**Εισαγωγή**

Στην εργασία αυτή εξετάζω το πρόβλημα ανάθεσης (assignment problem), το οποίο στοχεύει στην αντιστοίχιση κάθε εργασίας (job) σε έναν εργαζόμενο (worker) με τέτοιον τρόπο, ώστε να ελαχιστοποιείται το συνολικό κόστος. Χρησιμοποίησα τρεις διαφορετικές προσεγγίσεις:

1. **Ερώτημα 1:** Μαθηματική Μοντελοποίηση (με OR-Tools)
2. **Ερώτημα 2:** Σύγκριση με τον Ουγγρικό Αλγόριθμο (NetworkX)
3. **Ερώτημα 3:** Παραλλαγή με πρόσθετο περιορισμό ομαδοποίησης κάθε πεντάδας εργασιών.

Στις επόμενες ενότητες παρουσιάζω τα αποτελέσματα.

**Ερώτημα 1 – Μαθηματική Μοντελοποίηση με OR-Tools**

Για το πρώτο μέρος, κατασκεύασα ένα μοντέλο Ακέραιου Προγραμματισμού χρησιμοποιώντας τη βιβλιοθήκη **OR-Tools** (solvers SCIP/CBC). Στόχος ήταν να αντιστοιχίσω n εργασίες σε n εργαζόμενους, ώστε να μειώσω το άθροισμα κόστους.

**Αποτελέσματα Εκτέλεσης (erotima1.py):**

* assign4.txt: Κόστος = 161.0, Χρόνος = 0.00 sec
* assign100.txt: Κόστος = 305.0, Χρόνος = 0.19 sec
* assign200.txt: Κόστος = 475.0, Χρόνος = 0.97 sec
* assign300.txt: Κόστος = 626.0, Χρόνος = 2.70 sec
* assign400.txt: Κόστος = 804.0, Χρόνος = 6.71 sec
* assign500.txt: Κόστος = 991.0, Χρόνος = 13.00 sec
* assign600.txt: Κόστος = 1176.0, Χρόνος = 20.16 sec
* assign700.txt: Κόστος = 1362.0, Χρόνος = 29.75 sec
* assign800.txt: Κόστος = 1552.0, Χρόνος = 45.77 sec

Βλέπουμε ότι όσο αυξάνεται το μέγεθος n, αυξάνεται και ο χρόνος επίλυσης. Ωστόσο, το OR-Tools κατορθώνει να δώσει βέλτιστη λύση ακόμη και για 800 εργασίες, σε λιγότερο από 1 λεπτό.

**Ερώτημα 2 – Σύγκριση Με Ουγγρικό Αλγόριθμο**

Στη συνέχεια, συνέκρινα τα αποτελέσματα και τον χρόνο εκτέλεσης του μοντέλου OR-Tools με τον **Ουγγρικό αλγόριθμο** (μέσω της συνάρτησης *minimum\_weight\_full\_matching* της βιβλιοθήκης **NetworkX**).

**Αποτελέσματα Σύγκρισης (erotima2.py):**

* **assign100.txt**:
  + OR-Tools: χρόνος 0.21 sec
  + Ουγγρικός αλγόριθμος: χρόνος 0.92 sec
* **assign200.txt**:
  + OR-Tools: 0.99 sec
  + Ουγγρικός αλγόριθμος: 0.03 sec
* **assign300.txt**:
  + OR-Tools: 2.74 sec
  + Ουγγρικός αλγόριθμος: 0.08 sec
* **assign400.txt**:
  + OR-Tools: 7.35 sec
  + Ουγγρικός αλγόριθμος: 0.14 sec
* **assign500.txt**:
  + OR-Tools: 13.51 sec
  + Ουγγρικός αλγόριθμος: 0.27 sec
* **assign600.txt**:
  + OR-Tools: 28.68 sec
  + Ουγγρικός αλγόριθμος: 0.46 sec
* **assign700.txt**:
  + OR-Tools: 40.83 sec
  + Ουγγρικός αλγόριθμος: 0.54 sec
* **assign800.txt**:
  + OR-Tools: 65.25 sec
  + Ουγγρικός αλγόριθμος: 0.83 sec

Παρατηρούμε ότι ο Ουγγρικός αλγόριθμος είναι ιδιαίτερα γρήγορος για μεγαλύτερα μεγέθη (ειδικά από 200 εργασίες και πάνω), ενώ σε 100 εργασίες παρουσίασε λίγο μεγαλύτερο χρόνο. Σε όλες τις περιπτώσεις, τόσο η μαθηματική μοντελοποίηση όσο και ο Ουγγρικός αλγόριθμος επιτυγχάνουν βέλτιστο ή κοντά στο βέλτιστο κόστος.

**Ερώτημα 3 – Παραλλαγή (Ομαδοποίηση Πεντάδων)**

Για το τρίτο ερώτημα, πρόσθεσα έναν νέο περιορισμό: για κάθε πεντάδα διαδοχικών εργασιών (0-4, 5-9, κ.λπ.), τουλάχιστον δύο από αυτές να εκτελούνται από εργαζομένους της ίδιας πεντάδας (π.χ. εργασία 0 στον εργαζόμενο 1, εργασία 4 στον εργαζόμενο 3).

Το μοντέλο διευρύνθηκε με επιπλέον περιορισμούς γραμμικού προγραμματισμού, κάτι που αυξάνει την πολυπλοκότητα.

**Αποτελέσματα (erotima3.py) – Χρήση SCIP Group Constraint:**

* assign4.txt: Κόστος = 161.0, Χρόνος = 0.00 sec
* assign100.txt: Κόστος = 551.0, Χρόνος = 0.16 sec
* assign200.txt: Κόστος = 1063.0, Χρόνος = 0.81 sec
* assign300.txt: Κόστος = 1296.0, Χρόνος = 3.10 sec
* assign400.txt: Κόστος = 1872.0, Χρόνος = 5.59 sec
* assign500.txt: Κόστος = 2092.0, Χρόνος = 10.34 sec
* assign600.txt: Κόστος = 2906.0, Χρόνος = 19.06 sec
* assign700.txt: Κόστος = 3163.0, Χρόνος = 26.86 sec
* assign800.txt: Κόστος = 3601.0, Χρόνος = 41.04 sec

Βλέπουμε ότι το κόστος αυξάνεται σημαντικά σε σχέση με το βασικό πρόβλημα, πράγμα λογικό, αφού προσθέτουμε επιπλέον «περιορισμούς ομαδοποίησης» που στενεύουν τις διαθέσιμες λύσεις. Ωστόσο, ο χρόνος επίλυσης παραμένει αποδεκτός για μεγέθη έως 800 εργασίες, παρότι είναι μεγαλύτερος από το βασικό πρόβλημα χωρίς αυτόν τον περιορισμό.

**Συμπεράσματα**

1. **Βασικό Πρόβλημα (Ερώτημα 1):**
   * Η μαθηματική μοντελοποίηση με OR-Tools εντοπίζει βέλτιστες λύσεις με λογικό χρόνο εκτέλεσης για ώς και 800 εργασίες.
2. **Σύγκριση με Ουγγρικό Αλγόριθμο (Ερώτημα 2):**
   * Ο Ουγγρικός αλγόριθμος αποδίδει ιδιαίτερα γρήγορα σε μεγαλύτερα n, όμως στα χαμηλά μεγέθη υπάρχει κάποια διακύμανση.
   * Και οι δύο μέθοδοι οδηγούν σε άριστες λύσεις, απλώς η μαθηματική μοντελοποίηση είναι πιο ευέλικτη στην προσθήκη πρόσθετων περιορισμών.
3. **Παραλλαγή με Ομαδοποίηση (Ερώτημα 3):**
   * Ο πρόσθετος περιορισμός αυξάνει σημαντικά το κόστος, αλλά και το χρόνο εκτέλεσης του μοντέλου.
   * Παρόλα αυτά, τα αποτελέσματα είναι εφικτά/βέλτιστα σε λογικούς χρόνους ακόμα και για 800 εργασίες.

**Οδηγίες Εκτέλεσης**

1. **Προετοιμασία Δεδομένων:** Βεβαιωθείτε ότι τα αρχεία εισόδου (π.χ. assign100.txt) βρίσκονται στον φάκελο dataset/.
2. **Βασική Εκτέλεση (Ερώτημα 1):**
   * python erotima1.py
   * Τα αποτελέσματα αποθηκεύονται στον φάκελο solutions/erotima1/.
3. **Σύγκριση (Ερώτημα 2):**
   * python erotima2.py
   * Παράγονται γραφήματα (κόστος/χρόνος) στον φάκελο comparison\_img/.
4. **Παραλλαγή με Περιορισμούς (Ερώτημα 3):**
   * python erotima3.py
   * Αποθηκεύει τα αποτελέσματα στον φάκελο solutions/erotima3/.