Διάλεξη 4

Διαμόρφωση γραμμικών και ακέραιων γραμμικών προγραμμάτων, Πέμπτη 1/3/18

Παράδειγμα 4: Το πρόβλημα της συσκευασίας κιβωτίων

Θεωρείστε ότι μετακομίζετε και θα πρέπει να συσκευάσετε όλα τα προσωπικά σας αντικείμενα σε όσο το δυνατόν λιγότερα κουτιά. (Πχ, η εταιρία που έχει αναλάβει τη μεταφορά σας χρεώνει ανάλογα με τον αριθμό των κουτιών.)

Συγκεκριμένα θεωρείστε ότι έχετε n αντικείμενα όπου το i-οστό καταλαμβάνει όγκο v_i για $i = 1, \ldots, n$. Θεωρείστε ότι έχετε στη διάθεσή σας m κιβώτια διαφορετικής χωρητικότητας σε όγκο το καθένα, όπου το j-οστό χωράει V_j μονάδες όγκου για $j = 1, \ldots, m$. Επιπλέον θεωρείστε ότι ο αριθμός των κιβωτίων δεν είναι περιοριστικός, δηλαδή έχετε αρκετά κιβώτια για να χωρέσετε όλα τα αντικείμενα. Το πρόβλημα αυτό είναι γνωστό ως **bin packing** (συσκευασία σε κιβώτια) και μπορεί να διαμορφωθεί ως πρόβλημα ακέραιου προγραμματισμού ως εξής:

$$\min \sum_{j=1}^{m} f_j$$

έτσι ώστε $\sum_{j=1}^{m} x_{ij} = 1$ για κάθε $i = 1, \ldots, n$

$\sum_{i=1}^{n} x_{ij} \leq M f_j$ για κάθε $j = 1, \ldots, m$

$\sum_{i=1}^{n} v_i x_{ij} \leq V_j$ για κάθε $j = 1, \ldots, m$

$x_{ij}, f_j \in \{0, 1\}$ για κάθε i, j

Η μεταβλητή f_j υποδεικνύει εάν το j-οστό κιβώτιο χρησιμοποιείται ($f_j = 1$) ή είναι άδειο ($f_j = 0$). Η αντικειμενική συνάρτηση είναι ο αριθμός των κιβωτίων που χρησιμοποιούνται. Η μεταβλητή x_{ij} υποδεικνύει εάν το αντικείμενο i έχει τοποθετηθεί στο κιβώτιο j λαμβάνοντας τιμή 1 ή 0 αντίστοιχα. Ο πρώτος περιορισμός απαιτεί κάθε αντικείμενο να τοποθετείται ακριβώς σε ένα κιβώτιο. Στον δεύτερο περιορισμό η M είναι μια οποιαδήποτε σταθερά μεγαλύτερη του

n (δεν επηρεάζει τη βέλτιστη λύση) και ο περιορισμός αυτός συσχετίζει τις μεταβλητές x_{ij} με τις f_j: εάν κάποιο αντικείμενο τοποθετείται στο j-οστό κιβώτιο τότε το κιβώτιο αυτό χρησιμοποιείται, δηλαδή $f_j = 1$. Ο τρίτος περιορισμός δεν επιτρέπει το συνολικό όγκο που καταλαμβάνουν τα αντικείμενα που τοποθετούνται στο κιβώτιο j να υπερβεί τη χωρητικότητα του κιβωτίου αυτού. Τέλος, οι ακέραιοι περιορισμοί για τις μεταβλητές απόφασης.

Server consolidation

Στα μεγάλα υπολογιστικά κέντρα ο υπολογιστικός φόρτος που δέχονται οι υπολογιστές έχει μεγάλη διακύμανση στον χρόνο: στις ώρες αιχμής οι web servers δέχονται περισσότερες αιτήσεις εξυπηρέτησης από ότι τις νυχτερινές ώρες. Χρησιμοποιώντας τεχνολογία virtualization είναι δυνατό ο αριθμός των ενεργών υπολογιστών να μεταβάλλεται κατά τη διάρκεια της ημέρας έτσι ώστε να επιτυγχάνεται εξοικονόμηση ενέργειας κατά τη διάρκεια της νύχτας και ικανοποιητική απόδοση κατά τις ώρες αιχμής. Κατά τη διάρκεια αυτών των μεταβολών το λογισμικό των servers μένει ανέπαφο και εξακολουθούν να λειτουργούν κανονικά, παρόλο που η εκτέλεσή τους ενδέχεται να πραγματοποιείται σε διαφορετικούς υπολογιστές λόγω μετακίνησης. Αυτό είναι δυνατό λόγω της ψευδαίσθησης εκτέλεσης του λογισμικού σε εικονικές μηχανές. Για να μην υπάρχει μεταβολή στην ταχύτητα εκτέλεσης ενός server καθώς μετακινείται σε διαφορετικούς υπολογιστές, οι εικονική μηχανή στην οποία εκτελείται, λειτουργεί με σταθερή ταχύτητα (κύκλους μηχανής το δευτερόλεπτο). Κατά συνέπεια, ένας υπολογιστής δε μπορεί να εκτελεί τον φόρτο περισσότερων εικονικών μηχανών από ότι επιτρέπει η ταχύτητα του επεξεργαστή του. Πχ., ένας υπολογιστής με ταχύτητα επεξεργαστή 4 GHz δεν μπορεί να εκτελέσει 2 εικονικές μηχανές ταχύτητας 3 GHz η κάθε μια.

Ένα τυπικό πρόβλημα σε υπολογιστικό κέντρο είναι η αντιστοίχηση εικονικών μηχανών σε υπολογιστές που θα εκτελεστούν με τέτοιον τρόπο όπου αφενός τα όρια ταχύτητας των επεξεργαστών να μην ξεπερνιόνται αφετέρου να ενεργοποιούνται όσο το δυνατόν λιγότεροι υπολογιστές ώστε να εξοικονομείται ενέργεια.

Θεωρήστε μια απλή μορφή αυτού του προβλήματος όπου υπάρχουν 5 εικονικές μηχανές με απαιτήσεις σε ταχύτητα, 1.5GHz, 2GHz, 2.5GHz, 1GHz, 1GHz αντίστοιχα. Ένας ενεργός υπολογιστής έχει ταχύτητα επεξεργαστή 4GHz και καταναλώνει 50Watt. Ποια είναι η καλύτερη αντιστοίχηση εικονικών μηχανών σε υπολογιστές, σύμφωνα με τα κριτήρια που αναφέραμε παραπάνω;

Μπορούμε να διαμορφώσουμε το πρόβλημα αυτό ως ένα πρόβλημα συσκευασίας κιβωτίων: οι εικονικές μηχανές αντιστοιχούν σε αντικείμενα, οι απαιτήσεις ταχύτητας σε όγκο και οι υπολογιστές σε κιβώτια. Έτσι, εδώ έχουμε $n = 5, m = 5$ (δε θα χρειαστούν πάνω από 5 υπολογιστές), $V_j = 4$ για κάθε j και $v_1 = 1.5, v_2 = 2, v_3 = 2.5, v_4 = v_5 = 1$.

Παράδειγμα 2: εύρεση συντομότερου μονοπατιού

Ένα από τα γνωστότερα αλγοριθμικά προβλήματα είναι η εύρεση του συντομότερου μονοπατιού σε ένα κατευθυνόμενο γράφημα. Έστω V το σύνολο των κορυφών και $E \subset V \times V$ το σύνολο των ακμών. Για κάθε ακμή (i,j) δίδεται η "απόσταση" d_{ij} η οποία θεωρούμε ότι μπορεί να λάβει και αρνητικές τιμές. Έστω ότι η κορυφή αφετηρίας και προορισμού είναι η $s \in V$ και $t \in V$ αντίστοιχα. Ένα μονοπάτι από την αφετηρία στον προορισμό είναι ένα σύνολο από ακμές $\{(i_0,i_1),(i_1,i_2),(i_2,i_3)\ldots,(i_{n-1},i_n)\} \subseteq E$ με $i_0 = s$ και $i_n = t$. Το μήκος ενός μονοπατιού $\Pi \subseteq E$ είναι το άθροισμα των αποστάσεων $\sum_{(i,j)\in\Pi} d_{ij}$ των ακμών που το απαρτίζουν, δηλαδή . Το συντομότερο μονοπάτι από το s στο t είναι το μονοπάτι με το μικρότερο μήκος.

Πρόβλημα συντομότερου μονοπατιού: ποιο είναι το συντομότερο μονοπάτι από το το s στο t;

Το πρόβλημα διατυπώνεται σε μορφή ακέραιου προγράμματος ως εξής:

$\min \sum_{i,j} d_{ij} x_{ij}$ έτσι ώστε $\sum_j x_{sj} = 1$
$\sum_i x_{it} = 1$
$x_{ij} = \sum_k x_{jk}$ για κάθε $i \neq s, j \neq t$ $u_i + n x_{ij} \leq u_j + n - 1$ για κάθε i,j $x_{ij} \in \{0,1\}, u_i \geq 0$