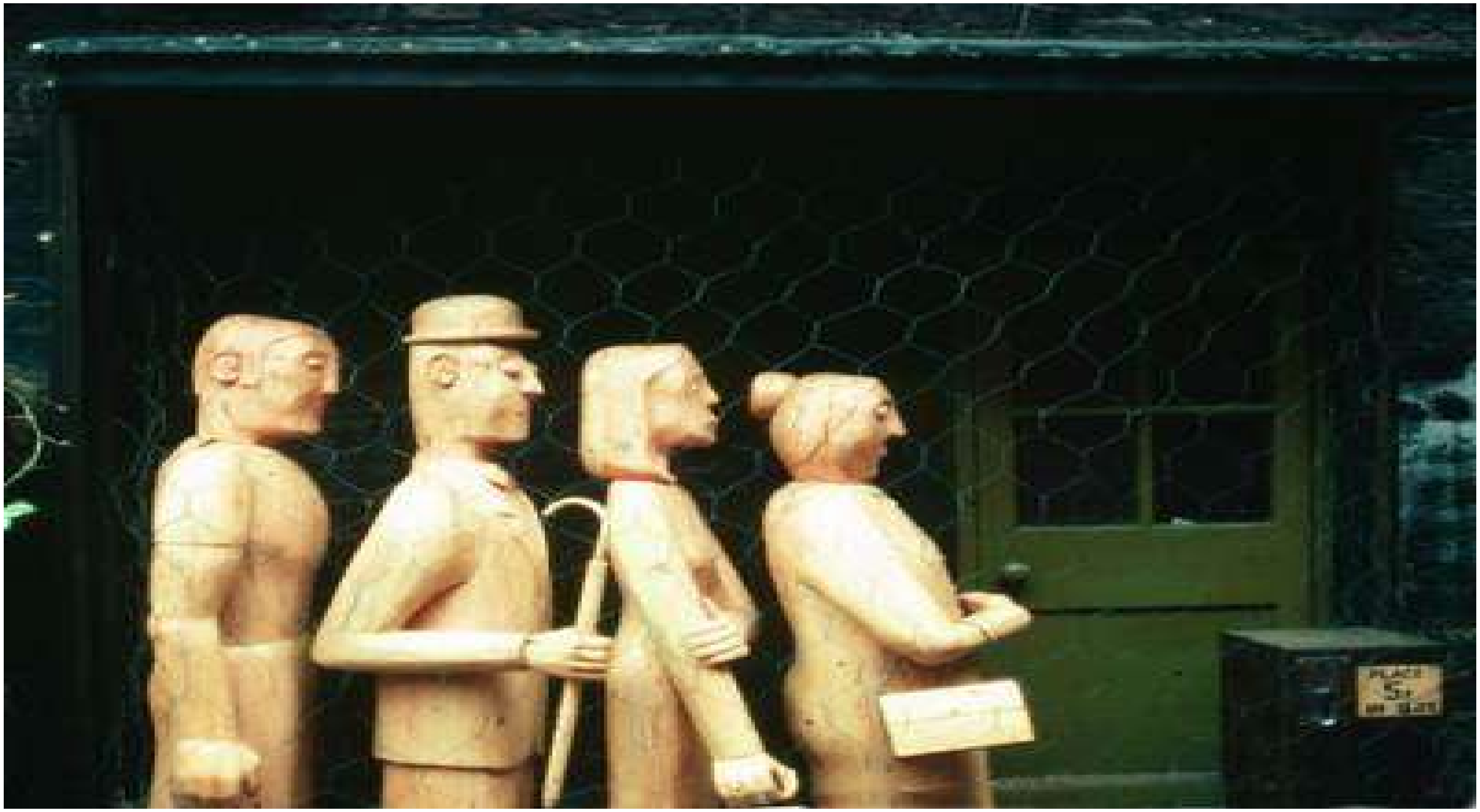
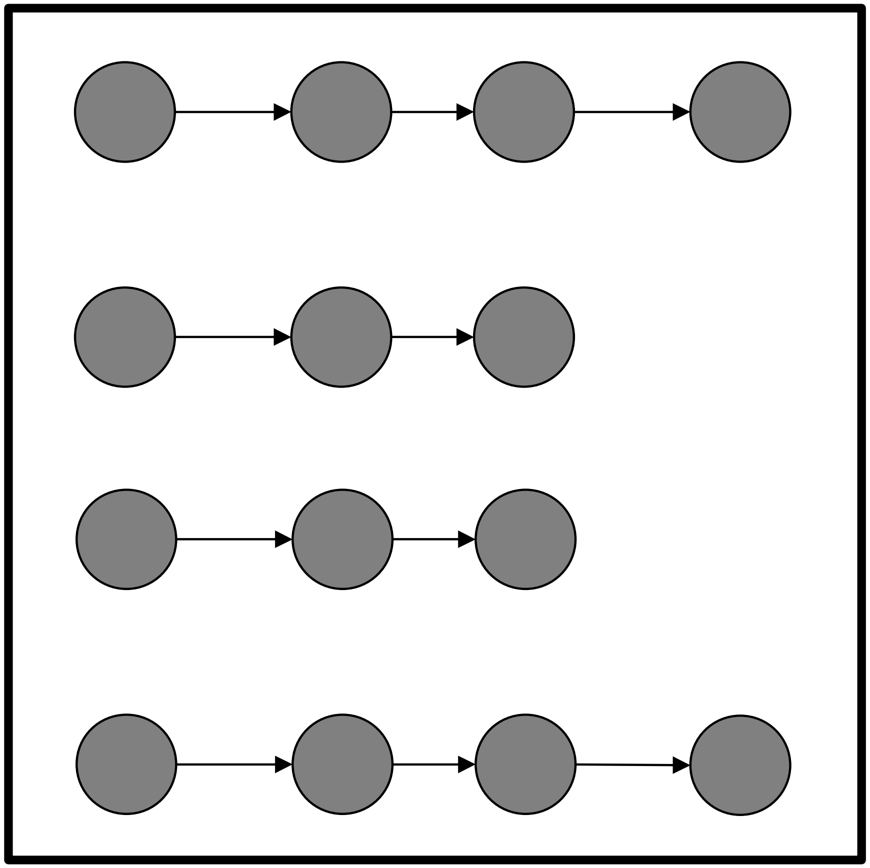
1η ΣΥΝΔΥΑΣΤΙΚΗ ΔΟΜΗ:
ΔΙΑΤΑΞΗ ΣΤΟΙΧΕΙΩΝ



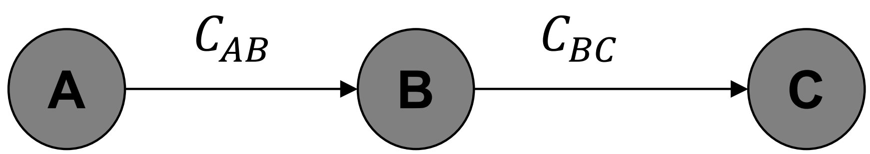
1η ΣΥΝΔΥΑΣΤΙΚΗ ΔΟΜΗ:
ΔΙΑΤΑΞΗ ΣΤΟΙΧΕΙΩΝ → ΠΡΟΒΛΗΜΑΤΑ ΔΙΑΤΑΞΗΣ

- Ένα πρόβλημα συνδυαστικής βελτιστοποίησης καλείται πρόβλημα διάταξης όταν η δομή της λύσης του προβλήματος εκφράζεται ως μια ή περισσότερες διατάξεις στοιχείων.



1η ΣΥΝΔΥΑΣΤΙΚΗ ΔΟΜΗ:
ΔΙΑΤΑΞΗ ΣΤΟΙΧΕΙΩΝ → ΠΡΟΒΛΗΜΑΤΑ ΔΙΑΤΑΞΗΣ

- Η έννοια της «διάταξης στοιχείων» είναι διαφορετική από αυτή της έννοιας του «συνόλου στοιχείων», λόγω του γεγονότος ότι τα στοιχεία μιας διάταξης (θα μπορούσε να ειπωθεί και ως ακολουθία στοιχείων) εμφανίζονται πάντα διατεταγμένα (δηλαδή υπάρχει το 1ο στοιχείο της διάταξης, το 2ο στοιχείο της διάταξης που βρίσκεται αμέσως μετά το 1ο στοιχείο κτλ.).
- Αντιθέτως, τα στοιχεία ενός συνόλου δεν είναι διατεταγμένα. Π.χ. τα σύνολα {1, 2, 3} και {3, 2, 1} αποτελούν δυο ταυτόσημους τρόπους παρουσίασης του ίδιου συνόλου στοιχείων.
- Διαφορετικές διατάξεις στοιχείων παράγουν διαφορετικές λύσεις, που αξιολογούνται από μια αντικειμενική συνάρτηση



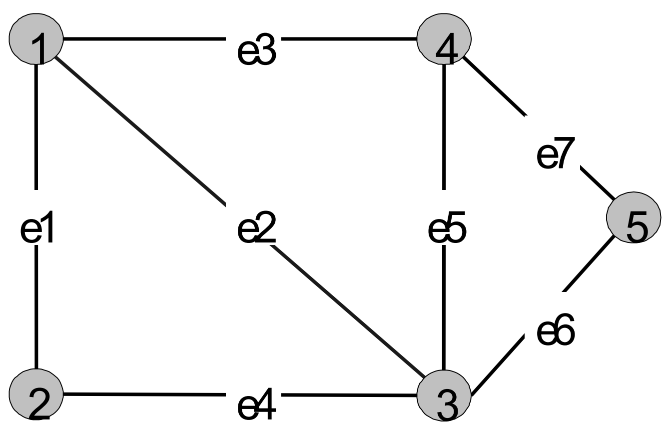
ΔΙΟΙΚΗΤΙΚΗ ΤΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ & ΠΡΟΒΛΗΜΑΤΑ ΔΙΑΤΑΞΗΣ

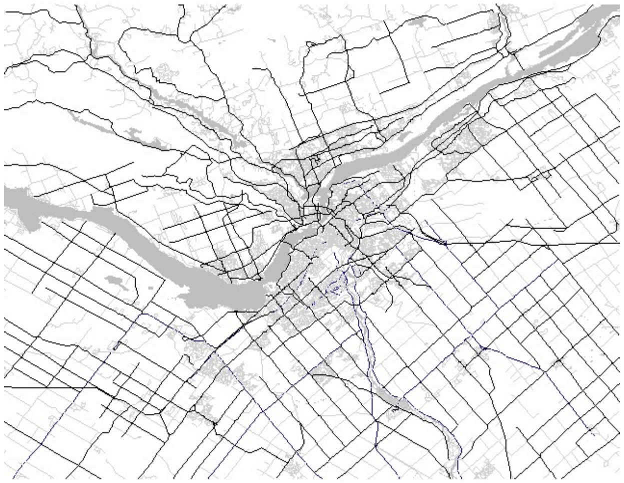
- Στόχος των Προβλημάτων Διάταξης: Η εύρεση του συνόλου των βέλτιστων διατάξεων (μιας ή περισσοτέρων) των στοιχείων της λύσης ενός προβλήματος.

- Παραδείγματα προβλημάτων διάταξης που περιγράφουν πλήθος εφαρμογών του χώρου της διοικητικής των επιχειρήσεων:
  - Το Πρόβλημα του Περιοδεύοντος Πωλητή (Traveling Salesman Problem - TSP).
  - Το Πρόβλημα Δρομολόγησης Στόλου Οχημάτων (Vehicle Routing Problem - VRP).

ΔΙΟΙΚΗΤΙΚΗ ΤΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ & ΠΡΟΒΛΗΜΑΤΑ ΔΙΑΤΑΞΗΣ

- Τα προβλήματα διάταξης αναπαρίστανται από γράφους
- Ένας γράφος (graph) αποτελείται από
  - Κόμβοι: $V = \{1, 2, \dots, m\}$ (Τοποθεσίες, πόλεις, σημεία πώλησης, κτλ.)
  - Τόξα: $A = \{e_1, e_2, \dots, e_n\}$, συνδέσεις μεταξύ δύο κόμβων
  - Αν μία σύνδεση έχει κατεύθυνση, η σύνδεση ονομάζεται τόξο
  - Αν μία σύνδεση δεν έχει κατεύθυνση (συμμετρικά δίκτυα), η σύνδεση ονομάζεται ακμή





ΔΙΟΙΚΗΤΙΚΗ ΤΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ & ΠΡΟΒΛΗΜΑΤΑ ΔΙΑΤΑΞΗΣ

- Ένας Γράφος π.χ. μπορεί να αναπαραστήσει το οδικό δίκτυο εντός του οποίου πραγματοποιείται μια διαδικασία (π.χ. μεταφορά προϊόντων ή ανθρώπων). Συγκεκριμένα,
  - Κόμβοι: Πελάτες ή το κέντρο αναχώρησης / άφιξης των οχημάτων ή τις διασταυρώσεις δρόμων
  - Τόξα: Μετακίνηση μεταξύ δύο κόμβων του δικτύου
- Κάθε σύνδεση ενός γράφου χαρακτηρίζεται από ένα αριθμό που ονομάζεται βάρος (weight) ή κόστος.
  - Το βάρος μπορεί να δηλώνει τον χρόνο που χρειάζεται καλυφθεί ένα κομμάτι δρόμου (βλέπε σύνδεση) εκφρασμένη είτε σε χρόνο είναι σε μήκος

ΔΙΟΙΚΗΣΗ ΜΕΤΑΦΟΡΩΝ – ΔΙΑΝΟΜΩΝ
ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

- Πολυεθνική εταιρία έχει ως κύρια δραστηριότητα τη διανομή ενός μεγάλου φάσματος πετρελαιοειδών (βενζινών, πετρελαίου, μαζούτ, λιπαντικών και υγραερίου) στην ευρύτερη περιοχή της Αθήνας.
- Στα πλαίσια αυτής της συγκεκριμένης δραστηριότητας, το βυτιοφόρο της εταιρείας θα εξυπηρετήσει 4 μεγάλους βιομηχανικούς πελάτες της.
- Η εξυπηρέτηση γίνεται με μία και μόνο μία επίσκεψη του οχήματος σε κάθε πελάτη
- Έχουμε δεδομένα του χρόνου ο οποίος χρειάζεται για να μετακινηθούμε μεταξύ κάθε ζεύγους πελατών
- Στόχος μας: Επιθυμούμε να ελαχιστοποιήσουμε τη διάκριση ως προς το χρόνο εξυπηρέτησης μεταξύ των πελατών μας
  - Δε θέλουμε να δώσουμε το δικαίωμα σε κανέναν πελάτη να παραπονεθεί για άδικη μεταχείριση

ΔΙΟΙΚΗΣΗ ΜΕΤΑΦΟΡΩΝ – ΔΙΑΝΟΜΩΝ
ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

- Αντικειμενική Συνάρτηση: Για να ελαχιστοποιήσουμε τη διάκριση μεταξύ των πελατών μας, επιθυμούμε να μειώσουμε το συνολικό χρόνο που θα περάσει μεταξύ του πελάτη ο οποίος εξυπηρετείται πρώτος και του πελάτη που εξυπηρετείται τελευταίος

ΔΙΟΙΚΗΣΗ ΜΕΤΑΦΟΡΩΝ – ΔΙΑΝΟΜΩΝ
ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

- Δίνονται οι χρόνοι (σε ώρες) μετάβασης μεταξύ των πελατών στον παρακάτω πίνακα.
- Να περιγράψετε και να χρησιμοποιήσετε μια ΠΣΕ για την επίλυση του συγκεκριμένου προβλήματος

Προς

Από

	A	B	C	D
A	0	3	14	17
B	3	0	12	16
C	13	12	0	4
D	14	15	2	0

ΕΠΙΛΥΣΗ του ΠΑΡΑΔΕΙΓΜΑΤΟΣ

- Για να σχεδιάσουμε μια ΠΣΕ θα πρέπει αρχικώς να κατανοήσουμε τη φύση του προβλήματος:
  - Ποια θα είναι η «μορφή της λύσης» του υπό εξέταση προβλήματος;
  - Ποια είναι η αντικειμενική συνάρτηση του προβλήματος;
  - Τι θα ορίσουμε ως «στοιχείο λύσης» για το συγκεκριμένο πρόβλημα;
  - Ποιο θα είναι το «κριτήριο επιλογής» βάσει του οποίου θα επιλεγεί το «στοιχείο της λύσης» που θα προστεθεί στην ημιτελή λύση του προβλήματος;

ΕΠΙΛΥΣΗ του ΠΑΡΑΔΕΙΓΜΑΤΟΣ ΒΑΣΕΙ ΤΟΥ ΚΡΙΤΗΡΙΟΥ ΤΟΥ ΠΛΗΣΙΕΣΤΕΡΟΥ ΓΕΙΤΟΝΑ

- Μορφή της λύσης:
  - Αναζητούμε της λύση που θα ελαχιστοποιήσει τη χρονική περίοδο μεταξύ της πρώτης και τελευταίας επίσκεψης στους πελάτες
  - Επομένως, η λύση είναι μία διάταξη των τεσσάρων κόμβων του προβλήματος
- Αντικειμενική Συνάρτηση:
  - Σύμφωνα με την εκφώνηση του προβλήματος, θέλουμε να ελαχιστοποιήσουμε το χρόνο μεταξύ της πρώτης και τελευταίας επίσκεψης σε πελάτη
  - Ελαχιστοποίηση του συνολικού χρόνου των τριών συνδέσεων οι οποίες ενώνουν τους τέσσερεις πελάτες
  - $S = (A, C, B, D), z(S) = t_{AC} + t_{CB} + t_{BD}$
- Στοιχείο της λύσης:
  - Στοιχείο λύσης που θα προστίθεται σε κάθε επανάληψη της ΠΣΕ είναι το τόξο που ενώνει δύο πελάτες

ΕΠΙΛΥΣΗ του ΠΑΡΑΔΕΙΓΜΑΤΟΣ ΒΑΣΕΙ ΤΟΥ ΚΡΙΤΗΡΙΟΥ ΤΟΥ ΠΛΗΣΙΕΣΤΕΡΟΥ ΓΕΙΤΟΝΑ

- Κριτήριο Επιλογής:
  - Αν η ημιτελής λύση περιέχει ήδη κάποιο κόμβο, η σύνδεση η οποία εκκινεί από τον τελευταίο πελάτη της ημιτελούς λύσης και ελαχιστοποιεί το χρόνο μετάβασης σε έναν πελάτη ο οποίος δεν έχει εξυπηρετηθεί
  - Αν η λύση είναι κενή, η σύνδεση με το ελάχιστο κόστος

ΕΠΙΛΥΣΗ του ΠΑΡΑΔΕΙΓΜΑΤΟΣ ΒΑΣΕΙ ΤΟΥ ΚΡΙΤΗΡΙΟΥ ΤΟΥ ΠΛΗΣΙΕΣΤΕΡΟΥ ΓΕΙΤΟΝΑ

– Επανάληψη 1. Κατατάσσουμε τα εφικτά στοιχεία της λύσης (δηλαδή όλες τις εφικτές συνδέσεις) και επιλέγουμε τη σύνδεση με τη μικρότερη απόσταση. Η σύνδεση με το ελάχιστο κόστος (απόσταση) είναι η (D,C) η οποία και προστίθεται στη δομή της ημιτελούς λύσης S

  Συνεπώς s: $\{(D,C)\}$

– Επανάληψη 2. Από τις συνδέσεις οι οποίες εκκινούν από τον τελευταίο επισκεπτόμενο πελάτη επιλέγουμε τη σύνδεση (C,B)

  Συνεπώς s: $\{(D,C),(C,B)\}$

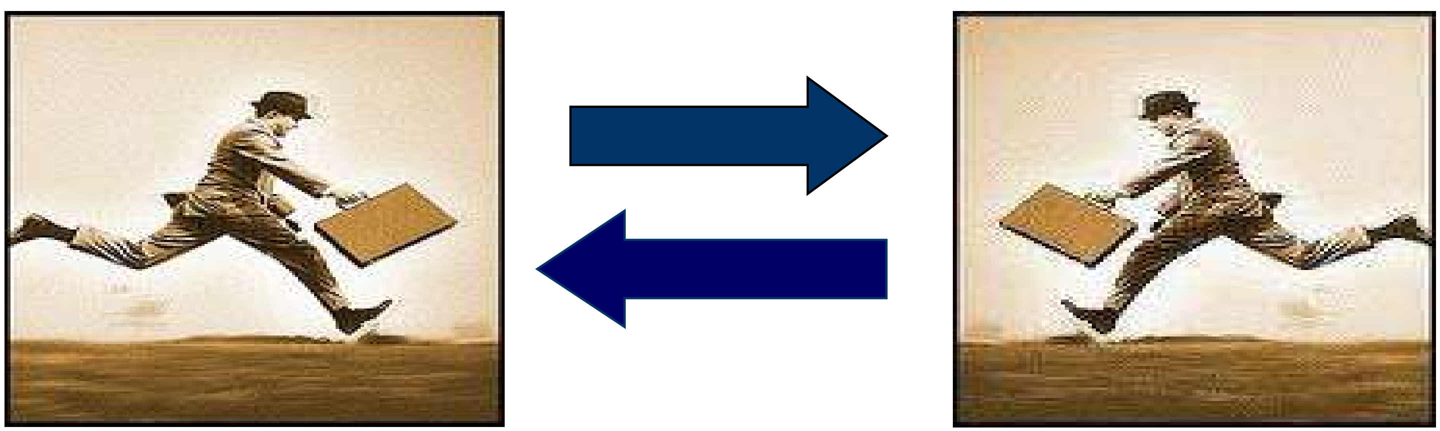
– Επανάληψη 3. Ομοίως, επιλέγουμε τη σύνδεση (B,A) για να προστεθεί στη δομή της ημιτελούς λύσης s

  Επομένως η τελική λύση είναι η s: $\{(D,C)(C,B)(B,A)\}$

  Η αντικειμενική συνάρτηση της τελικής λύσης είναι $z(s) = 17$

ΤΟ ΠΡΟΒΛΗΜΑ του ΠΕΡΙΟΔΕΥΟΝΤΟΣ ΠΩΛΗΤΗ (TSP)

- Το Πρόβλημα του Περιοδεύοντος Πωλητή (Traveling Salesman Problem - TSP) έχει ως στόχο, δεδομένων των αποστάσεων (μήκος ή χρόνος) μεταξύ των σημείων πώλησης που επισκέπτεται ο πωλητής, την εύρεση αυτής της διαδρομής για τον «πωλητή» ώστε:
  - η συνολική απόσταση (συνολικό κόστος) που διανύει ο πωλητής να ελαχιστοποιείται
  - ο πωλητής να επιστρέφει στο σημείο πώλησης από όπου ξεκίνησε (κλειστή διαδρομή)
  - εκτός του αρχικού σημείου πώλησης, ο πωλητής επισκέπτεται μια μόνο φορά το κάθε σημείο πώλησης



ΤΟ ΠΡΟΒΛΗΜΑ του ΠΕΡΙΟΔΕΥΟΝΤΟΣ ΠΩΛΗΤΗ (TSP)

Dantzig-Fulkerson-Johnson formulation

Label the cities with the numbers 1, …, *n* and define:

$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

Take c_{ij} to be the distance from city *i* to city *j*. Then TSP can be written as the following integer linear programming problem:

$$\begin{aligned} \min \sum_{i=1}^{n} \sum_{j\neq i, j=1}^{n} c_{ij} x_{ij} &: \\ 0 \le x_{ij} \le 1 & \qquad i, j = 1, \ldots, n; \\ \sum_{i=1, i\neq j}^{n} x_{ij} = 1 & \qquad j = 1, \ldots, n; \\ \sum_{j=1, j\neq i}^{n} x_{ij} = 1 & \qquad i = 1, \ldots, n; \\ \sum_{i\in Q} \sum_{j\in Q} x_{ij} \le |Q| - 1 & \qquad \forall Q \subsetneq \{1, \ldots, n\}, |Q| \ge 2 \end{aligned}$$

ΤΟ ΠΡΟΒΛΗΜΑ του ΠΕΡΙΟΔΕΥΟΝΤΟΣ ΠΩΛΗΤΗ (TSP)

Miller-Tucker-Zemlin formulation

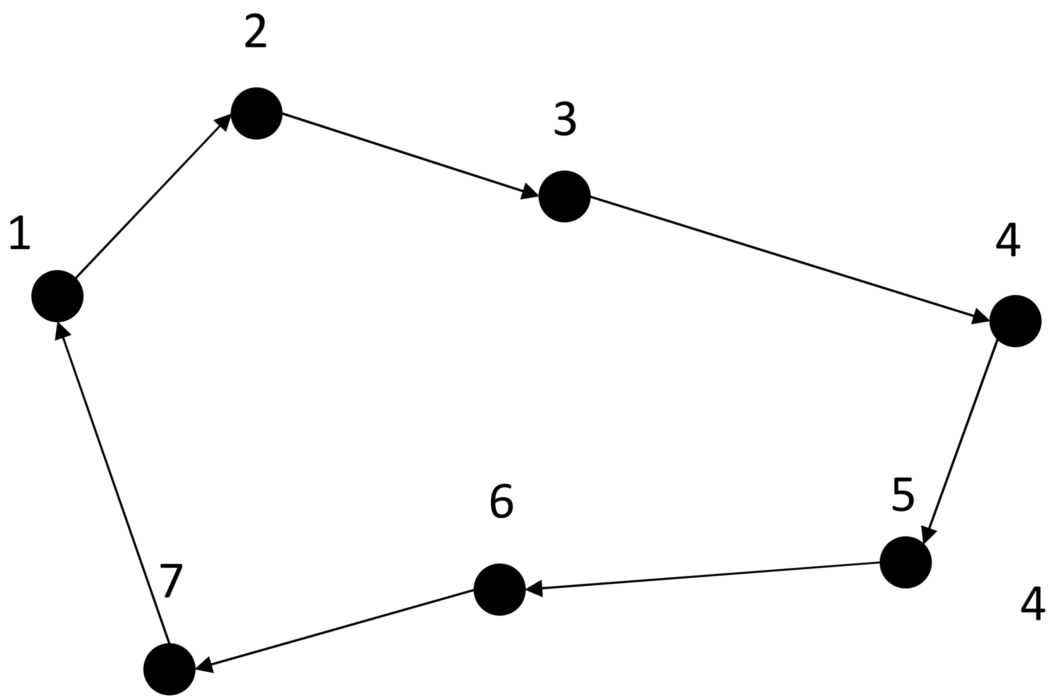
Label the cities with the numbers 1, …, *n* and define:

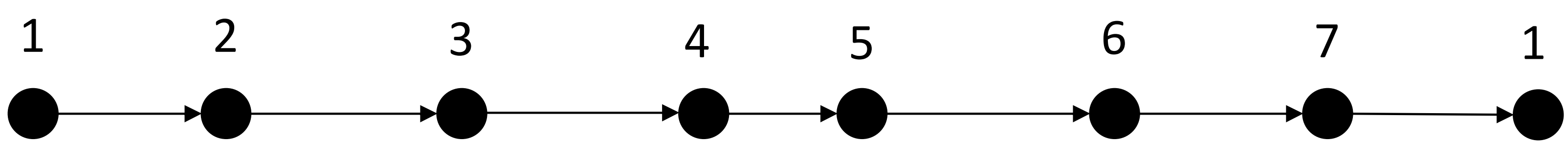
$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

For *i* = 1, …, *n*, let u_i be a dummy variable, and finally take c_{ij} to be the distance from city *i* to city *j*. Then TSP can be written as the following integer linear programming problem:

$$\begin{aligned}
\min \sum_{i=1}^{n} \sum_{j\neq i,j=1}^{n} & c_{ij} x_{ij} \colon && \\
& x_{ij} \in \{0,1\} && i,j = 1,\ldots,n; \\
& u_i \in \mathbf{Z} && i = 2,\ldots,n; \\
& \sum_{i=1,i\neq j}^{n} x_{ij} = 1 && j = 1,\ldots,n; \\
& \sum_{j=1,j\neq i}^{n} x_{ij} = 1 && i = 1,\ldots,n; \\
& u_i - u_j + n x_{ij} \leq n-1 && 2 \leq i \neq j \leq n; \\
& 0 \leq u_i \leq n-1 && 2 \leq i \leq n.
\end{aligned}$$

ΓΡΑΦΙΚΗ ΕΠΙΛΥΣΗ ΤΟΥ TSP





ΠΑΡΑΔΕΙΓΜΑ TSP

- Βρισκόμαστε στην κεντρική αποθήκη ενός εργοστασίου
- Πρέπει να ικανοποιήσουμε τη ζήτηση προϊόντος η οποία προέρχεται από 5 πελάτες
- Προς το σκοπό αυτό θα δρομολογήσουμε το ιδιόκτητο όχημα της εταιρείας μας
- Το όχημα αυτό θα ξεκινήσει από την κεντρική αποθήκη, θα επισκεφθεί μία φορά τον κάθε πελάτη και στη συνέχεια θα επιστρέψει πίσω στην κεντρική αποθήκη

ΠΑΡΑΔΕΙΓΜΑ TSP

- Παρακάτω δίνεται ο συμμετρικός πίνακας κόστους

	0	**1**	**2**	**3**	**4**	**5**
0	0	52	37	46	47	28
1	52	0	84	90	97	41
2	37	84	0	11	15	47
3	46	90	11	0	20	50
4	47	97	15	20	0	61
5	28	41	47	50	61	0

ΠΑΡΑΔΕΙΓΜΑ TSP

- Επιλύστε το πρόβλημα με δύο Πλεονεκτικές στρατηγικές
  - $1^{η}$ ΠΣΕ (Nearest Neighbor):

Ξεκινώντας από την κεντρική αποθήκη (0), επαναληπτικά τοποθέτησε μετά τον τελευταίο πελάτη της μερικής λύσης, τον πελάτη που (α) δεν βρίσκεται ήδη στη λύση και (β) ελαχιστοποιεί το κόστος που απαιτείται για την μετακίνηση σε αυτόν από τον τελευταίο πελάτη της μερικής λύσης

  - $2^{η}$ ΠΣΕ (Minimum Insertions):

Ξεκινώντας από τη μερική λύση (0, 0), διαδοχικά τοποθέτησε σε οποιοδήποτε σημείο της λύσης τον πελάτη που θα οδηγήσει στη μικρότερη αύξηση της αντικειμενικής συνάρτησης

Σημείωση:

Όλοι οι μη τοποθετημένοι πελάτες πρέπει να ελεγχθούν

Όλες οι δυνατές θέσεις εισαγωγής πρέπει να ελεγχθούν

ΠΑΡΑΔΕΙΓΜΑ TSP

- ΠΣΕ 1 – Nearest Neighbor
  - Αρχικοποιημένη Λύση: $s = \{0\}$
  - Λύση μετά την 1η εισαγωγή: $s = \{0, 5\}$
  - Λύση μετά την 2η εισαγωγή: $s = \{0, 5,1\}$
  - Λύση μετά την 3η εισαγωγή: $s = \{0, 5,1,2\}$
  - Λύση μετά την 4η εισαγωγή: $s = \{0, 5,1,2,3\}$
  - Λύση μετά την 5η εισαγωγή: $s = \{0, 5,1,2,3,4\}$
  - Λύση μετά την εισαγωγή της κεντρικής αποθήκης $s = \{0, 5,1,2,3,4,0\}$

  - Αντικειμενική Συνάρτηση Τελικής Λύσης: $z(s) = 231$

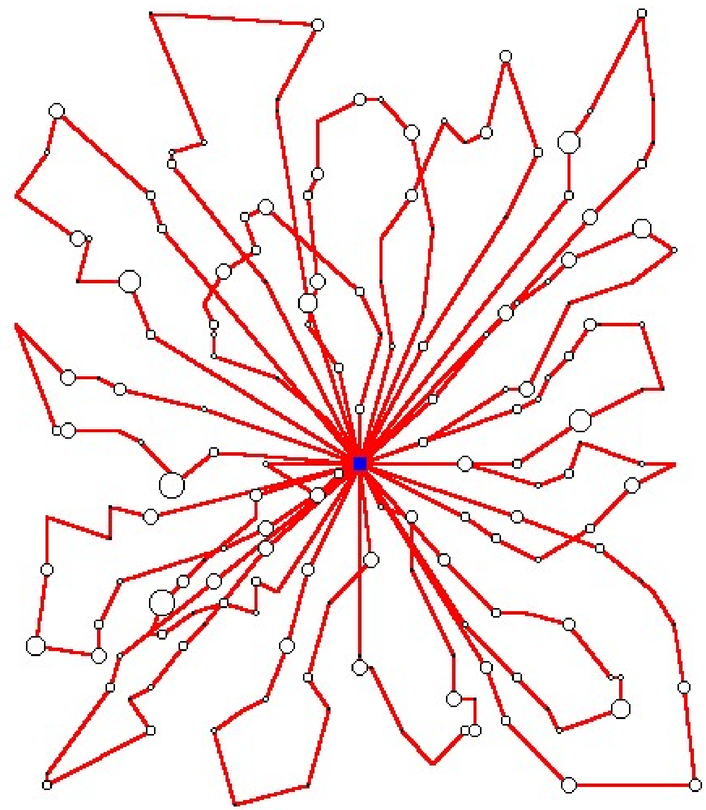
ΠΑΡΑΔΕΙΓΜΑ TSP

- ΠΣΕ 2 – Minimum Cost Insertions
  - Αρχικοποιημένη Λύση: $s = \{0,0\}$
  - Λύση μετά την $1^{η}$ εισαγωγή (Πελάτης 5, Θέση 1): $s = \{0, 5,0\}$
  - Λύση μετά την $2^{η}$ εισαγωγή (Πελάτης 2, Θέση 2): $s = \{0, 2, 5,0\}$
  - Λύση μετά την $3^{η}$ εισαγωγή (Πελάτης 3, Θέση 2): $s = \{0, 2, 3, 5,0\}$
  - Λύση μετά την $4^{η}$ εισαγωγή (Πελάτης 4, Θέση 2): $s = \{0, 2, 4, 3, 5,0\}$
  - Λύση μετά την $5^{η}$ εισαγωγή (Πελάτης 1, Θέση 5): $s = \{0, 2, 4, 3, 5, 1, 0\}$

  - Αντικειμενική Συνάρτηση Τελικής Λύσης: $z(s) = 215$

ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΣΤΟΛΟΥ ΟΧΗΜΑΤΩΝ

VEHICLE ROUTING PROBLEM (VRP)



ΤΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΣΤΟΛΟΥ ΟΧΗΜΑΤΩΝ - VRP

Το Πρόβλημα Δρομολόγησης Στόλου Οχημάτων (Vehicle Routing Problem – VRP) έχει ως στόχο, <u>δεδομένων των αποστάσεων</u> (μήκος ή χρόνος) που εκφράζουν το κόστος διαδρομής μεταξύ των πελατών που εξυπηρετούνται (για διανομή ή για παραλαβή), την εύρεση διαδρομών για ένα ομοιόμορφο στόλο οχημάτων συγκεκριμένης χωρητικότητας, τέτοιων ώστε:

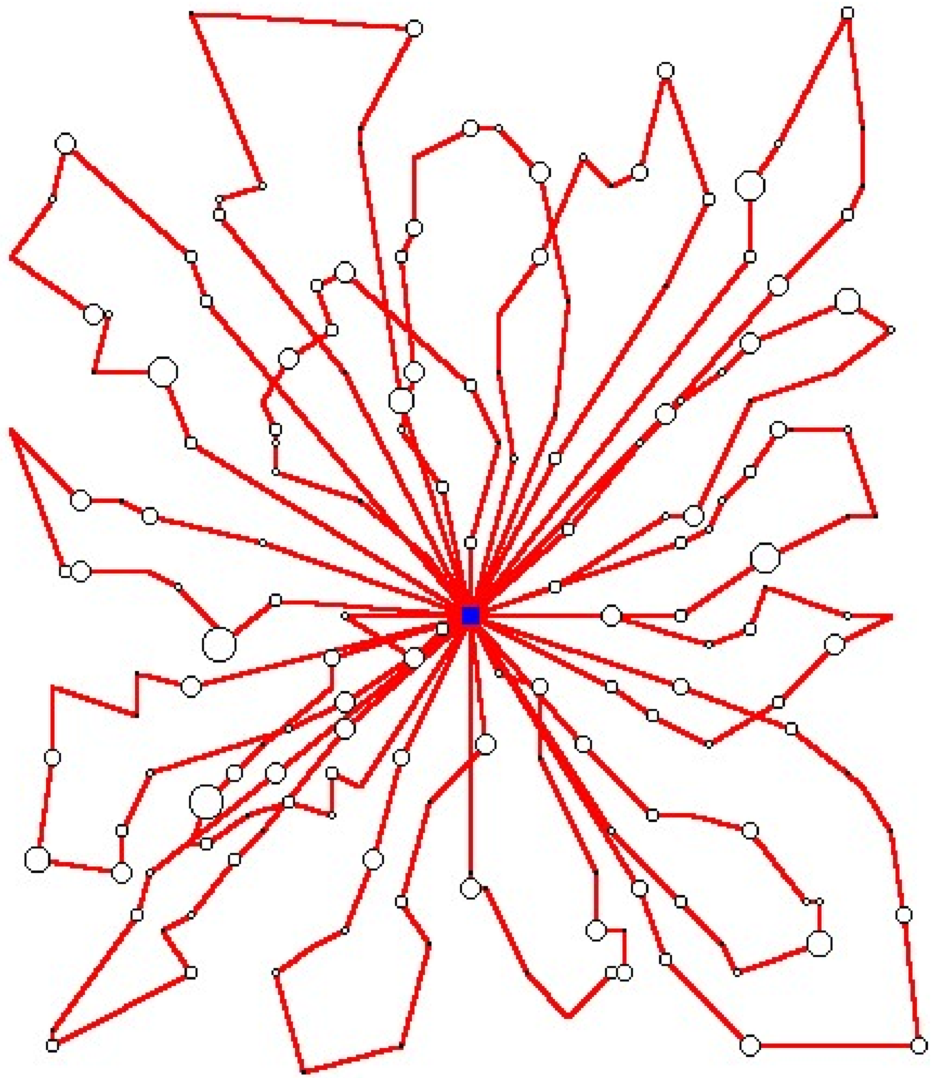
– το συνολικό κόστος (απόσταση εκφρασμένη σε μήκος ή χρόνο και/ή αριθμός οχημάτων που χρησιμοποιούνται) να ελαχιστοποιείται

– η ζήτηση να μην υπερβαίνει τη χωρητικότητα των οχημάτων

– κάθε πελάτης να έχει μία προκαθορισμένη ζήτηση, η οποία πρέπει να ικανοποιείται από μία και μοναδική επίσκεψη ενός οχήματος

– κάθε όχημα πρέπει να αναχωρεί, και αφού εξυπηρετήσει τους πελάτες, να επιστρέφει στον κεντρικό χώρο αποθήκευσης.

Τυπική λύση του VRP



ΜΙΑ ΑΝΑΛΥΤΙΚΗ ΛΥΣΗ ΕΝΟΣ VRP

– 199 πελάτες, 17 οχήματα

0	6	85	61	16	141	44	119	191	91	193	98	92	151	0	
0	26	149	195	179	110	198	72	74	171	73	180	53	0		
0	5	84	173	17	113	86	140	38	14	192	100	37	97	117	0
0	18	114	8	174	45	125	199	83	60	118	166	0			
0	69	101	70	30	160	131	32	181	63	126	90	108	10	189	0
0	54	130	165	55	25	170	67	39	187	139	4	155	0		
0	1	122	128	20	188	66	9	120	81	33	157	50	0		
0	27	167	127	190	31	162	132	176	111	28	0				
0	152	58	2	178	57	15	43	142	42	172	144	87	13	0	
0	51	103	161	71	65	136	35	135	164	34	78	169	129	0	
0	82	46	124	168	47	36	143	49	64	11	17	107	19	123	0
0	105	40	21	197	56	186	23	75	133	22	415	145	115	137	0
0	102	185	79	158	3	77	116	196	76	184	138	0			
0	156	112	0												
0	183	94	95	59	93	99	104	96	147	89	0				
0	146	88	148	159	62	182	48	7	194	106	153	52	0		
0	109	177	134	163	24	29	121	68	80	150	12	154	0		

Συνολική Απόσταση = 1311,48

ΔΙΑΧΕΙΡΙΣΗ ΔΙΑΝΟΜΩΝ –
ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

- Μια επιχείρηση διανομής πετρελαίου θέρμανσης (κόμβος 0) έχει ιδιόκτητο στόλο *δύο οχημάτων* χωρητικότητας 21,000 lt το καθένα, τα οποία χρησιμοποιεί για να καλύψει την ζήτηση όλων των πελατών της. Η επιχείρηση θέλει να βρει τη βέλτιστη ακολουθία εξυπηρέτησης των 9 πελατών (κόμβοι 1-9) της από τον υπάρχων στόλο.
- Περιορισμοί της επιχείρησης:
  - Τα οχήματα ξεκινούν από την επιχείρηση και πρέπει να επιστρέψουν σε αυτή.
  - Πρέπει να εξυπηρετηθούν όλοι οι πελάτες.
  - Η δυναμικότητα των οχημάτων και συνεπώς και η καλυπτόμενη ζήτηση δεν μπορεί να υπερβαίνει τα 21,000 lt το καθένα.
  - Ένας πελάτης μπορεί να εξυπηρετηθεί από ένα και μόνο όχημα, μία και μόνο φορά.

ΔΙΑΧΕΙΡΙΣΗ ΔΙΑΝΟΜΩΝ – ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

Παρακάτω παρατίθενται οι πίνακες των αποστάσεων και της ζήτησης.

	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

ΔΙΑΧΕΙΡΙΣΗ ΔΙΑΝΟΜΩΝ – ΕΠΙΛΥΣΗ ΜΕ ΠΣΕ

Πίνακας ζήτησης πελατών (σε χιλ lt)

Πελάτης	1	2	3	4	5	6	7	8	9
Ποσότητα	2	4	5	1	7	3	5	4	4

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΠΣΕ

- Μορφή Λύσης: Δύο διατάξεις κόμβων κάθε μία από τις οποίες ξεκινά και καταλήγει στον κόμβο 0.
- Στοιχείο Λύσης: Ένα τόξο που ενώνει δύο πελάτες
- Αρχικοποίηση της λύσης: Προσθήκη του κόμβου αφετηρίας (Αποθήκη) και στα δύο οχήματα
- Κριτήριο Επιλογής:

  Σε κάθε επανάληψη της ΠΣΕ πρόσθεσε στη μερική λύση το τόξο

  1. Ξεκινά από τους τελευταίους κόμβους των διαδρομών
  2. Καταλήγει σε πελάτη που δεν παραβιάζει κανέναν από τους περιορισμούς του προβλήματος
  3. Ελαχιστοποιεί την απόσταση που πρέπει να καλυφθεί

(Τροποποίηση του ΝΝ, η οποία λαμβάνει υπόψιν τους τελευταίους κόμβους όλων των διαδρομών)

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΠΣΕ

Τα υποψήφια προς προσθήκη τόξα δεν πρέπει να παραβιάζουν κανέναν από τους περιορισμούς του προβλήματος:

a) Δεν πρέπει να οδηγούν σε πελάτες οι οποίοι βρίσκονται ήδη στην μερική λύση (απαιτείται μία και μόνο επίσκεψη ανά πελάτη)

b) Δεν πρέπει να οδηγούν σε πελάτες που παραβιάζουν το όριο της χωρητικότητας των 21.000 lt των οχημάτων

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΠΣΕ

- Αρχικοποιούμε την ημιτελή μας λύση:

$$s_0 = \{\{0\}, \{0\}\}$$

Όχημα 1 Όχημα 2

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΠΣΕ

- Από τις υποψήφιες συνδέσεις, οι οποίες ξεκινούν από τον τελευταίο κόμβο κάθε οχήματος, επιλέγουμε τη σύνδεση με τη μικρότερη απόσταση
- Επανάληψη 1. $s_1 = \{\{\mathbf{0}, \mathbf{1}\}, \{0\}\}$
- Επανάληψη 2. $s_2 = \{\{0, \mathbf{1}, \mathbf{6}\}, \{0\}\}$
- Επανάληψη 3. $s_3 = \{\{0, 1, 6\}, \{\mathbf{0}, \mathbf{5}\}\}$
- Επανάληψη 4. $s_4 = \{\{0, 1, 6\}, \{0, \mathbf{5}, \mathbf{4}\}\}$
- Επανάληψη 5. $s_5 = \{\{0, 1, \mathbf{6}, \mathbf{7}\}, \{0, 5, 4\}\}$
- Επανάληψη 6. $s_6 = \{\{0, 1, 6, \mathbf{7}, \mathbf{8}\}, \{0, 5, 4\}\}$

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΠΣΕ

– Επανάληψη 7. $s_7 = \{\{0,1,6,7,8\},\{0,5,\mathbf{4},\mathbf{3}\}\}$

– Επανάληψη 8. $s_8 = \{\{0,1,6,7,8\},\{0,5,4,\mathbf{3},\mathbf{9}\}\}$

– Επανάληψη 9. $s_9 = \{\{0,1,6,7,8\},\{0,5,4,3,\mathbf{9},\mathbf{2}\}\}$

– Επανάληψη 10. $S = s_{10} = \{\{0,1,6,7,\mathbf{8},\mathbf{0}\},\{0,5,4,3,9,\mathbf{2},\mathbf{0}\}\}$
(Επιστροφή στην αποθήκη και για τα δύο οχήματα)

Αντικειμενική Συνάρτηση τελικής λύσης: $\boldsymbol{z(S) = 311\ km}$

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕ ΠΣΕ

- Πλεονεκτικές στρατηγικές για το VRP:
  - Clarke & Wright heuristic
  - Sweep method

Clarke & Wright heuristic

- Με δεδομένο ένα σύνολο n κόμβων
  - Αποθήκη: 0
  - Πελάτες: $\{1, 2, \dots, n\}$
  - Πίνακας κόστους $c_{ij},\ i = 0, \dots, n,\ j = 0, \dots, n$
- Για κάθε σύνδεση i, j υπολογίζουμε την ποσότητα:

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}$$

- Στη συνέχεια κατατάσσουμε τις διάφορες συνδέσεις (i, j) σε φθίνουσα σειρά
- Δημιουργούμε ένα δρομολόγιο (όχημα) για κάθε πελάτη
  - {0, 1, 0}
  - {0, 2, 0}
  - {0, n, 0}

Clarke & Wright heuristic

- Σε κάθε επανάληψη της ΠΣΕ
  - Πάρε την ψηλότερη σύνδεση στην διατεταγμένη λίστα με τις συνδέσεις: (i, j) (για συμμετρικά προβλήματα θεώρησε μόνο συνδέσεις με $i < j$)
  - Ένωσε τα δρομολόγια που περιέχουν τους κόμβους (i, j) αν και μόνο αν
    - Ο κόμβος i και ο κόμβος j ανήκουν σε διαφορετικά δρομολόγια
    - Ο κόμβος i είναι ο πρώτος ή ο τελευταίος στο δρομολόγιο του
    - Ο κόμβος j είναι ο πρώτος ή ο τελευταίος στο δρομολόγιο του
    - Η ένωση των δρομολογίων των i και j δεν οδηγούν σε παραβίαση του περιορισμού χωρητικότητας
  - Τερμάτισε αν δεν επιτρέπονται άλλες συνδέσεις δρομολογίων λόγω της χωρητικότητας, ή εφόσον έχεις εξετάσει όλες τις συνδέσεις

Παράδειγμα Clarke & Wright heuristic

- Πίνακας Κόστους

c_{ij}	0	1	2	3	4	5	6	7	8	9
0	-	12	11	7	10	10	9	8	6	12
1	12	-	8	5	9	12	14	16	17	22
2	11	8	-	9	15	17	8	18	14	22
3	7	5	9	-	7	9	11	12	12	17
4	10	9	15	7	-	3	17	7	15	18
5	10	12	17	9	3	-	18	6	15	15
6	9	14	8	11	17	18	-	16	8	16
7	8	16	18	12	7	6	16	-	11	11
8	6	17	14	12	15	15	8	11	-	10
9	12	22	22	17	18	15	16	11	10	-

Παράδειγμα Clarke & Wright heuristic

- Ζήτηση Πελατών

i	1	2	3	4	5	6	7	8	9
d_i	10	15	18	17	3	5	9	4	6

- Χωρητικότητα Οχήματος: $Q = 40$

Παράδειγμα Clarke & Wright heuristic

- Υπολογισμός savings

s_{ij}	1	2	3	4	5	6	7	8	9
1		15	14	13	10	7	4	1	2
2			9	6	4	12	1	3	1
3				10	8	5	3	1	2
4					17	2	11	1	4
5						1	12	1	7
6							1	7	5
7								3	9
8									8

Παράδειγμα Clarke & Wright heuristic

- Κατάταξη savings σε φθίνουσα σειρά:

(4,5), (1,2), (1,3), (1,4), (2,6), (5,7), (4,7), (1,5), (3,4), (2,3), (7,9), (3,5), (8,9), (1,6), (5,9), (6,8), (2,4), ...

Παράδειγμα Clarke & Wright heuristic

- Σύνοψη των βημάτων

Edge [4,5]: Join cycles 0-4-0 and 0-5-0: result **0-4-5-0**, load $d_4 + d_5 = 20 < K$.
Edge [1,2]: Join 0-1-0 and 0-2-0: result **0-1-2-0**, load $d_1 + d_2 = 25 < K$.
Edge [1,3]: Capacity limit: $d_1+d_2+d_3 = 43 > K$.
Edge [1,4]: Capacity limit: $d_1+d_2+d_4 = 42 > K$.
Edge [2,6]: Join cycles 0-1-2-0 and 0-6-0: result **0-1-2-6-0**, load $d_1+d_2+d_6 = 30 < K$.
Edge [5,7]: Join cycles 0-4-5-0 and 0-7-0: result **0-4-5-7-0**, load $d_4+d_5+d_7 = 29 < K$.
Edge [4,7]: Condition 4(i) doesn't hold: nodes belong to same cycle.
Edge [1,5]: Condition 4(iii) doesn't hold: node 5 is an interior node of its route.
Edge [3,4]: Capacity limit: $d_3+d_4+d_5+d_7 = 47 > K$.
Edge [2,3]: Condition 4(iii) doesn't hold: node 2 is an interior node.
Edge [7,9]: Join cycles 0-4-5-7-0 and 0-9-0: result **0-4-5-7-9-0**,
load $d_4+d_5+d_7+d_9 = 35 < K$.
Edge [3,5]: Condition 4(iii) doesn't hold: node 5 is an interior node.
Edge [8,9]: Join cycles 0-4-5-7-9-0 and 0-8-0: result **0-4-5-7-9-8-0**,
load $d_4+d_5+d_7+d_9+d_8 = 39 < K$.
Edge [1,6]: Condition 4(i) doesn't hold: nodes belong to same cycle.
Edge [5,9]: Condition 4(i) doesn't hold: nodes belong to same cycle.
Edge [6,8]: Capacity limit: $(d_1+d_2+d_6)+(d_4+d_5+d_7+d_9+d_8) = 69 > K$.

Παράδειγμα Clarke & Wright heuristic

- Τελική Λύση $s =$

Route:	Load:	Cost:
0-3-0	18	14
0-1-2-6-0	30	37
0-4-5-7-9-8-0	39	46

- Με αντικειμενική συνάρτηση:

$$z(s) = 97$$

Παράδειγμα

- Εργάζεστε στο HR dept μία εταιρείας η οποία θέλει να προσλάβει ένα νέο στέλεχος
- Θα γίνουν τέσσερεις τελικές συνεντεύξεις, μία με κάθε υποψήφιο
- Ανάλογα με τις δεξιότητες των υποψηφίων στο πάνελ της συνέντευξης θα βρίσκονται διαφορετικά στελέχη της εταιρείας σας (συνάδελφοί σας)
- Οι συνεντεύξεις θα γίνουν στα 4 διαστήματα: [9:00-10:00],[10:00:11.00],[11:00:12.00],[12:00-13.00]
- Προτείνετε το πρόγραμμα των συνεντεύξεων, ώστε να ελαχιστοποιηθούν οι φορές που ένας συνάδελφός σας θα βρίσκεται σε συνέντευξη σε δύο διαδοχικές ώρες
- Ζητείται: Μορφή Λύσης, Αντικειμενική συνάρτηση
- Αν είχατε στην κατοχή σας έναν TSP solver, τι δεδομένα θα του δίνατε για να αντιμετωπίσετε το πρόβλημα;

	Υποψήφιος Α	Υποψήφιος Β	Υποψήφιος Γ	Υποψήφιος Δ
Έλενα	*			*
Πέτρος		*	*	
Μαρία	*	*	*	*
Γεωργία	*		*	
Δημήτρης		*	*	*