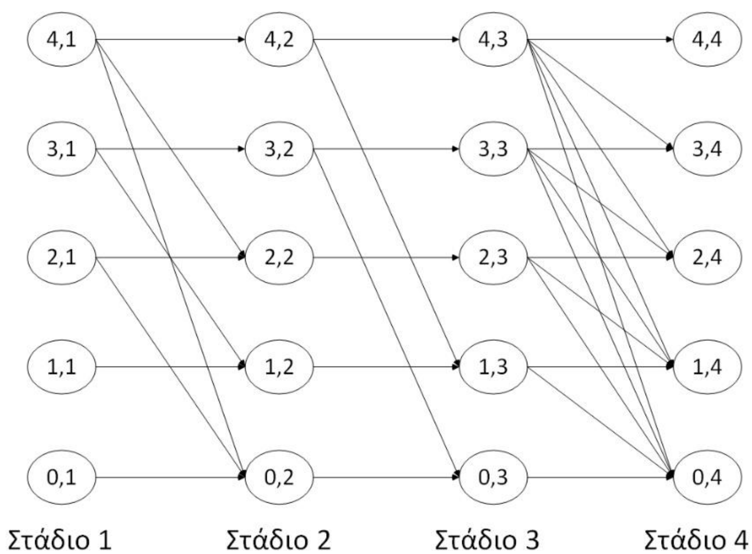
Πίνακας 14

Τώρα πλέον είναι εύκολο να φτάσουμε στη βέλτιστη λύση του προβλήματός μας. Ο πίνακας 10 του σταδίου 1 μας δείχνει ότι η κατάσταση $x_1 = 4$ δίνει την καλύτερη λύση για $m_1^* = 2$, που σημαίνει ότι από το αντικείμενο 1 θα φορτωθούν 2 μονάδες στο φορτηγό. Τότε, από τον τύπο $x_{i+1} = x_i - w_i m_i$, προκύπτει ότι $x_2 = x_1 - 2m_1 = 4 - 2 \cdot 2 = 0$. Στο στάδιο 2, το $x_2 = 0$ δίνει $m_2^* = 0$ κι έτσι προκύπτει $x_3 = x_2 - 3m_2^* = 0 - 3 \cdot 0 = 0$. Στο στάδιο 3, το $x_3 = 0$ δίνει $m_3^* = 0$.

Συνολικά, η βέλτιστη λύση μας δίνει $m_1^* = 2$, $m_2^* = 0$ και $m_3^* = 0$, επομένως, αναμένουμε να αποκτήσουμε ένα όφελος 58 χρηματικών μονάδων. Η μέγιστη αξία του φορτίου που μπορεί να μεταφερθεί με το φορτηγό από την αποθήκη έως το κατάστημα επιτυγχάνεται αν από το μηχάνημα 1 φορτώσουμε 2 κομμάτια και από τα μηχανήματα 2 και 3 φορτώσουμε 0 κομμάτια. Το φορτηγό, δηλαδή, μεταφέρει μόνο κομμάτια του μηχανήματος 1 κατά τη βέλτιστη λύση.

3.4.2 Αναπαράσταση του προβλήματος σακιδίου σε δίκτυο

Το να βρούμε τη λύση ενός προβλήματος, όπως του παραδείγματος 3, είναι ισοδύναμο με το να βρούμε τη μεγαλύτερη διαδρομή από τον κόμβο (4, 1) προς οποιονδήποτε κόμβο του σταδίου 4. Στην εικόνα 5 παρακάτω, κάθε κόμβος (d,t), για $t \leq 3$, παριστάνει μία κατάσταση, όπου d είναι οι μονάδες βάρους που μπορούν να διατεθούν για αντικείμενα του τύπου $t, t+1, \ldots, 3$. Ο κόμβος $(d,4)$ παριστάνει αχρησιμοποίητο χώρο d μονάδων βάρους. Κάθε τόξο από κάποιο κόμβο του σταδίου t σε κάποιο κόμβο του σταδίου $t+1$ αντιπροσωπεύει μία απόφαση σχετικά με τον αριθμό των αντικειμένων του τύπου t που τοποθετούνται μέσα στο σακίδιο.



Εικόνα 7: Αναπαράσταση παραδείγματος σακιδίου σε δίκτυο

Παραδείγματος χάριν, το τόξο που συνδέει τους κόμβους $(3,1)$ και $(1,2)$ παριστάνει την τοποθέτηση αντικειμένου του τύπου 1 στο σακίδιο και αφήνει χώρο να τοποθετηθούν 3-1=2 μονάδες βάρους των αντικειμένων 2 και 3. Το μήκος αυτού του τόξου είναι 41, όσο δηλαδή είναι και το όφελός μας από την τοποθέτηση ενός αντικειμένου του τύπου 1 στο σακίδιο.

Αυτό, λοιπόν, το δίκτυο της εικόνας 7 μας δείχνει ότι η μέγιστη διαδρομή από τον κόμβο $(3,1)$ σε κάποιον από τους κόμβους του σταδίου 4 επιτυγχάνεται αν ακολουθήσουμε τη ροή $(3,1)-(1,2)-(1,3)-(1,4)$. Επισημαίνουμε ότι η βέλτιστη λύση σε ένα πρόβλημα σακιδίου δεν είναι απαραίτητο να χρησιμοποιήσει όλο το διαθέσιμο βάρος [12][13].

3.5 ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΛΑΝΟΔΙΟΥ ΠΩΛΗΤΗ (TRAVELLING SALESMAN PROBLEM)

Στην παράγραφο αυτή θα μελετήσουμε ένα κλασικό πρόβλημα που είναι γνωστό ως το “πρόβλημα του πλανόδιου πωλητή”[9]. Έχουμε έναν περιπλανώμενο πωλητή, ο οποίος επισκέπτεται διάφορες πόλεις με σκοπό να πουλήσει τα προϊόντα του. Το ζητούμενο σε αυτό το πρόβλημα είναι να βρεθεί η μικρότερη διαδρομή που ενώνει τις πόλεις αυτές, δεδομένου ότι ξεκινάει από την πόλη του και θέλει να επισκεφτεί ακριβώς μία φορά κάθε πόλη από το σύνολο των πόλεων που έχει στο πλάνο του και τελικά να επιστρέψει στην πόλη του. Αν είναι σε θέση να γνωρίζει την απόσταση κάθε ζεύγους πόλεων, τότε μπορεί να κάνει ένα σχεδιασμό - φαινομενικά τουλάχιστον, διότι δεν είναι διόλου βέβαιο ότι θα χρησιμοποιήσει σωστά τα δεδομένα αυτά, ώστε να βρει

[9] Το πρόβλημα του περιπλανώμενου πωλητή έχει τις ρίζες του στο 19^{ο} αιώνα, όταν ο Ιρλανδός μαθηματικός William Rowan Hamilton κατασκεύασε το Icosian Puzzle (από το αρχαίο ελληνικό είκοσι) που περιλαμβάνει την εύρεση ενός κύκλου από τις άκρες ενός δωδεκαέδρου, έτσι ώστε το αρχικό σημείο να συμπίπτει με το τελικό και κάθε κόμβος να επισκέπτεται ακριβώς μία φορά. Προς τιμήν του, ο κύκλος αυτός ονομάστηκε “κύκλος Hamilton”. Ένα μονοπάτι του Hamilton είναι ένα μονοπάτι σε μη κατευθυνόμενο γράφημα που επισκέπτεται μία μόνο φορά.

τη λύση στο πρόβλημά του - με τη σειρά με την οποία θα τις επισκεφτεί. Το πρόβλημα αυτό λύνεται με τη βοήθεια του Δυναμικού Προγραμματισμού.

Στη γενική του μορφή το πρόβλημα διατυπώνεται ως εξής: έχουμε ένα σύνολο από N κόμβους με αυθαίρετη αρίθμηση σε καθέναν από αυτούς και τις μεταξύ τους αποστάσεις γνωστές. Θέλουμε να βρούμε τη βέλτιστη (ελάχιστη) διαδρομή που θα ακολουθήσουμε, ώστε με αφετηρία τον κόμβο 1 να τερματίσουμε πάλι σε αυτόν, έχοντας προηγουμένως περάσει από τον κάθε κόμβο ακριβώς μία φορά.

Θα παραθέσουμε ένα παράδειγμα για να γίνουν αντιληπτά όσα περιγράφηκαν παραπάνω.

3.5.1 Εφαρμογή

Το τελευταίο Σαββατοκύριακο της 5ης Ιουλίου 2015 για τη διεξαγωγή του ελληνικού δημοψηφίσματος, η περιοδεία του πρωθυπουργού θα ολοκληρωνόταν με την επίσκεψή του στη Θεσσαλονίκη, τη Λάρισα και την Καλαμάτα κι επιστροφή στην έδρα του, την Αθήνα. Ο πρωθυπουργός ήθελε να ελαχιστοποιήσει τη συνολική απόσταση που έπρεπε να διανύσει. Ποια ήταν άραγε η σωστή σειρά για τις πόλεις που είχε να επισκεφτεί; Στη συνέχεια δίνεται ο πίνακας 15, όπου αναγράφονται οι αποστάσεις (σε χιλιόμετρα) όλων των πόλεων μεταξύ τους.

Γνωρίζουμε ότι το πρόγραμμα περιελάμβανε την επίσκεψη καθεμιάς από τις τρεις πόλεις, δεδομένου ότι θα εκκινούσε από την Αθήνα και θα επέστρεφε τελικά σε αυτή. Όταν έχει απομείνει μόνο μία πόλη να επισκεφτεί, το πρόβλημα είναι τετριμμένο: από την τρέχουσα πόλη πρέπει να πάει στην Αθήνα. Για ακόμη μία φορά θα χρησιμοποιήσουμε την οπισθοδρομική επαγωγή για την επίλυση ενός προβλήματος στο οποίο ο πρωθυπουργός βρίσκεται σε κάποια πόλη και του απομένουν μόνο δύο πόλεις να επισκεφτεί και τελικά βρίσκουμε τη συντομότερη διαδρομή που ξεκινάει από την Αθήνα και απομένουν ακόμα τέσσερις πόλεις προς επίσκεψη. Ο αριθμός των σταδίων αλλάζει κάθε φορά, ανάλογα με τον αριθμό των πόλεων που έχει ήδη επισκεφτεί έως τώρα. Σε κάθε στάδιο, για να αποφασίσουμε ποια είναι η επόμενη πόλη προς επίσκεψη, αρκεί να γνωρίζουμε ποια είναι η τρέχουσα τοποθεσία του, καθώς και ποιες πόλεις έχει ήδη επισκεφτεί. Η κατάσταση σε κάθε στάδιο αποτελείται από την πόλη που επισκέφτηκε τελευταία και από το σύνολο των πόλεων που έχει ήδη επισκεφτεί.

Πόλη	Αθήνα	Θεσσαλονίκη	Λάρισα	Καλαμάτα
Αθήνα	—	503	322	238
Θεσσαλονίκη	503	—	152	687
Λάρισα	322	152	—	549
Καλαμάτα	238	687	549	—

Πίνακας 15: Η χιλιομετρική απόσταση των πόλεων σταθμών της περιοδείας

Ορίζουμε ως $f_t(i,C)$ την ελάχιστη απόσταση που πρέπει να διανύσει ο πρωθυπουργός για να ολοκληρωθεί η περιοδεία, εάν έχει ήδη επισκεφτεί $t-1$ πόλεις του συνόλου C των πόλεων και η πόλη i είναι αυτή που επισκέφτηκε τελευταία. Επίσης, ορίζουμε d_{ij} να είναι η απόσταση μεταξύ των πόλεων i και j.

Προφανώς, έχουμε τέσσερα στάδια για τη διαδικασία επίλυσης του προβλήματος.

Στάδιο 4: Εδώ βρισκόμαστε στο τελευταίο στάδιο (αν και παρουσιάζεται πρώτο λόγω της προς-τα-πίσω επαγωγής), δηλαδή έχουμε επιστρέψει στην πόλη της Αθήνας (πόλη 1), απ' όπου ξεκινήσαμε, συνεπώς το σύνολο που έχουμε ήδη επισκεφτεί θα είναι το $C=\{2,3,4\}$. Το ζεύγος (i,C) μπορεί τότε να είναι ένα εκ των $(2,\{2,3,4\})$, $(3,\{2,3,4\})$ ή $(4,\{2,3,4\})$. Στο στάδιο αυτό, θέλουμε να μεταβούμε από την τρέχουσα πόλη στην Αθήνα. Έτσι, λαμβάνουμε τα ακόλουθα:

$$f_4(2,\{2,3,4\}) = d_{21} = 503 \;\rightarrow\; \text{Μετάβαση από την πόλη 2 στην πόλη 1}$$
$$f_4(3,\{2,3,4\}) = d_{31} = 322 \;\rightarrow\; \text{Μετάβαση από την πόλη 3 στην πόλη 1}$$
$$f_4(4,\{2,3,4\}) = d_{41} = 238 \;\rightarrow\; \text{Μετάβαση από την πόλη 4 στην πόλη 1}$$

Στάδιο 3: Προχωρώντας ένα βήμα πιο πίσω, παίρνουμε την αναδρομική σχέση:

$$f_3(i,C) = \min_{\substack{j \notin C \\ j \neq 1}} \; d_{ij} + f_4(j, C \cup \{j\}) \; .$$

Ας εξηγήσουμε λίγο αυτή τη σχέση. Αν ο πρωθυπουργός βρίσκεται στην πόλη i και ταξιδέψει μετά στην πόλη j, θα διανύσει απόσταση d_{ij}. Τότε, στο στάδιο 4, η τελευταία πόλη που επισκέφτηκε είναι η j και προφανώς, έχει ήδη επισκεφτεί τις πόλεις του συνόλου $C \cup \{j\}$. Άρα, το μήκος της υπόλοιπης περιοδείας του θα είναι $f_4(j, C \cup \{j\})$. Χρησιμοποιώντας την αναδρομική σχέση, στο στάδιο 3 έχει επισκεφτεί ένα από τα $\{2,3\}$, $\{2,4\}$ ή $\{3,4\}$ ζεύγη πόλεων και πρέπει ακόμα να επισκεφτεί (ανάλογα με το ζεύγος που επισκέφτηκε) την πόλη που δεν ανήκει στο σύνολο (ζεύγος) αυτό και η οποία δεν πρέπει να είναι η πόλη 1. Έτσι, βρίσκουμε την $f_3(\cdot)$ για όλες τις δυνατές καταστάσεις:

$$f_3(2,\{2,3\}) = d_{24} + f_4(4,\{2,3,4\}) = 687 + 238 = 925 \;\rightarrow\; \text{Μετάβαση από την 2 στην 4}$$
$$f_3(3,\{2,3\}) = d_{34} + f_4(4,\{2,3,4\}) = 549 + 238 = 787 \;\rightarrow\; \text{Μετάβαση από την 3 στην 4}$$
$$f_3(2,\{2,4\}) = d_{23} + f_4(3,\{2,3,4\}) = 152 + 322 = 474 \;\rightarrow\; \text{Μετάβαση από την 2 στην 3}$$
$$f_3(4,\{2,4\}) = d_{43} + f_4(3,\{2,3,4\}) = 549 + 322 = 871 \;\rightarrow\; \text{Μετάβαση από την 4 στην 3}$$
$$f_3(3,\{3,4\}) = d_{32} + f_4(2,\{2,3,4\}) = 152 + 503 = 655 \;\rightarrow\; \text{Μετάβαση από την 3 στην 2}$$
$$f_3(4,\{3,4\}) = d_{42} + f_4(2,\{2,3,4\}) = 687 + 503 = 1190 \;\rightarrow\; \text{Μετάβαση από την 4 στην 2}$$

Η γενική μορφή της αναδρομικής σχέσης είναι:

$$f_t(i,C) = \min_{\substack{j \notin C \\ j \neq 1}} \; d_{ij} + f_{t+1}(j, C \cup \{j\}) \;\; , \;\; t = 1,2,3 \, .$$

Η εξήγηση για τη σχέση αυτή είναι παρόμοια με την προηγούμενη. Αν η τρέχουσα θέση του πρωθυπουργού είναι η πόλη i και η επόμενη πόλη που θα επισκεφτεί είναι η j, τότε θα διανύσει απόσταση d_{ij}. Η υπόλοιπη περιοδεία του θα ξεκινάει τώρα από την πόλη j (που είναι και η τελευταία πόλη που επισκέφτηκε) και θα έχει επισκεφτεί τις πόλεις του $C \cup \{j\}$, ενώ το μήκος της θα είναι $f_{t+1}(j, C \cup \{j\})$.

Στάδιο 2: Στο στάδιο αυτό, γνωρίζουμε ότι ο πρωθυπουργός έχει επισκεφτεί μόνο μία πόλη, επομένως, οι δυνατές καταστάσεις είναι $(2,\{2\})$, $(3,\{3\})$ και $(4,\{4\})$. Εφαρμόζοντας την προηγούμενη αναδρομική σχέση λαμβάνουμε:

$$f_2(2,\{2\}) = \min \begin{cases} c_{23} + f_3(3,\{2,3\}) = 152 + 787 = 939 \;\rightarrow\; \text{Μετάβαση από την 2 στην 3} \\ c_{24} + f_3(4,\{2,4\}) = 687 + 871 = 1558 \;\rightarrow\; \text{Μετάβαση από την 2 στην 4} \end{cases}$$

$$f_2(3,\{3\}) = \min \begin{cases} c_{34} + f_3(4,\{3,4\}) = 549 + 1190 = 1739 \;\rightarrow\; \text{Μετάβαση από την 3 στην 4} \\ c_{32} + f_3(2,\{2,3\}) = 152 + 925 = 1077 \;\rightarrow\; \text{Μετάβαση από την 3 στην 2} \end{cases}$$

$$f_2(4,\{4\}) = \min \begin{cases} c_{42} + f_3(2,\{2,4\}) = 687 + 474 = 1161 \;\rightarrow\; \text{Μετάβαση από την 4 στην 2} \\ c_{43} + f_3(3,\{3,4\}) = 549 + 655 = 1204 \;\rightarrow\; \text{Μετάβαση από την 4 στην 3} \end{cases}$$

Στάδιο 1: Προχωρώντας ένα βήμα πιο πίσω, είμαστε στο στάδιο που βρίσκεται στην Αθήνα και ακόμη δεν έχει επισκεφτεί καμία πόλη, άρα η κατάσταση του σταδίου 1 είναι η $f_1(1,\{\cdot\})$. Εφαρμόζοντας την προηγούμενη αναδρομική σχέση παίρνουμε:

$$f_1(1,\{\cdot\}) = \min \begin{cases} d_{12} + f_2(2,\{2\}) = 503 + 939 = 1442 \;\rightarrow\; \text{Μετάβαση από την 1 στην 2} \\ d_{13} + f_2(3,\{3\}) = 322 + 1077 = 1399 \;\rightarrow\; \text{Μετάβαση από την 1 στην 3} \\ d_{14} + f_2(4,\{4\}) = 238 + 1161 = 1399 \;\rightarrow\; \text{Μετάβαση από την 1 στην 4} \end{cases}$$

Βλέπουμε, λοιπόν, ότι από την πόλη 1 (Αθήνα) μπορεί να μεταβεί είτε στην πόλη 3 (Λάρισα), είτε στην πόλη 4 (Καλαμάτα). Έστω ότι αυθαίρετα επιλέγει να επισκεφτεί πρώτα την πόλη 3. Τότε, πρέπει να επιλέξει την πόλη που του εξασφαλίζει $f_2(3,\{3\})$, άρα η πόλη 2 (Θεσσαλονίκη) θα είναι η επόμενη στάση του. Στη συνέχεια, πρέπει να επιλέξει να επισκεφτεί την πόλη που πετυχαίνει το $f_3(2,\{2,3\})$, άρα η πόλη 4 θα είναι η επόμενη στάση του. Τέλος, πρέπει να επιλέξει να επισκεφτεί την πόλη που πετυχαίνει $f_4(4,\{2,3,4\})$, που είναι η πόλη 1, όπως άλλωστε αναμέναμε.

Επομένως, η βέλτιστη διαδρομή που μπορεί να ακολουθήσει ο πρωθυπουργός είναι η $1 \to 3 \to 2 \to 4 \to 1$ ($\text{Αθήνα} \to \text{Λάρισα} \to \text{Θεσσαλονίκη} \to \text{Καλαμάτα} \to \text{Αθήνα}$), με το μήκος αυτής της διαδρομής να είναι 322+152+687+238=1399 χλμ., αφού οι αποστάσεις μεταξύ των ζευγών πόλεων που μας ενδιαφέρουν είναι:

$$d(\text{Αθήνα,Λάρισα}) = 322 \text{ χλμ.}$$
$$d(\text{Λάρισα,Θεσσαλονίκη}) = 152 \text{ χλμ.}$$
$$d(\text{Θεσσαλονίκη,Καλαμάτα}) = 687 \text{ χλμ.}$$
$$d(\text{Καλαμάτα,Αθήνα}) = 238 \text{ χλμ.}$$

Σημειώνουμε ότι το ίδιο αποτέλεσμα ως προς τη συνολική διανυθείσα απόσταση θα προέκυπτε στην περίπτωση που ο πρωθυπουργός επέλεγε αρχικά να επισκεφτεί την πόλη 4, με διαφορετική βεβαίως διαδρομή.

3.6 ΤΟ ΜΟΝΤΕΛΟ ΜΕΓΕΘΟΥΣ ΕΡΓΑΤΙΚΟΥ ΔΥΝΑΜΙΚΟΥ (THE WORKFORCE SIZE MODEL)

Έστω μια κατασκευαστική εταιρεία που έχει αναλάβει την πραγματοποίηση ενός έργου. Κατά τη διάρκεια αυτού του έργου, πραγματοποιούνται αλλεπάλληλες προσθήκες και μειώσεις εργατικού δυναμικού, ούτως ώστε να παραμένουν πάντα οι εργάτες που είναι χρήσιμοι. Τόσο οι προσλήψεις, όσο και απολύσεις συνεπάγονται επιπλέον κόστος για την εταιρεία, οπότε το ζητούμενο είναι να βρεθεί ένας τρόπος να διατηρηθεί το εργατικό δυναμικό σε όλη τη διάρκεια ζωής του έργου.

Ορίζουμε τα παρακάτω μεγέθη:

e_i : ο αριθμός των εργατών που συνιστούν το ελάχιστο δυναμικό προσωπικό που απαιτείται την i βδομάδα (αν και ιδανικά θα θέλαμε να είναι ακριβώς e_i)

n : οι εβδομάδες που αναμένεται να ολοκληρωθεί το έργο

x_i : ο αριθμός των εργατών που δουλεύουν την i βδομάδα

Ανάλογα με τις παραμέτρους κόστους, πιθανώς να είναι πιο συμφέρον να παρουσιάζει διακυμάνσεις το μέγεθος του εργατικού δυναμικού. Έτσι, μέσα στην i βδομάδα μπορεί να προκύψουν δύο ειδών κόστη:

$c_1(x_i - e_i)$: το κόστος διατήρησης μιας επιπλέον μονάδας εργατικού δυναμικού

$c_2(x_i - x_{i-1})$: το κόστος πρόσληψης επιπλέον $x_i - x_{i-1}$ εργατών

Έχουμε, λοιπόν, ότι το στάδιο i παριστάνει την i βδομάδα, για $i = 1, 2, \ldots, n$ και οι τιμές που μπορεί να πάρει είναι x_i . Η κατάσταση i παριστάνει τον αριθμό των διαθέσιμων εργατών x_{i-1} στο στάδιο $i-1$. Η αναδρομική σχέση γράφεται ως εξής:

$$f_i(\mathrm{x}_{i-1}) = \min_{x_i \geq e_i} \; c_1(x_i - e_i) + c_2(x_i - x_{i-1}) + f_{i+1}(\mathrm{x}_i)$$

και ισχύει $f_{n+1}(x_n) \equiv 0$.

3.6.1 Εφαρμογή

Ένας εργολάβος εκτιμά ότι για το έργο που έχει αναλάβει, το απαιτούμενο εργατικό δυναμικό θα είναι 5,7,8,4 και 6 εργάτες για τις επόμενες 5 βδομάδες. Αν οποιαδήποτε βδομάδα υπάρξουν νέες προσλήψεις, θα έχουν σταθερό κόστος 400 χρηματικές μονάδες συν 200 χρηματικές μονάδες ανά εργαζόμενο ανά εβδομάδα, ενώ παράλληλα το πλεονάζον εργατικό δυναμικό που παραμένει στην εταιρεία έχει κόστος 300 χρηματικές μονάδες. Θέλουμε να βοηθήσουμε τον εργολάβο να πάρει τη βέλτιστη για τον ίδιο και την εταιρεία του απόφαση, όσον αφορά τους εργάτες και την παραμονή τους ή μη στο πέρασμα των 5 βδομάδων.

Λύση

Από τα δεδομένα της εκφώνησης έχουμε ότι:

$e_1 = 5$, $e_2 = 7$, $e_3 = 8$, $e_4 = 4$, $e_5 = 6$ για τις 5 βδομάδες, αντίστοιχα

$c_1(x_i - e_i) = 3(x_i - e_i),\ \mathrm{x}_i > e_i,\ i = 1,\ldots,5$

$c_2(x_i - x_{i-1}) = 4 + 2(x_i - x_{i-1}),\ \mathrm{x}_i > x_{i-1},\ i = 1,\ldots,5$

(όπου τα c_1 και c_2 είναι εκφρασμένα σε χρηματικές μονάδες επί εκατό).

Χρησιμοποιούμε τη μέθοδο της οπισθοδρομικής επαγωγής.

Στάδιο 5: Ο αριθμός των εργατών την 5η βδομάδα είναι $e_1 = 5$. Λαμβάνουμε τον πίνακα 16:

	$c_1(x_5 - 6) + c_2(x_5 - x_4)$	Βέλτιστη λύση	
x_4	$x_5 = 6$	$f_5(x_4)$	x_5^*
4	$3 \cdot 0 + 4 + 2 \cdot 2 = 8$	8	6
5	$3 \cdot 0 + 4 + 2 \cdot 1 = 6$	6	6
6	$3 \cdot 0 + 0 = 0$	0	6

Πίνακας 16

Στάδιο 4: Πηγαίνοντας ένα βήμα πιο πίσω και με $e_2 = 7$ έχουμε:

	$c_1(x_4-4)+c_2(x_4-x_3)+f_5(x_4)$			Βέλτιστη λύση	
x_3	$x_4=4$	$x_4=5$	$x_4=6$	$f_4(x_3)$	x_4^*
8	$3\cdot 0+0+8=8$	$3\cdot 1+0+6=9$	$3\cdot 2+0+0=6$	6	6

Πίνακας 17

Στάδιο 3: Πηγαίνοντας ένα βήμα πιο πίσω και με $e_3 = 8$ έχουμε:

	$c_1(x_3-8)+c_2(x_3-x_2)+f_4(\mathrm{x}_3)$	Βέλτιστη λύση	
x_2	$x_3=8$	$f_3(x_2)$	x_3^*
7	$3\cdot 0+4+2\cdot 1+6=12$	12	8
8	$3\cdot 0+0+6=6$	6	8

Πίνακας 18

Στάδιο 2: Πηγαίνοντας ένα βήμα πιο πίσω και με $e_4 = 4$ έχουμε:

	$c_1(x_3-8)+c_2(x_3-x_2)+f_4(\mathrm{x}_3)$		Βέλτιστη λύση	
x_1	$x_2=7$	$x_2=8$	$f_2(x_1)$	x_2^*
5	$3\cdot 0+4+2\cdot 2+12=20$	$3\cdot 1+4+2\cdot 3+6=19$	19	8
6	$3\cdot 0+4+2\cdot 1+12=18$	$3\cdot 1+4+2\cdot 2+6=17$	17	8
7	$3\cdot 0+0+12=12$	$3\cdot 1+4+2\cdot 1+6=15$	12	7
8	$3\cdot 0+0+12=12$	$3\cdot 1+0+6=9$	9	8

Πίνακας 19

Στάδιο 1: Τέλος, στο στάδιο 1 με $e_5 = 6$ έχουμε:

x_0	$c_1(x_1-5)+c_2(x_1-x_0)+f_2(\mathrm{x}_1)$				Βέλτιστη λύση	
	$x_1 = 5$	$x_1 = 6$	$x_1 = 7$	$x_1 = 8$	$f_1(x_0)$	x_1^*
0	$3 \cdot 0+4+2 \cdot 5$ $+19=33$	$3 \cdot 1+4+2 \cdot 6$ $+17=36$	$3 \cdot 2+4+2 \cdot 7$ $+12=36$	$3 \cdot 2+4+2 \cdot 8$ $+9=35$	33	5

Πίνακας 20

Επομένως, η βέλτιστη λύση στο πρόβλημά μας ακολουθεί τη ροή $x_0 = 0 \to x_1^* = 5 \to x_2^* = 8 \to x_3^* = 8 \to x_4^* = 6 \to x_5^* = 6$. Άρα, ο εργολάβος αποφασίζει να δράσει ως εξής: την 1η βδομάδα προσλαμβάνει 5 εργάτες, τη 2η προσλαμβάνει 3 εργάτες (8-5=3), την 3η δεν υπάρχει μεταβολή, την 4η (6-8=-2) απολύει 2 εργάτες και την 5η βδομάδα δεν αλλάζει τίποτα.

3.7 ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ ΑΝΤΙΚΑΤΑΣΤΑΣΗΣ ΕΞΟΠΛΙΣΜΟΥ (THE EQUIPMENT REPLACEMENT PROBLEM)

Μία εταιρεία επιθυμεί να αγοράσει ένα συγκεκριμένο τύπο μηχανής για τα επόμενα n χρόνια. Όπως είναι φυσικό, όσο αυξάνεται η “ηλικία” της μηχανής, δηλαδή όσο περισσότερο μένει σε λειτουργία, τόσο αυξάνεται το ετήσιο κόστος συντήρησής της και μειώνεται η παραγωγικότητά της, συνεπώς, πρέπει να αντικατασταθεί από μία καινούρια. Το πρόβλημά μας έγκειται στον προσδιορισμό της πιο οικονομικής ηλικίας της μηχανής (δηλαδή του ορίου της ηλικίας της που πρέπει να αντικαθίσταται, ώστε να είναι πιο συμφέρουσα για την εταιρεία).

Στο ξεκίνημα κάθε χρονιάς, η εταιρεία αποφασίζει αν θα κρατήσει τη μηχανή για έναν ακόμη χρόνο ή θα την αντικαταστήσει με μία καινούρια. Έστω t η ηλικία της μηχανής στην αρχή της χρονιάς και $a(t)$, $c(t)$ και $p(t)$ τα ετήσια έσοδα, τα λειτουργικά έξοδα και το κόστος συντήρησης της μηχανής για ένα χρόνο, αντίστοιχα. Έστω, επίσης, C το κόστος αγοράς καινούριας μηχανής κάθε χρόνο. Θέλουμε να προσδιορίσουμε την πολιτική αντικατάστασης της μηχανής που ελαχιστοποιεί το συνολικό κόστος της μηχανής για τα επόμενα n χρόνια, δεδομένου, όπως είπαμε νωρίτερα, ότι η απόφαση για πιθανή αντικατάσταση μπορεί να ληφθεί μόνο στην αρχή κάθε χρονιάς.

Η ανάλυση του προβλήματος με χρήση Δυναμικού Προγραμματισμού περιλαμβάνει n στάδια που αντιστοιχούν σε n χρόνια. Στην αρχή του i σταδίου, η κατάσταση του συστήματος μπορεί να προσδιοριστεί πλήρως από την ηλικία της μηχανής που έχει χρησιμοποιηθεί και την προηγούμενη χρονιά και η απόφαση είναι είτε να παραμείνει η μηχανή σε λειτουργία, είτε να αντικατασταθεί από καινούρια. Αν αποφασιστεί να παραμείνει η μηχανή, τότε, το μοναδικό κόστος θα είναι το λειτουργικό κόστος της μηχανής για το τρέχον στάδιο. Αν, όμως, η μηχανή αντικατασταθεί, τότε, το ετήσιο κόστος θα περιλαμβάνει και το κόστος απόκτησης και κόστος λειτουργίας της νέας μηχανής. Το κόστος στο στάδιο i εξαρτάται από την ηλικία της τρέχουσας μηχανής και την απόφαση διατήρησής της ή απόκτησης καινούριας.

Τα χαρακτηριστικά του μοντέλου του Δυναμικού Προγραμματισμού είναι :

- το στάδιο i αντιστοιχεί στη χρονιά i, για $i = 1, 2, ..., n$
- η βέλτιστη πολιτική απόφασης στο στάδιο i είναι είτε να κρατήσουμε την ήδη υπάρχουσα μηχανή, είτε να αγοράσουμε καινούρια
- η κατάσταση του συστήματος στο στάδιο i είναι η ηλικία της μηχανής στην αρχή του σταδίου (χρονιάς) i

- η βέλτιστη συνάρτηση τιμής $f_i(t)$ είναι το μέγιστο κέρδος που μπορεί να προκύψει από την κατοχή μιας μηχανής από την έναρξη της χρονιάς i έως το τέλος της χρονιάς n, δεδομένου ότι η ηλικία της είναι t στην αρχή της χρονιάς i

Ορίζουμε λοιπόν την αναδρομική σχέση:

$$f_i(t) = \max \begin{cases} a(t) - c(t) + f_{i+1}(t+1) & \text{, αν κρατήσουμε τη μηχανή} \\ a(0) + p(t) - C - c(0) + f_{i+1}(1) & \text{, αν αντικαταστήσουμε τη μηχανή} \end{cases}$$

με $f_{n+1}(\cdot) \equiv 0$.

3.7.1 Εφαρμογή

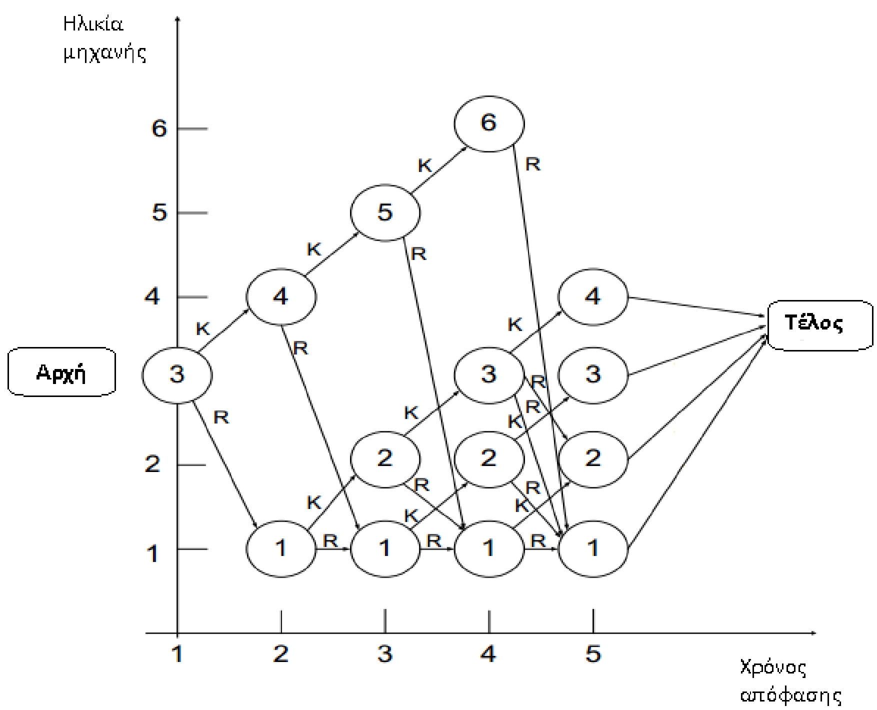
Έστω μία εταιρεία που θέλει να αποφασίσει την πολιτική που θα ακολουθήσει για τα επόμενα 4 χρόνια. Στην αρχή της 1ης χρονιάς, η ηλικία της μηχανής είναι 3 έτη. Το κόστος απόκτησης μιας νέας μηχανής είναι 80.000 χρηματικές μονάδες. Η εταιρεία απαιτεί η διάρκεια ζωής μιας μηχανής να είναι το πολύ 6 έτη. Τα ετήσια έσοδα, τα λειτουργικά έξοδα και το κόστος συντήρησης ανά έτος παρουσιάζονται στον πίνακα 21. Τα κόστη δίνονται σε χιλιάδες χρηματικές μονάδες.

Έτος (t)	Ετήσια έσοδα $a(t)$	Λειτουργικά έξοδα $c(t)$	Κόστος συντήρησης $p(t)$
0	30	0,27	-
1	27	0,5	60
2	23,5	0,8	50
3	21,8	1	35
4	19	1,3	25
5	17,6	1,5	17
6	15	1,8	13

Πίνακας 21

Λύση

Επειδή είναι σχετικά δύσκολο να καθοριστούν οι δυνατές τιμές της ηλικίας της μηχανής σε κάθε στάδιο, παραθέτουμε και τη σχηματική αναπαράσταση του προβλήματος (εικόνα 8), όπου φαίνονται πιο καθαρά οι αποφάσεις σε κάθε στάδιο.



Εικόνα 8: Αναπαράσταση προβλήματος σε δίκτυο

Συμβολισμός: **K**: η απόφαση να κρατήσει η εταιρεία τη μηχανή

R: η απόφαση να την αντικαταστήσει

Ας εξηγήσουμε λίγο τη σχήμα μας. Την 1η χρονιά, αν η εταιρεία έχει αποφασίσει να αντικαταστήσει τη μηχανή, τότε η καινούρια θα είναι 1 έτους, ενώ η παλιά 4 ετών. Γενικά, σε κάθε χρονιά ισχύει η ίδια λογική. Επίσης, αν μία μηχανή 1 έτους αντικατασταθεί στην αρχή μιας νέας χρονιάς, η αντικατάστασή της θα είναι 1 έτους στην αρχή της νέας χρονιάς. Στην αρχή της 4ης χρονιάς, αν η μηχανή είναι 6 ετών θα αντικατασταθεί οπωσδήποτε, λόγω της πολιτικής της εταιρείας. Στο τέλος της 4ης χρονιάς, συγκεντρώνονται όλες οι υπάρχουσες μηχανές.

Εφόσον ξεκινάμε την 1η χρονιά με μία μηχανή 3 ετών, στην αρχή της 2ης χρονιάς η ηλικία της μηχανής που θα έχουμε θα είναι 4 έτη και 1 έτος. Στην αρχή της 3ης χρονιάς, η μηχανή μπορεί να πάρει τις τιμές 1,2 και 5, ενώ στην αρχή της 4ης χρονιάς μπορεί να πάρει τις τιμές 1,2,3 και 6.

Θέλουμε να βρούμε τη μεγαλύτερη διαδρομή από τον 1ο έως τον 4ο χρόνο. Η επίλυση του προβλήματος θα γίνει μέσω της χρήσης πινάκων για κάθε στάδιο και της οπισθοδρομικής επαγωγής. Σημειωτέον ότι αν μια μηχανή αντικατασταθεί στο τέλος της 4ης χρονιάς (που είναι και το τέλος του χρονικού πλάνου της εταιρείας), στα έσοδα συμπεριλαμβάνονται το κόστος συντήρησης της καινούριας μηχανής, αλλά και το κόστος συντήρησης της μηχανής που αντικαταστάθηκε.

Στάδιο 4:

	K	**R**	Βέλτιστη λύση	
t	$a(t)+p(t+1)-c(t)$	$a(0)+p(t)+p(1)-c(0)-C$	$f_4(t)$	Απόφαση
1	27+50-0,5=76,5	30+60+60-0,27-80=69,73	76,5	**K**
2	23,5+35-0,8=57,7	30+50+60-0,27-80=59,73	59,73	**R**
3	21,8+25-1=45,8	30+35+60-0,27-80=44,73	45.8	**K**
6	**R**	30+13+60-0,27-80=22,73	22,73	**R**

Στάδιο 3:

	K	R	Βέλτιστη λύση	
t	$a(t)-c(t)+f_4(t+1)$	$a(0)+p(t)-c(0)-C+f_4(1)$	$f_3(t)$	Απόφαση
1	27-0,5+59,82=86,32	30+60-0,27-80+76,5=86,23	86,32	**K**
2	23,5-0,8+45,8=68,5	30+50-0,27-80+76,5=76,32	76,23	**R**
5	17,6-1,5+22,73=38,83	30+17-0,27-80+76,5=43,23	43,23	**R**

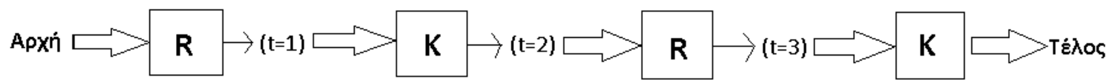
Στάδιο 2:

	K	R	Βέλτιστη λύση	
t	$a(t)-c(t)+f_3(t+1)$	$a(0)+p(t)-c(0)-C+f_3(1)$	$f_2(t)$	Απόφαση
1	27-0,5+76,23=102,73	30+60-0,27-80+86,32=96,05	102,73	**K**
4	19-1,3+43,23=60,93	30+25-0,27-80+86,32=61,05	61,05	**R**

Στάδιο 1:

	K	R	Βέλτιστη λύση	
t	$a(t)-c(t)+f_2(t+1)$	$a(0)+p(t)-c(0)-C+f_2(1)$	$f_1(t)$	Απόφαση
3	27-0,5+61,05=87,55	30+60-0,27-80+102,73=112,46	112,46	**R**

Η βέλτιστη λύση που προκύπτει από το σχήμα της εικόνας 4 μας λέει ακριβώς τούτο: στην αρχή της 1ης χρονιάς, όπου η μηχανή είναι 3 ετών, η καλύτερη απόφαση είναι να αντικατασταθεί. Η καινούρια μηχανή στην αρχή της 2ης χρονιάς θα είναι 1 έτους και αποφασίζεται η διατήρησή της, οπότε στην αρχή της 3ης χρονιάς θα γίνει 2 ετών. Τότε, αποφασίζεται να αντικατασταθεί, οπότε η καινούρια μηχανή θα είναι 1 έτους. Στην αρχή του 4ου έτους αποφασίζεται η διατήρηση της 2 ετών πλέον μηχανής. Συνοπτικά, η λύση μας φαίνεται στην εικόνα 5. Το συνολικό κόστος είναι 112.460 χρηματικές μονάδες [11][17][18].



Εικόνα 9: Συνοπτική λύση του προβλήματος

3.8 ΤΟ ΕΠΕΝΔΥΤΙΚΟ ΜΟΝΤΕΛΟ

Ο Δυναμικός Προγραμματισμός βρίσκει εφαρμογή ακόμη και σε προβλήματα χρηματοοικονομικής και χρηματιστηρίου. Εκεί, οι αποφάσεις που πρέπει να ληφθούν είναι πολλές και αλληλοεξαρτώμενες μεταξύ τους και στόχος είναι πάντα το βέλτιστο τελικό αποτέλεσμα. Ας δούμε ένα σχετικό παράδειγμα.

3.8.1 *Εφαρμογή*

Έστω ένας πελάτης μίας τράπεζας που επιθυμεί να επενδύσει το ποσό των 10.000€ για τα επόμενα τρία χρόνια. Επενδύσεις μπορούν να γίνουν μόνο για ένα χρόνο στην αρχή κάθε χρονιάς σε πολλαπλάσια των 10.000€. Οι διαθέσιμοι τύποι επενδύσεων αυτή την περίοδο είναι δύο: η Α και η Β. Αν ο πελάτης αποφασίσει να επενδύσει στην Α για ένα συγκεκριμένο χρόνο , τότε στο τέλος αυτής της χρονιάς μπορεί να χάσει τα χρήματά του με πιθανότητα 0,3 ή να τα διπλασιάσει με πιθανότητα 0,7. Αν τα επενδύσει στην Β που είναι πιο συντηρητική από την Α, τότε στο τέλος της χρονιάς θα πάρει πίσω τα χρήματά του στο ακέραιο με πιθανότητα 0,9 ή θα τα διπλασιάσει με πιθανότητα 0,1. Στόχος μας σε αυτό το πρόβλημα είναι να βρούμε τη βέλτιστη πολιτική επένδυσης που μπορεί να ακολουθήσει ο πελάτης, ώστε να μεγιστοποιήσει τη συνολική αναμενόμενη επιστροφή χρημάτων στο τέλος της τρίτης χρονιάς. Θεωρούμε για ευκολία ότι μπορεί να επιλέξει μόνο μία εκ των δύο δυνατών επενδύσεων.

Λύση

Όπως είναι προφανές, το σύνολο των δυνατών αποφάσεων σε κάθε στάδιο εξαρτάται από το ποσό των χρημάτων που διαθέτει ο πελάτης στην αρχή μιας χρονιάς. Αν π.χ. έχει στην κατοχή του λιγότερα από 10.000€, η μόνη του επιλογή είναι να κρατήσει τα χρήματά του. Αν, όμως, έχει διαθέσιμα για επένδυση 10.000€ ή και περισσότερα, πολλαπλάσια των 10.000€ μπορούν να επενδυθούν στην Α ή Β επένδυση.

Ας ορίσουμε τώρα τα μεγέθη που μας χρειάζονται για την επίλυση του προβλήματος με χρήση Δυναμικού Προγραμματισμού.

- Το στάδιο i αντιπροσωπεύει τη χρονιά i, για $i = 1, 2, 3$.
- Η κατάσταση s_i αντιπροσωπεύει το ποσό των χρημάτων που διαθέτει ο πελάτης στην αρχή της χρονιάς i. Προφανώς, την πρώτη χρονιά θα είναι $s_1 = 10.000€$.
- Η μεταβλητή απόφασης θα είναι η $x_i = x_{i1}, x_{i2}$, δηλαδή η απόφαση στο στάδιο i αντιπροσωπεύει το ποσό x_{i1} που μπορεί να επενδύσει σε πολλαπλάσια των 10.000€ και $x_{i2} \in \{0, \mathrm{A}, \mathrm{B}\}$ τον τύπο της επένδυσης που θα επιλέξει, με το 0 να υπονοεί ότι δε θα κάνει καμία επένδυση.

Ορίζουμε:

$f_i(s_i)$: το μέγιστο αναμενόμενο ποσό χρημάτων από την αρχή της i χρονιάς ως το τέλος της τρίτης χρονιάς, δεδομένου ότι τα διαθέσιμα χρήματα είναι s_i στην αρχή της χρονιάς i

$p_i(s_i)=x_i^*$: το βέλτιστο επενδυτικό πλάνο για την i χρονιά, δεδομένου ότι τα διαθέσιμα χρήματα είναι s_i

Για την αναδρομική σχέση λαμβάνουμε τα δεδομένα $0 \le s_i < 10$, $x_i = x_{i1}, x_{i2} = (0,0)$ και $f_i(s_i)=s_i$. Για $s_i \ge 10$ γράφουμε:

$$f_i(s_i)=\max\begin{cases} f_{i+1}(s_i) & ,\ \text{για } (x_{i1},x_{i2})=(0,0) \\ \max\limits_{x_{i1}\in \mathrm{A}} \ 0,3 f_{i+1}(s_i - x_{i1}) + 0,7 f_{i+1}(s_i + x_{i1}) & ,\ \text{για } x_{i2}\text{=A} \\ \max\limits_{x_{i1}\in \mathrm{A}} \ 0,9 f_{i+1}(s_i) + 0,1 f_{i+1}(s_i + x_{i1}) & ,\ \text{για } x_{i2}\text{=B} \end{cases},$$

όπου: $\mathrm{A}=\{a:\ a=10,20,...,a\le s_i\}$ είναι το σύνολο των δυνατών ποσοτήτων προς επένδυση και τα ποσά είναι εκφρασμένα σε χιλιάδες ευρώ.

Οι συνοριακές συνθήκες λαμβάνονται για $0 \le s_3 < 10$, $x_3 = x_{31}, x_{32} = (0,0)$ και $f_3(s_3)=s_3$. Για $s_3 \ge 10$ γράφουμε:

$$f_3(s_3)=\max\begin{cases} s_3 & ,\ \text{για } (x_{31},x_{32})=(0,0) \\ \max\limits_{x_{31}\in \mathrm{A}} \ 0,3(s_3 - x_{31}) + 0,7(s_3 + x_{31}) & ,\ \text{για } x_{32}\text{=A} \\ \max\limits_{x_{31}\in \mathrm{A}} \ 0,9(s_3) + 0,1(s_3 + x_{31}) & ,\ \text{για } x_{32}\text{=B} \end{cases},$$

όπου: $\mathrm{A}=\{a:\ a=10,20,...,a\le s_3\}$.

Η απάντηση στο πρόβλημά μας είναι το $f_1(10)$. Για να φτάσουμε στην απάντηση αυτή θα χρησιμοποιήσουμε για ακόμη μία φορά τη μέθοδο της οπισθοδρομικής επαγωγής.

***Στάδιο 3* ($i=3$):**

		s_3	$0,3(s_3-x_{31})+0,7(s_3+x_{31})$	$0,9(s_3)+0,1(s_3+x_{31})$			
s_3	x_{31}	$x_{32}=0$	$x_{32}=\mathrm{A}$	$x_{32}=\mathrm{B}$	$f_3(s_3)$	x_{31}^*	x_{32}^*
0	0	0	-	-	0	0	0
10	0	10	-	-			
	10	-	14	11	14	10	A
20	0	20	-	-			
	10	-	24	21			
	20	-	28	22	28	20	A
30	0	30	-	-			
	10	-	34	31			
	20	-	38	32			
	30	-	42	33	42	30	A
40	0	40	-	-			
	10	-	44	41			
	20	-	48	42			
	30	-	52	43			
	40	-	56	44	56	40	A

***Στάδιο 2* ($i=2$):**

s_2	x_{21}	$f_3(s_2)$ $x_{22}=0$	$0,3f_3(s_2-x_{21})+0,7f_3(s_2+x_{21})$ $x_{22}=\mathrm{A}$	$0,9f_3(s_2)+0,1f_3(s_2+x_{21})$ $x_{22}=\mathrm{B}$	$f_2(s_2)$	x_{21}^*	x_{22}^*
0	0	$f_3(0)=0$	-	-	0	0	0
10	0	$f_3(10)=14$	-	-			
	10	-	19,6		19,6	10	A
20	0	$f_3(20)=28$	-	-			
	10	-	33,6	29,4			
	20	-	39,2	30,8	39,2	20	A

***Στάδιο 1* ($i=1$):**

s_1	x_{11}	$f_2(s_1)$ $x_{12}=0$	$0,3f_2(s_1-x_{11})+0,7f_2(s_1+x_{11})$ $x_{12}=\mathrm{A}$	$0,9f_2(s_1)+0,1f_2(s_1+x_{11})$ $x_{12}=\mathrm{B}$	$f_1(s_1)$	x_{11}^*	x_{12}^*
10	0	$f_2(0)=19,6$	-	-			
	10	-	27,3	21,56	27,3	10	A

Μετά από όλα αυτά γίνεται φανερό ότι η βέλτιστη λύση για τον πελάτη είναι να επενδύσει τα 10.000€ στην επένδυση Α, αφού προέκυψε $x_1^*=x_{11}^*,x_{12}^*=(10,\mathrm{A})$, ενώ το συνολικό αναμενόμενο ποσό των χρημάτων που θα έχει στο τέλος της τρίτης χρονιάς είναι $f_1(10)=27.300€$ [11][22].

3.9 ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΕΠΙΤΡΕΠΤΩΝ ΖΗΜΙΩΝ (DETERMINING REJECT ALLOWANCES)

Έστω μία κατασκευαστική εταιρεία που έχει αναλάβει μία παραγγελία για προμήθεια ενός συγκεκριμένου τύπου προϊόντος. Οι αυστηρές απαιτήσεις ποιότητας που έχει επιβάλλει ο πελάτης, ωστόσο, επιβάλλει στην εταιρεία να κατασκευάσει πιθανώς περισσότερα από ένα τέτοια προϊόντα, ώστε να έχει την ασφάλεια ότι το προϊόν που τελικά θα προμηθεύσει θα γίνει αποδεκτό. Ο αριθμός των επιπλέον προϊόντων που παράγονται σε έναν κύκλο παραγωγής ονομάζεται *επιτρεπτή ζημιά* και αποτελεί συνήθη πρακτική κατά την παραγωγή της παραγγελίας ενός πελάτη.

Ο κατασκευαστής εκτιμά ότι κάθε προϊόν αυτού του τύπου που παράγεται έχει $\frac{1}{2}$ πιθανότητα να είναι αποδεκτό και $\frac{1}{2}$ να είναι ελαττωματικό (χωρίς να υπάρχει δυνατότητα επανεπεξεργασίας όμως). Ο αριθμός των αποδεκτών προϊόντων ενός μεγέθους παραγωγής L θα ακολουθεί τη διωνυμική κατανομή (δηλαδή, η πιθανότητα παραγωγής απορριφθέντων τελικά προϊόντων σε ένα τέτοιο δείγμα είναι $\left(\frac{1}{2}\right)^L$). Το οριακό κόστος παραγωγής αυτού του προϊόντος εκτιμάται στις 100 χρηματικές μονάδες ανά μονάδα προϊόντος (ακόμα κι αν είναι ελαττωματικό), με τα πλεονάζοντα προϊόντα να είναι άνευ αξίας. Κάθε φορά που ξεκινά η παραγωγική περίοδος για το προϊόν αυτό, το κόστος εγκατάστασης υπολογίζεται στις 300 χρηματικές μονάδες και απαιτείται, επίσης, μία νέα εγκατάσταση ιδίου κόστους για κάθε επόμενη παραγωγική περίοδο, αν έπειτα από επιθεώρηση διαπιστωθεί ότι σε μία ολοκληρωμένη παρτίδα προέκυψε μη αποδεκτό προϊόν. Τότε, ο κατασκευαστής έχει τη δυνατότητα να διεξάγει το πολύ μέχρι τρεις παραγωγικές περιόδους. Αν ούτε και μετά την τρίτη παραγωγική περίοδο παραχθεί αποδεκτό προϊόν, το κόστος σε χαμένες πωλήσεις και σε ποινή θα αγγίζει τις 1600 χρηματικές μονάδες.

Στόχος μας είναι να καθορίσουμε την πολιτική που πρέπει να ακολουθήσει η κατασκευάστρια εταιρεία σχετικά με το μέγεθος της παρτίδας για την απαιτούμενη περίοδο παραγωγής (ή περιόδους παραγωγής, ανάλογα), ώστε να ελαχιστοποιείται το αναμενόμενο συνολικό κόστος.

Λύση

Θα ορίσουμε τα μεγέθη που μας χρειάζονται για την επίλυση του προβλήματος αυτού με χρήση του Δυναμικού Προγραμματισμού.

- Το στάδιο n αντιπροσωπεύει τη n περίοδο παραγωγής, για $n = 1, 2, 3$.
- Το μέγεθος της παραγωγής στο στάδιο n είναι x_n.
- Η κατάσταση s_n του συστήματος είναι ο αριθμός των αποδεκτών προϊόντων που είναι ακόμη απαραίτητα (0 ή 1) στην αρχή του σταδίου n.

Άρα, στο στάδιο 1 η κατάσταση θα είναι $s_1 = 1$ (αφού δεν έχει παραχθεί ακόμα κανένα προϊόν). Αν στη συνέχεια παραχθεί τουλάχιστον ένα αποδεκτό προϊόν, η κατάσταση αλλάζει σε $s_n = 0$ (αφού πλέον ο στόχος έχει επιτευχθεί) και δεν υπάρχει επιπρόσθετο κόστος.

Ορίζουμε:

$f_n(s_n, x_n)$: το συνολικό αναμενόμενο κόστος για τα στάδια $n, \ldots, 3$, αν το σύστημα ξεκινώντας στο στάδιο n βρίσκεται στην κατάσταση s_n, η άμεση απόφαση είναι x_n και έπειτα λαμβάνονται βέλτιστες αποφάσεις

$f_n^*(s_n) = \min\limits_{x_n = 0,1,\ldots} f_n(s_n, x_n)$, όπου $f_n^*(0) = 0$.

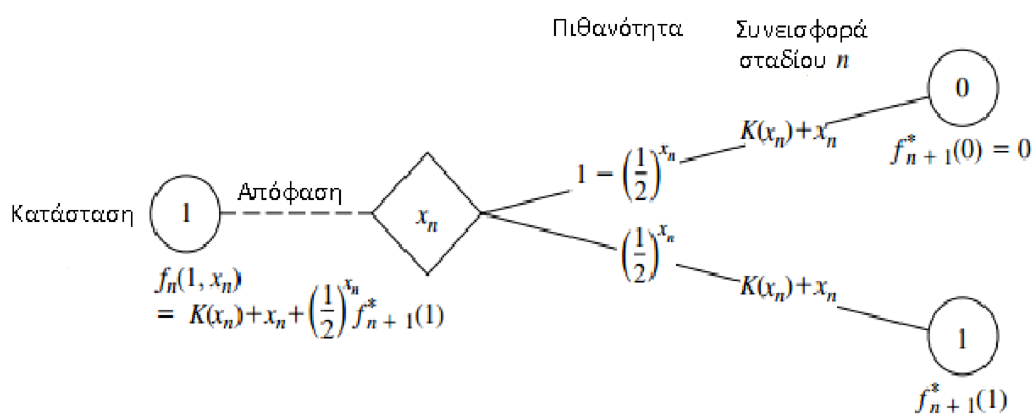
Χρησιμοποιώντας ως μονάδα χρήματος τις 100 εκατό χρηματικές μονάδες, η συνεισφορά του σταδίου n στο κόστος ανεξάρτητα από την επόμενη κατάσταση είναι $K(x_n)+x_n$,

όπου: $K(x_n)$: συνάρτηση του x_n, τέτοια ώστε $K(x_n)=\begin{cases}0, \text{ αν } x_n=0\\3, \text{ αν } x_n>0\end{cases}$.

Άρα, για $s_n=1$ έχουμε:

$f_n(1,x_n)=K(x_n)+x_n+\left(\frac{1}{2}\right)^{x_n} f_{n+1}^*(1)+\left[1-\left(\frac{1}{2}\right)^{x_n}\right] f_{n+1}^*(0)=K(x_n)+x_n+\left(\frac{1}{2}\right)^{x_n} f_{n+1}^*(1)$, όπου: $f_4^*(1)=16$ (με βάση τα δεδομένα του προβλήματος) είναι το τελικό κόστος αν δεν παραχθεί κανένα αποδεκτό προϊόν.

Όλα αυτά μπορούμε να τα δούμε και διαγραμματικά στην εικόνα 10.



Εικόνα 10: Διαγραμματική απεικόνιση του προβλήματος επιτρεπτών ζημιών

Τώρα μπορούμε να γράψουμε και την αναδρομική σχέση:

$$f_n^*(1)=\min_{x_n=0,1,\ldots}\left\{K(x_n)+x_n+\left(\frac{1}{2}\right)^{x_n} f_{n+1}^*(1)\right\}, \text{ για } n=1,2,3\,.$$

Με τη βοήθεια της οπισθοδρομικής επαγωγής υπολογίζουμε την αναδρομική σχέση για τα τρία στάδια.

Στάδιο 3 **(** $n=3$ **):**

s_3 \ x_3	$f_3(1,x_3)=K(x_3)+x_3+\left(\frac{1}{2}\right)^{x_3} f_4^*(1)$						$f_3^*(s_3)$	x_3^*
	0	1	2	3	4	5		
0	0	-	-	-	-	-	0	0
1	16	12	9	8	8	8,5	8	3 ή 4

όπου: $f_4^*(1)=16$, όπως είπαμε και νωρίτερα.

Στάδιο 2 **(** $n=2$ **):**

s_2 \ x_2	$f_2(1,x_2)=K(x_2)+x_2+\left(\frac{1}{2}\right)^{x_2} f_3^*(1)$					$f_2^*(s_2)$	x_2^*
	0	1	2	3	4		
0	0	-	-	-	-	0	0
1	8	8	7	7	7,5	7	2 ή 3

Στάδιο 1 **(** $n=1$ **):**

s_1 \ x_1	$f_1(1,x_1)=K(x_1)+x_1+\left(\frac{1}{2}\right)^{x_1} f_2^*(1)$					$f_1^*(s_1)$	x_1^*
	0	1	2	3	4		
1	7	7,5	6,75	6,875	7,4375	6,75	2

Συμπεραίνουμε, λοιπόν, ότι η βέλτιστη πολιτική που μπορεί να ακολουθήσει η εταιρεία είναι να παράγει δύο προϊόντα στην 1η περίοδο παραγωγής. Αν δεν προκύψει κανένα αποδεκτό προϊόν, στη 2η περίοδο παραγωγής θα πρέπει να παράγει δύο ή τρία προϊόντα. Αν και πάλι δεν προκύψει κανένα αποδεκτό προϊόν, τότε, στην 3η περίοδο παραγωγής θα πρέπει να παράγει τρία ή τέσσερα προϊόντα. Το συνολικό αναμενόμενο κόστος αυτής της πολιτικής είναι 675 χρηματικές μονάδες [10].

3.10 ΝΙΚΗ ΣΕ ΤΥΧΕΡΟ ΠΑΙΧΝΙΔΙ (WINNING IN A LUCKY GAME)

Έστω ότι παίζεται ένα τυχερό παιχνίδι στο καζίνο. Ένας μαθηματικός θεωρεί ότι έχει ανακαλύψει ένα σύστημα που θα τον οδηγήσει στη νίκη. Οι φίλοι του τον αμφισβητούν και βάζουν στοίχημα μαζί του ότι αν ξεκινήσει το παιχνίδι με τρεις μάρκες, μετά από τρεις παρτίδες θα έχει το πολύ πέντε μάρκες. Σε κάθε παρτίδα, οι παίκτες μπορούν να στοιχηματίσουν όσες από τις διαθέσιμες μάρκες τους θέλουν και μετά είτε κερδίζουν, είτε χάνουν αυτά που έχουν ποντάρει. Ο μαθηματικός πιστεύει ότι βάσει του συστήματός του θα κερδίσει μία συγκεκριμένη παρτίδα με πιθανότητα $\frac{2}{3}$.

Αν υποθέσουμε ότι το σύστημα του μαθηματικού είναι σωστό, θέλουμε χρησιμοποιώντας το Δυναμικό Προγραμματισμό να προσδιορίσουμε τη βέλτιστη πολιτική σχετικά με τον αριθμό των μαρκών που θα πρέπει να ποντάρει σε καθεμιά από τις τρεις παρτίδες. Η απόφαση που λαμβάνεται σε κάθε παρτίδα εξαρτάται από τα αποτελέσματα των προηγούμενων παρτίδων. Στόχος μας είναι να μεγιστοποιήσουμε την πιθανότητα να κερδίσει το στοίχημα με τους φίλους του.

Λύση

Θα ορίσουμε τα μεγέθη που μας χρειάζονται για την επίλυση του προβλήματος με χρήση Δυναμικού Προγραμματισμού.

- Το στάδιο n αντιπροσωπεύει τη n-στή παρτίδα, για $n=1,2,3$.
- Το x_n παριστάνει τον αριθμό των μαρκών που στοιχηματίζει ο παίκτης στο στάδιο n.
- Η κατάσταση s_n του συστήματος είναι ο αριθμός των μαρκών που διαθέτει ο παίκτης ξεκινώντας στο στάδιο n.

Ο τρόπος με τον οποίο ορίζεται μία κατάσταση δίνει την πληροφορία για την τρέχουσα κατάσταση, ώστε να μπορεί ο παίκτης να αποφασίσει βέλτιστα τον αριθμό των μαρκών που θα ποντάρει στον επόμενο γύρο.

Εφόσον στόχος μας είναι η μεγιστοποίηση της πιθανότητας νίκης του μαθηματικού στο στοίχημα, η αντικειμενική μας συνάρτηση (που πρέπει να μεγιστοποιείται σε κάθε στάδιο) πρέπει να είναι η πιθανότητα ο μαθηματικός να τελειώσει τις τρεις παρτίδες έχοντας τουλάχιστον πέντε μάρκες. Σημειωτέον ότι το στοίχημα κερδίζεται όταν τελειώνοντας τις τρεις παρτίδες έχει ακριβώς ή περισσότερες από πέντε μάρκες.

Ορίζουμε:

$f_n(s_n, x_n)$: την πιθανότητα να τελειώσει ο παίκτης μας τις τρεις παρτίδες έχοντας τουλάχιστον πέντε μάρκες, δεδομένου ότι ξεκινάει στο στάδιο n όντας στην κατάσταση s_n, η άμεση απόφασή του είναι x_n και έπειτα λαμβάνονται βέλτιστες αποφάσεις

$$f_n^*(s_n) = \max_{x_n=0,1,\ldots,s_n} f_n(s_n, x_n)$$

Η έκφραση $f_n(s_n, x_n)$ υπονοεί ότι είναι δυνατό να συγκεντρώσει τελικά πέντε μάρκες, ακόμη κι αν χάσει στην επόμενη παρτίδα.

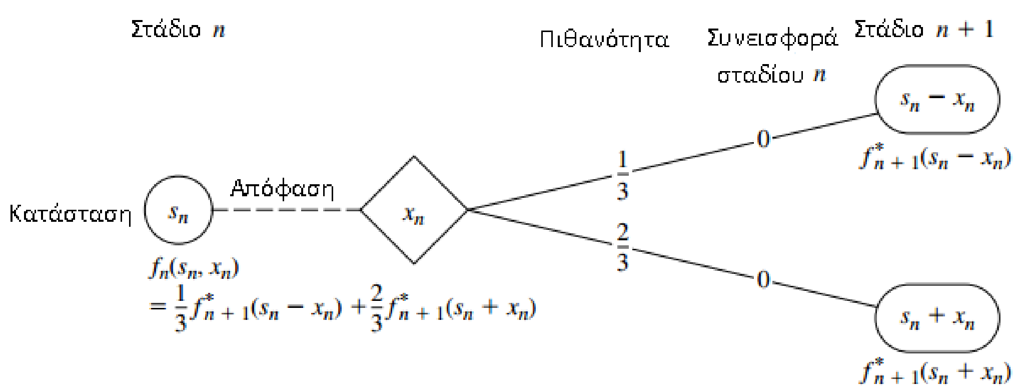
- Αν χάσει, η κατάσταση στο επόμενο στάδιο θα είναι $s_n - x_n$ και η πιθανότητα να τερματίσει το παιχνίδι έχοντας τουλάχιστον πέντε μάρκες είναι $f^*_{n+1}(s_n) = s_n - x_n$.
- Αν, όμως, κερδίσει το επόμενο παιχνίδι, η κατάσταση θα είναι $s_n + x_n$ και η αντίστοιχη πιθανότητα $f^*_{n+1}(s_n) = s_n + x_n$.

Από τα δεδομένα του προβλήματος, η πιθανότητα νίκης σε μία παρτίδα είναι $\frac{2}{3}$, οπότε:

$$f_n(s_n, x_n) = \frac{1}{3} f^*_{n+1}(s_n - x_n) + \frac{2}{3} f^*_{n+1}(s_n + x_n),$$

όπου: $f^*_4(s_4) = \begin{cases} 0, & \text{αν } s_4 < 5 \\ 1, & \text{αν } s_4 \geq 5 \end{cases}$.

Όλα αυτά συνοψίζονται στο παρακάτω διάγραμμα.



Εικόνα 11: Διαγραμματική απεικόνιση του προβλήματος νίκης σε τυχερό παιχνίδι.

Η αναδρομική σχέση γράφεται:

$$f^*_n(s_n) = \max_{x_n = 0,1,\ldots,s_n} \left\{ \frac{1}{3} f^*_{n+1}(s_n - x_n) + \frac{2}{3} f^*_{n+1}(s_n + x_n) \right\}, \text{ για } n = 1, 2, 3.$$

Με τη βοήθεια της οπισθοδρομικής επαγωγής υπολογίζουμε την αναδρομική σχέση για τα τρία στάδια.

***Στάδιο 3* ($n=3$):**

s_3	$f_3^*(s_3)$	x_3^*
0	0	-
1	0	-
2	0	-
3	2/3	2 (ή περισσότερες)
4	2/3	1 (ή περισσότερες)
≥ 5	1	0 (ή $\leq s_3-5$)

***Στάδιο 2* ($n=2$):**

s_2 \ x_2	$f_2(s_2,x_2)=\frac{1}{3}f_3^*(s_2-x_2)+\frac{2}{3}f_3^*(s_2+x_2)$					$f_2^*(s_2)$	x_2^*
	0	1	2	3	4		
0	0	-	-	-	-	0	-
1	0	0	-	-	-	0	-
2	0	4/9	4/9	-	-	4/9	1 ή 2
3	2/3	4/9	2/3	2/3	-	2/3	0, 2 ή 3
4	2/3	8/9	2/3	2/3	2/3	8/9	1
≥ 5	1	-	-	-	-	1	0 (ή $\leq s_2-5$)

***Στάδιο 1* ($n=1$):**

$s_1 \backslash x_1$	$f_1(s_1,x_1)=\frac{1}{3}f_2^*(s_1-x_1)+\frac{2}{3}f_2^*(s_1+x_1)$				$f_1^*(s_1)$	x_1^*
	0	1	2	3		
3	2/3	20/27	2/3	2/3	20/27	1

Επομένως, η βέλτιστη πολιτική που μπορεί να ακολουθήσει ο μαθηματικός είναι:

$$x_1^*=1\begin{cases} \text{αν κερδίσει,} \quad x_2^*=1\begin{cases}\text{αν κερδίσει} \quad , \quad x_3^*=0 \\ \text{αν χάσει} \quad , \quad x_3^*=2 \text{ ή } 3\end{cases} \\ \text{αν χάσει,} \quad x_2^*=1 \text{ ή } 2\begin{cases}\text{αν κερδίσει,} \quad x_3^*=\begin{cases} 2 \text{ ή } 3 \ (\text{για } x_2^*=1) \\ 1,2,3 \text{ ή } 4 \ (\text{για } x_2^*=2)\end{cases} \\ \text{αν χάσει, το στοίχημα χάνεται}\end{cases}\end{cases}$$

Αυτή η πολιτική δίνει στο μαθηματικό πιθανότητα 20/27 να κερδίσει το στοίχημα με τους φίλους του[10].

4 ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε ένα πρόβλημα βελτιστοποίησης, προσπαθούμε να βρούμε την καλύτερη λύση μέσα από ένα σύνολο εναλλακτικών. Προς την κατεύθυνση αυτή, απαιτείται η μαθηματική διατύπωση του προβλήματος και η εξεύρεση της κατάλληλης αναδρομικής σχέσης που θα μας βοηθήσουν να λύσουμε αποδοτικά το πρόβλημα στα διάφορα στάδιά του. Η εργασία αυτή παρουσιάζει την τεχνική του Δυναμικού Προγραμματισμού, μία αδιαμφισβήτητα αποτελεσματική προσέγγιση των προβλημάτων βελτιστοποίησης, ενώ παραθέτει αρκετές εφαρμογές επιχειρησιακού χαρακτήρα που αναπτύχθηκαν με τη συγκεκριμένη μέθοδο. Εφαρμογές που εντάσσονται τόσο στον τομέα της βιομηχανίας λύνοντας προβλήματα διαχείρισης ανθρώπινου δυναμικού, καθώς και εξοπλισμού, όσο και σε προβλήματα της καθημερινότητας και που διακρίνονται σε δύο κατηγορίες. Από τη μία, η κατηγορία του Ντετερμινιστικού Δυναμικού Προγραμματισμού, όπου τα δεδομένα είναι γνωστά με βεβαιότητα και από την άλλη, αυτή του Πιθανοθεωρητικού ή Στοχαστικού Δυναμικού Προγραμματισμού, όπου τα δεδομένα χαρακτηρίζονται από αβεβαιότητα και απαιτείται να συνοδεύονται από μία ορισμένη κατανομή πιθανότητας.

Σε πολλά προβλήματα του πραγματικού κόσμου, η δυσκολία του Δυναμικού Προγραμματισμού είναι ο μεγάλος αριθμός καταστάσεων που πρέπει να συμπεριληφθούν στην επίλυσή τους. Αυτή είναι η “κατάρα των διαστάσεων”, όπως την αποκάλεσε ο Bellman το 1952. Στην παρούσα εργασία, για προφανείς λόγους, ασχοληθήκαμε μόνο με προβλήματα πεπερασμένου αριθμού σταδίων και καταστάσεων. Ένα μοντέλο Δυναμικού Προγραμματισμού πρέπει να είναι όσο πιο απλό γίνεται, ώστε να μπορούμε να το αντιληφθούμε και ταυτόχρονα να το επαληθεύσουμε.

Μέσω των παραδειγμάτων, λοιπόν, γίνεται σαφής η αποδοτικότητα της μεθόδου, καθότι βασίζεται στην τεχνική της μερικής απαρίθμησης, ενώ παράλληλα εντοπίζει τον καλύτερο συνδυασμό αποφάσεων, εγγυώμενη τη βέλτιστη λύση, ακόμη και για τα πλέον απαιτητικά (υπολογιστικά) προβλήματα. Για παράδειγμα, αν είχαμε ένα πρόβλημα με δέκα στάδια, δέκα καταστάσεις και δέκα πιθανές αποφάσεις σε κάθε στάδιο, η μέθοδος της ολικής (εξαντλητικής) απαρίθμησης, η οποία επίσης εγγυάται τον εντοπισμό βέλτιστης λύσης, θα έδινε έως και δέκα δισεκατομμύρια συνδυασμούς, εν αντιθέσει με το Δυναμικό Προγραμματισμό που δε θα χρειαζόταν περισσότερο από χίλιους υπολογισμούς (δέκα για κάθε κατάσταση κάθε σταδίου).

5 ΠΑΡΑΡΤΗΜΑ

5.1 Η ΜΕΘΟΔΟΣ SIMPLEX

Η μέθοδος Simplex είναι μία επαναληπτική διαδικασία προσέγγισης της βέλτιστης λύσης ενός προβλήματος. Εκκινώντας από κάποια αυθαίρετη υπόθεση, προχωρά σε συνεχείς μερικές βελτιστοποιήσεις της, ώστε να τερματίσει όταν δεν είναι δυνατό να βελτιωθεί περαιτέρω, οπότε και την επιστρέφει ως λύση.

Ξεκινώντας από μία τυχαία γωνιακή τιμή (από το γωνιακό σημείο του κυρτού πολυγώνου ή κυρτού πολυέδρου του χώρου των εφικτών λύσεων) της αντικειμενικής συνάρτησης, η μέθοδος Simplex προσπαθεί να βρει επαναληπτικά μία άλλη γωνιακή τιμή που να βελτιώνει την προηγούμενη. Η διερεύνηση γίνεται στην ακμή του πολυγώνου (ή στις κορυφές του πολυέδρου, αν ο αριθμός των μεταβλητών είναι μεγαλύτερος). Καθώς ο αριθμός των κορυφών είναι πεπερασμένος, πάντοτε θα καθίσταται δυνατή η εύρεση της λύσης.

Πιο συνοπτικά, θα μπορούσαμε να πούμε ότι η μέθοδος Simplex βασίζεται στην ακόλουθη ιδιότητα: **"Αν η αντικειμενική συνάρτηση F δε λαμβάνει τη μέγιστη τιμή της στην κορυφή Κ, τότε υπάρχει ένα άκρο που ξεκινάει από το Κ, κατά μήκος του οποίου, η τιμή της συνάρτησης μεγαλώνει."**

Στη θεωρία του Γραμμικού Προγραμματισμού ανακύπτει ένα πολύ βασικό πρόβλημα: η βελτιστοποίηση μιας αντικειμενικής συνάρτησης που εξαρτάται από πολλές μεταβλητές, υποκείμενες σε κάποιους περιορισμούς. Εδώ είναι που υπεισέρχεται η μέθοδος αυτή. Πρέπει, όμως, να τονίσουμε πως η Simplex δουλεύει μόνο την ανισότητα "≤" και ανεξάρτητες μεταβλητές μεγαλύτερες ή ίσες του 0 και έτσι σχηματίζουμε τους περιορισμούς για τον αλγόριθμο. Σε οποιαδήποτε άλλη περίπτωση, δηλαδή εάν εμφανιστεί η αντίθετη ανισότητα "≥" ή περιορισμός ισότητας, τότε δοκιμάζουμε εναλλακτικούς τρόπους επίλυσης.

- ✓ Το βασικό μοντέλο είναι το παρακάτω:

Αντικειμενική συνάρτηση:

$$c_1 \cdot x_1 + c_2 \cdot x_2 + \ldots + c_n \cdot x_n$$

Υπό τους περιορισμούς:

$$a_{11} \cdot x_1 + a_{12} \cdot x_2 + \ldots + a_{1n} \cdot x_n = b_1$$

$$a_{21} \cdot x_1 + a_{22} \cdot x_2 + \ldots + a_{2n} \cdot x_n = b_2$$

…

$$a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \ldots + a_{mn} \cdot x_n = b_m$$

$$x_1, \ldots, x_n \geq 0$$

και ακολουθεί τους εξής κανόνες:

1. Η αντικειμενική συνάρτηση πρέπει να είναι maximize ή minimize.
2. Όλοι οι περιορισμοί πρέπει να είναι ισότητες με μη-αρνητικό δεξιό μέλος.
3. Όλες οι μεταβλητές πρέπει να είναι μη-αρνητικές.

Παραθέτουμε στο σημείο αυτό ένα παράδειγμα, ώστε να γίνει περισσότερο κατανοητή η μέθοδος Simplex.

<u>Παράδειγμα 1</u>: Έστω το πρόβλημα Γραμμικού Προγραμματισμού με 2 μεταβλητές.

Αντικειμενική συνάρτηση:

$$\max z = x_1 + 1{,}5x_2$$

Υπό τους περιορισμούς:

$$x_1 + 4x_2 \le 4$$
$$2x_1 + 3x_2 \le 6$$

Μη-αρνητικότητας περιορισμός:

$$x_1, x_2 \ge 0$$

Μέσω της γραφικής επίλυσης προκύπτει ο χώρος των εφικτών λύσεων του Σχήματος 1 που είναι το γραμμοσκιασμένο κυρτό πολύγωνο. Βάσει της μεθόδου Simplex, για να υπολογίσουμε αλγεβρικά το χώρο των εφικτών λύσεων λαμβάνουμε τους ακόλουθους περιορισμούς:

Περιορισμοί:

$$\begin{aligned} x_1 + 4x_2 + s_1 \quad\quad &= 4 \\ 2x_1 + 3x_2 \quad\quad + s_2 &= 6 \\ x_1, x_2, s_1, s_2 &\ge 0 \end{aligned}$$

όπου: s_1, s_2 οι (μη-αρνητικές) ***χαλαρές μεταβλητές*** (slack variables) που εισάγουμε στο αριστερό μέλος των περιορισμών, ώστε να μετατραπούν οι ανισότητες σε ισότητες.

Όπως βλέπουμε, το πλήθος των εξισώσεων (m=2) είναι μικρότερο από το πλήθος των μεταβλητών (n=4). Συνεπώς, έχουμε σύστημα άπειρων λύσεων και για την αλγεβρική επίλυσή του θέτουμε n-m=4-2=2 πλεονάζουσες μεταβλητές ίσες με το μηδέν. Επιλύουμε τώρα το σύστημα για τις δύο εναπομείνασες μεταβλητές, θέτοντας x_1=0 και x_2=0 και βρίσκουμε ότι s_1=4 και s_2=6. Η πρώτη εφικτή λύση που βρίσκουμε αντιστοιχεί στο σημείο O(0,0) της αρχής των αξόνων στο σχήμα μας.

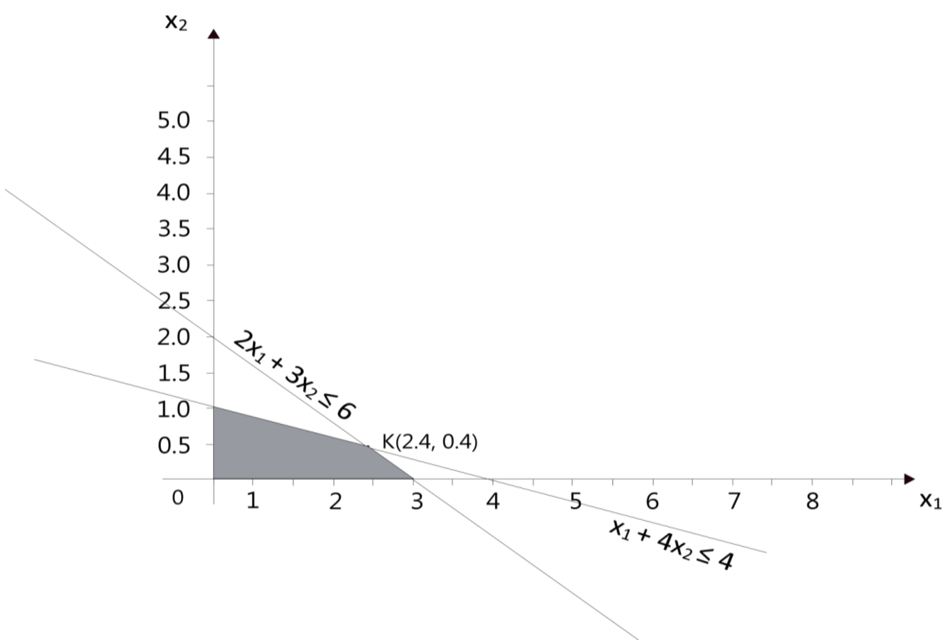
Στη συνέχεια, μηδενίζοντας τις χαλαρές μεταβλητές s_1, s_2 και λύνοντας το σύστημα

$$x_1 + 4x_2 = 4$$

$$2x_1 + 3x_2 = 6$$

βρίσκουμε μια καινούρια εφικτή λύση, τη $(x_1, x_2, s_1, s_2) = (2.4, 0.4, 0, 0)$, η οποία παριστάνεται από το γωνιακό σημείο K(2.4, 0.4) στο σχήμα 3. Το K είναι προφανώς βέλτιστο.

Εξακολουθεί, όμως, να υπάρχει το πρόβλημα εύρεσης των κατάλληλων μεταβλητών που θα μηδενίζονται κάθε φορά (***μη-βασικές*** μεταβλητές), ώστε από τη λύση του συστήματος που έχει προκύψει να δύνανται να υπολογιστούν και οι υπόλοιπες μεταβλητές (***βασικές*** μεταβλητές) και να λαμβάνουμε ένα γωνιακό σημείο (***βασική εφικτή λύση*** του προβλήματος, δηλαδή αυτή που έχει το πολύ όλες τις βασικές μεταβλητές ως προς τη βάση της διάφορες του μηδενός και όλες τις μη-βασικές μηδέν). Για να ξεπεράσουμε αυτό το εμπόδιο, μπορούμε να υπολογίσουμε εξαντλητικά το σύνολα των γωνιακών σημείων, αν επιλύσουμε όλα τα προκύπτοντα συστήματα για όλους τους δυνατούς συνδυασμούς των (n-m) μεταβλητών.



Σχήμα 3: Γραφική επίλυση του προβλήματος

Το συγκεκριμένο πρόβλημα είναι 2 διαστάσεων, άρα σε κάθε αλγεβρική λύση που βρήκαμε αντιστοιχίσαμε και ένα γωνιακό σημείο. Τα σημεία (3,0), (4,0), (0,1) και (0,2) είναι τα γωνιακά σημεία στα οποία αντιστοιχούν οι λύσεις. Ωστόσο, οι δύο από τις τέσσερις κορυφές του τετραπλεύρου, ήτοι τα σημεία (3,0) και (0,1) είναι ***εφικτές*** (δηλαδή ικανοποιούν τους περιορισμούς μη-αρνητικότητας), ενώ τα άλλα δύο σημεία είναι **μη-εφικτές** και άρα δε συγκαταλέγονται στις υποψήφιες ***βέλτιστες εφικτές λύσεις*** (δηλαδή στις εφικτές λύσεις που βελτιστοποιούν την αντικειμενική συνάρτηση)[9]. ■

6 ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Variational Methods In Optimization, Donald R. Smith, Dover Publications, INC (1998)

[2] A history of Analysis, Hans Niels Jahnke (2003)

[3] J. Von Neumann, American Mathematical Society (1998)

[4] Concepts of Optimality and Their Uses, Tjalling C. Koopmans, The American Economic Review (1977)

[5] The Legend of John Von Neumann, P.R. Halmos, The American Mathematical Monthly (1973)

[6] Von Neumann, Morgenstern and the Creation of Game Theory, From Chess to Social Science, 1900-1960, Robert Leonard (2010)

[7] Journal of Political Economy, Gary S. Becker (1993)

[8] On thinking about George Stigler, C.R. McCann, Jr. and Mark Perlman, The Economic Journal (1993)

[9] Εισαγωγή στην Επιχειρησιακή Έρευνα, Κολέτσος Ιωάννης, Στογιάννης Δημήτριος (2012)

[10] Hillier F.S. and Lieberman (2001). Introduction to Operations Research, 7th edition, McGraw-Hill, New York

[11] Bellman R.E. and S.E. Dreyfus (1962), Applied Dynamic Programming, Princeton University Press, Princeton, N.J.

[12] Bertsekas D. (1976), Dynamic Programming and Stochastic Control, Academic Press, New York

[13] Bertsekas D.P.: Dynamic Programming: Deterministic and Stochastic Models, Prentice-Hall, Englewood Cliffs, NJ, 1987

[14] Denardo, E.V.: Dynamic Programming Theory and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1982

[15] Howard R. A.:“Dynamic Programming” Management Science,12:317–345, 1966

[16] Smith, D. K.: Dynamic Programming: A Practical Introduction, Ellis Horwood, London, 1991

[17] Derman C. (1963), “An Optimal Replacement Rules when Changes of State are Markovian ” in Mathematical Optimization Techniques (R.E. Bellman, Ed.), University of California Press, Berkeley, CA

[18] Dreyfus S. and A. Law (1976), The Art of Dynamic Programming, Academic Press, New York

[19] Wagner H.M. and T. Whitin (1957), “Dynamic Problems in the Theory of the Firm” in Theory of Inventory Management, 2nd Ed. (T. Whitin Ed.), Princeton University Press, Princeton, NJ

[20] Winston W.L. (2004), Operations Research: Applications and Algorithms, 4th Ed., Brooks/Cole-Thomson, Belmont, CA

[21] Ross S.M. (1983), Introduction to Stochastic Dynamic Programming, Academic Press, New York

[22] Manne A.S. and A.F. Veinott Jr (1967), Chapter 11, in Investments for Capacity Expansion: Size, Location and Time-Phasing (A.S. Manne, Ed.), MIT Press, Cambridge, MA

7 ΠΕΡΙΕΧΟΜΕΝΑ ΣΧΗΜΑΤΩΝ