ΜΕΘΟΔΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΤΗ ΔΙΟΙΚΗΤΙΚΗ ΕΠΙΣΤΗΜΗ (ΜΕ.ΒΕ.Δ.Ε)

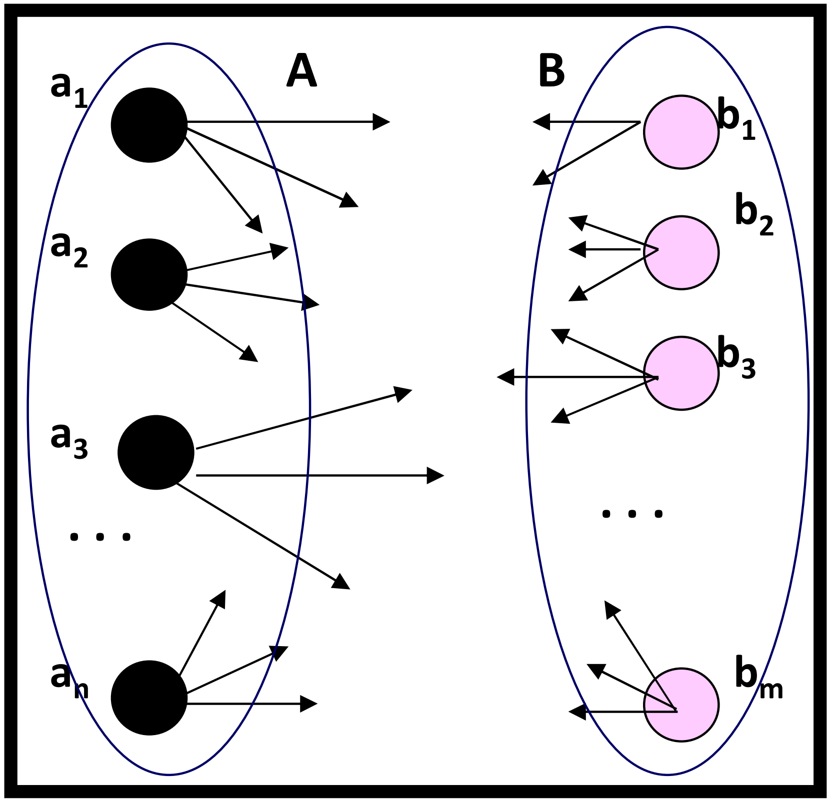
Emmanouil Zachariadis, Assistant Professor
Department of Management Science and Technology,
Athens University of Economics and Business
E: ezach@aueb.gr
A: Evelpidon 47A and Lefkados 33 street, 8th floor, Room 811, 11362, Athens, Greece
T: +30-210-8203 467

ΣΥΝΔΥΑΣΤΙΚΗ ΔΟΜΗ: ΑΝΤΙΣΤΟΙΧΙΣΗ

- Προβλήματα Αντιστοίχισης:

  Ένα πρόβλημα καλείται ***πρόβλημα αντιστοίχισης*** όταν η μορφή της λύσης του εκφράζεται ως αντιστοιχίσεις μεταξύ στοιχείων δύο (ή περισσότερων) συνόλων

- Διαφορετικές αντιστοιχίσεις στοιχείων αντιστοιχούν σε διαφορετικές λύσεις.



ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΡΟΒΛΗΜΑΤΩΝ ΑΝΤΙΣΤΟΙΧΙΣΗΣ

- Τα Προβλήματα Αντιστοίχισης αφορούν την εύρεση της βέλτιστης αντιστοίχισης στοιχείων των δυο αυτών συνόλων, με σκοπό την επίτευξη κάποιου στόχου

- Παραδείγματα Προβλημάτων Αντιστοίχισης:
  - Αντιστοίχιση εγκαταστάσεων σε τοποθεσίες
  - Αντιστοίχιση εργαζομένων σε χρονικά διαστήματα
  - Αντιστοίχιση τουριστών σε λεωφορεία
  - Αντιστοίχιση διαφημίσεων σε συγκεκριμένα χρονικά διαστήματα

ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΗΣ ΣΥΣΚΕΥΑΣΙΑΣ/ΤΟΠΟΘΕΤΗΣΗΣ

- Πολλές εφαρμογές στη Διοικητική των Επιχειρήσεων έχουν ως στόχο την τοποθέτηση στοιχείων σε χώρους
  - Προϊόντα σε συσκευασίες
  - Εργασίες σε Βάρδιες
  - Τουρίστες σε λεωφορεία
- Αυτά τα προβλήματα ονομάζονται προβλήματα συσκευασίας (Packing Problems)
- <u>Packing Problems</u>: Αντιστοιχίσεις αντικειμένων σε χώρους
- Τα packing προβλήματα μπορούν να έχουν διάφορες αντικειμενικές συναρτήσεις
  - Μεγιστοποίηση της χρησιμοποίησης των χώρων
  - Ευστάθεια των μεταφερόμενων αντικειμένων
  - Εύκολες διαδικασίες εκφόρτωσης
  - …

BIN PACKING PROBLEM - BPP

- Bin Packing Problem
- Ένα από τα σημαντικότερα προβλήματα αντιστοίχισης
- Αντικειμενική Συνάρτηση: Τοποθέτηση *όλων* των αντικειμένων σε χώρους (bins), έτσι ώστε ο αριθμός των containers που χρησιμοποιούνται να είναι ο ελάχιστος
  - Αντικείμενα: Διάφορα Μεγέθη
  - Χώροι: Ίδιο Μέγεθος

BIN PACKING PROBLEM - BPP

- Το BPP περιγράφεται ως εξής:
  - Σύνολο n αντικειμένων J
  - Κάθε αντικείμενο $j \in J$ εχει μέγεθος w_j $(j = \{1, \dots, n\})$
  - Έστω ένα ομοιογενές σύνολο αποθηκευτικών χώρων (bins) με μέγιστο βάρος φόρτωσης Q
  - Το συνολικό βάρος των αντικειμένων που περιέχει κάθε αποθηκευτικός χώρος δεν πρέπει να υπερβαίνει την χωρητικότητά του αποθηκευτικού χώρου Q
  - Επομένως, για να είναι εφικτό το πρόβλημα: $w_j \leq Q, \forall j = 1, \dots, n$
  - Σκοπός του προβλήματος είναι η τοποθέτηση όλων των αντικειμένων σε αποθηκευτικούς χώρους, έτσι ώστε να ελαχιστοποιούνται οι χρησιμοποιούμενοι αποθηκευτικοί χώροι

BIN PACKING PROBLEM - BPP

- Ποιο είναι ένα προφανές κατώτατο όριο για την τιμή της αντικειμενικής συνάρτησης;
  - Ποιο είναι ο χαμηλότερος αριθμός χώρων ο οποίος θα μπορούσε να φιλοξενήσει όλα τα αντικείμενα
- Ποιο είναι ένα προφανές ανώτατο όριο για την τιμή της αντικειμενικής συνάρτησης;
  - Ποιο είναι ο υψηλότερος αριθμός χώρων ο οποίος θα μπορούσε να φιλοξενήσει όλα τα αντικείμενα

BIN PACKING PROBLEM - BPP

- Ποιο είναι ένα προφανές κατώτατο όριο για την τιμή της αντικειμενικής συνάρτησης;
  - Lower Bound:

$$LB = \left\lceil \frac{\sum_{i=1}^{n} w_i}{Q} \right\rceil$$

- Ποιο είναι ένα προφανές ανώτατο όριο για την τιμή της αντικειμενικής συνάρτησης;
  - Upper Bound:

$$UB = n$$

BIN PACKING PROBLEM - BPP

- Αντικειμενική Συνάρτηση:

$$minimize\ B = \sum_{i=1}^{n} y_i$$

- Περιορισμοί:

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad \forall j \in \{1, \dots, n\}$$

Όλα τα αντικείμενα θα τοποθετηθούν σε ένα χώρο

$$\sum_{j=1}^{n} w_j \cdot x_{ij} \leq Q \cdot y_i, \qquad \forall i \in \{1, \dots, n\}$$

Περιορισμός χωρητικότητας

$$y_i \in \{0,1\}, \qquad \forall i \in \{1, \dots, n\}$$

Ο Αποθηκευτικός Χώρος i χρησιμοποιείται

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, n\}$$

Αντικείμενο j στον Αποθηκευτικό Χώρο i

ΕΦΑΡΜΟΓΗ μιας ΠΣΕ για το BPP

- <u>Μορφή Λύσης</u>

  Ένα σύνολο:

  – $\boldsymbol{n}$ αντιστοιχίσεων

  – *Κάθε μία από τις n αντιστοιχίσεις:*

  (Αντικείμενο j, Αποθηκευτικός Χώρος j)

- Στοιχείο της Λύσης:

  Μια αντιστοίχιση αντικειμένου - αποθηκευτικού χώρου.

Σχεδιασμός μίας ΠΣΕ

Προτάσεις για Κριτήριο Επιλογής του Στοιχείου της Λύσης…

ΕΦΑΡΜΟΓΗ μιας ΠΣΕ για το BPP

Online Bin-Packing

Οι αποφάσεις λαμβάνονται με την άφιξη ενός αντικειμένου

Η εισαγωγή ενός αντικειμένου σε ένα χώρο γίνεται με πλήρη άγνοια των επόμενων αντικειμένων

- Δύο γνωστοί αλγόριθμοι οι οποίοι αντιστοιχούν σε δύο κριτήρια επιλογής

- 1ο Κριτήριο: Επίλεξε το επόμενο αντικείμενο το οποίο δεν έχει ήδη τοποθετηθεί και τοποθέτησέ το στον ***πρώτο*** διαθέσιμο χώρο που μπορεί να το φιλοξενήσει (First-Fit Heuristic)

- 2ο Κριτήριο: Επίλεξε το επόμενο αντικείμενο το οποίο δεν έχει ήδη τοποθετηθεί και τοποθέτησέ το στον ***πιο γεμάτο*** (με το μικρότερο διαθέσιμο περιθώριο χωρητικότητας) διαθέσιμο χώρο που μπορεί να το φιλοξενήσει (Best-Fit Heuristic)

ΕΦΑΡΜΟΓΗ μιας ΠΣΕ για το BPP

Offline Bin-Packing

Οι αποφάσεις λαμβάνονται όταν όλα τα αντικείμενα είναι γνωστά

- Οι αλγόριθμοι First-Fit and Best-Fit μπορούν να βελτιωθούν με ένα προπαρασκευαστικό βήμα:

  Κατάταξε τα αντικείμενα σε φθίνουσα σειρά μεγέθους

  (το μεγαλύτερο αντικείμενο πρώτο)

- Όταν η κατάταξη προηγείται της εφαρμογής τους, οι αλγόριθμοι καλούνται
  - First-Fit Decreasing
  - Best-Fit Decreasing

ΕΦΑΡΜΟΓΗ μιας ΠΣΕ για το BPP

- Αντικειμενική Συνάρτηση:

Πλήθος των μη-κενών αποθηκευτικών χώρων μετά την τοποθέτηση όλων των αντικειμένων

ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

- 11 Αντικείμενα, τα ύψη των οποίων είναι:

$$w_1 = 8, w_2 = 7, w_3 = 14, w_4 = 9, w_5 = 6, w_6 = 9\ , w_7 = 2,$$
$$w_8 = 15, w_9 = 6, w_{10} = 7, w_{11} = 8$$

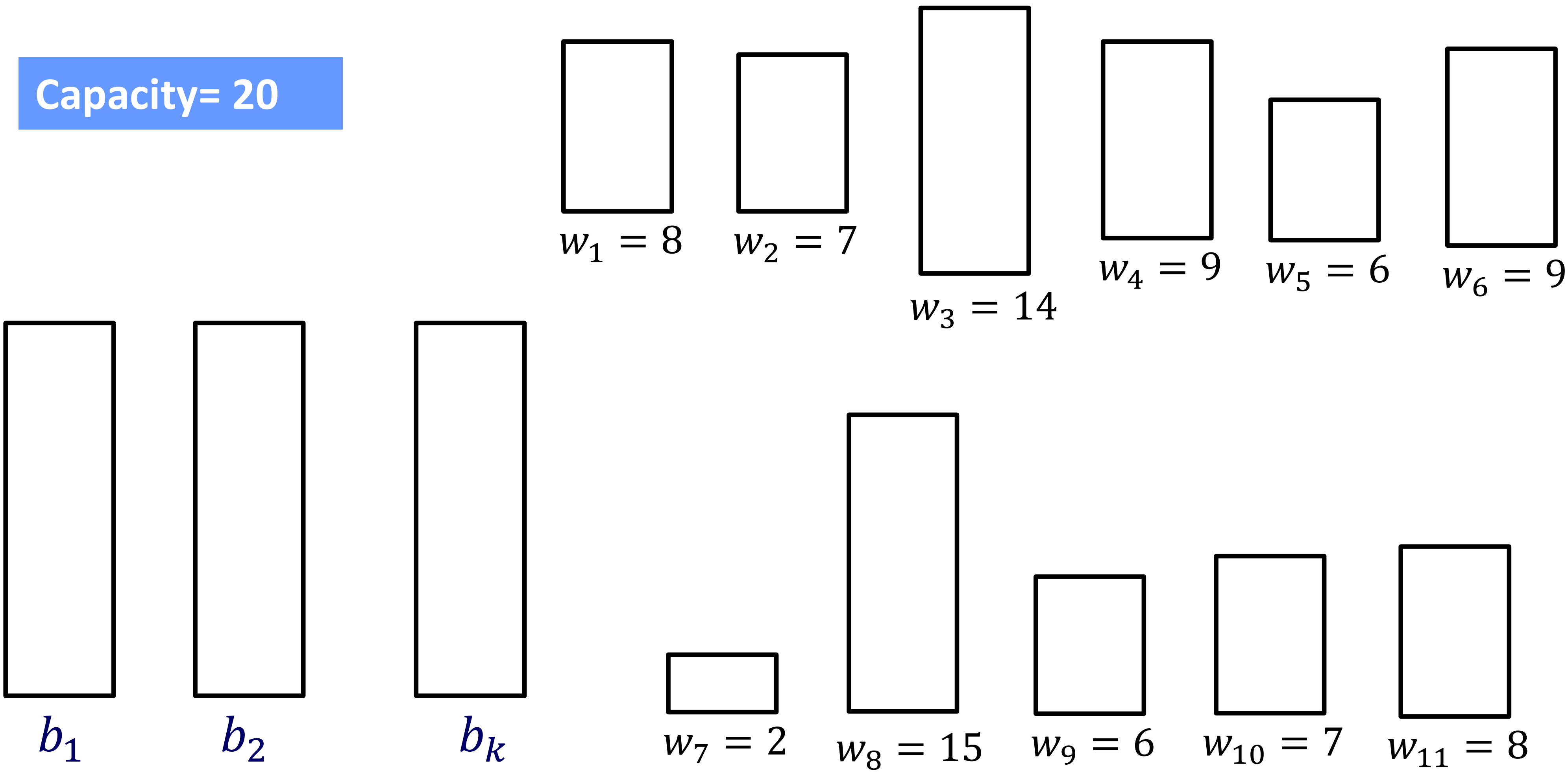
- Οι διαθέσιμοι αποθηκευτικοί χώροι $B = \{b_1, b_2, \ldots, b_n\}$ έχουν ύψος $Q = 20$

- Στόχος μας είναι η ελαχιστοποίηση των αποθηκευτικών χώρων για την τοποθέτηση όλων των containers.

- Εφαρμογή του Αλγορίθμου First-Fit Decreasing Algorithm για το πρόβλημα

ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

Capacity= 20



ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

- Μορφή Λύσης: Ένα σύνολο 11 αντιστοιχίσεων αποθηκευτικού χώρου - container.

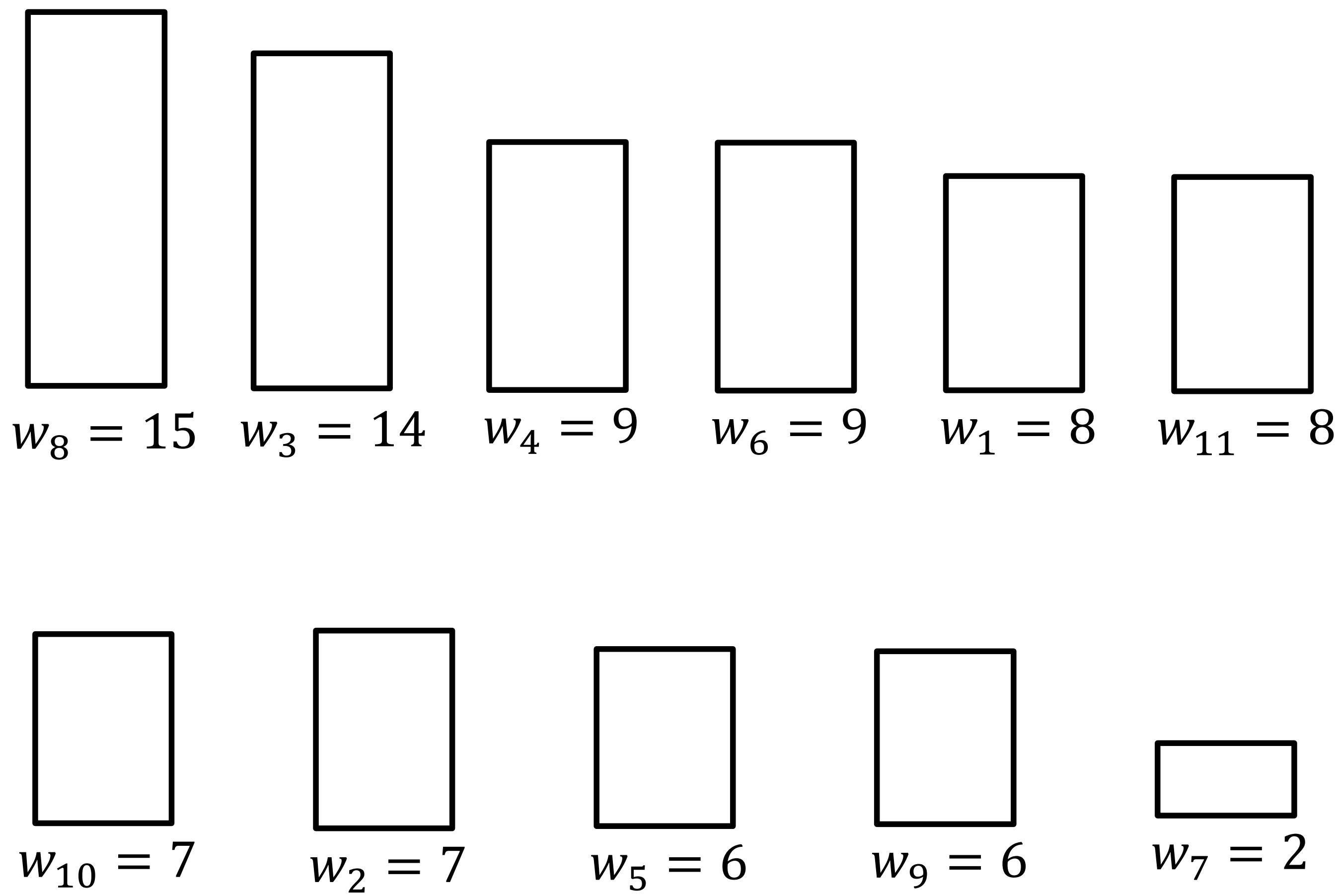
- Στοιχείο Λύσης: Μια αντιστοίχιση αποθηκευτικού χώρου - container.

- Κριτήριο Επιλογής: Κατάταξε τα containers από το ψηλότερο στο κοντύτερο. Ακολούθως, αντιστοίχισε κάθε container στον πρώτο διαθέσιμο αποθηκευτικό χώρο

  (First-Fit Decreasing)

ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

Ordering of Items



ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

Η συγκεκριμένη ΠΣΕ εφαρμόζεται ως εξής:

- Επανάληψη 1. Το Αντικείμενο $w_8 = 15$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 1

$$s = \{(w_8, b_1)\}$$

- Επανάληψη 2. Το Αντικείμενο $w_3 = 14$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 2

$$s = \{(w_8, b_1), (w_3, b_2)\}$$

- Επανάληψη 3. Το Αντικείμενο $w_4 = 9$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 3

$$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3)\}$$

ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

- Επανάληψη 4. Το container $w_6 = 9$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 3

$$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3), (w_6, b_3)\}$$

- Επανάληψη 5. Το Αντικείμενο $w_1 = 8$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 4

$$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3), (w_6, b_3), (w_1, b_4)\}$$

- Επανάληψη 6. Το Αντικείμενο $w_{11} = 8$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 4

$$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3), (w_6, b_3), (w_1, b_4), (w_{11}, b_4)\}$$

ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

- Επανάληψη 7. Το Αντικείμενο $w_2 = 7$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 5
  $$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3), (w_6, b_3), (w_1, b_4), (w_{11}, b_4), (w_2, b_5)\}$$
- Επανάληψη 8. Το Αντικείμενο $w_{10} = 7$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 5
  $$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3), (w_6, b_3), (w_1, b_4), (w_{11}, b_4), (w_2, b_5), (w_{10}, b_5)\}$$
- Επανάληψη 9. Το Αντικείμενο $w_5 = 6$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 2
  $$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3), (w_6, b_3), (w_1, b_4), (w_{11}, b_4), (w_2, b_5), (w_{10}, b_5), (w_5, b_2)\}$$

ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

- Επανάληψη 10. Το Αντικείμενο $w_9 = 6$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 5

$$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3), (w_6, b_3), (w_1, b_4), \\ (w_{11}, b_4), (w_2, b_5), (w_{10}, b_5), (w_5, b_2), (w_9, b_5)\}$$

- Επανάληψη 11. Το Αντικείμενο $w_7 = 2$ αντιστοιχίζεται στον Αποθηκευτικό Χώρο 1

$$s = \{(w_8, b_1), (w_3, b_2), (w_4, b_3), (w_6, b_3), (w_1, b_4), \\ (w_{11}, b_4), (w_2, b_5), (w_{10}, b_5), (w_5, b_2), (w_9, b_5), (w_7, b_1)\}$$

ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

s :

b1:	w_8	w_7	
b2:	w_3	w_5	
b3:	w_4	w_6	
b4:	w_1	w_{11}	
b5:	w_2	w_{10}	w_9

Αξιολόγηση της λύσης $z(s) = 5$

ΕΠΙΛΥΣΗ ΜΕ ΔΙΑΦΟΡΕΤΙΚΟ ΚΡΙΤΗΡΙΟ ΕΠΙΛΟΓΗΣ

- Αντιμετώπιση του ίδιου προβλήματος με τον Best-Fit Decreasing Αλγόριθμο
- Κατάταξε τα αντικείμενα σε φθίνουσα σειρά μεγέθους (το μεγαλύτερο πρώτο),
- Επαναληπτικά, τοποθέτησε κάθε αντικείμενο πιο γεμάτο αποθηκευτικό χώρο, ο οποίος μπορεί φυσικά να φιλοξενήσει το αντικείμενο χωρίς να παραβιάζεται ο περιορισμός χωρητικότητας
- «Πιο γεμάτος αποθηκευτικός χώρος»: Ο χώρος που στην τρέχουσα λύση φιλοξενεί τα αντικείμενα με το μεγαλύτερο συνολικό βάρος

ΕΠΙΛΥΣΗ ΜΕ ΔΙΑΦΟΡΕΤΙΚΟ ΚΡΙΤΗΡΙΟ ΕΠΙΛΟΓΗΣ

Η συγκεκριμένη ΠΣΕ εφαρμόζεται ως εξής:

- Επανάληψη 1. Το Αντικείμενο W_8=15 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 1

s: (W_8b1).

- Επανάληψη 2. Το Αντικείμενο W_3=14 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 2

s: (W_8b1, W_3b2).

- Επανάληψη 3. Το Αντικείμενο W_4=9 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 3

s: (W_8b1, W_3b2, W_4b3).

ΕΠΙΛΥΣΗ ΜΕ ΔΙΑΦΟΡΕΤΙΚΟ ΚΡΙΤΗΡΙΟ ΕΠΙΛΟΓΗΣ

- Επανάληψη 4. Το Αντικείμενο W_6=9 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 3, σύμφωνα με το κριτήριο επιλογής

s: (W_8b1, W_3b2, W_4b3, W_6b3).

- Επανάληψη 5. Το Αντικείμενο W_1=8 (επιλέχθηκε στοχαστικά σε σχέση με το W_{11}) αντιστοιχίζεται στον Αποθηκευτικό Χώρο 4

s: (W_8b1, W_3b2, W_4b3, W_6b3, W_1b4)

- Επανάληψη 6. Το Αντικείμενο W_{11}=8 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 4

s: (W_8b1, W_3b2, W_4b3, W_6b3, W_1b4, W_{11}b4)

ΕΠΙΛΥΣΗ ΜΕ ΔΙΑΦΟΡΕΤΙΚΟ ΚΡΙΤΗΡΙΟ ΕΠΙΛΟΓΗΣ

- Επανάληψη 7. Το Αντικείμενο W_2=7 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 5

s: (W_8b1, W_3b2, W_4b3, W_6b3, W_1b4, W_{11}b4, W_2b5)

- Επανάληψη 8. Το Αντικείμενο W_{10}=7 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 5

s: (W_8b1, W_3b2, W_4b3, W_6b3, W_1b4, W_{11}b4, W_2b5, W_{10}b5)

- Επανάληψη 9. Το Αντικείμενο W_5=6 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 2

s: (W_8b1, W_3b2, W_4b3, W_6b3, W_1b4, W_{11}b4, W_2b5, W_{10}b5, W_5b2)

ΕΠΙΛΥΣΗ ΜΕ ΔΙΑΦΟΡΕΤΙΚΟ ΚΡΙΤΗΡΙΟ ΕΠΙΛΟΓΗΣ

- Επανάληψη 10. Το Αντικείμενο W_9=6 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 5

s: (W_8b1, W_3b2, W_4b3, W_6b3, W_1b4, W_{11}b4, W_2b5, W_{10}b5, W_5b2, W_9b5).

- Επανάληψη 11. Το Αντικείμενο W_7=2 αντιστοιχίζεται στον Αποθηκευτικό Χώρο 5

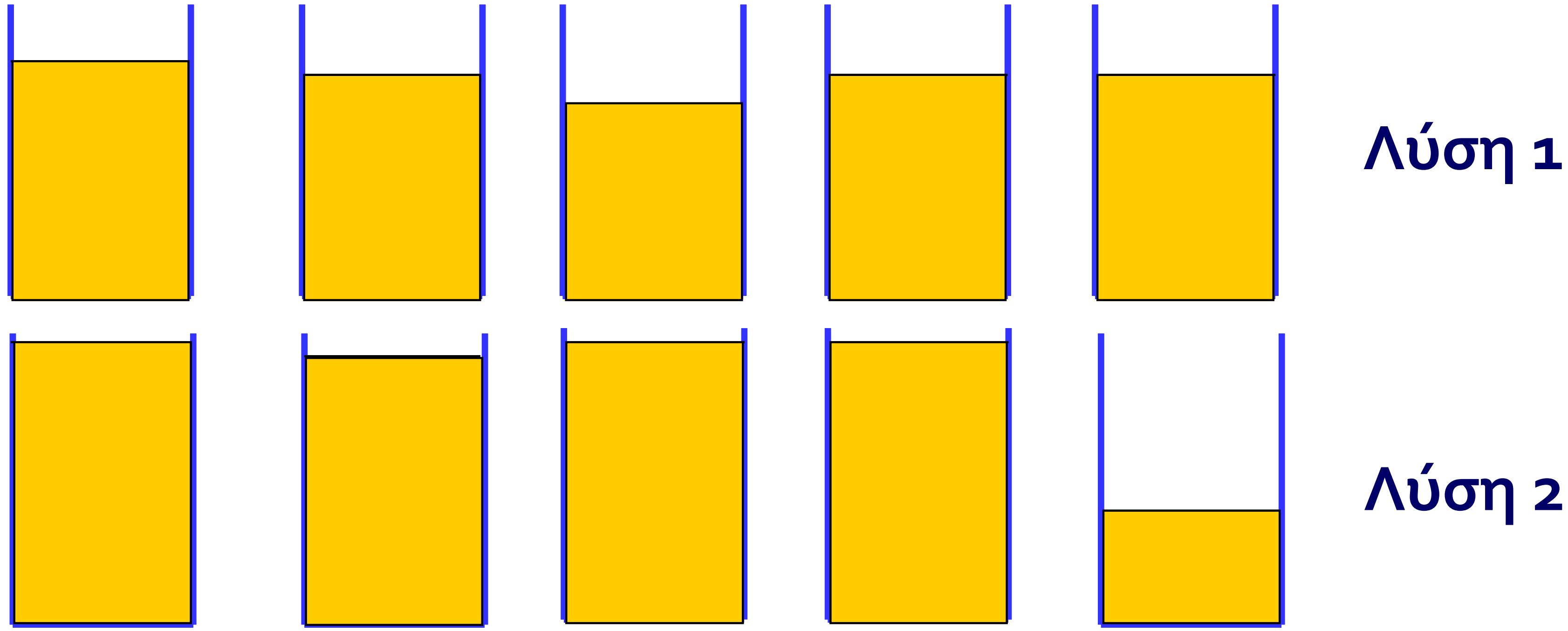
s: (W_8b1, W_3b2, W_4b3, W_6b3, W_1b4, W_{11}b4, W_2b5, W_{10}b5, W_5b2, W_9b5, W_7b3)

s :

b1:	$\mathbf{W_8}$		
b2:	$\mathbf{W_3}$	$\mathbf{W_5}$	
b3:	$\mathbf{W_4}$	$\mathbf{W_6}$	$\mathbf{W_7}$
b4:	$\mathbf{W_1}$	$\mathbf{W_{11}}$	
b5:	$\mathbf{W_2}$	$\mathbf{W_{10}}$	$\mathbf{W_9}$

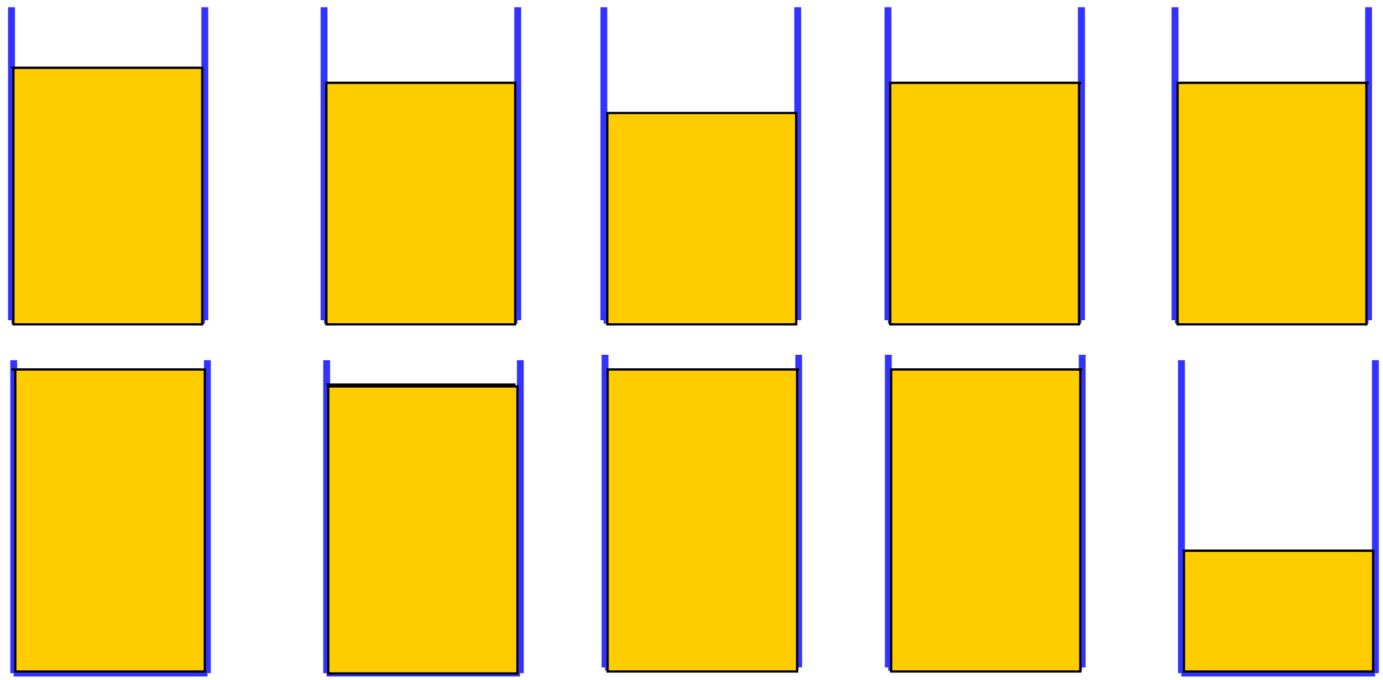
Αξιολόγηση της τελικής λύσης $z(s) = 5$

Εναλλακτική αντικειμενική συνάρτηση για το Bin Packing



Θεώρησε ένα bin packing πρόβλημα και δυο λύσεις του ίδιου αυτού προβλήματος. Οι δύο αυτές λύσεις έχουν κατασκευαστεί από δύο Διαφορετικά κριτήρια επιλογής. Σύμφωνα με το κλασικό κριτήριο αξιολόγησης ενός Προβλήματος bin packing, δηλαδή τον υπολογισμό των bins που χρησιμοποιούνται, οι δυο αυτές λύσεις έχουν την ίδια ποιότητα, καθώς και οι δύο χρησιμοποιούν 5 αποθηκευτικές χώρους.

Εναλλακτική αντικειμενική συνάρτηση για το Bin Packing



Λύση 1

Λύση 2

Ωστόσο, σε πλήθος εφαρμογών η Λύση 2 υπερέχει σε σύγκριση με την Λύση 1, λόγω της καλύτερης αξιοποίησης των bins που χρησιμοποιούνται.
Συνεπώς, υπάρχει ανάγκη να διατυπωθεί ένα εναλλακτικό κριτήριο αξιολόγησης, διαφορετικού του κλασικού, η εφαρμογή του οποίου να αντανακλά την υπεροχή της Λύσης 2 σε σχέση με τη Λύση 1

Εναλλακτική αντικειμενική συνάρτηση για το Bin Packing

- Εναλλακτικό Κριτήριο Αξιολόγησης της λύσης
- Έστω μία λύση s η οποία έχει τα εξής χαρακτηριστικά
  - Περιλαμβάνει p αποθηκευτικούς χώρους
  - Έστω u_i ο χώρος που καταλαμβάνουν τα αντικείμενα τα οποία είναι τοποθετημένα στον αποθηκευτικό χώρο $i = 1, \dots, p$
  - Ο αποθηκευτικός χώρος $i = 1, \dots, p$, φιλοξενεί q_i αντικείμενα
  - Έστω t_{ij} το μέγεθος του j-στου αντικειμένου που έχει αποθηκευτεί στον αποθηκευτικό χώρο i, $(i = 1, \dots, p, \kappa\alpha\iota\ j = 1, \dots, q_i)$
- Το κριτήριο αξιολόγησης

$$z(s) = \sum_{i=1}^{p} {u_i}^2 = \sum_{i=1}^{p} \left(\left(\sum_{j=1}^{q_i} t_{ij} \right)^2 \right)$$

αυξάνεται όσο αυξάνεται η πληρότητα των αποθηκευτικών χώρων

Εναλλακτική αντικειμενική συνάρτηση για το Bin Packing

- Επομένως, η μεγιστοποίηση του κριτηρίου αξιολόγησης
  - Ευνοεί την παραγωγή λύσεων με τη χρήση των ελάχιστων αποθηκευτικών χώρων
  - Ευνοεί την αύξηση της πληρότητας των χρησιμοποιούμενων αποθηκευτικών χώρων
- Ποιος εκ των First-Fit και Best-Fit αλγορίθμων είναι πιο συμβατός με τη μεγιστοποίηση της αντικειμενικής συνάρτησης:

$$z(s) = \sum_{i=1}^{p} \left(\left(\sum_{j=1}^{q_i} t_{ij} \right)^2 \right)$$

  - Ο Best-Fit αλγόριθμος καθώς μεγιστοποιεί την πληρότητα των χρησιμοποιούμενων αποθηκευτικών χώρων

Εναλλακτική αντικειμενική συνάρτηση για το Bin Packing

- Η αξιολόγηση της ολοκληρωμένης που κατασκευάστηκε στο bin packing problem μέσω της First Fit μεθόδου, εφαρμόζοντας το κριτήριο του αθροίσματος των τετραγώνων της χρήσης (εναλλακτικό κριτήριο) είναι:

s :

b1:	$\mathbf{W_8}$	$\mathbf{W_7}$	
b2:	$\mathbf{W_3}$	$\mathbf{W_5}$	
b3:	$\mathbf{W_4}$	$\mathbf{W_6}$	
b4:	$\mathbf{W_1}$	$\mathbf{W_{11}}$	
b5:	$\mathbf{W_2}$	$\mathbf{W_{10}}$	$\mathbf{W_9}$

Αντικειμενική Συνάρτηση της λύσης:

$$z(s) = 17^2 + 20^2 + 18^2 + 16^2 + 20^2 = 1669$$

Εναλλακτικό κριτήριο αξιολόγησης για το Bin Packing

- Η αξιολόγηση της ολοκληρωμένης που κατασκευάστηκε στο bin packing problem μέσω της Best Fit μεθόδου, εφαρμόζοντας το κριτήριο του αθροίσματος των τετραγώνων της χρήσης (εναλλακτικό κριτήριο) είναι:

s :

b1:	$\mathbf{w_8}$		
b2:	$\mathbf{w_3}$	$\mathbf{w_5}$	
b3:	$\mathbf{w_4}$	$\mathbf{w_6}$	$\mathbf{w_7}$
b4:	$\mathbf{w_1}$	$\mathbf{w_{11}}$	
b5:	$\mathbf{w_2}$	$\mathbf{w_{10}}$	$\mathbf{w_9}$

Αντικειμενική Συνάρτηση της λύσης:

$$\mathbf{z}(s) = \mathbf{15^2 + 20^2 + 20^2 + 16^2 + 20^2 = 1681}$$

Πρόβλημα

- Οι διαδικασίες διανομής των προϊόντων καθορίζουν σε μεγάλο βαθμό την ποιότητα και την αποτελεσματικότητα των υπηρεσιών της εφοδιαστικής αλυσίδας.
- Γαλακτοβιομηχανία διαθέτει οκτώ (8) ομοιογενή φορτηγά
- Η βιομηχανία μπορεί να χρησιμοποιήσει έως 8 από τα οχήματα **της, τα οποία βρίσκονται στο σημείο παραγωγής (Κόμβος 0) ώστε να ικανοποιήσει 9 παραγγελίες πελατών**
- **Πελάτες: Κόμβοι {1,2,…,9}**
- Η χωρητικότητα κάθε οχήματος είναι 17.000 lt.

Πρόβλημα

- Τα οχήματα ξεκινούν από την κεντρική αποθήκη και αφού ικανοποιήσουν τη ζήτηση των πελατών, επιστρέφουν στην κεντρική αποθήκη
- Ένα όχημα μπορεί να εξυπηρετήσει περισσότερους από έναν πελάτες
- Ένας πελάτης εξυπηρετείται από ακριβώς μία επίσκεψη ενός οχήματος
- Η συνολική ζήτηση των πελατών που εξυπηρετεί ένα όχημα δεν μπορεί να ξεπερνά τη χωρητικότητα του οχήματος.

Πρόβλημα

- Οι αποστάσεις ανάμεσα σε κάθε ζεύγος κόμβων του δικτύου δίνονται στον παρακάτω Πίνακα:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	21	56	89	56	25	64	41	48	55
1	21	0	29	34	52	36	12	55	35	67
2	56	29	0	32	72	56	64	81	89	26
3	89	34	32	0	36	64	81	95	99	18
4	56	52	72	36	0	29	55	47	65	39
5	25	36	56	64	29	0	40	56	62	46
6	64	12	64	81	55	40	0	29	31	43
7	41	55	81	95	47	56	29	0	11	46
8	48	35	89	99	65	62	31	11	0	36
9	55	67	26	18	39	46	43	46	36	0

- Η ζήτηση του κάθε πελάτη περιγράφεται στον παρακάτω Πίνακα:

Πελάτης	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Ζήτηση	2.000	4.000	5.000	1.000	7.000	3.000	5.000	4.000	4.000

Minimum Cut Stocking Problem

- Παραγγελίες πελατών: Σύνολα αντικειμένων με δεδομένο μέγεθος
- Πρώτη Ύλη: Αντικείμενα με δεδομένο μέγεθος
  - Φύλλα Χαρτιού
  - Μεταλλικά Φύλλα
  - Ξύλινες Επιφάνειες
- Σκοπός:
  - Ελαχιστοποίηση κόστους πρώτης ύλης για την παραγωγή όλων των παραγγελιών
  - Ανάθεση των παραγγελιών στα ελάχιστα αντικείμενα της πρώτης ύλης

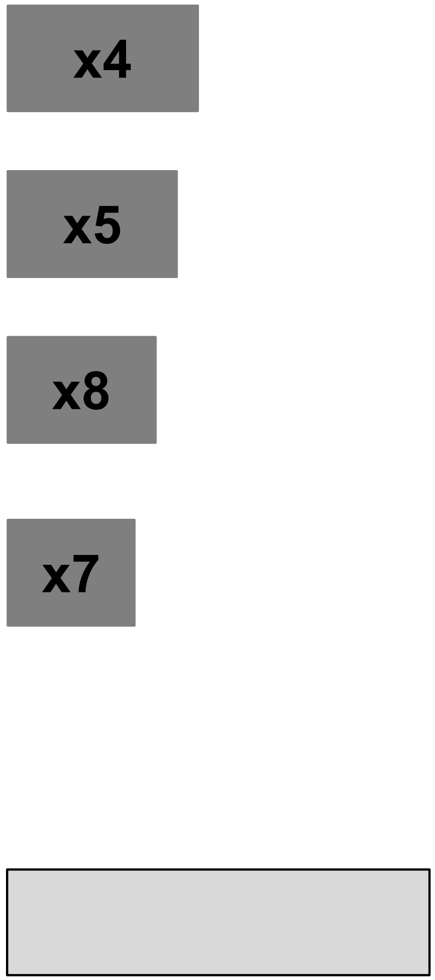
Minimum Cut Stocking Problem

- Το Minimum Cut Stocking Problem είναι πολύ στενά συνδεδεμένο με το 1-D Bin Packing Problem
  - Ελαχιστοποίηση αντικειμένων τα οποία θα φιλοξενήσουν μικρότερα αντικείμενα
- Οι παραγγελίες αποτελούνται από σύνολα πολλαπλών αντικειμένων με κοινό μέγεθος:
  - Ένας αριθμός αντικειμένων με μέγεθος w_1
  - Ένας αριθμός αντικειμένων με μέγεθος w_2
  - Ένας αριθμός αντικειμένων με μέγεθος w_3
  - …
- Πολλαπλά ομοειδή αντικείμενα

Minimum Cut Stocking Problem

- Παραγγελίες:
  - 4 αντικείμενα πλάτους 12
  - 5 αντικείμενα πλάτους 9
  - 8 αντικείμενα πλάτους 6
  - 7 αντικείμενα πλάτους 4

- Πρώτη ύλη:
  - Φύλλα αντικειμένων πλάτους 20



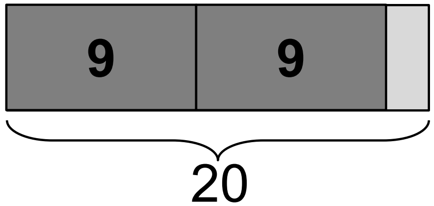
Minimum Cut Stocking Problem

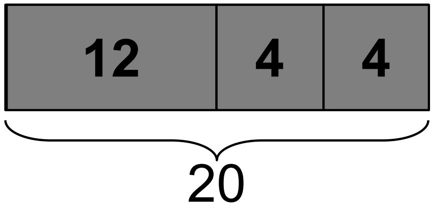
- Το Minimum Cut Stocking Problem θα μπορούσε να αντιμετωπιστεί ως ένα Bin Packing Problem
  – 24 αντικείμενα με διάφορα μεγέθη:
    - 4 αντικείμενα πλάτους 12
    - 5 αντικείμενα πλάτους 9
    - 8 αντικείμενα πλάτους 6
    - 7 αντικείμενα πλάτους 4
  – Ελαχιστοποίηση των αντικειμένων μεγέθους 20 που θα φιλοξενήσουν τα 24 αντικείμενα

- Τι δεν λαμβάνει υπόψιν αύτη η προσέγγιση;

Minimum Cut Stocking Problem

- Αντιμετωπίζοντας το πρόβλημα σαν ένα Bin Packing πρόβλημα δεν εκμεταλλευόμαστε το γεγονός πως οι παραγγελίες είναι ομαδοποιημένες
- Η ομαδοποιημένη φύση των παραγγελιών αλλάζει ιδιαίτερα τη φύση του προβλήματος
  - Μας οδηγεί να σκεφτούμε καλές ομαδοποιήσεις (patterns)
  - Μπορούμε να εξετάσουμε ένα σύνολο δυνατών ομαδοποιήσεων (patterns) των αντικειμένων στο αντικείμενο της πρώτης ύλης
- Παράδειγμα cutting patterns:





Minimum Cut Stocking Problem

- Ο προσδιορισμός καλών patterns μπορεί να μικρύνει το μέγεθος του προβλήματος σε σχέση με την αντιμετώπιση τους ως ένα bin packing πρόβλημα
- Επίσης μπορεί να βελτιώσει το μυωπικό χαρακτηριστικό των bin packing αλγορίθμων στους οποίους τα αντικείμενα επιλέγονται διαδοχικά για να τοποθετηθούν.
  - Με τον προσδιορισμό καλών packing patterns μπορούμε να δούμε “πιο μακριά” κατά την επίλυση του προβλήματος

Minimum Cut Stocking Problem

- Η έννοια των patterns χρησιμοποιείται και για τη μοντελοποίηση του προβλήματος
- Σύνολο από patterns $P = p_1, p_2, \dots, p_k$
- Σύνολο παραγγελιών $D = d_1, d_2, \dots, d_m$
  - Ζητούνται d_1 αντικείμενα τύπου 1, d_2 αντικείμενα τύπου 2, ...
- Ορίζεται ως $\alpha_{ij}, i = 1, \dots, m, j = 1, \dots, k,$ οι φορές που ένα αντικείμενο τύπου i εμφανίζεται στο pattern j
- $x_i, i = 1, \dots, k$: οι φορές που χρησιμοποιείται το pattern p_i

Minimum Cut Stocking Problem

- Το μοντέλο μπορεί να γραφτεί ως:

$$min \sum_{i=1}^{k} x_i$$

(Ελαχιστοποίηση των μονάδων πρώτης ύλης)

- Με περιορισμούς

$$\sum_{i=1}^{k} a_{ij} \cdot x_i \geq d_j \,, \forall j = 1, \dots, m$$

(Το πλήθος των παραγόμενων αντικειμένων τύπου j είναι τουλάχιστον ίσα με το ύψος της παραγγελίας d_j)

$$x_i \geq 0, \forall i = 1, \dots, k$$

(Ακέραιο Πλήθος χρησιμοποιούμενων patterns)

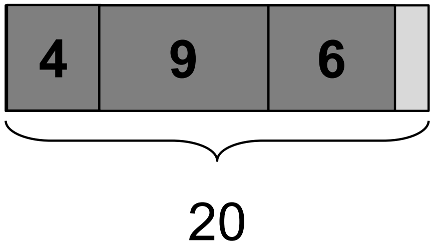
Minimum Cut Stocking Problem

Σχόλια:

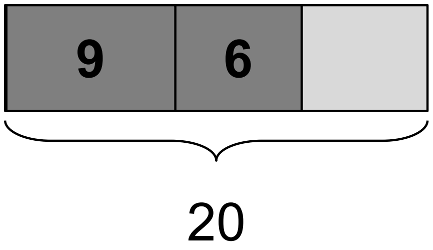
- Αν στον περιορισμό $\sum_{i=1}^{k} a_{ij} \cdot x_i \geq d_j$ χρησιμοποιηθεί ισότητα το πρόβλημα είναι ίδιο με το bin packing.
- Ο περιορισμός ‘$\geq$’ απλά επιτρέπει την παραγωγή παραπάνω κομματιών από τις παραγγελίες. Το πρόβλημα όμως αποσκοπεί στην ελαχιστοποίησης των κομματιών πρώτης ύλης, επομένως αν παραχθούν παραπάνω παραγγελίες, αυτό θα γίνει γιατί χρησιμοποιώντας ένα αποτελεσματικό pattern αυτό μπορεί να περιέχει και κάποιους τύπου αντικειμένων για τα οποία η ζήτηση έχει καλυφθεί

Σχεδιασμός ενός Greedy Αλγορίθμου για το Minimum Cutting Stock Problem

- Με ποια patterns πρέπει να ασχοληθούμε;
  - Πρέπει να δημιουργήσουμε όλα τα patterns όλες τις δυνατές ομάδες απαιτούμενων αντικειμένων με συνολικό μέγεθος ίσο ή μικρότερο από αυτό της πρώτης ύλης
  - Προφανώς, δεν υπάρχει νόημα να ασχολούμαστε με patterns τα οποία έχουν φύρα ίση ή μεγαλύτερη από το μικρότερο τύπο αντικειμένου (χρήση πρώτης ύλης άσκοπα)
  - Το Pattern B κυριαρχεί του Pattern A



Pattern A
(waste = 1)



Pattern B
(waste = 5)

Dominating Patterns

- Κάθε pattern το οποίο έχει φύρα μικρότερη από το ελάχιστο μέγεθος των αντικειμένων ονομάζεται dominating pattern
- Με άλλα λόγια, ένα pattern **δεν είναι** dominating όταν σε αυτό θα μπορούσε να φιλοξενηθεί τουλάχιστον ένα ακόμα αντικείμενο

Εντοπισμός όλων των Dominating Patterns

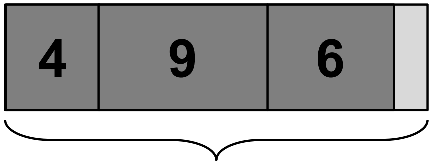
- Ένας απλός τρόπος είναι να εξετάσουμε όλα τα μείγματα τύπων αντικειμένων και να διατηρήσουμε μόνο αυτά τα οποία
  - Μπορούν να χωρέσουν σε μία μονάδα πρώτης ύλης
  - Είναι dominating
- Για τον έλεγχο όλων των μειγμάτων μπορούμε να εξετάσουμε όλους τους συνδυασμούς με επανατοποθέτηση των τύπων αντικειμένων
- Μέχρι πόσα αντικείμενα μπορεί να περιέχει κάθε ένας από τους ελεγχόμενους συνδυασμούς;
  - Από 1
  - Μέχρι Χωρητικότητα / Ελάχιστο μέγεθος αντικειμένων

Επιλογή Patterns μέχρι την πλήρη ικανοποίηση των παραγγελιών

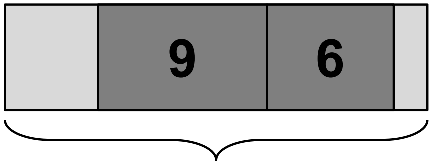
- Επαναληπτικός αλγόριθμος
- Όσο δεν έχει ικανοποιηθεί η ζήτηση όλων των τύπων αντικειμένων
  - Επιλογή του pattern ($p*$) που ελαχιστοποιεί τη φύρα
  - Ανανέωση της ακάλυπτης ζήτησης βάσει της επιλεγμένης παρτίδας ($p*$)

Επιλογή Patterns μέχρι την πλήρη ικανοποίηση των παραγγελιών

- Προσοχή στον υπολογισμό της φύρας κάθε pattern κατά την εκτέλεση του αλγορίθμου:
  - Για τον υπολογισμό της φύρας δεν πρέπει να λαμβάνονται υπόψιν τα αντικείμενα που ανήκουν στο pattern αλλά η ζήτηση τους έχει ήδη ικανοποιηθεί!
  - Π.χ. αν έχω το παρακάτω pattern και **εφόσον η ζήτηση των αντικειμένων με μέγεθος 4 έχει ικανοποιηθεί**, η φύρα του pattern πρέπει να υπολογιστεί ίση με 5 (και όχι ίση με ένα):



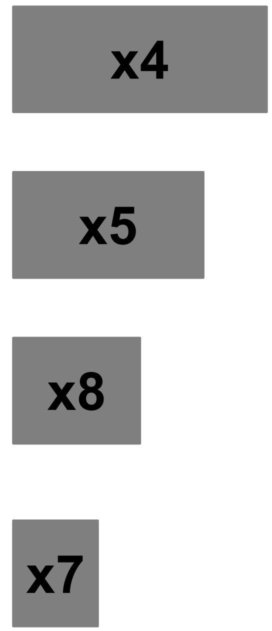
(original waste = 1)



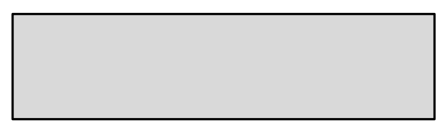
(current waste = 5)

Minimum Cut Stocking Problem – Greedy Algorithm

- Παραγγελίες
  - 4 αντικείμενα πλάτους 12
  - 5 αντικείμενα πλάτους 9
  - 8 αντικείμενα πλάτους 6
  - 7 αντικείμενα πλάτους 4

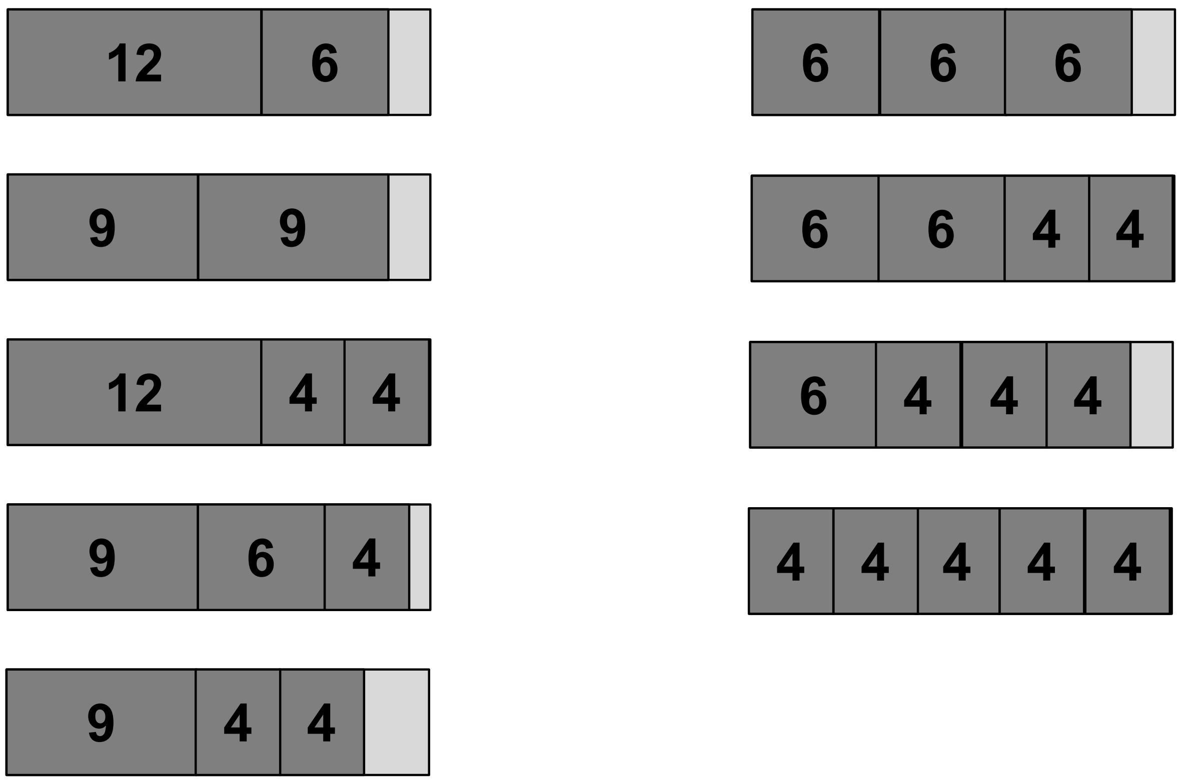


- Πρώτη ύλη:
  - Φύλλα αντικειμένων πλάτους 20



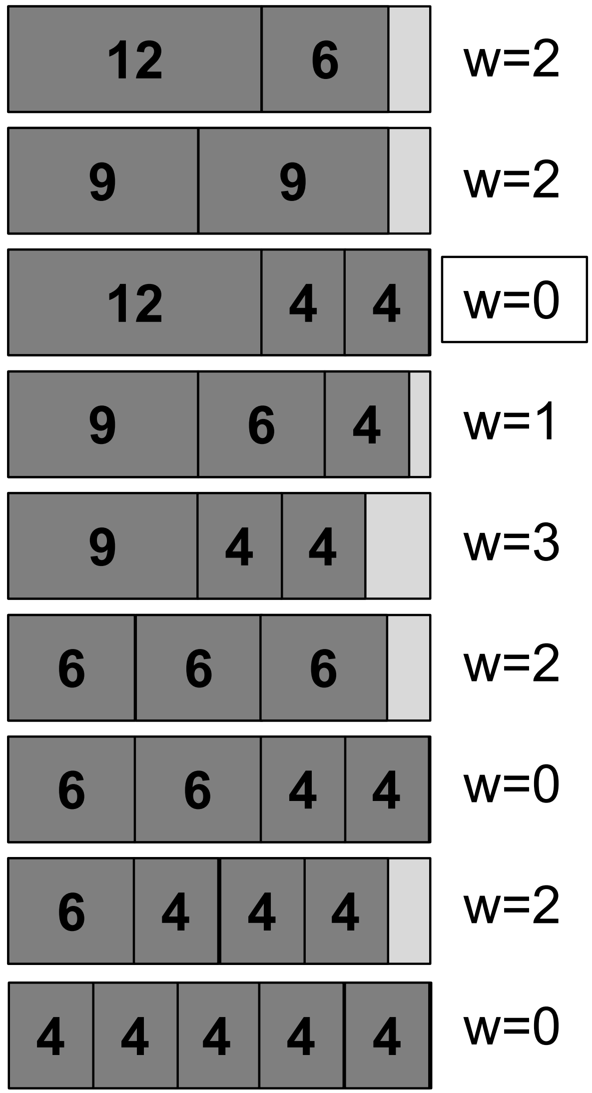
Minimum Cut Stocking Problem – Greedy Algorithm

- Προσδιορισμός Dominating Patterns
- 9 Dominating Patterns



Minimum Cut Stocking Problem – Greedy Algorithm

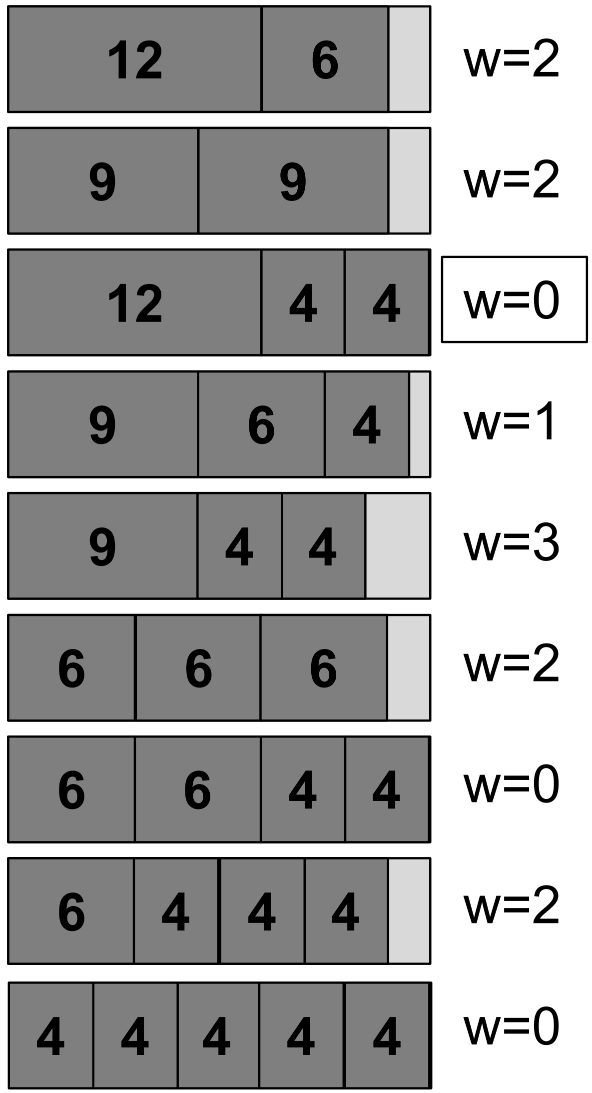
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 4 αντικείμενα
  - 9: 5 αντικείμενα
  - 6: 8 αντικείμενα
  - 4: 7 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: **3** αντικείμενα
  - 9: 5 αντικείμενα
  - 6: 8 αντικείμενα
  - 4: **5** αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

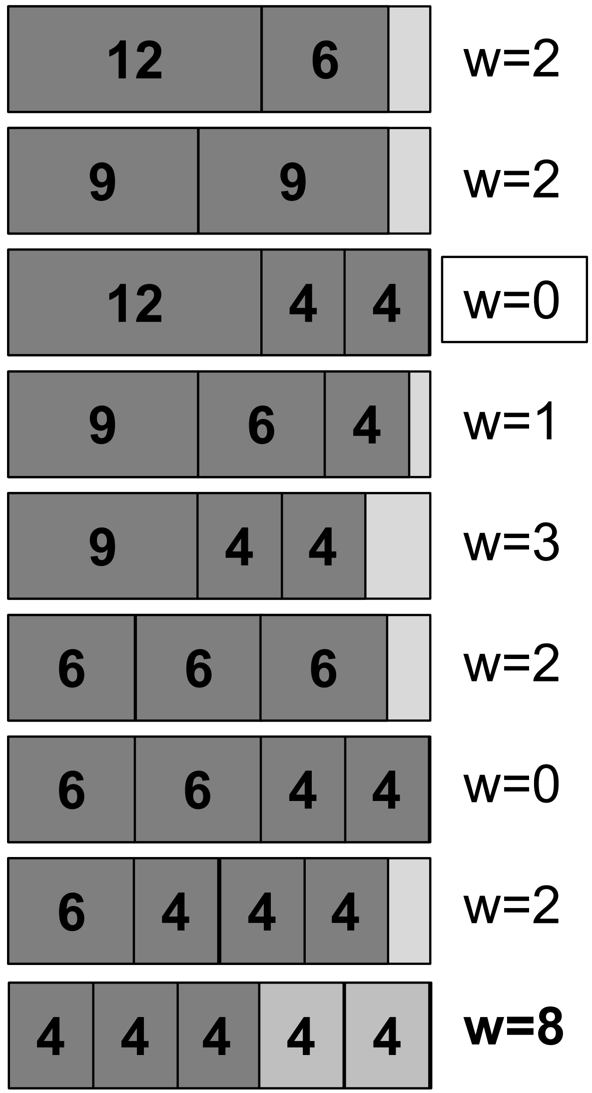
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 3 αντικείμενα
  - 9: 5 αντικείμενα
  - 6: 8 αντικείμενα
  - 4: 5 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: **2** αντικείμενα
  - 9: 5 αντικείμενα
  - 6: 8 αντικείμενα
  - 4: **3** αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

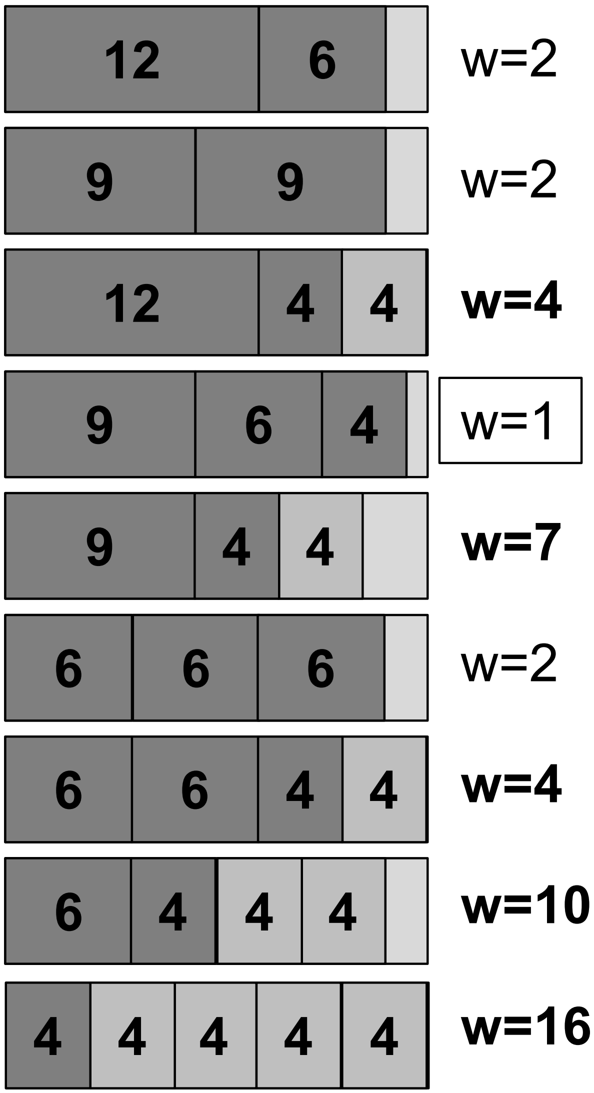
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 2 αντικείμενα
  - 9: 5 αντικείμενα
  - 6: 8 αντικείμενα
  - 4: 3 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: **1** αντικείμενα
  - 9: 5 αντικείμενα
  - 6: 8 αντικείμενα
  - 4: **1** αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

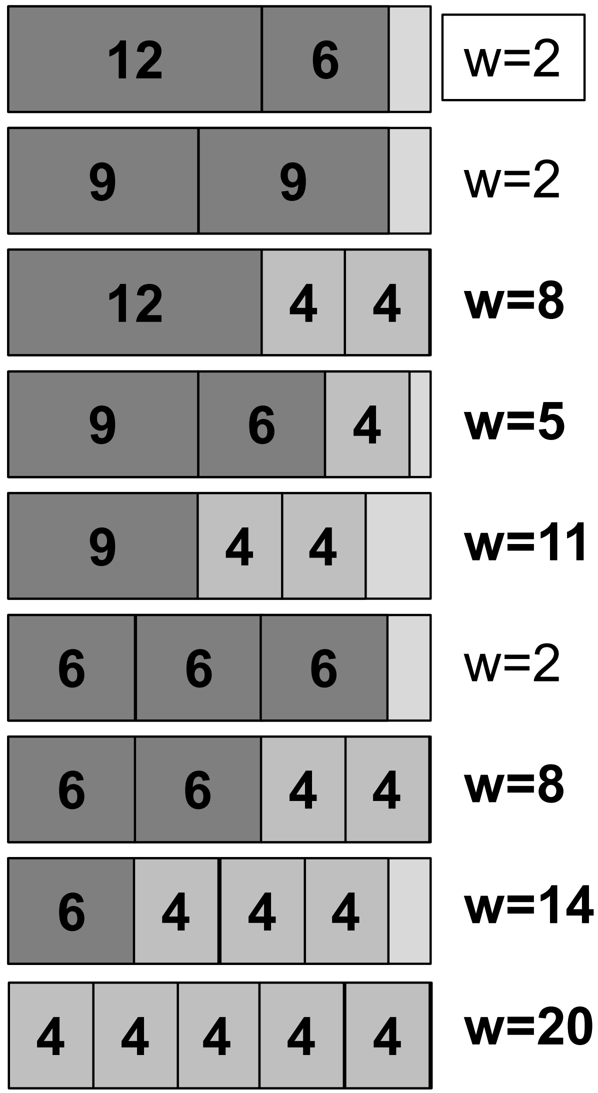
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 1 αντικείμενα
  - 9: 5 αντικείμενα
  - 6: 8 αντικείμενα
  - 4: 1 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: 1 αντικείμενα
  - 9: **4** αντικείμενα
  - 6: **7** αντικείμενα
  - 4: **0** αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

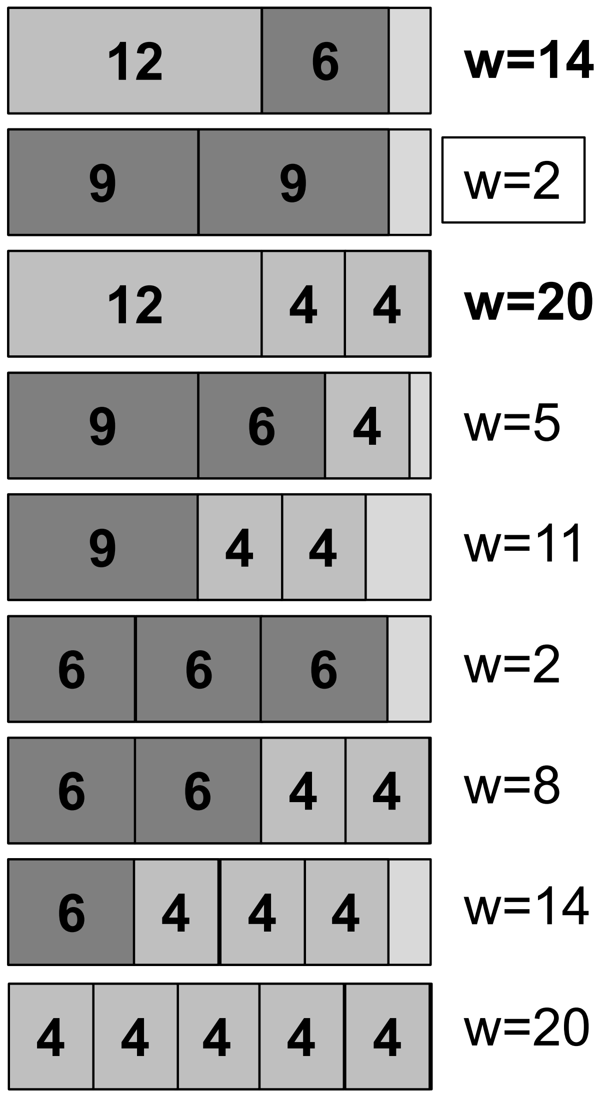
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 1 αντικείμενα
  - 9: 4 αντικείμενα
  - 6: 7 αντικείμενα
  - 4: 0 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: 0 αντικείμενα
  - 9: **4** αντικείμενα
  - 6: **6** αντικείμενα
  - 4: **0** αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

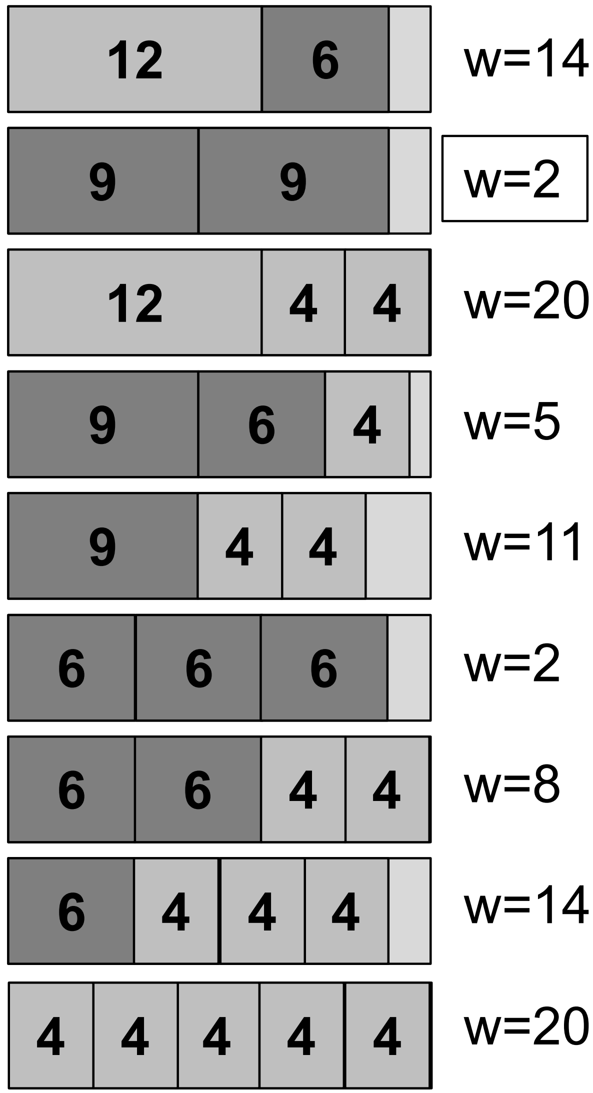
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 0 αντικείμενα
  - 9: 4 αντικείμενα
  - 6: 6 αντικείμενα
  - 4: 0 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: 0 αντικείμενα
  - 9: **2** αντικείμενα
  - 6: **6** αντικείμενα
  - 4: **0** αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

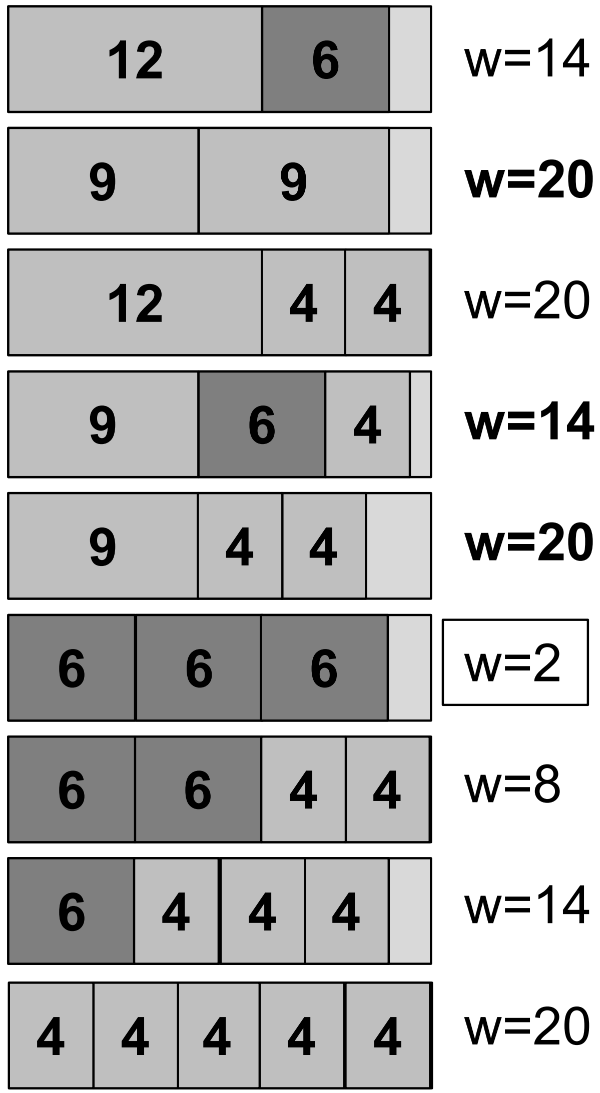
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 0 αντικείμενα
  - 9: 2 αντικείμενα
  - 6: 6 αντικείμενα
  - 4: 0 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: 0 αντικείμενα
  - 9: **0** αντικείμενα
  - 6: 6 αντικείμενα
  - 4: 0 αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

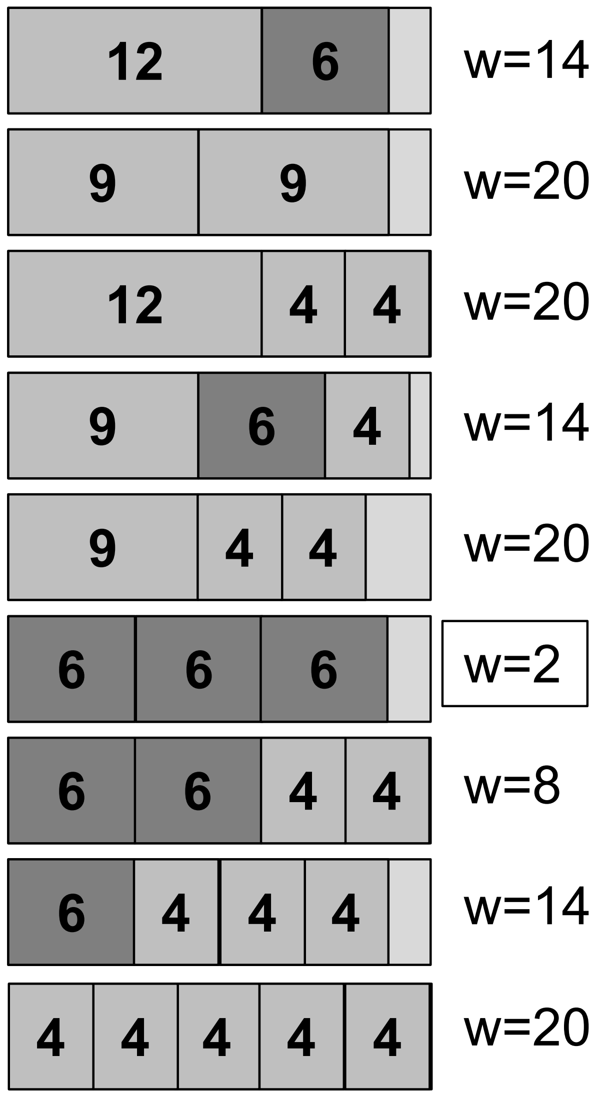
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 0 αντικείμενα
  - 9: 0 αντικείμενα
  - 6: 6 αντικείμενα
  - 4: 0 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: 0 αντικείμενα
  - 9: 0 αντικείμενα
  - 6: 3 αντικείμενα
  - 4: 0 αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

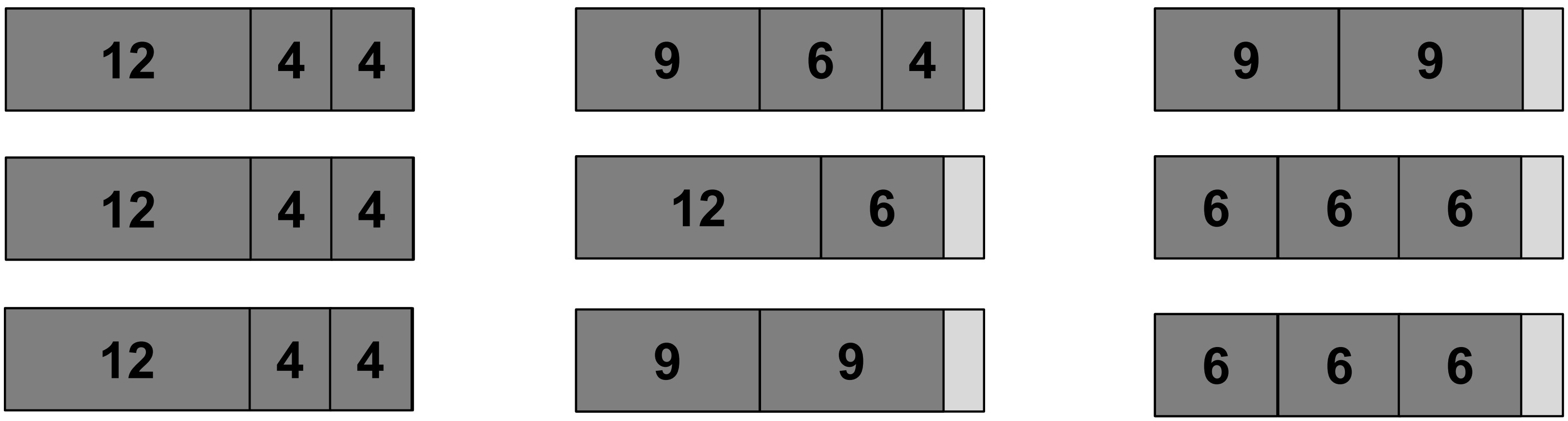
- Επεξεργασία Παραγγελιών
- Επανάληψη 1
- Εκκρεμείς Παραγγελίες
  - 12: 0 αντικείμενα
  - 9: 0 αντικείμενα
  - 6: 3 αντικείμενα
  - 4: 0 αντικείμενα



- Ανανέωση Εκκρεμών Παραγγελιών
  - 12: 0 αντικείμενα
  - 9: 0 αντικείμενα
  - 6: **0** αντικείμενα
  - 4: 0 αντικείμενα

Minimum Cut Stocking Problem – Greedy Algorithm

- Τελική Λύση:
- 9 μονάδες πρώτης ύλης



Πρόβλημα

- Καλούμαστε να επιτύχουμε δύο στόχους:
  - Πρωταρχικός σκοπός: Να ελαχιστοποιηθεί ο αριθμός των οχημάτων που θα χρησιμοποιηθούν για την εξυπηρέτηση των 9 πελατών της εταιρείας
  - Δευτερεύων σκοπός: Να ελαχιστοποιηθούν τα διανυόμενα χιλιόμετρα των χρησιμοποιούμενων οχημάτων
- Προτείνετε μία κατάλληλη αντικειμενική συνάρτηση για το πρόβλημα
- Να σχεδιαστεί και να εφαρμοστεί (να εκτελεστούν οι επαναλήψεις) μια πλεονεκτική στρατηγική επίλυσης (ΠΣΕ)

ΔΙΟΙΚΗΤΙΚΗ ΤΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ & ΠΡΟΒΛΗΜΑΤΑ ΑΝΤΙΣΤΟΙΧΙΣΗΣ

- Στην κατοχή της ναυτιλιακής εταιρίας βρίσκονται πέντε πλοία
- Ανάλογα με τα χαρακτηριστικά τους,
  - Scrubbers
  - Ικανότητα Διέλευσης από τη Διώρυγα του Σουέζ
  - Οικονομία Κινητήρα
  - Παγοθραυστικές Δυνατότητες

  κάθε πλοίο έχει διαφορετικό κόστος παραμονής σε κάθε μία από τις διαφορετικές περιοχές
- Για να εξασφαλίζεται παγκόσμια παρουσία της εταιρίας μας, τα πλοία της εταιρίας πρέπει να βρίσκονται σε διαφορετικές γεωγραφικές περιοχές
- Το εμπορικό τμήμα της εταιρίας μας κατασκεύασε έναν πίνακα με την κερδοφορία κάθε πλοίου μας για κάθε μία από τις πέντε διαφορετικές περιοχές του πλανήτη

ΠΡΟΒΛΗΜΑ ΑΝΤΙΣΤΟΙΧΗΣΗΣ: The linear sum assignment

Πλοίο

Περιοχή

	A	B	C	D	E
Med	38	53	61	36	66
Continent	100	60	9	79	34
Baltic	30	37	36	72	24
Arabian Gulf	61	95	21	14	64
W. Africa	89	90	4	5	79

ΠΡΟΒΛΗΜΑ ΑΝΤΙΣΤΟΙΧΗΣΗΣ: The linear sum assignment

- Σκοπός που τίθεται είναι ο ορισμός μία περιοχής για κάθε ένα από τα διαθέσιμα πλοία της εταιρίας μας

Περιορισμοί

- Κάθε περιοχή πρέπει να φιλοξενεί ακριβώς ένα πλοίο

Αντικειμενική Συνάρτηση

- Το συνολικό ημερήσιο κόστος για τη λειτουργία των καραβιών ελαχιστοποιείται

ΠΡΟΒΛΗΜΑ ΑΝΤΙΣΤΟΙΧΗΣΗΣ: The linear sum assignment

- Το συγκεκριμένο πρόβλημα είναι ένα πρόβλημα αντιστοίχισης
- Σύνολο 1: Περιοχές
- Σύνολο 2: Πλοία
- Κάθε ένα στοιχείο του συνόλου Α πρέπει να αντιστοιχηθεί με ακριβώς ένα στοιχείο του συνόλου Β
- Στόχος το άθροισμα του κόστους των αντιστοιχίσεων ελαχιστοποιείται

ΠΡΟΒΛΗΜΑ ΑΝΤΙΣΤΟΙΧΗΣΗΣ: The linear sum assignment

- Μορφή μία Λύσης

$$s = \{(Med, ship1), (Cont, ship2), (Baltic, ship3), (AG, ship4), (WAfr, ship5)\}$$

- Πόσες είναι οι δυνατές λύσεις;
  - Αν κλειδώσουμε τις περιοχές στο διάνυσμα της λύσης μας

$$s = \{(\boldsymbol{Med}, ***), (\boldsymbol{Cont}, ***), (\boldsymbol{Baltic}, ***), (\boldsymbol{AG}, ***), (\boldsymbol{WAfr}, ***)\}$$

  - Κάθε διάταξη πλοίων αποτελεί και μία διαφορετική λύση του προβλήματος μας (μία μοναδική αντιστοίχιση)
  - Επομένως, οι δυνατές λύσεις είναι 5!

ΠΡΟΒΛΗΜΑ ΑΝΤΙΣΤΟΙΧΗΣΗΣ: The linear sum assignment

- Αντικειμενική Συνάρτηση:

Η λύση

$$s = \{(Med, A), (Cont, C), (Baltic, B), (AG, D), (WAfr, E)\}$$

Έχει συνολικό κόστος:

$$z(S) = 38 + 9 + 37 + 14 + 79$$

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

ΠΡΟΒΛΗΜΑ ΑΝΤΙΣΤΟΙΧΗΣΗΣ: The linear sum assignment

- Προτείνετε έναν πλεονεκτικό αλγόριθμο για την επίλυση του προβλήματος

Ένας Πλεονεκτικός Αλγόριθμος για το linear sum assignment

- Ένας πλεονεκτικός αλγόριθμος για το πρόβλημα, μπορεί να λειτουργεί ως εξής: Έστω R_{comp} το σύνολο των περιοχών και V_{comp} το σύνολο των πλοίων

$s = \{\}$
$V = \{\}$
$R = \{\}$
while $|s| < 5$
    Εντόπισε την αντιστοίχιση $(i, j), i \in R_{comp}, j \in V_{comp}, i \notin R, j \notin V$ με το ελάχιστο κόστος
    $s \longleftarrow s \cup \{(i, j)\}$
    $R \longleftarrow R \cup \{i\}$
    $V \longleftarrow V \cup \{j\}$

Ένας Πλεονεκτικός Αλγόριθμος για το linear sum assignment

- Επανάληψη 1

$$s = \{\}$$
$$V = \{\}$$
$$R = \{\}$$

$$(i, j) = (WAfr, C)$$

$$s = \{(WAfr, C)\}$$
$$V = \{C\}$$
$$R = \{WAfr\}$$

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

Ένας Πλεονεκτικός Αλγόριθμος για το linear sum assignment

- Επανάληψη 2

$$s = \{(WAfr, C)\}$$
$$V = \{C\}$$
$$R = \{WAfr\}$$

$$(i, j) = (AG, D)$$

$$s = \{(WAfr, C), (AG, D)\}$$
$$V = \{C, D\}$$
$$R = \{WAfr, AG\}$$

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

Ένας Πλεονεκτικός Αλγόριθμος για το linear sum assignment

- Επανάληψη 3

$$s = \{(WAfr, C), (AG, D)\}$$
$$V = \{C, D\}$$
$$R = \{WAfr, AG\}$$

$$(i, j) = (Baltic, G)$$

$$s = \{(WAfr, C), (AG, D), (Baltic, E)\}$$
$$V = \{C, D, G\}$$
$$R = \{WAfr, AG, Baltic\}$$

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

Ένας Πλεονεκτικός Αλγόριθμος για το linear sum assignment

- Επανάληψη 4

$$s = \{(WAfr, C), (AG, D), (Baltic, E)\}$$
$$V = \{C, D, E\}$$
$$R = \{WAfr, AG, Baltic\}$$

$$(i, j) = (Med, A)$$

$$s = \{(WAfr, C), (AG, D), (Baltic, E), (Med, A)\}$$
$$V = \{C, D, E, A\}$$
$$R = \{WAfr, AG, Baltic, Med\}$$

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

Ένας Πλεονεκτικός Αλγόριθμος για το linear sum assignment

- Επανάληψη 5

$$s = \{(WAfr, C), (AG, D), (Baltic, E), (Med, A)\}$$
$$V = \{C, D, E, A\}$$
$$R = \{WAfr, AG, Baltic, Med\}$$

$$(i, j) = (Cont, B)$$

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

$$s = \{(WAfr, C), (AG, D), (Baltic, E), (Med, A), (Cont, B)\}$$
$$V = \{C, D, E, A, B\}$$
$$R = \{WAfr, AG, Baltic, Med, Cont\}$$

Ένας Πλεονεκτικός Αλγόριθμος για το linear sum assignment

- Ο Αλγόριθμος τερματίζεται με τελική λύση την

$$s = \{(WAfr, C), (AG, D), (Baltic, E), (Med, A), (Cont, B)\}$$

Με κόστος

$$z(S) = 4 + 14 + 24 + 38 + 60 = 140$$

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

Hungarian Algorithm

- Είναι η λύση που παρασκευάστηκε η βέλτιστη λύση;
  - Δε γνωρίζουμε
  - Είναι μία καλή λύση η οποία κατασκευάστηκε γρήγορα βάσει ενός απλού πλεονεκτικού αλγορίθμου
- Για το συγκεκριμένο πρόβλημα έχει προταθεί ένας πλεονεκτικός αλγόριθμος (πιο σύνθετος από αυτόν που μόλις παρουσιάσαμε), ο οποίος έχει τη δυνατότητα να παράγει τις βέλτιστες λύσεις
- Ο Αλγόριθμος προτάθηκε από τους Kuhn & Munkres algorithm
- Βασίζεται στην ερευνητική δραστηριότητα των Ούγγρων Dénes Kőnig and Jenő Egerváry (Hungarian Algorithm)
- Αντιμετωπίζει προβλήματα ελαχιστοποίησης κόστους σε τετραγωνικούς πίνακες
- <u>Σχόλιο</u>: Θα συζητήσουμε πώς αντιμετωπίζονται προβλήματα μεγιστοποίησης κέρδους και προβλήματα ορισμένα σε μη-τετραγωνικούς πίνακες

Hungarian Algorithm

- Βήματα του Αλγορίθμου

Βήμα 1. Βρες την ελάχιστη τιμή κάθε γραμμής και αφαίρεσέ την από όλα τα στοιχεία της γραμμής

Βήμα 2. Βρες την ελάχιστη τιμή κάθε στήλης και αφαίρεσέ την από όλα τα στοιχεία της στήλης

Βήμα 3. Κάλυψε όλα τα μηδενικά με τον ελάχιστο αριθμό γραμμών (σε γραμμές και στήλες)

Αν ο αριθμός των γραμμών είναι μικρότερος του n, Εκτέλεσε Βήμα 4

Βήμα 4. Εντόπισε την ελάχιστη τιμή (θ) η οποία δεν είναι καλυμμένη από οποιαδήποτε γραμμή.

Αφαίρεσε το θ από όλα τα ακάλυπτα στοιχεία

Αν υπάρχει στοιχείο το οποίο είναι καλυμμένο από δύο γραμμές πρόσθεσε σε αυτό θ και πήγαινε στο Βήμα 3

Hungarian Algorithm

Βήμα 1. Βρες την ελάχιστη τιμή κάθε γραμμής και αφαίρεσέ τη από όλα τα στοιχεία της γραμμής

	A	B	C	D	E
Med	38	53	61	**36**	66
Cont	100	60	**9**	79	34
Baltic	30	37	36	72	**24**
AG	61	95	21	**14**	64
WAfr	89	90	**4**	5	79

	A	B	C	D	E
Med	2	17	25	0	30
Cont	91	51	0	70	25
Baltic	6	13	12	48	0
AG	47	81	7	0	50
WAfr	85	86	0	1	75

Hungarian Algorithm

Βήμα 2. Βρες την ελάχιστη τιμή κάθε στήλης και αφαίρεσέ την από όλα τα στοιχεία της στήλης

	A	B	C	D	E
Med	**2**	17	25	**0**	30
Cont	91	51	**0**	70	25
Baltic	6	**13**	12	48	**0**
AG	47	81	7	**0**	50
WAfr	85	86	0	1	75

	A	B	C	D	E
Med	0	4	25	0	30
Cont	89	38	0	70	25
Baltic	4	0	12	48	0
AG	45	68	7	0	50
WAfr	83	73	0	1	75

Hungarian Algorithm

Βήμα 3. Κάλυψε όλα τα μηδενικά με τον ελάχιστο αριθμό γραμμών (σε γραμμές και στήλες)

Αν ο αριθμός των γραμμών είναι μικρότερος του n, Εκτέλεσε Βήμα 4

	A	**B**	**C**	**D**	**E**
Med	0	4	25	0	30
Cont	89	38	0	70	25
Baltic	4	0	12	48	0
AG	45	68	7	0	50
WAfr	83	73	0	1	75

Παρατηρούμε πως οι γραμμές είναι $4 < 5$, επομένως εκτελούμε Βήμα 4

Hungarian Algorithm

Βήμα 4. Εντόπισε την ελάχιστη τιμή (θ) η οποία δεν είναι καλυμμένη από οποιαδήποτε γραμμή.

Αφαίρεσε το θ από όλα τα ακάλυπτα στοιχεία

Αν υπάρχει στοιχείο το οποίο είναι καλυμμένο από δύο γραμμές πρόσθεσε σε αυτό θ και εκτέλεσε το Βήμα 3

$$\theta = 1$$

	A	**B**	**C**	**D**	**E**
Med	0	4	25	0	30
Cont	89	38	0	70	25
Baltic	4	0	12	48	0
AG	45	68	7	0	50
WAfr	83	73	0	***1***	75

	A	**B**	**C**	**D**	**E**
Med	0	4	**26**	0	30
Cont	**88**	**37**	0	**69**	**24**
Baltic	4	0	**13**	48	0
AG	45	68	**8**	0	50
WAfr	**82**	**72**	0	**0**	**74**

Hungarian Algorithm

Βήμα 3. Κάλυψε όλα τα μηδενικά με τον ελάχιστο αριθμό γραμμών (σε γραμμές και στήλες)

Αν ο αριθμός των γραμμών είναι μικρότερος του n, Εκτέλεσε Βήμα 4

	A	B	C	D	E
Med	0	4	26	0	30
Cont	88	37	0	69	24
Baltic	4	0	13	48	0
AG	45	68	8	0	50
WAfr	82	72	0	0	74

Παρατηρούμε πως οι γραμμές είναι $4 < 5$, επομένως εκτελούμε Βήμα 4

Hungarian Algorithm

Βήμα 4. Εντόπισε την ελάχιστη τιμή (θ) η οποία δεν είναι καλυμμένη από οποιαδήποτε γραμμή.

Αφαίρεσε το θ από όλα τα ακάλυπτα στοιχεία

Αν υπάρχει στοιχείο το οποίο είναι καλυμμένο από δύο γραμμές πρόσθεσε σε αυτό θ και εκτέλεσε το Βήμα 3

$$\theta = 24$$

	A	B	C	D	E
Med	0	4	26	0	30
Cont	88	37	0	69	***24***
Baltic	4	0	13	48	0
AG	45	68	8	0	50
WAfr	82	72	0	0	74

	A	B	C	D	E
Med	0	4	**<u>50</u>**	**<u>24</u>**	30
Cont	**64**	**13**	0	**69**	**0**
Baltic	4	0	**<u>37</u>**	**<u>72</u>**	0
AG	**21**	**44**	8	0	**26**
WAfr	**58**	**48**	0	0	**50**

Hungarian Algorithm

Βήμα 3. Κάλυψε όλα τα μηδενικά με τον ελάχιστο αριθμό γραμμών (σε γραμμές και στήλες)

Αν ο αριθμός των γραμμών είναι μικρότερος του n, Εκτέλεσε Βήμα 4

	A	B	C	D	E
Med	0	4	50	24	30
Cont	64	13	0	69	0
Baltic	4	0	37	72	0
AG	21	44	8	0	26
WAfr	58	48	0	0	50

Παρατηρούμε πως οι γραμμές είναι $5 = n$, επομένως ο αλγόριθμος τερματίζεται

Hungarian Algorithm

Πώς εντοπίζουμε τη βέλτιστη αντιστοίχιση;

Από τον Πίνακα διαλέγουμε τα μηδενικά, έτσι ώστε κάθε γραμμή και κάθε στήλη να έχει ακριβώς ένα επιλεγμένο μηδενικό

	A	B	C	D	E
Med	**0**	4	50	24	30
Cont	64	13	0	69	**0**
Baltic	4	**0**	37	72	0
AG	21	44	8	**0**	26
WAfr	58	48	**0**	0	50

Η τελική λύση του αλγορίθμου είναι η:

$$s = \{(WAfr, C), (AG, D), (Baltic, B), (Med, A), (Cont, E)\}$$

Hungarian Algorithm

Η τελική λύση του αλγορίθμου είναι η:

$$s = \{(WAfr, C), (AG, D), (Baltic, B), (Med, A), (Cont, E)\}$$

Η οποία έχει κόστος

$$z(S) = 4 + 14 + 37 + 38 + 34 = 127$$

	A	B	C	D	E
Med	***38***	53	61	36	66
Cont	100	60	9	79	***34***
Baltic	30	***37***	36	72	24
AG	61	95	21	***14***	64
WAfr	89	90	***4***	5	79

Hungarian Algorithm

Σχόλιο 1:

- Το **Βήμα 3** αναφέρει**:** *Κάλυψε όλα τα μηδενικά με τον ελάχιστο αριθμό γραμμών (σε γραμμές και στήλες)*
- Δεν εκτελείται πάντα με προφανή τρόπο
- Αντιστοιχεί στην επίλυση ενός Set Covering προβλήματος
  - Universe: Όλα τα μηδενικά στοιχεία του Πίνακα
  - Υποψήφια σύνολα: Οι δυνατές γραμμές
  - Κάθε γραμμή περιέχει τα μηδενικά στοιχεί του πίνακα τα οποία σβήνει
- Μια πλεονεκτική μέθοδος επιλέγει επαναληπτικά τις γραμμές που σβήνουν τα περισσότερα ακάλυπτα μηδενικά
- Σε περιπτώσεις όμως «ισοβαθμίας», η τυχαία επιλογή γραμμής ή στήλης μπορεί να μας οδηγήσει στο μη ελάχιστο αριθμό επιλεγμένων γραμμών
- Στα προβλήματα της τάξης, αυτό θα μπορεί να ξεπεραστεί με λίγες δοκιμές και με παρατήρηση

Hungarian Algorithm

<u>Σχόλιο 2:</u>

- Μετά τον τερματισμό του αλγορίθμου, η επιλογή των μηδενικών τιμών (αντιστοιχίσεων) επιτάσσει πως κάθε γραμμή και κάθε στήλη πρέπει να έχουν ακριβώς ένα επιλεγμένο μηδενικό στοιχείο
- Σε μεγάλα προβλήματα, η επιλογή μπορεί να μην είναι προφανής
- Στα μικρά προβλήματα που θα λύσουμε στην τάξη και πάλι μέσω παρατήρησης ο μοναδικός αυτός τρόπος θα μπορεί να εντοπιστεί

- Ως εκ τούτου μία *απλή* υλοποίηση του Hungarian Algorithm δεν είναι πάντα επαρκής για να αντιμετωπίζει κάθε δυνατό πρόβλημα
- Θα δούμε μία βιβλιοθήκη της Python

Hungarian Algorithm

- Πώς θα τροποποιούσαμε την εργασία μας αν οι τιμές στον Πίνακά μας αναπαριστούσαν την κερδοφορία των διάφορων πλοίων
- Πρόβλημα Μεγιστοποίησης Κέρδους
- Πως θα έπρεπε να εργασθούμε για να λύσουμε το πρόβλημα μεγιστοποίησης;
  – Μπορεί ο Hungarian Algorithm που αντιμετωπίζει προβλήματα ελαχιστοποίησης να εφαρμοσθεί;

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

Hungarian Algorithm

- Hungarian Algorithm επιλύει προβλήματα ελαχιστοποίησης
- Μετατροπή Κερδών σε Ζημιές ($\times\ -1$)

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

	A	B	C	D	E
Med	-38	-53	-61	-36	-66
Cont	-100	-60	-9	-79	-34
Baltic	-30	-37	-36	-72	-24
AG	-61	-95	-21	-14	-64
WAfr	-89	-90	-4	-5	-79

Hungarian Algorithm

- Ο Hungarian Algorithm εφαρμόζεται σε μη-αρνητικούς πίνακες
- Εντοπισμός της μικρότερης εγγραφής στον Πίνακα (Αρνητική Τιμή)
- Πρόσθεση της απόλυτης (εντοπισμένης) τιμής σε όλα τα στοιχεία του Πίνακα
- Εφαρμογή του Hungarian Algorithm για την ελαχιστοποίηση του τροποποιημένου Πίνακα

	A	B	C	D	E
Med	-38	-53	-61	-36	-66
Cont	-100	-60	-9	-79	-34
Baltic	-30	-37	-36	-72	-24
AG	-61	-95	-21	-14	-64
WAfr	-89	-90	-4	-5	-79

	A	B	C	D	E
Med	62	47	39	64	34
Cont	0	40	91	21	66
Baltic	70	63	64	28	76
AG	39	5	79	86	36
WAfr	11	10	96	95	21

Hungarian Algorithm

- Η αντιστοίχιση μέγιστου κέρδους στον αρχικό Πίνακα: $(Cont, A)$ με κέρδος 100
- Η αντιστοίχιση $(Cont, A)$ έχει το ελάχιστο κόστος στον Πίνακα Κόστους
- Αποτελεί το πιο επιθυμητό στοιχείο και στους δύο Πίνακες

Πίνακας Κέρδους

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79

Πίνακας Κόστους

	A	B	C	D	E
Med	62	47	39	64	34
Cont	0	40	91	21	66
Baltic	70	63	64	28	76
AG	39	5	79	86	36
WAfr	11	10	96	95	21

Hungarian Algorithm - Μη-Τετραγωνικοί Πίνακες

- Θεωρήστε την περίπτωση κατά την οποία αντιμετωπίζουμε ένα πρόβλημα αντιστοίχισης το οποίο δεν αντιστοιχεί σε έναν τετραγωνικό Πίνακα
- Π.χ. Περισσότερες περιοχές στον κόσμο

Πίνακας Κόστους

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79
USG	72	64	12	25	41
Spore	45	47	52	36	70

Hungarian Algorithm - Μη-Τετραγωνικοί Πίνακες

- Στο συγκεκριμένο πρόβλημα, κάθε πλοίο πρέπει να αντιστοιχηθεί σε μία περιοχή
- Προφανώς, κάποια περιοχή θα μείνει χωρίς πλοίο

Πίνακας Κόστους

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79
USG	72	64	12	25	41
Spore	45	47	52	36	70

Hungarian Algorithm - Μη-Τετραγωνικοί Πίνακες

- Πώς αντιμετωπίζουμε αυτό το πρόβλημα;
- Δημιουργούμε επιπρόσθετες στήλες (ή γραμμές), θεωρώντας mock πλοία
- Οι τιμές των στηλών θέτονται ίσες με μία κοινή τιμή
  - Κατά προτίμηση 0 για προβλήματα minimization
  - Κατά προτίμηση η μεγαλύτερη τιμή του Πίνακα σε προβλήματα maximization
  - Ο τελικός πίνακας θα επιλυθεί γρηγορότερα (άμεση διαγραφή μηδενικών)

Πίνακας Κόστους

	A	B	C	D	E	V1	V2
Med	38	53	61	36	66	0	0
Cont	100	60	9	79	34	0	0
Baltic	30	37	36	72	24	0	0
AG	61	95	21	14	64	0	0
WAfr	89	90	4	5	79	0	0
USG	72	64	12	25	41	0	0
Spore	45	47	52	36	70	0	0

Hungarian Algorithm - Μη-Τετραγωνικοί Πίνακες

- Με την προσθήκη των δύο στηλών, κάθε αντιστοίχιση των δύο εικονικών πλοίων με περιοχές, επιβαρύνει με ίσο τρόπο το συνολικό κόστος
- Πρακτικά κάθε αντιστοίχιση των νέων στηλών δε διαφοροποιεί τη βέλτιστη αντιστοίχιση των πραγματικών πλοίων σε κάποιες από τις περιοχές
- Η βέλτιστη αντιστοίχιση στον τροποποιημένο τετραγωνικό Πίνακα, θα περιέχει τη βέλτιστη αντιστοίχιση μεταξύ των πέντε πλοίων μας και των επτά περιοχών

Πίνακας Κόστους

	A	B	C	D	E	V1	V2
Med	38	53	61	36	66	0	0
Cont	100	60	9	79	34	0	0
Baltic	30	37	36	72	24	0	0
AG	61	95	21	14	64	0	0
WAfr	89	90	4	5	79	0	0
USG	72	64	12	25	41	0	0
Spore	45	47	52	36	70	0	0

Μη-Τετραγωνικοί Πίνακες

- Από μία λύση της μορφής
  $s = \{(USG, C), (AG, D), (Baltic, B), (Med, A), (Cont, E), \cancel{(Spore, V1), (WAfr, V2)}\}$
  αγνοούμε όλες τις αντιστοιχίσεις, των εικονικών πλοίων με περιοχές
- Επομένως, η λύση η οποία οδηγεί στο ελάχιστο κόστος είναι η
  $$s = \{(USG, C), (AG, D), (Baltic, B), (Med, A), (Cont, E)\}$$
- Δηλαδή, οι περιοχές $WAfr$ και $Cont$ δε θα φιλοξενήσουν κανένα πλοίο της εταιρίας μας

Πίνακας Κόστους

	A	B	C	D	E	V1	V2
Med	38	53	61	36	66	0	0
Cont	100	60	9	79	34	0	0
Baltic	30	37	36	72	24	0	0
AG	61	95	21	14	64	0	0
WAfr	89	90	4	5	79	0	0
USG	72	64	12	25	41	0	0
Spore	45	47	52	36	70	0	0

Hungarian Algorithm - Αντιστοίχιση 1-M

- Μία άλλη παραλλαγή του προβλήματος μπορεί να προκύψει αν η σχέση της αντιστοίχισης δεν είναι 1-1, αλλά 1-m
- Π.χ. μία περιοχή μπορεί να φιλοξενήσει πάνω από 1 πλοία
  - Η Med μπορεί να φιλοξενήσει μέχρι 2 πλοία
  - Η Baltic μπορεί να φιλοξενήσει μέχρι 3 πλοία
- Πως θα μπορούσαμε να τροποποιήσουμε τον Πίνακα Κόστους για την αντιμετώπιση του νέου προβλήματος;

Πίνακας Κόστους

	A	B	C	D	E
Med	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79
USG	72	64	12	25	41
Spore	45	47	52	36	70

Hungarian Algorithm - Αντιστοίχιση 1-Μ

- Προσθήκη επιπρόσθετων γραμμών (κλώνων) για κάθε μία γραμμή για την οποία επιτρέπονται πάνω από μία αντιστοιχίσεις
- Συνολικές γραμμές για κάθε περιοχή πρέπει να είναι ίσες με το άνω όριο πλοίων ανά περιοχή
- Δημιουργία
  - 1 επιπρόσθετη γραμμή για τη Med: Med2
    2 Med γραμμές συνολικά
  - 2 επιπρόσθετων γραμμών για τη Baltic: Baltic2, Baltic3
    3 Baltic γραμμές συνολικά
- Σχόλιο: Αν τα πλοία ήταν στις γραμμές και οι περιοχές στις στήλες, θα δημιουργούσαμε προφανώς νέες στήλες κλώνους

	A	B	C	D	E
Med	38	53	61	36	66
Med2	38	53	61	36	66
Cont	100	60	9	79	34
Baltic	30	37	36	72	24
Baltic2	30	37	36	72	24
Baltic3	30	37	36	72	24
AG	61	95	21	14	64
WAfr	89	90	4	5	79
USG	72	64	12	25	41
Spore	45	47	52	36	70

Hungarian Algorithm

- Σε ένα εργοστάσιο βρίσκονται 5 πρέσες παραγωγής προφίλ αλουμινίου: M1-M5
- Το εργοστάσιο μας πρέπει σήμερα να παρασκευάσει 4 ειδών προφίλ για την κάλυψη παραγγελιών: Π1-Π4
- Βάσει των χαρακτηριστικών προφίλ προέκυψε ο παρακάτω πίνακας κέρδους για την ανάθεση κάθε προφίλ σε μία από τις πέντε μηχανές

	P1	P2	P3	P4
M1	32	10	30	20
M2	45	34	50	15
M3	21	52	55	23
M4	30	20	**62**	40
M5	60	30	44	55

Hungarian Algorithm

- Βάσει όσων είπαμε, μετατροπή Πίνακα Κέρδους σε Πίνακα Κόστους

	P1	P2	P3	P4
M1	32	10	30	20
M2	45	34	50	15
M3	21	52	55	23
M4	30	20	**62**	40
M5	60	30	44	55

	P1	P2	P3	P4
M1	30	52	32	42
M2	17	28	12	47
M3	41	10	7	39
M4	32	42	0	22
M5	2	32	18	7

Hungarian Algorithm

Βήμα 2 - Μετατροπή σε τετραγωνικό Πίνακα

	P1	P2	P3	P4
M1	30	52	32	42
M2	17	28	12	47
M3	41	10	7	39
M4	32	42	0	22
M5	2	32	18	7

	P1	P2	P3	P4	P5
M1	30	52	32	42	0
M2	17	28	12	47	0
M3	41	10	7	39	0
M4	32	42	0	22	0
M5	2	32	18	7	0

Hungarian Algorithm

Συνέχεια με τα βήματα του Hungarian Algorithm

	P1	P2	P3	P4	P5
M1	30	52	32	42	0
M2	17	28	12	47	0
M3	41	10	7	39	0
M4	32	42	0	22	0
M5	2	32	18	7	0

Hungarian Algorithm

Βήμα 1. Βρες την ελάχιστη τιμή κάθε γραμμής και αφαίρεσέ τη από όλα τα στοιχεία της γραμμής (καμία διαφοροποίηση αφού κάθε γραμμή περιέχει ήδη ένα μηδέν)

	P1	P2	P3	P4	P5
M1	30	52	32	42	0
M2	17	28	12	47	0
M3	41	10	7	39	0
M4	32	42	0	22	0
M5	2	32	18	7	0

Hungarian Algorithm

Βήμα 2. Βρες την ελάχιστη τιμή κάθε στήλης και αφαίρεσέ την από όλα τα στοιχεία της στήλης

	P1	P2	P3	P4	P5
M1	30	52	32	42	0
M2	17	28	12	47	0
M3	41	**10**	**7**	39	0
M4	32	42	0	22	0
M5	**2**	32	18	**7**	0

	P1	P2	P3	P4	P5
M1	28	42	32	35	0
M2	15	18	12	40	0
M3	39	0	7	32	0
M4	30	32	0	15	0
M5	0	22	18	0	0

Hungarian Algorithm

Βήμα 3. Κάλυψε όλα τα μηδενικά με τον ελάχιστο αριθμό γραμμών (σε γραμμές και στήλες)

Αν ο αριθμός των γραμμών είναι μικρότερος του n, Εκτέλεσε Βήμα 4

	P1	P2	P3	P4	P5
M1	28	42	32	35	0
M2	15	18	12	40	0
M3	39	0	7	32	0
M4	30	32	0	15	0
M5	0	22	18	0	0

Hungarian Algorithm

Βήμα 4. Εντόπισε την ελάχιστη τιμή (θ) η οποία δεν είναι καλυμμένη από οποιαδήποτε γραμμή.

Αφαίρεσε το θ από όλα τα ακάλυπτα στοιχεία

Αν υπάρχει στοιχείο το οποίο είναι καλυμμένο από δύο γραμμές πρόσθεσε σε αυτό θ και πήγαινε στο Βήμα 3 ($\theta = 12$)

	P1	P2	P3	P4	P5
M1	28	42	32	35	~~0~~
M2	15	18	12	40	~~0~~
M3	~~39~~	~~0~~	~~7~~	~~32~~	~~0~~
M4	~~30~~	~~32~~	~~0~~	~~15~~	~~0~~
M5	~~0~~	~~22~~	~~18~~	~~0~~	~~0~~

	P1	P2	P3	P4	P5
M1	16	30	20	23	0
M2	3	6	0	28	0
M3	39	0	7	32	12
M4	30	32	0	15	12
M5	0	22	18	0	12

Hungarian Algorithm

Βήμα 3. Κάλυψε όλα τα μηδενικά με τον ελάχιστο αριθμό γραμμών (σε γραμμές και στήλες)

Αν ο αριθμός των γραμμών είναι μικρότερος του n, Εκτέλεσε Βήμα 4

$$\gamma\rho\alpha\mu\mu\varepsilon\varsigma = 4$$

	P1	P2	P3	P4	P5
M1	16	30	20	23	0
M2	3	6	0	28	0
M3	39	0	7	32	12
M4	30	32	0	15	12
M5	0	22	18	0	12

Hungarian Algorithm

Βήμα 4. Εντόπισε την ελάχιστη τιμή (θ) η οποία δεν είναι καλυμμένη από οποιαδήποτε γραμμή.

Αφαίρεσε το θ από όλα τα ακάλυπτα στοιχεία

Αν υπάρχει στοιχείο το οποίο είναι καλυμμένο από δύο γραμμές πρόσθεσε σε αυτό θ και πήγαινε στο Βήμα 3 ($\theta = 3$)

	P1	P2	P3	P4	P5
M1	16	30	20	23	0
M2	3	6	0	28	0
M3	39	0	7	32	12
M4	30	32	0	15	12
M5	0	22	18	0	12

→

	P1	P2	P3	P4	P5
M1	13	27	20	20	0
M2	0	3	0	25	0
M3	39	0	10	32	15
M4	27	29	0	12	12
M5	0	22	21	0	15

Hungarian Algorithm

Βήμα 3. Κάλυψε όλα τα μηδενικά με τον ελάχιστο αριθμό γραμμών (σε γραμμές και στήλες)

Αν ο αριθμός των γραμμών είναι μικρότερος του n, Εκτέλεσε Βήμα 4

$$\gamma\rho\alpha\mu\mu\varepsilon\varsigma = 5$$

	P1	P2	P3	P4	P5
M1	~~13~~	~~27~~	~~20~~	~~20~~	~~0~~
M2	~~0~~	~~3~~	~~0~~	~~25~~	~~0~~
M3	~~39~~	~~0~~	~~10~~	~~32~~	~~15~~
M4	~~27~~	~~29~~	~~0~~	~~12~~	~~12~~
M5	~~0~~	~~22~~	~~21~~	~~0~~	~~15~~

Hungarian Algorithm

Τερματισμός & Εντοπισμός Βέλτιστης Λύσης

Από τον Πίνακα διαλέγουμε τα μηδενικά, έτσι ώστε κάθε γραμμή και κάθε στήλη να έχει ακριβώς ένα επιλεγμένο μηδενικό

Επομένως, η βέλτιστη αντιστοίχιση είναι η αντιστοίχιση

$$s = \{(M1, P5), (M2, P1), (M3, P2), (M4, P3), (M5, P4)\}$$

Εφόσον, η $P5$ δεν αποτελεί υπαρκτή εργασία, η αντιστοίχιση είναι η

$$s = \{(M2, P1), (M3, P2), (M4, P3), (M5, P4)\}$$

	P1	P2	P3	P4	P5
M1	13	27	20	20	0
M2	0	3	0	25	0
M3	39	0	10	32	15
M4	27	29	0	12	12
M5	0	22	21	0	15

Hungarian Algorithm

$$s = \{(M2, P1), (M3, P2), (M4, P3), (M5, P4)\}$$

Η τελική αντιστοίχιση οδηγεί σε συνολικό κέρδος:

$$z(s) = 45 + 52 + 62 + 55 = 214$$

	P1	P2	P3	P4
M1	32	10	30	20
M2	45	34	50	15
M3	21	52	55	23
M4	30	20	62	40
M5	60	30	44	55