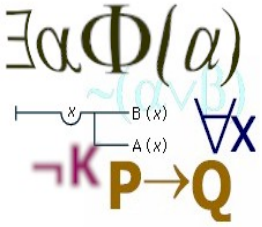
Πανεπιστήμιο Μακεδονίας
Τμ. Εφαρμοσμένης Πληροφορικής

ECLiPSe Constraints III Scheduling



Ηλίας Σακελλαρίου

Δομή

- Χρονοπρογραμματισμός (Μηχανών)
- Περιορισμοί που αφορούν χρονοπρογραμματισμό μηχανών.
  - disjunctive
  - cumulative

Χρονοπρογραμματισμός

Εισαγωγή

Χρονοπρογραμματισμός (1/2)

- Διαδικασία ανάθεσης **πόρων** (resources) σε **εργασίες** (jobs) σε ένα **χρονικό διάστημα**, δεδομένων κάποιων **περιορισμών** και ενός **κριτηρίου** – στόχου (*Μη-αυστηρός ορισμός*)**.**
- *Παραδείγματα Εργασιών*: διεργασίες υλοποίησης IT έργων, διεργασίες κατασκευής προϊόντων, εκτέλεση προγραμμάτων, μαθήματα, απογειώσεις / προσγειώσεις κλπ.
- *Παραδείγματα Πόρων*: ομάδες προγραμματιστών, μηχανές παραγωγής, υπολογιστικοί πόροι,αίθουσες, αεροδιάδρομοι, κλπ

Χρονοπρογραμματισμός (2/2)

- *Παραδείγματα Περιορισμών*: περιορισμοί διάταξης των διεργασιών, καταληκτική ημερομηνία, κλπ
- *Παραδείγματα Κριτηρίων*: ελαχιστοποίηση συνολικού χρόνου εκτέλεσης, ελαχιστοποίηση συνολικού χρόνου αργοπορίας σε σχέση με καταληκτικές ημερομηνίες, ολοκλήρωση διεργασιών εντός των ορίων της καταληκτικής ημερομηνίας, κλπ

Ελαχιστοποίηση Χρόνου

- Ένα έργο αποτελείται από **10 εργασίες (tasks)**, κάθε μια από τις οποίες έχει **μια καθορισμένη διάρκεια**.
- Υπάρχουν **περιορισμοί διάταξης**, πχ. η εργασία 1 πρέπει να εκτελεστεί πριν από τις εργασίες 2 και 3, η 7 μετά την 5, κοκ. Δεν υπάρχουν περιορισμοί ανάμεσα σε όλες τις εργασίες (μερική διάταξη).
- **Διαθέσιμες είναι 4 ομάδες προγραμματιστών,** όμως κάθε ομάδα μπορεί να υλοποιήσει ένα υποσύνολο από τις διαθέσιμες εργασίες.
- *Ποια είναι η ανάθεση εργασιών στις ομάδες ώστε να ελάχιστη η διάρκεια του συνολικού έργου;*

Εργασίες

Περιορισμοί

Πόροι

Περιορισμοί

Κριτήριο

Βιομηχανικές Εφαρμογές Χρονοπρογραμματισμού

- Χρονοπρογραμματισμός επιτελεί σημαντικό ρόλο σε πλήθος βιομηχανικών εφαρμογών.
- Μεγάλη αλληλεπίδραση με άλλες διεργασίες της επιχείρησης
  - Μέρος ενός ERP συστήματος το οποίο τροφοδοτεί με τρέχοντα δεδομένα τον χρονοπρογραμματιστή.
- Παραγωγή
  - Σύνδεση με διαχείριση παραγγελιών, ικανοποίηση παραγγελιών μεγάλης προτεραιότητας, διαχείριση αποθεμάτων πρώτων υλών, κλπ
- Υπηρεσίες
  - Σύνδεση με διαχείριση διαθέσιμων πόρων (ανθρώπινων και μη), συστήματα λήψης απόφασης κλπ

Εργοστάσιο Κατασκευής Χάρτινων Σακουλών

- **Πρώτη ύλη**: ρολά χαρτιού
- **Στάδια Παραγωγής**: Εκτύπωση βιομηχανικού σήματος, επικόλληση της μιας πλευράς συρραφή των άκρων των σακουλών
- **Μηχανές**: Για κάθε εργασία υπάρχει μια ή περισσότερες μηχανές που διαφέρουν στον τύπο / μέγεθος σακουλών που μπορούν να διαχειριστούν, την ταχύτητα παραγωγής, κλπ
- **Παραγγελίες**: ορίζουν την ποσότητα, το είδος και την ημερομηνία παράδοσης.
- **Στόχοι προγράμματος:**
  - Καθυστέρηση στην παραγγελία επιφέρει "ποινές" – στόχος η ελαχιστοποίηση των ποινών.
  - Αλλαγή τύπου κατασκευαζόμενης σακούλας σε μια μηχανή απαιτεί χρόνο που εξαρτάται από τις διαφορές στην κατασκευή των δύο τύπων – Στόχος η ελαχιστοποίηση του χρόνου αυτού.

Διαχείριση Πυλών σε Αεροδρόμιο

- **Πόροι**: Πύλες (gates) με διαφορετικά χαρακτηριστικά συνδεδεμένες με αίθουσες αναμονής
- Κάθε αεροπλάνο που προσγειώνεται θα πρέπει να κατευθύνεται στην κατάλληλη πύλη.
  - *Διεργασίες*: αποβίβαση επιβατών, εξυπηρέτηση αεροσκάφους από προσωπικό εδάφους, επιβίβαση επιβατών, κλπ
- Πτήσεις λαμβάνουν χώρα βάσει προγράμματος, το οποίο όμως επηρεάζεται από πολλούς παράγοντες πχ. καιρικές συνθήκες.
- **Στόχοι**:
  - κατάλληλη πύλη βάσει τύπου αεροσκάφους
  - ελαχιστοποίηση εργασιών προσωπικού εδάφους
  - ελαχιστοποίηση καθυστερήσεων στις πτήσεις

Χαρακτηριστικά Εργασιών

- Απαιτήσεις σε πόρους
  - ποιους **πόρους** και σε τι **ποσότητα**
- Χρόνος εκκίνησης s_i, χρόνος ολοκλήρωσης c_i, διάρκεια d_i
  - δίνονται συνήθως ως κλειστά (χρονικά) διαστήματα
    - s_i *ανήκει στο* $[s_{i\ min}, s_{i\ max}]$
    - c_i *ανήκει στο* $[c_{i\ min}, c_{i\ max}]$
  - η διάρκεια είναι δυνατό να εξαρτάται από τον τύπο του πόρου που ανατέθηκε στην εργασία (χρόνος επεξεργασίας-processing time p_{ij})
- Βάρος w_i της εργασίας, που δηλώνει την σημασία της σε σχέση με άλλες εργασίες του προβλήματος.

Είδη Εργασίων

- **Μη προεκτοπιστικές (non-preemptive)** : δεν μπορούν να διακοπούν
  - $d_i = c_i - s_i$
- **Προεκτοπιστικές (preemptive)**: μπορούν να διακοπούν και να επανεκκινήσουν.
  - $d_i = \Sigma(d_{kl}) \leq c_i - s_i \mid k \text{ in } l\}$
  - Μπορούν να υπάρχουν περιορισμοί στα παραπάνω διαστήματα (πχ. εργασία λαμβάνει χώρα μόνο μέρα).

Πόροι (1/3)

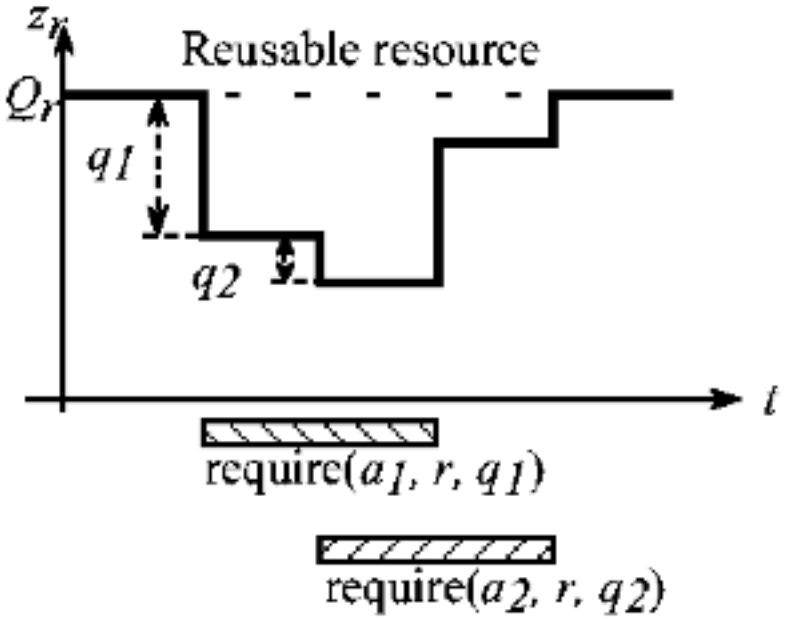
- **Επαναχρησιμοποιήσιμοι πόροι r**
  - δεσμεύονται για ένα χρονικό διάστημα από την εργασία και έπειτα ελευθερώνονται
    - πχ. πύλες αεροδρομίου, εργαλεία, αίθουσες, ομάδες κλπ.
  - **συνολική χωρητικότητα Q_r**, μπορεί να παίρνει συνεχείς ή διακριτές τιμές
  - **τρέχον επίπεδο $z_r(t)$** ανήκει στο $[0,Q_r]$
    - πχ. πέντε πύλες αεροδρομίου $Q_r = 5$, και $5 \geq z_r(t) \geq 0$
  - Αν μια εργασία απαιτεί ποσότητα q του πόρου r, τότε μειώνεται το z_r κατά q όταν η εργασία ξεκινήσει (s_i), και αυξάνεται κατά q μετά την ολοκλήρωση της (c_i).

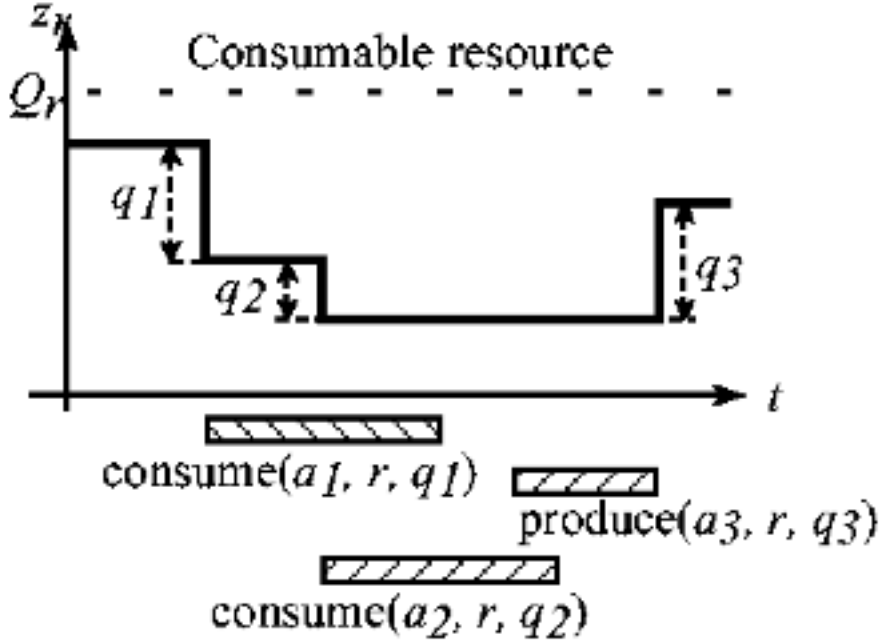
Πόροι (2/3)

- **Πόροι που καταναλώνονται r**
  - καταναλώνονται (ή παράγονται) από μια εργασία
  - μετά το πέρας της εργασίας το τρέχον επίπεδο του πόρου **δεν επανέρχεται** στην αρχική τιμή του
    - πχ. καύσιμα αεροσκαφών, πολλά χαρτιού κλπ.
  - συνολική χωρητικότητα Q_r
  - τρέχον επίπεδο zr(t) ανήκει στο $[0,Q_r]$

Πόροι (3/3)

- Οι απαιτήσεις σε πόρους μιας εργασίας μπορεί να είναι μια **σύζευξη**
  - consume(a,r_j,q_j) & consume(a,r_k,q_k) …
- ή **διάζευξη** αν η εργασία μπορεί να "καταναλώσει" εναλλακτικούς πόρους
  - consume(a,r_j,q_j) v consume(a,r_m,q_m)…





Περιορισμοί

- Χρονικά όρια
  - χρόνος ανακοίνωσης (ri release date):
    - η εργασία δεν μπορεί να ξεκινήσει πριν από αυτό το όριο
    - πχ. άφιξη αεροσκάφους
  - χρόνος παράδοσης (due date δi)
    - εργασία μπορεί να παραδοθεί μετά τον παραπάνω χρόνο με κάποια ποινή.

Περιορισμοί

- Περιορισμοί διάταξης
  - μια εργασία πρέπει να λάβει χώρα πριν από μια άλλη
  - πχ. αποβίβαση πριν επιβίβαση επιβατών επόμενης πτήσης
- Περιορισμοί καταλληλότητας πόρων
  - μια εργασία μπορεί/πρέπει να καταναλώσει συγκεκριμένους πόρους
  - περιορισμοί διαθεσιμότητας πόρων
  - χρόνοι αρχικοποίησης (setup) πόρων ανάμεσα σε δύο εργασίες
  - ...και διάφοροι άλλοι περιορισμοί

Συναρτήσεις Βελτιστοποίησης

- Ελαχιστοποίηση των ακόλουθων μεγεθών
  (c_i δηλώνει το χρόνο ολοκλήρωσης- **completion time**
  δ_i χρόνο παράδοσης -**due date, deadline**)
  - χρόνος ολοκλήρωσης χρονοπρογράμματος (makespan) $\max(c_1,..,c_j)$
  - σταθμισμένο άθροισμα όλων των χρόνων ολοκλήρωσης (total weighted completion time) $\Sigma(w_i c_i)$
  - μέγιστη αργοπορία (lateness): $\max(l_1,..,l_n)$
    - αργοπορία $l_i=c_i - \delta_i$
    - Μπορεί να πάρει και αρνητικές τιμές
  - μέγιστη καθυστέρηση (tardiness): $\max(\tau_1,...,\tau_n)$
    - $\tau_i=\max(0, c_i-\delta_i)$

Συναρτήσεις Βελτιστοποίησης

- σταθμισμένο άθροισμα καθυστέρησης (total weighted tardiness) $\Sigma w_i T_i$
- συνολικός αριθμός καθυστερημένων εργασιών
- συνολικό κόστος (σε σχέση με πόρους)
- σταθμισμένο άθροισμα εργασιών που καθυστέρησαν (weighted sum of late jobs)
- ...

Πώς μπορούν να εκφραστούν τυποκρατικά (formally) όλα τα προηγούμενα;

Χρονοπρογραμματισμός Μηχανών

Machine Scheduling

Χρονοπρογραμματισμός Μηχανών

- Έχει συγκεντρώσει ερευνητικό ενδιαφέρον από την δεκαετία του 50.
- Ανάθεση εργασιών σε μηχανές που αντιπροσωπεύουν πόρους.
- **Μηχανές έχουν χωρητικότητα 1**
  - μπορούν να επεξεργαστούν μια μόνο εργασία την φορά.
- **Εργασίες (jobs) αποτελούνται (συνήθως) από ένα αριθμό διεργασιών (tasks/operations)**
  - πχ. J_i αποτελείται από O_{ij} διεργασίες
  - Δύο διεργασίες της ίδιας εργασίας **δεν μπορούν να εκτελεστούν ταυτόχρονα** (μια εργασία δεν μπορεί ταυτόχρονα να εκτελείται σε δύο μηχανές)
  - Δύο διεργασίες διαφορετικών εργασιών **είναι ανεξάρτητες**, δηλ. μπορούν να εκτελεστούν με οποιαδήποτε σειρά.

Ορισμός Προβλήματος Χρονοπρογραμματισμού Μηχανών (1/2)

- Ένα πρόβλημα Χρονοπρογραμματισμού Μηχανών **X** αναπαριστάται με μία τετράδα <**M,J,C,F**>, όπου:
  - Το **M**={M_1,M_2,..M_m} είναι ένα σύνολο από μηχανές (machines)
  - Το **J**={J_1,J_2,..J_n} είναι ένα σύνολο από εργασίες (jobs)
  - Το **C** είναι ένα σύνολο από περιορισμούς που καθορίζουν για κάθε J_k την καταλληλότητα των μηχανών και τον χρόνο έναρξης καθώς και διάφορα άλλα χαρακτηριστικά
  - Το **F** είναι μία συνάρτηση κόστους (cost function)

Ορισμός Προβλήματος Χρονοπρογραμματισμού Μηχανών (2/2)

- Ένα **χρονοπρόγραμμα** (Schedule) είναι μία ανάθεση του **J** στο **M**, τέτοια ώστε να ικανοποιείται το **C.**
- Ένα χρονοπρόγραμμα ονομάζεται εφικτό (feasible) αν:
  - δεν περιέχει επικαλύψεις εργασιών στην ίδια μηχανή,
  - κάθε διεργασία μιας εργασίας δεν επικαλύπτεται από μια άλλη, και
  - ικανοποιούνται όλοι οι υπόλοιποι περιορισμοί.
- Ένα χρονοπρόγραμμα ονομάζεται βέλτιστο (optimal) αν ελαχιστοποιεί την **F.**

Χρονοπρογραμματισμός μηχανών ενός σταδίου (1/2)

- Κάθε εργασία αποτελείται **από μια διεργασία**, η οποία μπορεί να εκτελεστεί σε οποιαδήποτε μηχανή
  - Single stage machine scheduling
- ***1: Μιας μηχανής:*** υπάρχει μόνο μια μηχανή
- ***Pm*: Πανομοιότυπων Παράλληλων Μηχανών** (Identical Parallel Machines)
  - Οι εργασίες εκτελούνται στον ίδιο χρόνο σε οποιαδήποτε μηχανή ή σε κάποιο υποσύνολο τους.
  - Μπορεί να χρησιμοποιηθεί για μοντελοποίηση πόρων με χωρητικότητα μεγαλύτερη του ένα.

Χρονοπρογραμματισμός μηχανών ενός σταδίου (2/2)

- ***Qm*: Ομοιόμορφα Σχετιζόμενων Παράλληλων Μηχανών** (Uniformly Related Parallel Machines)
  - Οι μηχανές έχουν διαφορετική ταχύτητα επεξεργασία, κοινή για όλες τις διεργασίες.
  - Χρόνος εκτέλεσης της εργασίας p_j/u_i (u_i ταχύτητα μηχανης/p_j χρόνος επεξεργασίας εργασίας)
- ***Rm:* Μη Σχετιζόμενων Παράλληλων Μηχανών** (Unrelated Parallel Machines)
  - Οι μηχανές έχουν διαφορετική ταχύτητα επεξεργασίας η οποία είναι συνάρτηση της εκάστοτε εκτελούμενης εργασίας.

Χρονοπρογραμματισμός μηχανών πολλαπλών σταδίων (1/2)

- Κάθε εργασία αποτελείται από περισσότερες της μίας διεργασίες
- Κάθε διεργασία **έχει μια μόνο μηχανή** στην οποία εκτελείται
  - Multiple-stage scheduling problems
- ***Fm*: Προβλήματα Ροής Καταστημάτων** (flow-shop problems):
  - κάθε εργασία j έχει ακριβώς m διεργασίες $\{O_{ji} | i = 1, \ldots, m\}$
  - Κάθε O_{ji} πρέπει να εκτελεστεί στην μηχανή i
  - οι εργασίες πρέπει να εκτελεστούν με την σειρά $O_{j1}, O_{j2}, \ldots, O_{jm}$ (ίδια σειρά για όλες τις εργασίες)
  - Οι ουρές (queues) των μηχανών είναι FIFO

Χρονοπρογραμματισμός μηχανών πολλαπλών σταδίων (2/2)

- ***Om*: Προβλήματα Ανοικτών Καταστημάτων** (open-shop problems)
  - παρόμοιο με το πρόβλημα ροής αλλά χωρίς διάταξη ανάμεσα στις διεργασίες μιας εργασίας.
- ***Jm*: Προβλήματα Καταστημάτων Εργασιών** (job-shop problems)
  - γενική περίπτωση των προβλημάτων ροής
  - οι διεργασίες μιας εργασίας πρέπει να εκτελεστούν με την συγκεκριμένη σειρά O_{j1}, O_{j2}, …, O_{jm}
  - κάθε διεργασία εκτελείται σε συγκεκριμένη μηχανή $O_{ij}(m_k)$ και γενικά $k \neq j$.

Σημειογραφία Προβλημάτων Χρονοπρογραμματισμού

- $\alpha \mid \beta \mid \gamma$
  - α = κατηγορία προβλήματος :
    - *P* (πανομοιότυπες), *Q* (ομοιόμορφα σχετιζόμενων), *R* (μη σχετιζόμενων) παράλληλων μηχανών
    - *F* (ροής), *O* (ανοικτά), *J* (εργασιών) καταστημάτων
  - β = χαρακτηριστικά εργασιών (καταληκτικές ημερομηνίες, χρόνοι αρχικοποίησης, περιορισμοί διάταξης), κενό αν δεν υπάρχουν περιορισμοί.
  - γ = η συνάρτηση βελτιστοποίησης
- Παραδείγματα:
  - *Pm* | δ_j | $\Sigma_j w_j c_j$ *m* πανομοιότυπες παράλληλες μηχανές, deadlines on jobs, ελαχιστοποίηση σταθμισμένου αθροίσματος ολοκλήρωσης εργασιών
  - *J* | *prec* | *makespan* πρόβλημα καταστήματος εργασιών με τυχαίο αριθμό μηχανών και περιορισμούς διάταξης μεταξύ εργασιών με στόχο την ελαχιστοποίηση συνολικού χρόνου ολοκλήρωσης εργασιών.

Μέθοδοι Επίλυσης

- Κανόνες Διεκπεραίωσης (dispatching rules)
- Μαθηματικές Μέθοδοι
- Μέθοδοι Τοπικής Αναζήτησης
- Ικανοποίηση Περιορισμών

Χρονοπρογραμματισμός σε ECLiPSe

i.e. make your life easier with CLP...

Εύρεση Χρόνου Εκτέλεσης Εργασιών

- Υπάρχουν 5 **εργασίες** και 2 **ομάδες προγραμματιστών**.
- Κάθε εργασία έχει
  - μια προκαθορισμένη **διάρκεια**.
  - απαιτεί για την υλοποίησής της μια **συγκεκριμένη ομάδα.**
- Υπάρχουν **περιορισμοί διάταξης** (μερικής) μεταξύ των εργασιών.
- Κάθε ομάδα μπορεί να εκτελέσει **μια εργασία** σε **κάθε** χρονική στιγμή.
- Ποιοι είναι οι **χρόνοι έναρξης των εργασιών** ώστε να **ελαχιστοποιείται η διάρκεια** του συνολικού έργου;

Λεπτομέρειες Προβλήματος

Εργασία	Διάρκεια	Διάταξη	Ομάδα
1	5	πριν από 2	A
2	4		B
3	7	πριν από 5	B
4	1		A
5	9		A

Αναπαράσταση Χρόνου

- Χρονικές Στιγμές (ακέραιοι αριθμοί)
  - Αρχή του χρόνου: 0
  - Εργασία E1, χρόνος έναρξης St1, διάρκεια 5

Αναπαράσταση Προβλήματος

- **Κάθε εργασία έχει μια προκαθορισμένη διάρκεια d_i.**
- Άρα για κάθε εργασία μπορώ να ορίσω **δύο μεταβλητές S_i και E_i** που αντιστοιχούν στην αρχή και το τέλος της εργασίας, και έχουν τον περιορισμό:
  - **$St_i + d_i = End_i$**
  - Σε Eclipse (παράδειγμα): St1 + 5 #= End1.

Αναπαράσταση Προβλήματος

- **Υπάρχουν περιορισμοί διάταξης (μερικής) μεταξύ των εργασιών.**
  - Πχ. η εργασία 1 πρέπει να λάβει χώρα πριν την εργασία 2.
  - Σε ECLiPSe:

    End1 #<= St2,

    End3 #<= St5,
- Απαιτεί για την υλοποίησής της **μια συγκεκριμένη ομάδα**.
  - Άρα χωρίζουμε τις εργασίες σε λίστες ανάλογα με την ομάδα η οποία πρέπει να τις εκτελέσει.

Ζητούμενο

- *Ποιοι είναι οι χρόνοι έναρξης των εργασιών ώστε να* ***ελαχιστοποιείται η διάρκεια του συνολικού έργου;***
- Ποια είναι η τελευταία εργασία, για να πάρω τον χρόνο λήξης της;
  - maxlist(List,Var) (βιβλιοθήκη ic_global)
  - Η Var είναι η μεταβλητή με τη μέγιστη τιμή από την List.
- Έτσι:

```
maxlist([End1,End2,End3,End4,End5],MakeSpan),
```

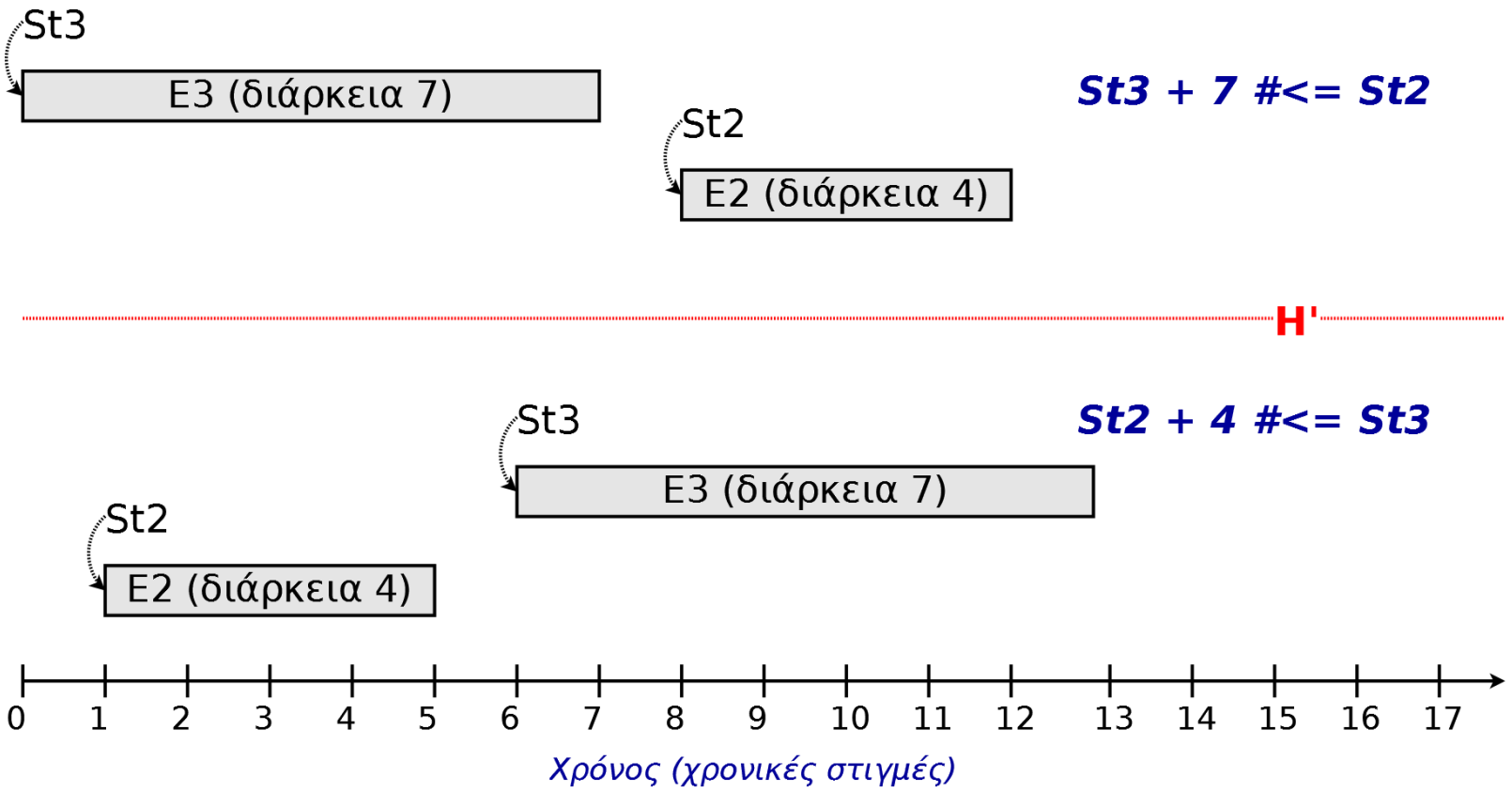
Αναπαράσταση Προβλήματος

- **Κάθε ομάδα μπορεί να εκτελέσει μια εργασία σε κάθε χρονική στιγμή.**
- Άρα **αν** πρέπει να εκτελέσει η πρώτη ομάδα τις εργασίες 1,4,5 με χρόνους 5,1,9 αντίστοιχα τότε ισχύει για κάθε δυνατό ζεύγος εργασιών, πχ 1 και 4:

  St1 + 5 #< St4 **ή** St4 + 1 #< St1

  Ομοίως St2 + 4 #=< St3 **ή** St3 + 7 #=< St2
- Πρόβλημα: Οι **διευζευκτικοί περιορισμοί** (disjunctive constraints) δημιουργούν πολλά σημεία επιλογής (choice points)!
  - Χρειάζονται **ειδικοί αλγόριθμοι διάδοσης** περιορισμών.

Διαζευκτικοί Περιορισμοί



Βιβλιοθήκη ic_edge_finder

- :-use_module(library(ic_edge_finder)).
- Η βιβλιοθήκη περιέχει κατηγορήματα ειδικά για την έκφραση/μοντελοποίηση διαζευκτικών περιορισμών που υπάρχουν σε προβλήματα χρονοπρογραμματισμού.
  - Υπάρχει και η ic_edge_finder3 (με ισχυρότερη διάδοση περιορισμών και μεγαλύτερη πολυπλοκότητα.
- Σημαντικότερα κατηγορήματα
  - disjunctive/2
  - cumulative/4

Ο περιορισμός disjunctive

- disjunctive(+StartTimes, +Durations)
  - Θέτει τον περιορισμό ότι οι διεργασίες με αρχικού χρόνους **StartTimes** και διάρκειες **Durations** δεν μπορούν να αλληλεπικαλύπτονται χρονικά.
  - StartTimes: Λίστα με τις μεταβλητές περιορισμών χρόνου έναρξης των εργασιών.
  - Durations: Λίστα με τις διάρκειες των εργασιών (integers).

Παράδειγμα Περιορισμών Ομάδων

- Εφόσον η **πρώτη ομάδα** έχει να υλοποιήσει τις εργασίες 1,4,5 και 8 μ,ε χρόνους έναρξης **St1, St4, St5, St8** και με διάρκειες **5,1,9,4** αντίστοιχα, και
- Κάθε ομάδα μπορεί να αναλάβει μια εργασία σε κάθε χρονική στιγμή:

  disjunctive([St1,St4,St5],[5,1,9]),...
- ...και ομοίως για κάθε ομάδα...

Τελικός Κώδικας (1/2)

```
schedule_start_times(teamA([St1,St4,St5]),
                                teamB([St2,St3]),MakeSpan):-
    Starts = [St1,St2,St3,St4,St5],
    Ends = [End1,End2,End3,End4,End5],
    Starts #:: 0..inf,   Ends #:: 0..inf,
    End1 #= 5 + St1,  %%% Start End Times and Durations
    End2 #= 4 + St2,     End3 #= 7 + St3,
    End4 #= 1 + St4,     End5 #= 9 + St5,
    End1 #<= St2,   %% Ordering Constraints
    End3 #<= St5,
    ic_global:maxlist(Ends,MakeSpan),
```

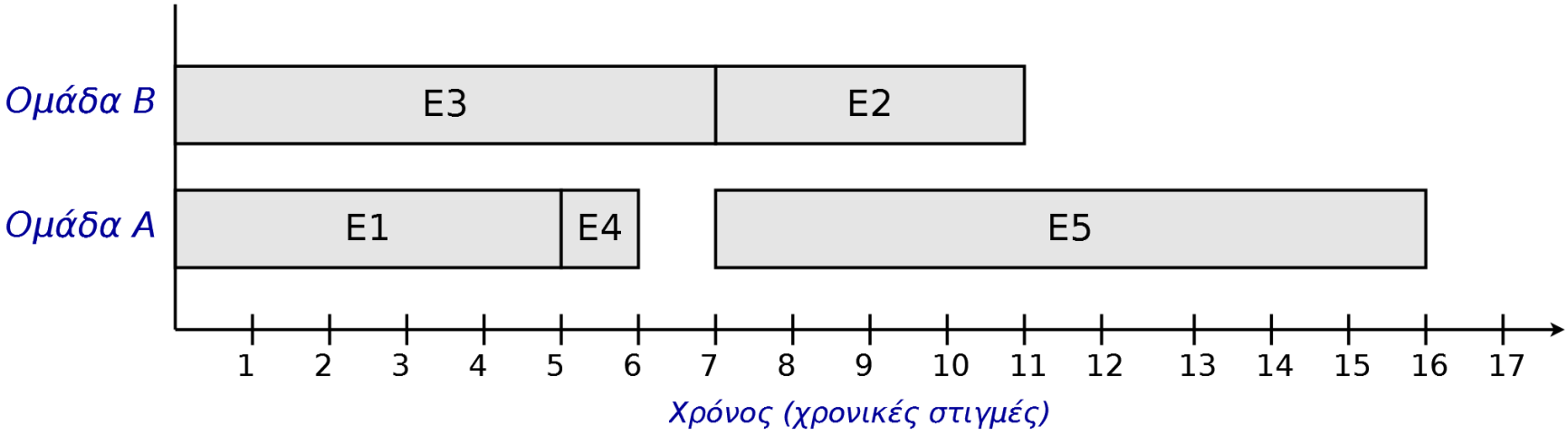
Τελικός Κώδικας (2/2)

```
%%% No overlapping Constraint
% Assignment to teams and
% no overlapping constraints
% team A
disjunctive([St1,St4,St5],[5,1,9]),
% team B
disjunctive([St2,St3],[4,7]),

% Search for a Solution
 bb_min(labeling(Starts),
        MakeSpan,bb_options{strategy:restart}).
```

Εκτέλεση και Πρόγραμμα

```
?- schedule_start_times(T1, T2, M).
T1 = teamA([0, 5, 7])
T2 = teamB([7, 0])
M = 16
Yes (0.00s cpu)
```



Ο περιορισμός cumulative

- Ο περιορισμός Cumulative.
- cumulative(+StartTimes, +Durations, +Resources, + +ResourceLimit)
  - StartTimes: Λίστα με τις μεταβλητές περιορισμών **χρόνου έναρξης** των εργασιών.
  - Durations: Λίστα με τις **διάρκειες** των εργασιών.
  - Resources: Πόσοτητα πόρου **καταναλώνει η κάθε εργασία**.
  - ResourceLimit: Ποια είναι **η συνολική χωρητικότητα** του πόρου.

Επεξήγηση

- Ο περιορισμός είναι αντίστοιχος του περιορισμού **disjunctive** με τις διαφορές:
  - ο **πόρος που καταναλώνεται έχει χωρητικότητα ίση με N**,
  - κάθε **εργασία καταναλώνει K** από τον πόρο
  - και ποτέ δεν μπορεί εργασίες των οποίων το άθροισμα της κατανάλωσης **πόρων είναι μεγαλύτερο του N**, να αλληλεπικαλύπτονται.

Παράδειγμα

- Παράδειγμα: η ομάδα Α έχει 5 μέλη, και η ομάδα Β έχει 4 μέλη. Οι εργασίες απαιτούν:
  - η εργασία 1, ένα μέλος
  - οι εργασίες 4 και 5 από τρία μέλη

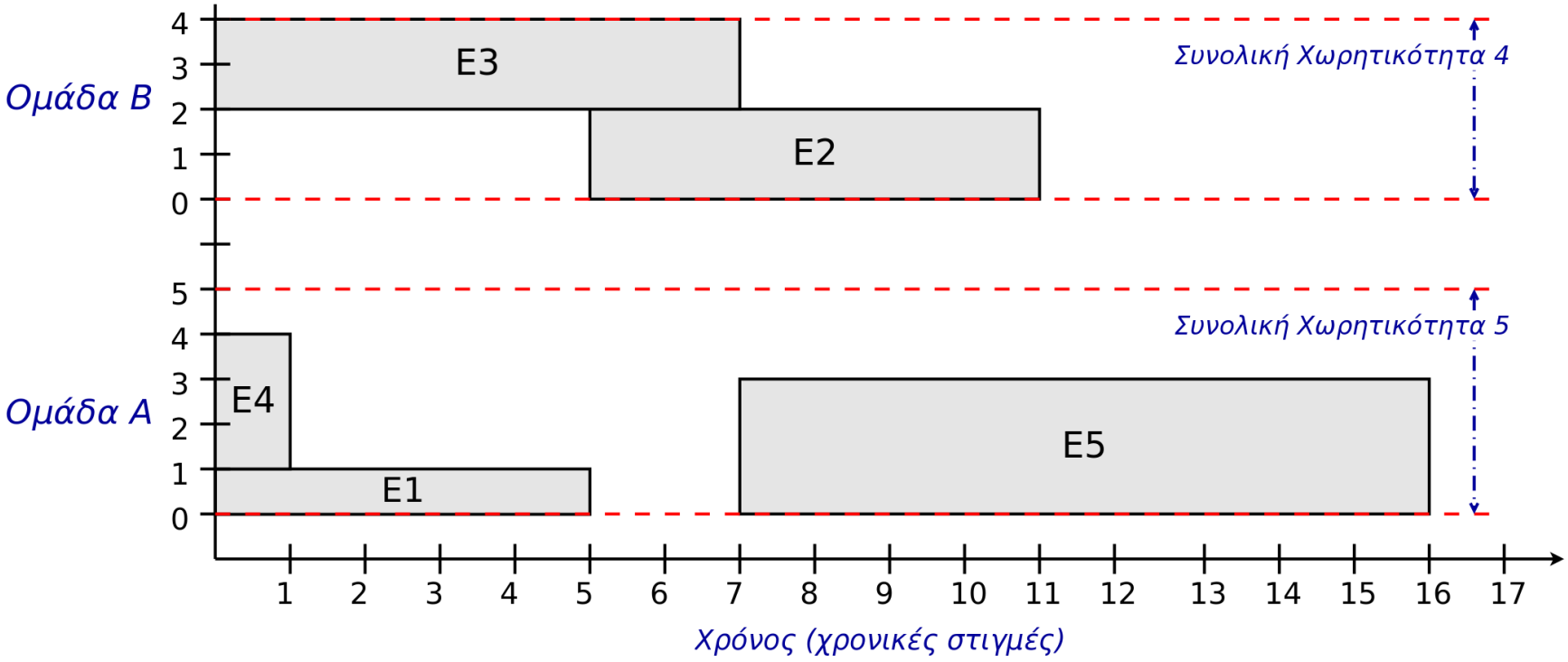
```
cumulative([St1,St4,St5], [5,1,9], [1,3,3], 5)
```

  - οι εργασίες 2 και 3 από δύο μέλη,

```
cumulative([St2,St3], [4,7], [2,2], 4)
```

Εκτέλεση και Πρόγραμμα

```
?- schedule_start_times_r(T1, T2, M).
T1 = teamA([0, 0, 7])
T2 = teamB([5, 0])
M = 16
```



Παράδειγμα 2

- Έξι φορτηγά αυτοκίνητα που έχουν διαφορετικές ταχύτητες και διαφορετικό βάρος πρέπει να περάσουν μια γέφυρα, η οποία μπορεί να αντέξει **μέχρι 20 τόνους**. Τα χαρακτηριστικά των αυτοκινήτων δίνονται σαν Prolog γεγονότα όπως φαίνεται παρακάτω.
- Πότε πρέπει να ξεκινήσει κάθε αυτοκίνητο για να περάσουν όλα με ασφάλεια την γέφυρα, στον **ελάχιστο δυνατό χρόνο**?

Δεδομένα για κάθε αυτοκίνητο

- **Speed**, ο χρόνος που απαιτείται για να διέλθει το αυτοκίνητο τη γέφυρα.
- **Weight**, το βάρος.

```
%%% car(Name, Weight, Speed)
car(alpha, 10, 4).
car(beta, 13,5).
car(gamma, 8, 3).
car(delta, 5, 4).
car(ephilon, 7, 1).
car(zita, 9, 3).
car(eta, 11, 6).
```

Μοντελοποίηση

- Μια μεταβλητή S που δηλώνει το χρόνο έναρξης της διέλευσης του αυτοκινήτου και μια E, που δηλώνει το πότε ολοκληρώνεται η διέλευση.
  - Άρα S + Speed #= E.
- Δεδομένου ότι τα πάντα είναι με την μορφή γεγονότων?
  - findall/3

Κώδικας

```
solve(Starts,MakeSpan):-
    findall(C,car(C,_,_),Cars),
    findall(W,car(_,W,_),Weights),
    findall(S,car(_,_,S),Speeds),
    length(Cars,N),
    length(Starts,N), %% My vars
   Starts #:: 0..inf,
   state_crossing_times(Starts,Speeds,Ends),
    ...
```

Constraints

```
state_crossing_times([],[],[]).
state_crossing_times([S|Starts],[Sp|Speeds],[E|Ends]):-
        S + Sp #= E,
        state_crossing_times(Starts,Speeds,Ends).
```

Μοντελοποίηση Πόρων

- Πόρος = Γέφυρα
- Χωρητικότητα = Βάρος που αντέχει η γέφυρα (max 20).
- Επαναχρησιμοποιήσιμος.
  - Κάθε φορτηγό χρησιμοποιεί Weight από τον πόρο όσο διαρκεί το ταξείδι του (Speed).
- Μοντελοποίηση με περιορισμό cumulative

  cumulative(Starts,Speeds,Weights,20),

Κώδικας 2/2

```
…,
ic:maxlist(Ends,MakeSpan),
cumulative(Starts,Speeds,Weights,20),
bb_min(labeling(Starts),MakeSpan,
                bb_options{strategy:restart}),
pretty_print(Cars,Starts,Ends).
```

Πλήρης Κώδικας

```
solve(Starts,MakeSpan):-
    findall(C,car(C,_,_),Cars),
    findall(W,car(_,W,_),Weights),
    findall(S,car(_,_,S),Speeds),
    length(Cars,N),
    length(Starts,N), %% My vars
    Starts #:: 0..inf,
    state_crossing_times(Starts,Speeds,Ends),
    ic:maxlist(Ends,MakeSpan),
    cumulative(Starts,Speeds,Weights,20),
    bb_min(labeling(Starts),MakeSpan,
                        bb_options{strategy:restart}),
    pretty_print(Cars,Starts,Ends).
```

Δομή

- Χρονοπρογραμματισμός Μηχανών
- Περιορισμοί που αφορούν χρονοπρογραμματισμό μηχανών.
  - disjunctive
  - cumulative