Απόλυτη Προσέγγιση

Συνδυαστική Βελτιστοποίηση

Βασίλης Ζησιμόπουλος

Θεωριτική Πληροφορική
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Αποθήκευση k αρχείων

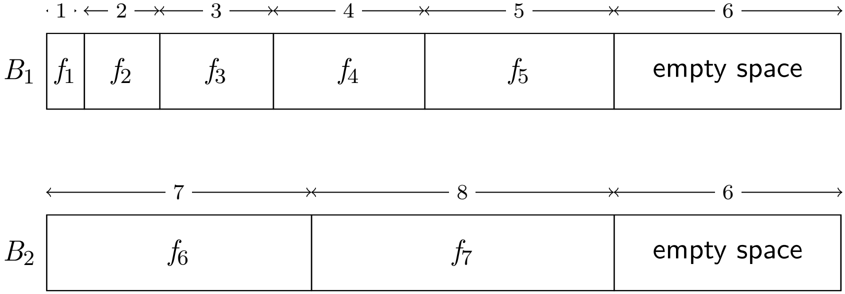
Δίδονται δύο μαγνητικές ταινίες B_1 και B_2 του ίδιου μήκους L και n αρχεία οποιουδήποτε μεγέθους. Θέλουμε να αντιγράψουμε το μέγιστο αριθμό αρχείων πάνω στις ταινίες B_1 και B_2. Υποθέτουμε ότι το πλάτος των ταινιών και αυτό των αρχείων είναι το ίδιο, έτσι ώστε να μην παίζει κανένα ρόλο. Να δοθεί ένας γρήγορος αλγόριθμος ο οποίος επιτρέπει να αντιγράψουμε k αρχεία όταν η βέλτιστη λύση είναι $k+1$ αρχεία.

Παράδειγμα

(Αποθήκευση 7 αρχείων)

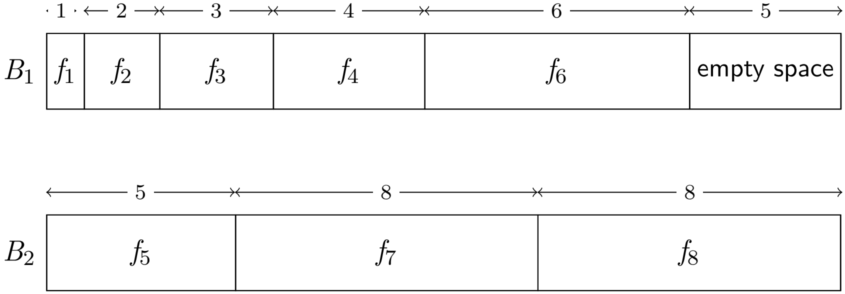
Έστω 8 αρχεία $\{f1, f2, \ldots, f8\}$ με μεγέθη $\{1, 2, 3, 4, 5, 7, 8, 8\}$ αντίστοιχα. Το μήκος των ταινιών είναι $L = 21$. Ο αλγόριθμός μας θα πρέπει να αποθηκεύει 7 αρχεία, τα $\{f1, f2, \ldots, f7\}$. Η καλύτερη λύση που μπορούμε να πετύχουμε για τα 8 αρχεία είναι να τα αποθηκεύσουμε με την εξής σειρά :
$\{f1, f2, f3, f4, f6\}$ στην ταινία B_1 και $\{f5, f7, f8\}$ στη B_2.

Παράδειγμα



Σχήμα: Η τοποθέτηση των 7 αρχείων που δίνει ο αλγόριθμος

Παράδειγμα



Σχήμα: Η βέλτιστη τοποθέτηση των 8 αρχείων στις ταινίες.

1-absolute προσεγγιστικός αλγόριθμος

Algorithm 1 Store k files

1: Ταξινόμησε τα αρχεία κατά αύξουσα σειρά μεγέθους
2: Τοποθέτησε με αυτή τη σειρά τα αρχεία πρώτα στη B_1 και μετά στη B_2
3: **επέστρεψε** τα σύνολα αρχείων που αποθηκεύτηκαν σε κάθε ταινία

Θεωρούμε τις λύσεις (και αρκεί αυτό) όπου τοποθετούμε πρώτα τα πιο μικρά αρχεία. Πράγματι, αν μπορούμε να αποθηκεύσουμε k αρχεία οποιουδήποτε μεγέθους, τότε θα μπορούμε να αντικαταστήσουμε αυτά τα αρχεία με τα πιο μικρά.

1-absolute προσεγγιστικός αλγόριθμος

- $k = A(I)$ και S η αντίστοιχη τοποθέτηση
- S^* βέλτιστη τοποθέτηση με $k + 2$ αρχεία.
- f_{k+1} και f_{k+2} το $(k + 1)$-στό και το $(k + 2)$-στό μικρότερο αρχείο αντίστοιχα.
- Στην S^* βάζουε τα k πιο μικρά αρχεία μετά το f_{k+1} και στο τέλος το f_{k+2}
- f_d το τελευταίο αρχείο που τοποθετήθηκε στη B_1 (και επομένως f_{d+1} το πρώτο αρχείο που τοποθετήθηκε στη B_2) στην τοποθέτηση S

1-absolute προσεγγιστικός αλγόριθμος

- V_1, V_2 ο κενός χώρος στις ταινίες B_1 και B_2 αντίστοιχα στην τοποθέτηση S
- Παρατηρούμε ότι: $size(f_{k+1}) + size(f_{k+2}) \leq V_1 + V_2$ (διαφορετικά δε θα μπορούσαμε να έχουμε $k+2$ αρχεία στην S^*)
- $V_1 < size(f_{d+1})$ και $V_2 < size(f_{k+1})$ (διαφορετικά το f_{d+1} θα ήταν στην B_1 και το f_{k+1} στην B_2. στην τοποθέτηση S)
- $size(f_{d+1}) \leq size(f_{k+2})$
- Συνεπώς: $V_1 + V_2 < size(f_{k+2}) + size(f_{k+1}) \leq V_1 + V_2$. Αδύνατον!

Δεύτερη απόδειξη

- $\hat{F}(I)$ πλήθος αρχείων από τον αλγόριθμο
- $F^*(I)$ πλήθος αρχείων της βέλτιστης λύσης
- Θέλουμε να δείξουμε ότι: $|F^*(I) - \hat{F}(I)| \leq 1$
- 2 ταινίες μήκους L, τότε $\hat{F}(I) = k$
- Αν η μια ταινία έχει μήκος $2L$, τότε $F^*(I) \leq p$ (p αρχεία κατά αύξουσα σειρά μεγέθους) και $\sum_{i=1}^{p} l_i \leq 2L$

Δεύτερη απόδειξη

- $\sum_{i=1}^{j} l_i \leq L$ (όπου j ο μεγαλύτερος δείκτης θέσης) $\Rightarrow j \leq p$.
- Ο αλγόριθμος θα αποθηκεύσει τα j μικρότερα αρχεία στην ταινία B_1
- $\sum_{i=j+1}^{p-1} l_i \leq \sum_{i=j+2}^{p} l_i \leq L$
- Ο αλγόριθμος θα αποθηκεύσει τα $j+1, j+2, \ldots, p-1$ αρχεία στην ταινία B_2
- $\hat{F}(I) \geq p-1$
- $|F^*(I) - \hat{F}(I)| \leq 1$
- $k-1$-absolute (για k ταινίες)

Η απόλυτη προσέγγιση είναι NP-hard

- 2 ταινίες NP-complete $\Rightarrow$ Abs-Appr. 1 $\Rightarrow$ πολλά άλλα NP-complete!
- <u>αλλά</u> η απόλυτη προσέγγιση για το knapsack είναι NP-hard

Η απόλυτη προσέγγιση είναι NP-hard

- Αλγόριθμος A με $\rho = |F^*(I) - F_A(I)|$
- Στιγμιότυπο του προβλήματος του σακιδίου: $n = 3, B = 100, (p_1, p_2, p_3) = (1, 2, 3), (s_1, s_2, s_3) = (50, 60, 30)$

Η απόλυτη προσέγγιση είναι NP-hard

- Αλγόριθμος A με $\rho = |F^*(I) - F_A(I)|$
- Στιγμιότυπο του προβλήματος του σακιδίου: $n = 3, B = 100, (p_1, p_2, p_3) = (1, 2, 3), (s_1, s_2, s_3) = (50, 60, 30)$

$I:$

$$\begin{array}{ccc} (1,0,0) & 1 & 5 \\ (0,1,0) & 2 & 10 \\ (0,0,1) & 3 & 15 \\ (1,0,1) & 4 & 20 \\ (0,1,1) & 5 & 25 \end{array}$$

$I' = (Pp_1, Pp_2, Pp_3) \underset{P=5}{=} (5, 10, 15)$

$k = 4 \Rightarrow (0, 1, 1)$ βέλτιστη λύση

$0-1$ Knapsack reduces to k-abs approximation knapsack

- A πολυωνυμικός αλγόριθμος με $|F^*(I) - F_A(I)| \leq k \ \forall I$ και k σταθερό.
- $(p_i, s_i), \ \ 1 \leq i \leq n, \ \ p_i \in \mathbb{N}$
- $I^{'} : ((k+1)p_i, s_i), \ \ 1 \leq i \leq n$
- $I, I^{'}$ τις ίδιες εφικτές λύσεις, καθώς και τις ίδιες βέλτιστες λύσεις
- $F^*(I^{'}) = (k+1)F^*(I)$

Η απόλυτη προσέγγιση είναι NP-hard

- Επειδή $p_i \in \mathbb{N}$ για κάθε εφικτή λύση στο $I^{'}$, θάχουμε τιμή $F^*(I^{'})$ ή μικρότερη από $F^*(I^{'}) - (k+1)$
- Εάν $F_A(I^{'}) \Rightarrow F^*(I^{'}) - F_A(I^{'}) = \begin{cases} 0 \\ \geq k+1 \end{cases}$
- Εάν $F^*(I^{'}) - F_A(I^{'}) \leq k \Rightarrow F^*(I^{'}) = F_A(I^{'}) \Rightarrow A$ βρίσκει την βέλτιστη λύση για το $I^{'}$ και άρα για το I
- $length(I^{'}) \leq (logk)(length(I))$ (κατασκευή του $I^{'}$ από το I σε πολυωνυμικό χρόνο)

Coloring

- $G = (V, E)$
- Ελάχιστος αριθμός χρωμάτων με δύο γειτονικούς κόμβους να έχουν διαφορετικό χρώμα.
- $p = logn$
- $\frac{H}{OPT} \leq logn \Rightarrow H \leq OPT * logn$