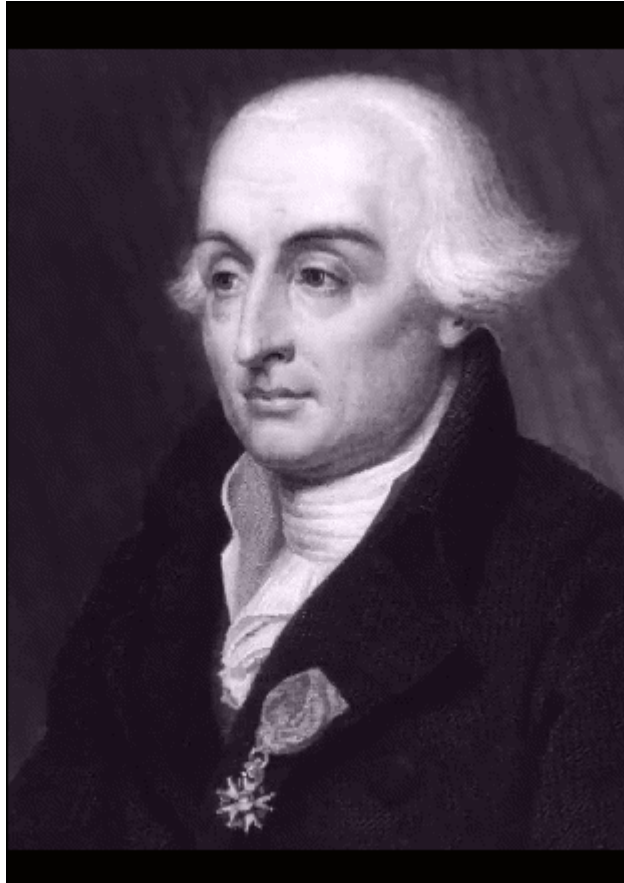




ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ



**Σύγκριση Αλγορίθμων Χαλάρωσης Lagrange για την επίλυση
κάτω φραγμένου τετραγωνικού προβλήματος με έναν
περιορισμό**

Επιτροπή:
Ε. Γρηγορούδης
Μ. Δούμπος
Ν. Ματσατσίνης

Διπλωματική Εργασία: Αντώνης Λεντζάκης

Χανιά, Οκτώβριος 2006

Ευχαριστίες

Θέλω να ευχαριστήσω την οικογένειά μου, τον καθηγητή κύριο Μυγδαλά Αθανάσιο, τον καθηγητή κύριο Κουικόγλου Βασίλειο, την κυρία Καρακίτσιου Αθανασία, τον κύριο Μαρινάκη Γιάννη και τέλος, τους πολύ καλούς μου φίλους Αγγελόπουλο Σπύρο, Ανδρέου Χάρη, Καββαδά Παναγιώτη, Κιμιωνή Γιάννη, Κωστάκη Μιχάλη, Μιχαηλίδη Νίκο, Μπελλώνια Λουκά, Μπραΐμη Ηλία, Παπαδομανωλάκη Μάνο και Προτονοτάριο Βασίλη για την βοήθεια και την πολύ σημαντική υποστήριξη που μου παρείχαν.

Περιεχόμενα

1	Γενική Εισαγωγή	2
1.1	Εισαγωγή	2
1.2	Τετραγωνικό Πρόβλημα Σακιδίου (Quadratic Knapsack Problem)	3
2	Χαλάρωση Lagrange	7
2.1	Βασική Ιδέα	7
2.2	Τρόποι Χρήσης της Χαλάρωσης Lagrange	13
2.3	Δυσκό Πρόβλημα Lagrange	22
3	Αλγόριθμοι για την επίλυση κάτω φραγμένου, τετραγωνικού προβλήματος σακιδίου με έναν περιορισμό	24
3.1	Εισαγωγικά	24
3.2	Βασικές Έννοιες	25
3.3	Ανάλυση βιβλιογραφίας	27
3.4	Θεωρία για τον αλγόριθμο προσεγγιστικής διαμέσου	30
3.5	Ένας αλγόριθμος προσεγγιστικής διαμέσου	33
3.6	Θεωρία για τον αλγόριθμο Brucker	39
3.7	Ο αλγόριθμος Brucker	42
4	Αποτελέσματα	47
4.1	Αποτελέσματα	47
4.2	Συμπεράσματα	48

Κεφάλαιο 1

Γενική Εισαγωγή

1.1 Εισαγωγή

Σε αυτήν την εργασία θα γίνει προσπάθεια επίλυσης Τετραγωνικών Προβλημάτων Σακιδίου μεγέθους n , όπου οι συντελεστές καθώς και οι μεταβλητές παίρνουν τυχαίες τιμές μεταξύ του 0 και του 1, ενώ η τιμή της χωρητικότητας προκύπτει ομοίως τυχαία μεταξύ του 0 και n , κάνοντας χρήση τροποποιημένων αλγορίθμων που πρότειναν οι Bitran και Hax [13], Nou [48] και Brucker [16]. Οι τρεις αυτοί αλγόριθμοι κάνουν χρήση της Χαλάρωσης Lagrange, μίας ευρετικής τεχνικής που αναλύεται εκτενώς στο Κεφάλαιο 2. Ο σκοπός της εργασίας αυτής είναι η παρουσίαση της θεωρίας πάνω στην οποία στηρίζεται ο κάθε αλγόριθμος, γενικά και ειδικά, η παρουσίαση σε ψευδοκώδικα του κάθε αλγορίθμου ξεχωριστά, η παρουσίαση τεχνικών ταξινόμησης που χρησιμοποιούνται σε κάθε αλγόριθμο (αν χρησιμοποιούνται), η παρουσίαση των διαφορών, των πλεονεκτημάτων και των μειονεκτημάτων του εκάστοτε αλγορίθμου και τέλος η σύγκριση της απόδοσης των τριών αυτών αλγορίθμων κατά την επίλυση του Τετραγωνικού Προβλήματος Σακιδίου. Η εργασία είναι οργανωμένη ως ακολούθως. Στο 1ο μέρος γίνεται μία εισαγωγή στο Τετραγωνικό Πρόβλημα Σακιδίου, στο 2ο μέρος γίνεται ανάλυση της τεχνικής της Χαλάρωσης Lagrange που χρησιμοποιήθηκε και από τους τρεις αλγορίθμους για την εύρεση αποτελεσμάτων για τα παραχθέντα Τετραγωνικά Προβλήματα Σακιδίου και τέλος στο 3ο μέρος γίνεται χαρακτηρισμός της βέλτιστης λύσης του προβλήματος (3.1) με χρήση των αντίστοιχων συνθηκών Kuhn-Tucker,

ανάλυση αλγοριθμικών προσεγγίσεων για την επίλυση του προβλήματος (3.1) και παρουσίαση θεωρητικών αποτελεσμάτων. Τέλος, στο 3ο μέρος γίνεται παρουσίαση των τριών αλγορίθμων που χρησιμοποιήθηκαν. Στο 4ο μέρος γίνεται αναφορά των υπολογιστικών αποτελεσμάτων και, τέλος, στο 5ο μέρος παρουσιάζονται τα συμπεράσματά μας.

1.2 Τετραγωνικό Πρόβλημα Σακιδίου (Quadratic Knapsack Problem)

Το *Τετραγωνικό Πρόβλημα Σακιδίου* (QKP) έχει σαν σκοπό (απαιτεί) την μεγιστοποίηση μίας τετραγωνικής αντικειμενικής συνάρτησης η οποία υπόκειται σε έναν περιορισμό τύπου σακιδίου (knapsack constraint), όπου όλοι οι συντελεστές θεωρούνται μη αρνητικοί και όλες οι μεταβλητές είναι δυαδικές. Το πρόβλημα αυτό βρίσκει εφαρμογές στον προσδιορισμό θέσης (location) και στην υδρολογία (hydrology) και αποτελεί γενίκευση του προβλήματος ελέγχου για το εάν ένα γράφημα περιέχει μία κλίκα (clique) δεδομένου μεγέθους.

Δεδομένων n αντικειμένων, το j -οστό αντικείμενο διαθέτει ένα θετικό ακέραιο βάρος w_j , μία θετική ακέραια χωρητικότητα σακιδίου c και έναν $n \times n$ μη αρνητικό ακέραιο πίνακα $P = (p_{ij})$, όπου p_{ij} αποτελεί ένα κέρδος που αντιστοιχεί στην επιλογή του αντικειμένου j , και, για $j > i$, $p_{ij} + p_{ji}$ αποτελεί ένα κέρδος που αντιστοιχεί στην επιλογή τόσο του αντικειμένου i όσο και του αντικειμένου j . Το *Τετραγωνικό Πρόβλημα Σακιδίου* (QKP) έχει σαν σκοπό την επιλογή ενός υποσυνόλου αντικειμένων του οποίου το συνολικό βάρος δεν υπερβαίνει την χωρητικότητα του σακιδίου, έτσι ώστε να μεγιστοποιήσει το συνολικό κέρδος. Για ευκολία στον συμβολισμό, έστω ότι με $N = \{1, \dots, n\}$ συμβολίζουμε το σύνολο αντικειμένων και με $q_j = p_{jj}$ συμβολίζουμε τα στοιχεία της διαγωνίου του πίνακα P . Αν εισάγουμε μία δυαδική μεταβλητή x_{ij} που παίρνει την τιμή 1 αν γίνει επιλογή του αντικειμένου j και 0 αλλιώς, το πρόβλημα διατυπώνεται μαθηματικά ως εξής:

$$\begin{array}{ll} \text{Μεγιστοποίηση} & z(QKP) = \sum_{i \in N} \sum_{j \in N} p_{ij} x_i x_j \\ \text{υπό} & \sum_{j \in N} w_j x_j \leq c \\ & x_j \in \{0, 1\}, j \in N. \end{array} \quad (1.1)$$

Υποθέτουμε, χωρίς απώλεια της γενικότητας, ότι ισχύει $\max_{j \in N} w_j \leq c < \sum_{j \in N} w_j$ και ότι ο πίνακας κέρδους είναι συμμετρικός, ήτοι, $p_{ij} = p_{ji}$, $\forall i, j \in N$, $j > i$.

Το QKP αποτελεί γενίκευση του *Προβλήματος Σακιδίου* (Knapsack Problem, KP), το οποίο προκύπτει αν ισχύει $p_{ij} = 0$ για όλα τα $i \neq j$. Επιπροσθέτως, το QKP διαθέτει την ακόλουθη άμεση γραφο-θεωρητική ερμηνεία. Δεδομένου ενός πλήρους, μη διατεταγμένου γραφήματος πάνω στο σύνολο κόμβων N , όπου κάθε κόμβος j διαθέτει ένα κέρδος q_j και ένα βάρος w_j και κάθε ακμή (i, j) έχει ένα κέρδος $p_{ij} + p_{ji}$, πρέπει να γίνει επιλογή ενός υποσυνόλου κόμβων $S \subseteq N$ τέτοιου ώστε το συνολικό βάρος να μην υπερβαίνει το c και το συνολικό κέρδος να είναι μέγιστο. Το συνολικό κέρδος προκύπτει από το άθροισμα των κερδών των κόμβων στο S και των ακμών, των οποίων και οι δύο άκρες ανήκουν στο S . Συνεπώς, είναι εύκολο να παρατηρηθεί ότι το QKP είναι επίσης μία γενίκευση του προβλήματος *Κλίκας* (Clique problem). Αυτό το τελευταίο πρόβλημα, στην εκδοχή του σαν πρόβλημα αναγνώρισης, έχει σαν σκοπό τον έλεγχο εάν, για ένα δεδομένο θετικό ακέραιο k , ένα δεδομένο μη διατεταγμένο γράφημα $G = (V, E)$ περιέχει ένα πλήρες υπογράφημα με k κόμβους. Μία πιθανή εκδοχή βελτιστοποίησης του προβλήματος Κλίκας δίδεται από το λεγόμενο *Πρόβλημα Πυκνού Υπογραφήματος* (*Dense Subgraph Problem*), στο οποίο είναι επιθυμητή η επιλογή ενός υποσυνόλου κόμβων $K \subseteq V$ με αριθμό στοιχείων $|K| = k$ τέτοιου ώστε το υπογράφημα του G που προκύπτει από το K να περιέχει όσο το δυνατόν περισσότερες ακμές. Αυτό το πρόβλημα μπορεί να διατυπωθεί μαθηματικά όπως το (1.1) θέτοντας $n = |V|$, $c = k$, $w_j = 1$ για $j \in N$, $p_{ij} = p_{ji} = 1$ εάν $(i, j) \in E$ και $p_{ij} = p_{ji} = 0$ σε διαφορετική περίπτωση, για $i, j \in N$. Πρέπει να σημειωθεί ότι σε αυτήν την περίπτωση ο περιορισμός σακιδίου εκφυλίζεται σε περιορισμό αριθμού στοιχείων, και θα ικανοποιείται με ισότητα κατά την βέλτιστη λύση. Προφανώς, η απάντηση στο πρόβλημα Κλίκας είναι θετική αν και μόνο αν η βέλτιστη λύση αυτού του QKP έχει τιμή $k(k - 1)$. Η πιο γνωστή εκδοχή βελτιστοποίησης του προβλήματος Κλίκας, το Max Clique, έχει σαν σκοπό ένα πλήρες υπογράφημα με ένα μέγιστο αριθμό κόμβων. Αυτό το τελευταίο πρόβλημα μπορεί να επιλυθεί μέσω ενός αλγορίθμου QKP κάνοντας χρήση δυαδικής αναζήτησης.

Το πρόβλημα Max Clique, εκτός του ότι είναι (ισχυρά) NP-hard (Μη Αιτιοκρατικό Πολυωνυμικού Χρόνου δύσκολα επιλύσιμο), αποτελεί ένα από τα δυσκολότερα προβλήματα συνδυαστικής βελτιστοποίησης που μελετάται στην βιβλιογραφία, τόσο από την πλευρά της θεωρητικής προσεγγισιμότητας όσο και από την πλευρά της πρακτικής επιλυσιμότητας. Οι ίδιες ιδιότητες εμφανίζονται και στο QKP, με

αποτέλεσμα να είναι πιο δύσκολο στην επίλυσή του από το κλασικό τετραγωνικό πρόβλημα. Συγκεκριμένα, η εύρεση μίας προσεγγιστικής λύσης του προβλήματος QKP με τιμή μεγαλύτερη από την τιμή της βέλτιστης διαιρεμένη με n^ϵ είναι NP-hard για οποιοδήποτε $\epsilon < \frac{1}{4}$ [5]. Παίρνοντας υπόψιν αυτά τα αποτελέσματα, θα ήταν αναμενόμενο ότι οποιοδήποτε άνω φράγμα, το οποίο μπορεί να προκύψει με αποδοτικό τρόπο, θα ήταν εξαιρετικά κακό για ορισμένες περιπτώσεις.

Το QKP μελετήθηκε για πρώτη φορά από τους Gallo, Hammer και Simeone [13], οι οποίοι πρότειναν ακριβείς αλγορίθμους όπου τα άνω φράγματα υπολογίζονταν με χρήση των ανώτερων επιπέδων (*upper planes*), τα οποία είναι γραμμικές συναρτήσεις αυτών των δυαδικών μεταβλητών που δεν είναι μικρότερες από από την αντικειμενική συνάρτηση του QKP πάνω στο σύνολο εφικτών λύσεων του εν λόγω QKP. Φαίνεται πως, μέχρι και πριν από λίγα χρόνια, το πρόβλημα δεν είχε μελετηθεί εκτεταμένα, αλλά έχει προσφάτως προσελκύσει μεγάλο ερευνητικό ενδιαφέρον. Οι Billionnet και Calmels [6] ακολουθούν μία προσέγγιση διακλάδωσης και τομής (*branch-and-cut*) στο εν λόγω πρόβλημα, κάνοντας χρήση μίας κλασικής διατύπωσης ILP (ακεραίου γραμμικού προγραμματισμού) με $O(n^2)$ μεταβλητές και περιορισμούς. Προσεγγίσεις που κάνουν χρήση της Χαλάρωσης Lagrange περιγράφονται από τους Chaillou, Hansen και Mahieu [9], Michelon και Veilleux [24], Hammer και Rader [14] και Billionnet, Faye και Soutif [7]. Οι Helmberg, Rendl και Weismantel [17] παίρνουν μία πιο γενική εκδοχή του προβλήματος όπου τα στοιχεία του πίνακα P μπορούν να πάρουν αρνητικές τιμές, και προτείνουν μία συνδυαστική προσέγγιση που κάνει χρήση επιπέδων τομής (*cutting planes*) και ημιορισμένου προγραμματισμού (*semidefinite programming*) και επιτρέπει τον υπολογισμό πολύ ισχυρών άνω φραγμάτων. Το Ακέραιο QKP, όπου οι μεταβλητές μπορούν να λάβουν μία ακέραια τιμή μεταξύ ενός κάτω και άνω φράγματος, μελετήθηκε από τους Bretthauer, Shetty και Syam [8], αλλά περιορίστηκαν από το γεγονός ότι οι πίνακες κέρδους P είναι διαγώνιοι, ήτοι να ισχύει $p_{ij} = 0$ για $i \neq j$. Πρέπει να σημειωθεί ότι το γενικό ακέραιο QKP μπορεί εύκολα να διατυπωθεί ως ένα QKP εφαρμόζοντας τον ίδιο μετασχηματισμό που περιγράφηκε από τους Martello και Toth [21], για την μετατροπή του Φραγμένου Προβλήματος Σακιδίου σε Πρόβλημα Σακιδίου.

Όπως είναι αναμενόμενο, λόγω της γενικότητάς του, το QKP διαθέτει ένα ευρύ πεδίο εφαρμογών. Ο Witzgall [27] παρουσίασε ένα πρόβλημα που προκύπτει στις τηλεπικοινωνίες όταν πρέπει να επιλεγεί ένας αριθμός τοποθεσιών για δορυφορικούς σταθμούς, τέτοιος ώστε η παγχώστια κυκλοφορία ψηφιακών σημάτων μεταξύ

αυτών των σταθμών να μεγιστοποιείται υπό έναν περιορισμό προϋπολογισμού. Αυτό το πρόβλημα φαίνεται πως αποτελεί QKP. Παρόμοιες διατυπώσεις προβλημάτων προκύπτουν κατά την επιλογή τοποθεσίας αεροδρομίων, σιδηροδρομικών σταθμών ή σταθμών διαχείρισης φορτίων [26]. Οι Johnson, Mehrotra και Nemhauser [18] αναφέρουν ένα πρόβλημα σχεδίασης μεταγλωττιστή (compiler) που μπορεί να διατυπωθεί σαν ένα QKP, όπως περιγράφεται στο [17]. Οι Dijkhuizen και Faigle [11] και οι Park, Lee και Park [25] θεωρούν ένα σταθμισμένο πρόβλημα maximum b-clique. Άν όλα τα βάρη των ακμών είναι μη αρνητικά, τότε αυτό το πρόβλημα αποτελεί ειδική περίπτωση του QKP και προκύπτει όταν $w_j = 1$ για $j \in N$ και $b = c$. Τέλος, το QKP εμφανίζεται σαν το υποπρόβλημα παραγωγής στηλών (column generation) κατά την επίλυση του προβλήματος διαχωρισμού γραφήματος που περιγράφεται από τους Johnson, Mehrotra και Nemhauser [18].

Κεφάλαιο 2

Χαλάρωση Lagrange

2.1 Βασική Ιδέα

Θεωρούμε το γενικό πρόβλημα

$$(P) \min z = f(x) \tag{2.1}$$

$$\text{υπό } g(x) \leq 0 \tag{2.2}$$

$$h(x) = 0 \tag{2.3}$$

$$x \in X \tag{2.4}$$

όπου $g(x) = [g_1(x), g_2(x), \dots, g_p(x)]^T$, $h(x) = [h_1(x), h_2(x), \dots, h_q(x)]^T$, $x = [x_1, x_2, \dots, x_n]^T$ και $m = p + q$.

Θα υποθέσουμε ότι το σύνολο Q είναι τέτοιο που η εξάλειψη των περιορισμών (2.2)-(2.3) από το πρόβλημα (P) θα καθιστούσε το εναπομείνον πρόβλημα

$$(P_1) \min z_1 = f(x) \tag{2.5}$$

$$\text{υπό } x \in X \tag{2.6}$$

‘εύκολο’ προς επίλυση, όπου η ευκολία αναφέρεται είτε στην ύπαρξη γνωστού αλγορίθμου για την επίλυση του (P_1) είτε στην παρουσία ιδιοτήτων που θα επέτρεπαν την ανάπτυξη νέου ειδικού αλγορίθμου για την επίλυση του (P_1) .

Το πρόβλημα (P_1) λαμβάνεται ως **χαλάρωση** (relaxation) του προβλήματος (P) , διότι όπως και στην περίπτωση της **γραμμικής χαλάρωσης** του ακέραιου γραμμικού προγραμματισμού (της αντικατάστασης δηλαδή περιορισμών του τύπου $x \in \{0, 1\}$ ή $x \in \{0, 1, \dots, u\}$ από τους περιορισμούς $0 \leq x \leq 1$ και $0 \leq x \leq u$ αντίστοιχα), η εξάλειψη των περιορισμών οδηγεί σε χαλάρωση των απαιτήσεων που οι εφικτές λύσεις του (P_1) , πρέπει να ικανοποιούν. Συγκεκριμένα, εάν συμβολίσουμε με $F(P)$ και $F(P_1) \equiv X$ τα σύνολα των εφικτών λύσεων των (P) και (P_1) αντίστοιχα, με x^* και x_1^* τις βέλτιστες λύσεις των (P) και (P_1) αντίστοιχα, και με z^* και z_1^* τις βέλτιστες αντικειμενικές τιμές $f(x^*)$ και $f(x_1^*)$ των (P) και (P_1) αντίστοιχα, τότε, όπως φαίνεται και από το Σχήμα 2.1, ισχύουν οι ακόλουθες σχέσεις:

$$F(P) \subseteq F(P_1) \quad (2.7)$$

$$x^* \in F(P_1) \quad (2.8)$$

$$z^* \geq z_1^* \quad (2.9)$$

Γενικώς αναμένουμε να ισχύει ότι

$$x_1^* \notin F(P), \quad (2.10)$$

αν και η πιθανότητα να είναι $x_1^* \in F(P)$ δεν μπορεί να παραβλεφθεί πλήρως, π.χ. όταν η σχέση (2.7) ισχύει ως ταυτότητα διότι οι εξαλειφθέντες περιορισμοί είναι όλοι τους πλεονάζοντες (redundant). Σε μια τέτοια περίπτωση αναγκαστικά έχουμε $z^* = z_1^*$ οπότε το x_1^* αποτελεί εναλλακτική βέλτιστη λύση του (P) ή ταυτίζεται με το x^* . Με άλλα λόγια, όταν η σχέση (2.10) δεν ισχύει, το **χαλαρωμένο πρόβλημα** (P_1) μπορεί να αντικαταστήσει το (P) χωρίς οιαδήποτε απώλεια. Το χαλαρωμένο πρόβλημα (relaxed problem) δύναται να αντικαταστήσει πλήρως το αρχικό (P) και στην περίπτωση που η σχέση (2.7) ισχύει μεν αλλά οι εξαλειφθέντες περιορισμοί είναι όλοι ανενεργοί (inactive, unbinding) στην βέλτιστη λύση x^* . Η

εξάλειψη μη-πλεοναζόντων περιορισμών οι οποίοι είναι ενεργοί (active, binding) στη βέλτιστη λύση του (P) συνεπάγεται εν γένει τόσο την σχέση (2.10) όσο και αυστηρές ανισότητες στις σχέσεις (2.7) και (2.9).

Στη χαλάρωση ενός προβλήματος δύναται επιπρόσθετα να συμμετέχει και η τροποποίηση της αντικειμενικής συνάρτησης. Θα λέγεται ότι το πρόβλημα

$$(P_\theta) \min z_\theta = \theta(x) \quad (2.11)$$

$$\text{υπό } x \in F_\theta \quad (2.12)$$

είναι μία χαλάρωση του (P) εάν και μόνο εάν οι ακόλουθες σχέσεις ισχύουν:

$$F(P) \subseteq F_\theta \quad (2.13)$$

$$f(x) \geq \theta(x), \quad \forall x \in F(P) \quad (2.14)$$

Παρατηρούμε ότι η σχέση (2.14) συνεπάγεται αναγκαστικά την ανισότητα

$$z^* \geq z_\theta^* \quad (2.15)$$

η οποία αναλογεί στην (2.9).

Η **χαλάρωση Lagrange** εφαρμοζόμενη στο (P) είναι η ειδική περίπτωση του (2.11)-(2.12), όπου

$$\theta(x) = f(x) + \lambda^T g(x) + \pi^T h(x), \quad (2.16)$$

$$F_\theta \equiv X, \quad (2.17)$$

και η συνάρτηση $\theta(x)$ ορίζεται για αυθαίρετες παγιομένες τιμές στις παραμέτρους $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_p]^T \geq 0$ και $\pi = [\pi_1, \pi_2, \dots, \pi_q]^T$, όπου το π είναι ελεύθερο από περιορισμούς προθέματος.

Το ούτως λαμβανόμενο χαλαρωμένο πρόβλημα

$$(LSUB) \theta(\lambda, \pi) = \min_{x \in X} \theta(x) = f(x) + \lambda^T g(x) + \pi^T h(x) \quad (2.18)$$

$$\text{υπό } x \in X \quad (2.19)$$

αναφέρεται συνήθως ως **υποπρόβλημα Lagrange** (Lagrangian sub-problem), οι παράμετροι $\lambda \geq 0$ και π ως **πολλαπλασιαστές Lagrange** (Lagrangian multipliers), ενώ το $\theta(\lambda, \pi)$ αντιπροσωπεύει την βέλτιστη αντικειμενική τιμή του υποπροβλήματος. Η συνάρτηση

$$L(x, \lambda, \pi) = f(x) + \lambda^T g(x) + \pi^T h(x) \quad (2.20)$$

με σύνολο ορισμού το $X \times \mathbb{R}_{0+}^p \times \mathbb{R}^q$ ονομάζεται **συνάρτηση Lagrange**. Παρατηρούμε ότι εάν οι τιμές των πολλαπλασιαστών **Lagrange** σταθεροποιηθούν σε εφικτές τιμές τότε $\theta(x) \equiv L(x, \lambda, \pi)$.

Η βασική ιδέα επίλυσης του (P) που η χαλάρωση Lagrange συνεπάγεται είναι να ανευρεθεί η βέλτιστη λύση του (P) μέσω επίλυσης μιας ακολουθίας ‘εύκολων’ υποπροβλημάτων (LSUB) για διαφορετικές των παραμέτρων λ και π με τέτοιο τρόπο ώστε αυτές να ενεργούν ως ποινές που προσπαθούν να επιβάλλουν την ικανοποίηση των χαλαρωμένων περιορισμών από τις διαδοχικές βέλτιστες λύσεις των αντίστοιχων (LSUB), εάν και εφόσον βεβαίως υπάρχουν τιμές λ^* και π^* για τις οποίες το (LSUB) δύναται να παράξει την βέλτιστη λύση x^* του (P). Συνθήκες υπό τις οποίες αυτό είναι δυνατόν θα μελετηθούν σε παρακάτω υποενότητες. Ακόμη όμως και όταν οι συνθήκες αυτές δεν ικανοποιούνται, η βασική ιδέα επίλυσης του (P) προσφέρεται για την δημιουργία ευρετικών αλγορίθμων προσεγγιστικής επίλυσης του (P).

Απομένει να αποδείξουμε ότι το υποπρόβλημα (LSUB) είναι όντως χαλάρωση του (P), δηλαδή ότι ικανοποιεί τις σχέσεις (2.13)-(2.14). Η ικανοποίηση της πρώτης σχέσης είναι επακόλουθο της προηγούμενης συζήτησης (δες και Σχήμα 2.1). Για την απόδειξη της ανισότητας (2.14) θεωρούμε για αυθαίρετα $\lambda \geq 0$ και π την βέλτιστη λύση $\hat{x} \in X$ του (LSUB). Έχουμε συνεπώς $f(\hat{x}) + \lambda^T g(\hat{x}) + \pi^T h(\hat{x})$, $\forall x \in X$. Από την άλλη πλευρά ως είναι x^* η βέλτιστη λύση του (P). Τότε ισχύει

$g(x^*) \leq 0$ και $h(x^*) = 0$, και επειδή $\lambda \geq 0$, έχουμε $\lambda^T g(x^*) \leq 0$ και φυσικά $f(x^*) + \lambda^T g(x^*) + \pi^T h(x^*) = f(x^*) + \lambda^T g(x^*) + 0 \leq f(x^*)$. Επίσης αναγκαστικά $x^* \in X$ και συνεπώς από την πρώτη ανισότητα έχουμε $f(\hat{x}) + \lambda^T g(\hat{x}) + \pi^T h(\hat{x}) \leq f(x^*) + \lambda^T g(x^*) + \pi^T h(x^*) \leq f(x^*)$. Τέλος, επειδή $f(x^*) \leq f(x), \forall x \in F(P)$, η ανισότητα (2.14) ικανοποιείται. Έχουμε αποδείξει το ακόλουθο αποτέλεσμα:

Θεώρημα 2.1 Ας είναι $\theta(\lambda, \pi)$ η βέλτιστη αντικειμενική τιμή του (LSUB) για αυθαίρετες τιμές $\lambda \geq 0$ και π , και z^* η βέλτιστη αντικειμενική τιμή του (P). Τότε ισχύει η ανισότητα

$$z^* \geq \theta(\lambda, \pi).$$

Το υποπρόβλημα Lagrange (LSUB) μας παρέχει σύμφωνα με το Θεώρημα 2.1 ένα **κάτω φράγμα** (lower bound, LBD) στην ζητούμενη βέλτιστη αντικειμενική τιμή του αρχικού προβλήματος (P). Μάλιστα εάν οι πολλαπλασιαστές Lagrange επιλεγθούν έτσι ώστε αυτό το κάτω φράγμα να είναι το μέγιστο δυνατόν, τότε, στις περιπτώσεις κυρίου ενδιαφέροντος, αυτό το φράγμα είναι "καλύτερο" από το φράγμα που θα εδύνατο να παραχθεί επιλύοντας το χαλαρωμένο πρόβλημα (P₁), αγνοώντας δηλαδή παντελώς τους εξαιρεφθέντες περιορισμούς. Με άλλα λόγια, στις περιπτώσεις αυτές, το μέγιστο κάτω φράγμα από το υποπρόβλημα Lagrange παρεμβάλλεται του κάτω φράγματος από το χαλαρωμένο πρόβλημα (P₁) και την βέλτιστη αντικειμενική τιμή του αρχικού προβλήματος (P) και συνεπώς είναι το ισχυρότερο από τα δύο.

Εάν κάποια εφικτή λύση \bar{x} του αρχικού προβλήματος είναι γνωστή ή δύναται να παραχθεί με κάποιον τρόπο, τότε αναγκαστικά $\bar{z} = f(\bar{x}) \geq f(x^*)$ αποτελεί ένα **άνω φράγμα** (upper bound, UBD) στην ζητούμενη βέλτιστη αντικειμενική τιμή z^* . Συνδυάζοντας συνεπώς με την ανισότητα (2.21), μία εφικτή αλλά αυθαίρετη λύση του (P) μας παρέχει την δυνατότητα να περιορίσουμε την ζητούμενη βέλτιστη αντικειμενική τιμή του (P) στο διάστημα $[\theta(\lambda, \pi), \bar{z}]$:

$$\theta(\lambda, \pi) \leq z^* \leq \bar{z}. \quad (2.21)$$

Το υποπρόβλημα Lagrange δύναται δηλαδή να χρησιμοποιηθεί τόσο για την παραγωγή κάτω φράγματος $\theta(\lambda, \pi)$ όσο και - μέσω κατάλληλων μετασχηματισμών - άνω φράγματος \bar{z} .

Όπως θα γίνει εμφανές στο υποκεφάλαιο §2.2, η χαλάρωση Lagrange είναι μία γενική μεθοδολογία για την επίλυση προβλημάτων βελτιστοποίησης, η οποία προσφέρει σοβαρά πλεονεκτήματα:

- Η γενικότητά της επιτρέπει την εφαρμογή της σε πλείστα όσα προβλήματα ανεξάρτητα από τις επιμέρους ιδιότητές των. Είναι δηλαδή εφαρμόσιμη ανεξάρτητα από την γραμμικότητα ή μη-γραμμικότητα, την κυρτότητα ή μη-κυρτότητα, την συνέχεια ή ασυνέχεια, την ακεραιότητα και την δυαδικότητα των μεταβλητών και των σχέσεών τους στο αρχικό πρόβλημα.
- Η εφαρμογή της σε δεδομένο πρόβλημα δύναται να γίνει ποικιλοτρόπως και συνεπώς έχει έμφυτη την ιδιότητα τις ευελιξίας.
- Η εφαρμογή της επιτρέπει να εκμεταλλευθούμε τις ειδικές δομές που αναγνωρίζουμε στα διάφορα προβλήματα επιτυγχάνοντας την αποσύνδεση του αρχικού προβλήματος σε υποπροβλήματα για τα οποία είτε υπάρχουν ήδη αποτελεσματικοί και αποδοτικοί αλγόριθμοι είτε δύνανται να αναπτυχθούν νέοι.
- Όπως ήδη έχει αναφερθεί, η χαλάρωση Lagrange επιτρέπει την ταχεία παραγωγή κάτω φράγματος επί της βέλτιστης αντικειμενικής τιμής του αρχικού προβλήματος. Θα δούμε στο υποκεφάλαιο §2.3 ότι η ιδιότητα αυτή επιτρέπει την ανάπτυξη της θεωρίας δυικότητας Lagrange και την ανάπτυξη δυικού προς το αρχικό πρόβλημα. Έχουν αναπτυχθεί μέθοδοι επίλυσης του δυικού προβλήματος που οδηγούν στην περαιτέρω ισχυροποίηση του κάτω φράγματος, στην καλύτερη δηλαδή προσέγγιση εκ των κάτω της ζητούμενης βέλτιστης αντικειμενικής τιμής του αρχικού προβλήματος.
- Η χαλάρωση Lagrange επιτρέπει συχνά την ανάπτυξη ευρετικών αλγορίθμων για την αποτελεσματική εύρεση “καλών” - αν όχι βέλτιστων - λύσεων του αρχικού προβλήματος και την παραγωγή άνω φράγματος στην ζητούμενη βέλτιστη αντικειμενική τιμή του αρχικού προβλήματος.
- Η ανάπτυξη τόσο κάτω όσο και άνω φραγμάτων επί της ζητούμενης βέλτιστης αντικειμενικής τιμής του αρχικού προβλήματος επιτρέπει τον εγλωβισμό της τελευταίας και συνεπώς την παροχή εγγυήσεων για την παραχθείσα ευρετική λύση σε μορφή σχετικού σφάλματος. Εκτός τούτου, η ιδιότητα του

εγκλωβισμού της βέλτιστης αντικειμενικής τιμής του αρχικού προβλήματος δίδει την δυνατότητα ανάπτυξης αλγορίθμων διακλάδωσης και φράγματος (branch-and-bound) για την επακριβή επίλυση δύσκολων συνδυαστικών και ακεραίων προβλημάτων βελτιστοποίησης.

2.2 Τρόποι Χρήσης της Χαλάρωσης Lagrange

Η ιδέα της χαλάρωσης Lagrange δύναται να χρησιμοποιηθεί ποικιλοτρόπως σε κάθε δεδομένο πρόβλημα οδηγώντας σε χαλαρωμένα υποπροβλήματα με διαφορετικές ιδιότητες όσον αφορά στην δυνατότητα επίλυσής των και στην ποιότητα των λύσεων που δύνανται να παράξουν. Επειδή συνήθως η ποιότητα των λύσεων σχετίζεται με την ευκολία ή δυσκολία ανεύρεσής των, η τελική επιλογή του χαλαρωμένου υποπροβλήματος απαιτεί το σωστό στάθμισμα των δύο αυτών παραγόντων. Στην παρούσα υποενότητα θα γίνει ανάλυση δύο εκ των βασικών τρόπων εφαρμογής της χαλάρωσης Lagrange εξετάζοντας μία σειρά από εξειδικευμένα προβλήματα.

2.2.1 Ολική Χαλάρωση

Στην ολική χαλάρωση όλοι οι περιορισμοί πλην των απλούστερων αντιστοιχίζονται σε πολλαπλασιαστές Lagrange και προστίθενται στην αντικειμενική συνάρτηση για τον σχηματισμό του υποπροβλήματος Lagrange. Με αναφορά στο γενικό πρόβλημα (2.1)-(2.4), το σύνολο X που καθορίζει τους περιορισμούς του χαλαρωμένου προβλήματος (2.18)-(2.19) ορίζεται από πολύ απλές σχέσεις, όπως ανισότητες μη αρνητικότητας των μεταβλητών, άνω ή κάτω όρια στις τιμές των μεταβλητών, και απαιτήσεις δυαδικότητας ή ακεραιότητας των λύσεων.

Παράδειγμα 2.1 Γραμμικός Προγραμματισμός

Θεωρούμε το γραμμικό πρόβλημα βελτιστοποίησης:

$$(LP) \min z = c^T x \quad (2.22)$$

$$\text{υπό} \quad A(x) \geq b \quad (2.23)$$

$$x \geq 0. \quad (2.24)$$

Για να εφαρμόσουμε ολική χαλάρωση, αντιστοιχίζουμε στους περιορισμούς (2.23) τους πολλαπλασιαστές $\lambda \geq 0$ αφού προηγουμένως θέσουμε τους περιορισμούς σε \leq - μορφή. Το αποτέλεσμα της ολικής χαλάρωσης είναι το κάτωθι υποπρόβλημα Lagrange:

$$\theta(\lambda) = \lambda^T b + \min(c^T - \lambda^T A)x \quad (2.25)$$

$$\text{υπό } x \geq 0 \quad (2.26)$$

Το πρόβλημα αυτό επιλύεται με απλή επισκόπηση αφού εάν ο συντελεστής μιας μεταβλητής x_j είναι θετικός, η ελαχιστοποίηση απαιτεί να τεθεί η μεταβλητή στο κατώτατο όριό της, να της αποδοθεί δηλαδή η τιμή 0. Αντιθέτως, εάν ο συντελεστής της μεταβλητής είναι αρνητικός, τότε αναγκαστικά $x_j \rightarrow +\infty$. Έχουμε επομένως:

$$\theta(\lambda) = \begin{cases} \lambda^T b, & \text{εάν } c^T - \lambda^T A \geq 0 \\ -\infty & \text{αλλιώς} \end{cases} . \quad (2.27)$$

Το χαλαρωμένο πρόβλημα (2.25)-(2.26) έχει δηλαδή νόημα μόνον εφόσον οι πολλαπλασιαστές Lagrange λ ικανοποιούν τις συνθήκες:

$$A^T \lambda \leq c. \quad (2.28)$$

Η βέλτιστη αντικειμενική τιμή του χαλαρωμένου προβλήματος έχει τότε την τιμή $b^T \lambda$ που είναι ένα κάτω φράγμα (LBD) στην βέλτιστη αντικειμενική τιμή του αρχικού προβλήματος (LP).

Παράδειγμα 2.2 Ακέραιος Προγραμματισμός

Θεωρούμε το γραμμικό πρόβλημα βελτιστοποίησης με μεταβλητές δυαδικής μορφής:

$$(BP) \min z = c^T x \quad (2.29)$$

$$\text{υπό } A(x) \geq b \quad (2.30)$$

$$x \in \{0, 1\}^n \quad (2.31)$$

Για να εφαρμόσουμε ολική χαλάρωση, αντιστοιχίζουμε στους περιορισμούς (2.30) $A(x) \geq b$ τους πολλαπλασιαστές $\lambda \geq 0$ αφού προηγουμένως θέσουμε τους περιορισμούς σε \leq - μορφή. Το αποτέλεσμα της ολικής χαλάρωσης είναι το κάτωθι υποπρόβλημα Lagrange:

$$\theta(\lambda) = \lambda^T b + \min (c^T - \lambda^T A)x \quad (2.32)$$

$$\text{υπό } x \in \{0, 1\}^n \quad (2.33)$$

Παράδειγμα 2.3 Μη-Γραμμικός Προγραμματισμός

Θεωρούμε το μη-γραμμικό πρόβλημα βελτιστοποίησης:

$$\text{(LP) } \min z = \sum_{j=1}^n f_j(x_j) \quad (2.34)$$

$$\text{υπό } A(x) \geq b \quad (2.35)$$

$$x \geq 0 \quad (2.36)$$

Για να εφαρμόσουμε ολική χαλάρωση, αντιστοιχίζουμε στους περιορισμούς (2.35) $A(x) \geq b$ τους πολλαπλασιαστές $\lambda \geq 0$ αφού προηγουμένως θέσουμε τους περιορισμούς σε \leq - μορφή. Το αποτέλεσμα της ολικής χαλάρωσης είναι το κάτωθι υποπρόβλημα Lagrange:

$$\theta(\lambda) = \lambda^T b + \min \sum_{j=1}^n f_j(x_j) - \lambda^T Ax \quad (2.37)$$

$$\text{υπό } x \geq 0 \quad (2.38)$$

Θέτοντας $d(\lambda) = A^T \lambda$, παρατηρούμε ότι το υποπρόβλημα διαχωρίζεται σε n προβλήματα μίας μεταβλητής:

$$\theta(\lambda) = \min f_j(x_j) - d(\lambda)_j x_j \quad (2.39)$$

$$\text{υπό } x \geq 0 \quad (2.40)$$

Υπό την προϋπόθεση ότι οι συναρτήσεις $f_j(\cdot)$ κατέχουν ορισμένες ιδιότητες, τα προβλήματα αυτά επιλύονται σχετικά εύκολα είτε με μεθόδους γραμμικής αναζήτησης (line search) είτε ακόμη και αναλυτικά.

Το κάτω φράγμα που παράγεται από την χαλάρωση είναι:

$$\theta(\lambda) = \lambda^T b + \sum_{j=1}^n \theta_j(\lambda). \quad (2.41)$$

Παράδειγμα 2.4 Απλό Πρόβλημα Χωροθέτησης

Θεωρούμε το κάτωθι πρόβλημα χωροθέτησης:

$$\text{(SPL) } \min z = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{i=1}^n f_i y_i \quad (2.42)$$

$$\text{υπό } \sum_{i=1}^n x_{ij} = 1, \forall j \quad (2.43)$$

$$x_{ij} - y_i \leq 0, \forall i, \forall j \quad (2.44)$$

$$x_{ij} \geq 0, \forall i, \forall j \quad (2.45)$$

$$y_i \in \{0, 1\}, \forall i. \quad (2.46)$$

Για να εφαρμόσουμε ολική χαλάρωση, αντιστοιχίζουμε στους περιορισμούς (2.43) $\sum_{i=1}^n x_{ij} = 1, \forall j$ τους πολλαπλασιαστές $\pi \geq 0$ αφού προηγουμένως θέσουμε τους περιορισμούς σε \leq - μορφή. Το αποτέλεσμα της ολικής χαλάρωσης είναι το κάτωθι υποπρόβλημα Lagrange:

$$\theta(\pi) = \sum_{j=1}^m \pi_j + \min \sum_{i=1}^n \sum_{j=1}^m (c_{ij} - \pi_j) x_{ij} + \sum_{i=1}^n f_i y_i \quad (2.47)$$

$$\text{υπό } x_{ij} - y_i \leq 0, \forall i, \forall j \quad (2.48)$$

$$x_{ij} \geq 0, \forall i, \forall j \quad (2.49)$$

$$y_i \in \{0, 1\}, \forall i. \quad (2.50)$$

Παρατηρούμε ότι το υποπρόβλημα είναι αθροιστικά διαχωρίσιμο με το i σε n υποπροβλήματα, ένα για κάθε i , της μορφής:

$$\theta_i(\pi) = \min \sum_{j=1}^m (c_{ij} - \pi_j)x_{ij} + f_i y_i \quad (2.51)$$

$$\text{υπό } x_{ij} - y_i \leq 0, \forall j \quad (2.52)$$

$$x_{ij} \geq 0, \forall j \quad (2.53)$$

$$y_i \in \{0, 1\}. \quad (2.54)$$

Φυσικά κάτω φράγμα υπολογίζεται από την σχέση:

$$\theta(\pi) = \sum_{i=1}^n \pi_i + \sum_{i=1}^n \theta_i(\pi). \quad (2.55)$$

Για την επίλυση των υποπροβλημάτων παρατηρούμε ότι εάν το $f_i + \sum_{j=1}^n \min\{0, (c_{ij} - \pi_j)\} < 0$ τότε το $y_i = 1$ και το $x_{ij} = 1$ εάν το $c_{ij} - \pi_i \leq 0$, διαφορετικά $y_i = 0$ και $x_{ij} = 0$.

2.2.2 Διατήρηση Γνωστού Υποπροβλήματος

Συχνά αναγνωρίζουμε στους περιορισμούς ενός προβλήματος κάποιο υποσύνολο περιορισμών που περιγράφουν τις ιδιότητες που πρέπει να έχει η λύση γνωστού προβλήματος. Λέγεται τότε ότι το τελευταίο εμφανίζεται ως **υποπρόβλημα** στο αρχικό πρόβλημα. Σε τέτοιες περιπτώσεις είναι συνήθως επιθυμητό να επιτευχθεί η χαλάρωση με τέτοιο τρόπο ώστε το γνωστό υποπρόβλημα να εμφανισθεί στην θέση του υποπροβλήματος Lagrange (2.18)-(2.19). Για την επίτευξη αυτού του σκοπού, θεωρούμε ότι το X αντιστοιχεί στο αναγνωρίσιμο υποπρόβλημα, ενώ αντιστοιχίζουμε πολλαπλασιαστές Lagrange στους υπόλοιπους περιορισμούς

και τους προσθέτουμε στην αντικειμενική συνάρτηση για να διαμορφώσουμε το υποπρόβλημα Lagrange.

Είναι δυνατόν να εμφανίζονται στους περιορισμούς ενός προβλήματος δομές που αντιστοιχούν σε διαφορετικά γνωστά προβλήματα. Η επιλογή ενός εξ αυτών για την διαμόρφωση του υποπροβλήματος Lagrange βασίζεται στο στάθμισμα των εξής παραγόντων:

- την πρόσβαση σε ή την ευκολία ανάπτυξης κώδικα επίλυσης του υποπροβλήματος,
- την πολυπλοκότητα, απαίτηση δηλαδή σε χρόνο εκτέλεσης, του αλγορίθμου επίλυσης του υποπροβλήματος,
- την διατήρηση ή μη της σπουδαιότερης ιδιότητας που πρέπει να έχει η λύση του αρχικού προβλήματος, και
- την ποιότητα του κάτω φράγματος που δύναται να παραχθεί από το υποπρόβλημα.

Συχνά οι παράγοντες αυτοί είναι αλληλοσυγκρουόμενοι και για τον λόγο αυτό η γνώση και η εμπειρία υπεισέρχονται στην διαμόρφωση του σταθμίσιματος και την τελική επιλογή υποπροβλήματος. Τα παραδείγματα που ακολουθούν θα διαφωτίσουν τις πτυχές αυτού του προβληματισμού.

Παράδειγμα 2.5 Περιορισμένα Προβλήματα Ελαχίστων Διαδρομών

Προβλήματα ανεύρεσης διαδρομών υπό διάφορους περιορισμούς εμφανίζονται συχνά στην πράξη, π.χ. στις επικοινωνίες η διαδρομή πρέπει να είναι η συντομότερη όσον αφορά τον χρόνο αλλά ταυτόχρονα και η πιο αξιόπιστη ή να χρησιμοποιεί περιορισμένο αριθμό τόξων για λόγους αξιοπιστίας και ποιότητας, ενώ στις διανομές η διανομή πρέπει να είναι η φθηνότερη από άποψη κόστους αλλά ταυτόχρονα να μην υπερβαίνει κάποιο χρονικό όριο εγγυούμενη έτσι υψηλό επίπεδο υπηρεσιών προς τους πελάτες.

Θεωρούμε ένα δίκτυο $G = (N, A)$ με δύο διακεκριμένους κόμβους $s, t \in N$ και στα τόξα $(i, j) \in A$ του οποίου έχουν καταχωρηθεί δύο βάρη c_{ij} και t_{ij} που αντιστοιχούν στο κόστος και τον χρόνο διάνυσής των. Αναζητείται η διαδρομή ελαχίστου συνολικού κόστους από το s στο t που ο συνολικός χρόνος διάνυσής της δεν υπερβαίνει ένα δεδομένο χρονικό όριο T .

Το πρόβλημα διατυπώνεται μαθηματικά ως εξής:

$$(CSP) \min z = \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.56)$$

$$\text{υπό} \quad \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = \begin{cases} 1, & \text{εάν } i = s \\ 0, & \text{εάν } i \neq s, t, \forall i \in N \\ -1, & \text{εάν } i = t \end{cases} \quad (2.57)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq T \quad (2.58)$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in A. \quad (2.59)$$

Παρατηρώντας το μαθηματικό πρότυπο μπορούμε να διαπιστώσουμε ότι η απουσία του περιορισμού (2.58) θα το καθιστούσε ένα απλό πρόβλημα ελάχιστης διαδρομής (shortest path problem) το οποίο επιλύεται σε χρόνο $O(|N|^2)$ και ότι η απουσία των περιορισμών (2.57) θα το μετέτρεπε στο γνωστό πρόβλημα σακιδίου (knapsack problem) το οποίο επιλύεται με δυναμικό προγραμματισμό σε ψευδοπολυωνυμικό χρόνο $O(T|A|)$ ή με την μέθοδο διακλάδωσης και φράγματος (branch-and-bound). Αναγνωρίζουμε δηλαδή δύο γνωστές δομές τους περιορισμούς του (CSP) και επομένως η χαλάρωση Lagrange δύναται να εφαρμοσθεί με δύο διαφορετικούς τρόπους: χαλαρώνοντας είτε τον περιορισμό (2.58) είτε τους περιορισμούς (2.57). Στην πράξη, η τελική επιλογή θα εξαρτηθεί από την προσπάθεια ή την ευκολία ανάπτυξης κώδικα επίλυσης για το ένα ή το άλλο πρόβλημα και από τον προσδοκώμενο χρόνο εκτέλεσης του κώδικα επίλυσης για την συγκεκριμένη εφαρμογή. Αμφότεροι οι παράγοντες συνηγορούν στην διατήρηση του υποπροβλήματος ελάχιστης διαδρομής και στην χαλάρωση του περιορισμού (2.58). Επιπροσθέτως, η διατήρηση των περιορισμών (2.57) εγγυάται την συνεκτικότητα της λύσης, εγγυάται δηλαδή την σπουδαιότερη ιδιότητα μιας διαδρομής - άσχετα από το εάν μια τέτοια διαδρομή ικανοποιεί ή όχι τον χαλαρωμένο περιορισμό (2.58). Από την άλλη πλευρά, μία λύση που ικανοποιεί τον περιορισμό (2.58) αλλά όχι τους

περιορισμούς (2.57) δεν είναι κατά ανάγκη συνεκτική και συνεπώς ούτε αποτελεί διαδρομή.

Εισάγοντας τον πολλαπλασιαστή Lagrange $\lambda \geq 0$ για τον περιορισμό (2.58), το υποπρόβλημα Lagrange λαμβάνει την μορφή:

$$\theta(\lambda) = -\lambda T + \min_{(i,j) \in A} \sum (c_{ij} + \lambda t_{ij}) x_{ij} \quad (2.60)$$

$$\text{υπό} \quad \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = \begin{cases} 1, & \text{εάν } i = s \\ 0, & \text{εάν } i \neq s, t, \forall i \in \mathcal{N} \\ -1, & \text{εάν } i = t \end{cases} \quad (2.61)$$

$$x_{ij} \geq 0, \forall (i, j) \in A. \quad (2.62)$$

Για δεδομένο $\lambda \geq 0$, το γινόμενο $-\lambda T$ είναι σταθερό και το πρόβλημα ελαχιστοποίησης διεξάγει αναζήτηση στο δίκτυο G για μία διαδρομή που εκκινεί από τον κόμβο s και τερματίζει στον κόμβο t με ελάχιστο συνολικό κόστος όταν σε κάθε τόξο (i, j) του δικτύου έχει καταχωρηθεί ένα κόστος διέλευσης $c_{ij} + \lambda t_{ij}$. Στην περίπτωση αυτή ο πολλαπλασιαστής Lagrange λ δύναται να ερμηνευθεί ως συντελεστής μετατροπής των μονάδων χρόνου σε χρηματικές μονάδες.

Παράδειγμα 2.6 Πρόβλημα Πλανόδιου Πωλητή

Το πρόβλημα του πλανόδιου πωλητή (traveling salesman problem) αφορά την εύρεση της συντομότερης (σε χρόνο, απόσταση ή κάποιο άλλο κόστος) διαδρομής για ένα όχημα (ή πωλητή) με αφετηρία κάποιο σημείο, π.χ. ένα κέντρο διανομής, και επιστροφή στο ίδιο σημείο αφού επισκεφθεί έναν σταθερό αριθμό πελατών ακριβώς μία φορά τον καθένα. Η αντίληψη που έχουμε για αυτήν τη διαδρομή είναι επομένως ένας κύκλος που διέρχεται από τους κόμβους που αντιστοιχούν στο σημείο αφετηρίας και τους πελάτες ακριβώς μία φορά. Για την διαμόρφωση ενός μαθηματικού προτύπου θα υποθέσουμε ότι οι κόμβοι ανήκουν σε ένα μη-διατεταγμένο, πλήρες γράφημα. Ας είναι $i = 2, \dots, n$ οι κόμβοι των πελατών και $i = 1$ ο κόμβος αφετηρίας. Από την υπόθεση, κάθε μη-διατεταγμένο ζεύγος $\{i, j\}$ με $i \neq j$, $i = 1, \dots, n$, $j = 1, \dots, n$, αντιστοιχεί σε ένα σύνδεσμο ή ακμή του γραφήματος. Σε κάθε τέτοιο σύνδεσμο αντιστοιχίζουμε ένα βάρος c_{ij} που είναι ίσο με το κόστος (όπως και αν νοείται αυτό) της διαδρομής του οχήματος από το i στο j ή αντίστροφα. Αν ορίσουμε τις δυαδικές μεταβλητές

$$x_{ij} = \begin{cases} 1, & \text{εάν το όχημα κάνει χρήση του συνδέσμου } \{i, j\}, \\ 0, & \text{αλλιώς, } \forall i, \forall j, i \neq j \end{cases}$$

τότε το πρότυπο διαμορφώνεται ως εξής:

$$\text{(TSP) } \min z = \sum_{i=1}^n \sum_{j=1, i \neq j}^n c_{ij} x_{ij} \quad (2.63)$$

$$\text{υπό } \sum_{j=1}^n x_{ij} = 2, i = 1, \dots, n \quad (2.64)$$

$$\sum_{i \in C} \sum_{j \in \bar{C}} x_{ij} \geq 1, \forall C \subset \{1, \dots, n\}, C \neq \emptyset \quad (2.65)$$

$$x_{ij} \in \{0, 1\}, \forall i, \forall j, i \neq j \quad (2.66)$$

Οι περιορισμοί (2.64) επιβάλλουν στην λύση να διαθέτει δύο συνδέσμους σε κάθε κόμβο, έτσι ώστε το όχημα να εισέλθει κατά μήκος του ενός και να εξέλθει κατά μήκος του άλλου, ενώ οι περιορισμοί (2.65) αποβλέπουν στην εξάλειψη κυκλικών διαδρομών (υποκύκλων) που δεν διέρχονται από όλους τους κόμβους απαιτώντας από κάθε εν δυνάμει υπόκυκλο, που αντιπροσωπεύεται από ένα κατάλληλο μη-κενό υποσύνολο C των κόμβων, να διαθέτει στην λύση του τουλάχιστον έναν σύνδεσμο που οδηγεί στο συμπληρωματικό του υποσύνολο $\bar{C} = \{1, \dots, n\} \setminus C$.

Σε προβλήματα πλανόδιου πωλητή, ο σκοπός είναι η μετατροπή τους σε ένα ευκολότερο πρόβλημα ώστε η να διευκολυνθεί επίλυσή του. Πιο συγκεκριμένα, αν γίνει εισαγωγή του πολλαπλασιαστή Lagrange $\lambda \geq 0$ για τον περιορισμό (2.64), το υποπρόβλημα Lagrange που προκύπτει λαμβάνει την μορφή ενός προβλήματος MST (Minimum Spanning Tree το οποίο επιλύεται σαφώς πιο εύκολα και έτσι θα έχουμε:

$$\theta(\lambda) = \sum_{i=1}^n 2 \cdot \lambda_i + \min \sum_{i=1}^n \sum_{j=1, i \neq j}^n (c_{ij} - \lambda_i) x_{ij}$$

$$\text{υπό } \sum_{i \in C} \sum_{j \in \bar{C}} x_{ij} \geq 1, \forall C \subset \{1, \dots, n\}, C \neq \emptyset$$

$$x_{ij} \in \{0, 1\}, \forall i, \forall j, i \neq j$$

2.3 Δυικό Πρόβλημα Lagrange

Διαπιστώσαμε στο Θεώρημα 2.1 της υποενότητας §2.1 ότι για αυθαίρετες τιμές των πολλαπλασιαστών $\lambda \geq 0$ και π , η βέλτιστη αντικειμενική τιμή $\theta(\lambda, \pi)$ του χαλαρωμένου υποπροβλήματος (LSUB) αποτελεί ένα κάτω φράγμα για την βέλτιστη αντικειμενική τιμή του αρχικού προβλήματος (P). Είναι επομένως επιθυμητό αντί για την αυθαίρετη επιλογή τιμών να γίνει συνειδητή τέτοιων τιμών που θα ισχυροποιήσουν, όσο είναι δυνατό, το κάτω φράγμα, που θα το φέρουν δηλαδή όσο το δυνατό εγγύτερα στην βέλτιστη αντικειμενική τιμή του αρχικού προβλήματος. Η σκέψη αυτή μας οδηγεί στην διατύπωση του κάτωθι προβλήματος:

$$(D) \max \theta(\lambda, \pi) \tag{2.67}$$

$$\text{υπό } \lambda \geq 0 \tag{2.68}$$

Το πρόβλημα αυτό όπου αναζητούνται οι τιμές των πολλαπλασιαστών που καθιστούν το κάτω φράγμα $\theta(\lambda, \pi)$ μέγιστο καλείται **δυικό (dual) πρόβλημα Lagrange** του (P), που με την σειρά του αναφέρεται ως το **πρωταρχικό (primal) πρόβλημα**.

Παρατηρούμε ότι η συνάρτηση $\theta(\lambda, \pi)$ δεν δίδεται σε **αναλυτική μορφή** (explicit form) αλλά σε **πεπλεγμένη μορφή** (implicit form) καθώς ορίζεται ως η βέλτιστη αντικειμενική τιμή ενός προβλήματος βελτιστοποίησης, του (LSUB).

Εάν επαναφέρουμε την συνάρτηση (2.20), το δυικό πρόβλημα (D) δύναται να τεθεί σε μορφή **προβλήματος maxmin** ή **σαγματικού προβλήματος** (saddle point problem):

$$(SPP) \max \min_{\lambda \geq 0} \min_{x \in X} L(x, \lambda, \pi) = f(x) + \lambda^T g(x) + \pi^T h(x) \tag{2.69}$$

Στο πρόβλημα αυτό η συνάρτηση $L(x, \lambda, \pi)$ ελαχιστοποιείται ως προς το $x \in X$ και μεγιστοποιείται ως προς το $\lambda \geq 0$ και το π . Η ελαχιστοποίηση αντιστοιχεί στο υποπρόβλημα (LSUB) και η μεγιστοποίηση στο δυικό πρόβλημα (D).

Κεφάλαιο 3

Αλγόριθμοι για την επίλυση κάτω φραγμένου, τετραγωνικού προβλήματος σακιδίου με έναν περιορισμό

3.1 Εισαγωγικά

Σε αυτή την εργασία, θεωρούμε το ακόλουθο τετραγωνικό πρόγραμμα

$$\begin{array}{ll} \text{Ελαχιστοποίηση} & \sum_{i=1}^n \frac{1}{2}g_i x_i^2 + h_i x_i \\ \text{υπό} & \sum_{i=1}^n b_i x_i = b_0, \\ & x_i \geq 0, i = 1, \dots, n, \end{array} \quad (3.1)$$

όπου $g_i, b_i > 0$ για $i = 0, \dots, n$. Τα κάτω φράγματα, l_i , των μεταβλητών δεν προσθέτουν κάποια επιπλέον πολυπλοκότητα, εφόσον μπορούν να απαλειφούν με

μία αλλαγή των μεταβλητών (υποθέτοντας ότι το πρόβλημα είναι εφικτό, ήτοι, $\sum_{i=1}^n b_i l_i \leq b_0$). Το πρόβλημα (3.1) εμφανίζεται συχνά σαν υποπρόβλημα σε διάφορους τομείς εφαρμογών: Σε προβλήματα μέγιστης ροής πολλαπλών αγαθών (multicommodity maximum flow problems), [31], και προβλήματα εξισορρόπησης κυκλοφορίας (traffic equilibrium) και προβλήματα οικονομίας δικτύων (network economics), [47], για να αναφέρουμε μερικά. Επίσης εμφανίζεται σαν υποπρόβλημα στην βελτιστοποίηση υποκλίτου (subgradient optimization), [14]. Έχουν προταθεί διάφοροι αλγόριθμοι για την επίλυση του προβλήματος (3.1). Μπορούμε να αναφέρουμε για παράδειγμα τους [20, 31]. Αρκετοί συγγραφείς έχουν θεωρήσει το πιο γενικό πρόβλημα όπου υπάρχουν επίσης και άνω φράγματα για τις μεταβλητές, δες π.χ. [13, 16, 32, 39, 49, 54]. Οι αντίστοιχοι αλγόριθμοι μπορούν, βέβαια, να εφαρμοσθούν στο πρόβλημα (3.1) σαν μία ειδική περίπτωση.

3.2 Βασικές Έννοιες

Το πρόβλημα (3.1) μπορεί να επαναδιατυπωθεί σαν

$$\text{Ελαχιστοποίηση} \quad \sum_{i=1}^n \frac{1}{2} g_i x_i^2 + h_i x_i \quad (3.2)$$

$$\text{υπό} \quad b_0 - \sum_{i=1}^n b_i x_i = 0, \quad (3.3)$$

$$-x_i \leq 0, i = 1, \dots, n, \quad (3.4)$$

Έστω ότι λ και u_i , $i = 1, \dots, n$ είναι οι πολλαπλασιαστές Kuhn-Tucker για τους περιορισμούς (3.3) και (3.4) αντίστοιχα. Οι συνθήκες Kuhn-Tucker για το πρόβλημα (3.2)-(3.4) διατυπώνονται ως

$$g_i x_i + h_i - v_i - \lambda b_i = 0, \quad \forall i \quad (3.5)$$

$$v_i x_i = 0, \quad \forall i \quad (3.6)$$

$$v_i \geq 0, \quad \forall i \quad (3.7)$$

μαζί με τους περιορισμούς (3.3) και (3.4). Πρέπει να σημειωθεί ότι οι συνθήκες

Kuhn-Tucker είναι αναγκαίες και ικανές συνθήκες για ένα ολικό βέλτιστο στο πρόβλημα (3.2)-(3.4). Για δεδομένο λ προκύπτει η ακόλουθη λύση:

$$\begin{aligned} x_i(\lambda) &= \max\{(\lambda b_i - h_i)/g_i, 0\}, \\ v_i(\lambda) &= \max\{h_i - \lambda b_i, 0\}. \end{aligned} \quad (3.8)$$

Θεώρημα 3.1. Η λύση (3.8) ικανοποιεί όλες τις συνθήκες Kuhn-Tucker εκτός πιθανόν από την (3.3).

Απόδειξη. Ελέγχεται εύκολα.

Έτσι, για την επίλυση του προβλήματος (3.2)-(3.4) πρέπει να βρεθεί ένα κατάλληλο λ τέτοιο ώστε ο περιορισμός (3.3) να ικανοποιείται. Αν θέσουμε

$$\tilde{x}_i(\lambda) = (\lambda b_i - h_i)/g_i, \quad \forall i \quad (3.9)$$

$$y_i(\lambda) = h_i/b_i. \quad \forall i \quad (3.10)$$

Επομένως, το y αποτελεί ένα διάνυσμα τιμών θλάσης (breakpoint values), εφόσον ισχύει

$$x_i(\lambda) = \begin{cases} 0 & , \text{αν } \lambda < y_i, \\ \tilde{x}_i(\lambda) & , \text{αν } \lambda \geq y_i, \end{cases} \quad (3.11)$$

Πρέπει να σημειωθεί ότι η $x_i(\lambda)$ είναι τμηματικά γραμμική, αύξουσα και κυρτή. Έστω τώρα

$$b(\lambda) = b_0 - \sum_{i=1}^n b_i x_i(\lambda).$$

Η $b(\lambda)$ είναι προφανώς τμηματικά γραμμική, φθίνουσα και κοίλη. Παίρνει όλες τις τιμές μεταξύ του b_0 και $-\infty$, επομένως (εφόσον $b_0 > 0$) υπάρχει μία μοναδική τιμή για τον πολλαπλασιαστή λ^* τέτοια ώστε $b(\lambda^*) = 0$, που υποδηλώνει ικανοποίηση

του περιορισμού (3.3). Πρέπει να σημειωθεί ότι όλες οι μεταβλητές x_i για τις οποίες ισχύει $y_i > \lambda^*$, είναι μηδέν στη βέλτιστη λύση. Ο αριθμός των μη μηδενικών μεταβλητών στη βέλτιστη λύση εκφράζεται με το s^* , ήτοι,

$$s^* = |\{i \in \{1, \dots, n\} | x_i(\lambda^*) > 0\}|.$$

Αν η τιμή του s^* ήταν γνωστή εκ των προτέρων, τότε θα μπορούσε να γίνει επίλυση του προβλήματος (3.2)-(3.4) με ταυτοποίηση των δεικτών των s^* μικρότερων τιμών θλάσης (βλέπε π.χ. [55]), και μετέπειτα θα γινόταν επιλογή του λ έτσι ώστε να ικανοποιείται ο περιορισμός (3.3).

3.3 Ανάλυση βιβλιογραφίας

Μία πιθανή προσέγγιση για εύρεση λύσης του προβλήματος (3.1) θα ήταν να γίνει ταξινόμηση των τιμών θλάσης (breakpoints) κατά αύξουσα σειρά και μετά αναζήτηση του s^* . Οι αλγόριθμοι [20, 31, 32, 39] ακολουθούν αυτήν την προσέγγιση. Η διαφορά μεταξύ τους έγκειται στην τεχνική αναζήτησης για την εύρεση του s^* . Κάνοντας χρήση ενός αποδοτικού αλγορίθμου ταξινόμησης έχει σαν αποτέλεσμα η πολυπλοκότητα των εν λόγω αλγορίθμων να γίνεται $O(n \log n)$.

Ο Brucker, [16], αντικαθιστά την ταξινόμηση, που απαιτείται από τους παραπάνω αλγόριθμους, με τον υπολογισμό της διαμέσου (median) ενός μειούμενου συνόλου τιμών θλάσης (breakpoints) σε κάθε επανάληψη, με αποτέλεσμα έναν αλγόριθμο πολυπλοκότητας $O(n)$. Οι Pardalos και Kononov, [49], προτείνουν μία τυχαία κατανεμημένη (τυχοποιημένη) εκδοχή του αλγορίθμου [16] με χειρότερη υπολογιστική πολυπλοκότητα αλλά γρηγορότερους χρόνους επίλυσης.

Το κοινό που έχουν όλοι οι παραπάνω αλγόριθμοι είναι ότι, στην επανάληψη k , θέτουν $\lambda^k = h_s/b_s$, για κάποιο $s \in \{1, \dots, n\}$. Βάσει του προσήμου του $b(\lambda^k)$ μειώνουν το σύνολο δεικτών στο οποίο πρέπει να βρίσκεται το s^* . Τέλος, ο προσδιορισμός του λ^* είναι απλός, αφού έχει βρεθεί το s^* .

Οι Bitran και Hax, [13], προτείνουν έναν αλγόριθμο που διαφέρει ουσιαστικά από τους προαναφερθέντες αλγόριθμους, εφόσον δεν απαιτείται ταξινόμηση ή υπολογισμός, ακριβών ή προσεγγιστικών, διαμέσων. Ο αλγόριθμος έχει στην χείριστη

περίπτωση πολυπλοκότητα $O(n^2)$, αλλά στην πράξη αποδίδει πολύ καλά. Θα γίνει σύγκριση του προτεινόμενου αλγορίθμου με μια αποδοτική εκδοχή του αλγορίθμου Bitran-Hax, τον οποίο περιγράφουμε παρακάτω στην υποπαράγραφο 3.1.

Τέλος, οι Robinson et al., [54], προτείνουν έναν αλγόριθμο που βασίζεται στην γεωμετρία του προβλήματος. Επιλύουν μία σειρά απλών προβλημάτων προβολής (μετά τη χρήση ενός κατάλληλα επιλεγμένου μετασχηματισμού) και αναφέρουν ευνοϊκά υπολογιστικά αποτελέσματα (σε σχέση με τους αλγορίθμους [16, 32, 49] την εφαρμογή της μεθόδου Newton στο δυϊκό πρόβλημα του προβλήματος (3.1) με άνω φράγματα). Αν εκφράσουμε τα μεγέθη τους σε μεταβλητές του αρχικού προβλήματος, προκύπτει ο αλγόριθμος Bitran-Hax.

3.3.1 Ο αλγόριθμος Bitran-Hax

Εν συντομία, ο αλγόριθμος Bitran-Hax μπορεί να περιγραφεί ως ακολούθως. Γίνεται χαλάρωση των περιορισμών φραγμάτων στις μεταβλητές (στην περίπτωσή τους, τόσο στα άνω όσο και στα κάτω φράγματα) και επιλύουν βέλτιστα το προκύπτον χαλαρωμένο πρόβλημα. Έπειτα, είτε για όλες τις μεταβλητές που παραβιάζουν τα κάτω φράγματά τους, είτε για όλες τις μεταβλητές που παραβιάζουν τα άνω φράγματά τους, οι τιμές τους διορθώνονται (στα αντίστοιχα φράγματα), με αποτέλεσμα την απαλοιφή τους από το πρόβλημα. Για την απόφαση του ποιο από τα δύο σύνολα μεταβλητών μπορεί να διορθωθεί πρέπει να υπολογιστούν συγκεκριμένες τιμές (δες [13]). Η διαδικασία επαναλαμβάνεται μέχρι όλες οι μεταβλητές να ικανοποιούν τους περιορισμούς φραγμάτων τους.

Θεώρημα 3.2. *Αν υποθέσουμε ότι με y εκφράζεται το διάνυσμα τιμών θλάσης (breakpoints) που αντιστοιχεί στο (3.1), και ότι το λ δίδεται από*

$$\lambda = (b_0 + \sum_{i=1}^n \frac{b_i x_i}{g_i}) / \sum_{i=1}^n \frac{b_i^2}{g_i}. \quad (3.12)$$

Τότε όλες οι μεταβλητές x_i τέτοιες ώστε να ισχύει $y_i > \lambda$, $i = 1, \dots, n$, θα έχουν τιμή μηδέν στη βέλτιστη λύση.

Απόδειξη. Προκύπτει από [13].

Δεν είναι απαραίτητος ο υπολογισμών όλων των μεγεθών από την αρχή σε κάθε επανάληψη, κάτι που αποκαλύπτεται στον ακόλουθο αλγόριθμο.

Αλγόριθμος 1. Bitran-Hax

Βήμα 0. (Initialization)

Θέσε $k = 1$, και $F = \{1, \dots, n\}$.

Για όλα τα $i \in F$ θέσε $y_i = h_i/b_i$, και $\tilde{g}_i = b_i^2/g_i$.

Θέσε $N = b_0 + \sum_{i \in F} y_i \cdot \tilde{g}_i$, και $D = \sum_{i \in F} \tilde{g}_i$ (cf. (12)).

Βήμα 1. (Multiplier calculation)

Θέσε $\lambda^k = N/D$, και $L' = \{\}$.

Βήμα 2. (Pegging)

Για όλα τα $i \in F$ Do

If $y_i \geq \lambda^k$ (που συνεπάγεται ότι $x_i(\lambda^k) \leq 0$) then $L' = L' \cup \{i\}$, $x_i^* = 0$.

End Do

Βήμα 3. (Convergence check)

If $L' = \{\}$ then

Πήγαινε στο Βήμα 4.

Else

Θέσε $F = F \setminus L'$, $N = N - \sum_{i \in L'} y_i \cdot \tilde{g}_i$ και $D = D - \sum_{i \in L'} \tilde{g}_i$.

Θέσε $k = k + 1$. Πήγαινε στο Βήμα 1.

End If

Βήμα 4. (Optimal solution)

Για όλα τα $i \in F$ Do

$x_i^* = (\lambda^k b_i - h_i)/g_i$

End Do

Πρέπει να σημειωθεί ότι οι πολλαπλασιαστές, λ^k , δεν είναι συνήθως ίσοι με οποιαδήποτε από τις τιμές θλάσης (breakpoints) y_i , $i = 1, \dots, n$.

3.4 Θεωρία για τον αλγόριθμο προσεγγιστικής διαμέσου

Σε αυτή την παράγραφο θεμελιώνεται ο προτεινόμενος αλγόριθμος του Νου [48] παρέχοντας ορισμένα θεωρητικά αποτελέσματα. Ανακεφαλαιώνοντας, ο λ^* εκφράζει τον βέλτιστο πολλαπλασιαστή *Kuhn-Tucker* που σχετίζεται με τον περιορισμό (3.3), και τα \tilde{x}_i^* , $i = 1, \dots, n$ και y δίδονται από τις (3.9) και (3.10) αντίστοιχα.

Θεώρημα 3.3. Έστω $s \in \{1, \dots, n\}$. Αν υποθέσουμε ότι το διάνυσμα τιμών θλάσης (*breakpoints*) y διαχωρίζεται κατά τέτοιο τρόπο ώστε (μετά από αλλαγή δεικτών)

$$\max\{y_1, \dots, y_s\} < \min\{y_{s+1}, \dots, y_n\},$$

και ότι ο $\lambda(s)$ δίδεται από

$$\lambda(s) = (b_0 + \sum_{i=1}^s \frac{b_i x_i}{g_i}) / \sum_{i=1}^s \frac{b_i^2}{g_i}. \quad (3.13)$$

Τότε θα ισχύει $\sum_{i=1}^s b_i \tilde{x}_i(\lambda(s)) = b_0$ και $\lambda(s) \geq \lambda^*$, με την ανισότητα να γίνεται αυστηρή όποτε το $s < s^*$.

Απόδειξη. Δεδομένης της (3.13) μπορεί εύκολα να επαληθευθεί ότι $\sum_{i=1}^s b_i \tilde{x}_i(\lambda(s)) = b_0$. Κάνοντας χρήση της (3.11) παίρνουμε

$$\sum_{i=1}^s b_i x_i(\lambda(s)) \geq \sum_{i=1}^s b_i \tilde{x}_i(\lambda(s)) = b_0. \quad (3.14)$$

Θεωρούμε δύο περιπτώσεις: (i) Έστω ότι $s < s^*$. Αν $s^* > s$ τότε αυτό σημαίνει ότι $\lambda^* > \max\{y_1, \dots, y_s\}$, που συνεπάγεται ότι όλες οι μεταβλητές x_i , $i = 1, \dots, s$, θα είναι μη μηδενικές στην βέλτιστη λύση. Κάνοντας χρήση της (3.14) παίρνουμε

$$\sum_{i=1}^s b_i x_i(\lambda^*) = \{s < s^*\} < \sum_{i=1}^{s^*} b_i x_i(\lambda^*) = b_0 \leq \sum_{i=1}^s b_i x_i(\lambda(s)). \quad (3.15)$$

(ii) Έστω ότι $s \geq s^*$.

$$\sum_{i=1}^s b_i x_i(\lambda^*) = \sum_{i=1}^{s^*} b_i x_i(\lambda^*) = b_0 \leq \sum_{i=1}^s b_i x_i(\lambda(s)). \quad (3.16)$$

Εφόσον η $x_i(\lambda)$ είναι συνεχής, μονοτονικά αύξουσα και ισχύει $b_i > 0$ για όλα τα i , προκύπτει από τις (3.15) και (3.16) ότι ισχύει $\lambda(s) \geq \lambda^*$, με την ανισότητα να γίνεται αυστηρή όποτε το $s < s^*$.

Πόρισμα 3.4. Έστω $s < s^*$, τότε $\max\{y_1, \dots, y_s\} < \lambda^* < \lambda(s)$.

Απόδειξη. Το κάτω φράγμα προκύπτει από τον ορισμό του s^* , ενώ το άνω φράγμα από το Θεώρημα 3.3.

Τώρα θα δειχθεί ότι για $s = s^*, \dots, n-1$, η $\lambda(s)$ δίδει ένα αυστηρότερο όριο στον λ^* σε σχέση με την $\lambda(n)$. Αρχικά θα αποδειχθούν κάποια βοηθητικά αποτελέσματα.

Λήμμα 3.5. Έστω $s \in \{s^* + 1, \dots, n\}$. Τότε ισχύει $\max\{y_1, \dots, y_s\} > \lambda(s)$.

Απόδειξη. Έστω ότι $\max\{y_1, \dots, y_s\} \leq \lambda(s)$. Αυτό συνεπάγεται ότι $\tilde{x}_i(\lambda(s)) \geq 0$, για $i = 1, \dots, s$. Κάνοντας χρήση του Θεωρήματος 3.3, παίρνουμε

$$b_0 = \sum_{i=1}^{s^*} b_i \tilde{x}_i(\lambda^*) \leq \sum_{i=1}^{s^*} b_i \tilde{x}_i(\lambda(s)) = \{s^* < s\} < \sum_{i=1}^s b_i \tilde{x}_i(\lambda(s)) = b_0,$$

κάτι που είναι άτοπο.

Λήμμα 3.6. Έστω $s \in \{s^*, \dots, n\}$. Τότε ισχύει $\tilde{x}_j(\lambda(s)) < 0$ για $j = s+1, \dots, n$.

Απόδειξη. Έστω ότι $s = s^*$. Συνεπάγεται από τον ορισμό του s^* ότι ισχύει $\tilde{x}_j(\lambda(s)) < 0$ για $j = s+1, \dots, n$. Έστω ότι $s > s^*$. Χρησιμοποιώντας το Θεώρημα 3.3 και το Λήμμα 3.5 προκύπτει ότι $y_j > \max\{y_1, \dots, y_s\} > \lambda(s)$, επομένως ισχύει ότι $\tilde{x}_j(\lambda(s)) < 0$ για $j = s+1, \dots, n$ μέσω της (3.9).

Τώρα δύναται να αποδειχθεί ότι η $\lambda(s)$ είναι μονοτονικά φθίνουσα για $s \in \{1, \dots, s^* - 1\}$ και μονοτονικά αύξουσα για $s \in \{s^*, \dots, n-1\}$.

Θεώρημα 3.7. (i) Έστω $s \in \{1, \dots, s^* - 1\}$. Τότε $\lambda(s) > \lambda(s + 1)$.

(ii) Έστω $s \in \{s^*, \dots, n - 1\}$. Τότε $\lambda(s) < \lambda(s + 1)$.

Απόδειξη. (i) Έστω ότι ισχύει $\lambda(s) \leq \lambda(s + 1)$. Εφόσον $s < s^*$ συνεπάγεται ότι $\tilde{x}_{s+1}(\lambda(s)) > 0$ και κάνοντας χρήση του Θεωρήματος 3.3, προκύπτει

$$b_0 = \sum_{i=1}^{s+1} b_i \tilde{x}_i(\lambda(s+1)) > \sum_{i=1}^s b_i \tilde{x}_i(\lambda(s+1)) > \sum_{i=1}^s b_i \tilde{x}_i(\lambda(s)) = b_0,$$

κάτι που είναι άτοπο.

(ii) Έστω ότι ισχύει $\lambda(s) \geq \lambda(s + 1)$. Τότε, κάνοντας χρήση του Θεωρήματος 3.3 και του Λήμματος 3.6, προκύπτει

$$b_0 = \sum_{i=1}^{s+1} b_i \tilde{x}_i(\lambda(s+1)) \leq \sum_{i=1}^{s+1} b_i \tilde{x}_i(\lambda(s)) = b_0 + b_{s+1} \tilde{x}_{s+1}(\lambda(s)) < b_0,$$

κάτι που είναι άτοπο.

Πόρισμα 3.8. Αν $s \in \{s^*, \dots, n - 1\}$, τότε $\lambda(s) < \lambda(n)$.

Απόδειξη. Συνεπάγεται από το Θεώρημα 3.7.

Το ακόλουθο θεώρημα δηλώνει ότι η $\lambda(s)$ μπορεί να μας καθοδηγήσει στην αναζήτηση για το s^* .

Θεώρημα 3.9. (i) Έστω ότι $\lambda(s) < \max\{y_1, \dots, y_s\}$. Τότε $s^* < s$.

(ii) Έστω $\max\{y_1, \dots, y_s\} \leq \lambda(s) \leq \min\{y_{s+1}, \dots, y_n\}$. Τότε $s^* = s$.

(iii) Έστω ότι $\lambda(s) > \min\{y_{s+1}, \dots, y_n\}$. Τότε $s^* > s$.

Απόδειξη. (i) Προκύπτει από [13].

(ii) Εφόσον $\sum_{i=1}^n b_i x_i(\lambda(s)) = \sum_{i=1}^s b_i \tilde{x}_i(\lambda(s)) = b_0$, προκύπτει από το Θεώρημα 3.1 ότι $s^* = s$.

(iii) Από την ανισότητα $\lambda(s) > \min\{y_{s+1}, \dots, y_n\}$ συνεπάγεται ότι $\tilde{x}_j(\lambda(s)) > 0$ για τουλάχιστον $j = 1, \dots, s+1$. Έστω ότι $s^* = s$. Προκύπτει,

$$\sum_{i=1}^n b_i x_i(\lambda(s)) \geq \sum_{i=1}^s b_i \tilde{x}_i(\lambda(s)) + b_{s+1} \tilde{x}_{s+1}(\lambda(s)) > b_0,$$

που είναι άτοπο για βέλτιστη λύση. Έστω ότι $s^* < s$. Μέσω του Θεωρήματος 3.7, $\lambda^* < \lambda(s)$ και έτσι προκύπτει

$$b_0 = \sum_{i=1}^{s^*} b_i \tilde{x}_i(\lambda^*) < \sum_{i=1}^{s^*} b_i \tilde{x}_i(\lambda(s)) < \sum_{i=1}^s b_i \tilde{x}_i(\lambda(s)) = b_0$$

που είναι άτοπο.

3.5 Ένας αλγόριθμος προσεγγιστικής διαμέσου

Αυτή η παράγραφος είναι οργανωμένη ως ακολούθως. Στην υποπαράγραφο 3.5.1 γίνεται ερμηνεία των αλγορίθμων [13, 16] σε όρους ορισμένων μεγεθών (οντοτήτων) s^k , που συγκλίνουν στο s^* . Έπειτα, στην υποπαράγραφο 3.5.2 προτείνεται ένας αλγόριθμος προσεγγιστικής διαμέσου για την επίλυση του προβλήματος (3.1).

3.5.1 Κίνητρο

Όπως προαναφέρθηκε στην παράγραφο 3.2, η αποδοτική επίλυση του προβλήματος (3.1) ισοδυναμεί με αποδοτικά μέσα προσδιορισμού της τιμής του s^* . Έστωσαν \underline{s} και \bar{s} τα οποία εκφράζουν τα κάτω και άνω φράγματα, αντίστοιχα, του s^* . Μία δυνατή ερμηνεία των αλγορίθμων της παραγράφου 3.3 είναι ότι προχωράνε επαναληπτικά, υπολογίζοντας σε κάθε επανάληψη δοκιμαστικές τιμές του s^k , με συνεπακόλουθη μείωση του διαστήματος $\{\underline{s}, \bar{s}\}$. Η διαφορά τους έγκειται στον τρόπο με τον οποίο προκύπτει η σειρά $\{s^k\}_{k=1}^K$, όπου $s^K = s^*$. Κατά την σχεδίαση ενός αλγορίθμου για την επίλυση του προβλήματος (3.1), είναι σημαντικό να υπάρχει μία

ισορροπία μεταξύ του υπολογιστικού φόρτου κατά τον διαχωρισμό του διανύσματος τιμών θλάσης (breakpoints) (με την πλήρη ταξινόμηση να αποτελεί το ακραίο) και το μήκος της σειράς $\{s^k\}_{k=1}^K$, που ορίζεται σαν: $L(\{s^k\}) = \sum_{k=1}^K |s^k - s^{k-1}|$, όπου $s^0 \equiv 0$.

Σε κάθε επανάληψη του αλγορίθμου Bitran-Hax, [13], έστω ότι το s^k εκφράζει τον αριθμό των στοιχείων με τιμή μικρότερη ή ίση της τιμής του πολλαπλασιαστή λ^k . Οι πολλαπλασιαστές αποτελούν άνω φράγματα για τον βέλτιστο πολλαπλασιαστή λ^* , δηλαδή $s^k \geq s^*$. Εφόσον ισχύει επίσης ότι $s^{k+1} < s^k$, ο αλγόριθμος παράγει μία σειρά $\{s^k\}$ τέτοια ώστε $n = s^1 > \dots > s^K \equiv s^*$. Πρέπει να σημειωθεί ότι ο αλγόριθμος υποθέτει ότι όλες οι μεταβλητές είναι μη μηδενικές στην αρχική βέλτιστη λύση ($s^1 = n$), ενώ στις επόμενες επαναλήψεις το s^k μειώνεται διαδοχικά έως ότου να συγκλίνει στο s^* . Επίσης, πρέπει να σημειωθεί ότι ο υπολογιστικός φόρτος του υπολογισμού του $\{\lambda^k\}_{k=1}^K$ (Αλγόριθμος 1, Βήματα 0 και 3) είναι ανάλογος του μήκους $L(\{s^k\})$.

Ο αλγόριθμος του Brucker, [16], υπολογίζει διαμέσους μίας αλληλουχίας διανυσμάτων τιμών θλάσης (breakpoints), μειούμενων μεγεθών. Κάνοντας χρήση των διαμέσων σαν δοκιμαστικά σημεία λ^k και βάσει της τιμής του $b(\lambda^k)$ γίνεται μείωση του συνόλου των εναπομείνουσων τιμών θλάσης (breakpoints). Έτσι, παράγεται μία σειρά $\{s^k\}$ τέτοια ώστε $\lfloor \frac{1}{2}(1+n) \rfloor = s^1, \dots, s^k = s^*$. Σε σχέση με τον αλγόριθμο των Bitran και Hax, η αρχική υπόθεση για την τιμή του s^* είναι πιο λογική, αλλά ο υπολογιστικός φόρτος του υπολογισμού των διαμέσων αποτρέπει τον εν λόγω αλγόριθμο από το να αποδίδει ικανοποιητικά, όσον αφορά τους χρόνους επίλυσης.

Ο αλγόριθμος Bitran-Hax ελαχιστοποιεί τον φόρτο λόγω διαχωρισμού, αλλά προκύπτει μεγαλύτερο μήκος σειράς (πάντα μεγαλύτερο του n), ενώ ο αλγόριθμος του Brucker ελαχιστοποιεί το αναμενόμενο μήκος της σειράς $\{s^k\}$, (πάντα μικρότερο του n), αλλά αυτό γίνεται με σοβαρό υπολογιστικό κόστος κατά τον υπολογισμό των διαμέσων. Δεδομένων των Θεωρημάτων 3.3 και 3.9, καθώς και αυτών που συζητήθηκαν παραπάνω, προτείνεται ο υπολογισμός μίας προσεγγιστικής διαμέσου (*approximate median*) στη θέση της ακριβούς διαμέσου.

3.5.2 Περιγραφή αλγορίθμου προσεγγιστικής διαμέσου

Αρχικά έχουμε $\underline{s} = 1$ και $\bar{s} = n$. Δεν διαλέγουμε μία τιμή για το s^1 , αλλά προτείνουμε την ακόλουθη υπολογιστικά φτηνότερη προσέγγιση. Έστω ότι το s^1

είναι ίσο με τον αριθμό των στοιχείων στο y των οποίων η τιμή είναι μικρότερη ή ίση μίας προσεγγιστικής διαμέσου, y_{appr} . Η y_{appr} θεωρούμε ότι είναι ίση με τη διάμεσο τριών τυχαία επιλεγμένων στοιχείων του διανύσματος τιμών θλάσης (breakpoints), γύρω από τις οποίες διαχωρίζεται το διάνυσμα y (Median-of-Three partitioning cf. [55]).

Δεδομένου του s^1 , γίνεται υπολογισμός του $\lambda(s^1)$ σύμφωνα με την (3.13) και εφόσον ο $\lambda(s^1)$ αποτελεί άνω φράγμα για τον λ^* , απαλείφουμε όλα τα στοιχεία $y_i > \lambda(s^1)$ και ενημερώνουμε το \bar{s} κατάλληλα. Αν $s^1 < s^*$ ενημερώνουμε το κάτω φράγμα στο s^* ($\underline{s} = s^1 + 1$). Όταν $s^1 > s^*$ προτείνουμε την χρήση του αλγορίθμου Bitran-Hax για να ελαχιστοποιήσουμε το μήκος της εναπομείνουσας σειράς. Όταν $s^1 < s^*$ είμαστε ουσιαστικά πίσω στην αρχική κατάσταση (αλλά με μικρότερο διάστημα $\{\underline{s}, \bar{s}\}$), και θα μπορούσαμε να επαναλάβουμε την παραπάνω διαδικασία. Μπορούμε όμως να προχωρήσουμε και κάνοντας χρήση του αλγορίθμου Bitran-Hax επιλέγοντας $s^2 = \bar{s}$. Αυτό πρέπει να είναι υπολογιστικά αποδοτικό όποτε ο $\lambda(s^1)$ αποτελεί ένα καλό άνω φράγμα για τον λ^* . Αυτό ακριβώς προτείνεται, όπως διαφαίνεται και από την παρακάτω λεπτομερή περιγραφή του αλγορίθμου.

Αλγόριθμος 2. Ένα αλγόριθμος προσεγγιστικής διαμέσου

Βήμα 0. (Initialization)

Θέσε $k = 1$, και $F = \{1, \dots, n\}$.

Για όλα τα $i \in F$ θέσε $y_i = h_i/b_i$.

Βήμα 0.5. (Approximate median partitioning)

Επέλεξε y_{appr} , και διαχώρισε το σύνολο F σε δύο νέα σύνολα, F_{\leq} και $F_{>}$, τέτοια ώστε $\max_{i \in F_{\leq}} \{y_i\} = y_{appr} < \min_{i \in F_{>}} \{y_i\}$.

Θέσε $N = b_0 + \sum_{i \in F_{\leq}} y_i \cdot \tilde{g}_i$, και $D = \sum_{i \in F_{\leq}} \tilde{g}_i$, όπου $\tilde{g}_i = b_i^2/g_i$.

Θέσε $\lambda^1 = N/D$.

If $y_{appr} > \lambda^1$ then

Θέσε $F = F_{\leq}$, και $F^1 = \{\}$. Πήγαινε στο Βήμα 2.

Else If $y_i < \lambda^1$ για κάποιο $i \in F_{>}$ then

Θέσε $F = F_{>}$, και $F^1 = F_{\leq}$.

Για όλα τα $i \in F$ Do

If $y_i \geq \lambda^1$ then $F = F \setminus \{i\}$.

End Do

Θέσε $N = N + \sum_{i \in F_{\leq}} y_i \cdot \tilde{g}_i$, και $D = D + \sum_{i \in F_{\leq}} \tilde{g}_i$, όπου $\tilde{g}_i = b_i^2/g_i$.

Θέσε $k = k + 1$. Πήγαινε στο Βήμα 1.

Else

Πήγαινε στο Βήμα 4.

End If

Βήμα 1. (Multiplier calculation)

Θέσε $\lambda^k = N/D$, και $L' = \{\}$.

Βήμα 2. (Pegging)

Για όλα τα $i \in F$ Do

If $y_i \geq \lambda^k$ (που συνεπάγεται ότι $x_i(\lambda^k) \leq 0$) then $L' = L' \cup \{i\}$, $x_i^* = 0$.

End Do

Βήμα 3. (Convergence check)

If $L' = \{\}$ then

Πήγαινε στο Βήμα 4.

Else

Θέσε $F = F \setminus L'$, $N = N - \sum_{i \in L'} y_i \cdot \tilde{g}_i$ και $D = D - \sum_{i \in L'} \tilde{g}_i$.

Θέσε $k = k + 1$. Πήγαινε στο Βήμα 1.

End If

Βήμα 4. (Optimal solution)

Για όλα τα $i \in F^1 \cup F$ Do

$$x_i^* = (\lambda^k b_i - h_i)/g_i$$

End Do

Αν γνωρίζουμε εκ των προτέρων την τιμή του s^* (για παράδειγμα $s^* \leq 0.3n$), αυτή η πληροφορία μπορεί να χρησιμοποιηθεί στο Βήμα 0.5 θέτοντας επιπλέον συνθήκες

στο s^1 . Πρέπει να σημειωθεί ότι τα Βήματα 1-3 στους Αλγορίθμους 1 και 2 είναι πανομοιότυπα και ότι υπάρχει μόνο μία μικρή διαφορά στο Βήμα 4.

3.5.3 Ταξινόμηση με Διαχωρισμό (Partition Sort)

Σε αυτή την υποενότητα θα γίνει περαιτέρω ανάπτυξη της έννοιας της Ταξινόμησης Διαχωρισμού (Partition Sort) που χρησιμοποιήθηκε στον αλγόριθμο προσεγγιστικής διαμέσου για τον διαχωρισμό του συνόλου F σε δύο νέα σύνολα, F_{\leq} και $F_{>}$, τέτοια ώστε $\max_{i \in F_{\leq}} \{y_i\} = y_{appr} < \min_{i \in F_{>}} \{y_i\}$ για ένα κατάλληλα επιλεγμένο y_{appr} . Η διαδικασία μπορεί να περιγραφεί με τον ακόλουθο αλγόριθμο: Επέλεξε με κατάλληλο τρόπο ένα στοιχείο του συνόλου (έστω το στοιχείο y_{appr}). Σάρωσε από τα αριστερά τα στοιχεία του συνόλου έως ότου να βρεθεί ένα στοιχείο $y_i > y_{appr}$ και έπειτα σάρωσε από τα δεξιά τα στοιχεία του συνόλου έως ότου να βρεθεί ένα στοιχείο $y_j < y_{appr}$. Τώρα κάνε ανταλλαγή των θέσεων των δύο αυτών στοιχείων και συνέχισε αυτή την διαδικασία “σάρωσης και ανταλλαγής” έως ότου οι δύο σαρώσεις (εξ αριστερών και εκ δεξιών) συναντηθούν κάπου στη μέση του συνόλου. Αυτό έχει σαν αποτέλεσμα τον διαχωρισμό του συνόλου σε ένα αριστερό τμήμα με στοιχεία τα οποία είναι μικρότερα του y_{appr} (αλλά όχι απαραίτητα ταξινομημένα κατά αύξουσα σειρά) και σε ένα δεξιό τμήμα με στοιχεία τα οποία είναι μεγαλύτερα του y_{appr} (αλλά όχι απαραίτητα ταξινομημένα κατά αύξουσα σειρά). Αυτή η διαδικασία διαχωρισμού είναι δυνατόν να διατυπωθεί στην ακόλουθη μορφή υπορουτίνας

υπορουτίνα: διαχωρισμός

Βήμα 0. (Variable declaration)

Θέσε μεταβλητή διαμέσου y_{appr}

Θέσε προσωρινή μεταβλητή $temp$

Βήμα 1. (Selection of mean and Initialization)

Επέλεξε τυχαία 3 στοιχεία του συνόλου και θέσε ως y_{appr} το στοιχείο του οποίου η τιμή βρίσκεται ανάμεσα στις τιμές των άλλων 2 (Median-Of-Three)

Θέσε $i = 1, j = n$

Βήμα 2. (Partitioning)

Do

While $y_i < y_{appr}$ then Θέσε $i = i + 1$

While $y_{appr} < y_j$ then Θέσε $j = j - 1$

If $i \leq j$ then

 Θέσε $temp = y_i$

 Θέσε $y_i = y_j$

 Θέσε $y_j = temp$

 Θέσε $i = i + 1$

 Θέσε $j = j - 1$

End If

While $i > j$

Σαν ένα παράδειγμα, αν το μεσαίο στοιχείο 42 επιλεγθεί σαν το y_{appr} τότε το σύνολο στοιχείων

44 55 12 42 94 6 18 67

απαιτεί δύο ανταλλαγές για να προκύψει το διαχωρισμένο σύνολο

18 6 12 |42| 94 55 44 67

οι τελικές τιμές δεικτών είναι $i = 5$ και $j = 3$. Τα στοιχεία y_1, \dots, y_{i-1} είναι μικρότερα ή ίσα του στοιχείου $y_{appr} = 42$, ενώ τα στοιχεία y_{j+1}, \dots, y_n είναι μεγαλύτερα ή ίσα του στοιχείου y_{appr} . Συνεπώς, υπάρχουν δύο τμήματα, ήτοι,

$$y_k \leq y_{appr}, \text{ για } k = 1, \dots, i - 1$$

$$y_k \geq y_{appr}, \text{ για } k = j + 1, \dots, n$$

και συνεπακόλουθα,

$$y_k = y_{appr}, \text{ για } k = j + 1, \dots, i - 1$$

Αυτός ο αλγόριθμος είναι αρκετά σαφής και αποδοτικός διότι τα απαραίτητα στοιχεία σύγκρισης y_i, y_j και y_{appr} μπορούν να καταχωρηθούν σε γρήγορα μητρώα (καταλόγους) κατά τη διάρκεια της σάρωσης του συνόλου. Όπως, όμως, μπορεί να φανεί και στην περίπτωση n ίδιων στοιχείων, που έχει να αποτέλεσμα $n/2$ ανταλλαγές, μπορεί να γίνει πολύ αργός. Παρόλα ταύτα, η απλότητα των συνθηκών του αλγορίθμου υπερσχύει του κόστους των επιπλέον επαναλήψεων που, ούτως ή άλλως, συμβαίνουν σχετικά σπάνια στην συνήθη “τυχαία” περίπτωση.

3.6 Θεωρία για τον αλγόριθμο Brucker

Όπως αναφέρθηκε και παραπάνω, ο Brucker, [16], υπολογίζει τη διάμεσο (median) ενός μειούμενου συνόλου τιμών θλάσης (breakpoints) σε κάθε επανάληψη k , θέτοντας $\lambda^k = h_s/b_s$, για κάποιο $s \in \{1, \dots, n\}$. Βάσει του προσήμου του $b(\lambda^k)$ μειώνεται το σύνολο δεικτών στο οποίο πρέπει να βρίσκεται το s^* . Τέλος, ο προσδιορισμός του λ^* είναι απλός, αφού έχει βρεθεί το s^* . Έτσι, προκύπτει ένας αλγόριθμος πολυπλοκότητας $O(n)$.

Έστω το ακόλουθο τετραγωνικό πρόγραμμα

$$\min \quad \sum_{i=1}^n \frac{1}{2} g_i x_i^2 - h_i x_i \quad (3.17)$$

$$\text{υπό} \quad \sum_{i=1}^n b_i x_i = b_0, \quad (3.18)$$

$$u_i \geq x_i \geq 0, i = 1, \dots, n \quad (3.19)$$

3.6.1 Μία παραμετρική προσέγγιση

Για κάθε τιμή της παραμέτρου $t \in \mathbb{R}$ θεωρούμε το πρόβλημα

$$\begin{aligned} P(t) \min \quad & \sum_{i=1}^n \frac{1}{2} g_i x_i^2 + (b_i t - h_i) x_i \\ \text{υπό} \quad & u_i \geq x_i \geq 0, i = 1, \dots, n. \end{aligned} \quad (3.20)$$

Το παραπάνω πρόβλημα έχει την μοναδική λύση $x(t)$

$$x_i(t) = \begin{cases} 0, & \text{εάν } (h_i - b_i t)/g_i \leq 0, \\ u_i & \text{εάν } (h_i - b_i t)/g_i \geq u_i, \\ (h_i - b_i t)/g_i & \text{αλλιώς.} \end{cases} \quad (3.21)$$

Έστω ότι $z : \mathbb{R} \rightarrow \mathbb{R}$ είναι η συνάρτηση που ορίζεται από

$$z(t) = \sum_{i=1}^n b_i x_i(t). \quad (3.22)$$

Αν ορίσουμε τις κρίσιμες παραμέτρους $t_i^U \leq t_i^L$ ως $t_i^L = h_i/b_i$ και $t_i^U = (h_i - u_i g_i)/b_i$ για $i = 1, \dots, n$ και έστω ότι οι $t_1 < t_2 < \dots < t_r$ αποτελούν διακριτές τιμές των κρίσιμων παραμέτρων. Επίσης, έστω ότι $t_0 = -\infty$, $t_{r+1} = +\infty$. Τότε η δομή του $x(t)$ δεν αλλάζει μέσα σε κάθε διάστημα (t_i, t_{i+1}) , $i = 0, \dots, r$. Το ακόλουθο θεώρημα δηλώνει ορισμένες χρήσιμες ιδιότητες της συνάρτησης z .

Θεώρημα 3.10. Η συνάρτηση $z : \mathbb{R} \rightarrow \mathbb{R}$ που ορίζεται από την (3.22) διαθέτει τις ακόλουθες ιδιότητες:

- (a) Η z είναι μονοτονικά φθίνουσα.
- (b) Εάν $z(t) = b_0$ τότε η $x(t)$ αποτελεί βέλτιστη λύση του προβλήματος.
- (c) Εάν $z(t) > b_0$ ή $z(t) < b_0$ τότε $\forall t \in \mathbb{R}$ δεν υπάρχει εφικτή λύση του προβλήματος.

Απόδειξη. Γράφουμε

$$f(x) = \sum_{i=1}^n (\frac{1}{2} g_i x_i^2 - h_i x_i), \quad bx = \sum_{i=1}^n b_i x_i$$

και

$$f(x, t) = f(x) + tbx.$$

- (a) Έστω ότι $t \in \mathbb{R}$, $t^* = t + \Delta t$ όπου $\Delta t > 0$, και $z(t) = bx(t) < bx(t^*) = z(t^*)$.

Τότε, η βελτιστότητα του $x = x(t)$ ($x^* = x(t^*)$) για το $P(t)$ ($P(t^*)$) υποδηλώνει ότι

$$f(x) + tbx \leq f(x^*) + tbx^*$$

$$(f(x^*) + (t + \Delta t)bx^* \leq f(x) + (t + \Delta t)bx).$$

Έτσι προκύπτει η αντίφαση

$$f(x^*) + (t + \Delta t)bx^* \leq f(x) + (t + \Delta t)bx < f(x^*) + (t + \Delta t)bx^*.$$

(b) Έστω ότι x' είναι μία αυθαίρετη εφικτή λύση του προβλήματος. Τότε η x' αποτελεί εφικτή λύση για το $P(t)$ και η βελτιστότητα της $x(t)$ για το $P(t)$ υποδηλώνει ότι

$$f(x(t)) + bx(t) \leq f(x') + bx'$$

το οποίο σημαίνει ότι $f(x(t)) \leq f(x')$ διότι $b(x(t)) = b(x') = b_0$.

(c) Έστω ότι ισχύει $z(t) = bx(t) > b_0$, $\forall t \in \mathbb{R}$ και ότι το πρόβλημα διαθέτει μία εφικτή λύση x . Τότε $bx = b_0$. Επιπροσθέτως, υπάρχει μία τιμή παραμέτρου t^* τέτοια ώστε η $x(t^*)$ να είναι βέλτιστη για όλα τα $P(t)$ εφόσον $t \geq t^*$ (δες (3.21)). Έτσι, για όλα τα $t \geq t^*$ έχουμε

$$f(x(t^*)) + tbx(t^*) \leq f(x) + tbx,$$

$$f(x(t^*)) - f(x) \leq t(bx - bx(t^*)) = t(b_0 - bx(t^*))$$

το οποίο για μεγάλα $t \geq t^*$ οδηγεί σε αντίφαση διότι $b_0 - bx(t^*) < 0$.

Αν αντί για $z(t) > b_0$ υποθέσουμε ότι $z(t) < b_0$, $\forall t \in \mathbb{R}$, το επιθυμητό αποτέλεσμα προκύπτει κατά όμοιο τρόπο.

Αν $z(t_1) \geq b_0$ και $z(t_r) \leq b_0$ τότε υπάρχει ένας δείκτης j ώστε να ισχύει $z(t_j) \geq b_0$ και $z(t_{j+1}) \leq b_0$. Η κατασκευή μίας βέλτιστης λύσης είναι δυνατή με τη χρήση αυτού του δείκτη j ως ακολούθως. Ορίζουμε τα σύνολα δεικτών

$$I_L = \{i | t_i^L \leq t_j\},$$

$$I_U = \{i | t_{j+1} \leq t_i^U\},$$

$$I_M = \{i | t_i^L > t_j; t_{j+1} > t_i^U\}.$$

Τότε, για κάθε $t \in [t_j, t_{j+1}]$ τα στοιχεία της λύσης $x(t)$ μπορούν να εκφραστούν ως εξής

$$x_i(t) = \begin{cases} 0, & \text{εάν } i \in I_L, \\ u_i & \text{εάν } i \in I_U, \\ (h_i - b_i t)/g_i & \text{εάν } i \in I_M. \end{cases} \quad (3.23)$$

Η βέλτιστη λύση $x(t_{opt})$ του προβλήματος προκύπτει υποκαθιστώντας στην (3.23) την παραμετρική τιμή

$$t_{opt} = \left(\sum_{i \in I_U} b_i u_i + \sum_{i \in I_M} \left(\frac{b_i h_i}{g_i} \right) - b_0 \right) / \sum_{i \in I_M} \frac{b_i^2}{g_i}. \quad (3.24)$$

Πρέπει να σημειωθεί ότι η t_{opt} αποτελεί λύση της εξίσωσης $bx(t) = b_0$.

Λόγω της μονοτονικότητας της συνάρτησης z , ο δείκτης j μπορεί να βρεθεί μέσω δυαδικής αναζήτησης του συνόλου δεικτών των κρίσιμων τιμών.

3.7 Ο αλγόριθμος Brucker

Η κύρια ιδέα του αλγορίθμου περιγράφεται ως ακολούθως. Έστω ότι $z(t_1) \geq b_0$ και $z(t_r) \leq b_0$. Αν θέσουμε $t_{min} = t_1$ και $t_{max} = t_r$, τότε μπορούμε να είμαστε σίγουροι ότι ισχύει η ιδιότητα

$$t_{opt} \in [t_{min}, t_{max}] \quad (3.25)$$

Σε κάθε επανάληψη του αλγορίθμου φροντίζουμε να ικανοποιείται η ιδιότητα (3.25) και μειώνουμε σταδιακά το διάστημα $[t_{min}, t_{max}]$ έως ότου φθάσει στην μορφή $[t_j, t_{j+1}]$. Αυτό επιτυγχάνεται επιλύοντας το P(t) για διαφορετικές τιμές $t = t_i$. Έστω ότι $t_{min} < t_i < t_{max}$. Αν $z(t_i) > b_0$ τότε μπορεί να γίνει αντικατάσταση του t_{min} με t_i . Άλλως, αν $z(t_i) < b_0$ τότε το t_{max} αντικαθίσταται από το t_i . Αν, τέλος, $z(t_i) = b_0$ θα έχουμε $t_{opt} = t_i$.

Το σημαντικό σημείο είναι ότι ταυτόχρονα, τα σχετικά σύνολα κρίσιμων παραμέτρων

$$T^U = \{t_i^U | i = 1, \dots, n\} \quad (T^L = \{t_i^L | i = 1, \dots, n\})$$

μειώνονται απαλείφοντας τα άκρα των διαστημάτων $[t_i^U, t_i^L]$. Επιλύοντας το πολύ δύο από τα προβλήματα $P(t^L)$ και $P(t^U)$, κάθε σύνολο από τα T^U, T^L μειώνεται κατά τουλάχιστον $\lceil \frac{1}{4} |T^U| \rceil$ ($\lceil \frac{1}{4} |T^L| \rceil$) στοιχεία. Επιπρόσθετα, όταν απαλείφουμε τα t_i^U, t_i^L , γνωρίζουμε την δομή των $x_i(t_{opt})$ (ήτοι, γνωρίζουμε εάν $x_i(t_{opt}) = u_i$ ή $x_i(t_{opt}) = 0$ ή $x_i(t_{opt}) = (h_i - b_i t_{opt})/g_i$). Έτσι η πολυπλοκότητα για την επίλυση των προβλημάτων $P(t)$ μειώνεται αναλόγως. Για την συνολική πολυπλοκότητα του αλγορίθμου παίρνουμε το άνω φράγμα

$$cn + \frac{3}{4}cn + (\frac{3}{4})^2cn + \dots = 4cn = O(n)$$

όπου c είναι κάποια σταθερά. Επιλέγουμε τα t^U, t^L κατά τέτοιο τρόπο ώστε

$$t^L = \text{διάμεσος}(T^L) \text{ και } t^U = \text{διάμεσος}(\{t_i^U | t_i^L \in T^L; t_i^L \geq t^L\})$$

όπου ως διάμεσος(S) συμβολίζεται η διάμεσος ενός συνόλου S .

Το $P(t^L)$ ($P(t^U)$) επιλύεται μόνο όταν $t_{min} < t^L < t_{max}$ ($t_{min} < t^U < t_{max}$) και γίνεται ανανεώση των τιμών των t_{min} και t_{max} . Ένα τέταρτο των συνόλων T^U και T^L μπορεί να απαλειφθεί και η δομή των $x_i(t)$ θα είναι γνωστή για τις αντίστοιχες μεταβλητές για κάθε $t \in [t_{min}, t_{max}]$. Για αυτό πρέπει να θεωρήσουμε τρεις περιπτώσεις.

Περίπτωση 1: $t^L \leq t_{min}$. Τότε $t_i^L \leq t_{min} \leq t_{opt}$ για τουλάχιστον $\lceil \frac{1}{2} |T^L| \rceil$ διαστήματα $[t_i^U, t_i^L]$. Για αυτά τα i μπορούμε να απαλείψουμε το t_i^U (t_i^L) από το σύνολο T^U (T^L) και να θέσουμε $x_i(t) = 0, \forall t \in [t_{min}, t_{max}]$.

Περίπτωση 2: $t^L > t_{min}$ και $z(t^U) \leq b_0$ (ήτοι, $t^U \geq t_{max}$). Τότε $t_{opt} \leq t_{max} \leq t_i^U$ και κατασκευάζοντας το t^U για τουλάχιστον $\lceil \frac{1}{4} |T^L| \rceil$ διαστήματα $[t_i^U, t_i^L]$ θα έχουμε $t_{opt} \leq t^U \leq t_i^U$. Για αυτά τα i μπορούμε να απαλείψουμε το t_i^U (t_i^L) από το σύνολο T^U (T^L) και να θέσουμε $x_i(t) = u_i, \forall t \in [t_{min}, t_{max}]$.

Περίπτωση 3: $t^L > t_{min}$ και $z(t^U) > b_0$. Τότε $t^U < t_{opt}$ και $t_{opt} \leq t^L$ διότι διαφορετικά θα ίσχυε $t_{opt} > t^L > t_{min}$ που αντιτίθεται στο γεγονός ότι το t_{min} ανανεώθηκε κατά την επίλυση του $P(t^L)$. Τώρα η σχέση $t^U \leq t_{opt} \leq t^L$ υποδηλώνει ότι για τουλάχιστον $\lceil \frac{1}{4} |T^L| \rceil$ διαστήματα $[t_i^U, t_i^L]$ θα έχουμε $t_i^U \leq t_{min} \leq t_{opt} \leq t_{max} \leq t_i^L$. Για αυτά τα i μπορούμε να απαλείψουμε το t_i^U (t_i^L) από το σύνολο T^U (T^L) και να θέσουμε $x_i(t) = (h_i - b_i t)/g_i, \forall t \in [t_{min}, t_{max}]$.

Λεπτομέρειες της διαδικασίας περιγράφονται από τον ακόλουθο αλγόριθμο. Πριν την εφαρμογή του αλγορίθμου υπολογίζονται όλες οι κρίσιμες τιμές t_i^U, t_i^L . Υποθέτουμε ότι αμέσως μετά της χρήσης της $z(t)$ γίνεται επίλυση του $P(t)$. Επίσης, το σύνολο I συμβολίζει το σύνολο των δεικτών των μεταβλητών $x_i(t)$ που δεν έχουν πάρει σταθερή τιμή. Πρέπει να σημειωθεί ότι τα $T^U = \{t_i^U | i \in I\}$ και $T^L = \{t_i^L | i \in I\}$ είναι τα τρέχοντα σύνολα των τιμών των παραμέτρων.

Αλγόριθμος 3. Ο αλγόριθμος Brucker

Βήμα 0. (Optimality Check)

If $z(t_1) = b_0$ OR $z(t_r) = b_0$ then

Θέσε $x_i = (h_i - b_i t_{1ORr}) / g_i, \forall i$. (Βέλτιστη Λύση)

Else If $z(t_1) < b_0$ OR $z(t_r) > b_0$ then

Έξοδος (Δεν υφίσταται Εφικτή Λύση)

End If

Βήμα 1. (Initialization)

Θέσε $t_{min} = t_1, t_{max} = t_r$ και $I = \{1, \dots, n\}$

Βήμα 2. (Median Selection)

Για όλα τα $i \in I$ Do

Θέσε $t^L = \text{διάμεσος}(\{t_i^L | i \in I\})$

Θέσε $t^U = \text{διάμεσος}(\{t_i^U | i \in I; t_i^L \geq t^L\})$

Για $t = t^L, t^U$ If $t_{min} < t < t_{max}$ then

Do

If $z(t) = b_0$ then

Θέσε $x_i = (h_i - b_i t) / g_i, \forall i$. (Βέλτιστη Λύση)

Else If $z(t) > b_0$ then

$t_{min} = \max\{t_{min}, t\}$

Else

$t_{max} = \min\{t_{max}, t\}$

End If

End Do

Για όλα τα $i \in I$ Do

If $t_i^L \leq t_{min}$ then

Θέσε $I = I \setminus \{i\}$ και $x_i = 0$

End If

If $t_{max} \leq t_i^U$ then

Θέσε $I = I \setminus \{i\}$ και $x_i = u_i$

End If

If $t_i^U \leq t_{min} \leq t_{max} \leq t_i^L$ then

Θέσε $I = I \setminus \{i\}$ και $x_i = (h_i - b_i t) / g_i$

End If

End Do

End Do

Βήμα 3. (Convergence check)

If $I = \{\}$ then

Πήγαινε στο Βήμα 4

Else

Πήγαινε στο Βήμα 2.

End If

Βήμα 4. (Optimal solution)

Για όλα τα i Do

If $x_i = u_i$ then

Θέσε μία προσωρινή μεταβλητή $t_{temp} = 0$

Θέσε $t_{temp} = \sum b_i u_i$

End If

If $x_i \neq 0$ OR $x_i \neq u_i$ then

Θέσε $t_{opt} = \sum \left(\frac{b_i h_i}{g_i} \right) / \sum \frac{b_i^2}{g_i}$

$$\text{Θέσε } t_{opt} = t_{opt} + \left((t_{temp} - b_0) / \sum \frac{b_i^2}{g_i} \right)$$

End If

End Do

Για όλα τα i Do

If $x_i \neq 0$ OR $x_i \neq u_i$ then

$$\text{Θέσε } x_i = (h_i - b_i t_{opt}) / g_i$$

End If

End Do

Σε κάθε Βήμα του αλγορίθμου, ο υπολογισμός του $z(t)$ γίνεται ως ακολούθως

$$z(t) = \sum_{i \in I} b_i x_i(t) + \sum_{i \in I^U} b_i u_i + \sum_{i \in I^M} h_i b_i / g_i - \sum_{i \in I^M} (b_i^2 / g_i) t, \quad \forall t \in [t_{min}, t_{max}].$$

Όπως φαίνεται και στο Βήμα 4, για την εύρεση της βέλτιστης λύσης έγινε χρήση της εξίσωσης (3.24) ώστε πρώτα να βρεθεί μία βέλτιστη τιμή για την παράμετρο $t = t_{opt}$.

Πρέπει να σημειωθεί ότι, λόγω του γεγονότος ότι οι προηγούμενοι δύο αλγόριθμοι επιλύουν **κάτω φραγμένα** τετραγωνικά προβλήματα σακιδίου τα οποία υπόκεινται σε έναν περιορισμό, έγινε τροποποίηση του άνωθεν αλγορίθμου ώστε και αυτός να επιλύει **μόνο κάτω φραγμένα** τετραγωνικά προβλήματα σακιδίου με έναν περιορισμό, ήτοι, οι σχέσεις, οι περιορισμοί και οι συνθήκες που εμπεριέχουν τα άνω φράγματα u_i αγνοούνται παντελώς.

Κεφάλαιο 4

Αποτελέσματα

4.1 Αποτελέσματα

Για μία σχολαστική υπολογιστική εξέταση αλγορίθμων για το πρόβλημα (3.1) με άνω και κάτω φράγματα πρέπει να γίνει αναφορά στους [54]. Η υπολογιστική εργασία έγινε σε έναν προσωπικό υπολογιστή με επεξεργαστή Athlon64 +3400. Η υλοποίηση των ανωτέρω αλγορίθμων έγινε σε γλώσσα προγραμματισμού C και το λειτουργικό που χρησιμοποιήθηκε ήταν το GNU/Linux Ubuntu. Τα προγράμματα έτρεξαν στον μεταγλωττιστή (compiler) gcc. Τα προβλήματα που χρησιμοποιήθηκαν για την σύγκριση των αλγορίθμων προήλθαν από γεννήτρια τυχαίων αριθμών. Η εν λόγω γεννήτρια ήταν μία γραμμική πολλαπλασιαστική γεννήτρια υπολοίπων (linear multiplicative congruential generator), η οποία βασίζεται στην φορητή γεννήτρια τυχαίων αριθμών UNIRAN των Marse και Roberts (Η υλοποίησή της σε γλώσσα προγραμματισμού C έγινε από τον καθηγητή Β. Κουικόγλου στα πλαίσια του προπτυχιακού μαθήματος Προσομοίωση). Πιο συγκεκριμένα, οι συντελεστές των προβλημάτων είναι ομοιόμορφα κατανομημένοι στο διάστημα $[0, 1]$. Για την τιμή του b_0 το αντίστοιχο διάστημα ήταν $[0, n]$, όπου n το μέγεθος του προβλήματος. Για την σύγκριση των τριών αλγορίθμων χρησιμοποιήθηκε η εντολή του UNIX, `time filename.executable` η οποία επιστρέφει τρεις χρόνους: τους `real`, `user`, `sys` που αφορούν το εκάστοτε εκτελέσιμο αρχείο. Το άθροισμα των χρόνων `user` και `sys` μας δίνει το λεγόμενο CPU Time το οποίο μετράται σε δευτερόλεπτα. Για κάθε αλγόριθμο τέθηκαν δέκα διαφορετικές αρχικές τιμές (seeds) για την γεν-

νήτρια τυχαίων αριθμών, με αποτέλεσμα την δημιουργία δέκα τυχαία παραχθέντων προβλημάτων με μέγεθος $n = 1000$. Επίσης, για κάθε αρχική τιμή έγιναν δέκα επαναλήψεις της εντολής time. Στον πίνακα που ακολουθεί παρατίθενται οι μέσες τιμές των CPU Time για τις δέκα συνολικά επαναλήψεις για κάθε πρόβλημα.

	Μέσος CPU Time (s)		
Πρόβλημα	Bitran-Hax	Αλγ. Πρ. Διαμέσου	Brucker
1.	0.0072	0.0136	0.1136
2.	0.0056	0.0168	0.0792
3.	0.0056	0.0128	0.0712
4.	0.0096	0.0288	0.1664
5.	0.0128	0.0280	0.1640
6.	0.0096	0.0296	0.1640
7.	0.0120	0.0312	0.1808
8.	0.0112	0.0296	0.1752
9.	0.0120	0.0336	0.1760
10.	0.0096	0.0344	0.1804

Όπως φαίνεται και από τα δεδομένα του παραπάνω πίνακα, ο τροποποιημένος αλγόριθμος Bitran-Hax έχει τους πιο γρήγορους χρόνους επίλυσης, κάτι που συνάδει και με τα θεωρητικά δεδομένα, εφόσον ο εν λόγω αλγόριθμος ελαχιστοποιεί τον φόρτο λόγω διαχωρισμού ή στην γενικότερη περίπτωση, ταξινόμησης. Αντίθετα, ο τροποποιημένος αλγόριθμος Brucker έχει τους μεγαλύτερους χρόνους επίλυσης ανάμεσα στους τρεις αλγόριθμους, κάτι που επίσης συνάδει με τα θεωρητικά δεδομένα, εφόσον προκύπτει μεγάλος υπολογιστικός φόρτος κατά την εύρεση μίας διαμέσου του συνεχώς μειούμενου συνόλου. Τέλος, ο αλγόριθμος προσεγγιστικής διαμέσου είναι όπως φαίνεται 5 με 8 φορές γρηγορότερος, όσον αφορά τους χρόνους επίλυσης, σε σχέση με τον αλγόριθμο Brucker, αλλά κατά 2 με 4 φορές πιο αργός από τον αλγόριθμο Bitran-Hax. Αυτό, επίσης, επιβεβαιώνεται από τη θεωρία που αναπτύχθηκε στο Κεφάλαιο 3.

4.2 Συμπεράσματα

Λόγω του γεγονότος ότι το πρόβλημα (3.1) αποτελεί κοινό υποπρόβλημα σε διάφορα σχήματα αποσύνδεσης (decomposition schemes), όπως επίσης και σε σχέση

με προβλήματα βελτιστοποίησης υποκλίτου (subgradient optimization), οποιαδήποτε βελτίωση στην απόδοση των υπαρχόντων αλγορίθμων είναι πολύ σημαντική. Όπως φαίνεται και από τα υπολογιστικά αποτελέσματα, ο αλγόριθμος Brucker έχει την χειρότερη απόδοση στα συγκεκριμένα προβλήματα που εξετάστηκαν, ενώ ο αλγόριθμος Bitran-Hax αποδίδει καλύτερα από τους άλλους δύο αλγορίθμους. Πρέπει, όμως, να σημειωθεί ότι αυτό μπορεί να είναι συνέπεια του τρόπου επιλογής των συγκεκριμένων τυχαία παραχθέντων προβλημάτων και ότι σε διαφορετικές περιπτώσεις προβλημάτων, (δες [48]), υπολογιστικά δεδομένα έχουν δείξει ότι ο αλγόριθμος προσεγγιστικής διαμέσου αποδίδει καλύτερα από τον αλγόριθμο Bitran-Hax. Τέλος, πρέπει να παρατηρηθεί ότι, εκτός της Χαλάρωσης Lagrange, μπορούν να χρησιμοποιηθούν και άλλες προσεγγίσεις για την επίλυση του τετραγωνικού προβλήματος σακιδίου. Για παράδειγμα, μπορεί να γίνει χρήση Υπερχαλάρωσης (*Over-relaxation*), ήτοι, στην επανάληψη k , αντί να γίνεται χρήση του λ^k σαν στοιχείο διαχωρισμού (partitioning element) όπως φαίνεται στο Βήμα 3 του Αλγορίθμου 2, να γίνει χρήση του $\lambda_{over}^k = f_{over} \cdot \underline{\lambda} + (1 - f_{over}) \cdot \lambda^k$, όπου $f_{over} \in [0, 1]$ μία παράμετρος χαλάρωσης και $\underline{\lambda}$ ένα κάτω φράγμα στο λ^* . Ενώ η εν λόγω προσέγγιση δίδει μεγαλύτερο μήκος σειράς $L(\{s^k\})$, ο αριθμός των επαναλήψεων μειώνεται αρκετά σε σχέση με τον αλγόριθμο προσεγγιστικής διαμέσου.

Βιβλιογραφία

- [1] W.P. Adams and H.D. Sherali (1986), *A Tight Linearization and an Algorithm for Zero-One Quadratic Programming Problems*, Management Science 32, 1274-1290.
- [2] P.B. Afentakis, B. Gavish and U. Karmarkar (1984), *Computationally efficient optimal solutions to the lot-sizing problem in multistage assembly systems*, Management Science 30, 222-239.
- [3] A. Aho, J. Hopcroft and J. Ullman (1974), *The design and analysis of computer algorithms*, Addison-Wesley, Reading MA.
- [4] R.K. Ahuja, T.L. Magnanti and J.B. Orlin (1993), *Network Flows*, Prentice-Hall, N.J.
- [5] E. Balas, S. Ceria and G. Cornuejols (1993), *A Lift-and-Project Cutting Plane Algorithm for Mixed 0-1 Programs*, Mathematical Programming 58, 295-324.
- [6] E. Balas, S. Ceria, G. Cornuejols and G. Pataki (1996), *Polyhedral Methods for the Maximum Clique Problem*, in D.S. Johnson and M.A. Trick (eds.) *Cliques, Colouring and Satisfiability: Second DIMACS Implementation Challenge*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS Press.
- [7] E. Balas and E. Zemel (1980), *An Algorithm for Large Zero-One Knapsack Problems*, Operations Research, 28, 1130-1154.
- [8] M.S. Bazaraa and C.M. Shetty (1979), *Nonlinear Programming: Theory and Algorithms*, John Wiley and Sons, N.Y.
- [9] M. Bellare, O. Goldreich and M. Sudan (1995), *Free Bits, PCPs and Non-Approximability - Towards Tight Results*, Proceedings of the 36th FOCS Conference, 422-431, IEEE Computer Society Press.
- [10] D.P. Bertsekas, A. Nedic and A.E. Ozdaglar (2003), *Convex Analysis and Optimization*, Athena Scientific, Belmont, Mass.

- [11] A. Billionet and F. Calmels (1996), *Linear Programming for the 0-1 Quadratic Knapsack Problem*, European Journal of Operational Research 92, 310-325.
- [12] A. Billionet, A. Faye and E. Soutif (1997), *A New Upper-Bound and an Exact Algorithm for the 0-1 Quadratic Knapsack Problem*, presented at ISMP'97, Lausanne, EPFL, Augus 24-29, 1997.
- [13] G.R. Bitran and A.C. Hax (1981), *Disaggregation and resource allocation using convex knapsack problems with bounded variables*, Management Science 27, 431-441.
- [14] U. Brannlund (1993), *On Relaxation Methods for Nonsmooth Convex Optimization*, PhD Thesis, Department of Mathematics, Royal Institute of Technology, Stockholm.
- [15] K.M. Bretthauer, B. Shetty and S. Syam (1995), *A Branch-and-Bound Algorithm for Integer Quadratic Knapsack Problems*, ORSA Journal on Computing 7, 109-116.
- [16] P. Brucker (1984), *An $O(n)$ algorithm for quadratic knapsack problems*, Operations Research Letters, 163-166.
- [17] A. Caprara, D. Pisinger and P. Toth (1999), *Exact Solution of the Quadratic Knapsack Problem*, Informs Journal on Computing, 11(2), 125-137.
- [18] P. Chailou, P. Hansen and Y. Mahieu (1986), *Best Network Flow Bounds for the Quadratic Knapsack Problem*, Lecture Notes in Mathematics 1403, 226-235.
- [19] P. Carraresi and F. Malucelli (1994), *A Reformulation Scheme and New Lower Bounds for the QAP*, in P.M. Pardalos and H. Wolkowicz (eds.), Quadratic Assignment and Related Problems, DIMACS series in Discrete Mathematics and Theoretical Computer Science, 147-160, AMS Press.
- [20] S.C. Dafermos and F.T. Sparrow (1969), *The traffic assignment problem for a general network*, Journal of Research of the National Bureau of Standards 73B(2), 91-118.
- [21] G. Dijkhuizen and U. Faigle (1993), *A Cutting-Plane Approach to the Edge-Weighted Maximal Clique Problem*, European Journal of Operational Research 69, 121-130.
- [22] H. Everett (1963), *Generalized Lagrange multiplier method for solving problems of optimum allocation of resources*, Operations Research 11, 399-417.
- [23] M.L. Fisher (1981), *The Lagrangian Relaxation Method for Solving Integer Programming Problems*, Management Science 27, 1-18.
- [24] G. Gallo, P.L.Hammer and B. Simeone (1980), *Quadratic Knapsack Problems*, Mathematical Programming 12, 132-149.

- [25] A.M. Geoffrion (1970), *Elements of large-scale mathematical programming*, Management Science 16, 652-691.
- [26] A.M. Geoffrion (1974), *Lagrangian relaxation for integer programming*, Mathematical Programming Study 2, 82-114.
- [27] A.M. Geoffrion and R. Marsten (1972), *Integer programming: A framework and the state-of-the-art survey*, Management Science 18, 465-491.
- [28] P.L. Hammer and D.J. Rader, Jr. (1997), *Efficient Methods for Solving Quadratic 0-1 Knapsack Problems*, INFOR 35, 170-182.
- [29] M. Held and R. M. Karp (1970), *The Traveling Salesman Problem and Minimum Spanning Trees*, Operations Research 18, 1138-1162.
- [30] M. Held and R. M. Karp (1971), *The Traveling Salesman Problem and Minimum Spanning Trees: Part II*, Mathematical Programming 1, 6-25.
- [31] M. Held, P. Wolfe and H.P. Crowder (1974), *Validation of Subgradient Optimization*, Mathematical Programming 6, 62-88.
- [32] R. Helgason, J. Kennington and H. Lall (1980), *A polynomially bounded algorithm for a singly constrained quadratic program*, Mathematical Programming 18, 338-343.
- [33] C. Helmberg, F. Rendl and R. Weismantel (1996), *Quadratic Knapsack Relaxations Using Cutting Planes and Semidefinite Programming*, in W.H. Cunningham, S.T. McCormick and M. Queyranne (eds.), Proceedings of the Fifth IPCO Conference, Lecture Notes in Computer Science 1084, 175-189, Springer Verlag.
- [34] J.B. Hiriart-Urruty and C. Lemarechal (1991), *Convex Analysis and Minimization Algorithms*, Volumes I and II, Springer-Verlag, Berlin.
- [35] E.L. Johnson, A. Mehrotra and G.L. Nemhauser (1993), *Min-Cut Clustering*, Mathematical Programming 62, 133-152.
- [36] D.S. Johnson and M.A. Trick (eds.) (1996), *Cliques, Colouring and Satisfiability: Second DIMACS Implementation Challenge*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS Press.
- [37] L.S. Lasdon (1970), *Optimization Theory for Large Systems*, MacMillan Publishing Co., N.Y.
- [38] L. Lovasz and A. Schrijver (1991), *Cones of Matrices and Set-Functions and 0-1 Optimization*, SIAM Journal on Optimization 1, 166-190.
- [39] H. Luss and S.K. Gupta (1975), *Allocation of effort resources among competing activities*, Operations Research 23, 360-366.

- [40] Y. Marinakis, A. Migdalas and P.M. Pardalos (2005), *A hybrid genetic-GRASP algorithm using Lagrangean relaxation for the traveling salesman problem*, Journal of Combinatorial Optimization 10, 311-326.
- [41] S. Martello and P. Toth (1990), *Knapsack Problem: Algorithms and Computer Implementations*, Wiley, Chichester, England.
- [42] S. Martello, D. Pisinger and P. Toth (1999), *Dynamic Programming and Strong Bounds for the 0-1 Knapsack Problem*, Management Science 45(3), 414-424.
- [43] R.K. Martin (1999), *Large Scale Linear and Integer Optimization: A Unified Approach*, Kluwer Academic Publishers, Boston.
- [44] P. Michelon and N. Maculan (1993), *Lagrangean Methods for 0-1 Quadratic Programming*, Discrete Applied Mathematics 42, 257-269.
- [45] P. Michelon and L. Veilleux (1996), *Lagrangean Methods for the 0-1 Quadratic Knapsack Problem*, European Journal of Operational Research 92, 326-341.
- [46] A. Migdalas and M. Gothe-Lundgren (1994), *Kombinatorisk Optimering: Problem, Algoritmer och Komplexitet*, LiTH, Sweden.
- [47] A. Nagurney (1993), *Network Economics*, Kluwer Academic Publishers, N.Y.
- [48] A. Nou (1997), *An algorithm for a singly constrained quadratic program subject to lower bounds*, Technical Report TRITA/MAT-97-OS2, Department of Mathematics, KTH, Stockholm, Sweden.
- [49] P.M. Pardalos and N. Kovoov (1990), *An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds*, Mathematical Programming 46, 321-328.
- [50] K. Park, K. Lee, S. Park (1996), *An Extended Formulation Approach to the Edge-Weighted Maximal Clique Problem*, European Journal of Operational Research 95, 671-682.
- [51] B.T. Polyak (1967), *A general method for solving extremal problems*, Soviet Math. Doklady 8, 593-597.
- [52] B.T. Polyak (1969), *Minimization of unsmooth functions*, USSR Comp. Math. and Math. Physics 9, 509-521.
- [53] J. Rhys (1970), *A Selection Problem of Shared Fixed Costs and Network Flows*, Management Science 17, 200-207.
- [54] A.G. Robinson, N. Jiang and C.S. Lerne (1992), *On the quadratic knapsack problem*, Mathematical Programming 55(1), 99-108.

- [55] R. Sedgewick (1998), *Algorithms*, Addison-Wesley, Boston.
- [56] J.F. Shapiro (1979), *Mathematical Programming: Structures and Algorithms*, John Wiley and Sons, N.Y.
- [57] K. Shimizu, Y. Ishizuka and J.F. Bard (1997), *Non-Differentiable and Two-Level Mathematical Programming*, Kluwer Academic Publishers, Boston.
- [58] N.Z. Shor (1985), *Minimization Methods for Non-Differentiable Functions*, Springer-Verlag, Berlin.
- [59] N. Wirth (1976), *Algorithms+Data Structures=Programs*, Prentice-Hall, N.J.
- [60] C. Witzgall (1975), *Mathematical Methods of Site Selection for Electronic Message Systems (EMS)*, NBS internal report.