MIN COST FLOW PROBLEM

Μοντελοποίηση και Επίλυση

με Τεχνικές Γραμμικού Προγραμματισμού

ΓΕΩΡΓΙΟΣ ΠΑΠΟΥΤΣΑΣ

ΑΜ: 1083738

Έτος: 4ο

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών
Πανεπιστήμιο Πατρών
Ακαδημαϊκό Έτος 2023-2024

Περιεχόμενα

1. Εισαγωγή:

1.1 Επισκόπηση του Min Flow Cost Problem:

Το πρόβλημα Ελαχίστου Κόστους Ροής (MCF) είναι ένα θεμελιώδες πρόβλημα στη βελτιστοποίηση δικτύων, το οποίο αφορά στην εξεύρεση της λιγότερο δαπανηρής μεθόδου μεταφοράς μιας συγκεκριμένης ποσότητας ροής μέσω ενός δικτύου από κόμβους προμήθειας (πηγές) προς κόμβους ζήτησης (καταβόθρες). Το δίκτυο αναπαρίσταται ως ένας κατευθυνόμενος Γράφος, όπου κάθε ακμή (ή τόξο) έχει ένα σχετικό κόστος ανά μονάδα ροής και μια χωρητικότητα που περιορίζει τη μέγιστη ροή που μπορεί να μεταφέρει. Ο στόχος του προβλήματος MCF είναι να προσδιορίσει τη βέλτιστη ροή μέσω του δικτύου που ικανοποιεί τους περιορισμούς προμήθειας και ζήτησης με το ελάχιστο δυνατό συνολικό κόστος.

1.2 Σημασία και Εφαρμογές του Προβλήματος:

Το πρόβλημα MCF αποτελεί γενίκευση πολλών γνωστών προβλημάτων βελτιστοποίησης, όπως το πρόβλημα της συντομότερης διαδρομής, το πρόβλημα της μέγιστης ροής και το πρόβλημα μεταφοράς. Βρίσκει εφαρμογές σε ένα ευρύ φάσμα πραγματικών σεναρίων, όπως στη διαχείριση εφοδιαστικής αλυσίδας και logistics, στις τηλεπικοινωνίες, στη δρομολόγηση κυκλοφορίας και στα χρηματοοικονομικά δίκτυα. Για παράδειγμα, στη διαχείριση logistics, το πρόβλημα μπορεί να αφορά στην εξεύρεση του πιο οικονομικού τρόπου διανομής αγαθών από πολλές αποθήκες σε διάφορες τοποθεσίες λιανικής, με στόχο την ελαχιστοποίηση του κόστους μεταφοράς και τη συμμόρφωση με τους περιορισμούς χωρητικότητας των οχημάτων.

2. Ορισμός του Προβλήματος:

Ένα δίκτυο ροής είναι ένας κατευθυνόμενος γράφος $G = (V, E)$ με ένα κόμβο source $s \in V$ και ένα κόμβο sink $t \in V$. Κάθε κόμβος $i \in V$ έχει προσφορά ή ζήτηση $s(i)$. Για $s(i) > 0$, λέμε ότι ο κόμβος έχει προσφορά και για $s(i) < 0$ έχει ζήτηση. Κάθε ακμή $(u, v) \in E$ έχει χωρητικότητα (capacity) $c(u, v) > 0$, ροή (flow) $f(u, v)$ και κόστος (cost) $a(u, v)$, με τους περισσότερους min cost flow αλγορίθμους να υποστηρίζουν ακμές με αρνητικά κόστη. Το κόστος μεταφοράς κάποιας ροής στην ακμή (u, v) είναι $f(u, v) \cdot a(u, v)$. Το πρόβλημα απαιτεί ένα ποσό ροής d να σταλθεί από το source s στο sink t.

Ο ορισμός του προβλήματος είναι να ελαχιστοποιήσει το συνολικό κόστος της ροής σε όλες της ακμές:

$$\sum_{(u,v)\in E} a(u, v) \cdot f(u, v)$$

Υπό τους περιορισμούς:

Περιορισμοί χωρητικότητας: $f(u, v) \leq c(u, v)$

Αντισυμμετρικότητα: $f(u, v) = -f(v, u)$

Διατήρηση ροής: $\sum_{w\in V} f(u, w) = 0 \; για \; κάθε \; u \neq s, t$

Απαιτούμενη ροή: $\sum_{w\in V} f(s, w) = d \; και \; \sum_{w\in V} f(w, t) = d$

Το πρόβλημα δεν έχει λύση εάν η προσφορά δεν είναι ίση με τη ζήτηση:

$$\sum_{i\in V} s(i) \neq 0$$

3. Μοντελοποίηση με Γραμμικό Προγραμματισμό:

3.1 Διατύπωση του Min Flow Cost Problem ως Γραμμικό Πρόγραμμα:

Αντικειμενική συνάρτηση:

$$\min \sum_{e \in E} a(e) \cdot f(e)$$

Περιορισμοί:

- Προσφορά και ζήτηση καλύπτονται για όλους τους κόμβους:

$$\sum_{j:(i,j)\in E} f(i,j) - \sum_{k:(k,i)\in E} f(k,i) = s(i) \; \gamma\iota\alpha \; \kappa\acute{\alpha}\theta\varepsilon \; i \in V$$

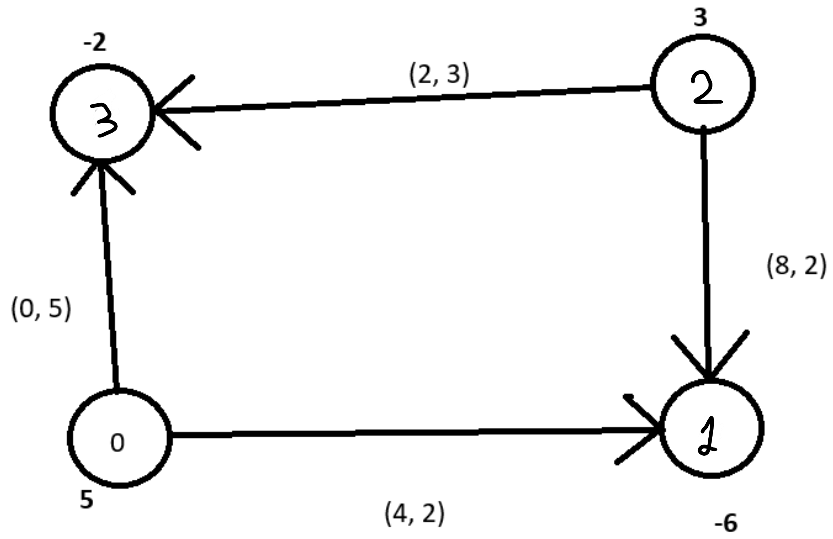
- Χωρητικότητες ακμών ικανοποιούνται:

$$f(i,j) \leq c(i,j) \; \gamma\iota\alpha \; \kappa\acute{\alpha}\theta\varepsilon \; (i,j) \in E$$

- Οι ροές σε κάθε ακμή είναι μεγαλύτερες ή ίσες με 0:

$$f(i,j) \geq 0 \; \gamma\iota\alpha \; \kappa\acute{\alpha}\theta\varepsilon \; (i,j) \in E$$

3.3 Παράδειγμα Διατύπωσης ΓΠ για ένα Απλό Δίκτυο:



Εικόνα 1: Παράδειγμα δικτύου για διατύπωση του ΓΠ

Στην Εικόνα 1 φαίνεται ένα απλό δίκτυο ροής. Ο αριθμός με bold έξω από τον κόμβο είναι η προσφορά ή ζήτηση και σε κάθε ακμή υπάρχει ένα ζευγάρι (κόστος, χωρητικότητα).

Το πρόβλημα σύμφωνα με την ενότητα 3.1 διατυπώνεται ως εξής:

$$\min Z = 0x_1 + 4x_2 + 2x_3 + 8x_4$$

Υπό τους περιορισμούς:

Περιορισμοί διατήρησης ροής:

$$x_1 + x_2 = 5$$

$$x_1 + x_3 = 6$$

$$x_3 + x_4 = 3$$

$$-x_2 + x_4 = -2$$

Περιορισμοί χωρητικοτήτων:

$$0 \leq x_1 \leq 5$$

$$0 \leq x_2 \leq 2$$

$$0 \leq x_3 \leq 3$$

$$0 \leq x_2 \leq 2$$

Όπου:

x_1 είναι η ροή που θα σταλθεί από τον κόμβο $0 \rightarrow 1$

x_2 είναι η ροή που θα σταλθεί από τον κόμβο $0 \rightarrow 3$

x_3 είναι η ροή που θα σταλθεί από τον κόμβο $2 \rightarrow 1$

x_4 είναι η ροή που θα σταλθεί από τον κόμβο $2 \rightarrow 3$

4. Εφαρμογή σε ένα πραγματικό δίκτυο

4.1 Διατύπωση του προβλήματος

Μία πραγματική εφαρμογή του αλγορίθμου είναι η διανομή αγαθών μιας εταιρείας προς κάποιους πελάτες είτε μέσω αποθηκών είτε απευθείας. Θα χρησιμοποιήσουμε δεδομένα από την Crown distributors company στην Kegalle της Sri Lanka για να δείξουμε την πολυπλοκότητα του αλγορίθμου όταν εφαρμόζεται σε ένα αληθινό σενάριο.

Η εταιρεία έχει δύο εργοστάσια(P_1, P_2), τέσσερεις αποθήκες (W_1, W_2, W_3, W_4) και πουλάει τα προϊόντα της σε έξι διαφορετικούς πελάτες ($C_1, C_2, C_3, C_4, C_5, C_6$). Τα κόστη διανομής είναι γνωστά και παρατίθενται στον παρακάτω Πίνακα 1. Η παύλα σημαίνει αδυναμία αποστολής από κάποιον προμηθευτή (εργοστάσιο ή αποθήκη) προς κάποια αποθήκη ή πελάτη και το κενό σημαίνει ότι μια αποθήκη δεν μπορεί να προμηθεύσει τον εαυτό της (προφανώς) ή άλλες αποθήκες.

Πίνακας 1

Προμηθεύει	Προμηθευτής					
	P_1	P_2	W_1	W_2	W_3	W_4
W_1	0.5	-				
W_2	0.5	0.3				
W_3	1.0	0.5				
W_4	0.2	0.2				
C_1	1.0	2.0	-	1.0	-	-
C_2	-	-	1.5	0.5	1.5	-
C_3	1.5	-	0.5	0.5	2.0	0.2
C_4	2.0	-	1.5	1.0	-	1.5
C_5	-	-	-	0.5	0.5	0.5
C_6	1.0	-	1.0	-	1.5	1.5

Κάθε εργοστάσιο έχει ετήσια χωρητικότητα η οποία δεν μπορεί να την υπερβεί:

$$P_1\ 150000$$

$$P_2\ 200000$$

Κάθε αποθήκη έχει μέγιστη ετήσια παροχή/χωρητικότητα (throughput):

$$W_1\ 70000$$

$$W_2\ 50000$$

$$W_3\ 100000$$

$$W_4\ 40000$$

Κάθε πελάτης έχει μηνιαίες παραγγελίες/απαιτήσεις αγαθών:

$$C_1\ 50000$$

$$C_2\ 10000$$

$$C_3\ 40000$$

$$C_4\ 35000$$

$$C_5\ 60000$$

$$C_6\ 20000$$

- Οι μεταβλητές απόφασης είναι οι εξής:

x_{ij} : ποσότητα που στέλνεται από το εργοστάσιο i προς την αποθήκη j

$$i = 1, 2, \qquad j = 1, 2, 3, 4$$

y_{ik} : ποσότητα που στέλνεται από το εργοστάσιο i προς τον πελάτη k

$$i = 1, 2, \qquad k = 1, 2, 3, 4, 5, 6$$

z_{jk} : ποσότητα που στέλνεται από την αποθήκη j προς τον πελάτη k

$$j = 1, 2, 3, 4, \quad k = 1, 2, 3, 4, 5, 6$$

Προκύπτουν 29 μεταβλητές απόφασης. Όλα τα πιθανά μονοπάτια αποστολής είναι 44 όμως έχουμε εξαιρέσει εκείνα στα οποία η αποστολή είναι αδύνατη (παύλες στον Πίνακα 1).

- Η αντικειμενική συνάρτηση είναι η εξής:

Πρέπει να βρούμε το ελάχιστο κόστος του αγαθού της εταιρείας μέσα από το δίκτυο εργοστασίων/αποθηκών/πελατών. Άρα:

$$\sum_{i=1,j=1}^{i=2,j=4} c_{ij}x_{ij} + \sum_{i=1,k=1}^{i=2,k=6} d_{ik}y_{ik} + \sum_{j=1,k=1}^{j=4,k=6} e_{jk}z_{jk}$$

Όπου οι συντελεστές c_{ij}, d_{ik}, e_{jk} προκύπτουν από τον Πίνακα 1.

- Οι περιορισμοί είναι οι εξής και θα διατυπωθούν λίγο διαφορετικά από τα απλά δίκτυα διότι βοηθούν στην κατανόηση του προβλήματος:

1. Περιορισμοί Εργοστασίων:
$$\sum_{j=1}^{2} x_{ij} + \sum_{k=1}^{6} y_{ik} \leq \chi\omega\rho\eta\tau\iota\kappa\acute{o}\tau\eta\tau\alpha[i], \qquad i = 1, 2$$
2. Ποσότητες προς αποθήκες:
$$\sum_{i=1}^{2} x_{ij} \leq \pi\alpha\rho\sigma\chi\acute{\eta}[j], \qquad j = 1, 2, 3, 4$$
3. Ποσότητες από αποθήκες:
$$\sum_{k=1}^{6} z_{jk} = \sum_{i=1}^{2} x_{ij}\,, \qquad j = 1, 2, 3, 4$$
4. Απαιτήσεις πελατών:
$$\sum_{i=1}^{2} y_{ik} + \sum_{j=1}^{4} z_{jk} = \alpha\pi\alpha\acute{\iota}\tau\eta\sigma\eta[k], \qquad k = 1, 2, 3, 4, 5, 6$$

4.2 Διατύπωση του προβλήματος ως ισορροπημένος γράφος

Πέρα από αυτή την μοντελοποίηση σε Γραμμικό Πρόβλημα, μπορούμε επίσης να διατυπώσουμε το πρόβλημα της εταιρείας ως έναν γράφο ως εξής:

- Κόμβοι: Τα εργοστάσια P_1, P_2 (πηγές/sources), οι αποθήκες W_1, W_2, W_3, W_4 (ενδιάμεσοι κόμβοι) και οι πελάτες $C_1, C_2, C_3, C_4, C_5, C_6$ (καταβόθρες/sinks).
- Οι ακμές καθώς και τα κόστη κάθε ακμής δίνονται στον Πίνακα 1.
- Οι χωρητικότητες κάθε ακμών προκύπτουν ως εξής:
    - Οι χωρητικότητες των ακμών από εργοστάσια προς αποθήκες είναι το ελάχιστο της χωρητικότητας του εκάστοτε εργοστασίου και της χωρητικότητας της αποθήκης
    - Οι χωρητικότητες των ακμών από εργοστάσια προς πελάτες είναι το ελάχιστο της χωρητικότητας του εκάστοτε εργοστασίου και της ζήτησής του πελάτη
    - Οι χωρητικότητες των ακμών από αποθήκες προς πελάτες είναι το ελάχιστο της χωρητικότητας της εκάστοτε αποθήκης και της ζήτησής του πελάτη
- Η προσφορά των εργοστασίων δίνεται καθώς και η ζήτηση των πελατών. Οι κόμβοι των αποθηκών έχουν μηδενική προσφορά/ζήτηση διότι αποτελούν ενδιάμεσα στάδια ροής προϊόντος.

Παρατηρούμε όμως ότι η συνολική προσφορά από τα εργαστήρια είναι 350000 ενώ η ζήτηση είναι 215000, άρα ο το δίκτυο μας δεν είναι ισορροπημένο και δεν θα βρεθεί λύση με την μοντελοποίηση της Ενότητας 3.1. Για να το ισορροπήσουμε μπορούμε να προσθέσουμε ένα εικονικό κόμβο με ζήτηση 350000 – 215000 = 135000. Ο κόμβος αυτός θα συνδεθεί με τα δύο εργοστάσια και οι αντίστοιχες ακμές θα έχουν μηδενικά κόστη (για να μην επηρεαστεί το υπόλοιπο δίκτυο) και η επιπλέον προσφορά θα πρέπει να κατανεμηθεί στις χωρητικότητες των κάθε ακμών. Παρατηρήθηκε ότι διαφορετικές κατανομές χωρητικοτήτων βρίσκουν διαφορετικά αποτελέσματα. Πειραματικά βρέθηκε ότι η κατανομή 80000, 55000 βρίσκει ίδια αποτελέσματα με την διατύπωση της Ενότητας 4.1.

Έχοντας τον γράφο μπορούμε να μοντελοποιήσουμε την εταιρεία σε Γραμμικό Πρόβλημα όπως στην Ενότητα 3.1.

5. Μέθοδοι Επίλυσης του Min Flow Cost Problem:

5.1 Αλγόριθμος Ακύρωσης Κύκλων:

Ο αλγόριθμος ακύρωσης κύκλων για την εύρεση ελάχιστου κόστους ροής σε ένα δίκτυο λειτουργεί εντοπίζοντας και ακυρώνοντας επανειλημμένα κύκλους αρνητικού κόστους στο υπολειπόμενο γράφημα, όπου κάθε κύκλος αντιπροσωπεύει μια διαδρομή μέσω της οποίας το συνολικό κόστος της ροής μπορεί να μειωθεί. Ξεκινώντας από μια αρχική εφικτή ροή, ο αλγόριθμος ανιχνεύει κύκλους αρνητικού κόστους (π.χ., με τον αλγόριθμο Bellman-Ford) και αυξάνει τη ροή κατά μήκος αυτών των κύκλων μέχρι να μην υπάρχουν άλλοι κύκλοι αρνητικού κόστους, εξασφαλίζοντας έτσι ότι η τελική ροή έχει το ελάχιστο δυνατό κόστος. Δεν θα αναλυθεί περαιτέρω, όμως έχει επισυναπτηθεί ένα script που λύνει προβλήματα min cost flow με την βιβλιοθήκη or-tools της google που βασίζεται σε αυτόν τον αλγόριθμο.

5.2 Χρήση Ακέραιου Γραμμικού Προγραμματισμού και υλοποίηση σε Python:

Χρησιμοποιώντας μεθόδους Simplex και συγκεκριμένα για προβλήματα με ακέραιες τιμές παίρνουμε την εξής λύση για το απλό δίκτυο:

$$x_1 = 3, x_2 = 2, x_3 = 3, x_4 = 0$$

$$z = 14$$

Δηλαδή η ροή από $0 \rightarrow 1$ θα είναι 3, από $0 \rightarrow 3$ θα είναι 2, από $2 \rightarrow 1$ θα είναι 3 και από $2 \rightarrow 3$ θα είναι 0 και το ελάχιστο κόστος θα είναι 14. Η συνάρτηση που αναπτύχθηκε (αναλυτικότερα στο zip):

```python
def min_cost_flow_ilp(nodes:list,
                      edges:list,
                      cost:dict,
                      capacity:dict,
                      supply:dict):
    """Solves the minimum cost flow problem when given the nodes, edges, costs, capacities and supplies using Gurobi's ILP solver.

    Args:
        nodes (list): List of nodes in the network.
        edges (list): List of edges in the network.
        cost (dict): Dictionary of costs for each edge.
        capacity (dict): Dictionary of capacities for each edge.
        supply (dict): Dictionary of supplies for each node.

    Returns:
        None"""
```

Αυτή η συνάρτηση μπορεί να χρησιμοποιηθεί και για την επίλυση του προβλήματος της εταιρείας Crown Distributors όταν το πρόβλημα μετατραπεί σε μορφή ισορροπημένου γράφου όπως στην ενότητα 4.2. Για το περίπλοκο δίκτυο της εταιρείας παίρνουμε την εξής λύση:

```
Objective value found: 198500.0
f[p1,w1] = -0.0
f[p1,w2] = 0.0
f[p1,w3] = -0.0
f[p1,w4] = 40000.0
f[p2,w2] = 50000.0
f[p2,w3] = 55000.0
f[p2,w4] = 0.0
f[p1,c1] = 50000.0
f[p1,c3] = -0.0
f[p1,c4] = -0.0
f[p1,c6] = 20000.0
f[p2,c1] = -0.0
f[w1,c2] = -0.0
f[w1,c3] = 0.0
f[w1,c4] = -0.0
f[w1,c6] = -0.0
f[w2,c1] = 0.0
f[w2,c2] = 10000.0
f[w2,c3] = -0.0
f[w2,c4] = 35000.0
f[w2,c5] = 5000.0
f[w3,c2] = -0.0
f[w3,c3] = -0.0
f[w3,c5] = 55000.0
f[w3,c6] = -0.0
f[w4,c3] = 40000.0
f[w4,c4] = -0.0
f[w4,c5] = 0.0
f[w4,c6] = -0.0
```

Η συνάρτηση που αναπτύχθηκε για την διατύπωση της ενότητας 4.1 (αναλυτικότερα στο zip):

```python
def min_cost_flow_ilp_factory(nodes:list,
                              edges:list,
                              costs:dict,
                              capacities:dict,
                              throughputs:dict,
                              demands:dict,
                              plants_n:list,
                              warehouses_n:list,
                              customers_n:list,
                              plants_to_warehouses:list,
                              plants_to_customers:list,
                              warehouses_to_customers:list):
    """Solves the minimum cost flow problem for the Crown Distributors Company example using Gurobi's ILP solver.

    Args:
        nodes (list): List of nodes in the network. (plants, warehouses, customers)
        edges (list): List of edges in the network. (plants to warehouses, plants to customers, warehouses to customers)
        costs (dict): Dictionary of costs for each edge.
        capacities (dict): Dictionary of capacities for each plant.
        throughputs (dict): Dictionary of throughputs for each warehouse.
        demands (dict): Dictionary of demands for each customer.
        plants_n (list): List of plants.
        warehouses_n (list): List of warehouses.
        customers_n (list): List of customers.
        plants_to_warehouses (list): List of edges from plants to warehouses.
        plants_to_customers (list): List of edges from plants to customers.
        warehouses_to_customers (list): List of edges from warehouses to customers.

    Returns:
        None
    """
```

6. Συμπεράσματα

Συμπερασματικά, η μοντελοποίηση και επίλυση προβλημάτων ελάχιστου κόστους ροής μέσω γραμμικού προγραμματισμού προσφέρει μια ισχυρή και ευέλικτη προσέγγιση για τη βελτιστοποίηση της ροής σε δίκτυα. Χρησιμοποιώντας τις μεταβλητές απόφασης για να αναπαραστήσουμε τις ροές μεταξύ κόμβων, μπορούμε να ελαχιστοποιήσουμε το συνολικό κόστος ενώ σεβόμαστε τους περιορισμούς χωρητικότητας και ισοζυγίου προσφοράς-ζήτησης. Αυτή η μέθοδος επιτρέπει την εφαρμογή σε ποικίλα προβλήματα πραγματικού κόσμου, όπως οι μεταφορές, οι τηλεπικοινωνίες και η διαχείριση πόρων, προσφέροντας βέλτιστες λύσεις με αποτελεσματικό και μαθηματικά αυστηρό τρόπο.

7. Βιβλιογραφία

1) https://en.wikipedia.org/wiki/Minimum-cost_flow_problem
2) https://www.youtube.com/watch?v=r9L6CQRxgy0
3) https://developers.google.com/optimization/flow/mincostflow#python
4) https://www.topcoder.com/thrive/articles/Minimum%20Cost%20Flow%20Part%20Three:%20Applications
5) https://www.ijser.org/researchpaper/Application-of-Minimum-Cost-Flow-Problem-A-Case-Study-of-Crown-Distributors-in-Kegalle-Sri-Lanka.pdf