Άπληστοι Αλγόριθμοι

Ένα πρόβλημα επιλογής δραστηριοτήτων

- Δίνεται κάποιο σύνολο $S = a_1, a_2, \ldots, a_n$ από n ***δραστηριότητες*** που απαιτούν τη χρήση ενός πόρου ο οποίος μπορεί να χρησιμοποιείται μόνο από μία δραστηριότητα κάθε φορά.
- Η κάθε δραστηριότητα a_i έχει ***χρόνο έναρξης*** s_i και ***χρόνο λήξης*** f_i, όπου $0 < s_i < f_i < \infty$.
- Οι δραστηριότητες a_i και a_j χαρακτηρίζονται ***συμβατές*** εάν τα διαστήματα $[s_i, f_i)$ και $[s_j , f_j)$ δεν επικαλύπτονται
- Το ***πρόβλημα της επιλογής δραστηριοτήτων:*** η εύρεση ενός μέγιστου συνόλου αμοιβαία συμβατών δραστηριοτήτων.

Ένα πρόβλημα επιλογής δραστηριοτήτων

Οι δραστηριότητες είναι διατεταγμένες κατά μονότονα αύξουσα σειρά ως προς τον χρόνο λήξης:

$$f_1 \leq f_2 \leq f_3 \leq \cdots \leq f_{n-1} \leq f_n$$

Π.χ.:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	8	9	10	11	12	13	14

Αμοιβαία συμβατές δραστηριοτήτες: $\{a_3, a_9, a_{11}\}$
Μέγιστα σύνολα: $\{a_1, a_4, a_8, a_{11}\}$, $\{a_2, a_4, a_9, a_{11}\}$.

Ορίζουμε

$$S_{ij} = \{a_k \in S : f_i \leq s_k < f_k \leq s_j\}$$

c[i,j] = το πλήθος των δραστηριοτήτων σε ένα μέγιστο υποσύνολο αμοιβαία συμβατών δραστηριοτήτων στο S_{ij}.

Προσθέτουμε τις φανταστικές δραστηριότητες a_0 με χρόνο λήξης 0 και a_{n+1} με χρόνο έναρξης ∞.

Η απαιτούμενη λύση: c[0, n+1].

$$c[i,j] = \begin{cases} 0 & \text{if } S_{ij} = \varnothing \\ \max_{\substack{i<k<j \\ a_k \in S_{ij}}} \{c[i,k] + c[k,j] + 1\} & \text{if } S_{ij} \neq \varnothing \end{cases}$$

Η αναδρομική εξίσωση οδηγεί σε λύση δυναμικού προγραμματισμού.

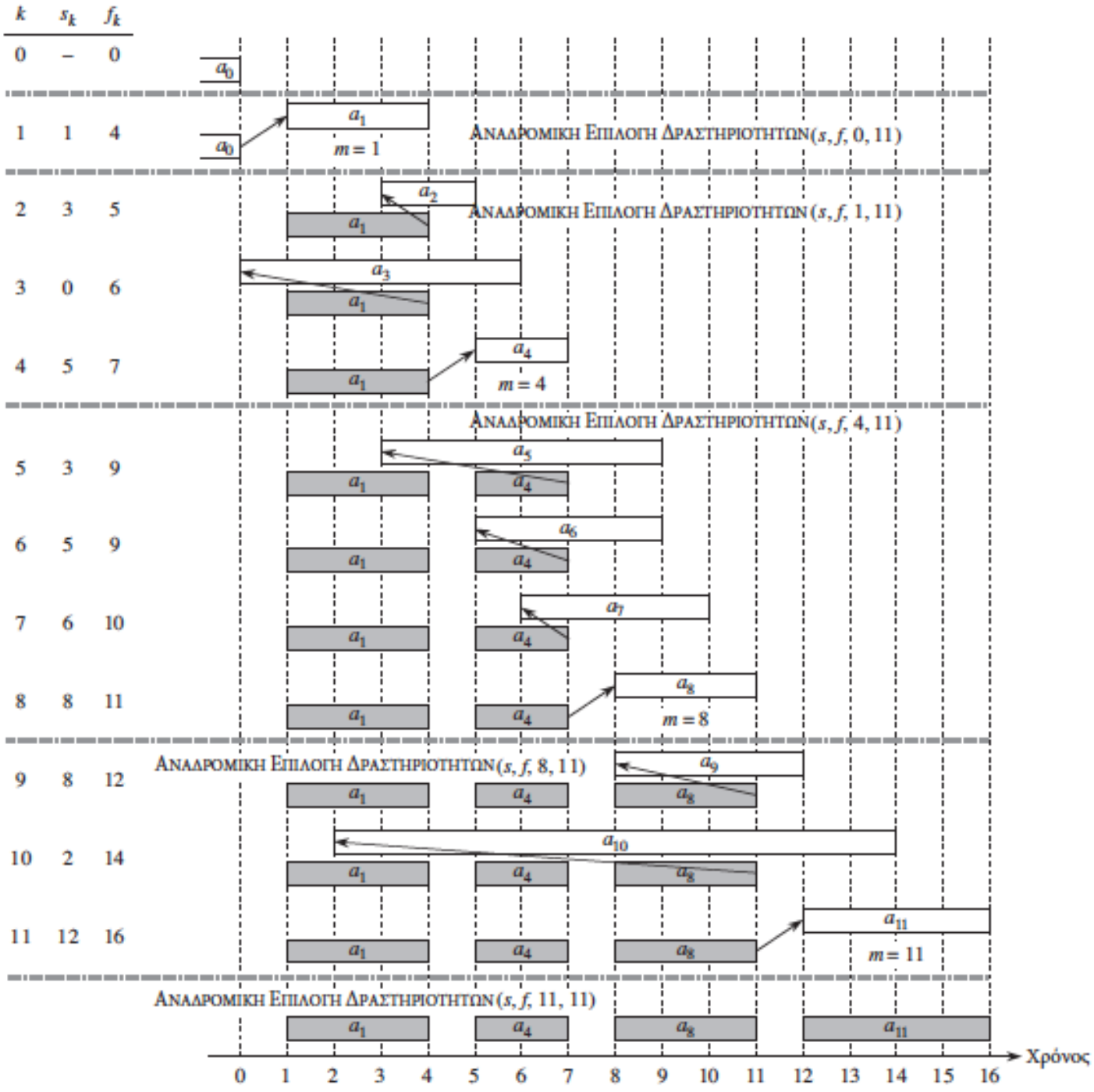
Όμως υπάρχει και γρηγορότερη λύση:

Θεώρημα: Έστω a_m η δραστηριότητα του S_{ij} με το μικρότερο χρόνο λήξης δηλ., $f_m = \min\{f_k : a_k \in S_{ij}\}$. Τότε η δραστηριότητα a_m χρησιμοποιείται σε κάποιο μεγίστου πλήθους υποσύνολο αμοιβαία συμβατών δραστηριοτήτων του S_{ij}.

Το υποπρόβλημα S_{im} κενό, επομένως απομένει μόνο ένα υποπρόβλημα S_{mj} μη κενό.

```
RECURSIVE-ACTIVITY-SELECTOR(s, f, k, n)
1  m = k + 1
2  while m ≤ n and s[m] < f[k]        // find the first activity in S_k to finish
3      m = m + 1
4  if m ≤ n
5      return {a_m} ∪ RECURSIVE-ACTIVITY-SELECTOR(s, f, m, n)
6  else return ∅
```

```
GREEDY-ACTIVITY-SELECTOR(s, f)
1  n = s.length
2  A = {a_1}
3  k = 1
4  for m = 2 to n
5      if s[m] ≥ f[k]
6          A = A ∪ {a_m}
7          k = m
8  return A
```


k
s_k
f_k
0 – 0
1 1 4
2 3 5
3 0 6
4 5 7
5 3 9
6 5 9
7 6 10
8 8 11
9 8 12
10 2 14
11 12 16
a_0
a_1
a_2
a_3
a_4
a_5
a_6
a_7
a_8
a_9
a_{10}
a_{11}
$m = 1$
$m = 4$
$m = 8$
$m = 11$
ΑΝΑΔΡΟΜΙΚΗ ΕΠΙΛΟΓΗ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ$(s, f, 0, 11)$
ΑΝΑΔΡΟΜΙΚΗ ΕΠΙΛΟΓΗ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ$(s, f, 1, 11)$
ΑΝΑΔΡΟΜΙΚΗ ΕΠΙΛΟΓΗ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ$(s, f, 4, 11)$
ΑΝΑΔΡΟΜΙΚΗ ΕΠΙΛΟΓΗ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ$(s, f, 8, 11)$
ΑΝΑΔΡΟΜΙΚΗ ΕΠΙΛΟΓΗ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ$(s, f, 11, 11)$
Χρόνος
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Άπληστη στρατηγική έναντι δυναμικού προγραμματισμού

Το ***ακέραιο πρόβλημα του σακιδίου:***

- Ένας διαρρήκτης παραβιάζει ένα κατάστημα και βρίσκει n αντικείμενα.
- Το i-οστό αντικείμενο έχει αξία v_i ευρώ και βάρος w_i κιλά, όπου τα v_i και w_i είναι ακέραιοι αριθμοί.
- Ο διαρρήκτης θέλει να αποκομίσει όσο το δυνατόν πολυτιμότερη λεία
- Μπορεί να μεταφέρει στο σακίδιό του φορτίο βάρους το πολύ ίσου με W κιλά, όπου W ακέραιος αριθμός.
- Ποια αντικείμενα θα πρέπει να πάρει;

Άπληστη στρατηγική έναντι δυναμικού προγραμματισμού

Κλασματικό πρόβλημα του σακιδίου:

Έχουμε το ίδιο σκηνικό μόνο που ο διαρρήκτης δεν είναι υποχρεωμένος να επιλέξει «διαζευκτικά» για το κάθε αντικείμενο, αλλά μπορεί να πάρει και κλάσματα αντικειμένων.

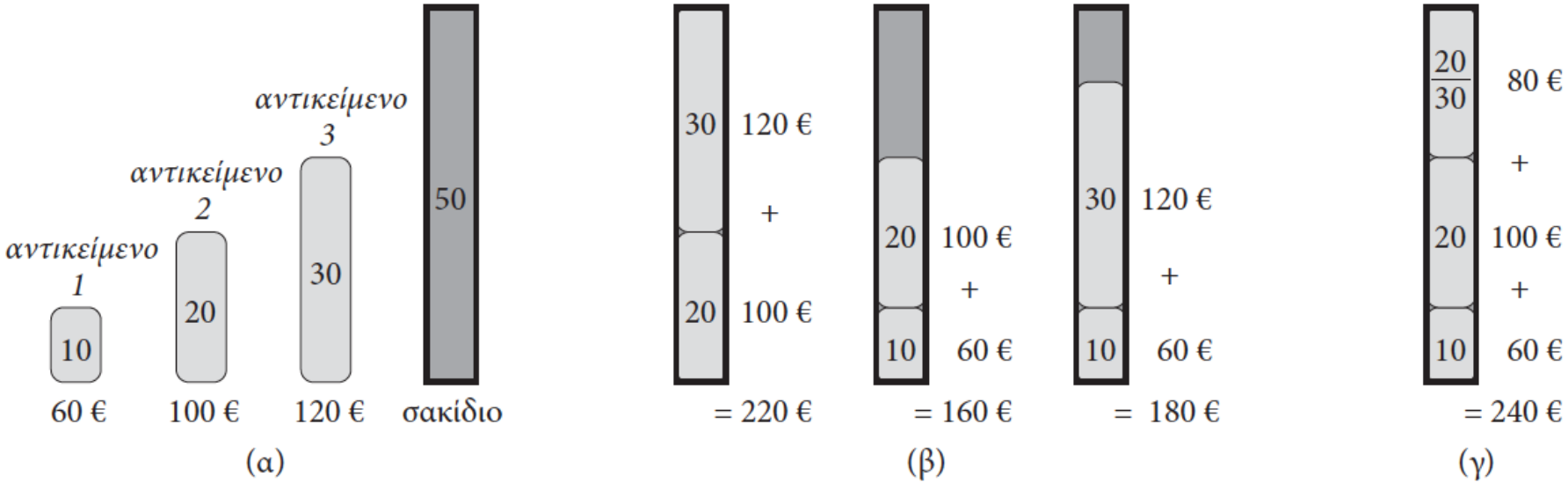
Δυναμικός Προγραμματισμός για το ακέραιο πρόβλημα

- Αν *k* είναι όλα τα αντικείμενα και *V*[*k*,*w*] είναι η μέγιστη συνολική αξία που μπορούν να χωρέσουν στο σάκο χωρίς το συνολικό βάρος να υπερβεί το *w*, ισχύει η ακόλουθη αναδρομική σχέση:

$$V[k,w] = \begin{cases} V[k-1,w] & \text{if } w_k > w \\ \max\{V[k-1,w], V[k-1,w-w_k]+v_k\} & \text{αλλιώς} \end{cases}$$

Κλασματικό πρόβλημα σακιδίου

- Ταξινομούμε τα αντικείμενα ως προς το λόγο v_i/w_i κατά φθίνουσα.
- Με αυτή τη σειρά εισάγουμε τα αντικείμενα στο σάκο.
- Αν κάποιο δεν χωράει ολόκληρο παίρνουμε ακριβώς το κλάσμα από αυτό το αντικείμενο μέχρι να το συνολικό βάρος να φθάσει στο όριο w.
- Άπληστη λογική.



Σχήμα 16.2 Ένα παράδειγμα που δείχνει ότι η άπληστη στρατηγική δεν δίνει σωστά αποτελέσματα για το ακέραιο πρόβλημα του σακιδίου. (α) Ο διαρρήκτης θα πρέπει να επιλέξει ένα υποσύνολο των τριών εικονιζόμενων αντικειμένων το βάρος του οποίου να μην υπερβαίνει τα 50 κιλά. (β) Το βέλτιστο υποσύνολο αποτελείται από τα αντικείμενα 2 και 3. Οποιαδήποτε λύση που περιλαμβάνει το αντικείμενο 1 είναι χειρότερη της βέλτιστης, παρ' όλο που το αντικείμενο αυτό έχει τη μέγιστη αξία ανά μονάδα βάρους. (γ) Για το κλασματικό πρόβλημα του σακιδίου, η επιλογή των αντικειμένων κατά φθίνουσα σειρά αξίας ανά μονάδα βάρους δίνει μια βέλτιστη λύση.