

Εργασία στο μάθημα "Πληροφοριακά Συστήματα Οργάνωσης Παραγωγής"

Θεματική ενότητα: "Χωροταξικός Σχεδιασμός Συστημάτων Παραγωγής"

Προθεσμία παράδοσης: Πέμπτη 11 Απριλίου 2019

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C που να επιλύει το πρόβλημα της **εξισορρόπησης γραμμών συναρμολόγησης έκδοση E** (Assembly Line Balancing Problem, **ALBP-E**). Λεπτομέρειες για το πρόβλημα θα συζητηθούν στα πλαίσια του μαθήματος.

Το πρόγραμμα θα έχει τα εξής χαρακτηριστικά:

1. Θα δίδει τη δυνατότητα στο χρήστη μέσω κατάλληλου μενού επιλογών να επιλέγει την κατηγορία πειραμάτων που θέλει να εκτελέσει. Θα εμφανίζεται σχετική λίστα προβλημάτων.
2. Αφού επιλεγεί η κατηγορία των πειραμάτων κατόπιν θα διαβάζονται από τα σχετικά αρχεία δεδομένων τα εξής:
 - a. Ο γράφος προήγησης των εργασιών (με τις εξαρτήσεις μεταξύ των εργασιών)
 - b. Η διάρκεια (ο χρόνος επεξεργασίας) κάθε εργασίας.
 - c. Δύο ακέραιες τιμές (mmin, mmax) για κάθε διαφορετικό γράφο προήγησης που αναφέρονται στο ελάχιστο και στο μέγιστο επιτρεπτό πλήθος των σταθμών εργασίας στη γραμμή. Αρχείο εισόδου **stations_range.txt**.
 - d. Για κάθε διαφορετικό γράφο προήγησης (αντιστοιχεί σε μια χωριστή κατηγορία προβλήματος) περιλαμβάνονται ένα πλήθος διαφορετικών πειραμάτων αναφοράς (benchmarks).
 - e. Το αρχείο Readme.doc περιγράφει τον τρόπο που διαβάζονται τα δεδομένα στα αρχεία εισόδου.
3. Ακολούθως θα καθορίζεται ο κανόνας με βάση τον οποίο θα γίνεται η ανάθεση των εργασιών στους σταθμούς. Ο κανόνας θα επιλέγεται από ένα διαθέσιμο κατάλογο με συγκεκριμένους ευρετικούς κανόνες ανάθεσης (βλέπε Πίνακα 1 πιο κάτω).
4. Στην έξοδο του το πρόγραμμα θα τυπώνει το πλήθος m^* των σταθμών που επιτεύχθηκε, τον cycle time c^* , τις εργασίες που ανατίθενται σε κάθε σταθμό με τη σειρά που ανατίθενται, τον χρόνο σταθμού και τον αδρανή χρόνο κάθε σταθμού.
5. Για κάθε πείραμα αναφοράς (test) τα αποτελέσματα να φυλάσσονται σε ένα **ενιαίο κοινό αρχείο** κειμένου που θα έχει όνομα το όνομα του αλγόριθμου που τα επέλυσε και επέκταση "out". Για παράδειγμα, αν πρόκειται για τον αλγόριθμο STT (Shortest Task Time rule) το αρχείο αυτό θα ονομάζεται **"STT.out"**.

6. Τα αποτελέσματα στο κάθε αρχείο εξόδου θα δίδονται με τη μορφή:

Precedence graph	mMin	mMax	Idle*	m*	c*	CPU Time (sec)	Tasks in each station
Mertens	3	5					
Mertens	4	5					
Bowman	3	5					
Bowman	4	5					
Jaeschke	3	7					
Jaeschke	4	7					
Jaeschke	5	7					
Jaeschke	6	7					
.....					
Scholl	50	51					

7. Οι αλγόριθμοι που θα υλοποιηθούν θα τρέξουν επί ενός συνόλου πειραμάτων δοκιμασίας που θα βρίσκονται στον φάκελο *dataSets*.

Πίνακας 1: Ευρετικοί κανόνες ανάθεσης εργασιών στους σταθμούς.

1. Longest task time	Επέλεξε τη διαθέσιμη εργασία με τον μεγαλύτερο χρόνο επεξεργασίας.
2. Shortest task time	Επέλεξε τη διαθέσιμη εργασία με τον μικρότερο χρόνο επεξεργασίας.
3. Most following tasks	Επέλεξε τη διαθέσιμη εργασία με τον μεγαλύτερο πλήθος εργασιών που έπονται αυτής στο γράφο προήγησης.
4. Least number of following tasks	Επέλεξε τη διαθέσιμη εργασία με τον μικρότερο πλήθος εργασιών που έπονται αυτής στο γράφο προήγησης.
5. Most number of immediate following tasks	Επέλεξε τη διαθέσιμη εργασία με το μεγαλύτερο πλήθος εργασιών που έπονται άμεσα αυτής.
6. Least number of immediate following tasks	Επέλεξε τη διαθέσιμη εργασία με το μικρότερο πλήθος εργασιών που έπονται άμεσα αυτής.
7. Ranked positional weight	Επέλεξε τη διαθέσιμη εργασία με τον μεγαλύτερο συντελεστή βαρύτητας θέσης.
8.Αλγόριθμος VNS	Αλγόριθμος έρευνας μεταβλητής γειτονιάς (Variable Neighbourhood Search , VNS).

Ο αλγόριθμος υλοποίησης της VNS δίδεται στην σελίδα 3.

Γενικός Αλγόριθμος για την εφαρμογή των 7 πρώτων κανόνων στο πρόβλημα ALBP-E:
Δοθέντων των m_{min} και m_{max} υπολόγισε τα m^* και c^* .

Αλγόριθμος 1

1. Θέσε $I^* = \text{Maxint}$; /* $I^* = \text{total idle time of the best-so-far solution}$ */
2. Θέσε $m = m_{min}$
3. Θέσε $c = \text{LBc}$. /* **Lower bound of cycle time** */
- 5 **Επανάλαβε τα 5.1 & 5.2 μέχρι να βρεθεί έγκυρη λύση εξισορρόπησης***:
 - 5.1. Προσπάθησε να εντοπίσεις μια έγκυρη λύση εξισορρόπησης για το ζεύγος τιμών (m, c) . /* **εδώ γίνεται χρήση κάθε φορά του κανόνα που ορίστηκε προκειμένου να επιλεγεί εργασία για ανάθεση** */
 - 5.2. Αν δεν υπάρχει τέτοια λύση τότε αύξησε το c κατά 1.
- 6 Υπολόγισε τον συνολικό αδρανή χρόνο I της λύσης που επιτεύχθηκε.
- 7 if $I < I^*$ Θέσε $c^* = c$, $m^* = m$ και $I^* = I$
- 8 Αύξησε τον αριθμό των σταθμών κατά 1, $m = m + 1$
- 9 If $(m \leq m_{max})$ Επανάλαβε από το **βήμα 3**
- 10 Επέστρεψε την καλύτερη λύση εξισορρόπησης που βρέθηκε μαζί με τις σχετικές τιμές των m^* , c^* και I^* .
- 11 Τέλος αλγορίθμου

- Όσον αφορά την επιλογή της επόμενης εργασίας για ανάθεση σε σταθμό, αν υπάρχει ισοπαλία στην τιμή της προτεραιότητας θα επιλέγεται τυχαία μια από τις υποψήφιες. Προσοχή όχι εφαρμογή του κανόνα FCFS.

Αλγόριθμος 2: The VNS algorithm

1. Ξεκίνα από μια τυχαία λύση x από δεκαδικούς αριθμούς τυχαία επιλεγμένους στο διάστημα $[0, 1]$. Η x είναι ένας μονοδιάστατος πίνακας μεγέθους n με n το πλήθος των εργασιών συναρμολόγησης. Κάθε δεκαδικός αριθμός στο x ορίζει την προτεραιότητα της αντίστοιχης εργασίας (βλέπε Εικόνα 1).
2. **Επανάλαβε** τα επόμενα βήματα **μέχρι** να ικανοποιηθεί ένα κριτήριο τερματισμού.
3. **Επέλεξε** έναν τυχαίο ακέραιο αριθμό r στο διάστημα $[m_{min}, m_{max}]$.
4. Θέσε $m = r$
5. **Εφάρμοσε τον Αλγόριθμο 1 με διάνυσμα προτεραιοτήτων το x . Οι τιμές στο x θα χρειαστούν στο βήμα 2.1 του Αλγορίθμου για την επιλογή της εργασίας με τη μεγαλύτερη προτεραιότητα.**
6. Θέσε $k = 1$.
7. **Επανάλαβε** τα επόμενα 4 βήματα **μέχρι** $k = k_{max}$.
 - 7.1 Δημιούργησε τυχαία μια νέα λύση x' στη γειτονιά της x εφαρμόζοντας κατά σειρά τις τεχνικές: **Exchange(x); Shift(x); Exchange(x); Shift(x); Shift(x);**
 - 7.2. Δημιούργησε μια νέα λύση x'' στη γειτονιά της x' εφαρμόζοντας την τεχνική **2-opt(x')**.
 - 7.3. **Εφάρμοσε τον Αλγόριθμο 1 με διάνυσμα προτεραιοτήτων το x'' .**

* Λύση εξισορρόπησης είναι η τελική (έγκυρη, νόμιμη) ανάθεση όλων των εργασιών σε σταθμούς.

7.4. **Αν** η νέα λύση (τιμές στα m^* , c^*) που αντιστοιχεί στο x'' είναι καλύτερη από τη λύση που αντιστοιχεί στο x **Τότε** θέσε $x=x''$ και $k=1$.

Διαφορετικά θέσε $k=k+1$.

8. Επέστρεψε τη x
9. Τέλος αλγορίθμου

Εικόνα 1: Ο πίνακας x για ένα πρόβλημα με 8 εργασίες. Η εργασία 1 έχει προτεραιότητα 0.32, η εργασία 2 έχει προτεραιότητα 0.83, η 3 προτεραιότητα 0.05 κλπ.

Job	1	2	3	4	5	6	7	8
priority	0.32	0.83	0.05	0.24	0.17	0.45	0.09	0.61

Υποδείξεις για τον αλγόριθμο VNS:

- Η τιμή της μεταβλητής k_{max} ορίζεται από τον χρήστη. Προτείνεται να σχετίζεται με το μέγεθος του προβλήματος. Για παράδειγμα, να θέσετε $k_{max}=n$.
- Ως κριτήριο τερματισμού στο **βήμα 2** ορίστε το εξής: Ο αλγόριθμος να σταματά μετά από n συνεχόμενες επαναλήψεις.

- Όσον αφορά την επιλογή της επόμενης εργασίας για ανάθεση σε σταθμό, αν υπάρχει ισοπαλία στην τιμή της προτεραιότητας θα επιλέγεται τυχαία μια από τις υποψήφιες. Προσοχή όχι εφαρμογή του κανόνα FCFS.