ΕΚΔΟΣΕΙΣ ΚΛΕΙΔΑΡΙΘΜΟΣ

Υπεύθυνος σειράς:
ΚΩΝΣΤΑΝΤΙΝΟΣ ΖΟΠΟΥΝΙΔΗΣ

ΣΕΙΡΑ: ΔΙΟΙΚΗΣΗ & ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗ

Μεθευρετικοί και Εξελικτικοί Αλγόριθμοι σε Προβλήματα Διοικητικής Επιστήμης

Ιωάννης Μαρινάκης, Μαγδαληνή Μαρινάκη,
Νικόλαος Φ. Ματσατσίνης και Κωνσταντίνος Ζοπουνίδης

Περιεχόμενα

Κεφάλαιο 1

Βελτιστοποίηση

Ο σκοπός αυτού του κεφαλαίου είναι να εισαγάγει τον αναγνώστη στη θεωρία της βελτιστοποίησης και να δώσει κάποιους ορισμούς που θα τον βοηθήσουν να κατανοήσει καλύτερα τα προβλήματα που επιλύονται στα επόμενα κεφάλαια καθώς και τους λόγους που κάποιες από τις μεθόδους που παρουσιάζονται σε αυτό το βιβλίο είναι κατάλληλες για την επίλυση κάποιου προβλήματος ενώ δεν μπορούν να χρησιμοποιηθούν εύκολα για την επίλυση κάποιου άλλου. Για τον αναγνώστη που ενδιαφέρεται να αναζητήσει περισσότερες λεπτομέρειες μπορεί να ανατρέξει στην ακόλουθη βιβλιογραφία [2, 19, 28, 36, 42, 43, 44, 46, 53, 61, 77, 82, 83, 96, 97, 123, 135, 146, 150, 159, 171, 193, 232, 239, 245, 305, 319, 320, 329, 335, 345, 361, 362, 380, 392, 423, 424, 427].

Η **Βελτιστοποίηση** (**Optimization**) είναι η διαδικασία απόκτησης του βέλτιστου αποτελέσματος κάτω από δεδομένες καταστάσεις. Στο σχεδιασμό, στην εφαρμογή και στη συντήρηση οποιουδήποτε επιχειρηματικού σχεδίου οι αποφασίζοντες θα πρέπει να πάρουν πάρα πολλές τεχνολογικές και διοικητικές αποφάσεις σε πάρα πολλά επίπεδα. Ο στόχος των αποφάσεων είναι είτε η ελαχιστοποίηση του κόστους είτε η μεγιστοποίηση του κέρδους. Το κόστος και το κέρδος σε οποιοδήποτε πρόβλημα μπορεί να εκφραστεί ως μια συνάρτηση καθορισμένων μεταβλητών απόφασης, και έτσι, η βελτιστοποίηση μπορεί να καθορισθεί ως η διαδικασία εύρεσης των συνθηκών που δίνουν τη μέγιστη ή την ελάχιστη τιμή μιας συνάρτησης. Οι μέθοδοι αναζήτησης του βέλτιστου είναι γνωστές και ως τεχνικές μαθηματικού προγραμματισμού και αποτελούν μέρος της επιχειρησιακής έρευνας.

Η **Επιχειρησιακή Έρευνα** (**Operations Research**) είναι ο κλάδος των μαθηματικών που ασχολείται με την εφαρμογή επιστημονικών μεθόδων και τεχνικών σε προβλήματα λήψης αποφάσεων με στόχο την εύρεση και την εφαρμογή της καλύτερης (βέλτιστης) λύσης. Το πεδίο της επιχειρησιακής έρευνας και της

διοικητικής επιστήμης (**management science**) ασχολείται με την ανάπτυξη και την εφαρμογή ποσοτικών τεχνικών για την επίλυση προβλημάτων που αντιμετωπίζουν αποφασίζοντες σε διάφορους τομείς.

Τα βήματα που απαιτούνται για την λήψη μιας απόφασης σε ένα οργανισμό είναι τα ακόλουθα:

1. Αναγνώριση της ανάγκης. Η αντίληψη ότι κάποιες δράσεις πρέπει να γίνουν ή να γίνουν καλύτερα.

2. Καθορισμός του προβλήματος. Στο βήμα αυτό περιλαμβάνονται η ακριβής περιγραφή των στόχων της μελέτης, ο καθορισμός των εναλλακτικών αποφάσεων του συστήματος και η αναγνώριση των ορίων, των περιορισμών και των απαιτήσεων του συστήματος.

3. Κατασκευή του μαθηματικού προτύπου. Τα μαθηματικά πρότυπα, όπως θα δούμε και αναλυτικά στη συνέχεια, καθορίζουν ρητά τη σχέση των δεδομένων εισόδου (μεταβλητές, περιορισμοί και παράμετροι) με τα δεδομένα εξόδου (τιμή του κριτηρίου βελτιστοποίησης που εκφράζεται με την αντικειμενική συνάρτηση).

4. Συλλογή δεδομένων. Τα δεδομένα του προβλήματος απεικονίζουν τις πραγματικές συνθήκες στην επιχείρηση.

5. Επίλυση του μαθηματικού προτύπου. Για την επίλυση του μαθηματικού προτύπου έχει αναπτυχθεί ένας πολύ μεγάλος αριθμός από τεχνικές, ανάλογα με την δομή του προβλήματος.

6. Επεξεργασία της λύσης του μαθηματικού προτύπου. Αφού επιλυθεί κάποιο πρότυπο είναι αναγκαία η ύπαρξη ανάλυσης ευαισθησίας. Αυτή η ανάλυση θα εστιασθεί στην αξιοπιστία τόσο του προτύπου όσο και της λύσης.

7. Υλοποίηση των τελικών αποτελεσμάτων.

Τα βήματα που περιγράφτηκαν προηγουμένως δεν πρέπει να θεωρηθούν ως ένα άκαμπτο σύνολο από βήματα όπου κάποιος εισέρχεται στο ένα άκρο και εξέρχεται από το άλλο άκρο, αντίθετα αναμένεται στην επίλυση οποιουδήποτε προβλήματος ένας πολύ μεγάλος αριθμός από επαναλήψεις διαφόρων βημάτων και από τροποποιήσεις της στρατηγικής σε διάφορα βήματα.

Στη μορφοποίηση ενός προβλήματος βελτιστοποίησης περιλαμβάνονται τα ακόλουθα τρία βασικά σύνολα στοιχείων:

1. **Μεταβλητές απόφασης** (**decision variables**). Οι μεταβλητές απόφασης είναι οι άγνωστοι που πρέπει να καθοριστούν από την επίλυση του προτύπου. Οι παράμετροι αποτελούν τις μεταβλητές ελέγχου του συστήματος.

2. **Περιορισμοί** (**constraints**). Για την μέτρηση των φυσικών περιορισμών του συστήματος, το πρότυπο πρέπει να περιέχει περιορισμούς που να περιορίζουν τις μεταβλητές απόφασης στις εφικτές (επιτρεπτές) τιμές του. Ένα πρόβλημα κατατάσσεται σε διαφορετικές κατηγορίες βάση της φύσης των περιορισμών του (γραμμικοί, ακέραιοι, μη γραμμικοί κ.τ.λ.).

3. **Αντικειμενική συνάρτηση** (**objective function**). Η αντικειμενική συνάρτηση καθορίζει το μέτρο της αποτελεσματικότητας του συστήματος ως μαθηματική συνάρτηση των μεταβλητών απόφασής του. Για παράδειγμα, αν ο αντικειμενικός στόχος του συστήματος είναι η μεγιστοποίηση του συνολικού κέρδους, η αντικειμενική συνάρτηση πρέπει να καθορίζει το κέρδος σε όρους των μεταβλητών απόφασης. Γενικά η **βέλτιστη** (**optimal**) λύση ενός μοντέλου επιτυγχάνεται όταν οι αντίστοιχες τιμές των μεταβλητών απόφασης οδηγούν στην **βέλτιστη** τιμή της αντικειμενικής συνάρτησης ικανοποιώντας ταυτόχρονα όλους τους περιορισμούς.

Τα προβλήματα βελτιστοποίησης μπορούν να ταξινομηθούν λαμβάνοντας υπόψη τα ακόλουθα [131]:

- Τον αριθμό των μεταβλητών που επηρεάζουν την αντικειμενική συνάρτηση. Υπάρχει διαφορετική αντιμετώπιση στην επίλυση του προβλήματος αν υπάρχει μία ή αν υπάρχουν παραπάνω μεταβλητές.

- Τον τύπο των μεταβλητών. Αν οι μεταβλητές είναι συνεχείς τότε λέμε ότι έχουμε ένα **πρόβλημα συνεχούς βελτιστοποίησης** (**continuous optimization problem**). Εάν οι μεταβλητές είναι ακέραιες τότε το πρόβλημα θεωρείται ως ένα **πρόβλημα ακέραιας** ή **διακριτής βελτιστοποίησης** (**integer** ή **discrete optimization problem**). Ένα πρόβλημα που έχει και συνεχείς και ακέραιες μεταβλητές ονομάζεται **πρόβλημα μεικτής ακέραιας βελτιστοποίησης** (**mixed integer optimization problem**). Προβλήματα ακέραιας βελτιστοποίησης όπου η μοντελοποίηση και οι λύσεις τους αντιστοιχούν σε μοντέλα δικτύου και μπορούν να περιγραφούν από ένα γράφημα ονομάζονται **προβλήματα συνδυαστικής βελτιστοποίησης** (**combinatorial optimization problems**).

- Το βαθμό της μη γραμμικότητας της αντικειμενικής συνάρτησης. Τα **γραμμικά προβλήματα βελτιστοποίησης** (**linear programming prob-**

lems) έχουν γραμμική αντικειμενική συνάρτηση και γραμμικούς περιορισμούς. Ένα πρόβλημα **τετραγωνικού προγραμματισμού** (**quadratic optimization problem**) είναι ένα πρόβλημα μη-γραμμικού προγραμματισμού με τετραγωνική αντικειμενική συνάρτηση και γραμμικούς περιορισμούς. Όταν είτε η αντικειμενική συνάρτηση είτε/ή και κάποιος ή και ακόμα όλοι οι περιορισμοί είναι μη γραμμικοί τότε το πρόβλημα ονομάζεται πρόβλημα **μη γραμμικού προγραμματισμού** (**nonlinear programming problems**).

- Τους περιορισμούς που χρησιμοποιούνται. Ένα πρόβλημα μπορεί να μη χρησιμοποιεί περιορισμούς οπότε ονομάζεται **πρόβλημα βελτιστοποίησης χωρίς περιορισμούς** (**unconstrained optimization problem**) ή να χρησιμοποιεί περιορισμούς είτε ισότητας είτε ανισότητας οπότε ονομάζεται **πρόβλημα βελτιστοποίησης με περιορισμούς** (**constrained optimization problem**).

- Αν το πρόβλημα έχει **μία** ή **περισσότερες λύσεις** (**unimodal** ή **multimodal**).

- Αν το πρόβλημα έχει **ένα** ή **περισσότερα κριτήρια (στόχους)** (**single objective** ή **multiobjective optimization problem**).

- Αν κάποια ή και όλες οι μεταβλητές είναι πιθανολογικές (μη - αιτιοκρατικές ή στοχαστικές) τότε το πρόβλημα ονομάζεται **πρόβλημα στοχαστικής βελτιστοποίησης** (**stochastic optimization problem**).

- Αν η αντικειμενική συνάρτηση ή και οι περιορισμοί μεταβάλλονται στο χρόνο τότε το πρόβλημα ονομάζεται **πρόβλημα δυναμικής βελτιστοποίησης** (**dynamic optimization problem**).

Το γενικευμένο **πρόβλημα βελτιστοποίησης (ελαχιστοποίησης) χωρίς περιορισμούς** μοντελοποιείται ως ακολούθως:

$$min f(x) \tag{1.1}$$

υπό

$$x_i, i = 1, \cdots, n \in dom(x_i) \tag{1.2}$$

όπου f είναι η αντικειμενική συνάρτηση του προβλήματος, x είναι οι μεταβλητές και $dom(x)$ είναι το πεδίο τιμών της κάθε μεταβλητής. Το πρόβλημα θα

μπορούσε να είναι μεγιστοποίησης όπου αντί για min θα είχαμε max. Επίσης, αν το πρόβλημα είναι συνεχούς βελτιστοποίησης τότε οι μεταβλητές x παίρνουν τιμές από το σύνολο των πραγματικών αριθμών R ενώ αν το πρόβλημα είναι ακέραιας βελτιστοποίησης τότε οι μεταβλητές x παίρνουν τιμές από το σύνολο των ακεραίων αριθμών Z.

Το γενικευμένο **πρόβλημα βελτιστοποίησης (ελαχιστοποίησης) με περιορισμούς** μοντελοποιείται ως ακολούθως:

$$min f(x) \tag{1.3}$$

υπό

$$h_j(x) = 0, j = 1, \cdots, p \tag{1.4}$$

$$g_k(x) \leq 0, k = 1, \cdots, l \tag{1.5}$$

$$x_i, i = 1, \cdots, n \in dom(x_i) \tag{1.6}$$

όπου με h_j συμβολίζουμε τους p περιορισμούς ισότητας και με g_k συμβολίζουμε τους l περιορισμούς ανισότητας του προβλήματος.

Η **βελτιστοποίηση πολλαπλών αντικειμενικών συναρτήσεων** (multiobjective optimization) αποτελεί ένα πρόβλημα σχεδιασμού ενός μοντέλου το οποίο χαρακτηρίζεται από περισσότερες από μια συναρτήσεις ή κριτήρια (στόχους). Αν οι συναρτήσεις δεν έχουν κάποιο κοινό χαρακτηριστικό τότε το πρόβλημα που προκύπτει είναι η εύρεση εκείνου του μοντέλου που ικανοποιεί όλες τις αντικρουόμενες συναρτήσεις. Σκοπός είναι η επίλυση του προβλήματος βελτιστοποίησης που υπόκειται σε συγκεκριμένες συναρτήσεις και περιορισμούς. Το πρόβλημα αυτό καλείται **πρόβλημα βελτιστοποίησης πολλαπλών κριτηρίων** [474]. Ένα πρόβλημα βελτιστοποίησης πολλαπλών αντικειμενικών συναρτήσεων μπορεί να μοντελοποιηθεί ως ακολούθως [92]:

$$min f(x) = [f_1(x), f_2(x), \cdots, f_m(x)] \tag{1.7}$$

υπό

$$h_j(x) = 0, j = 1, \cdots, p \tag{1.8}$$

$$g_k(x) \leq 0, k = 1, \cdots, l \tag{1.9}$$

όπου $x = [x_1, x_2, \cdots, x_n]$ το διάνυσμα των μεταβλητών απόφασης, $f_i, i = 1, \cdots, m$ οι αντικειμενικές συναρτήσεις του προβλήματος, και $g_k, k = 1, \cdots, l$ και $h_j, j = 1, \cdots, p$ είναι οι περιορισμοί του προβλήματος.

Η **Δυναμική Βελτιστοποίηση** αποτελείται από προβλήματα βελτιστοποίησης που περιλαμβάνουν αλλαγές μέσα στο χρόνο. Οι αλλαγές μπορεί να εμφανίζονται στην αντικειμενική συνάρτηση ή/και στους περιορισμούς. Όταν εμφανιστεί η αλλαγή οι λύσεις που έχουν βρεθεί μπορεί να μην είναι πλέον εφικτές οπότε πρέπει να βρεθεί μία διαδικασία που να οδηγεί ξανά σε βέλτιστες λύσεις.

Η μοντελοποίηση ενός γενικευμένου προβλήματος δυναμικής βελτιστοποίησης είναι η ακόλουθη:

$$max f(X, t) \tag{1.10}$$

$$\text{υπό} \quad h_j(X, t) = 0, j = 1, \cdots, p \tag{1.11}$$

$$g_k(X, t) \leq 0, k = 1, \cdots, l \tag{1.12}$$

$$X = [x_1, x_2, \cdots, x_n]^T \tag{1.13}$$

όπου f είναι η αντικειμενική συνάρτηση του προβλήματος, X το διάνυσμα των μεταβλητών απόφασης, t το χρονικό διάστημα που μας ενδιαφέρει να εξετάσουμε το πρόβλημα, h_j οι περιορισμοί ισότητας και g_k οι περιορισμοί ανισότητας του προβλήματος.

Στη **συνδυαστική βελτιστοποίηση** έχει αναπτυχθεί ένας πολύ μεγάλος αριθμός αλγορίθμων για την επίλυση των διαφόρων προβλημάτων. Τα πιο σημαντικά προβλήματα είναι:

- τα προβλήματα **εύρεσης της συντομότερης διαδρομής**. Είναι από τα βασικότερα προβλήματα ροών σε δίκτυα. Το πρόβλημα της συντομότερης διαδρομής εμφανίζεται άμεσα ή έμμεσα σε ένα μεγάλο αριθμό από εφαρμογές της πραγματικής ζωής, όπως στα προβλήματα δρομολόγησης οχημάτων, σε προβλήματα επενδύσεων, σε προβλήματα ελέγχου αποθεμάτων κτλ. Οι βασικότεροι αλγόριθμοι επίλυσης προβλημάτων ροής ελαχίστου κόστους για την εύρεση της συντομότερης διαδρομής είναι ο αλγόριθμος Dijkstra και ο αλγόριθμος Ford.

- Το **πρόβλημα της μέγιστης ροής**. Το πρόβλημα της μέγιστης ροής και το πρόβλημα της εύρεσης της συντομότερης διαδρομής είναι συμπληρωματικά μεταξύ τους. Αυτό συμβαίνει γιατί και τα δύο μπορούν να θεωρηθούν ως υποπροβλήματα σε αλγόριθμους για την επίλυση του προβλήματος ροής ελαχίστου κόστους. Τα δύο αυτά προβλήματα όμως διαφέρουν μεταξύ τους γιατί ενώ τα προβλήματα εύρεσης της συντομότερης διαδρομής εξε-

τάζουν - μορφοποιούν τα κόστη των τόξων αλλά όχι τις χωρητικότητες των τόξων, τα προβλήματα μέγιστης ροής εξετάζουν - μορφοποιούν χωρητικότητες των τόξων άλλα όχι τα κόστη. Οι βασικότεροι αλγόριθμοι για την επίλυση του προβλήματος μέγιστης ροής είναι ο αλγόριθμος Ford-Fulkerson και ο αλγόριθμος Dinic.

- Το πρόβλημα του **ελάχιστου τανύοντος δένδρου** (**M**inimal **S**panning **T**ree - MST) είναι αυτό στο οποίο αναζητείται ένα τανύον δένδρο του $\mathcal{G}$ με τον ελάχιστο ολικό σταθμό c_T. Το πρόβλημα της εύρεσης ενός ελάχιστου τανύοντος δένδρου εμφανίζεται στις εφαρμογές σχεδιασμού δικτύων (για παράδειγμα επικοινωνίας υπολογιστών, μεταφοράς υγρών καυσίμων, κλπ.) ή αυτούσιο ή ως υποπρόβλημα. Οι βασικότεροι αλγόριθμοι για την επίλυση του προβλήματος ελαχίστου τανύοντος δένδρου είναι ο αλγόριθμος Prim και ο αλγόριθμος Kruskall.

Ένας αλγόριθμος που μπορεί να χρησιμοποιηθεί στη συνδυαστική βελτιστοποίηση αλλά και γενικότερα σε προβλήματα βελτιστοποίησης είναι η μέθοδος του δυναμικού προγραμματισμού. Ο **δυναμικός προγραμματισμός** είναι μια μέθοδος για την επίλυση προβλημάτων βελτιστοποίησης βασιζόμενος σε μια μη-ρητή (αλλά πλήρη) απαρίθμηση όλων των πιθανών λύσεων. Η ιδέα του δυναμικού προγραμματισμού είναι πολύ απλή και στηρίζεται στην αρχή του βέλτιστου που διατυπώθηκε από τον Bellman το 1952. Αρχικά παρουσιάστηκε για τη βελτιστοποίηση της ακολουθητικής διαδικασίας αποφάσεων. Τα προβλήματα που μπορούν να επιλυθούν με τη χρήση του δυναμικού προγραμματισμού είναι προβλήματα που η λύση τους μπορεί να παρουσιαστεί ως το αποτέλεσμα μιας ακολουθίας αποφάσεων. Για πολλά προβλήματα αυτής της μορφής, μια βέλτιστη ακολουθία αποφάσεων μπορεί να βρεθεί παίρνοντας μία απόφαση τη φορά με βέλτιστο κόστος.

Ο **Δυναμικός Προγραμματισμός** [123] συνήθως μειώνει δραστικά το μέγεθος των δυνατών λύσεων προσπαθώντας να απορρίψει την απαρίθμηση κάποιων λύσεων που δεν μπορούν με κανένα τρόπο να οδηγήσουν στο βέλτιστο. Στον δυναμικό προγραμματισμό μια βέλτιστη ακολουθία αποφάσεων προέρχεται κάνοντας χρήση της **Αρχής του Βέλτιστου** (Principle of Optimality) [123]. Αυτή η αρχή αναφέρει ότι *μια βέλτιστη ακολουθία αποφάσεων έχει την ιδιότητα ότι οποιαδήποτε και αν είναι η αρχική κατάσταση και οι αρχικές αποφάσεις, οι εναπομείναντες αποφάσεις πρέπει να αποτελούν μια βέλτιστη ακολουθία αποφάσεων σε σχέση με την κατάσταση που έχει έρθει το πρόβλημα βάσει της αρχικής απόφασης.*

Ένας πολύ σημαντικός αλγόριθμος που οδηγεί στην εύρεση του ολικού βέλτιστου σε προβλήματα συνδυαστικής βελτιστοποίησης είναι ο **αλγόριθμος δια-**

κλάδωσης και οριοθέτησης (branch and bound algorithm) [335] όπου με τη χρήση ελεγχόμενης απαρίθμησης και χαλάρωσης ή / και δυϊκότητας, προβλήματα ακεραίου προγραμματισμού λύνονται με τη υποδιαίρεση ενός εφικτού συνόλου S σε ένα σύνολο από υποσύνολα $\{S^i : i = 1, \cdots, k\}$ και έπειτα με την επίλυση του προβλήματος για κάθε ένα από τα υποσύνολα αυτά. Λέμε ότι το $\{S^i : i = 1, \cdots, k\}$ είναι μια υποδιαίρεση του S αν $\cup_{i=1}^{k} S^i = S$. Μια υποδιαίρεση ονομάζεται τμηματοποίηση (partition) αν $S^i \cap S^j = \emptyset$ για $i, j = 1, \cdots, k, i \neq j$. Η μέθοδος στηρίζεται στη γνώριμη ιδέα του διαίρει και βασίλευε. Με άλλα λόγια αν είναι τόσο δύσκολο να βελτιστοποιήσεις στο σύνολο S, ίσως το πρόβλημα να μπορεί να λυθεί κάνοντας βελτιστοποίηση σε μικρότερα σύνολα και στη συνέχεια βάζοντας τα αποτελέσματα μαζί.

Μέρος ΙΙ

Απλοί Ευρετικοί Αλγόριθμοι και Αλγόριθμοι Τοπικής Αναζήτησης

Κεφάλαιο 2

Απλοί Ευρετικοί Αλγόριθμοι και Αλγόριθμοι Τοπικής Αναζήτησης

2.1 Απλοί Ευρετικοί Αλγόριθμοι

Σε αυτό το κεφάλαιο παρουσιάζονται οι ευρετικοί αλγόριθμοι. Η επίλυση ενός προβλήματος βελτιστοποίησης και ιδιαίτερα συνδυαστικής βελτιστοποίησης γίνεται ολοένα και δυσκολότερη όσο αυξάνει το μέγεθος του προβλήματος και πολλές φορές το να προσπαθούμε να βρούμε την ολικά βέλτιστη λύση σε λογικό χρόνο είναι πρακτικά αδύνατο. Για να επιλυθούν προβλήματα αυτής της μορφής συχνά καταφεύγουμε σε διαφορετικές τεχνικές που μας οδηγούν σε μια σχεδόν βέλτιστη, αλλά ικανοποιητική λύση. Μια λύση ενός ευρετικού αλγορίθμου γίνεται αποδεκτή αν ικανοποιεί κάποια κριτήρια όπως η ποιότητα της λύσης, δηλαδή η απόκλισή της από τη βέλτιστη, η ευκολία απόκτησης μιας λύσης, η λογική πάνω στην οποία στηρίζονται οι κανόνες του ευρετικού αλγορίθμου που χρησιμοποιήθηκαν για να οδηγηθούμε στη λύση. Για κάθε πρόβλημα βελτιστοποίησης δεν υπάρχει μονάχα ένας ευρετικός αλγόριθμος που να δίνει τη βέλτιστη λύση, αλλά έχουν αναπτυχθεί αρκετοί αλγόριθμοι οι οποίοι συγκρινόμενοι μεταξύ τους, οδηγούν ολοένα και σε καλύτερες λύσεις.

Ένα σημείο που πρέπει να διευκρινιστεί αφορά την ποιότητα της λύσης. Σε μερικά προβλήματα, όπως αναφέρθηκε, είναι αδύνατο να βρεθεί η βέλτιστη λύση για κάποιο πρόβλημα σε ικανοποιητικό χρόνο, και έτσι γεννιέται το ερώτημα: Πώς είναι δυνατό να είμαστε βέβαιοι για την ποιότητα της λύσης που προέκυψε από τον αλγόριθμο που αναπτύξαμε; Η απάντηση δεν είναι εύκολο να δοθεί. Ένας απλός αλλά και ο πιο συνηθισμένος τρόπος απόδειξης είναι να δημιουργήσουμε μικρότερα παραδείγματα από αυτό που θέλουμε να λύσουμε, τα οποία μπορούμε να τα λύσουμε με κάποια ακριβή μέθοδο, και να δούμε πό-

σο κοντά στο βέλτιστο είναι η λύση που παίρνουμε με τη χρήση του ευρετικού αλγορίθμου. Ένας άλλος τρόπος είναι η δημιουργία ενός φράγματος αποδεκτής λύσης με την επίλυση ενός χαλαρωμένου προβλήματος, για παράδειγμα με τη διαδικασία διακλάδωσης και οριοθέτησης (branch and bound), και έτσι όλες οι λύσεις που θα πάρουμε με τη χρήση του ευρετικού αλγορίθμου και που η τιμή τους δεν θα παραβιάζει την τιμή του φράγματος θα είναι ικανοποιητικές.

Σε αυτό το κεφάλαιο παρουσιάζονται στη συνέχεια διάφορες κατηγορίες ευρετικών αλγορίθμων, η κάθε μία από τις οποίες έχει κάποια ιδιαίτερα χαρακτηριστικά που θα περιγραφούν στη συνέχεια. Οι κατηγορίες των αλγορίθμων αυτών είναι οι ακόλουθες:

- Αλγόριθμοι απληστίας (greedy algorithms). Οι αλγόριθμοι απληστίας προσπαθούν να οδηγήσουν σε μια εφικτή λύση του προβλήματος, αλλά πολλές φορές χρειάζονται πάρα πολύ μεγάλο χρόνο γιατί είναι μυωπικοί αλγόριθμοι, δηλαδή βλέπουν μόνο μπροστά.

- Προσεγγιστικοί αλγόριθμοι (approximation algorithms). Οι προσεγγιστικοί αλγόριθμοι προσπαθούν να λύσουν αυτό το πρόβλημα χρησιμοποιώντας επιπλέον πληροφορία.

- Αλγόριθμοι τοπικής αναζήτησης (local search algorithms). Οι αλγόριθμοι τοπικής αναζήτησης προσπαθούν ξεκινώντας από μια αρχική εφικτή λύση να βελτιώσουν τη λύση με κάποια μέθοδο αναζήτησης στη γειτονιά της λύσης.

Η βασική διαφορά των τριών κατηγοριών είναι ότι οι δύο πρώτες κατηγορίες χρησιμοποιούνται για να παράγουμε μία αρχική λύση ενώ η τρίτη, και για αυτό το λόγο παρουσιάζεται σε χωριστή παράγραφο, προσπαθεί να βελτιώσει μία ήδη υπάρχουσα λύση. Το παράδειγμα που ακολουθεί [188] παρακίνησε τη μελέτη των προσεγγιστικών αλγορίθμων και είναι το πρώτο που αντιμετωπίστηκε με το μεθοδολογικό πλαίσιο των προσεγγιστικών αλγορίθμων.

Φανταστείτε τον εαυτό σας ότι βρίσκεστε στην εργασία σας στις 9 το πρωί, έχετε οχτώ μηχανήματα τα οποία δεν μπορούν να είναι έτοιμα για δουλειά πριν της 10 το πρωί και 147 διαφορετικές δουλειές που πρέπει να γίνουν από τα μηχανήματα, κάθε μία από αυτές με διαφορετική χρονική διάρκεια. Επίσης, στις 9 το βράδυ έχει στην τηλεόραση ένα αγώνα μπάσκετ που δεν θέλετε με τίποτα να χάσετε, αλλά που πρέπει για να τον δείτε να έχουν τα μηχανήματα ολοκληρώσει όλες τις δουλειές. Ο στόχος σας είναι να εκχωρήσετε δουλειές στα μηχανήματα με τέτοιο τρόπο ώστε να είστε στο σπίτι όσο νωρίτερα γίνεται.

Το πρόβλημα αυτό ανήκει στην κατηγορία των **μη πολυωνυμικά δύσκολων (NP-Hard)** προβλημάτων [345], που σημαίνει ότι όχι μόνο δεν είναι γνωστός κάποιος εφικτός αλγόριθμος για την επίλυση του προβλήματος αλλά είναι απίθανο να υπάρχει ένας τέτοιος αλγόριθμος. Για πολλά χρόνια προβλήματα αυτού του τύπου επιλύονταν είτε με εργαλεία από τον ακέραιο προγραμματισμό είτε με τη βοήθεια κάποιων πρωταρχικών ευρετικών αλγορίθμων (ιδιαίτερα αναφέρονται πρωταρχικοί αλγόριθμοι απληστίας). Αν χρησιμοποιήσουμε ακέραιο προγραμματισμό, πρέπει να βρούμε πάνω και κάτω όρια για να περιορίσουμε το πρόβλημα με την ελπίδα να πέσουμε σε μια βέλτιστη λύση. Ο χρόνος που χρειάζεται για την επίλυση ακόμα και ενός μικρού μεγέθους προβλήματος με αυτές τις μεθόδους είναι υπερβολικός. Αντίθετα, ο χρήστης διακόπτει τη διαδικασία απαρίθμησης είτε όταν η τρέχουσα λύση κρίνεται ικανοποιητική, είτε όταν ο χρόνος επίλυσης έχει περάσει κάποιο λογικό όριο. Όμως οι αλγόριθμοι ακεραίου προγραμματισμού δεν παρέχουν καμία εγγύηση. Κανείς δεν μπορεί να εγγυηθεί ότι μετά από πέντε λεπτά που θα τρέχει ακόμα ο αλγόριθμος θα μπορούμε να πάρουμε μια σημαντικά καλύτερη λύση, ή μετά από πέντε ακόμα μέρες που θα τρέχει ο αλγόριθμος θα μπορούμε να οδηγηθούμε σε μια μικρή έστω βελτίωση της λύσης. Εκτός από την εγγύηση της ποιότητας της λύσης και τον λογικό χρόνο επίλυσης, το κλασικό εργαλείο ακεραίου προγραμματισμού, ο αλγόριθμος διακλάδωσης και οριοθέτησης (Branch and Bound), χρειάζεται πολύ καλά φράγματα τα οποία είναι εφικτές λύσεις, όπως επίσης και πολύ καλές εκτιμήσεις των τιμών του βέλτιστου (κάτω φράγματα για προβλήματα ελαχιστοποίησης - πάνω φράγματα για προβλήματα μεγιστοποίησης). Οι προσεγγιστικοί αλγόριθμοι καλύπτουν τόσο την εγγύηση για την εύρεση λύσης όσο και ότι θα έχουμε στη διάθεση μας μια καλή εφικτή λύση.

Επιστρέφουμε στο πρόβλημα μας, η ώρα είναι λίγα λεπτά πριν από τις 10 και είναι πολύ αργά για να αρχίσουμε να δοκιμάζουμε μια διαδικασία διακλάδωσης και οριοθέτησης, αφού μπορεί να φθάσουμε να συγκρίνουμε τη λύση με μια λύση που μπορούμε να μαντέψουμε τυχαία. Αντίθετα, δεν θα ήταν καλύτερο να χρησιμοποιήσουμε κάποια μέθοδο (για παράδειγμα ανάθεση εργασιών στα μηχανήματα με βάση τη φορά των δεικτών του ρολογιού) για να βρούμε μια λογική λύση; Έστω, εκχωρούμε μια δουλειά σε ένα μηχάνημα μόλις αυτό γίνει διαθέσιμο. Αυτό ακούγεται λογικό, αλλά με ένα γρήγορο υπολογισμό ανακαλύπτουμε ότι η τελευταία δουλειά θα τερματιστεί στις 10 το βράδυ. Ο αλγόριθμος που περιγράψαμε για να δώσουμε λύση στο πρόβλημα είναι ένας κλασικός ευρετικός αλγόριθμος.

Οι ευρετικοί αλγόριθμοι δουλεύουν γρήγορα και αποτελεσματικά. Η ποιότητα της λύσης όμως που παρέχουν είναι ένα εντελώς διαφορετικό θέμα. Πριν από την εμφάνιση της μεθόδου ανάλυσης των προσεγγιστικών αλγορίθμων η

απόδοση ενός ευρετικού αλγορίθμου κρινόταν από δοκιμαστικά τρεξίματα σε ένα σύνολο από παραδείγματα προβλημάτων αναφοράς και συγκρινόταν με την απόδοση άλλων αλγορίθμων που επέλυαν το ίδιο πρόβλημα. Αυτή η μέθοδος είχε πάντοτε κάποια προβλήματα. Αρχικά, το σύνολο των προβλημάτων που δοκιμαζόταν ήταν ένα μικρό δείγμα αλλά συνήθως όχι αντιπροσωπευτικό όλων των παραδειγμάτων ενός συγκεκριμένου προβλήματος. Επίσης, μια καλή βελτίωση της λύσης σε κάποιο παράδειγμα σε σύγκριση με κάποιον άλλο αλγόριθμο δεν σημαίνει και γενικά βελτίωση της λύσης σε οποιοδήποτε παράδειγμα. Είναι ξεκάθαρο ότι χρειάζεται μια ανάλυση για το πόσο καλή είναι η λύση που μπορεί να μας δώσει ένας αλγόριθμος απληστίας και γενικότερα ένας ευρετικός αλγόριθμος.

Η δημιουργία μίας αρχικής λύσης είναι μία πολύ σημαντική διαδικασία για την αποτελεσματικότητα του αλγορίθμου. Αν η αρχική λύση είναι καλή μπορεί να οδηγήσει πιο γρήγορα τον αλγόριθμο σε σύγκλιση. Από την άλλη μεριά μία πολύ κακή αρχική λύση θεωρητικά θα αργήσει να συγκλίνει σε κάποια καλή λύση. Συνήθως, υπάρχουν δύο τρόποι για να δημιουργηθεί μία αρχική λύση. Ο πρώτος τρόπος είναι με τη χρήση μίας τυχαιοποιημένης διαδικασίας κατάλληλα προσαρμοσμένης στο πρόβλημα που επιλύουμε και ο δεύτερος τρόπος με τη χρήση κάποιου αλγόριθμου απληστίας. Από τη μέθοδο που θα χρησιμοποιήσουμε στη φάση της τοπικής αναζήτησης εξαρτάται και ο τρόπος που θα επιλέξουμε για να δημιουργήσουμε τις αρχικές λύσεις. Έτσι, για παράδειγμα, αν έχουμε ένα συνεχές πρόβλημα είναι καλύτερο να δημιουργήσουμε τις αρχικές λύσεις με τυχαίο τρόπο. Αντίθετα, σε ένα πρόβλημα συνδυαστικής βελτιστοποίησης είναι πιο αποτελεσματικό να δημιουργήσουμε τις αρχικές λύσεις με ένα αλγόριθμο απληστίας. Επίσης, αν έχουμε ένα αλγόριθμο που χρησιμοποιεί ένα πληθυσμό από λύσεις για να βελτιώσει τις λύσεις του κατά τη διάρκεια των επαναλήψεων, τότε είτε θα δημιουργήσουμε τις αρχικές λύσεις με τυχαίο τρόπο είτε θα πρέπει να χρησιμοποιήσουμε πολλούς διαφορετικούς αλγορίθμους απληστίας ή συνδυασμό των δύο μεθόδων (για παράδειγμα τη διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (παράγραφος 3.6)). Το βασικό στοιχείο σε αυτή τη κατηγορία αλγορίθμων είναι ότι εξαρτώνται από το πρόβλημα που επιλύουν. Έτσι διαφορετικοί αλγόριθμοι απληστίας χρησιμοποιούνται στο πρόβλημα του πλανόδιου πωλητή (κεφάλαιο 13) και διαφορετικοί σε κάποιο άλλο πρόβλημα, για παράδειγμα στο πρόβλημα του σακιδίου.

Ένας προσεγγιστικός αλγόριθμος είναι πάντοτε πολυωνυμικός και εκτιμάται από τη χειρότερη περίπτωση του πιθανώς σχετικού λάθους σε όλα τα πιθανά παραδείγματα του προβλήματος. Ένας αλγόριθμος Α μπορεί να θεωρηθεί ως ένας δ - προσεγγιστικός αλγόριθμος για ένα πρόβλημα ελαχιστοποίησης P αν για κάθε παράδειγμα I του P δίνει μια λύση η οποία είναι το πολύ δ φορές

μεγαλύτερη από τη βέλτιστη. Φυσικά, $\delta > 1$ και όσο πιο κοντά είναι στη μονάδα τόσο καλύτερα. Ομοίως, για προβλήματα μεγιστοποίησης ένας δ - προσεγγιστικός αλγόριθμος δίνει για κάθε παράδειγμα I μια λύση που είναι τουλάχιστον δ φορές μεγαλύτερη από τη μέγιστη. Σε αυτή την περίπτωση $\delta < 1$. Το δ αναφέρεται συνήθως ως ρυθμός προσέγγισης, ή ως εγγύηση απόδοσης, ή ως ρυθμός μελέτης χειρότερης περίπτωσης. Ο αλγόριθμος που περιγράφτηκε προηγούμενα για τη λύση του προβλήματος μας δίνει μια λύση με σχετικό λάθος όχι παραπάνω από 100%. Πράγμα που σημαίνει ότι είναι ένας 2 - προσεγγιστικός αλγόριθμος. Έχουν βρεθεί και ευρετικοί αλγόριθμοι που δίνουν καλύτερα αποτελέσματα από ότι αυτός για το συγκεκριμένο πρόβλημα, για παράδειγμα με $\frac{4}{3}$ ρυθμό λάθους. Έτσι, με τον τελευταίο αλγόριθμο σε 10 ώρες θα έχουν τελειώσει όλες οι μηχανές, δηλαδή στις 8 το βράδυ.

Στη συνέχεια θα δούμε επιπλέον στοιχεία για τους προσεγγιστικούς αλγόριθμους. Ακόμα υποθέτουμε ότι ένας πολυωνυμικός αλγόριθμος δίνει μια εφικτή λύση σε κάποιο NP-Hard πρόβλημα που έχει ένα σύνολο από παραδείγματα I.

Ένας πολυωνυμικός αλγόριθμος A, λέγεται ότι είναι ένας δ - προσεγγιστικός αλγόριθμος εάν για κάθε παράδειγμα προβλήματος I με μια βέλτιστη λύση $OPT(I)$ ισχύει ότι

$$OPT(I) \leq \delta. \tag{2.1}$$

Όπως αναφέρθηκε προηγουμένως το $\delta \geq 1$ ισχύει για προβλήματα ελαχιστοποίησης και το $\delta \leq 1$ για προβλήματα μεγιστοποίησης. Η μικρότερη τιμή που μπορεί να πάρει το δ είναι ο ρυθμός προσέγγισης (απόδοσης) R_A του αλγορίθμου A.

Η τιμή του δ αναφέρεται συνήθως με τους ακόλουθους όρους:

- Όριο χειρότερης περίπτωσης (Worst case bound).
- Απόδοση χειρότερης περίπτωσης (Worst case performance).
- Παράγοντας προσέγγισης (Approximation factor).
- Ρυθμός προσέγγισης (Approximation ratio).
- Ρυθμός λάθους (Error ratio).

Ο απόλυτος ρυθμός απόδοσης, R_A, ενός προσεγγιστικού αλγορίθμου A είναι:

$$R_A = inf\{r \geq 1 | R_A(I) \leq r \text{ για όλα τα παραδείγματα } I\} \tag{2.2}$$

ενώ ο ασυμπτωτικός ρυθμός απόδοσης, R_A^∞ για το Α είναι:

$$R_A^\infty = inf\{r \geq 1 | \exists n \in Z^+, R_A(I) \leq r \forall I s.t. I \geq n\} \quad (2.3)$$

Η αποτελεσματικότητα ενός προσεγγιστικού αλγορίθμου είναι ένα άλλο πολύ σημαντικό σημείο. Αφού υποθέτουμε ότι κάθε προσεγγιστικός αλγόριθμος είναι πολυωνυμικός, υπάρχει μια τεράστια ποικιλία πολυωνυμικών αλγορίθμων που κάποιοι από αυτούς είναι αναποτελεσματικοί και χωρίς καμία πρακτική σημασία. Ένα άλλο σημείο που μας ενδιαφέρει είναι ο χρόνος που απαιτείται από ένα πολυωνυμικό αλγόριθμο έτσι ώστε να προσεγγίσουμε μια καλύτερη λύση.

Μια οικογένεια από προσεγγιστικούς αλγορίθμους για ένα πρόβλημα $P, \{A_\epsilon\}_\epsilon$, ονομάζεται πολυωνυμικό προσεγγιστικό διάγραμμα, εάν ο αλγόριθμος A_ϵ είναι $(1+\epsilon)-$προσεγγιστικός αλγόριθμος και ο χρόνος εκτέλεσης του είναι πολυωνυμικός στο μέγεθος των δεδομένων εισόδου για ένα καθορισμένο ϵ.

Μια οικογένεια από προσεγγιστικούς αλγορίθμους για ένα πρόβλημα $P, \{A_\epsilon\}_\epsilon$, ονομάζεται πλήρως πολυωνυμικό προσεγγιστικό διάγραμμα, εάν ο αλγόριθμος A_ϵ είναι $(1+\epsilon)-$προσεγγιστικός αλγόριθμος και ο χρόνος εκτέλεσης του είναι πολυωνυμικός στο μέγεθος των δεδομένων εισόδου και του $\frac{1}{\epsilon}$.

2.2 Αλγόριθμοι Τοπικής Αναζήτησης

Η τοπική αναζήτηση βασίζεται [345] στην αρχαιότερη μέθοδο βελτιστοποίησης, στη μέθοδο δοκιμής και σφάλματος. Η τοπική αναζήτηση έχει αποδειχθεί πολύ επιτυχημένη στην πράξη σε ένα πολύ μεγάλο αριθμό από προβλήματα συνδυαστικής βελτιστοποίησης. Στη συνέχεια περιγράφεται ένας γενικός αλγόριθμος.

Δοθέντος ενός προβλήματος βελτιστοποίησης (F, c), όπου F είναι ένα εφικτό σύνολο και c είναι το κόστος, επιλέγουμε τη γειτονιά (η επιλογή της γειτονιάς αναζήτησης εξαρτάται από το πρόβλημα που επιλύεται και από τον αλγόριθμο τοπικής αναζήτησης που χρησιμοποιείται για να επιλυθεί το πρόβλημα αυτό)

$$N : F \longrightarrow 2^F \quad (2.4)$$

στην οποία εφαρμόζεται αναζήτηση για να βρεθεί μια καλύτερη λύση από την αρχική αν υπάρχει κάποιο σημείο $s \in N$ που να έχει μικρότερο κόστος από το κόστος που έχουμε αυτή τη στιγμή. Για όσο χρονικό διάστημα μία καλύτερη

λύση υπάρχει, αντικαθιστούμε την υπάρχουσα και συνεχίζουμε την αναζήτηση μέχρι το σημείο που θα βρούμε κάποιο τοπικό ελάχιστο και η λύση που έχουμε δεν θα βελτιώνεται παραπάνω. Σε ένα γενικό αλγόριθμο τοπικής αναζήτησης αρχίζουμε από μια αρχική εφικτή λύση $t \in F$ και χρησιμοποιούμε μία υπορουτίνα για να ψάξουμε για μια καλύτερη λύση στη γειτονιά της αρχικής λύσης. Όσο βρίσκεται μια βελτιωμένη λύση, την εφαρμόζουμε και επαναλαμβάνουμε τη διαδικασία αναζήτησης από τη νέα λύση. Όταν φθάσουμε σε τοπικό ελάχιστο σταματάμε.

διαδικασία $local\_search$

begin

s μια αρχική λύση του προβλήματος

do while βρίσκεται μια βελτιωμένη λύση ($improve(s)$)

$s = improve(s)$

return s

end

Ένα πολύ σημαντικό χαρακτηριστικό της τοπικής αναζήτησης είναι το γεγονός ότι μπορεί να εκτελείται από πολλά διαφορετικά αρχικά σημεία και να επιλέγεται το καλύτερο ως βέλτιστη λύση. Σε αυτές τις περιπτώσεις μια άλλη απόφαση που πρέπει να παρθεί είναι πόσα θα είναι τα αρχικά σημεία που θα πρέπει να επιλέξουμε. Το επόμενο πρόβλημα που πρέπει να επιλυθεί για να είμαστε σε θέση να πούμε ότι η τοπική αναζήτηση που εφαρμόζουμε οδηγεί σε ικανοποιητικά αποτελέσματα είναι το ότι θα πρέπει να γίνει σωστή επιλογή της γειτονιάς που θα πραγματοποιηθεί η αναζήτηση.

Μία λύση s' στη γειτονιά της αρχικής λύσης s ($s \in N(s)$) ονομάζεται γείτονας της s. Ένας γείτονας δημιουργείται με την εφαρμογή ενός τελεστή κίνησης m που πραγματοποιεί μία μικρή διαταραχή στη λύση s. Η βασική αρχή που πρέπει να χαρακτηρίζει μία γειτονιά είναι η τοπικότητα. Τοπικότητα είναι η επίδραση στη λύση όταν πραγματοποιείται μία κίνηση (διαταραχή) στην αναπαράσταση της λύσης. Όταν γίνονται μικρές αλλαγές στην αναπαράσταση τότε στη λύση πρέπει να αντανακλώνται αυτές οι αλλαγές. Σε αυτή την περίπτωση ορίζουμε ότι η γειτονιά έχει ισχυρή τοπικότητα. Ο όρος ασθενής τοπικότητα χαρακτηρίζεται από μεγάλες αλλαγές στη λύση. Η δομή της γειτονιάς εξαρτάται από το πρόβλημα που θέλουμε να επιλύσουμε. Έτσι, για παράδειγμα αν έχουμε ένα συνεχές πρόβλημα τότε η γειτονιά είναι ένας κύκλος γύρω από το s με ακτίνα ϵ. Αν από την άλλη μεριά έχουμε ένα διακριτό πρόβλημα, η

γειτονιά είναι ένα σύνολο από σημεία, όπου το κάθε σημείο βρίσκεται σε μία απόσταση $d \leq \epsilon$ όπου το ϵ είναι και στις δύο περιπτώσεις μία παράμετρος που εξαρτάται από τη δομή του προβλήματος και τη μέθοδο που έχουμε επιλέξει για να κινηθούμε από το ένα σημείο στο άλλο.

Τέλος, το σημαντικότερο ίσως στοιχείο για την επιτυχία της διαδικασίας είναι η επιλογή της μεθόδου που θα χρησιμοποιηθεί για την αναζήτηση. Στη βιβλιογραφία του προβλήματος υπάρχουν αρκετές μέθοδοι που έχουν εφαρμοστεί για την επίλυση του προβλήματος η κάθε μία όμως από αυτές εστιάζεται στο πρόβλημα το οποίο επιλύουμε. Δηλαδή τελείως διαφορετικές μέθοδοι τοπικής αναζήτησης υπάρχουν για να επιλυθεί το πρόβλημα του πλανόδιου πωλητή και τελείως διαφορετικές για το πρόβλημα του σακιδίου. Κάποιες από αυτές παρουσιάζονται στο κεφάλαιο 13.

Συνοπτικά, τα σημαντικότερα προβλήματα που πρέπει να αντιμετωπιστούν κατά τη διάρκεια της φάσης της σχεδίασης ενός αλγορίθμου τοπικής αναζήτησης είναι:

- Το πρώτο, όπως είπαμε και παραπάνω, είναι η επιλογή της γειτονιάς, και αυτό σχετίζεται με την ανησυχία που υπάρχει για το αν είναι ή όχι εφικτή η λύση.
- Το δεύτερο πολύ σημαντικό στοιχείο που πρέπει να αντιμετωπίσουμε είναι η ποιότητα της αρχικής λύσης. Είναι ένα πολύ σημαντικό στοιχείο, γιατί όσο καλύτερη είναι η αρχική λύση τόσο περισσότερες πιθανότητες υπάρχουν να οδηγηθούμε ευκολότερα και γρηγορότερα σε βελτίωση της λύσης με τη χρήση της τοπικής αναζήτησης.
- Το τρίτο και σημαντικότερο στοιχείο για την επιτυχία του αλγορίθμου είναι η μέθοδος που θα χρησιμοποιηθεί για να βελτιώσει την αρχική λύση. Οι περισσότερες μέθοδοι που έχουν αναπτυχθεί στη βιβλιογραφία παρουσιάζουν σημαντικά προβλήματα όσον αφορά την ποιότητα της λύσης που επιστρέφουν και αυτό γιατί αν κάποια στιγμή πέσουν σε τοπικό ελάχιστο είναι πολύ δύσκολο να ξεκολλήσουν από εκεί. Από τις μεθόδους που έχουν αναπτυχθεί και που αποφεύγουν εύκολα την παγίδα του κολλήματος σε τοπικά ελάχιστα είναι και η μέθοδος Lin-Kernigham [255].

Κατά τη διάρκεια της διαδικασίας τοπικής αναζήτησης θα οδηγηθούμε σε κάποιο τοπικό ελάχιστο. Το τοπικό ελάχιστο εμφανίζεται όταν για όλα τα σημεία που μπορούμε να κινηθούμε μέσα στη γειτονιά αναζήτησης δεν υπάρχει πιθανότητα να βρεθεί κάποια καλύτερη λύση. Ο στόχος είναι να βρούμε ένα όσο το δυνατό καλύτερο τοπικό ελάχιστο. Δύο διαφορετικές γειτονιές αναζήτησης,

για το ίδιο πρόβλημα, θα παράγουν δύο διαφορετικά τοπικά ελάχιστα. Πολλές φορές έχουν αναπτυχθεί μέθοδοι που εξερευνούν μονάχα ένα μέρος από τη γειτονιά αναζήτησης και όχι ολόκληρη και στη συνέχεια χρησιμοποιούν μία διαφορετική γειτονιά αναζήτησης. Οι γειτονιές αναζήτησης που εξετάζουν ονομάζονται γειτονιές αναζήτησης μεταβλητού μήκους (variable depth search).

Αναλυτικά αυτό που μπορούμε να πούμε για την τοπική αναζήτηση είναι ότι μπορεί να θεωρηθεί ως ο παλιότερος και απλούστερος μεθευρετικός αλγόριθμος [425]. Όπως έχει ήδη αναφερθεί, αρχίζουμε με κάποια δεδομένη λύση, όπως όλοι οι μεθευρετικοί αλγόριθμοι που θα περιγράψουμε στη συνέχεια. Η λύση παράγεται είτε με τυχαίο τρόπο είτε με κάποιο αλγόριθμο απληστίας. Σε κάθε επανάληψη ο αλγόριθμος αντικαθιστά την τρέχουσα λύση με μία γειτονική λύση που βελτιώνει την αντικειμενική συνάρτηση. Η αναζήτηση σταματά όταν όλες οι υποψήφιες γειτονιές έχουν εξερευνηθεί και καμία καλύτερη λύση δεν έχει βρεθεί, πράγμα που σημαίνει ότι βρέθηκε ένα τοπικό ελάχιστο. Αν έχουμε μεγάλες γειτονιές αναζήτησης τότε οι υποψήφιες λύσεις μπορεί να είναι ένα υποσύνολο της συνολικής γειτονιάς. Η μη εξερεύνηση όλης της γειτονιάς γίνεται για να επιταχύνουμε τη διαδικασία. Οι αλγόριθμοι τοπικής αναζήτησης μπορεί να ταξινομηθούν ανάλογα με τον τρόπο που δημιουργείται η γειτονιά αναζήτησης για παράδειγμα αν η γειτονιά δημιουργείται με κάποιο καθορισμένο τρόπο ή αν η γειτονιά αναζήτησης παράγεται με στοχαστικό τρόπο είτε ανάλογα με το τρόπο που επιλέγεται μία καλύτερη λύση.

Από μία αρχική λύση s_0 ο αλγόριθμος τοπικής αναζήτησης θα δημιουργήσει μία ακολουθία από καλύτερες λύσεις $s_1, s_2, \cdots, s_k$ που έχουν τα ακόλουθα χαρακτηριστικά [425]:

- Υπό κάποιες συνθήκες, για παράδειγμα, αν το κριτήριο σταματήματος είναι ο μέγιστος αριθμός επαναλήψεων, το μέγεθος του k είναι γνωστό εκ των προτέρων
- $s_{i+1} \in N(s_i), \forall\ i \in [0, k-1]$.
- Για ένα πρόβλημα ελαχιστοποίησης ισχύει ότι $f(s_{i+1}) < f(s_i), \forall\ i \in [0, k-1]$.
- Το s_k είναι τοπικό ελάχιστο αν ισχύει ότι $f(s_k) < f(s), \forall\ s \in N(s_k)$.

Πολλές στρατηγικές μπορούν να χρησιμοποιηθούν για την επιλογή μίας καλύτερης γειτονικής λύσης. Η πρώτη είναι η στρατηγική που βρίσκει τη βέλτιστη βελτίωση (best improvement strategy). Στη συγκεκριμένη στρατηγική όλες οι πιθανές κινήσεις δοκιμάζονται και αυτή που δίνει τη μεγαλύτερη βελτίωση επιλέγεται. Αυτός ο τρόπος είναι πιο αποτελεσματικός από τους υπόλοιπους αλλά

Οι μεθευρετικοί και οι εξελικτικοί αλγόριθμοι είναι μέθοδοι επίλυσης που συνδυάζουν διαδικασίες τοπικής αναζήτησης και υψηλότερου επιπέδου στρατηγικές για να δημιουργήσουν μια διαδικασία που είναι ικανή να ξεφύγει από κάποιο τοπικό ελάχιστο. Τα τελευταία χρόνια ο χώρος των μεθευρετικών και εξελικτικών αλγορίθμων έχει μεγάλη ανάπτυξη καθώς έχει προταθεί ένας μεγάλος αριθμός από νέους αλγορίθμους που εμπνέονται από κάποιες φυσικές διαδικασίες. Ταυτόχρονα, οι συγκεκριμένοι αλγόριθμοι έχουν εφαρμοστεί σε πραγματικά προβλήματα με πολύ καλά αποτελέσματα καθώς η δομή τους παρέχει τη δυνατότητα επίλυσης υπολογιστικά δύσκολων προβλημάτων με μικρό υπολογιστικό φόρτο. Ενδεικτικά, αναφέρονται μερικοί τομείς που οι μέθοδοι αυτοί έχουν εφαρμοστεί, διοικητική επιστήμη, χρηματοοικονομική λήψη αποφάσεων, διαχείριση εφοδιαστικής αλυσίδας, πρόβλεψη ζήτησης. Σε αυτό το βιβλίο θα δούμε αναλυτικά ένα μεγάλο αριθμό από μεθευρετικούς και εξελικτικούς αλγορίθμους και αλγορίθμους εμπνευσμένους από τη φύση ενώ θα δοθούν και ορισμένες εφαρμογές για να παρουσιαστεί ο τρόπος χρήσης των αλγορίθμων και η αποτελεσματικότητά τους.

Μερικοί από τους αλγορίθμους που περιγράφονται και αναλύονται στο βιβλίο είναι ο αλγόριθμος της προσομοιωμένης ανόπτησης, ο αλγόριθμος της περιορισμένης αναζήτησης, οι γενετικοί αλγόριθμοι, ο αλγόριθμος της διαφορικής εξέλιξης, ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων, ο αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών, ο αλγόριθμος βελτιστοποίησης ζευγαρώματος μέλισσών, ο αλγόριθμος επιλογής κλώνων κ.α. Το παρόν βιβλίο απευθύνεται σε ένα αρκετά πλατύ κοινό το οποίο περιλαμβάνει διοικητικά στελέχη επιχειρήσεων, χρηματοοικονομικούς αναλυτές, τραπεζικά στελέχη, στελέχη που ασχολούνται με τη διαχείριση της εφοδιαστικής αλυσίδας, προπτυχιακούς και μεταπτυχιακούς φοιτητές, υποψήφιους διδάκτορες καθώς και σε ερευνητές που ασχολούνται με τη διοικητική επιστήμη, την επιχειρησιακή έρευνα, τη βελτιστοποίηση και γενικά οποιονδήποτε ασχολείται με την ανάπτυξη αλγορίθμων για επίλυση προβλημάτων βελτιστοποίησης. Ο στόχος της προσπάθειας αυτής ήταν να δημιουργηθεί ένα ολοκληρωμένο σύνολο μεθόδων, το οποίο θα προσφερθεί στον αναγνώστη, ο οποίος βλέποντας τα χαρακτηριστικά και τις διαφορές των μεθόδων θα μπορεί να επιλέξει την πιο κατάλληλη μέθοδο για το πρόβλημα βελτιστοποίησης που έχει να επιλύσει.

εκδόσεις
ΚΛΕΙΔΑΡΙΘΜΟΣ
Δαμασκού 4, Σταθμός Λαρίσης, 10440 ΑΘΗΝΑ, Τηλ. 210-5237635
www.klidarithmos.gr info@klidarithmos.gr

ISBN 978-960-461-422-6