

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	(Ελληνικά) Προσεγγιστικοί και Ευρετικοί Αλγόριθμοι για το πρόβλημα του προσανατολισμού. (Αγγλικά) Approximation Algorithms and Heuristics for the Orienteering Problem
Όνοματεπώνυμο Φοιτητή	Δημήτριος Ορφανός
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	ΜΠΠΛ/11009
Επιβλέπων	Χαράλαμπος Κωνσταντόπουλος, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Οκτώβριος 2015**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
ΒαθμίδαΌνομα Επώνυμο
ΒαθμίδαΌνομα Επώνυμο
Βαθμίδα**Χαράλαμπος Κωνσταντόπουλος**
Επίκουρος Καθηγητής**Άγγελος Πικράκης**
Επίκουρος Καθηγητής**Μιχαήλ Ψαράκης**
Επίκουρος Καθηγητής

Δημήτρης Ορφανός ΜΠΠΛ/11009

Προσεγγιστικοί και Ευρετικοί Αλγόριθμοι για το πρόβλημα του Προσανατολισμού (Orienteering Problem)

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	σελ. 6
Abstract	σελ. 7
1 Εισαγωγή	σελ.8
1.2 Ορισμός σχετικών προβλημάτων	σελ. 9
1.2.1 K-stroll problem	σελ. 9
1.2.2 Min-Excess problem	σελ. 9
2 Από το k-stroll στο undirected Orienteering μέσω Min-Excess	σελ.10
2.1 Προσέγγιση του Min-excess προβλήματος	σελ. 10
2.2 Προσεγγιστικοί Παράγοντες για το Min-Excess	σελ. 12
2.3 Προσέγγιση του Orienteering Problem	σελ. 13
2.4 Μια 3-προσέγγιση του u-v Orienteering	σελ. 14
2.5 Μια (2+ε) προσέγγιση για το OP σε μη κατευθυνόμενα γραφήματα	σελ. 15
2.6 Απόδειξη θεωρήματος 2.5.1	σελ. 17
2.7 Προσεγγιστική πολυπλοκότητα του OP	σελ. 23
3 Directed OP (OP με κατευθυνόμενο γράφημα)	σελ. 24
3.1 Minimum Ratio ATSP(Asymmetric Traveling Salesman Problem)	σελ. 24
3.2 Directed k-Path Problem (ή k-stroll)	σελ. 24
3.3 Directed Min-Excess Problem	σελ. 25
3.3.1 Προσέγγιση του directed Min-excess προβλήματος	σελ. 25
3.4 Προσέγγιση του directed OP	σελ. 26
4 Weighted Orienteering Problem	σελ. 29
5 Orienteering Problem with Time-Windows(OPTW)	σελ. 31
5.1 Το Γενικό Πλαίσιο	σελ. 31
5.1.1 Δυναμικός Προγραμματισμός για modular στιγμιότυπα	σελ. 32
5.2 Αλγόριθμοι	σελ. 33
5.2.1 Μια $O(\alpha \cdot \log L_{\max})$ -προσέγγιση	σελ. 34
5.2.2 Μια $O(\alpha \cdot \log OPT)$ -προσέγγιση όταν $L \leq 2$	σελ. 34
5.3 Γενικός Αλγόριθμος για το OPTW	σελ. 35

6 Team Orienteering Problem (TOP)	σελ 36
6.1 Ορισμός του TOP	σελ. 36
6.2 Η τεχνική Greedy Randomized Adaptive Search Procedure (GRASP) στο Team Orienteering Problem	σελ. 37
6.2.1 Συνδυασμός GRASP με Path Relinking στο Team Orienteering Πρόβλημα	σελ. 38
6.3 Ο ευρετικός αλγόριθμος CGW για το Team Orienteering πρόβλημα των Chao et. al.	σελ. 39
6.3.1 Το βήμα της αρχικοποίησης (Initialization)	σελ. 40
6.3.2 Το βήμα της βελτίωσης (Improvement)	σελ. 40
6.3.2.1 Εναλλαγή δυο σημείων (two-point exchange)	σελ. 40
6.3.2.2 Μετακίνηση ενός σημείου (One point movement)	σελ. 41
6.3.2.3 Διαδικασία Clean up (Μείωση μήκους μονοπατιών)	σελ. 42
6.3.3 Διαδικασίες Διαταραχής της λύσης	σελ. 42
6.3.3.1 ReInitialization I	σελ. 43
6.3.3.2 ReInitialization II	σελ. 43
6.3.4 Ψευδοκώδικας του CGW ευρετικού αλγορίθμου	σελ. 43
7 Team Orienteering Problem με Time Windows (TOPTW)	σελ. 44
7.1 Ορισμός του Προβλήματος	σελ. 44
7.1.1 Μαθηματική Μοντελοποίηση του Team Orienteering Problem με Time Windows (TOPTW)	σελ.44
7.2 Ο αλγόριθμος Iterated Local Search (ILS) των Vansteenwegen et al.	σελ. 45
7.2.1 Βήμα Εισαγωγής (Insertion Step)	σελ. 46
7.2.2 Βήμα Διαταραχής (Shake Step)	σελ. 47
7.2.3 Ο ευρετικός αλγόριθμος	σελ.47
7.3 Ο αλγόριθμος Average Slack	σελ.48
7.4 Ο Υβριδικός αλγόριθμος GRASP-ELS για το TOPTW	σελ. 49
7.4.1 Insertion Heuristics	σελ. 49
7.4.2 Sweep Heuristics	σελ 51
7.4.3 Ο αλγόριθμος GRASP-ELS	σελ. 52
7.4.3.1 Διαδικασία Variable Neighborhood Descent	σελ. 53
7.4.3.2 Διαδικασία διαταραχής (Perturbation)	σελ. 55
7.5 Προτεινόμενος Αλγόριθμος	σελ. 55
7.5.1 Πειραματικά Αποτελέσματα	σελ. 58
8. Συμπεράσματα	σελ. 60
9. Αναφορές	σελ. 61
10. Αναλυτικά αποτελέσματα	σελ. 62

Περίληψη

Σε αυτήν την εργασία θα ασχοληθούμε με το Orienteering Problem (OP) ή Selective Traveling Salesman Problem (STSP) ή Maximum Collection Problem όπως είναι μερικές από τις ονομασίες του.

Το OP είναι ένα πρόβλημα βελτιστοποίησης και αποτελεί μια παραλλαγή του προβλήματος του περιοδεύοντος πωλητή (Traveling Salesperson problem ή TSP). Αυτού του είδους τα προβλήματα έχουν μεγάλο ενδιαφέρον καθώς υπάρχουν αρκετές εφαρμογές τους.

Το πρόβλημα παίρνει ως είσοδο ένα γράφημα με βάρη στις ακμές του, έστω $G=(V,E)$ (κατευθυνόμενο ή μη), δύο κορυφές s και t και ένα μη αρνητικό χρονικό όριο B . Το ζητούμενο είναι η εύρεση ενός μονοπατιού από το s στο t μήκους B το πολύ, έτσι ώστε να μεγιστοποιηθεί ο αριθμός των κόμβων που επισκεπτόμαστε μέσω του μονοπατιού [10]. Στον παραπάνω ορισμό του προβλήματος, η κορυφή έναρξης s και η κορυφή τερματισμού t είναι δεδομένες. Σε αυτήν την περίπτωση το πρόβλημα λέγεται σημείο προς σημείο (point to point) OP. Το OP έχει διάφορες παραλλαγές όπως rooted OP, unrooted OP, OP με ίδιο ή διαφορετικό όφελος ανά κόμβο κ.ο.κ. Το unrooted OP ορίζεται όπως παραπάνω με τη διαφορά ότι ο κόμβος έναρξης και ο τελικός κόμβος δεν δίνονται στο πρόβλημα. Σε αυτήν την παραλλαγή ο κόμβος έναρξης μπορεί να είναι οποιοσδήποτε κόμβος στο γράφημα. Στο rooted OP έχουμε έναν αρχικό κόμβο αλλά όχι τερματικό κόμβο.

Το OP ως πρόβλημα ανήκει στην κλάση των NP-hard προβλημάτων ([58], [14]). Αυτό σημαίνει ότι δεν έχει βρεθεί κάποιος ακριβής αλγόριθμος επίλυσής του που να τρέχει σε πολυωνυμικό χρόνο. Ωστόσο έχουν αναπτυχθεί προσεγγιστικοί αλγόριθμοι για την επίλυσή του. Σε αυτήν την εργασία θα μοντελοποιήσουμε το πρόβλημα και θα παρουσιάσουμε κάποια αποτελέσματα για τους μέχρι τώρα προσεγγιστικούς αλγόριθμους.

Επίσης θα παρουσιάσουμε έναν νέο ευρετικό αλγόριθμο ο οποίος δίνει λύσεις στο Team Orienteering Problem με Time Windows. Το πρόβλημα του ομαδικού προσανατολισμού (Team Orienteering problem) αποτελεί γενίκευση του προβλήματος προσανατολισμού (Orienteering problem). Δεδομένου ενός γραφήματος $G=(V,E)$ με βάρη στις ακμές του, δύο κορυφών s και t του G , ενός μη αρνητικού ορίου κόστους B και ενός ακέραιου m , το ζητούμενο είναι να βρεθούν m διαδρομές από την κορυφή s στην κορυφή t με κόστος μικρότερο ή ίσο του B η καθεμία έτσι ώστε να μεγιστοποιείται ο συνολικός αριθμός των κορυφών που βρίσκονται μέσα στις m διαδρομές. Το πρόβλημα του ομαδικού προσανατολισμού με χρονικά παράθυρα (time windows) εμπεριέχει τον περιορισμό ότι η επίσκεψη σε κάθε κορυφή πρέπει να γίνει εντός ενός χρονικού διαστήματος, που είναι δεδομένο για κάθε κορυφή.

Abstract

In this thesis, we present the Orienteering Problem (OP) or Selective Travelling Salesman Problem (STSP) or Maximum Collection Problem which are some other names of the problem.

Orienteering Problem is an optimization problem and it is a variant of Travelling Salesperson Problem (TSP). This kind of problems have a lot of interest because they are applicable in real life problems.

The problem takes as an input a (directed or undirected) graph with some non-negative weights in its edges. Let $G=(V,E)$ be the graph, s and t two given vertices and a non-negative time budget B . The goal is to find a path from s to t with time cost at most B , so as to maximize the number of vertices visited by the path. In this definition of the problem, s and t are given. In this case, the problem is called point to point OP. In addition, OP has some variants like rooted OP, unrooted OP, OP with the same or different profit to his nodes etc. Unrooted OP is defined like above with the difference that s and t are not given. In this variant of the problem, starting node and ending node s and t can be any node which belongs to V . In rooted OP, the starting node s is given but the ending node t is not.

OP is a NP-hard problem which means that an exact algorithm for its solution in polynomial time, has not been found. Therefore some approximation algorithms have been proposed for solving the problem in acceptable time. In this thesis, the problem is formulated and some results for approximation algorithms are presented.

We present a new heuristic algorithm for Team Orienteering Problem with Time Windows. Team Orienteering Problem is a generalization of OP. Given an additional positive integer parameter m , the goal is to find m distinct paths from s to t , having cost less than or equal to B , so as to maximize the total number of vertices visited by the m paths.

OP with Time Windows has an extra constraint: the visit at each node must occur within a specific time interval for each node.

1. ΕΙΣΑΓΩΓΗ

Έστω ότι έχουμε έναν πράκτορα με ένα χάρτη του περιβάλλοντος χώρου, ο οποίος πρέπει να επισκεφτεί ένα αριθμό τοποθεσιών για κάποιο λόγο (π.χ. να παραδώσει πακέτα, να συλλέξει δείγματα, να βρει ένα χαμένο αντικείμενο κτλ.). Ένα κλασικό μοντέλο αυτού του σεναρίου είναι το πρόβλημα του περιοδεύοντος πωλητή (TSP), στο οποίο αναζητούμε ένα κλειστό μονοπάτι που να περιέχει όλες τις τοποθεσίες (κόμβους) και το μήκος του είναι το ελάχιστο δυνατόν. Ωστόσο υπάρχουν περιπτώσεις στις οποίες ο πράκτορας (robot) δεν μπορεί να επισκεφτεί όλες τις τοποθεσίες. Για παράδειγμα, μπορεί να έχει χαμηλό απόθεμα μπαταρίας. Το πρόβλημα του προσανατολισμού (Orienteering Problem ή OP) ορίζεται ως το πρόβλημα που αναζητά έναν κύκλο που περνά από τις τοποθεσίες έτσι ώστε να μεγιστοποιεί το συνολικό όφελος επισκεψιμότητας με τον περιορισμό ότι το συνολικό μήκος δεν πρέπει να ξεπερνά ένα άνω φράγμα B. Το όφελος μπορεί να ανταποκρίνεται στην αξία του πακέτου που παραδίδεται ή στην πιθανότητα να βρίσκεται ένα χαμένο αντικείμενο σε μια συγκεκριμένη τοποθεσία.

Το Orienteering Problem παρουσιάστηκε από τον Tsiligirides [5] και ονομάστηκε έτσι από ένα άθλημα που ονομάζεται orienteering. Το OP ορίζεται ως εξής: Έστω $G=(V,E)$ ένα γράφημα με βάρη στις ακμές του και τιμές στους κόμβους που αναπαριστούν το όφελος της επίσκεψης του συγκεκριμένου κόμβου. Δεδομένων ενός αρχικού κόμβου s ενός τελικού κόμβου t και ενός θετικού χρονικού ορίου (budget) B, ο στόχος είναι η εύρεση ενός μονοπατιού από το s στο t (ή κύκλου εάν $s=t$) με μέγιστο μήκος B, έτσι ώστε να μεγιστοποιείται το συνολικό όφελος από την επίσκεψη των κόμβων.

Οι διάφορες παραλλαγές του παραπάνω προβλήματος έχουν να κάνουν με το εάν το γράφημα είναι κατευθυνόμενο ή μη, εάν οι κόμβοι έχουν το ίδιο ή διαφορετικό όφελος, αν δίνεται μόνο αρχικός κόμβος s (ρίζα) χωρίς τελικό κόμβο (rooted OP), εάν δεν δίνεται ούτε αρχικός κόμβος (unrooted OP) και συνδυασμός των παραπάνω.

Το πρόβλημα μοντελοποιείται ως εξής ([6]): Έστω N ο αριθμός των κόμβων με ονομασίες $1,2,\dots,N$ με $s=1$ και $t=N$, p_i το όφελος από την επίσκεψη του κόμβου i και c_{ij} το κόστος μετάβασης από τον κόμβο i στον κόμβο j. Για κάθε μονοπάτι από το 1 έως το N, ορίζουμε τη μεταβλητή x_{ij} ίση με 1 εάν ο κόμβος i ακολουθείται από τον κόμβο j στο μονοπάτι και 0 αλλιώς. Επίσης το u_i δηλώνει τη θέση του κόμβου i στο μονοπάτι. Με αυτόν τον συμβολισμό έχουμε τις παρακάτω σχέσεις:

$$\max \sum_{i=2}^{N-1} \sum_{j=2}^N p_i x_{ij} \quad (1)$$

με περιορισμούς:

$$\sum_{j=2}^N x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1 \quad (2)$$

$$\sum_{i=1}^{N-1} x_{ir} = \sum_{j=2}^N x_{rj} \leq 1 \quad \text{για όλα τα } r=2,\dots,N-1 \quad (3)$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^N c_{ij} x_{ij} \leq B \quad (4)$$

$$2 \leq u_i \leq N, \quad \text{για όλα τα } i=2,\dots,N \quad (5)$$

$$u_i - u_j + 1 \leq (N-1)(1 - x_{ij}) \quad \text{για όλα τα } i,j=2,\dots,N \quad (6)$$

$$x_{ij} \in \{0,1\} \quad \text{για όλα τα } i,j=1,\dots,N \quad (7)$$

Η αντικειμενική συνάρτηση (1) σημαίνει ότι μεγιστοποιούμε το συνολικό όφελος των κόμβων που επισκεπτόμαστε. Με τον περιορισμό (2) εξασφαλίζουμε ότι το μονοπάτι αρχίζει από τον κόμβο 1 και

τελειώνει στον κόμβο N . Ο περιορισμός (3) επιβεβαιώνει ότι το μονοπάτι που αρχίζει από τον κόμβο 1 και τελειώνει στον κόμβο N , είναι συνεκτικό και κάθε κόμβος διατρέχεται το πολύ μια φορά. Ο περιορισμός (4) εξασφαλίζει ότι το μήκος του μονοπατιού δεν υπερβαίνει το χρονικό όριο B (budget). Τέλος οι περιορισμοί (5) και (6) εξασφαλίζουν ότι δεν υπάρχουν κλειστά υπο-μονοπάτια.

Θα παρουσιάσουμε κάποια αποτελέσματα που αφορούν προσεγγιστικούς αλγόριθμους στο OP. Κάποιες παρατηρήσεις για προσεγγίσεις πάνω σε διάφορες παραλλαγές του OP είναι οι εξής :

- Στους προσεγγιστικούς αλγόριθμους για το OP, το γράφημα εισόδου μπορεί να περιοριστεί σε γραφήματα των οποίων οι κόμβοι έχουν μοναδιαία οφέλη καθώς μια προσέγγιση για αυτό το πρόβλημα, οδηγεί σε μια προσέγγιση του OP με βεβαρυμένους κόμβους (weighted OP) που έχει παράγοντα προσέγγισης $\alpha \cdot (1+O(1))$, όπου α είναι ο παράγοντας προσέγγισης του OP με μοναδιαία οφέλη.
- Ένας τρόπος για να προσεγγίσουμε το OP χωρίς ρίζα (unrooted OP) σε μη κατευθυνόμενους γράφους προκύπτει από προσεγγιστικούς αλγόριθμους για το πρόβλημα k -TSP (εύρεση μιας κυκλικής διαδρομής ελάχιστου μήκους η οποία διατρέχει τουλάχιστον k κόμβους). Η βασική ιδέα είναι να σπάσουμε μια τέτοια διαδρομή σε τμήματα μήκους το πολύ B και να πάρουμε αυτή με το μεγαλύτερο όφελος. (βλ. [9])

1.2 Ορισμός σχετικών προβλημάτων

Για την προσέγγιση του OP χρησιμοποιούμε προσεγγίσεις άλλων προβλημάτων. Τα προβλήματα αυτά είναι το Min-excess Problem και το k -stroll problem. Έχει αποδειχθεί ότι μια προσέγγιση για το πρόβλημα min-excess συνεπάγεται μια προσέγγιση και για το OP [7]. Επιπλέον το min-excess μπορεί να προσεγγιστεί χρησιμοποιώντας αλγόριθμους για το k -stroll. Σε αυτό το σημείο θα ορίσουμε αυτά τα προβλήματα.

1.2.1 k -Stroll Problem

Έστω γράφημα $G=(V,E)$ με βάρη στις ακμές του και μοναδιαία οφέλη επισκεψιμότητας για κάθε κόμβο. Ζητάμε ένα μονοπάτι από έναν αρχικό κόμβο s σε έναν τελικό κόμβο t έτσι ώστε να επισκεπτόμαστε τουλάχιστον k κόμβους με το ελάχιστο δυνατόν κόστος [10].

1.2.2 Min-excess Problem ([7],[10])

Έστω γράφημα $G=(V,E)$. Ορίζουμε μια συνάρτηση απόστασης στις ακμές $d:E \rightarrow \mathbb{R}^+$ και μια συνάρτηση οφέλους στους κόμβους $\pi:V \rightarrow \mathbb{R}^+$. Με $\pi(u)$ συμβολίζουμε το όφελος στον κόμβο u . Με s συμβολίζουμε τον αρχικό κόμβο ή ρίζα. Για ένα μονοπάτι P που επισκέπτεται τον κόμβο u πριν από τον v , η έκφραση $d^P(u,v)$ συμβολίζει το μήκος από τον u στο v μέσω του μονοπατιού P . Η έκφραση $d(u,v)$ συμβολίζει το μήκος της συντομότερης διαδρομής από το u στο v στο γράφημα G . Έστω ένα σύνολο κόμβων V'

υποσύνολο του V . Ορίζουμε $\pi(V') = \sum_{u \in V'} \pi(u)$ για κάθε u που ανήκει στο V' . Για ένα σύνολο ακμών E'

υποσύνολο του E ορίζουμε $d(E') = \sum_{e \in E'} d(e)$ για κάθε e που ανήκει στο E' .

Για να ορίσουμε το πρόβλημα min-excess, ορίζουμε επιπλέον ότι $\text{excess}(s,t)$ ενός μονοπατιού P από το s στο t να είναι η διαφορά $d^P(s,t) - d(s,t)$. Δηλαδή excess είναι η διαφορά μήκους του s - t μονοπατιού στο P από το μήκος της συντομότερης διαδρομής του s - t στο γράφημα.

Δεδομένου δυο κόμβων s και t και ενός ακεραίου k , στόχος μας είναι να βρούμε ένα μονοπάτι s - t με ελάχιστο $\text{excess}(s,t)$ το οποίο επισκέπτεται τουλάχιστον k κόμβους στην περίπτωση που όλοι οι κόμβοι έχουν μοναδιαίο όφελος. Αν οι κόμβοι έχουν όφελος, το k είναι το επιθυμητό όφελος του μονοπατιού.

¹ Περισσότερες λεπτομέρειες υπάρχουν στο κεφάλαιο 4

2 Από το k-stroll στο undirected Orienteering μέσω Min-Excess

Σε ένα πρόβλημα ελαχιστοποίησης όπως είναι τα k-stroll, min-excess όταν λέμε ότι έχουμε μια p -προσέγγιση (p σταθερά) του προβλήματος εννοούμε ότι η προσεγγιστική λύση $AP \leq p \cdot OPT$ (όπου OPT είναι η τιμή της βέλτιστης λύσης και AP η τιμή της προσεγγιστικής λύσης). Αντίστοιχα σε προβλήματα μεγιστοποίησης όπως το Orienteering, ένας αλγόριθμος είναι p -προσεγγιστικός εάν $AP \geq (OPT/p)$.

Έχει αποδειχθεί ότι σε μη κατευθυνόμενα γραφήματα μία β -προσέγγιση (β -approximation) στο k-stroll

πρόβλημα συνεπάγεται μια $\left(\frac{3}{2} \cdot \beta - \frac{1}{2}\right)$ -προσέγγιση στο min-excess πρόβλημα. Επίσης μια γ -

προσέγγιση στο min-excess πρόβλημα οδηγεί σε μια $\lceil \gamma \rceil$ -προσέγγιση στο Orienteering ([10],[7]).

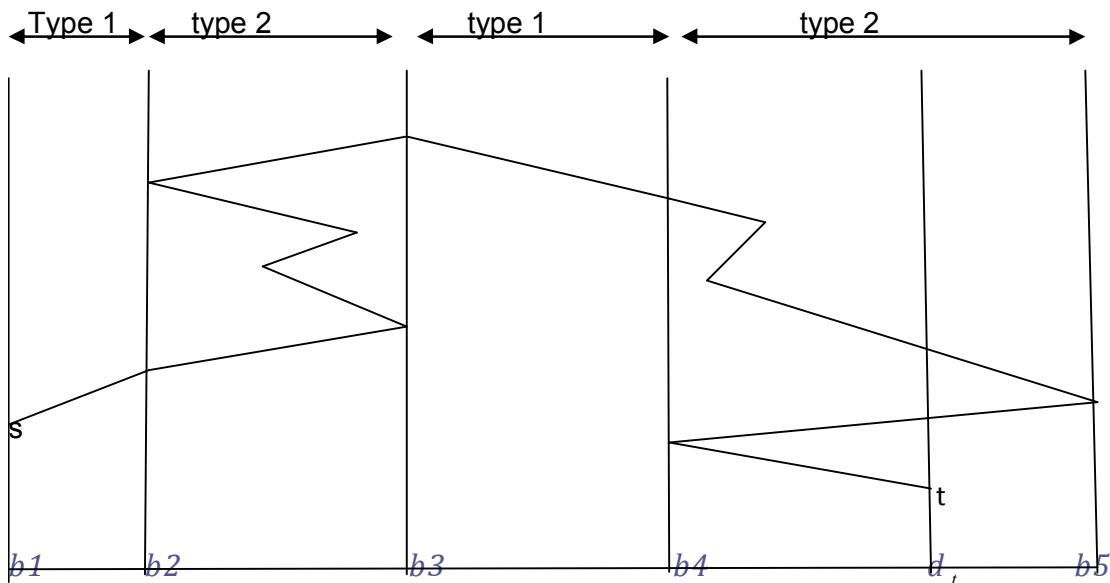
2.1 Προσέγγιση του Min-excess προβλήματος

Για την προσέγγιση του Min-excess προβλήματος στηρίζομαστε σε μια $(2+\epsilon)$ προσέγγιση για το k-stroll, όπως παρουσιάστηκε από τον Chaudhuri (βλ. [11]). Θα δείξουμε ότι μπορούμε να προσεγγίσουμε το min-excess με παράγοντα $a_{EP} = ((3/2) \cdot a_{CP} - 1/2)$, όπου a_{CP} ο παράγοντας προσέγγισης του k-stroll ([7],[10]).

Ένα βέλτιστο μονοπάτι το οποίο επισκέπτεται τους κόμβους του σε αύξουσα σειρά της απόστασής τους από την αρχή s , λέμε ότι είναι μονοτονικό. Το βέλτιστο μονοτονικό μονοπάτι μπορεί να βρεθεί με χρήση δυναμικού προγραμματισμού. Συγκεκριμένα για κάθε δυνατή τιμή οφέλους p και για κάθε κορυφή i σε αύξουσα σειρά απόστασης από τον αρχικό κόμβο s υπολογίζουμε το minimum excess μονοπάτι που αρχίζει από το s και καταλήγει στην i συλλέγοντας όφελος τουλάχιστον p . Ωστόσο οι λύσεις που ψάχνουμε μπορεί να μην αποτελούν μονοτονικά μονοπάτια. Σε αυτές τις περιπτώσεις χωρίζουμε το βέλτιστο μονοπάτι P σε διαστήματα τα οποία είτε είναι μονοτονικά είτε έχουν μεγάλο excess. Για τα μονοτονικά διαστήματα χρησιμοποιούμε δυναμικό προγραμματισμό και βρίσκουμε ακριβείς λύσεις σε αυτό το διάστημα. Για τα υπόλοιπα διαστήματα χρησιμοποιούμε τον αλγόριθμο του k-stroll.

Για κάθε πραγματικό αριθμό d , ορίζουμε τη συνάρτηση $f(d)$ ως τον αριθμό των ακμών του βέλτιστου μονοπατιού P , οι οποίες έχουν το ένα τους άκρο σε κόμβο που έχει απόσταση από την αρχή s μικρότερη από d και το άλλο άκρο σε απόσταση από το s μεγαλύτερη ή ίση από d .

Χωρίζουμε τη γραμμή των πραγματικών αριθμών σε μεγιστικά διαστήματα στα οποία είτε $f(d)=1$ είτε $f(d)>1$. Ονομάζουμε τα διαστήματα με $f(d)=1$ τύπου 1 και αυτά με $f(d)>1$ τύπου 2. Τα διαστήματα τύπου ένα αντιστοιχούν σε μονοτονικό κομμάτι ενώ τα διαστήματα τύπου δύο σε κομμάτι με μεγάλο excess. Αυτά τα διαστήματα διαμερίζουν τον άξονα των πραγματικών μέχρι τη μέγιστη απόσταση που φτάνει η βέλτιστη λύση. Τα διαστήματα εναλλάσσονται σε τύπου ένα και δύο. Η εικόνα 1 δίνει ένα παράδειγμα αυτής της διαμέρισης.



Εικόνα 1

Το d_i στην εικόνα 1 συμβολίζει τη συντομότερη απόσταση του τελικού κόμβου t από την αρχή s .

Έστω b_i να συμβολίζει το αριστερό άκρο του i διαστήματος. Ένα διάστημα από το b_i στο b_{i+1} είναι τύπου ένα εάν για κάθε d μέσα σε αυτό το διάστημα ισχύει $f(d)=1$. Έστω l_i το μήκος του μονοπατιού P στο i διάστημα.

Λήμμα 1: Σε κάθε διάστημα i το μήκος του μονοπατιού στο συγκεκριμένο διάστημα είναι τουλάχιστον $b_{i+1} - b_i$ αν είναι τύπου ένα, ενώ για τύπου δύο είναι τουλάχιστον $3 \cdot (b_{i+1} - b_i)$. Αν το διάστημα τύπου 2 περιέχει τον τελικό κόμβο t το μήκος του είναι τουλάχιστον $3 \cdot (d(t) - b_i)$.

Απόδειξη: Το μήκος του i διαστήματος έχει κάτω φράγμα το ολοκλήρωμα της $f(d)$ στο αντίστοιχο διάστημα. Για $f(d)=1$ (διαστήματα τύπου ένα) το αποτέλεσμα είναι προφανώς $(b_{i+1} - b_i)$. Παρατηρούμε ότι σε διαστήματα τύπου δύο πρέπει να ισχύει $f(d) \geq 3$ γιατί εάν το μονοπάτι διασχίζει απόσταση d μόνο δύο φορές, θα καταλήγει σε απόσταση μικρότερη από d . Παίρνοντας το ολοκλήρωμα παίρνουμε τουλάχιστον $3 \cdot (b_{i+1} - b_i)$. \square

Θεώρημα 1: [7] Το συνολικό μήκος των διαστημάτων τύπου δύο είναι το πολύ $3 \cdot \text{ex}(P)/2$ (όπου $\text{ex}(P)$ δηλώνει το excess του βέλτιστου μονοπατιού P).

Απόδειξη: Εξ ορισμού το μήκος του μονοπατιού P είναι $d_i + \text{ex}(P) = \sum l_i$. Επιπλέον

$$d_i \leq b_m = \sum_{i=1}^{m-1} (b_{i+1} - b_i) \leq \sum_{i \in \text{type1}} l_i + \sum_{i \in \text{type2}} \frac{l_i}{3} \quad (\text{βάσει του Λήμματος 1}). \text{ Άρα } \text{ex}(P) = \sum l_i - d_i \geq \sum_{i \in \text{type2}} \frac{2}{3} \cdot l_i$$

Πολλαπλασιάζοντας με $3/2$ προκύπτει το αποτέλεσμα. \square

Στη συνέχεια παρουσιάζεται ο προσεγγιστικός αλγόριθμος. Ο αλγόριθμος εφαρμόζει δυναμικό προγραμματισμό.

Για κάθε διάστημα που θα μπορούσε να ανήκει στη βέλτιστη λύση ο αλγόριθμος υπολογίζει ένα τμήμα που αντιστοιχεί στη λύση σε αυτό το διάστημα. Στη συνέχεια με δυναμικό προγραμματισμό ενώνει τα τμήματα χρησιμοποιώντας ακμές (που έχουν κόστος) οι οποίες διασχίζουν τα τμήματα. Τα τμήματα που υπολογίζονται, ορίζονται από 4 κορυφές. Αυτές είναι οι c και f που συμβολίζουν τα όρια του διαστήματος

τύπου 1 ή 2 του βέλτιστου μονοπατιού. Οι άλλες δυο κορυφές x και y , ορίζουν την πρώτη και τελευταία κορυφή του τμήματος της λύσης εντός του διαστήματος. Επίσης το κάθε τμήμα ορίζεται από το όφελος ρ που πρέπει να συγκεντρωθεί εντός του διαστήματος. Οι 4 κορυφές και το επιθυμητό όφελος ρ προσδιορίζονται με την τεχνική του guessing. Αυτό σημαίνει ότι δοκιμάζονται όλοι οι πιθανοί συνδυασμοί.

Για κάθε τμήμα υπολογίζεται μια βέλτιστη λύση για διάστημα τύπου 1 και προσεγγιστική λύση για διάστημα τύπου 2. Για διάστημα τύπου 1 το βέλτιστο μονοπάτι είναι μονοτονικό. Συνεπώς μπορεί να υπολογιστεί το βέλτιστο (συντομότερο) μονοτονικό μονοπάτι από το x στο y που συλλέγει όφελος ρ . Αν το διάστημα είναι τύπου 2 χρησιμοποιείται ο αλγόριθμος του k-stroll για να προσεγγίσει το μονοπάτι ελαχίστου μήκους που αρχίζει από το x και καταλήγει στο y , συλλέγει όφελος ρ και μένει εντός των ορίων του διαστήματος c και f .

Με δεδομένα τα βέλτιστα τμήματα τύπου 1 και τα κατά προσέγγιση τμήματα τύπου 2 μπορεί να βρεθεί ένας βέλτιστος τρόπος ώστε να επικολληθεί ένα υποσύνολό τους με δυναμικό προγραμματισμό.

Λήμμα: [7] Μία β -προσέγγιση του k-stroll συνεπάγεται μία $(3/2 \cdot \beta - 1/2)$ -προσέγγιση του Min-excess προβλήματος.

Απόδειξη: Έστω L το μήκος του βέλτιστου μονοπατιού P στο min-excess πρόβλημα. Εστιάζουμε στα διαστήματα που ανταποκρίνονται στο βέλτιστο μονοπάτι P . Με l_i συμβολίζουμε το μήκος του βέλτιστου μονοπατιού στο διάστημα i . Εάν το διάστημα είναι τύπου ένα τότε ο αλγόριθμος θα βρει το βέλτιστο μονοπάτι σε αυτό το διάστημα χρησιμοποιώντας δυναμικό προγραμματισμό. Στα διαστήματα τύπου δύο χρησιμοποιούμε έναν προσεγγιστικό αλγόριθμο του k-stroll που συλλέγει ένα επιθυμητό όφελος ελαχιστοποιώντας το μήκος. Με L_1 συμβολίζουμε το ολικό μήκος των μονοπατιών στα διαστήματα τύπου ένα. Με L_2 συμβολίζουμε το μήκος του βέλτιστου μονοπατιού στα διαστήματα τύπου δύο. Άρα το μήκος του μονοπατιού P' που θα βρούμε από τον προσεγγιστικό αλγόριθμο (για το min-excess) θα είναι το πολύ $L_1 + \beta \cdot L_2$. Ισχύει προφανώς ότι $L_1 = L - L_2$. Έχουμε $\text{excess}(P') = L_1 + \beta \cdot L_2 - d(t) = L - L_2 + \beta \cdot L_2 - d(t) = L - d(t) + (\beta - 1) \cdot L_2 = (\text{εξ ορισμού}) \text{excess}(P) + (\beta - 1) \cdot L_2 \leq \text{excess}(P) + (\beta - 1) \cdot 3/2 \cdot \text{excess}(P) = (3/2 \cdot \beta - 1/2) \text{excess}(P)$.

Συνεπώς $\text{excess}(P') \leq (3/2 \cdot \beta - 1/2) \text{excess}(P)$ \square

2.2 Προσεγγιστικοί Παράγοντες για το Min-Excess

Στην εργασία του Blum et al. [7] παρουσιάζεται μια βελτίωση της προσέγγισης του Min-Excess από $(2.5 + \epsilon)$ σε $(2 + \epsilon)$ για κάθε σταθερό ϵ θετικό. Το αποτέλεσμα αυτό στηρίζεται σε έναν βελτιωμένο αλγόριθμο για το πρόβλημα του min-cost (k-path) μονοπατιού όπως παρουσιάστηκε από τον Chaudhuri στην εργασία [11].

Ο Chaudhuri και οι συνεργάτες του [11] παρέχουν έναν αλγόριθμο για την κατασκευή ενός δέντρου, που περιέχει τους κόμβους έναρξης και τερματισμού s και t αντίστοιχα, το οποίο συλλέγει όφελος τουλάχιστον k και έχει κόστος το πολύ $(1 + \frac{1}{2} \cdot \delta)$ φορές το κόστος του βέλτιστου k-path μονοπατιού P^* από το s στο t .

Διπλασιάζοντας τις ακμές αυτού του δέντρου προκύπτει μια προσέγγιση του k-path με παράγοντα προσέγγισης $a_{CP} = (2 + \delta)$.

Αν το βέλτιστο μονοπάτι P^* έχει μήκος $l = d(s, t) + \epsilon^*$, τότε το δέντρο έχει μήκος το πολύ

$(1 + \frac{1}{2} \cdot \delta) \cdot l = (1 + \frac{1}{2} \cdot \delta) \cdot (d(s, t) + \epsilon^*)$. Μετατρέπουμε αυτό το δέντρο σε ένα s-t μονοπάτι

διπλασιάζοντας όλες τις ακμές εκτός από εκείνες στο μονοπάτι του δέντρου από το s στο t .

Παρατηρώντας ότι το μήκος των ακμών που δεν διπλασιάστηκαν είναι τουλάχιστον $d(s, t)$, προκύπτει ένα μονοπάτι από το s στο t μήκους το πολύ:

$$2 \cdot \left(1 + \frac{1}{2} \cdot \delta\right) \cdot (d(s, t) + \varepsilon^*) - d(s, t) = (2 + \delta) \cdot (d(s, t) + \varepsilon^*) - d(s, t) = (1 + \delta) \cdot (d(s, t) + \varepsilon^*) + \varepsilon^* = (1 + \delta) \cdot l + \varepsilon^* .$$

Η δεύτερη ισότητα προκύπτει με πράξεις.²

Αυτή η ισχυρότερη προσέγγιση του μήκους στο k-path δίνει μια καλύτερη προσέγγιση στο Min-Excess. Υποθέτουμε ότι εφαρμόζουμε τον αλγόριθμο του k-path σε ένα διάστημα τύπου 2 με άκρα u και v, και έχοντας ένα βέλτιστο μονοπάτι του Min-Excess με μήκος $l = d(u, v) + \varepsilon$. Τότε παίρνουμε ένα μονοπάτι από το u στο v με ίδιο όφελος και μήκος το πολύ $(1 + \delta) \cdot l + \varepsilon$ για κάθε σταθερά δ. Πριν εφαρμόσουμε το Θεώρημα 1 της παραγράφου 2.1 για να πάρουμε την προσέγγιση για το excess, παρατηρούμε ότι το ε της τελευταίας έκφρασης συμβολίζει το excess από το u στο v και διαφέρει από τη διαφορά του excess του βέλτιστου μονοπατιού P* του v μείον του u. Έστω ε_u το excess του P* από το s στο u και ε_v το excess από το s στο v. Τότε το μήκος του P* σε αυτό το διάστημα θα είναι $l = (d_u + \varepsilon_v) - (d_u + \varepsilon_u) \leq d(u, v) + \varepsilon_v - \varepsilon_u$. Άρα $\varepsilon \leq \varepsilon_v - \varepsilon_u$ (1) και ο αλγόριθμος του Chaudhuri et al. επιστρέφει ένα μονοπάτι μήκους το πολύ $(1 + \delta) \cdot l + \varepsilon_v - \varepsilon_u$.

Συνεπώς το δυναμικό πρόγραμμα βρίσκει ένα μονοπάτι ίδιου οφέλους με το P* και συνολικού μήκους :

$$L1 + (1 + \delta) \cdot L2 + \sum_{type2} (\varepsilon_v - \varepsilon_u) \leq L1 + (1 + \delta) \cdot L2 + \varepsilon(P^*) \text{ (λόγω της 1)} = L1 + L2 + \delta \cdot L2 + \varepsilon(P^*) = d_t + 2 \cdot \varepsilon(P^*) + \delta \cdot L2$$

(γιατί $L1 + L2 = d_t + \varepsilon(P^*) \leq d_t + 2 \cdot \varepsilon(P^*) + \frac{3}{2} \cdot \delta \cdot \varepsilon(P^*)$). Η τελευταία ανισότητα προκύπτει από το Θεώρημα 1 της παραγράφου 2.1. Συνεπώς προκύπτει μια $(2 + \delta)$ -προσέγγιση του Min-Excess προβλήματος για κάθε δ' θετική σταθερά.

2.3 Προσέγγιση του Orienteering Problem

Το παρακάτω Λήμμα ισχύει τόσο για κατευθυνόμενα όσο και μη κατευθυνόμενα γραφήματα.

Λήμμα 2.3 ([10]): Μια γ-προσέγγιση του Min-Excess προβλήματος συνεπάγεται μια $\lceil \gamma \rceil$ -προσέγγιση για το Orienteering με μοναδιαίο όφελος στους κόμβους.

Απόδειξη: Έστω P το βέλτιστο μονοπάτι το οποίο επισκέπτεται OPT κόμβους. Το χωρίζουμε σε διαδοχικά κομμάτια P_1, P_2, \dots, P_h όπου $h = \lceil \gamma \rceil$. Κάθε κομμάτι θα περιέχει OPT/h κόμβους (κορυφές). Για κάθε κομμάτι P_i έστω ότι γνωρίζουμε τον πρώτο και τελευταίο κόμβο του κομματιού. Συμβολίζουμε s_i και t_i αντίστοιχα. Μπορούμε να 'μαντέψουμε' τα s_i και t_i δοκιμάζοντας όλες τις δυνατές επιλογές. Με $ex(i)$ συμβολίζουμε το τοπικό excess του κομματιού P_i . Δηλαδή $ex(i) = d^P(s_i, t_i) - d(s_i, t_i)$. Έστω ότι P_j είναι το κομμάτι με τη μικρότερη τιμή του excess με αρχικό και τελικό κόμβο s_j και t_j αντίστοιχα.

Ισχύει $h \cdot ex(j) \leq \sum_{i=1}^h ex(i) = d^P(s, t) - \sum_{i=1}^h d(s_i, t_i)$ Χρησιμοποιούμε τον προσεγγιστικό αλγόριθμο του Min-Excess για να βρούμε ένα νέο $s_j - t_j$ μονοπάτι P_j' που θα επισκέπτεται τουλάχιστον OPT/h κόμβους.

² Συγκεκριμένα $(2 + \delta) \cdot (d(s, t) + \varepsilon^*) - d(s, t) = d(s, t) + 2 \cdot \varepsilon^* + \delta \cdot d(s, t) + \delta \cdot \varepsilon^* = (1 + \delta) \cdot d(s, t) + \varepsilon^* + \delta \cdot \varepsilon^* + \varepsilon^* = (1 + \delta) \cdot d(s, t) + (1 + \delta) \cdot \varepsilon^* + \varepsilon^* = (1 + \delta) \cdot (d(s, t) + \varepsilon^*) + \varepsilon^*$

Αφού ο αλγόριθμος για το min-excess έχει παράγοντα προσέγγισης γ τότε το P_j θα έχει το πολύ $\gamma \leq h$ φορές περισσότερο excess από το P_j . Δημιουργούμε ένα νέο μονοπάτι P' το οποίο ξεκινά από τον αρχικό κόμβο s , πηγαίνει κατευθείαν στο s_j , ακολουθεί το P_j από το s_j στο t_j και στη συνέχεια από το

t_j στο t . Το συνολικό μήκος του P' θα είναι το πολύ $\sum_{i=1}^h d(s_i, t_i) + h \cdot \text{ex}(j) \leq d^P(s, t) = \text{length}(P)$. Άρα θα

προκύψει ένα μονοπάτι που το μήκος του δε θα υπερβαίνει το όριο κόστους και θα επισκέπτεται το πολύ OPT/h κορυφές. Συνεπώς έχουμε μια $\lceil \gamma \rceil$ -προσέγγιση του Orienteering. \square

Συγκεντρώνοντας τα μέχρι τώρα αποτελέσματα προκύπτει ότι η βελτιωμένη προσέγγιση του Min-Excess με παράγοντα $(2+\epsilon)$ οδηγεί σε μια 3-προσέγγιση του Orienteering σύμφωνα με το Λήμμα 2.3. Αυτή η προσέγγιση παρουσιάζεται στην ενότητα 2.4. Ωστόσο έχει αποδειχθεί ότι υπάρχει ακόμα καλύτερη προσέγγιση για το OP (συγκεκριμένα $2+\epsilon$). Το αποτέλεσμα αυτό παρουσιάστηκε από τον Chekuri και συνεργάτες [10] εστιάζοντας στο k-stroll πρόβλημα (δηλ. να βρεθεί το ελάχιστο μονοπάτι από το s στο t το οποίο επισκέπτεται τουλάχιστον k κόμβους) και δίνοντας δι-κριτηριακές προσεγγίσεις για το k-stroll σύμφωνα με το μήκος του μονοπατιού και τον αριθμό των κόμβων που επισκεπτόμαστε. Στην ενότητα 2.5 θα παρουσιάσουμε την $(2+\epsilon)$ προσέγγιση του Orienteering.

2.4 Μια 3-προσέγγιση του u-v Orienteering

Για λόγους πληρότητας παρουσιάζεται η εργασία [12] των Blum και συνεργατών για το πρόβλημα σημείο προς σημείο orienteering (ή u-v Orienteering). Παρουσιάζουν έναν αλγόριθμο που οδηγεί σε μια προσέγγιση με παράγοντα 3 όπως προέκυψε και στην ενότητα 2.3. Ο αλγόριθμος χρησιμοποιεί ως υπορουτίνα μια $(2+\epsilon)$ -προσέγγιση του min-excess προβλήματος η οποία παρουσιάστηκε από το Blum [7].

Ορισμός: u-v orienteering είναι το OP όπως ορίστηκε παραπάνω με σημείο εκκίνησης (ρίζα) τον κόμβο u και τελικό σημείο τον κόμβο v . Δηλαδή ψάχνουμε ένα μονοπάτι από τον u στο v έτσι ώστε το κόστος να μην υπερβαίνει το άνω φράγμα B (budget) και να μεγιστοποιείται το συνολικό όφελος επισκεψιμότητας.

Ακολουθεί ο αλγόριθμος για την 3-προσέγγιση του OP όπως παρουσιάστηκε στην εργασία [12].

Αλγόριθμος:

Είσοδος: Γράφημα $G=(V,E)$, κόμβοι u και v , όριο κόστους B

Έξοδος: μονοπάτι P με $\lceil P \rceil \geq \frac{1}{3} \lceil OPT \rceil$ και μήκος το πολύ B

1. Για κάθε ζεύγος κόμβων x και y , θεωρούμε ένα μονοπάτι που ξεκινά από τον u στον x , στη συνέχεια επισκέπτεται κάποιους κόμβους καθώς πηγαίνει από τον x στον y και τέλος πηγαίνει απευθείας από τον y στον v . Το επιτρεπόμενο excess του μονοπατιού από το x στο y είναι $\text{excess}(x,y) = B - d(u,x) - d(x,y) - d(y,v)$. Χρησιμοποιούμε την $(2+\epsilon)$ -προσέγγιση του min-excess³ όπως προκύπτει από την εργασία [7], υπολογίζουμε ένα μονοπάτι από το x στο y με excess το πολύ $\text{excess}(x,y)$ το οποίο να επισκέπτεται τουλάχιστον τόσους κόμβους όσο το βέλτιστο μονοπάτι από το x στο y με excess ίση με $\frac{1}{3}\text{excess}(x,y)$.

2. Επιλέγουμε το ζεύγος (x,y) που μεγιστοποιεί το συνολικό όφελος του μονοπατιού. Στη συνέχεια πηγαίνουμε από το u στο x μέσω μιας γραμμής, μετά από το x στο y και τέλος από τον y στον v .

Θεώρημα 2: Ο παραπάνω αλγόριθμος αποτελεί μια 3-προσέγγιση στο point-to-point Orienteering.

Για να αποδείξουμε το Θεώρημα 2 θα παρουσιάσουμε πρώτα δύο λήμματα όσον αφορά κάποιες ιδιότητες του excess.

Έστω P τυχαίο μονοπάτι από το u στο v το οποίο επισκέπτεται διαδοχικά τους κόμβους

$$u = u_0, u_1, u_2, \dots, u_n = v$$

³ Η $2+\epsilon$ προσέγγιση του προβλήματος Min-Excess περιγράφεται στην ενότητα 2.2

Λήμμα 1: Το $excess^P(u_0, u_i)$ αυξάνεται όσο αυξάνεται το i .

Λήμμα 2: Για οποιουδήποτε κόμβους u_i, u_j, u_k με $0 \leq i \leq j \leq k \leq n$ ισχύει $excess^P(u_i, u_j) + excess^P(u_j, u_k) \leq excess^P(u_i, u_k)$ (Η συνάρτηση excess είναι υπο-προσθετική/sub-additive).

Απόδειξη Λημμάτων

Το πρώτο Λήμμα προκύπτει από την τριγωνική ανισότητα.

Για το δεύτερο Λήμμα γράφουμε :

$$\begin{aligned} excess^P(u_i, u_j) + excess^P(u_j, u_k) &= d^P(u_i, u_j) - d(u_i, u_j) + d^P(u_j, u_k) - d(u_j, u_k) = \\ d^P(u_i, u_k) - d(u_i, u_j) - d(u_j, u_k) &= d^P(u_i, u_k) - (d(u_i, u_j) + d(u_j, u_k)) \leq d^P(u_i, u_k) - d(u_i, u_k) \\ &= excess^P(u_i, u_k). \end{aligned}$$

Η ανισότητα προκύπτει λόγω της τριγωνικής ανισότητας. Ισχύει δηλαδή $d(u_i, u_j) + d(u_j, u_k) \geq d(u_i, u_k)$

□

Απόδειξη Θεωρήματος 2

Εκ κατασκευής ο αλγόριθμος παράγει ένα μονοπάτι μήκους το πολύ B . Αρκεί να δείξουμε ότι επισκέπτεται αρκετά σημεία.

Έστω OPT το βέλτιστο μονοπάτι από το u στο v . Χωρίζουμε το μονοπάτι σε τρία κομμάτια έτσι ώστε το καθένα να έχει τουλάχιστον το $\frac{1}{3}$ της συνολικού οφέλους και έστω x και y να είναι τα οριακά σημεία του μέρους με το μικρότερο excess (συμβ. $ex^{OPT}(x, y)$). Θεωρούμε τώρα ένα μονοπάτι O το οποίο πηγαίνει από απευθείας το u στο x , στη συνέχεια ακολουθεί το OPT από το x στο y και τέλος πηγαίνει απευθείας από το y στο v . Αφού επιλέξαμε τα x και y έτσι ώστε το excess να είναι το μικρότερο από τα τρία κομμάτια, και λόγω της τριγωνικής ανισότητας, πρέπει να ισχύει ότι ακολουθώντας το μονοπάτι O από το OPT εξοικονομούμε επιπλέον κόστος (excess) τουλάχιστον $2 \cdot ex^{OPT}(x, y)$. Δηλαδή $d^{OPT}(u, v) - d^O(u, v) \geq 2 \cdot ex^{OPT}(x, y)$ (8). Το κόστος αυτό είναι αρκετό να επικαλύψει το κόστος που προστίθεται από την εφαρμογή του προσεγγιστικού αλγορίθμου min-excess από το x στο y . Εξ ορισμού έχουμε $excess(x, y) = d^{OPT}(u, v) - d^O(u, v) + ex^{OPT}(x, y)$. Λόγω της ανισότητας (8) $excess(x, y) \geq 3 \cdot ex^{OPT}(x, y)$. Αφού ο αλγόριθμος για το min-excess πρόβλημα συλλέγει τουλάχιστον τόσο όφελος όσο το βέλτιστο μονοπάτι από το x στο y με excess το πολύ $\frac{1}{3} \cdot excess(x, y)$, είναι εγγυημένο ότι θα συλλέγει τουλάχιστον τόσο όσο το όφελος αυτού του κομματιού του βέλτιστου μονοπατιού δηλαδή $\frac{1}{3} \cdot \lceil OPT \rceil$. Επίσης το excess αυτού του μονοπατιού θα είναι το πολύ $excess(x, y)$ δηλαδή κάτω από το επιτρεπόμενο όριο για το excess και επομένως το μήκος του θα είναι κάτω από B . □

2.5 Μια (2+ε) προσέγγιση για το OP σε μη κατευθυνόμενα γραφήματα

Η (2+ε)-προσέγγιση προκύπτει πάλι μέσω του Min-Excess όπως περιγράψαμε παραπάνω αντικαθιστώντας την προσέγγιση του k-stroll με μια δικριτηριακή προσέγγιση όπως παρουσιάζεται από τους Chekuri και συνεργάτες [10].

Λέγοντας ότι κάποιος αλγόριθμος αποτελεί μια δικριτηριακή (α,β)-προσέγγιση του k-stroll εννοούμε ότι δεδομένου ενός γραφήματος G , δυο κόμβων s και t και ενός ακέραιου k , ο αλγόριθμος επιστρέφει ένα μονοπάτι $s-t$ που περιέχει τουλάχιστον k/α κορυφές και έχει μήκος το πολύ β φορές μεγαλύτερο από το μήκος του βέλτιστου μονοπατιού που επισκέπτεται k κόμβους.

Αρχικά θα αποδείξουμε ένα Λήμμα που φράσει το τοπικό excess στα διαστήματα τύπου 2. Θα χρησιμοποιήσουμε αυτό το Λήμμα στην απόδειξη του Θεωρήματος 2.5.2. Για κάθε διάστημα i ορίζουμε

το τοπικό excess του διαστήματος $ex(i)$ από την κορυφή u (που έχει απόσταση b_i από την κορυφή έναρξης s) στην κορυφή v (με απόσταση b_{i+1} από την s)⁴ ως την αύξηση του excess στο μονοπάτι P όταν πηγαίνει από το u στο v . Δηλαδή $ex(i) = excess^P(v) - excess^P(u)$.

Λήμμα 2.5.1[10]: Σε κάθε διάστημα τύπου 2 του μονοπατιού P ισχύει $ex(i) \geq \max\{l_i - d_i, 2l_i/3\}$ όπου $ex(i)$ η τοπική αύξηση του excess στο διάστημα i , l_i το μήκος του P στο διάστημα i και d_i η συντομότερη διαδρομή στο διάστημα i .

Απόδειξη: Έχουμε $ex(i) = (d^P(v) - d(v)) - (d^P(u) - d(u)) = (d^P(v) - d^P(u)) - (d(v) - d(u)) =$

$= l_i - (b_{i+1} - b_i)$ Από το Λήμμα 1 στην παράγραφο 2.1 ισχύει $l_i \geq 3(b_{i+1} - b_i)$ για διαστήματα τύπου 2.

Συνεπάγεται $(b_{i+1} - b_i) \leq \frac{l_i}{3} \Leftrightarrow l_i - (b_{i+1} - b_i) \geq l_i - \frac{l_i}{3} \Leftrightarrow ex(i) \geq \frac{2}{3}l_i$. Επιπλέον το συντομότερο μονοπάτι d_i

μεταξύ δυο κόμβων u και v που περιέχονται στο διάστημα i είναι τουλάχιστον $(b_{i+1} - b_i)$. Άρα $l_i - (b_{i+1} - b_i) \geq l_i - d_i$. Το αποτέλεσμα ισχύει και στο τελευταίο διάστημα με τη διαφορά ότι αντί για $(b_{i+1} - b_i)$ θα έχουμε $d(t) - b_i$. □

Θεώρημα 2.5.1 [10]: Για κάθε σταθερό $\varepsilon > 0$ υπάρχει ένας αλγόριθμος πολυπλοκότητας $O(n^{O(1/\varepsilon^2)})$ ο οποίος, δεδομένου ενός γραφήματος G , δυο κορυφών s και t και ενός ακεραίου k , βρίσκει ένα μονοπάτι από το s στο t μήκους το πολύ $\max\{1.5D, 2L - D\}$ που επισκέπτεται τουλάχιστον $(1-\varepsilon)k$ κορυφές όπου L είναι το μήκος του βέλτιστου s - t μονοπατιού που επισκέπτεται τουλάχιστον k κορυφές (k -stroll) και D είναι η συντομότερη διαδρομή από το s στο t .

Η απόδειξη του θεωρήματος υπάρχει στην παράγραφο 2.6.

Θεώρημα 2.5.2: Για κάθε σταθερό $\varepsilon > 0$ υπάρχει ένας αλγόριθμος πολυωνυμικού χρόνου ο οποίος βρίσκει ένα μονοπάτι από το s στο t που επισκέπτεται τουλάχιστον $(1-\varepsilon)k$ κορυφές με excess το πολύ διπλάσιο από αυτό του βέλτιστου μονοπατιού P που επισκέπτεται k κορυφές.

Απόδειξη: Έστω P το μονοπάτι της βέλτιστης λύσης για το Min-excess. Στα διαστήματα τύπου 1 βρίσκουμε το βέλτιστο μονοπάτι λόγω μονοτονικότητας. Στα διαστήματα τύπου 2 χρησιμοποιούμε τη δικριπριακή προσέγγιση του k -stroll. Με P' συμβολίζεται το μονοπάτι που προκύπτει από τα βέλτιστα τμήματα στα διαστήματα τύπου 1 και από την προσεγγιστική λύση στα διαστήματα τύπου 2 όπου χρησιμοποιείται η δικριπριακή προσέγγιση του k -stroll. Έχουμε μια $(1-\varepsilon)$ προσέγγιση ως προς τον αριθμό των κορυφών που επισκεπτόμαστε. Αυτό συνεπάγεται ότι το P' θα έχει τουλάχιστον $(1-\varepsilon)k$ κορυφές. Θα βρούμε ένα φράγμα του μήκους του P' . Έχουμε:

$$\begin{aligned} \text{length}(P') &\leq \sum_{i \in \text{type1}} l_i + \sum_{i \in \text{type2}} \max\{1.5d_i, 2l_i - d_i\} = \sum_{i \in \text{type1}} l_i + \sum_{i \in \text{type2}} \max\{1.5d_i, 2l_i - d_i\} + \sum_{i \in \text{type2}} l_i - \\ &\sum_{i \in \text{type2}} l_i \leq \sum_i l_i + \sum_{i \in \text{type2}} \max\{0.5d_i, l_i - d_i\} \\ &\leq \sum_i l_i + \sum_{i \in \text{type2}} ex(i) \leq \text{length}(P) + excess^P(t) = d(t) + 2 \cdot excess^P(t) \end{aligned}$$

Η πρώτη ανισότητα προκύπτει από το θεώρημα 2.5.1. Η δεύτερη ανισότητα προκύπτει αφαιρώντας το $\max\{1.5d_i, 2l_i - d_i\}$ με το l_i . Θα έχουμε $\max\{1.5d_i - l_i, l_i - d_i\}$. Ισχύει $1.5d_i - l_i \leq 1.5d_i - d_i = 0.5d_i$ (γιατί $d_i \leq l_i$). Η τρίτη ανισότητα προκύπτει από το Λήμμα 2.5.1. Τέλος έχουμε $\text{length}(P') \leq d(t) + 2 \cdot excess^P(t) \Leftrightarrow \text{length}(P') - d(t) \leq 2 \cdot excess^P(t) \Leftrightarrow$

⁴ Οι κορυφές u , v αποτελούν την πρώτη και τελευταία κορυφή του τμήματος της λύσης εντός του διαστήματος $[b_i, b_{i+1}]$ για κάθε i

$\text{excess}(P') \leq 2 \cdot \text{excess}^P(t)$. Η τελευταία σχέση αποδεικνύει το ζητούμενο \square

Στη συνέχεια θα αποδείξουμε ότι ο αλγόριθμος αυτός δίνει μια $(2+\varepsilon)$ προσέγγιση για το OP σε μη κατευθυνόμενα γραφήματα. Αναδιατυπώνουμε το Λήμμα 2.3 για την περίπτωση της δικριτιριακής προσέγγισης του Min-excess.

Λήμμα 2.5.2: Μια (α, β) προσέγγιση του Min-Excess δίνει μια $\lceil \beta \rceil$ -προσέγγιση στο Orienteering πρόβλημα.

Η απόδειξη του λήμματος 2.5.2 είναι ακριβώς η ίδια με την απόδειξη του θεωρήματος 4 στην ενότητα 3.4 και ισχύει και για τη μη κατευθυνόμενη εκδοχή του προβλήματος. Στην απόδειξη χρησιμοποιείται η ιδιότητα της υποπροσθετικότητας της excess που αποδείχθηκε στο λήμμα 2 της ενότητας 2.4 και ισχύει σε κατευθυνόμενα και μη κατευθυνόμενα γραφήματα.

Έχοντας συγκεντρώσει αυτά τα αποτελέσματα μπορούμε να διατυπώσουμε το παρακάτω θεώρημα που εξασφαλίζει μια $(2+\varepsilon)$ προσέγγιση στο Orienteering Problem.

Θεώρημα 2: Για κάθε σταθερό $\varepsilon > 0$ υπάρχει ένας αλγόριθμος με χρόνο εκτέλεσης $n^{O\left(\frac{1}{\varepsilon^2}\right)}$ που πετυχαίνει μια $(2+\varepsilon)$ προσέγγιση στο OP.

Απόδειξη: Το θεώρημα 2.5.2 δίνει μια δικριτιριακή προσέγγιση του Min-Excess που ισχύει για κάθε $\varepsilon > 0$.

Έστω ε' τέτοιο ώστε $2+\varepsilon = \frac{2}{1-\varepsilon'}$ (1) για κάθε $\varepsilon > 0$. Προφανώς θα πρέπει να ισχύει $\varepsilon' > 0$.⁵ Συνεπώς

εφαρμόζοντας το θεώρημα 2.5.2 για $\varepsilon' > 0$ συνεπάγεται ότι θα έχουμε ένα $\left(\frac{1}{1-\varepsilon'}, 2\right)$ -δικριτιριακό

προσεγγιστικό αλγόριθμο για το Min-Excess πρόβλημα ο οποίος τρέχει σε χρόνο $n^{O\left(\frac{1}{\varepsilon'^2}\right)}$. Βάσει του

Λήμματος 2.5.2 θα έχουμε μια $2 \cdot \frac{1}{1-\varepsilon'} = 2+\varepsilon$ προσέγγιση του Orienteering σε μη κατευθυνόμενα

γραφήματα. \square

2.6 Απόδειξη Θεωρήματος 2.5.1

Για την απόδειξη, ο Chekuri και συνεργάτες του [10] χρησιμοποιούν ένα θεώρημα που αποδεικνύεται στην [11] και αφορά το k-stroll.

Θεώρημα 2.6 [11]: Δεδομένου ενός γραφήματος G , δυο κορυφών s και t και ενός ακεραίου k , έστω L το μήκος ενός βέλτιστου μονοπατιού από το s στο t που καλύπτει k κορυφές. Για κάθε $\delta > 0$ υπάρχει ένας αλγόριθμος πολυωνυμικού χρόνου που βρίσκει ένα δέντρο⁶ μήκους το πολύ $(1+\delta) \cdot L$ και να περιέχει τουλάχιστον k κορυφές συμπεριλαμβανομένων και των s και t .

Ο αλγόριθμος του [11] 'μαντεύει' $O\left(\frac{1}{\delta}\right)$ κορυφές $s = w_1, w_2, w_3, \dots, w_{m-1}, w_m = t$ έτσι ώστε ένα βέλτιστο

μονοπάτι P (του k-stroll) επισκέπτεται αυτές τις κορυφές με αυτήν την σειρά και για κάθε i η απόσταση του w_i από το w_{i+1} στο μονοπάτι P είναι μικρότερη ή ίση από $\delta \cdot L$. Στη συνέχεια καλείται ο αλγόριθμος του [15] για το k-MST (Minimum Spanning Tree)⁷ με το σύνολο των (guessed) κορυφών που αναφέρουμε

⁵ Από τη σχέση (1) $\frac{2}{1-\varepsilon'} > 2$ γιατί $(2+\varepsilon) > 2$ καθώς $\varepsilon > 0 \Leftrightarrow \frac{1}{1-\varepsilon'} > 1 \Leftrightarrow 1-\varepsilon' < 1 \Leftrightarrow -\varepsilon' < 0 \Leftrightarrow \varepsilon' > 0$

⁶ Ως δέντρο ορίζεται ένα συνεκτικό γράφημα χωρίς κύκλους

⁷ Το k-MST ορίζεται ως το ελάχιστο ζευγνύον δέντρο το οποίο περιέχει k κορυφές. Δηλαδή δεδομένου ενός μη κατευθυνόμενου γραφήματος G και ενός ακεραίου k , το k-MST μας δίνει ένα δέντρο που περιέχει k κορυφές του G και έχει το ελάχιστο άθροισμα κόστους των ακμών που περιέχει

πιο πάνω. Προκύπτει ένα δέντρο που ικανοποιεί το θεώρημα 2.6. Αυτό το δέντρο περιέχει όλες τις (guessed) κορυφές. Μπορούμε να υποθέσουμε ότι όλες οι ακμές του δέντρου έχουν μήκος το πολύ $\delta \cdot L$.

Όσες ακμές έχουν μεγαλύτερο μήκος, μπορούν να υποδιαιρεθούν προσθέτοντας το πολύ $O\left(\frac{1}{\delta}\right)$

κορυφές.

Ο δικριτηριακός προσεγγιστικός αλγόριθμος για το k-stroll ξεκινά θέτοντας $\delta = \varepsilon^2$ και χρησιμοποιώντας τον αλγόριθμο του θεωρήματος 2.6 για να προκύψει ένα δέντρο k κορυφών T που περιέχει και τους κόμβους s και t . Από το θεώρημα εξασφαλίζεται ότι $\text{length}(T) \leq (1+\delta) \cdot L$ (1) (το L είναι το συντομότερο s - t μονοπάτι P που επισκέπτεται k κορυφές). Έστω $P_{s,t}^T$ να είναι το μονοπάτι στο T από το s στο t .

Μπορούμε να διπλασιάσουμε όλες τις ακμές που δεν ανήκουν στο $P_{s,t}^T$ για να προκύψει ένα μονοπάτι P_T από το s στο t που να περιέχει k κορυφές. Το μήκος του μονοπατιού P_T είναι $\text{length}(P_T) = 2 \cdot \text{length}(T) - \text{length}(P_{s,t}^T) \leq 2 \cdot \text{length}(T) - D$ (2) Εάν ισχύει κάποια από τις παρακάτω συνθήκες, τότε το μονοπάτι επισκέπτεται k κορυφές και έχει μήκος το πολύ $\max\{1.5D, 2L-D\}$ που είναι και το ζητούμενο αποτέλεσμα.

- Συνθήκη 1: Το ολικό μήκος του T είναι το πολύ $5D/4$ σε αυτήν την περίπτωση το μήκος του P_T είναι το πολύ $3/2 \cdot D$ λόγω της σχέσης (2).
- Συνθήκη 2: $\text{length}(P_{s,t}^T) \geq D + 2\delta L$ (σε αυτήν την περίπτωση το μήκος του P_T είναι

$$\text{length}(P_T) = 2 \cdot \text{length}(T) - \text{length}(P_{s,t}^T) \leq 2 \cdot (1+\delta) \cdot L - (D + 2\delta L) = 2L - D \text{ [από τις σχέσεις (1) και (2)]}$$

Θα αναφέρουμε αυτές τις συνθήκες easy doubling conditions. Στόχος μας είναι να δείξουμε ότι αν δεν ισχύει κάποια από τις easy doubling conditions, μπορεί να προκύψει από το T ένα νέο δέντρο T' που να περιέχει τα s και t μήκους το πολύ L και τουλάχιστον $(1 - O(\varepsilon)) \cdot k$ κορυφές. Στη συνέχεια διπλασιάζουμε τις ακμές του T' που δεν ανήκουν στο μονοπάτι s - t (του T') και προκύπτει ένα μονοπάτι μήκους το πολύ $2L - D$ που επισκέπτεται τουλάχιστον $(1 - O(\varepsilon)) \cdot k$ κορυφές.

Στη συνέχεια θα περιγράψουμε τη δομή που θα πρέπει να έχει το δέντρο T αν δεν ισχύει κάποια συνθήκη easy doubling και μετά πώς χρησιμοποιούμε αυτήν την πληροφορία για να προκύψει το T' .

Δομή του δέντρου T : Αν δεν ισχύει κάποια από τις easy doubling conditions, τότε

$\text{length}(T) > 5D/4 \Leftrightarrow D < 4 \cdot \text{length}(T)/5$. Επίσης $\text{length}(P_{s,t}^T) < D + 2\delta L$. Το ολικό μήκος των ακμών του T που δεν ανήκουν στο $P_{s,t}^T$ (συμβολίζουμε $T \setminus P_{s,t}^T$) είναι μεγαλύτερο από $(1/5 - 2\delta) \cdot L$. (Αυτό προκύπτει ως εξής:

$$\text{length}(T \setminus P_{s,t}^T) = \text{length}(T) - \text{length}(P_{s,t}^T) > \text{length}(T) - (D + 2\delta L) > \text{length}(T) - \left(\frac{4}{5} \cdot \text{length}(T) + 2\delta L\right) = \text{length}(T)/5 -$$

$$2\delta L > \left(\frac{1}{5} L - 2\delta L\right) = (1/5 - 2\delta) \cdot L \text{ (η τελευταία ανισότητα προκύπτει γιατί } \text{length}(T) \geq L \text{)}^8 \text{. Θα περιγράψουμε το$$

πώς κατασκευάζουμε το επιθυμητό δέντρο T' αφαιρώντας ένα μικρό κομμάτι του $T \setminus P_{s,t}^T$.

Θα λέμε ότι ένα σύνολο ακμών S που ανήκουν στο $T \setminus P_{s,t}^T$ είναι απομονωμένη συνιστώσα (isolated component) αν το ολικό μήκος των ακμών του S είναι λιγότερο από $\varepsilon \cdot L$ και το S είναι συνεκτική συνιστώσα του $T \setminus P_{s,t}^T$.

Πρόταση 2.6.1: Μπορούμε να αποσυνθέσουμε το σύνολο ακμών του $T \setminus P_{s,t}^T$ σε $\Omega(1/\varepsilon)$ ξένα κομμάτια έτσι ώστε:

⁸ Αν ισχύει $\text{length}(T) < L$ τότε θα ήταν δυνατό να παραχθεί ένα s - t μονοπάτι μήκους το πολύ $2L - D$ με όφελος k με διπλασιασμό των ακμών του T που δεν ανήκουν στο μονοπάτι s - t του T

- Κάθε κομμάτι είναι ένα συνεκτικό υπογράφημα του $\mathcal{T} \setminus P_{s,t}^T$ ή ένωση απομονωμένων συνιστωσών (isolated components) του $\mathcal{T} \setminus P_{s,t}^T$.
- Κάθε κομμάτι έχει μήκος που ανήκει στο διάστημα $[\varepsilon L, 3\varepsilon L]$ εκτός αν είναι ένωση όλων των απομονωμένων συνιστωσών του $\mathcal{T} \setminus P_{s,t}^T$ και έχει μήκος λιγότερο από εL .

Απόδειξη πρότασης 2.6.1 : Θεωρούμε τον παρακάτω άπληστο αλγόριθμο: θέτουμε ως ρίζα του δέντρου \mathcal{T} τον κόμβο s και θεωρούμε τον βαθύτερο κόμβο v στο $\mathcal{T} \setminus P_{s,t}^T$ για τον οποίο το ολικό μήκος ακμών του υποδέντρου με ρίζα τον v είναι τουλάχιστον εL . Αν το συνολικό μήκος όλων των ακμών του υποδέντρου είναι το πολύ $2\varepsilon L$ τότε το υποδέντρο είναι ένα κομμάτι που έχει το επιθυμητό μέγεθος (και είναι συνεκτικό). Διαφορετικά επιλέγουμε αρκετά ‘παιδιά’ του v έτσι ώστε το συνολικό μέγεθος όλων των υποδέντρων τους μαζί με τις ακμές που τα συνδέουν με το v , να είναι μεταξύ εL και $2\varepsilon L$. (Αυτό είναι πάντα εφικτό γιατί το υποδέντρο με ρίζα κάθε ‘παιδί’ του v έχει μήκος μικρότερο του εL και κάθε ακμή έχει μήκος μικρότερο ή ίσο του $\delta L < \varepsilon L$ (γιατί επιλέξαμε $\delta = \varepsilon^2$)). Αυτό σημαίνει ότι παίρνουμε πάλι ένα κομμάτι που είναι συνεκτικό και έχει το επιθυμητό μέγεθος.

Στη συνέχεια διαγράφουμε τις ακμές του κομματιού που βρήκαμε από το δέντρο \mathcal{T} και βρίσκουμε ένα άλλο κομμάτι επαναληπτικά. Όταν δε θα μπορούμε να βρούμε άλλα τέτοια κομμάτια είναι πιθανό να έχουμε μείνει με τμήματα μήκους $< \varepsilon L$ τα οποία κρέμονται από το s - t μονοπάτι. Για κάθε τέτοιο τμήμα που έχει ένα επιπλέον κομμάτι να κρέμεται από αυτό, το ενώνουμε με αυτό το κομμάτι αυξάνοντας το μήκος σε λιγότερο από $3\varepsilon L$. Τα κομμάτια που μένουν είναι απομονωμένα συστατικά και αν το ολικό τους μέγεθος δεν είναι μικρότερο από εL , είναι εύκολο να τα συνδυάσουμε σε ομάδες με ολικό μήκος στο διάστημα $[\varepsilon L, 3\varepsilon L]$. \square

Έστω T_1 το δέντρο που προκύπτει ως εξής: Θέτουμε s' την κορυφή που αντιστοιχεί στο $P_{s,t}^T$ και θέτουμε μια κορυφή για κάθε κομμάτι της πρότασης 2.6.1. Συνεπώς θα έχουμε $\Omega(1/\varepsilon)$ κορυφές. Υπάρχει μια ακμή μεταξύ των κορυφών $v_1, v_2 \in V(T_1)$ που αντιστοιχούν στα σύνολα ακμών S_1, S_2 αν και μόνο αν το S_1 περιέχει την πατρική ακμή στο \mathcal{T} μιας ελαχίστου βάθους ακμής που ανήκει στο S_2 , ή αντίστροφα. (Στην ειδική περίπτωση που $v_1 = s'$ και η ακμή ελάχιστου βάθους στο S_2 είναι ενωμένη με την κορυφή s του δέντρου \mathcal{T} , προσθέτουμε την ακμή μεταξύ των $v_1 = s'$ και v_2). Παρατηρούμε ότι κάθε κομμάτι που περιέχει απομονωμένες συνιστώσες (isolated components) γίνεται φύλλο στο T_1 γειτονικό του s' .

Πρόταση 2.6.2 Το δέντρο T_1 περιέχει μια κορυφή βαθμού 1 ή 2 που αντιστοιχεί σε ένα κομμάτι με μήκος στο $[\varepsilon L, 3\varepsilon L]$ και περιέχει το πολύ $32/\varepsilon k$ κορυφές του αρχικού δέντρου \mathcal{T} που δεν περιέχονται σε άλλα κομμάτια.

Απόδειξη 2.6.2: Ο αριθμός των κορυφών του T_1 είναι τουλάχιστον $\frac{(1/5 - 2\delta)L}{3\varepsilon L} = \frac{1}{15\varepsilon} - \frac{2\varepsilon}{3} \geq \frac{1}{16\varepsilon}$ (Θυμίζουμε ότι η ποσότητα $(1/5 - 2\delta)L$ είναι ένα κάτω φράγμα του συνολικού μήκους ακμών του $\mathcal{T} \setminus P_{s,t}^T$. Το μέγιστο μήκος κάθε κομματιού είναι $3\varepsilon L$. Έτσι προκύπτει το αρχικό κλάσμα). Τουλάχιστον περισσότερες από τις μισές αυτές κορυφές έχουν βαθμό 1 ή 2 καθώς το T_1 είναι δέντρο. Αυτό αποδεικνύεται με εις άτοπο απαγωγή. Έστω x ο αριθμός των κορυφών του δέντρου με βαθμό μεγαλύτερο του 2 και N ο αριθμός των κορυφών του δέντρου. Υποθέτουμε ότι $x \geq \frac{N}{2}$ και επιλέγουμε μια κορυφή που έχει τουλάχιστον βαθμό 3 ως ρίζα του δέντρου. Η ρίζα με τους απογόνους της περιλαμβάνει τουλάχιστον 4 κορυφές. Κάθε απόγονος βαθμού μεγαλύτερου του 2 έχει τουλάχιστον 2 απογόνους γιατί έχει βαθμό

τουλάχιστον 3. Πρέπει όμως να ισχύει $4+(x-1) \cdot 2 \leq N \Leftrightarrow 2 \cdot x + 2 \leq N \Leftrightarrow x \leq (N-2)/2$ που είναι άτοπο. Άρα δεν ισχύει η υπόθεση $x \geq \frac{N}{2}$ και αποδεικνύεται ο ισχυρισμός.

Αν η ένωση όλων των isolated components έχει μήκος λιγότερο από ϵL , αφαιρούμε την κορυφή που αντιστοιχεί σε αυτό το κομμάτι. Υπάρχουν τουλάχιστον $1/(32\epsilon)$ (τουλάχιστον οι μισές από τις $\frac{1}{16\epsilon}$ που

είναι συνολικά) από κορυφές βαθμού 1 ή 2. Αν καθεμία από αυτές αντιστοιχεί σε ένα κομμάτι με περισσότερες από $32\epsilon k$ κορυφές που δεν ανήκουν και σε άλλα κομμάτια, τότε ο συνολικός αριθμός των κορυφών που περιέχουν ξεπερνά τον αριθμό k , που είναι άτοπο. \square

Εάν το T_1 έχει ένα φύλλο που αντιστοιχεί σε ένα κομμάτι με το πολύ $32\epsilon k$ κορυφές, διαγράφουμε αυτό το κομμάτι από το T και παίρνουμε ένα δέντρο T' μήκους το πολύ $(1+\delta)L - \epsilon L < L$ με τουλάχιστον $(1-32\epsilon)k$ κορυφές (Όπως προαναφέρθηκε ισχύει $\text{length}(T) \leq (1+\delta)L$ και το κομμάτι που αφαιρέσαμε έχει μήκος στο $[\epsilon L, 3\epsilon L)$). Διπλασιάζουμε τις ακμές του T' που δεν ανήκουν στο s - t μονοπάτι που περιέχει, και έτσι προκύπτει μια διαδρομή από το s στο t που επισκέπτεται $(1-32\epsilon)k$ κορυφές και έχει μήκος το πολύ $2L-D$. Αυτό θέλαμε να πετύχουμε.

Αν δεν υπάρχει τέτοιο φύλλο μπορούμε να βρούμε μια κορυφή βαθμού 2 (βαθμός μιας κορυφής σε ένα undirected γράφημα ονομάζεται ο αριθμός των ακμών που συνδέονται με αυτήν) στο T_1 που αντιστοιχεί σε ένα συνεκτικό υπογράφημα/κομμάτι C του $T|_{P_{s,t}^T}$ μήκους $l \in [\epsilon L, 3\epsilon L)$ και το πολύ $32\epsilon k$ κορυφών.

Αφαιρώντας το C από το T θα προκύψουν δύο δέντρα T_{11} και T_2 . Έστω T_{11} το δέντρο που περιέχει τα s και t . Μπορούμε να επανασυνδέσουμε τα δέντρα χρησιμοποιώντας το συντομότερο μονοπάτι μεταξύ τους. Εάν το μήκος αυτού του μονοπατιού έχει μήκος το πολύ $l - \delta L$, έχουμε ένα δέντρο T' μήκους το πολύ L^9 που περιέχει τουλάχιστον $k - 32\epsilon k = (1-32\epsilon)k$ κορυφές. Αυτό μας αρκεί.

Ας υποθέσουμε τώρα ότι το συντομότερο μονοπάτι του G που συνδέει τα δέντρα T_{11} και T_2 έχει μήκος μεγαλύτερο από $l - \delta L$. (Το συνολικό μήκος του C είναι l). Μια συνέπεια είναι ότι το κομμάτι C είναι path-like. Δηλαδή, αν τα x και y είναι δυο κορυφές που ανήκουν στο $T-C$ με ακμές που καταλήγουν στο C , το μήκος του μονοπατιού από το x στο y μέσα στο κομμάτι C είναι περισσότερο από $l - \delta L$. Θα αναφερόμαστε σε αυτό το μονοπάτι από το x στο y ως spine του κομματιού. Συνεπάγεται ότι το ολικό μήκος των ακμών του C που δεν ανήκουν στο spine είναι μικρότερο από δL . Θα αναφερόμαστε στην κορυφή $x \in T_{11}$ που είναι γειτονική στο C ως head (κεφαλή) του spine, και στην κορυφή $y \in T_2$ και γειτονική του C ως foot του spine. Τέλος θα λέμε για οποιοδήποτε κορυφές p και q που ανήκουν στο C ότι η απόσταση κατά μήκος του spine μεταξύ των p και q είναι το μήκος εκείνων των ακμών στο p - q μονοπάτι που ανήκουν στο spine.

Υποθέτουμε προσωρινά ότι το T_2 περιέχει τουλάχιστον μια κορυφή από τις guessed κορυφές του αλγορίθμου του θεωρήματος 2.6. Έστω w_p η μεγαλύτερη κατά σειρά κορυφή από αυτές που μαντέψαμε στον αλγόριθμο του 2.6 και ανήκει στο T_2 . Αν λοιπόν η επόμενη guessed κορυφή w_{p+1} δεν ανήκει στο T_2 , δεν ανήκει ούτε στο T_{11} γιατί το συντομότερο μονοπάτι που συνδέει τα T_{11} και T_2 έχει μήκος τουλάχιστον $l - \delta L$ ενώ η ακμή που συνδέει τα w_p και w_{p+1} έχει μήκος μικρότερο ή ίσο από δL . Άρα η w_{p+1} πρέπει να ανήκει στο C . Ομοίως, αφού $\delta L \ll l - \delta L$ πρέπει και οι επόμενες κορυφές

⁹ Αυτό ισχύει γιατί το μήκος του T' θα είναι το πολύ όσο το μήκος του T χωρίς το κομμάτι C (που είναι l) συν το μήκος $l + \delta L$ που ενώνει τα δυο υποδέντρα που προκύπτουν από την αφαίρεση. Άρα $\text{length}(T') \leq \text{length}(T) - l + l + \delta L \leq (1+\delta)L - \delta L = L$

w_{p+2}, w_{p+3}, \dots να ανήκουν στο C. (Για την ακρίβεια υπάρχουν τουλάχιστον $\frac{l - \delta L}{\delta L} = \Omega(1/\varepsilon)$ διαδοχικές guessed κορυφές στο C). Έστω w_q η μεγαλύτερη κατά σειρά κορυφή από τις διαδοχικές guessed κορυφές που βρίσκονται στο C.

Με ένα παρόμοιο επιχείρημα μπορούμε να πούμε ότι αν w_b είναι η μικρότερη κατά σειρά κορυφή που ανήκει στο T_2 τότε οι κορυφές w_{b-1}, w_{b-2}, \dots βρίσκονται στο C. Έστω w_a η μικρότερη κατά σειρά από τις διαδοχικές κορυφές έτσι ώστε $w_a, w_{a+1}, \dots, w_{b-2}, w_{b-1}$ να βρίσκονται στο κομμάτι C.

Πρέπει να σημειωθεί ότι αν το δέντρο T_2 δεν περιέχει κάποια από τις guessed κορυφές μπορούμε να τροποποιήσουμε την παραπάνω διαδικασία βρίσκοντας την guessed κορυφή w η οποία βρίσκεται πιο κοντά στο foot του spine. Αφαιρούμε από το C το μονοπάτι από το w στο foot καθώς και τα τμήματα εκείνα του spine που είναι γειτονικά στο συγκεκριμένο μονοπάτι. Προσθέτουμε αυτές τις ακμές στο δέντρο T_2 . Τώρα το T_2 περιέχει μια από τις guessed κορυφές και μπορούμε να συνεχίσουμε.

Χωρίζουμε τώρα το C σε τμήματα ως εξής: αρχίζοντας από το x , την κεφαλή του spine, κόβουμε το C σε απόσταση $10\delta L$ κατά μήκος του spine από το x . Επαναλαμβάνουμε αυτήν την διαδικασία μέχρι το foot

του spine. Έτσι παίρνουμε τουλάχιστον $\frac{l - \delta L}{10\delta L} \geq \frac{1}{10\varepsilon} - \frac{1}{10}$ τμήματα. (η ανισότητα προκύπτει γιατί

$$\frac{l - \delta L}{10\delta L} = \frac{l}{10\delta L} - \frac{1}{10} \geq \frac{\varepsilon L}{10\varepsilon^2 L} - \frac{1}{10} \text{ καθώς } l \in [\varepsilon L, 3\varepsilon L])$$

Αφαιρούμε το τμήμα που βρίσκεται πιο κοντά στην κεφαλή x και τα δυο τμήματα πιο κοντά στο y (foot) και απαριθμούμε τα τμήματα που έμειναν από το 1 έως το r ξεκινώντας από την κεφαλή. Έχουμε τουλάχιστον $\frac{1}{10\varepsilon} - \frac{1}{10} - 3 \geq \frac{1}{15\varepsilon}$ τμήματα. Για κάθε τμήμα θα αναφερόμαστε στο άκρο που βρίσκεται πιο κοντά στο x (head of the spine) με τον όρο κορυφή (top) του τμήματος και στο άκρο που είναι πιο κοντά στο y (foot) με τον όρο bottom (κάτω μέρος) του τμήματος.

Εστιάζουμε την προσοχή μας στις guessed κορυφές στο εύρος από το w_a έως το w_{b-1} και w_{p+1} μέχρι w_q . Για κάθε τμήμα i ορίζουμε ως v_i^{low} να είναι η μικρότερη αριθμημένη guessed κορυφή στα τμήματα από το r στο i και v_i^{high} η μεγαλύτερη αριθμημένη guessed κορυφή από το r στο i .

Λήμμα 2.6: Για κάθε i

1. Η v_i^{low} βρίσκεται πριν από την v_{i+1}^{low} στο βέλτιστο μονοπάτι, επίσης η v_i^{high} βρίσκεται μετά από την v_{i+1}^{high} στο βέλτιστο μονοπάτι.
2. Η απόσταση κατά μήκος του spine από την κορυφή του τμήματος i στην κορυφή v_i^{low} είτε στην κορυφή v_i^{high} είναι το πολύ $2 \cdot \delta \cdot L$.
3. Η απόσταση μεταξύ των κορυφών v_i^{low} και v_{i+1}^{low} είναι τουλάχιστον $7 \cdot \delta \cdot L$. Ομοίως και για τις κορυφές v_i^{high} και v_{i+1}^{high} .

Απόδειξη Λήμματος 2.6: Η απόδειξη εστιάζει στους ισχυρισμούς για v_i^{low} και v_{i+1}^{low} . Οι ισχυρισμοί για τα v_i^{high} και v_{i+1}^{high} είναι παρόμοιοι. Αρχικά ισχύει ο ισχυρισμός ότι κάθε τμήμα (του C) περιέχει κάποια guessed κορυφή μεταξύ w_a και w_{b-1} . Έστω ότι υπάρχει ένα τμήμα i που δεν περιέχει. Έστω c ο δείκτης της guessed κορυφής με $c \geq a$ έτσι ώστε η w_c να μη βρίσκεται πάνω από το τμήμα i στο δέντρο.

(Πάντα υπάρχει μια τέτοια κορυφή αφού η w_a είναι πάνω από το τμήμα i και η w_b κάτω από αυτό.) Άρα η w_{c-1} βρίσκεται πάνω από το τμήμα i και η w_c από κάτω. Μπορεί να διαγραφεί το τμήμα i και να συνδεθεί το δέντρο με μια ακμή που συνδέει τις κορυφές w_{c-1} και w_c μήκους το πολύ δL . Αυτή η ένωση όμως συνεπάγεται ένα μονοπάτι από το x στο y μήκους το πολύ $l - 10\delta L + \delta L = l - 9\delta L$ που είναι αντίφαση γιατί το $l - 9\delta L$ είναι μικρότερο από $l - \delta L$ και έχουμε υποθέσει ότι το μήκος από το x στο y είναι μεγαλύτερο από $l - \delta L$.

Έστω τώρα v_i^{low} η guessed κορυφή w_j . Θα αποδειχθεί ότι βρίσκεται στο τμήμα i . Η θέση της w_{j-1} δε βρίσκεται στα τμήματα από i έως r εξ ορισμού. Άρα θα πρέπει να βρίσκεται στα τμήματα από 1 έως $i-1$. Εάν η w_j δε βρισκόταν στο διάστημα i , θα ήταν δυνατό να αφαιρεθεί το τμήμα i (μειώνοντας το μήκος κατά $10\delta L$) και να ενωθούν τα x και y μέσω της ακμής μεταξύ των w_j και w_{j-1} που έχει μήκος το πολύ δL . Έτσι προκύπτει ένα μονοπάτι από το x στο y που είναι μικρότερο κατά $9\delta L$ τουλάχιστον. Αυτό είναι άτοπο. Συνεπώς πρέπει για κάθε i , η κορυφή v_i^{low} να βρίσκεται στο τμήμα i .

Επειδή η guessed κορυφή με χαμηλότερο δείκτη στα τμήματα από το i στο r (v_i^{low}) βρίσκεται στο τμήμα i , έχει μικρότερο δείκτη από τη χαμηλότερου δείκτη κορυφή στα τμήματα από $i+1$ έως r . Αυτό σημαίνει ότι η v_i^{low} βρίσκεται πριν από την v_{i+1}^{low} στο βέλτιστο μονοπάτι. Έτσι αποδεικνύεται ο πρώτος ισχυρισμός του λήμματος 2.6.

Θα αποδειχθεί ότι για κάθε i η απόσταση κατά μήκος του spine από την κορυφή v_i^{low} στην κορυφή (top) του τμήματος i είναι το πολύ $2\delta L$. Έστω ότι αυτό δεν ισχύει. Τότε διαγράφονται οι ακμές κατά μήκος του spine από την v_i^{low} μέχρι την κορυφή top του τμήματος i και ενώνεται η v_i^{low} με την προηγούμενη guessed κορυφή που βρίσκεται στο τμήμα $i-1$. Αυτό θα είχε ως αποτέλεσμα η διαγραφή να μειώσει το μήκος κατά $2\delta L$ τουλάχιστον και η καινούρια ακμή που προστέθηκε αυξάνει το μήκος κατά δL το πολύ. Άρα προκύπτει μια μείωση του μήκους τουλάχιστον δL που είναι άτοπο.

Τώρα μπορεί να αποδειχθεί και ο τρίτος ισχυρισμός του λήμματος 2.6. Διαγράφοντας τις ακμές από την κορυφή v_i^{low} μέχρι το κάτω μέρος του τμήματος (bottom) το μήκος μειώνεται κατά $8\delta L$ τουλάχιστον βάσει του ισχυρισμού 2 του λήμματος. Αν η απόσταση από το v_i^{low} στο v_{i+1}^{low} ήταν μικρότερη από $7\delta L$ θα προέκυπτε μείωση του μήκους τουλάχιστον δL που είναι άτοπο. \square

Τώρα για κάθε τμήμα i ορίζουμε ως $gain(i)$ το άθροισμα του οφέλους που συλλέγει το βέλτιστο μονοπάτι μεταξύ των v_i^{low} και v_{i+1}^{low} και του οφέλους που συλλέγει το βέλτιστο μονοπάτι μεταξύ των v_{i+1}^{high} και v_i^{high} .

Αφού αυτά τα μέρη του μονοπατιού είναι ξένα, $\sum_i gain(i) \leq k$ και υπάρχουν τουλάχιστον $\frac{1}{15\epsilon}$ τέτοια τμήματα, πρέπει να υπάρχει κάποιο i τέτοιο ώστε $gain(i) \leq 15\epsilon k$. Μπορούμε να βρούμε ένα τέτοιο i απαριθμώντας όλες τις περιπτώσεις.

Συγχωνεύοντας το γράφημα G:

Έστω ότι βρίσκουμε ένα τμήμα i για το οποίο ισχύει $gain(i) \leq 15\epsilon k$. Θεωρούμε το νέο γράφημα H που προκύπτει από το G αν συγχωνεύσουμε τις 4 κορυφές v_i^{low} , v_i^{high} , v_{i+1}^{low} και v_{i+1}^{high} του G σε μια νέα κορυφή v' .

Πρόταση 2.6.3: Το γράφημα H έχει ένα μονοπάτι μήκους το πολύ $L - 14\delta L$ που επισκέπτεται τουλάχιστον $(1 - 15\epsilon)k$ κορυφές.

Απόδειξη: Θεωρούμε το βέλτιστο μονοπάτι P στο G και το τροποποιούμε έτσι ώστε να βρούμε ένα μονοπάτι P_H στο H αφαιρώντας το μέρος του μονοπατιού μεταξύ των v_i^{low} και v_{i+1}^{low} και το μέρος του μονοπατιού μεταξύ των v_{i+1}^{high} και v_i^{high} . Αφού $gain(i) \leq 15\epsilon k$ το μονοπάτι P_H επισκέπτεται τουλάχιστον $(1-15\epsilon)k$ κορυφές. Επιπλέον αφού η απόσταση του συντομότερου μονοπατιού από v_i^{low} στη v_{i+1}^{low} καθώς και από v_i^{high} σε v_{i+1}^{high} είναι τουλάχιστον $7\delta L$ σε κάθε περίπτωση, το P_H έχει μήκος το πολύ $L-14\delta L$. \square

Χρησιμοποιώντας τον αλγόριθμο του [15] μπορούμε να βρούμε ένα δέντρο T_H στο H μήκους το πολύ $L-13\delta L$ με τουλάχιστον $(1-15\epsilon)k$. Το T_H μπορεί να μην αντιστοιχεί σε δέντρο στο G (αν χρησιμοποιεί την νέα κορυφή v'). Ωστόσο μπορούμε να βρούμε ένα δέντρο T_i στο G , μήκους το πολύ $13\delta L$ που περιέχει και τις $v_i^{low}, v_i^{high}, v_{i+1}^{low}, v_{i+1}^{high}$. Μπορούμε στη συνέχεια να συνδυάσουμε τα T_H και T_i και να προκύψει ένα δέντρο T' στο G μήκους L .

Πρόταση 2.6.4: Υπάρχει ένα δέντρο T_i στο G που περιέχει τις κορυφές $v_i^{low}, v_i^{high}, v_{i+1}^{low}, v_{i+1}^{high}$ μήκους το πολύ $13\delta L$.

Απόδειξη πρότασης 2.6.4: Χρησιμοποιούμε όλο το τμήμα i και ένα μέρος του τμήματος $i+1$ για να φτάσουμε στις κορυφές v_{i+1}^{low} και v_{i+1}^{high} . Οι ακμές του τμήματος i κατά μήκος του spine έχουν μήκος το πολύ $10\delta L$. Οι v_{i+1}^{low} και v_{i+1}^{high} έχουν απόσταση (η καθεμία) κατά μήκος του spine το πολύ $2\delta L$ από την κορυφή top του τμήματος $i+1$ (Λόγω του λήμματος 2.6). Το συνολικό μήκος όλων των ακμών του C που δεν ανήκουν στο spine είναι το πολύ δL . Συνεπώς για να συνδεθούν οι 4 κορυφές $v_i^{low}, v_i^{high}, v_{i+1}^{low}, v_{i+1}^{high}$ χρησιμοποιούνται ακμές συνολικού μήκους το πολύ $(10+2+1)\delta L=13\delta L$. \square

Απόδειξη του θεωρήματος 2.5.1: Έστω $\epsilon' = \frac{\epsilon}{32}$ και εφαρμόζουμε τον αλγόριθμο του [15] με

$\delta = \epsilon'^2$ για να προκύψει ένα δέντρο T , k κορυφών με μήκος το πολύ $(1+\delta)L$. Αν ισχύει κάποια από τις συνθήκες easy doubling, μπορούμε να διπλασιάσουμε όλες τις ακμές του T που δεν ανήκουν στο $s-t$ μονοπάτι για να προκύψει μια νέα $s-t$ διαδρομή που διατρέχει k κορυφές μήκους το πολύ $\max\{1.5D, 2L-D\}$.

Εάν δεν ισχύει κάποια από τις συνθήκες easy doubling, χρησιμοποιούμε το T για να πετύχουμε ένα νέο δέντρο T' που περιέχει τις κορυφές s, t με μήκος το πολύ L και τουλάχιστον $(1-32\epsilon')$ - k κορυφές. Διπλασιάζοντας τις ακμές του T' που δεν ανήκουν στο $s-t$ μονοπάτι βρίσκουμε ένα νέο $s-t$ μονοπάτι που επισκέπτεται $(1-32\epsilon')$ - $k = (1-\epsilon)$ - k κορυφές και έχει μήκος το πολύ $2L-D$. \square

2.7 Προσεγγιστική πολυπλοκότητα του OP

Το Orienteering Problem ανήκει στην κατηγορία των NP-hard προβλημάτων ως γενίκευση του προβλήματος του περιοδεύοντος πωλητή. Τα προβλήματα αυτής της κατηγορίας δεν παρουσιάζουν ακριβή λύση σε πολυωνυμικό χρόνο. Σε αυτήν την παράγραφο θα δείξουμε ότι το OP ανήκει στην τάξη των APX-hard προβλημάτων. Τα APX είναι τα NP-hard προβλήματα βελτιστοποίησης που έχουν προσεγγιστικούς αλγόριθμους που τρέχουν σε πολυωνυμικό χρόνο με παράγοντα προσέγγισης (approximation ratio) που είναι φραγμένος από μια σταθερή τιμή.

Συγκεκριμένα θα δείξουμε ότι το να επιτύχουμε έναν αλγόριθμο με παράγοντα προσέγγισης καλύτερο από $1481/1480$ είναι NP-hard.

Θεώρημα [7]: Το OP είναι NP-hard να προσεγγιστεί με παράγοντα μικρότερο από $1481/1480$.

Απόδειξη: Θα ανάγουμε το TSP problem με μήκη ακμών στο σύνολο $\{1,2\}$ στο OP. Έστω πλήρες γράφημα $G=(V,E)$ με n κόμβους και μήκη πλευρών στο σύνολο $\{1,2\}$. Οι Engebretsen και Karpinski [22]

έδειξαν ότι είναι NP-hard να προσεγγιστεί το TSP με παράγοντα μικρότερο $1+\alpha = \frac{741}{740}$ σε αυτά τα γραφήματα.

Έστω $L=n+\delta n$ να είναι το μήκος της βέλτιστης λύσης του TSP (δοκιμάζουμε όλες τις τιμές μεταξύ n και $2n$). Υποθέτουμε ότι υπάρχει ένας αλγόριθμος που προσεγγίζει το OP με παράγοντα $(1+\beta)$, όπου $\beta \leq \frac{\alpha}{2} = \frac{1}{1480}$. Εφαρμόζουμε αυτόν τον αλγόριθμο στο G με όριο απόστασης (budget) το L . Παρατηρούμε ότι η

βέλτιστη λύση (που είναι το βέλτιστο TSP μονοπάτι) συλλέγει $n-1$ κόμβους εντός απόστασης L (όλους τους κόμβους εκτός του αρχικού s υποθέτοντας ότι το όφελος είναι 0 στον αρχικό κόμβο). Συνεπώς η λύση που παίρνουμε από τον αλγόριθμο που προσεγγίζει το OP με $1+\beta$, συλλέγει $\frac{1}{1+\beta} \cdot (n-1)$ κόμβους.

Μετατρέπουμε τη λύση αυτή σε έναν κύκλο (tour) ο οποίος περιέχει όλους τους κόμβους

χρησιμοποιώντας $(1 - \frac{1}{1+\beta})(n-1)+1$ ακμές μήκους το πολύ 2. Το μήκος της λύσης μας θα είναι το πολύ

$$L + 2 \cdot (1 - \frac{1}{1+\beta})(n-1) + 2 = L + \frac{2\beta}{1+\beta}(n-1) + 2 < L + 2\beta n \leq L + \alpha \cdot n \text{ (λόγω της υπόθεσης } \beta \leq \frac{\alpha}{2}) \leq L(1+\alpha).$$

Συνεπώς παίρνουμε μια προσέγγιση $(1+\alpha)$ για το TSP στο G , το οποίο έρχεται σε αντίθεση με το γεγονός ότι το να προσεγγίσουμε το TSP εντός του παράγοντα $(1+\alpha)$ είναι NP-hard όταν τα μήκη των ακμών

ανήκουν στο $\{1,2\}$. Άρα $\beta > \frac{1}{1480}$ που αποδεικνύει το ζητούμενο \square

3 Directed OP (OP με κατευθυνόμενο γράφημα)

Μία άλλη κατηγορία προβλημάτων είναι το directed Orienteering. Εδώ το γράφημα εισόδου είναι κατευθυνόμενο. Η μετρική συνάρτηση είναι ασύμμετρη. Το ζητούμενο είναι να βρούμε ένα μονοπάτι που να μεγιστοποιεί το όφελος επισκεψιμότητας χωρίς το κόστος να υπερβαίνει ένα άνω όριο B .

Και σε αυτήν την κατηγορία υπάρχουν διάφορες παραλλαγές π.χ. με ρίζα, χωρίς ρίζα, σημείο προς σημείο κλπ.

Όπως και παραπάνω θα ορίσουμε κάποια προβλήματα που θα χρησιμοποιήσουμε για να προσεγγίσουμε την κατευθυνόμενη εκδοχή.

3.1 Minimum Ratio ATSP (Asymmetric Traveling Salesman Problem)

Το πρόβλημα αναζητά μια κλειστή διαδρομή (tour) η οποία περιλαμβάνει τη ρίζα r έτσι ώστε να ελαχιστοποιείται ο λόγος του μήκους της διαδρομής προς τον αριθμό των κορυφών που περιέχει. Το πρόβλημα αυτό θα χρησιμεύσει στην δικριτηριακή προσέγγιση του directed k -path. Σύμφωνα με τον V.

Nagarajan [13] υπάρχει ένας $O(\frac{\log^2 n}{\log \log n})$ -προσεγγιστικός αλγόριθμος για το minimum ratio ATSP. Το

minimum ratio ATSP διαφέρει από το κλασικό ATSP στο ότι δεν περιλαμβάνει απαραίτητα όλες τις κορυφές του γραφήματος.

3.2 Directed k -Path Problem (ή k -stroll)

Δεδομένου ενός συνεκτικού κατευθυνόμενου γραφήματος $G=(V,E)$, ενός αρχικού κόμβου s και ενός τελικού t , και μιας τιμής k ζητάμε ένα s - t μονοπάτι με ελάχιστο μήκος (ή κόστος) το οποίο να επισκέπτεται τουλάχιστον k κορυφές (ή κόμβους) συμπεριλαμβανομένων και των κορυφών s, t .

3.3 Directed Min-Excess Problem

Δεδομένου ενός συνεκτικού κατευθυνόμενου γραφήματος $G=(V,E)$, ενός αρχικού κόμβου s και ενός τελικού t , και μιας τιμής k ζητάμε ένα s - t μονοπάτι με ελάχιστο excess το οποίο να επισκέπτεται τουλάχιστον k άλλες κορυφές.

3.3.1 Προσέγγιση του directed Min-excess προβλήματος

Η προσέγγιση είναι παρόμοια με αυτή για μη κατευθυνόμενο γράφημα, όπως παρουσιάστηκε στο 2.2. Η μόνη διαφορά βρίσκεται στο ότι φράζουμε το τοπικό excess.[10] Για κάθε διάστημα i , ορίζουμε ως τοπικό excess (συμβ. $ex(i)$) από έναν κόμβο u (απόστασης b_i από τον s) σε ένα κόμβο v (απόστασης b_{i+1} από τον s) την αύξηση του excess που παρατηρείται στο μονοπάτι P καθώς πηγαίνουμε από τον u στο v . Ισχύει $ex(i) = excess^P(v) - excess^P(u)$. Ορίζουμε ως l_i το μήκος του μονοπατιού στο i διάστημα και ως d_i το μήκος της συντομότερης διαδρομής από το u στο v τα οποία περιέχονται στο i διάστημα.

Λήμμα 2:[10] Για κάθε διάστημα i τύπου δύο ενός μονοπατιού P σε ένα κατευθυνόμενο γράφημα ισχύει $ex(i) \geq \max\{l_i - d_i, l_i/2\}$.

Απόδειξη: Έχουμε $ex(i) = excess^P(v) - excess^P(u) = (d^P(v) - d(v)) - (d^P(u) - d(u)) = (d^P(v) - d^P(u)) - (d(v) - d(u)) = l_i - (b_{i+1} - b_i)$. Όπως δείξαμε στο Λήμμα 1 [βλ. 21] για κάθε διάστημα i τύπου δύο ισχύει $l_i \geq 3 \cdot (b_{i+1} - b_i)$. (Για το τελευταίο διάστημα έχουμε $l_i \geq 3 \cdot (d(t) - b_i)$) Προφανώς η μικρότερη απόσταση d_i από τον πρώτο κόμβο του διαστήματος i μέχρι τον τελευταίο είναι τουλάχιστον $(b_{i+1} - b_i)$. Άρα προκύπτει $ex(i) \geq (l_i - d_i)$. Επίσης από το Λήμμα 1 $l_i \geq 3 \cdot (b_{i+1} - b_i) \geq 2 \cdot (b_{i+1} - b_i)$. Συμπεραίνουμε ότι $\frac{1}{2} \cdot l_i \geq (b_{i+1} - b_i) \Leftrightarrow -1/2 \cdot l_i \leq -(b_{i+1} - b_i) \Leftrightarrow l_i - 1/2 \cdot l_i \leq l_i - (b_{i+1} - b_i) \Leftrightarrow 1/2 \cdot l_i \leq ex(i) \Leftrightarrow ex(i) \geq 1/2 \cdot l_i$. \square

Η λογική του δυναμικού προγραμματισμού που περιγράφηκε στην ενότητα 2.1 ισχύει και στην κατευθυνόμενη εκδοχή. Για κάθε διάστημα που είναι πιθανό να ανήκει στη βέλτιστη λύση γίνεται guessing του επιθυμητού οφέλους καθώς και των ορίων του διαστήματος.

Λήμμα 3:[10] Σε κατευθυνόμενα γραφήματα μια β -προσέγγιση του k -stroll προβλήματος συνεπάγεται μια $(2\beta-1)$ -προσέγγιση του min-excess προβλήματος.

Απόδειξη: Στα διαστήματα τύπου ένα χρησιμοποιούμε δυναμικό προγραμματισμό και βρίσκουμε το βέλτιστο μονοπάτι. Στα διαστήματα τύπου δύο χρησιμοποιούμε έναν προσεγγιστικό αλγόριθμο για το k -stroll που βρίσκει ένα σύντομο μονοπάτι το οποίο συλλέγει το επιθυμητό όφελος. Έστω L το μήκος του βέλτιστου μονοπατιού, $L1$ το ολικό μήκος του P στα διαστήματα τύπου ένα και $L2$ το ολικό μήκος στα διαστήματα τύπου 2. Έστω P' το μονοπάτι που βρίσκουμε. Το συνολικό του μήκος θα είναι το πολύ $L1 + \beta \cdot L2$. Ο παράγοντας β δικαιολογείται επειδή χρησιμοποιούμε β -προσέγγιση του k -stroll. Το excess του βέλτιστου μονοπατιού θα είναι $excess(P) = L - d(t)$. Το excess του P' θα είναι $excess(P') = L1 + \beta \cdot L2 - d(t)$. Ισχύει ότι $L1 = L - L2$ γιατί στα διαστήματα τύπου ένα παίρνουμε ακριβείς λύσεις. Άρα $excess(P') = L -$

$L2 + \beta \cdot L2 - d(t) = L - d(t) + (\beta - 1) \cdot L2 = excess(P) + (\beta - 1) \cdot L2$. Έχουμε $excess(P) = \sum_i ex(i) \geq \sum_{i \in type2} ex(i) \geq \sum_{i \in type2} \frac{l_i}{2}$ (η

τελευταία ανισότητα προκύπτει από το Λήμμα 2). Αφού $\sum_{i \in type2} l_i = L2$ τελικά προκύπτει $excess(P) \geq 1/2 \cdot L2$

$\Leftrightarrow L2 \leq 2 \cdot excess(P)$. Αντικαθιστώντας λοιπόν στη σχέση 2 παίρνουμε $excess(P') = excess(P) + (\beta - 1) \cdot L2 \leq excess(P) + 2 \cdot (\beta - 1) \cdot excess(P) = (2\beta - 2 + 1) \cdot excess(P) = (2\beta - 1) \cdot excess(P)$.

Δείξαμε συνεπώς ότι το excess του μονοπατιού P' που βρίσκουμε προσεγγιστικά είναι το πολύ $(2\beta-1)$ φορές το $\text{excess}(P)$ όπου P το βέλτιστο μονοπάτι \square

3.4 Προσέγγιση του directed OP

Όπως αναφέρθηκε το Λήμμα 2.3 που αποδείχθηκε στην παράγραφο 2.3 ισχύει και για κατευθυνόμενα γραφήματα. Συνεπώς το directed OP ανάγεται στο directed Min-Excess. Δηλαδή μια γ -προσέγγιση του directed Min-Excess συνεπάγεται μια $\lceil \gamma \rceil$ -προσέγγιση για το directed OP.

Στο [10] χρησιμοποιείται μια δικριτηριακή προσέγγιση του k-stroll. Συγκεκριμένα αποδεικνύεται ότι μια (α, β) -προσέγγιση του k-stroll συνεπάγεται μια $(\alpha \cdot \lceil 2\beta - 1 \rceil)$ -προσέγγιση του Orienteering σε

κατευθυνόμενα γραφήματα. Η προσέγγιση που προκύπτει για το directed OP είναι $O(\log^2 \text{OPT})$ όπου OPT είναι ο αριθμός των κορυφών της βέλτιστης λύσης.

Στο [13] επιτυγχάνεται μια $O\left(\frac{\log^2 n}{\log \log n}\right)$ -προσέγγιση για το directed OP. Η διαδικασία είναι η εξής: με

βάση τον προσεγγιστικό αλγόριθμο για το minimum ratio ATSP προκύπτει ένας δικριτηριακός προσεγγιστικός αλγόριθμος για το directed k-path πρόβλημα και οι αναγωγές των Blum et al. [7] και Bansal et al. [12] (για τη μη κατευθυνόμενη εκδοχή) από το k-stroll πρόβλημα στο OP εφαρμόζονται στη συνέχεια και στην κατευθυνόμενη περίπτωση.

Ο αλγόριθμος για το κατευθυνόμενο Orienteering βασίζεται στις εξής ανάγωγές: το directed k-stroll στο minimum-ratio ATSP, το directed Min-Excess πρόβλημα στο directed k-stroll, και το directed OP στο directed Min-Excess. Οι δύο τελευταίες αναγωγές είναι παρόμοιες με τις αντίστοιχες αναγωγές για το undirected OP.

Θεώρημα: [1] Ένας p -προσεγγιστικός αλγόριθμος για το minimum ratio ATSP συνεπάγεται έναν $(3, 4 \cdot p)$ δικριτηριακό προσεγγιστικό αλγόριθμο για το directed k-path πρόβλημα.

Απόδειξη: Υποθέτουμε ότι γνωρίζουμε την βέλτιστη τιμή OPT του directed k-path. Ο επιθυμητός παράγοντας δικριτηριακής προσέγγισης είναι $(3, 4p)$. Εφαρμόζουμε δυαδική αναζήτηση στο υποτιθέμενο μήκος L της βέλτιστης λύσης προκειμένου να εντοπίσουμε το πραγματικό μήκος της βέλτιστης διαδρομής με ένα σταθερό παράγοντα. Συγκεκριμένα, αν ο αλγόριθμος επιστρέψει μια λύση της οποίας το κόστος

φράσσεται από $3 \cdot L$ και ο αριθμός των κόμβων που επισκέπτεται η λύση είναι τουλάχιστον $\frac{k}{4p}$ τότε

$\text{OPT} \leq L$. Αλλιώς αν δεν ικανοποιείται κάποια από τις δυο προηγούμενες συνθήκες, θα πρέπει να ισχύει $\text{OPT} > L$. Σε αυτήν την περίπτωση προσαρμόζουμε την τιμή του L ανάλογα για το επόμενο βήμα της δυαδικής αναζήτησης. Έτσι στο εξής υποθέτουμε ότι γνωρίζουμε το μήκος OPT της βέλτιστης λύσης.

Έστω G το κατευθυνόμενο γράφημα που αντιστοιχεί στη μετρική (V, d) (και έχει μια ακμή μήκους $d(u, v)$ από το u στο v για κάθε ζεύγος κορυφών $u, v \in V$). Επειδή αναφερόμαστε σε κατευθυνόμενα γραφήματα το (V, d) αποτελεί μη συμμετρικό μετρικό χώρο γιατί δεν ικανοποιεί το αξίωμα συμμετρίας. Τροποποιούμε το γράφημα G σε ένα γράφημα H ως εξής: α) διαγράφουμε όλες τις κορυφές v για τις οποίες ισχύει $d(s, v) > \text{OPT}$ ή $d(v, t) > \text{OPT}$ β) προσθέτουμε μια επιπλέον ακμή από το t στο s μήκους OPT. Η μετρική που προκύπτει από το γράφημα H θα συμβολίζεται (V, l) . Αφού υπάρχει ένα $s-t$ μονοπάτι μήκους OPT στη μετρική d που καλύπτει k κορυφές, όταν προσθέσουμε την ακμή (t, s) έχουμε έναν s -κύκλο σ^* μήκους το πολύ $2 \cdot \text{OPT}$ (στην l μετρική) που καλύπτει $k+1$ κορυφές.

Εφαρμόζουμε τον αλγόριθμο του minimum ratio ATSP με ρίζα s στην l μετρική, επαναληπτικά μέχρι είτε α) να καλυφθούν $k/2$ κορυφές και η ακμή (t, s) να μη χρησιμοποιείται στον τρέχοντα κύκλο στο γράφημα H , είτε β) η επιπλέον ακμή (t, s) χρησιμοποιείται για πρώτη φορά στον τρέχοντα κύκλο στο H . Έστω σ ο s -κύκλος (στο H) που προκύπτει στο τέλος αυτής της επανάληψης και h ο αριθμός των κορυφών που καλύπτονται. Παρατηρούμε ότι κάθε s -κύκλος που προστίθεται σε μια μόνο κλήση του minimum ratio ATSP μπορεί να χρησιμοποιεί την επιπλέον ακμή (t, s) το πολύ μια φορά. Άρα στην περίπτωση α) η ακμή (t, s) δεν βρίσκεται στο σ και στην περίπτωση β) η (t, s) χρησιμοποιείται ακριβώς μια φορά και είναι η τελευταία ακμή στο σ . Επίσης παρατηρούμε ότι σε κάθε κλήση του minimum ratio ATSP υπάρχει μια

εφικτή λύση με λόγο (ratio) $\frac{2 \cdot OPT}{k}$ (το σ^* περιορισμένο στις κορυφές που παρέμειναν). Έτσι ο λόγος του s -κύκλου σ θα είναι $\frac{l(\sigma)}{h} \leq p \cdot \frac{2 \cdot OPT}{k}$. Από το σ πετυχαίνουμε ένα s - t μονοπάτι τ στη μετρική d ως εξής: στην περίπτωση α) προσθέτουμε μια άμεση ακμή (s,t) : $\tau = \sigma \cup (s,t)$ β) στην περίπτωση αυτή διαγράφουμε το μοναδικό αντίγραφο της επιπλέον ακμής (t,s) : $\tau = \sigma \setminus \{(t,s)\}$. Σε κάθε περίπτωση, το μονοπάτι s - t δηλ. το τ περιέχει h κορυφές και έχει μήκος $d(\tau) \leq l(\sigma) + d(s,t) \leq h \cdot p \cdot \frac{2 \cdot OPT}{k} + OPT$.

Παρατηρούμε ότι στην περίπτωση α) $h \geq \frac{k}{2}$ και στη β) όπου χρησιμοποιείται η ακμή (t,s) ισχύει $\frac{OPT}{h} \leq \frac{l(\sigma)}{h} \leq p \cdot \frac{2 \cdot OPT}{k} \Leftrightarrow \frac{1}{h} \leq \frac{2 \cdot p}{k} \Leftrightarrow h \geq \frac{k}{2 \cdot p}$. Άρα

$$d(\tau) \leq ph \cdot \frac{2 \cdot OPT}{k} + OPT = \left(\frac{2 \cdot p \cdot h}{k} + 1 \right) \cdot OPT = \left(\frac{2ph + k}{k} \right) \cdot OPT \leq \frac{2ph + 2ph}{k} \cdot OPT \text{ (διότι}$$

$$h \geq \frac{k}{2 \cdot p} \Leftrightarrow k \leq 2 \cdot p \cdot h) \Leftrightarrow d(\tau) \leq \frac{4 \cdot p \cdot h}{k} \cdot OPT.$$

Στη συνέχεια διαμερίζουμε το τ σε μεγιστικά μονοπάτια καθένα από τα οποία έχει μήκος το πολύ OPT . Ο αριθμός των υπομονοπατιών είναι το πολύ $\frac{d(\tau)}{OPT} \leq \frac{4 \cdot p \cdot h}{k}$. Άρα ένα υπομονοπάτι έχει τουλάχιστον

$$\frac{h}{\lfloor \text{υπομονοπατια} \rfloor} \text{ κορυφές, δηλ. τουλάχιστον } \frac{h}{\frac{4ph}{k}} = \frac{k}{4p} \text{ κορυφές. Προσθέτοντας άμεσες ακμές από}$$

το s στην πρώτη κορυφή αυτού του μονοπατιού και από την τελευταία του κορυφή στο t , παίρνουμε ένα s - t μονοπάτι μήκους το πολύ $3 \cdot OPT$ που περιέχει τουλάχιστον $\frac{k}{4p}$ κορυφές. Οι άμεσες ακμές που

προσθέτουμε δεν μπορούν να υπερβαίνουν σε μήκος το OPT η καθεμία γιατί έχουν αφαιρεθεί στην αρχή όσες ακμές υπερβαίνουν το OPT . Έτσι προκύπτει η $(3,4p)$ δικριτηριακή προσέγγιση του directed k -path. □

Θεώρημα 3:[1] Ένας (α, β) δικριτηριακός προσεγγιστικός αλγόριθμος για το directed k -path πρόβλημα συνεπάγεται έναν $(2\alpha-1, \beta)$ δικριτηριακό προσεγγιστικό αλγόριθμο για το directed Min-Excess πρόβλημα.

Απόδειξη: [1] Ο αλγόριθμος αυτός είναι παρόμοιος με αυτόν του Blum [7] για μη κατευθυνόμενα γραφήματα. Αρχικά δείχνουμε ότι το βέλτιστο s - t μονοπάτι, έστω π , μπορεί να διαμεριστεί σε τμήματα που έχουν μια συγκεκριμένη δομή. Στη συνέχεια δείχνουμε τον τρόπο με τον οποίο προσεγγίζουμε λύσεις αυτής της δομής με χρήση δυναμικού προγραμματισμού. Υποθέτουμε ότι υπάρχει μια απαρίθμηση $s = v_1, v_2, \dots, v_n$ του συνόλου κορυφών V τέτοια ώστε $0 = d(s, v_1) < d(s, v_2) < d(s, v_3) < \dots < d(s, v_n)$. Για κάθε πραγματικό $l \geq 0$ ορίζουμε τη συνάρτηση $f(l)$ ως τον αριθμό των ακμών (x, y) στο βέλτιστο μονοπάτι π με $d(s, x) < l < d(s, y)$. Για τις τιμές του $l \geq d(s, t)$ για τις οποίες $f(l) = 1$ επαναορίζουμε την $f(l) = 2$. Προφανώς $f(l) \geq 1$ για κάθε l με $0 \leq l \leq d(s, t)$. Διαμερίζουμε τον άξονα των πραγματικών αριθμών σε διαστήματα δυο τύπων: τα διαστήματα τύπου ένα είναι μεγιστικά διαστήματα όπου $f(l) = 1$, τα διαστήματα τύπου δύο είναι μεγιστικά διαστήματα όπου $f(l) \geq 2$. Αριθμούμε τα όρια των διαστημάτων ως εξής: $0 = b_1 < b_2 < \dots < b_m$ και παρατηρούμε ότι $b_{m-1} < d(s, t) \leq b_m$. Το διάστημα i είναι το (b_i, b_{i+1}) και οι τύποι των διαστημάτων εναλλάσσονται μεταξύ 1 και 2. Συμβολίζουμε με $V_i = \{v \in V : b_i < d(s, v) \leq b_{i+1}\}$ να είναι οι κορυφές που βρίσκονται στο i διάστημα. Οι κορυφές κάποιου τυχαίου διαστήματος V_i αντιστοιχούν σε ένα συνεχές

υπομονοπάτι του π , έστω $S_i = \pi \cap V_i$. Το μήκος του π μπορεί να γραφτεί ως $d(\pi) = \sum_i l_i$ όπου $l_i = d(S_i)$

είναι το μήκος του π στο V_i . Κατά αντιστοιχία με το Blum [7] έχουμε το ακόλουθο λήμμα:

Λήμμα 3.3:[1] Εάν ένα διάστημα V_i είναι τύπου 1, τότε $l_i \geq b_{i+1} - b_i$. Αν το V_i είναι τύπου 2 και δεν περιέχει το t τότε $l_i \geq 2 \cdot (b_{i+1} - b_i)$. Αν το V_i είναι τύπου 2 και περιέχει το t τότε $l_i \geq 2 \cdot (d(s,t) - b_i)$. \square

Αν συμβολίσουμε με ε τη βέλτιστη τιμή του (directed) Min-Excess προβλήματος έχουμε

$$d(s,t) + \varepsilon = d(\pi) = \sum_i l_i. \text{ Ισχύει } d(s,t) = d(s,t) - b_{m-1} + b_{m-1} = d(s,t) - b_{m-1} + \sum_{i=1}^{m-2} (b_{i+1} - b_i) \text{ (γιατί}$$

$$b_{m-1} = \sum_{i=1}^{m-2} (b_{i+1} - b_i)) \Leftrightarrow \sum_{i \in \text{type1}} l_i + \frac{1}{2} \sum_{i \in \text{type2}} l_i \geq d(s,t) = d(\pi) - \varepsilon = \sum_{i \in \text{type1}} l_i + \sum_{i \in \text{type2}} l_i - \varepsilon$$

$$\Leftrightarrow \frac{1}{2} \sum_{i \in \text{type2}} l_i \leq \varepsilon \Leftrightarrow \sum_{i \in \text{type2}} l_i \leq 2 \cdot \varepsilon. \text{ (σχέση 1) (κατά αντιστοιχία με το Blum)}$$

Με βάση αυτή τη δομή της βέλτιστης λύσης γράφουμε ένα δυναμικό πρόγραμμα που ενώνει διαστήματα και μονοπάτια ώστε να προκύψει ένα s - t μονοπάτι. Τα υποπροβλήματα στο δυναμικό πρόγραμμα ορίζονται από πλειάδες (a, b, u, v, p, i) όπου τα $a, b \in V$ συμβολίζουν τα όρια του διαστήματος (το a είναι κοντά στο s και το b πιο μακριά), $u, v \in V$ είναι οι κορυφές έναρξης και τέλους του μονοπατιού στο συγκεκριμένο διάστημα, p ο αριθμός των κορυφών που καλύπτονται στο διάστημα και $i \in \{1, 2\}$ ο τύπος του διαστήματος.

Επίλυση των διαστημάτων τύπου 1: Μπορούμε να έχουμε μια ακριβή λύση σε πολυωνυμικό χρόνο με χρήση δυναμικού προγραμματισμού.

Επίλυση των διαστημάτων τύπου 2: Χρησιμοποιούμε τον αλγόριθμο του directed k -path για να βρούμε μια α -προσέγγιση ενός μονοπατιού u - v ελάχιστου μήκους που καλύπτει τουλάχιστον p/β κορυφές.

Συνδυασμός των διαστημάτων: Το δυναμικό πρόγραμμα για την επικόλληση των διαστημάτων θεωρεί τη διαμέριση του βέλτιστου μονοπατιού π όπως περιγράφηκε στην ενότητα 2.1. Το μήκος της λύσης που

$$\text{προκύπτει είναι το πολύ } \sum_{i \in \text{type1}} l_i + \sum_{i \in \text{type2}} a \cdot l_i =$$

$$= \sum_{i \in \text{type1}} l_i + \sum_{i \in \text{type2}} l_i - \sum_{i \in \text{type2}} l_i + \sum_{i \in \text{type2}} a \cdot l_i = \sum_i l_i + \sum_{i \in \text{type2}} (a-1) \cdot l_i = \sum_i l_i + (\alpha-1) \cdot \sum_{i \in \text{type2}} l_i = d(\pi) + (\alpha-1) \cdot \sum_{i \in \text{type2}} l_i \leq$$

$\leq d(\pi) + 2(\alpha-1) \cdot \varepsilon$ (λόγω της σχέσης 1) $= d(s,t) + \varepsilon + 2(\alpha-1) \cdot \varepsilon = d(s,t) + (2\alpha-1) \cdot \varepsilon$. Συνεπώς το excess της προσεγγιστικής μας λύσης είναι το πολύ $(2\alpha-1) \cdot \varepsilon$ όπου ε το excess του βέλτιστου μονοπατιού. Επιπλέον ο

αριθμός των κορυφών που καλύπτει το μονοπάτι είναι τουλάχιστον $\frac{k}{\beta}$. Έτσι προκύπτει μια $(2\alpha-1, \beta)$ -

προσέγγιση του (directed) Min-Excess προβλήματος. \square

Θεώρημα 4:[1] Ένας (α, β) δικριτηριακός προσεγγιστικός αλγόριθμος για το directed Min-Excess πρόβλημα συνεπάγεται έναν $\lceil \alpha \rceil \cdot \beta$ προσεγγιστικό αλγόριθμο για το directed Orienteering πρόβλημα.

Απόδειξη:[1] Διαμερίζουμε το βέλτιστο s - t μονοπάτι του OP, έστω π , σε $l = \lceil \alpha \rceil$ τμήματα, καθένα από τα οποία περιέχει τουλάχιστον k/l κορυφές (όπου k είναι η βέλτιστη τιμή του OP). Έστω ότι οι οριακές κορυφές αυτών τμημάτων να είναι $s = u_0, u_1, \dots, u_l = t$ με τη σειρά που εμφανίζονται στο π . Για δυο κορυφές u_i και u_j (με $i < j$) ορίζουμε το $\pi(u_i, u_j)$ ως το υπομονοπάτι του π από το u_i στο u_j . Ακόμα ορίζουμε $\varepsilon'(u_i, u_j) = d(\pi(u_i, u_j)) - d(u_i, u_j)$, δηλαδή το $\varepsilon'(u_i, u_j)$ συμβολίζει το τοπικό excess. Στην εργασία [12] έχει αποδειχθεί ότι η συνάρτηση ε' είναι υποπροσθετική. Αυτό ισχύει και για τα κατευθυνόμενα γραφήματα. Η

ιδιότητα της υποπροσθετικότητας είναι : για οποιαδήποτε $0 \leq h < i < j \leq l$ ισχύει ότι $\epsilon'(u_h, u_i) + \epsilon'(u_i, u_j) \leq \epsilon'(u_h, u_j)$. Από τα l κομμάτια επιλέγουμε το $\pi(u_i, u_{i+1})$ που έχει τη μικρότερη τιμή $\epsilon'(u_i, u_{i+1})$. Έστω σ το μονοπάτι από το s στο t με $\sigma = s, u_i, \pi(u_i, u_{i+1}), u_{i+1}, t$. Ισχύει $d(\pi) - d(\sigma) = \epsilon'(u_i, u_i) + \epsilon'(u_{i+1}, u_i) \geq (l-1) \cdot \epsilon'(u_i, u_{i+1})$ γιατί η ϵ' είναι υποπροσθετική και το τμήμα i του μονοπατιού έχει τη μικρότερη τιμή ϵ' . Ο αλγόριθμος “μαντεύει” τη βέλτιστη τιμή k και τις κορυφές u_i και u_{i+1} (για να μαντέψουμε αυτές τις τιμές δοκιμάζουμε όλους τους δυνατούς συνδυασμούς) και τρέχει τον αλγόριθμο του Min-Excess με έναρξη το u_i και τελική κορυφή το u_{i+1} και στόχο k/l κορυφές. Για τα κατάλληλα u_i και u_{i+1} το $\pi(u_i, u_{i+1})$ είναι μια εφικτή λύση και ο προσεγγιστικός Min-Excess αλγόριθμος βρίσκει ένα $u_i - u_{i+1}$ μονοπάτι το οποίο

καλύπτει τουλάχιστον $\frac{k}{\beta \cdot l}$ κορυφές με μήκος το πολύ $d(u_i, u_{i+1}) + \alpha \cdot \epsilon'(u_i, u_{i+1}) =$

$d(u_i, u_{i+1}) + \alpha \cdot \epsilon'(u_i, u_{i+1}) + \epsilon'(u_i, u_{i+1}) - \epsilon'(u_i, u_{i+1}) = d(\pi(u_i, u_{i+1})) + (\alpha - 1) \cdot \epsilon'(u_i, u_{i+1})$. Συνεπώς το $s-t$ μονοπάτι που προκύπτει ως αποτέλεσμα (επισυνάπτοντας τις ακμές (s, u_i) και (u_{i+1}, t)) καλύπτει

τουλάχιστον $\frac{k}{\beta \cdot l}$ κορυφές και έχει μήκος το πολύ $d(s, u_i) + d(\pi(u_i, u_{i+1})) + d(u_{i+1}, t) + (\alpha - 1) \cdot$

$\epsilon'(u_i, u_{i+1}) = d(\sigma) + (\alpha - 1) \cdot \epsilon'(u_i, u_{i+1}) \leq d(\sigma) + (l - 1) \cdot \epsilon'(u_i, u_{i+1})$ (αφού $l \geq \alpha$ εξ ορισμού) $\leq d(\pi) \leq D$ που είναι το

όριο μήκους. Άρα η λύση που βρήκαμε δεν υπερβαίνει το budget D και αφού καλύπτει $\frac{k}{\beta \cdot l}$ κορυφές

έχουμε μια $\beta \cdot l = \beta \cdot \lceil a \rceil$ -προσέγγιση του OP \square

Υπάρχει μια $O\left(\frac{\log^2 n}{\log \log n}\right)$ προσέγγιση του minimum ratio ATSP. Με βάση τα θεωρήματα 1,2,3 αυτής της

ενότητας (που χρησιμοποιούν δικριτηριακή προσέγγιση) πετυχαίνουμε μια $O\left(\frac{\log^2 n}{\log \log n}\right)$ προσέγγιση του directed OP.

4 Weighted Orienteering Problem

Μια παραλλαγή του Orienteering Problem είναι το weighted OP. Σε αυτήν την εκδοχή, η είσοδος αποτελείται από ένα γράφημα G με κάποιο κόστος στις ακμές, δύο κόμβους (κορυφές) s και t που ανήκουν στο γράφημα, ένα μη αρνητικό αριθμό B ο οποίος αποτελεί ένα άνω φράγμα στο κόστος (ή μήκος) της λύσης και μια συνάρτηση βάρους (weight function) w από το σύνολο των κορυφών στο \mathbb{R}^+ . Η συνάρτηση w αντιπροσωπεύει το όφελος που προσκομίζουμε από την επίσκεψη του εκάστοτε κόμβου. Στόχος μας είναι η εύρεση ενός $s-t$ μονοπατιού με μήκος B το πολύ που μεγιστοποιεί το όφελος από την επίσκεψη των κόμβων. Είναι δυνατόν να επισκεφτούμε μια κορυφή περισσότερες από μια φορές, όμως το όφελος προστίθεται μόνο μία φορά.

Το παρακάτω θεώρημα είναι το θεώρημα 2 της ενότητας 2.5 και η απόδειξη παρουσιάστηκε σε εκείνη την ενότητα.

Θεώρημα 4.1: Για κάθε σταθερό $\delta > 0$ υπάρχει ένας αλγόριθμος με χρόνο εκτέλεσης $O\left(n^{\frac{1}{\delta^2}}\right)$ που πετυχαίνει μία $(2+\delta)$ -προσέγγιση του Orienteering σε μη κατευθυνόμενα γραφήματα.

Όπως αποδείχθηκε στην εργασία [8], η βεβαρυμμένη εκδοχή του OP (weighted OP) μπορεί εύκολα να αναχθεί στην απλή εκδοχή. Αυτό γίνεται με χρήση scaling και rounding τεχνικών.

Λήμμα 4.1: Εάν υπάρχει μια β -προσέγγιση σε πολυωνυμικό χρόνο για το απλό Orienteering (χωρίς βάρη στους κόμβους) τότε υπάρχει ένας αλγόριθμος πολυωνυμικού χρόνου που επιστρέφει μια β -προσεγγιστική λύση σε στιγμιότυπα του weighted OP όπου όλα τα βάρη είναι ακέραιοι από 1 έως n^2 .

Απόδειξη: Δεδομένου ενός γραφήματος $G=(V,E)$ συμβολίζουμε με $w(v)$ το όφελος της κορυφής v με $v \in V$. Δημιουργούμε ένα νέο γράφημα G' αντικαθιστώντας κάθε κόμβο v του G με μία κλίκα από $w(v)$ αντίγραφα του v και θέτοντας μηδέν το μήκος των ακμών που συνδέουν τα αντίγραφα του v . Με τον όρο κλίκα εννοούμε ένα γράφημα στο οποίο κάθε κορυφή συνδέεται με όλες τις άλλες.

Για οποιεσδήποτε κορυφές s, t που ανήκουν στο G και οποιοδήποτε όριο B υπάρχει ένα s - t μονοπάτι στο G κόστους B και συνολικού βάρους (οφέλους) W αν και μόνο αν υπάρχει ένα s - t μονοπάτι στο G' κόστους B το οποίο επισκέπτεται W κορυφές. Αφού υποθέτουμε ότι όλα τα βάρη στο G κυμαίνονται από 1 έως n^2 τότε ο αριθμός των κορυφών στο G' θα είναι το πολύ n^3 . Συνεπώς μπορεί να εφαρμοστεί μια β -προσέγγιση σε πολυωνυμικό χρόνο για το Orienteering στο γράφημα G' , για να προκύψει μια β -προσέγγιση στο weighted OP του G . Μία β -προσέγγιση του OP σε πολυωνυμικό χρόνο είναι εφικτή βάσει του Θεωρήματος 4.1 \square

Λήμμα 4.2: Εάν υπάρχει μια ρ -προσέγγιση για το Orienteering σε πολυωνυμικό χρόνο, τότε υπάρχει μια $\rho(1+\epsilon(1))$ - προσέγγιση πολυωνυμικού χρόνου για το βεβαρυμένο OP (weighted OP).

Απόδειξη: Υποθέτουμε χωρίς βλάβη της γενικότητας ότι το γράφημα εισόδου είναι πλήρες και τα μήκη των πλευρών ικανοποιούν την τριγωνική ανισότητα. Για κάποιο μονοπάτι P χρησιμοποιούμε το συμβολισμό $w(P)$ όταν αναφερόμαστε στο συνολικό όφελος από τις ακμές που επισκέπτεται το P . Με τον όρο OPT συμβολίζουμε το βέλτιστο μονοπάτι στο G και με το $w(OPT)$ συνολικό βάρος των ακμών που επισκέπτεται.

Θα περιγράψουμε έναν αλγόριθμο που ανάγει την είσοδο σε ένα στιγμιότυπο του weighted OP όπου όλα τα βάρη των κορυφών είναι ακέραιοι μεταξύ 1 και n^2 . Μαντεύουμε¹⁰ την κορυφή u η οποία έχει το μέγιστο όφελος και ανήκει στο βέλτιστο μονοπάτι. Έστω G' το γράφημα το οποίο προκύπτει έχοντας

αφαιρέσει από το G όλες τις κορυφές με όφελος μεγαλύτερο από $w(u)$ και μικρότερο από $\frac{w(u)}{n^2}$. Έστω

OPT' το μονοπάτι στο G' που προκύπτει με συντόμηση του OPT ώστε να παρακάμψει τις κορυφές που έχουμε αφαιρέσει. Το μήκος του είναι το πολύ όσο είναι το προκαθορισμένο όριο B (budget) και το συνολικό όφελος $w(OPT')$ των κορυφών που επισκέπτεται είναι τουλάχιστον

$$w(OPT) - n \cdot \frac{w(u)}{n^2} = w(OPT) - \frac{w(u)}{n} \geq w(OPT) \left(1 - \frac{1}{n}\right) \text{ καθώς } w(OPT) \geq w(u).$$

Στη συνέχεια θέτουμε σε κάθε κορυφή v το όφελος $w'(v) = \left\lfloor \frac{n^2}{w(u)} \cdot w(v) \right\rfloor$ και παρατηρούμε ότι το $w'(v)$

είναι ακέραιος μεταξύ του 1 και του n^2 . Το νέο όφελος του OPT' δηλαδή το $w'(OPT')$ θα είναι: $w'(OPT') =$

$\frac{n^2}{w(u)} \cdot w(OPT') \geq \frac{n^2}{w(u)} \cdot \left(1 - \frac{1}{n}\right) \cdot w(OPT) - n$. Ο τελευταίος όρος n προκύπτει από τη συνάρτηση του

κατωφλιού καθώς το OPT' επισκέπτεται το πολύ n κορυφές. Λαμβάνοντας υπόψη ότι $w(OPT) \geq w(u)$

έχουμε $\frac{w(OPT)}{w(u)} \geq 1 \Leftrightarrow -n \cdot \frac{w(OPT)}{w(u)} \leq -n$. Βάσει αυτού και από τη σχέση $w'(OPT') \geq \frac{n^2}{w(u)} \cdot \left(1 - \frac{1}{n}\right) \cdot w(OPT) - n$ που δείξαμε προηγουμένως προκύπτει

$\frac{1}{n} \cdot w(OPT) - n$

¹⁰ Δηλαδή δοκιμάζουμε κάθε μια από τις n κορυφές

$$w'(OPT') \geq \frac{n^2}{w(u)} \cdot \left(1 - \frac{1}{n}\right) \cdot w(OPT) - n \cdot \frac{w(OPT)}{w(u)} = \frac{w(OPT)}{w(u)} \cdot n \cdot (n-2) = \frac{n^2}{w(u)} \cdot \left(1 - \frac{2}{n}\right) \cdot w(OPT).$$

Από το Λήμμα 4.1 μπορούμε να πετύχουμε μια β -προσεγγιστική λύση στο στιγμιότυπο του βεβαρυμένου Orienteering στο G' με βάρη κορυφών w' . Βρίσκουμε ένα μονοπάτι P τέτοιο ώστε $w'(P) \geq \frac{1}{\beta} \cdot w'(OPT') \geq \frac{1}{\beta} \cdot \left[\frac{n^2}{w(u)} \cdot \left(1 - \frac{2}{n}\right) \cdot w(OPT)\right]$. Αλλά για κάθε κορυφή v ισχύει $w(v) \geq \frac{w(u)}{n^2} \cdot w'(v)$. Για αυτό $w(P) \geq \frac{1}{\beta} \cdot \left(1 - \frac{2}{n}\right) \cdot w(OPT)$ που αποδεικνύει το ζητούμενο. \square

5 Orienteering Problem with Time-Windows(OPTW)

Το OPTW είναι μια γενίκευση του απλού Orienteering. Σε αυτό το πρόβλημα έχουμε έναν επιπλέον περιορισμό. Μας δίνεται ένα χρονικό διάστημα (time window) $[R(u), D(u)]$ για κάθε κορυφή u . Το όφελος επισκεψιμότητας μιας κορυφής προσμετράται μόνο εάν καλύψουμε αυτήν την κορυφή μέσα σε αυτό το διάστημα. Αν επισκεφτούμε μια κορυφή πριν από το $R(u)$ μπορούμε να επιλέξουμε να περιμένουμε μέχρι το $R(u)$.

Ο Chekuri [10] εμπνεόμενος από την τεχνική του Bansal [12] απέδειξαν ότι μια α προσέγγιση για το Orienteering οδηγεί σε έναν $O(\alpha \cdot \max\{\log OPT, \log L\})$ προσεγγιστικό αλγόριθμο του OPTW σε κατευθυνόμενα (και μη κατευθυνόμενα) γραφήματα. Με OPT συμβολίζουμε τον αριθμό των κόμβων της

βέλτιστης λύσης και με L συμβολίζουμε το λόγο $\frac{L_{\max}}{L_{\min}}$ όπου L_{\max} το μήκος του μεγαλύτερου χρονικού

διαστήματος και L_{\min} το μήκος του μικρότερου διαστήματος.

Τα αποτελέσματα του [10] μπορούν να εφαρμοστούν τόσο σε κατευθυνόμενα όσο και μη κατευθυνόμενα γραφήματα. Η κεντρική ιδέα είναι να χωρίσουμε το πρόβλημα σε υποπρόβληματα στα οποία μπορεί να εφαρμοστεί το απλό Orienteering. Συγκεκριμένα κάθε υποπρόβλημα περιέχει ένα υποσύνολο V' του συνόλου των κορυφών V , κορυφές s' και v' που συμβολίζουν τις κορυφές έναρξης και λήξης του μονοπατιού στο υποπρόβλημα και ένα χρονικό διάστημα I . Κατασκευάζουμε τα V' με τέτοιο τρόπο έτσι ώστε το time window κάθε κορυφής του V' να περιέχει ολόκληρο το διάστημα I . Με αυτόν τον τρόπο κάθε υποπρόβλημα αποτελεί ένα στιγμιότυπο του απλού Orienteering. Χρησιμοποιούμε έναν προσεγγιστικό αλγόριθμο του OP για κάθε υποπρόβλημα και στη συνέχεια συνδέουμε τις λύσεις των υποπροβλημάτων χρησιμοποιώντας δυναμικό προγραμματισμό.

Οι όποιες βελτιώσεις στην προσέγγιση του απλού OP οδηγούν σε βελτίωση στην προσέγγιση του OPTW (σε ασύμμετρες ή συμμετρικές μετρικές). Με α θα συμβολίζουμε τον προσεγγιστικό παράγοντα για το

απλό OP που όπως είδαμε είναι $(2+\epsilon)$ για μη κατευθυνόμενα γραφήματα και $O\left(\frac{\log^2 n}{\log \log n}\right)$ για

κατευθυνόμενα. Για κάθε στιγμιότυπο X του OPTW έστω $OPT(X)$ το όφελος της βέλτιστης λύσης για το X .

5.1 Το Γενικό Πλαίσιο

Δεδομένου ενός στιγμιότυπου του OPTW σε γράφημα $G=(V,E)$, θεωρούμε μια διαμέριση $\{V_1, V_2, \dots, V_m\}$ του V [8]. Αντιστοιχίζουμε τους χρόνους R_i και D_i στα V_i έτσι ώστε να ικανοποιούνται οι παρακάτω συνθήκες

- Για κάθε $u \in V_i$, $R(u) \leq R_i$ και $D(u) \geq D_i$

- Για $1 \leq i < m$, $D_i < R_{i+1}$
- Μια βέλτιστη λύση επισκέπτεται κάθε κορυφή στο V_i στο διάστημα $[R_i, D_i]$

Τότε μπορούμε να λύσουμε ένα στιγμιότυπο του Orienteering σε κάθε V_i ξεχωριστά και να συνδυάσουμε τις λύσεις με δυναμικό προγραμματισμό όπως θα δούμε παρακάτω. Ένα στιγμιότυπο του OPTW στο οποίο μπορούμε να κάνουμε μια τέτοια διαμέριση του συνόλου κορυφών καλείται modular στιγμιότυπο. Ένα τυχαίο στιγμιότυπο του OPTW είναι πιθανότερο να μην είναι modular. Γι αυτό ορίζουμε κάποιες περιορισμένες εκδοχές ενός δοθέντος στιγμιότυπου:

Όρισμός: Έστω A και B στιγμιότυπα του OPTW στον ίδιο υποκείμενο γράφο (με τα ίδια βάρη ακμών) και έστω $I_A(v)$ και $I_B(v)$ τα διαστήματα της κορυφής v στα A και B αντίστοιχα. Λέμε ότι το B είναι μια περιορισμένη εκδοχή του A (restricted version) αν για κάθε v $I_B(v)$ είναι υποδιάστημα του $I_A(v)$.

Προφανώς μια διαδρομή η οποία συλλέγει ένα συγκεκριμένο όφελος σε μια περιορισμένη εκδοχή ενός στιγμιότυπου, θα συλλέγει τουλάχιστον το ίδιο όφελος στο αρχικό στιγμιότυπο. Θα προσπαθήσουμε να επιλύσουμε το OPTW κατασκευάζοντας ένα σύνολο από περιορισμένες εκδοχές με τις οποίες είναι ευκολότερο να δουλέψουμε. Η κατασκευή είναι τέτοια ώστε το όφελος μιας βέλτιστης λύσης σε τουλάχιστον μια από τις περιορισμένες εκδοχές να αποτελεί ένα σημαντικό κομμάτι του οφέλους μιας βέλτιστης λύσης στο αυθεντικό στιγμιότυπο. Άρα μια προσέγγιση στη βέλτιστη λύση της 'καλύτερης' περιορισμένης εκδοχής, μας οδηγεί σε μια προσέγγιση στο αρχικό στιγμιότυπο.

Πρόταση 5.1: Έστω A ένα στιγμιότυπο του OPTW σε ένα γράφημα $G(V,E)$. Αν B_1, B_2, \dots, B_β είναι περιορισμένες εκδοχές του A και $I_A(v) = \bigcup_{1 \leq i \leq \beta} I_{B_i}(v)$ για κάθε κορυφή v, υπάρχει κάποιο B_j τέτοιο

$$\text{ώστε } \text{OPT}(B_j) \geq \frac{\text{OPT}(A)}{\beta}.$$

Απόδειξη: Ισχύει $\text{OPT}(A) = \text{OPT}(B_1 \cup B_2 \cup \dots \cup B_\beta) \leq \text{OPT}(B_1) + \text{OPT}(B_2) + \dots + \text{OPT}(B_\beta)$. Έστω B_j η περιορισμένη εκδοχή με μεγαλύτερο όφελος από τις υπόλοιπες. Θα έχουμε:

$$\text{OPT}(B_1) + \text{OPT}(B_2) + \dots + \text{OPT}(B_\beta) \leq \beta \cdot \text{OPT}(B_j). \text{ Άρα } \text{OPT}(A) \leq \beta \cdot \text{OPT}(B_j) \Leftrightarrow \text{OPT}(B_j) \geq \frac{\text{OPT}(A)}{\beta} \square$$

Οι περιορισμένες εκδοχές που κατασκευάζουμε συνήθως είναι modular στιγμιότυπα του OPTW. Ο γενικός αλγόριθμος για το OPTW είναι:

1. Κατασκευάζουμε ένα σύνολο από β περιορισμένες εκδοχές του δεδομένου στιγμιότυπου, κάθε περιορισμένη εκδοχή είναι modular στιγμιότυπο
2. Επιλέγουμε την καλύτερη περιορισμένη εκδοχή (δοκιμάζοντας όλες τις εκδοχές), βρίσκουμε μια κατάλληλη διαμέριση και χρησιμοποιούμε μια α -προσέγγιση του απλού Orienteering μαζί με δυναμικό προγραμματισμό για να λύσουμε το στιγμιότυπο.

Ο παραπάνω αλγόριθμος δίνει μια $(\alpha \cdot \beta)$ προσέγγιση του OPTW.

5.1.1 Δυναμικός Προγραμματισμός για modular στιγμιότυπα

Όπως αναφέρθηκε, ένα modular στιγμιότυπο είναι ένα στιγμιότυπο του OPTW με γράφημα $G(V,E)$ του οποίου το σύνολο κορυφών V μπορεί να διαμεριστεί σε υποσύνολα V_1, V_2, \dots, V_m έτσι ώστε μια βέλτιστη λύση επισκέπτεται τις κορυφές του V_i στο διάστημα (R_i, D_i) . Για κάθε κορυφή $v \in V_i$ ισχύει $R(v) \leq R_i$ και $D(v) \geq D_i$. Επιπλέον οι κορυφές του V_i εμφανίζονται στο μονοπάτι πριν από τις κορυφές του V_j για $i < j$.

Για να επιλύσουμε ένα modular στιγμιότυπο, για κάθε V_i 'μαντεύουμε' την πρώτη και τελευταία κορυφή μιας βέλτιστης λύσης (δηλαδή δοκιμάζουμε όλους τους δυνατούς συνδυασμούς) καθώς και τους χρόνους που αυτή η λύση επισκέπτεται την πρώτη και τελευταία κορυφή. Αν α είναι ο παράγοντας προσέγγισης του Orienteering, βρίσκουμε για κάθε V_i μια λύση που συλλέγει το $1/\alpha$ του βέλτιστου οφέλους και στη συνέχεια συνδυάζουμε αυτές τις λύσεις.

Μπορούμε να χρησιμοποιήσουμε το ακόλουθο δυναμικό πρόγραμμα: Για κάθε $u, v \in V_i$ θεωρούμε το γράφημα που περιέχει μόνο της κορυφές του V_i και έστω $\text{OPT}(u, v, t)$ να συμβολίζει το βέλτιστο όφελος ενός u - v μονοπατιού μήκους το πολύ t (αγνοώντας τα time windows). Ορίζουμε $\Pi_i(v, T)$ με $v \in V_i$ και $R_i \leq T \leq D_i$ ως το βέλτιστο όφελος μιας διαδρομής που ξεκινά από το s σε χρόνο 0 και καταλήγει στο v σε χρόνο μέχρι T . Γνωρίζοντας το $\text{OPT}(u, v, t)$ μπορούμε να υπολογίσουμε το $\Pi_i(v, T)$ χρησιμοποιώντας την παρακάτω αναδρομή:

$$\Pi_i(v, T) = \max_{u \in V_i, w \in V_{i-1}, t \leq T - R_i} \text{OPT}(u, v, t) + \Pi_{i-1}(w, T - t - d(w, u)) \quad (1)$$

Ωστόσο δεν μπορούμε να υπολογίσουμε ακριβώς το $\text{OPT}(u, v, t)$. Αντί αυτού χρησιμοποιούμε έναν α -προσεγγιστικό αλγόριθμο του Orienteering για να υπολογίσουμε μια προσέγγιση του $\text{OPT}(u, v, t)$ για όλα τα $u, v \in V_i$ και $t \leq D_i - R_i$. Έτσι παίρνουμε μια α -προσέγγιση του $\Pi_i(v, T)$ χρησιμοποιώντας την παραπάνω αναδρομή.

Ο χρόνος αυτού του αλγορίθμου εξαρτάται πολυωνυμικά από το T . Αυτό οδηγεί σε έναν ψευδο-πολυωνυμικό αλγόριθμο. Για να πετύχουμε έναν πολυωνυμικό αλγόριθμο χρησιμοποιούμε μια τεχνική δυναμικού προγραμματισμού η οποία βασίζεται στο όφελος αντί του χρόνου. Όπως αναφέρθηκε στο κεφάλαιο 4 μπορούμε να ανάγουμε το πρόβλημα με τυχαία οφέλη στους κόμβους στο πρόβλημα όπου κάθε κόμβος έχει μοναδιαίο όφελος. Η απώλεια στην προσέγγιση είναι $(1+O(1))$. Δηλαδή το μέγιστο όφελος είναι n .

Για να δημιουργήσουμε ένα δυναμικό πρόγραμμα που βασίζεται στο όφελος και όχι στο χρόνο πρέπει για κάθε $u, v \in V_i$ και για κάθε $k_i \in [0, |V_i|]$ να υπολογιστεί το συντομότερο u - v μονοπάτι που συλλέγει όφελος τουλάχιστον k_i . Αυτό δεν μπορεί να υπολογιστεί με κάποιον ακριβή αλγόριθμο. Αν δοκιμάσουμε μια προσεγγιστική λύση μπορεί να έχουμε μεγάλες αποκλίσεις για τα τελευταία διαστήματα V_j . Αντί για αυτό 'μαντεύουμε' το μήκος ενός βέλτιστου μονοπατιού με όφελος k_i (χρησιμοποιώντας δυαδική αναζήτηση στο μέγιστο μήκος μονοπατιού στο G) και για κάθε τέτοιο μήκος χρησιμοποιούμε μια α -προσέγγιση του Orienteering. Αυτό μας εξασφαλίζει ότι εάν υπάρχει ένα u - v μονοπάτι μήκους B που συλλέγει όφελος k_i τότε βρίσκουμε ένα u - v μονοπάτι μήκους το πολύ B που συλλέγει όφελος τουλάχιστον k_i/α . Τελικά για να βρούμε την προσέγγιση στο αρχικά δοσμένο στιγμιότυπο του OPTW ενώνουμε τις λύσεις από κάθε V_i χρησιμοποιώντας δυναμικό προγραμματισμό παρόμοιο με το (1).

5.2 Αλγόριθμοι

Με βάση το παραπάνω πλαίσιο θα αναπτύξουμε αλγόριθμους που πετυχαίνουν προσεγγιστικό λόγο που εξαρτάται από το μήκος των χρονικών διαστημάτων (time windows) [8]. Αρχικά θεωρούμε στιγμιότυπα που όλα τα άκρα των time windows είναι ακέραιοι και στη συνέχεια στιγμιότυπα που έχουν φραγμένο

λόγο $L = \frac{L_{\max}}{L_{\min}}$. Τέλος συνδυάζουμε αυτές τις ιδέες για να προκύψει μια $O(\alpha \cdot \max\{\log \text{OPT}, \log L\})$ -

προσέγγιση για όλα τα στιγμιότυπα του OPTW.

5.2.1 Μια $O(\alpha \cdot \log L_{\max})$ -προσέγγιση

Εστιαζόμαστε σε στιγμιότυπα του OPTW όπου για όλες τις κορυφές v , τα $R(v)$ και $D(v)$ είναι ακέραιοι. Ο αλγόριθμος βασίζεται στο παρακάτω λήμμα:

Λήμμα 5.2: Κάθε διάστημα μήκους $M > 1$ με ακέραια άκρα μπορεί να χωριστεί το πολύ σε $2 \lfloor \log M \rfloor$ ξένα υπο-διαστήματα έτσι ώστε το μήκος κάθε υποδιαστήματος να είναι δύναμη του 2 και κάθε υποδιάστημα μήκους 2^i να αρχίζει σε ένα πολλαπλάσιο του 2^i . Επιπλέον υπάρχουν το πολύ δυο υποδιαστήματα κάθε μήκους.

Απόδειξη: Θα κάνουμε επαγωγή στο μήκος του διαστήματος. Προφανώς το λήμμα ισχύει για διαστήματα μήκους 2 ή 3. Με χρήση το πολύ 2 υποδιαστημάτων μήκους 1 στην αρχή και στο τέλος του εξεταζόμενου διαστήματος, το υπολειπόμενο διάστημα αρχίζει και τερματίζει σε άρτιο ακέραιο. Για να καλυφθεί το υπολειπόμενο διάστημα διαιρούνται όλοι οι ακέραιοι του διαστήματος με 2 και εφαρμόζεται η υπόθεση της επαγωγής. Το σύνολο των υποδιαστημάτων που χρησιμοποιούνται είναι το πολύ $2 + (2 \cdot \lfloor \log(M/2) \rfloor) \leq 2 \cdot \lfloor \log M \rfloor$. Χρησιμοποιούνται το πολύ 2 υποδιαστήματα κάθε μήκους και τα διαστήματα μήκους 2^i χρησιμοποιούνται στο $i + 1$ επίπεδο της αναδρομής. \square

Συμβολίζουμε με l την ποσότητα $\lfloor \log L_{\max} \rfloor$ και υποθέτουμε ότι $L_{\max} \geq 2$. Δεδομένου ενός στιγμιότυπου του OPTW και για κάθε κορυφή v με διάστημα $I(v)$ χρησιμοποιούμε το λήμμα 5.2 για να διαμερίσουμε το $I(v)$ σε $2 \cdot l$ το πολύ υποδιαστήματα. Επισημαίνουμε τα υποδιαστήματα του $I(v)$ ως εξής: για κάθε $1 \leq i \leq l$, το πρώτο υποδιάστημα μήκους 2^i θα συμβολίζεται ως $I_i^1(v)$ και το δεύτερο $I_i^2(v)$. (ενδεχομένως να μην υπάρχει διάστημα μήκους 2^i)

Κατασκευάζουμε τώρα ένα σύνολο από $2 \cdot l$ το πολύ περιορισμένες εκδοχές του αρχικά δοσμένου στιγμιότυπου. Καλούμε αυτές τις εκδοχές $B_1^1, B_2^1, \dots, B_l^1$ και $B_1^2, B_2^2, \dots, B_l^2$ έτσι ώστε το διάστημα της κορυφής v στο B_i^b να είναι το $I_i^b(v)$. Αν το διάστημα $I_i^b(v)$ δεν χρησιμοποιείται στη διαμέριση του $I(v)$, η κορυφή v δεν εμφανίζεται στην περιορισμένη εκδοχή. (Ισοδύναμα έχει όφελος 0 ή έχει κενό time window)

Έστω ένα αυθαίρετο στιγμιότυπο B_i^b . Όλες οι κορυφές σε αυτό το στιγμιότυπο του OPTW έχουν διαστήματα μήκους 2^i και όλα τα time windows ξεκινούν από έναν ακέραιο που είναι πολλαπλό του 2^i . Άρα οποιοσδήποτε δυο κορυφές έχουν time windows που είτε ταυτίζονται είτε είναι τελείως ξένα μεταξύ τους. Αυτό σημαίνει ότι το B_i^b είναι modular στιγμιότυπο και μπορούμε να το χωρίσουμε σε υποπροβλήματα, σε καθένα από τα οποία εφαρμόζουμε μια α προσέγγιση του απλού Orienteering. Επικολλώντας τα υποπροβλήματα προκύπτει μια α προσέγγιση του περιορισμένου στιγμιότυπου (δηλ. του B_i^b).

Από την πρόταση 5.1 έχουμε ότι μια από τις περιορισμένες εκδοχές έχει βέλτιστη λύση η οποία συλλέγει όφελος τουλάχιστον $\frac{OPT}{2l}$. Χρησιμοποιώντας μια α -προσέγγιση του OP σε αυτήν την περιορισμένη εκδοχή παίρνουμε μια $\alpha \cdot 2l = O(\alpha \cdot \log L_{\max})$ προσέγγιση του OPTW όταν όλα τα άκρα των διαστημάτων είναι ακέραιοι.

5.2.2 Μια $O(\alpha \cdot \log OPT)$ -προσέγγιση όταν $L \leq 2$

Για ένα στιγμιότυπο του OPTW όπου $L = \frac{L_{\max}}{L_{\min}} \leq 2$ κάνουμε scaling στα άκρα των διαστημάτων (time windows) και στα μήκη των ακμών έτσι ώστε $L_{\min} = 1$ (συνεπώς $L_{\max} \leq 2$). Παρατηρούμε ότι ακόμα και αν

τα άκρα των διαστημάτων είναι ακέραια, μετά το scaling μπορεί να μην είναι ακέραια στην scaling εκδοχή. Στη scaling εκδοχή όλα τα διαστήματα θα είναι υποσύνολα του $[1,2]$.

Για κάθε κορυφή v , διαμερίζουμε το $I(v)=[R(v),D(v)]$ σε 3 υποδιαστήματα: $I_1(v)=[R(v),a]$, $I_2(v)=[a,b]$ και $I_3(v)=[b,D(v)]$, όπου $a=\lfloor R(v)+1 \rfloor$ (δηλ. ο επόμενος ακέραιος αυστηρά μεγαλύτερος από το $R(v)$) και $b=\lceil D(v)-1 \rceil$ (δηλ. ο μεγαλύτερος ακέραιος αυστηρά μικρότερος από το $D(v)$). Το $I_2(v)$ μπορεί να είναι σημείο. Σε αυτήν την περίπτωση αγνοούμε αυτό το υποδιάστημα.

Κατασκευάζουμε 3 περιορισμένες εκδοχές του αρχικά δοσμένου στιγμιότυπου έστω B_1, B_2, B_3 έτσι ώστε το διάστημα για κάθε κορυφή v στο B_i να είναι $I_i(v)$. Από την πρόταση 5.1 προκύπτει ότι κάποια από τις 3 περιορισμένες εκδοχές έχει βέλτιστη λύση η οποία συλλέγει τουλάχιστον το $\frac{1}{3}$ του οφέλους της βέλτιστης λύσης του αρχικού στιγμιότυπου. Υποθέτουμε ότι αυτή η περιορισμένη εκδοχή είναι το B_2 . Όλα τα time windows έχουν μήκος ακριβώς 1 με ακέραια άκρα. Συνεπώς το B_2 είναι modular στιγμιότυπο και μπορούμε να πάρουμε μια α -προσέγγιση της βέλτιστης λύσης του B_2 . Αυτή η προσέγγιση δίνει μια 3α -προσέγγιση στο αρχικό στιγμιότυπο.

Στην περίπτωση των B_1 και B_3 έχουμε πιο περίπλοκες περιπτώσεις καθώς δεν είναι modular. Κάθε διάστημα στο B_1 έχει μήκος το πολύ 1 και τελειώνει σε ακέραιο και κάθε διάστημα στο B_3 έχει μήκος το πολύ 1 και αρχίζει από ακέραιο. Στη συνέχεια θα παρουσιαστεί ένας τρόπος προσέγγισης της λύσης του B_3 με παράγοντα $O(\alpha \cdot \log OPT)$. Για το B_1 ο αλγόριθμος είναι ταυτόσημος εκτός από το ότι πρέπει να εναλλάσσουμε τα $R(v)$ και $D(v)$. (release times και deadlines).

Για το B_3 μπορούμε να διαμερίσουμε το σύνολο των κορυφών στα σύνολα V_1, V_2, \dots, V_m έτσι ώστε κάθε κορυφή στο V_i να καλύπτεται πριν από κάθε κορυφή στο V_j για $i < j$. Τα deadlines (δηλ. τα $D(v)$) για κορυφές που ανήκουν σε κάποιο V_i μπορεί να είναι όλα διακριτά. Μπορούμε να εφαρμόσουμε τον αλγόριθμο του Orient-deadline ξεχωριστά σε κάθε V_i και να ενώσουμε τις λύσεις με δυναμικό προγραμματισμό. Το Orient-Deadline έχει μελετηθεί στο [12] και αποδείχθηκε ότι υπάρχει μια $O(\log OPT)$ -προσέγγιση για αυτό το πρόβλημα. Είναι μια ειδική περίπτωση του OPTW στην οποία έχουμε $R(v)=0$ για κάθε κορυφή v . Η λύση που θα προκύψει για το B_3 θα συλλέγει τουλάχιστον $\Omega\left(\frac{1}{a \log OPT}\right)$ του

οφέλους της βέλτιστης λύσης του B_3 . Ο παράγοντας a προκύπτει από την τεχνική δυναμικού προγραμματισμού που περιγράφηκε στην ενότητα 5.1.1. Συνεπώς αυτό μας δίνει μια $3 \cdot O(\alpha \cdot \log OPT) = O(\alpha \cdot \log OPT)$ -προσέγγιση του αρχικά δοσμένου στιγμιότυπου.

Παρομοίως μπορούμε να πετύχουμε μια $O(\alpha \cdot \log OPT)$ -προσέγγιση για το B_1 χρησιμοποιώντας τον αλγόριθμο για το orienteering με release times που δίνει προσέγγιση $O(\log OPT)$. Άρα όταν $L \leq 2$, έχουμε μια $O(\alpha \cdot \log OPT)$ -προσέγγιση του OPTW.

5.3 Γενικός Αλγόριθμος για το OPTW

Ένα τυχαίο στιγμιότυπο του OPTW μπορεί να έχει $L > 2$ και τα άκρα των time-windows να μην είναι ακέραιοι. Ωστόσο μπορούμε να συνδυάσουμε τους παραπάνω δυο αλγορίθμους ώστε να επιλύουμε τέτοια στιγμιότυπα. Ξεκινάμε κάνοντας scaling στα άκρα των διαστημάτων έτσι ώστε το μικρότερο

διάστημα να έχει μήκος 1. Το μεγαλύτερο διάστημα τώρα έχει μήκος $\frac{L_{\max}}{L_{\min}}$ όπου L_{\max} και L_{\min} τα μήκη του μεγαλύτερου και του μικρότερου διαστήματος αντίστοιχα στο αρχικό στιγμιότυπο.

Κατασκευάζουμε 3 περιορισμένες εκδοχές του scaled στιγμιότυπου: B_1, B_2, B_3 . Για κάθε κορυφή v με $l(v)=[R(v), D(v)]$ στο scaled στιγμιότυπο δημιουργούμε 3 υποδιαστήματα. Αν $l(v) < 2$ θέτουμε $I_1(v)=[R(v), D(v)]$ και $I_2(v) = I_3(v) = \emptyset$. Διαφορετικά $I_1(v)=[R(v), a]$, $I_2(v)=[a, b]$ και $I_3(v)=[b, D(v)]$ όπου $a = \lfloor R(v) + 1 \rfloor$ και $b = \lceil D(v) - 1 \rceil$. Όπως και πριν το διάστημα του v στο στιγμιότυπο B_i είναι $I_i(v)$.

Μια από τις περιορισμένες εκδοχές συλλέγει τουλάχιστον το $\frac{1}{3}$ του οφέλους της βέλτιστης λύσης του αρχικού στιγμιότυπου (σύμφωνα με την πρόταση 5.1). Έστω ότι είναι η B_1 ή B_3 . Όλα τα διαστήματα στη B_1 ή στη B_3 έχουν μήκος μεταξύ 1 και 2 εκ κατασκευής. Γι αυτό μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο με $O(\alpha \cdot \log OPT)$ -προσέγγιση της παραγράφου 5.2.2 για να συλλέξουμε τουλάχιστον $\Omega(\frac{1}{\alpha} \cdot \log OPT)$ του οφέλους μιας βέλτιστης λύσης στο αυθεντικό στιγμιότυπο. Στην περίπτωση που το B_2

συλλέγει πάνω από το $\frac{1}{3}$ της βέλτιστης λύσης του αρχικού στιγμιότυπου μπορούμε να

χρησιμοποιήσουμε τον αλγόριθμο της παραγράφου 5.2.1 γιατί τα άκρα των διαστημάτων είναι ακέραιοι και το μεγαλύτερο διάστημα έχει μήκος μικρότερο από L . Σε αυτήν την περίπτωση παίρνουμε μια $O(\alpha \cdot \log L)$ -προσέγγιση

Συνεπώς ο αλγόριθμος αυτός που συνδυάζει τους δυο αλγορίθμους των προηγούμενων παραγράφων μας δίνει μια $O(a \cdot \max\{\log OPT, \log L\})$ - προσέγγιση του OPTW.

6 Team Orienteering Problem (TOP)

Το πρόβλημα του ομαδικού προσανατολισμού (Team Orienteering Problem) αποτελεί γενίκευση του απλού προβλήματος προσανατολισμού (Orienteering Problem) όπως αυτό παρουσιάστηκε παραπάνω. Εφαρμόζεται στο σχεδιασμό τουριστικών διαδρομών καθώς και στη δρομολόγηση των οχημάτων μιας εταιρείας.

6.1 Ορισμός του TOP

Έστω πλήρες γράφημα $G=(V, E)$ όπου $V=\{v_0, v_1, \dots, v_{n+1}\}$ το σύνολο των κορυφών και E είναι το σύνολο των ακμών. Οι κορυφές v_0 και v_{n+1} αντιστοιχούν στο σημείο αναχώρησης και τερματισμού αντίστοιχα (μπορούν και να ταυτίζονται). Κάθε κορυφή v_i σχετίζεται με ένα όφελος p_i το οποίο λαμβάνεται με την επίσκεψη στον κόμβο. Το όφελος των κορυφών v_0 και v_{n+1} είναι μηδενικό. Σε κάθε ακμή του γραφήματος (v_i, v_j) αντιστοιχίζεται ένα κόστος c_{ij} . Έστω m ο αριθμός των διαδρομών και B το όριο κόστους ανά διαδρομή. Το Team Orienteering πρόβλημα αφορά την εύρεση m διαδρομών από το v_0 στο v_{n+1} με μήκος το πολύ B η κάθε μια έτσι ώστε η κάθε κορυφή να διαπερνάται το πολύ μια φορά και να μεγιστοποιείται το συνολικό όφελος των διαδρομών [4]. Στη γενική περίπτωση του Team Orienteering Problem (TOP) οι διαδρομές μπορούν να έχουν διαφορετικούς κόμβους έναρξης και τερματισμού μεταξύ τους. Επίσης μπορούν να έχουν διαφορετικό όριο κόστους, να είναι κυκλικές (δηλ. να ξεκινούν και να τερματίζουν σε ένα συγκεκριμένο κόμβο ο οποίος μπορεί να είναι ξενοδοχείο) κοκ.

Για τη μαθηματική μοντελοποίηση του προβλήματος παραπέμπουμε στην παράγραφο 7.1.1 που ορίζεται μια επέκταση του TOP με έναν επιπλέον περιορισμό (time windows).

Το TOP είναι NP-hard και APX-hard γιατί αποτελεί γενίκευση του απλού OP. Σε αυτήν την παράγραφο θα παρουσιάσουμε κάποιους ευρετικούς αλγόριθμους για την επίλυση του προβλήματος. Πρόκειται για τους αλγόριθμους GRASP των Souffriau et al. [17] και τον CGW των Chao et al. [18].

6.2 Η τεχνική Greedy Randomized Adaptive Search Procedure (GRASP) στο Team Orienteering Problem

Σε αυτήν την ενότητα παρουσιάζουμε την τεχνική GRASP η οποία εφαρμόζεται στην επίλυση του Team Orienteering προβλήματος. Η GRASP παρουσιάστηκε αρχικά από τους Feo και Resende [16]. Οι Souffriau et al. [17] εφάρμοσαν αυτήν την τεχνική στο TOP και προέκυψε ένας ευρετικός αλγόριθμος.

Σε γενικές γραμμές η GRASP εκτελεί κάποιες επαναλήψεις κάθε μια από τις οποίες περιέχει μία κατασκευαστική μέθοδο (construct) και μια διαδικασία τοπικής αναζήτησης. Η κατασκευαστική μέθοδος δέχεται ως παράμετρο μια μεταβλητή (η οποία λέγεται βαθμός απληστίας ή greediness) που καθορίζει την αναλογία μεταξύ της απληστίας του αλγορίθμου και της τυχαιότητας. Οι επαναλήψεις είναι ανεξάρτητες μεταξύ τους και η καλύτερη λύση αποθηκεύεται και επιστρέφεται στο τέλος του αλγορίθμου ως αποτέλεσμα.

Ο ευρετικός αλγόριθμος των Souffriau et al. [17] έχει ως εξής: Αρχικά παράγεται μια κενή λύση στην οποία κάθε μια από τις m διαδρομές περιέχει μόνο τον αρχικό και τελικό κόμβο. Για να κατασκευάσουμε μια αρχική λύση δημιουργούμε μια λίστα με υποψήφιους κόμβους οι οποίοι είναι εφικτοί να εισαχθούν στη λύση. Αυτή τη λίστα θα την συμβολίζουμε CL (Candidate List). Μια κίνηση εισαγωγής ενός κόμβου l μεταξύ των κόμβων i και j θα τη συμβολίζουμε IM_{ij} . Προφανώς ο l δεν πρέπει να ανήκει στη λύση.

Μια τέτοια εισαγωγή αυξάνει το συνολικό όφελος της λύσης όσο είναι το όφελος του κόμβου l . Επιπλέον επιβαρύνει το χρόνο της διαδρομής. Έστω t_{ij} η αύξηση από την εισαγωγή του l μεταξύ των i και j .

Αυτή η ποσότητα θα είναι ίση με $t_{ij} = t_{il} + t_{lj} - t_{ij}$ (όπου t_{ij} ο χρόνος μετάβασης από τον i στον j). Για να είναι εφικτή μια εισαγωγή πρέπει ο χρόνος επιβάρυνσης να είναι μικρότερος ή ίσος από το υπολειπόμενο χρονικό όριο. Για κάθε κόμβο που ανήκει στη λίστα CL υπολογίζουμε μια ποσότητα που θα

χρησιμεύσει στη συνέχεια του αλγορίθμου. Αυτή η ποσότητα είναι $h_{ij} = \frac{p_l}{t_{ij}}$ και θα την ονομάζουμε

ευρετική τιμή. Έστω Min και Max η ελάχιστη και η μέγιστη τιμή αντίστοιχα της ευρετικής τιμής των κόμβων της CL. Υπολογίζουμε μια νέα τιμή th (threshold) ως εξής: $th = \text{Min} + \text{greediness} \times (\text{Max} - \text{Min})$. Δημιουργούμε μια νέα λίστα RCL (Restricted Candidate List) η οποία θα περιέχει μόνο τους κόμβους εκείνους από τη CL που έχουν ευρετική τιμή μεγαλύτερη από th . Από τη λίστα RCL επιλέγουμε τυχαία έναν κόμβο και τον τοποθετούμε στη λύση. Επανακατασκευάζουμε τη λίστα CL και επαναλαμβάνουμε την ίδια διαδικασία μέχρι να μη βρεθούν άλλοι κόμβοι εφικτοί προς εισαγωγή στη λίστα CL.

Στο τέλος κάθε επανάληψης η λύση που κατασκευάστηκε βελτιώνεται μέσω μιας διαδικασίας τοπικής αναζήτησης. Στη διαδικασία αυτή εκτελούνται δυο κινήσεις (2-opt και swap) που έχουν στόχο να μειώσουν το συνολικό κόστος των διαδρομών εναλλάσσοντας κορυφές μεταξύ των διαδρομών. Στη συνέχεια εκτελούνται κινήσεις αντικατάστασης (replace) που εναλλάσσουν κορυφές που ανήκουν στη λύση με κορυφές που δεν περιλαμβάνονται στη λύση. Τέλος εφαρμόζονται ξανά κινήσεις εισαγωγής. Οι τέσσερις αυτές κινήσεις εφαρμόζονται διαδοχικά μέχρι να μην υπάρχει βελτίωση της λύσης.

Η μέθοδος που περιγράψαμε παραπάνω αντιστοιχεί σε μια επανάληψη της GRASP για το Team Orienteering πρόβλημα. Ο αλγόριθμος εκτελείται μέχρι να μην προκύψει καλύτερη λύση έχοντας εκτελέσει έναν προκαθορισμένο αριθμό συνεχών επαναλήψεων. Δίνεται παρακάτω ο ψευδοκώδικας μιας επανάληψης της τεχνικής GRASP όπως παρουσιάζεται από τους Souffriau et al. [17].

Ψευδοκώδικας GRASP για το Team Orienteering (1 επανάληψη)

Solution ← empty solution;

CL ← κόμβοι εφικτοί προς εισαγωγή;

Ενώσω CL μη κενό **Κάνε**

Min ← ελάχιστη ευρετική τιμή του CL;

Max ← μέγιστη ευρετική τιμή του CL;

Threshold ← $\text{Min} + \text{greediness} \times (\text{Max} - \text{Min})$;

RCL ← empty list;

Για κάθε $c \in \text{CL}$ **Κάνε**

Αν ευρετική τιμή του $c \geq \text{Threshold}$ **Τότε**

Πρόσθεσε το c στην RCL

Τέλος Αν

Τέλος Για

Τελικός κόμβος προς εισαγωγή ← Τυχαία επιλογή από την RCL;

Solution ← $\text{Solution} \cup \{\text{Τελικός κόμβος προς εισαγωγή}\}$;

CL ← κόμβοι εφικτοί προς εισαγωγή;

Τέλος Ενώσω

Solution ← **Τοπική Αναζήτηση**(Solution);

Ψευδοκώδικας της Τοπικής Αναζήτησης στον αλγόριθμο GRASP

Ενώσω δεν έχει βρεθεί τοπικό μέγιστο **Κάνε**

2-opt

Swap

Replace

Insert

Τέλος Ενώσω

6.2.1 Συνδυασμός GRASP με Path Relinking στο Team Orienteering Πρόβλημα

Το γεγονός ότι στον αλγόριθμο που περιγράψαμε στην ενότητα 6.2, οι επαναλήψεις είναι ανεξάρτητες μεταξύ τους αποτελεί μειονέκτημα. Κάθε επανάληψη ξεκινά δημιουργώντας μια κενή λύση και δε λαμβάνονται υπόψη περιοχές που εξερευνήθηκαν νωρίτερα στο χώρο αναζήτησης και ενδεχομένως να συμβάλουν θετικά στη λύση. Για να αντιμετωπιστεί αυτό το πρόβλημα οι Souffriau et al. [17] παρουσιάζουν μια επέκταση της βασικής τεχνικής GRASP στο Team Orienteering πρόβλημα.

Συγκεκριμένα χρησιμοποιούν τη μέθοδο path relinking (επανασύνδεση μονοπατιού) σε συνδυασμό με GRASP. Ο ευρετικός αλγόριθμος περιγράφεται ως εξής [17]: Οι διαδικασίες της κατασκευής (construct) και της τοπικής αναζήτησης (local search) που χρησιμοποιήθηκαν στην απλή GRASP τεχνική της ενότητας 6.2 εφαρμόζονται και τώρα. Μέχρι να εκτελεστεί ένας προκαθορισμένος αριθμός συνεχών επαναλήψεων χωρίς να προκύψει βελτίωση στη λύση, εφαρμόζεται η construct ακολουθούμενη από τη local search. Αμέσως μετά εκτελείται path relinking εξερευνώντας τα μονοπάτια μεταξύ της προκύπτουσας λύσης και των στοιχείων ενός συνόλου (καλών) λύσεων (καλούμε το σύνολο αυτό pool of elite solutions) ανά ζεύγη κάθε φορά. Στη συνέχεια αυτό το σύνολο (pool) ενημερώνεται. Ο ψευδοκώδικας με path relinking παρουσιάζεται παρακάτω [17]:

Αλγόριθμος GRASP με path relinking για το TOP

Ενώσω ο αριθμός των επαναλήψεων χωρίς βελτίωση δεν υπερβαίνει μια σταθερά **Κάνε**

Construct;

Local search;

Link to elites;

Update elite pool;

Τέλος Ενόςω**Επέστρεψε** τη βέλτιστη λύση

Θα περιγράψουμε τη λειτουργία Link to elites και στη συνέχεια θα δώσουμε τον ψευδοκώδικα. Η Link to elites δέχεται δυο λύσεις ως ορίσματα, τη start solution και τη guiding solution (λύση οδηγός) και εξερευνά το μονοπάτι μεταξύ αυτών των λύσεων για να βρεθεί καλύτερη λύση. Δημιουργούνται τα σύνολα Set Intersection και Set featuresToAdd. Το πρώτο σύνολο αποτελείται από τους κοινούς κόμβους των starting solution και guiding solution. Το δεύτερο σύνολο αποτελείται από τους κόμβους της guiding solution χωρίς τους κόμβους του πρώτου συνόλου. Από το σύνολο featuresToAdd εισάγονται οι κόμβοι στην starting solution (με μια κίνηση insert) ένας προς έναν. Σε αντίθεση με τον αλγόριθμο του 6.2 μια κίνηση εισαγωγής μπορεί να υπερβαίνει το υπολειπόμενο time budget αν ο κόμβος ανήκει στο σύνολο featuresToAdd και η διαδρομή στην οποία θα εισαχθεί δεν υπερβαίνει το time budget μέχρι εκείνη τη στιγμή. Όταν και οι m διαδρομές υπερβούν το budget (είναι infeasible routes) δεν εισάγονται άλλοι κόμβοι. Στη συνέχεια καλείται μια διαδικασία που επαναφέρει τις διαδρομές σε επιτρεπτή κατάσταση. Για κάθε κόμβο l που ανήκει σε μια μη επιτρεπτή διαδρομή και βρίσκεται μεταξύ των κόμβων i και j ,

υπολογίζουμε την ποσότητα $\frac{1}{h_{ij}} = \frac{t_{ij}}{p_l}$. Ο κόμβος l με τη μεγαλύτερη ποσότητα $\frac{1}{h_{ij}}$ αφαιρείται από τη

διαδρομή. Η αφαίρεση κόμβων συνεχίζεται μέχρι όλες οι διαδρομές να είναι επιτρεπτές. Ακολουθεί ο ψευδοκώδικας της Link to elites [17].

Διαδικασία Link to elites

Set Intersection=το σύνολο των κοινών κόμβων των 2 λύσεων που περνούν ως ορίσματα στη διαδικασία
Set FeaturesToAdd=το σύνολο κόμβων που ανήκουν στη guided Solution και δεν ανήκουν στο Set Intersection

Ενόςω το Set FeaturesToAdd δεν είναι άδειο **Κάνε**

Εισαγωγή κόμβων στη starting solution με επιτρεπόμενη υπέρβαση του budget;

Διαγραφή κόμβων για να επανέλθουν οι διαδρομές σε επιτρεπτή κατάσταση;

Τέλος Ενόςω

Η καλύτερη λύση που βρέθηκε από τη διαδικασία Link to elites εξετάζεται για να εισαχθεί στο σύνολο pool of elites solutions. Για να αποφασίσουμε εάν θα εισαχθεί μια λύση στο pool of elites solutions, υπολογίζουμε μια μετρική ομοιότητας μεταξύ της προς εισαγωγή λύσης και των στοιχείων του pool of elite solutions. Ο ακόλουθος τύπος χρησιμοποιείται για να υπολογιστεί η ομοιότητα μεταξύ δυο λύσεων X και

Y : $\frac{2n_{X \cap Y}}{n_X + n_Y}$ όπου n_X και n_Y ο αριθμός των κόμβων των λύσεων X και Y και $n_{X \cap Y}$ ο αριθμός των

κοινών κόμβων των δυο λύσεων. Αν το elite pool δεν έχει φτάσει το μέγιστο αριθμό λύσεων-μελών και τα μέτρα ομοιότητας της υποψήφιας λύσης προς εισαγωγή και όλων των μελών της elite pool είναι κάτω από μια τιμή threshold, η λύση προστίθεται στο σύνολο elite pool. Αν η υποψήφια λύση είναι πολύ όμοια με ένα μέλος της elite pool, δεν εισάγεται στο σύνολο εκτός αν είναι καλύτερη από την πιο καλή λύση στο elite pool. Τέλος αν το elite pool είναι γεμάτο και η υποψήφια λύση προς εισαγωγή είναι καλύτερη από την χειρότερη λύση του συνόλου elite pool γίνεται αντικατάσταση της χειρότερης λύσης με την υποψήφια λύση.

6.3 Ο ευρετικός αλγόριθμος CGW για το Team Orienteering πρόβλημα των Chao et. al.

Ο ευρετικός αλγόριθμος CGW παρουσιάστηκε από τους I.-M. Chao, Bruce L. Golden και Edward A. Wasil [18] το 1996 και πήρε το όνομά του από τα αρχικά γράμματα του ονόματος των εμπνευστών του. Αποτελείται από δυο βήματα: αρχικοποίηση (initialization) και βελτίωση (improvement). Η τεχνική του

CGW είναι η εφαρμογή τοπικής αναζήτησης σε μια αρχική λύση με στόχο τη βελτίωσή της. Στη συνέχεια εφαρμόζονται δυο κινήσεις διαταραχής για περαιτέρω βελτίωση της λύσης.

Για να κατασκευάσουμε την αρχική λύση στη φάση της αρχικοποίησης δημιουργούμε μια έλλειψη στο χώρο αναζήτησης η οποία έχει ως εστίες τα σημεία έναρξης και τερματισμού της διαδρομής και το time budget B ως το μήκος του κύριου άξονα. Για να δημιουργήσουμε ένα μονοπάτι εστιάζουμε στα σημεία που βρίσκονται εντός της έλλειψης γιατί κάθε μονοπάτι που περιέχει ένα σημείο που βρίσκεται εκτός της έλλειψης θα παραβίαζε το χρονικό όριο B . Η διαδικασία της αρχικοποίησης (initialization) και της βελτίωσης (improvement) θα παρουσιαστούν αναλυτικά στη συνέχεια.

6.3.1 Το βήμα της αρχικοποίησης (Initialization)

Αρχικά ορίζουμε $L = \min(5, N)$ όπου N ο αριθμός των σημείων εντός της έλλειψης που αναφέραμε πιο πάνω. Αυτά τα N σημεία αποτελούν το σύνολο των υποψήφιων σημείων προς εισαγωγή στις m διαδρομές της λύσης. Από τα L σημεία επιλέγουμε m σημεία και εισάγουμε ένα σε κάθε διαδρομή πέραν των σημείων έναρξης και τερματισμού. Στη συνέχεια τα υπόλοιπα σημεία εισάγονται στις διαδρομές (με κριτήριο εισαγωγής να είναι το χαμηλό κόστος) μέχρι να γεμίσουν και οι m διαδρομές. Μια διαδρομή θεωρείται γεμάτη όταν οποιαδήποτε εισαγωγή σημείου σε αυτήν παραβιάζει τον χρονικό περιορισμό δηλ. έχει χρόνο μεγαλύτερο από το B .

Αν παραμείνουν κι άλλα σημεία, η διαδικασία δημιουργεί κι άλλες διαδρομές μέχρι να εξαντληθούν όλα τα υποψήφια προς εισαγωγή σημεία. Στη συνέχεια επιλέγονται οι m διαδρομές που έχουν μεγαλύτερο όφελος. Αυτές οι m διαδρομές αποτελούν την αρχική λύση και το άθροισμα των profit είναι το συνολικό ομαδικό όφελος [18].

Θα συμβολίζεται το σύνολο των m διαδρομών με το μεγαλύτερο συνολικό όφελος ως $paths_{top}$ και το σύνολο των υπόλοιπων διαδρομών ως $paths_{notop}$. Το βήμα της αρχικοποίησης που περιγράφηκε προϋποθέτει ότι τα σημεία βρίσκονται σε έναν δισδιάστατο Ευκλείδιο χώρο. Αν δεν ισχύει αυτό η αρχική λύση μπορεί να κατασκευαστεί με διαφορετικό τρόπο και στη συνέχεια να εφαρμοστεί το βήμα της βελτίωσης.

6.3.2 Το βήμα της βελτίωσης (Improvement)

Το βήμα Improvement χρησιμοποιεί κάποια άλλα βήματα ως υπο-ρουτίνες τα οποία θα περιγράψουμε αναλυτικά. Τα βήματα αυτά είναι: εναλλαγή δυο σημείων (two-point exchange), μετακίνηση ενός σημείου (one-point movement) και clean up. Στο βήμα Improvement επιτρέπεται μείωση του συνολικού οφέλους με την προοπτική να βρούμε τελικά μια καλύτερη λύση. Στόχος είναι η βελτίωση της λύσης που δημιουργήθηκε στο βήμα Initialization.

6.3.2.1 Εναλλαγή δυο σημείων (two-point exchange)

Σε αυτήν την υπο-ρουτίνα γίνεται μετακίνηση ενός σημείου i από κάποια διαδρομή στο $paths_{notop}$ και εισάγεται σε μια διαδρομή που ανήκει στο $paths_{top}$. Επίσης γίνεται μετακίνηση ενός σημείου j που ανήκει σε κάποια διαδρομή στο $paths_{top}$ και εισάγεται σε κάποια διαδρομή στο $paths_{notop}$. Η μετακίνηση των 2 σημείων γίνεται ταυτόχρονα και η εισαγωγή τους γίνεται με βάση το μικρότερο κόστος στη διαδρομή. Εάν δεν είναι εφικτή καμία εισαγωγή στο $paths_{notop}$ (δηλ. κάθε πιθανή εισαγωγή παραβιάζει τον χρονικό περιορισμό), δημιουργείται ένα νέο μονοπάτι που περιέχει το σημείο j καθώς και τους κόμβους έναρξης και λήξης.

Έστω $L(p)$ το μήκος του μονοπατιού p . Για να ελέγξουμε αν το μονοπάτι p ικανοποιεί το περιορισμό του time budget όταν εισάγεται σε αυτό ένα σημείο i και αφαιρείται ένα σημείο j , υπολογίζουμε την εξής ποσότητα:

$$L(p) - (c_{j,next(j)} + c_{prev(j),j} - c_{prev(j),next(j)}) + \min_{k \in p, k \neq 1, j} \{c_{i,k} + c_{i,prev(k)} - c_{k,prev(k)}\} \quad (1)$$

Όπου $c_{i,j}$ το κόστος μετάβασης από το σημείο i στο σημείο j . Με $prev(j)$ συμβολίζουμε το σημείο που προηγείται του j στο μονοπάτι p , με $next(j)$ το σημείο που έπεται του j στο μονοπάτι p και με $prev(k)$ το σημείο που προηγείται του k στο p μετά την αφαίρεση του σημείου j . Ο όρος μετά το πρώτο μείον που βρίσκεται στην παρένθεση είναι το κόστος που κερδίζουμε από την αφαίρεση του j . Ο όρος μετά το δεύτερο συν εκφράζει το κόστος που προκύπτει από την εισαγωγή του σημείου i στο μονοπάτι p . Η συνάρτηση \min εξασφαλίζει ότι η εισαγωγή του i στο p γίνεται στη θέση με το μικρότερο κόστος επιβάρυνσης. Αν η τιμή της έκφρασης (1) είναι μικρότερη ή ίση από το B (budget) τότε η εισαγωγή είναι εφικτή, διαφορετικά είναι ανέφικτη.

Για κάθε σημείο που ανήκει σε κάποια διαδρομή στο $paths_{top}$ εξετάζονται όλες οι πιθανές εναλλαγές με σημεία που ανήκουν στο $paths_{n\text{top}}$. Όταν μια πιθανή εναλλαγή αυξάνει το συνολικό όφελος (team score), η εναλλαγή γίνεται άμεσα και δεν εξετάζονται οι υπόλοιπες πιθανές εναλλαγές. Όταν για ένα σημείο στο $paths_{top}$ δεν υπάρχει υποψήφια εναλλαγή που να αυξάνει το συνολικό όφελος της λύσης, εξετάζονται εναλλαγές που μειώνουν το συνολικό όφελος κατά μια μικρή ποσότητα. Αν η μείωση αποφέρει ένα συνολικό όφελος που είναι πάνω από μια τιμή (threshold) τότε εκτελείται η εναλλαγή των σημείων. Διαφορετικά το σημείο παραμένει στη θέση του και εξετάζεται άλλο σημείο για εναλλαγή. Το όφελος της καλύτερης λύσης όπως προκύπτει από τη διαδικασία καλείται record και η επιτρεπτή μείωση του record στη διαδικασία two-point exchange καλείται deviation. Πρέπει να σημειωθεί ότι καθώς γίνονται εναλλαγές σημείων είναι πιθανό ένα μονοπάτι από το $paths_{n\text{top}}$ να αντικαταστήσει ένα μονοπάτι από το $paths_{top}$. Στο $paths_{top}$ κρατάμε πάντα τα m μονοπάτια με το μεγαλύτερο team score.

Ακολουθεί ο ψευδοκώδικας της two-point exchange.

Two-point exchange (Εναλλαγή 2 σημείων) για το TOP

Για κάθε σημείο j κάθε διαδρομής στο $paths_{top}$ **Κάνε** (A Loop)

Για κάθε σημείο i κάθε διαδρομής στο $paths_{n\text{top}}$ **Κάνε** (B Loop)

Αν η εναλλαγή των i και j είναι εφικτή και αυξάνεται το συνολικό όφελος **Τότε**

Κάνε την εναλλαγή;

Πήγαινε στην A Loop;

Αλλιώς

Best exchange = εφικτή εναλλαγή με το μεγαλύτερο συνολικό όφελος ανάμεσα στις άλλες εφικτές κινήσεις

Τέλος Αν

Τέλος Για (B Loop)

Αν το συνολικό όφελος από την Best exchange \geq record-deviation **Τότε**

Κάνε την εναλλαγή Best exchange;

Τέλος Αν

Τέλος Για (A Loop)

6.3.2.2 Μετακίνηση ενός σημείου (One point movement)

Σε αυτήν την διαδικασία μετακινούμε ένα σημείο κάθε φορά μεταξύ των μονοπατιών. Συγκεκριμένα μετακινούμε ένα σημείο i από την τρέχουσα θέση του σε ένα άλλο μονοπάτι μπροστά από κάποιο

σημείο. Αν η κίνηση είναι εφικτή και πρόκειται να αυξηθεί το συνολικό όφελος της λύσης, την εκτελούμε. Αν το συνολικό όφελος δεν αυξάνεται, αναζητούμε την κίνηση η οποία θα μειώσει λιγότερο το συνολικό όφελος.

Μια τέτοια μετακίνηση είναι πιθανό να μεταβάλλει το $paths_{top}$. Ένα σημείο μπορεί να μεταφερθεί από το $paths_{top}$ στο $paths_{ntop}$ και αντιστρόφως. Επίσης μπορούν να γίνουν μετακινήσεις μεταξύ διαδρομών που ανήκουν μόνο στο $paths_{top}$ ή μόνο στο $paths_{ntop}$. Όλες αυτές οι πιθανές μετακινήσεις εξετάζονται προκειμένου να βρεθεί το σύνολο μονοπατιών με το μεγαλύτερο συνολικό όφελος. Οι λεπτομέρειες αυτού του αλγορίθμου παρουσιάζονται στον ψευδοκώδικα που ακολουθεί. [18]

Μετακίνηση ενός σημείου (One point movement) για το TOP

Για κάθε σημείο i που ανήκει στο χώρο αναζήτησης (έλλειψη)(έστω $i \in q$ path) **Κάνε** (A loop)

Για κάθε σημείο j που ανήκει σε οποιοδήποτε μονοπάτι $p \in \{ paths_{top}, paths_{ntop} \}$ ($p \neq q$)

Κάνε (B Loop)

Αν η εισαγωγή του i μπροστά από το j είναι εφικτή και αυξάνεται το συνολικό

Όφελος **Τότε**

Κάνε την κίνηση

Πήγαινε στην A Loop

Αλλιώς

Best movement= η εφικτή κίνηση με το μεγαλύτερο συνολικό όφελος ανάμεσα στις άλλες εφικτές κινήσεις

Τέλος Αν

Τέλος Για (B Loop)

Αν το συνολικό όφελος από την Best movement \geq record-deviation **Τότε**

Εκτέλεσε Best movement;

Τέλος Αν

Τέλος Για (A Loop)

6.3.2.3 Διαδικασία Clean up (Μείωση μήκους μονοπατιών)

Με αυτήν την διαδικασία μειώνονται τα μήκη των μονοπατιών στο $paths_{top}$ με ανακατανομή των κόμβων σε κάθε μονοπάτι. Αυτό έχει ως αποτέλεσμα τα μονοπάτια να διατηρούν το ίδιο όφελος το κάθε ένα, καταναλώνοντας λιγότερο χρόνο. Αυτή η βελτίωση στα μήκη καθιστά πιο πιθανή την εισαγωγή και νέων κόμβων στα μονοπάτια (από το $paths_{ntop}$ στο $paths_{top}$) καθώς αυξάνεται το υπολειπόμενο budget ανά μονοπάτι.

Οι Chao et. al. [18] χρησιμοποιούν μια 2-opt διαδικασία και παραπέμπουν στη δουλειά του Lin 1965 [19]. Με τον όρο 2-opt σε προβλήματα βελτιστοποίησης εννοούμε έναν αλγόριθμο τοπικής αναζήτησης που εξετάζει κάθε πιθανή εναλλαγή σημείων.

6.3.3 Διαδικασίες διαταραχής της λύσης

Για να αποφύγουμε τον εγκλωβισμό της λύσης σε τοπικό βέλτιστο μέσα στο χώρο αναζήτησης, εφαρμόζουμε δυο διαδικασίες διαταραχής της λύσης. Η ιδέα είναι να αφαιρέσουμε μερικά σημεία της λύσης που προκύπτει από το βήμα της βελτίωσης (Improvement) και ξαναεκτελούμε την Improvement.

Στόχος είναι η περαιτέρω βελτίωση του συνολικού οφέλους της λύσης. Οι διαδικασίες περιγράφονται στη συνέχεια και ονομάζονται *ReInitialization I* και *ReInitialization II*.

6.3.3.1 ReInitialization I

Σε αυτήν την διαδικασία, αφαιρούμε έναν αριθμό k σημείων τα οποία έχουν το μικρότερο όφελος στο $paths_{top}$ και τα εισάγουμε στο $paths_{ntop}$. Καθώς αυξάνεται ο αριθμός των επαναλήψεων στο βήμα Improvement, αυξάνεται και ο αριθμός k των σημείων που αφαιρούνται από τα μονοπάτια στο $paths_{top}$.

6.3.3.2 ReInitialization II

Σε αυτήν την διαδικασία, επίσης αφαιρούμε k σημεία από τα μονοπάτια που ανήκουν στο $paths_{top}$ αλλά με διαφορετικό κριτήριο από τη *ReInitialization I* (στην οποία το κριτήριο ήταν το χαμηλό profit). Για να αποφασίσουμε ποια θα είναι τα k σημεία που θα αφαιρέσουμε, υπολογίζουμε για κάθε σημείο στο $paths_{top}$ το λόγο του οφέλους (profit) προς το κόστος εισαγωγής του κόμβου στη διαδρομή (το οποίο ορίστηκε παραπάνω). Επιλέγουμε τα k σημεία με το μικρότερο λόγο. Αυτά τα σημεία εισάγονται βάσει του ελάχιστου κόστους εισαγωγής στα μονοπάτια του $paths_{ntop}$. Σε αυτό το σημείο του κώδικα μειώνουμε το ποσοστό επιτρεπόμενης μείωσης του team score στο $p=2.5$. Το μέγεθος της επιτρεπόμενης μείωσης του συνολικού οφέλους της λύσης κατά την εκτέλεση του αλγορίθμου καλείται deviation. Το συνολικό όφελος της καλύτερης λύσης κατά τη διάρκεια του αλγορίθμου καλείται record. Ισχύει $deviation=p\% \cdot record$. Το ποσοστό επιτρεπόμενης μείωσης p αρχικά ισούται με 5%.

6.3.4 Ψευδοκώδικας του CGW ευρετικού αλγορίθμου

Σε αυτήν την ενότητα παρουσιάζουμε τον ψευδοκώδικα του αλγορίθμου των Chao et al. [18]

για το Team Orienteering πρόβλημα. Οι τιμές των K και I στους δυο βρόγχους του ψευδοκώδικα είναι 50 και 10 αντίστοιχα. Αυτές οι τιμές προέκυψαν μετά από εκτεταμένα πειράματα και παράγουν καλές λύσεις σε μικρό χρόνο CPU.

CGW αλγόριθμος για το TOP

Βήμα 1. Αρχικοποίηση

Εκτέλεση της αρχικοποίησης (Initialization)
 $record \leftarrow$ συνολικό όφελος της αρχικής λύσης
 $p \leftarrow 5$
 $deviation \leftarrow p\% \times record$

Βήμα 2. Βελτίωση

Για $k=1,2,\dots,K$ **Κάνε** (K Loop)

Για $i=1,2,\dots,I$ **Κάνε** (I Loop)

Εκτέλεση της εναλλαγής 2 σημείων (two-point exchange)
 Εκτέλεση της μετακίνησης ενός σημείου (one point movement)
 Εκτέλεση Clean Up

Αν δεν έγινε κάποια κίνηση παραπάνω **Τότε**

τερμάτισε το βρόγχο I Loop

Τέλος Αν

Αν προέκυψε καλύτερη λύση **Τότε**

$record \leftarrow$ συνολικό όφελος της λύσης που προέκυψε
 $deviation \leftarrow p\% \times record$

Τέλος Αν**Τέλος Για** (I Loop)**Αν** δεν προκύψει νέο record σε 5 επαναλήψεις **Τότε**

Πήγαινε στο Βήμα 3

Τέλος ΑνΕκτέλεσε την ReInitialization I (αφαίρεση k σημείων)**Τέλος Για** (K Loop)Βήμα 3 $p \leftarrow 2.5$ Εκτέλεσε την ReInitialization II (αφαίρεση K σημείων, όπου K η τιμή τερματισμού της (K Loop)

Εκτέλεσε το Βήμα 2 άλλη μια φορά

7 Team Orienteering Problem με Time Windows

Κατά αντιστοιχία με το απλό Orienteering Problem, οι κόμβοι του γραφήματος μπορούν να έχουν συγκεκριμένο χρονικό διάστημα στο οποίο μπορεί να γίνει η επίσκεψη. Αυτό το χρονικό διάστημα καλείται time window. Σε αυτήν την περίπτωση το πρόβλημα λέγεται Team Orienteering Problem with Time Windows (TOPTW). Το TOPTW έχει εφαρμογή στη χρήση ενός τουριστικού οδηγού σε μια πόλη με αξιοθέατα τα οποία θέλουν να επισκεφτούν οι τουρίστες χωρίς όμως να μπορούν να τα επισκεφτούν όλα λόγω περιορισμένου χρόνου. Ο αριθμός των διαδρομών k αντιστοιχεί στις διαθέσιμες μέρες των τουριστών και τα time windows στο ωράριο λειτουργίας των αξιοθέατων. Μια άλλη εφαρμογή του TOPTW εμφανίζεται στη δρομολόγηση των οχημάτων μιας εταιρείας η οποία θέλει να εξυπηρετήσει όσο το δυνατόν περισσότερους πελάτες (κόμβοι του γραφήματος). Ο αριθμός των οχημάτων της εταιρείας αντιστοιχεί στον αριθμό των διαφορετικών διαδρομών. Εδώ το όφελος θα μπορούσε να αντιστοιχεί στο μέγεθος της παραγγελίας κάθε πελάτη.

Αφού το OP είναι NP-hard όπως αποδείχθηκε στην παράγραφο 2.7, τότε το TOPTW δεν μπορεί να έχει βέλτιστη λύση σε πολυωνυμικό χρόνο. Για την εφαρμογή του τουριστικού οδηγού πρέπει να βρεθεί μία καλή λύση σε μερικά δευτερόλεπτα. Για το λόγο αυτό έχουν αναπτυχθεί κάποιοι ευρετικοί αλγόριθμοι (heuristics) που δίνουν μια αρκετά καλή λύση σε σύντομο χρόνο.

7.1 Ορισμός του Team Orienteering Problem με Time Windows (TOPTW)

Έστω πλήρες γράφημα $G=(V,E)$ όπου $V=\{v_0, v_1, \dots, v_{n+1}\}$ το σύνολο των κορυφών και E είναι το σύνολο των ακμών. Οι κορυφές v_0 και v_{n+1} αντιστοιχούν στο σημείο αναχώρησης και τερματισμού αντίστοιχα (μπορούν και να ταυτίζονται). Κάθε κορυφή v_i σχετίζεται με ένα όφελος p_i το οποίο λαμβάνεται με την επίσκεψη στον κόμβο. Το όφελος των κορυφών v_0 και v_{n+1} είναι μηδενικό. Σε κάθε ακμή του γραφήματος (v_i, v_j) αντιστοιχίζεται ένα κόστος c_{ij} . Ακόμα σε κάθε κορυφή v_i αντιστοιχίζεται ένας πραγματικός αριθμός u_i που εκφράζει το χρόνο που διαρκεί η επίσκεψη στο συγκεκριμένο κόμβο και ένα χρονικό παράθυρο (time window) $[R_i, D_i]$ εντός του οποίου πρέπει να ξεκινήσει η επίσκεψη στον κόμβο. Το χρονικό παράθυρο του κόμβου μπορεί να διαφέρει ανά διαδρομή [4]. Με δεδομένο έναν φυσικό αριθμό m και B το μέγιστο επιτρεπτό μήκος, το TOPTW αφορά την εύρεση m διακριτών διαδρομών $v_0 - v_{n+1}$ οι οποίες έχουν μήκος το πολύ B έτσι ώστε να μεγιστοποιείται το συνολικό όφελος (profit) και κάθε κόμβος να διαπερνάται το πολύ μία φορά χωρίς να παραβιάζεται το χρονικό παράθυρο.

7.1.1 Μαθηματική Μοντελοποίηση του Team Orienteering Problem με Time Windows (TOPTW)

Σύμφωνα με τους Vansteenwegen et. al [2], το TOPTW μπορεί να μοντελοποιηθεί ως πρόβλημα γραμμικού προγραμματισμού. Έστω $x_{ijd} = 1$ εάν στη διαδρομή d μία επίσκεψη στον κόμβο i ακολουθείται από μια επίσκεψη στον κόμβο j και 0 αλλιώς. Επίσης έστω $y_{id} = 1$ εάν ο κόμβος i περιλαμβάνεται στη διαδρομή d και 0 αλλιώς. Με s_{id} συμβολίζουμε την έναρξη της επίσκεψης στον κόμβο i στην διαδρομή d . M είναι μια μεγάλη σταθερά. Υποθέτουμε ότι οι τοποθεσίες κυμαίνονται από $1, 2, \dots, n$ όπου 1 είναι το σημείο έναρξης και n το σημείο τερματισμού. Χρησιμοποιώντας το συμβολισμό της ενότητας 7.1 έχουμε:

$$\text{Max} \sum_{d=1}^m \sum_{i=2}^{n-1} p_i y_{id} \quad (0)$$

$$\sum_{d=1}^m \sum_{j=2}^{n-1} x_{1jd} = \sum_{d=1}^m \sum_{i=2}^{n-1} x_{ind} = m \quad (1)$$

$$\sum_{i=1}^{n-1} x_{ikd} = \sum_{j=2}^n x_{kj d} = y_{kd} \quad (k=2, \dots, n-1; d=1, \dots, m) \quad (2)$$

$$s_{id} + u_i + c_{ij} - s_{jd} \leq M(1 - x_{ijd}) \quad (i, j = 1, \dots, n; d=1, \dots, m) \quad (3)$$

$$\sum_{d=1}^m y_{kd} \leq 1 \quad k=2, \dots, n-1 \quad (4)$$

$$\sum_{i=1}^{n-1} \left(u_i y_{id} + \sum_{j=2}^n c_{ij} x_{ijd} \right) \leq B \quad (d = 1, \dots, m) \quad (5)$$

$$R_i \leq s_{id} \quad (i = 1, \dots, n; d = 1, \dots, m) \quad (6)$$

$$s_{id} \leq D_i \quad (i = 1, \dots, n; d = 1, \dots, m) \quad (7)$$

$$x_{ijd}, y_{id} \in \{0, 1\} \quad (i, j = 1, \dots, n; d = 1, \dots, m) \quad (8)$$

Η αντικειμενική συνάρτηση στη σχέση (0) μεγιστοποιεί το συνολικό όφελος που συλλέγεται από τις m διαδρομές. Ο περιορισμός της σχέσης (1) εξασφαλίζει ότι κάθε διαδρομή ξεκινά από την τοποθεσία 1 και τελειώνει στην τοποθεσία n . Στις ευρετικές τεχνικές που θα παρουσιάσουμε στη συνέχεια έχουμε μια παραλλαγή του προβλήματος έτσι ώστε οι διαδρομές να αρχίζουν και να τερματίζουν στην ίδια τοποθεσία. Σε αυτήν την περίπτωση, ο περιορισμός (1) γίνεται:

$$\sum_{d=1}^m \sum_{j=2}^{n-1} x_{1jd} = \sum_{d=1}^m \sum_{i=2}^{n-1} x_{i1d} = m.$$

Οι περιορισμοί (2) και (3) εξασφαλίζουν την συνδεσιμότητα του κάθε μονοπατιού και ότι οι χρονικές στιγμές επίσκεψης στους κόμβους είναι συμβατές με την ακολουθία επίσκεψης των κόμβων στα μονοπάτια. Ο περιορισμός (4) εξασφαλίζει ότι κάθε κόμβος δέχεται επίσκεψη το πολύ μια φορά. Ο περιορισμός (5) επιβάλλει ότι κάθε μονοπάτι θα πρέπει να ικανοποιεί το χρονικό όριο (budget) B . Τέλος οι περιορισμοί (6) και (7) περιορίζουν την έναρξη της επίσκεψης (s) εντός του χρονικού παραθύρου κάθε κόμβου.

7.2 Ο αλγόριθμος Iterated Local Search (ILS)

Σε αυτήν την παράγραφο θα παρουσιάσουμε την τεχνική της επαναλαμβανόμενης τοπικής αναζήτησης και τον αλγόριθμο ILS των Vansteenwegen και συνεργατών [2] που χρησιμοποιεί αυτήν την τεχνική. Ο αλγόριθμος συνδυάζει ένα βήμα εισαγωγής (insertion step) και ένα βήμα διαταραχής (shake step) με το οποίο απεγκλωβίζεται από τα τοπικά μέγιστα. Σε γενικές γραμμές η τεχνική της επαναλαμβανόμενης τοπικής αναζήτησης ξεκινά από μια αρχική λύση και εισάγει κόμβους στη λύση μέχρι να βρεθεί μια τοπικά βέλτιστη λύση. Στη συνέχεια εκτελείται το βήμα διαταραχής όπου αφαιρείται ένας αριθμός κόμβων από κάθε διαδρομή της λύσης και ξαναεκτελείται το βήμα εισαγωγής στη λύση που προέκυψε. Η νέα τοπικά βέλτιστη λύση συγκρίνεται με την προηγούμενη ως προς το όφελος. Ο αλγόριθμος κρατά τη λύση με το μεγαλύτερο όφελος. Η διαδικασία θα τερματιστεί όταν ικανοποιηθεί μια συνθήκη. Η συνθήκη που χρησιμοποιείται συνήθως είναι όταν σε ένα συγκεκριμένο αριθμό επαναλήψεων δεν προκύπτει καλύτερη λύση.

7.2.1 Βήμα Εισαγωγής (Insertion Step)

Στο βήμα εισαγωγής εισάγεται στη λύση ένας νέος κόμβος που δεν ανήκει σε αυτήν. Για να είναι εφικτή μια εισαγωγή θα πρέπει να εξασφαλιστεί ότι δεν παραβιάζονται τα χρονικά παράθυρα των κόμβων που έπονται της εισαγωγής. Για να γίνει αποδοτικά η εκτίμηση για το κατά πόσο είναι εφικτή μια εισαγωγή αποθηκεύονται κάποιες επιπλέον πληροφορίες για κάθε κόμβο που ανήκει στη λύση. Αυτές είναι το Wait και το MaxShift. Το Wait εκφράζει το χρόνο αναμονής μέχρι να αρχίσει η επίσκεψη στον κόμβο. Αν η άφιξη στον κόμβο γίνει μετά το χρόνο έναρξης R_i του κόμβου, το Wait είναι μηδέν. Αλλιώς το Wait είναι η διαφορά της έναρξης του χρονικού παραθύρου με το χρόνο άφιξης. Συνεπώς συμβολίζοντας το χρόνο άφιξης σε έναν κόμβο i με $arrive_i$, ο χρόνος αναμονής στον κόμβο ορίζεται ως $Wait_i = \max(0, R_i - arrive_i)$. Το Maxshift ορίζεται ως ο μέγιστος χρόνος που μπορεί να παραταθεί η έναρξη της επίσκεψης χωρίς να παραβιάζονται οι χρονικοί περιορισμοί της διαδρομής. Το MaxShift υπολογίζεται αναδρομικά ως εξής: για τον τελευταίο κόμβο v_{n+1} πρέπει $MaxShift_{n+1} = B - arrive_{n+1}$ όπου B το χρονικό όριο της διαδρομής και για κάθε κόμβο i ισχύει $MaxShift_i = \min(D_i - s_i, Wait_{i+1} + MaxShift_{i+1})$ όπου s_i ο χρόνος έναρξης της επίσκεψης.

Η αύξηση του συνολικού χρόνου της διαδρομής από την εισαγωγή ενός κόμβου j σε κάποια διαδρομή (συμβολίζουμε με $Shift_j$) μεταξύ των κόμβων i και k ορίζεται ως

$$Shift_j = c_{ij} + Wait_j + u_j + c_{jk} - c_{ik}.$$

Για να είναι εφικτή η εισαγωγή του κόμβου j μεταξύ των i και k πρέπει να ικανοποιείται η συνθήκη $Shift_j \leq Wait_k + MaxShift_k$. Επίσης θα πρέπει να εξασφαλίζεται ότι η επίσκεψη στον κόμβο j γίνεται εντός του time window του j . Δηλαδή πρέπει $s_i + u_i + c_{ij} \leq D_j$. Για κάθε κόμβο που δεν ανήκει σε κάποια διαδρομή της λύσης εξετάζεται η κάθε πιθανή θέση εισαγωγής στη λύση και υπολογίζεται η θέση με το μικρότερο shift. Στη συνέχεια για κάθε υποψήφιο σημείο προς εισαγωγή υπολογίζεται ένας λόγος (ratio) που είναι το πηλίκο του τετραγώνου του οφέλους του σημείου προς το shift. Δηλαδή $ratio_i = p_i^2 / shift_i$. Το σημείο με το μεγαλύτερο ratio είναι αυτό που θα εισαχθεί στη λύση. Μετά την εισαγωγή πρέπει να ενημερωθούν οι μεταβλητές των υπολοίπων κόμβων της διαδρομής. Για τους κόμβους που έπονται της εισαγωγής πρέπει να ενημερωθούν το Wait, το arrive, ο χρόνος έναρξης της επίσκεψης (s) και το MaxShift. Για τους κόμβους που προηγούνται της εισαγωγής πρέπει να ενημερωθεί το MaxShift.

Ο ψευδοκώδικας για το βήμα εισαγωγής του αλγορίθμου ILS δίνεται παρακάτω:

Ψευδοκώδικας βήματος **Εισαγωγής** (Insertion Step)

Για κάθε κόμβο που δεν ανήκει στη λύση

Βρες την καλύτερη πιθανή θέση εισαγωγής με το μικρότερο shift
Υπολόγισε το ratio

Τέλος Για

Εισήγαγε τον κόμβο με το μεγαλύτερο ratio (έστω j αυτός ο κόμβος)στην αντίστοιχη θέση
Υπολόγισε τα wait, arrive και start του j

Για κάθε κόμβο μετά τον j

Ενημέρωσε τα wait, arrive, start και MaxShift

Τέλος Για

Ενημέρωσε το MaxShift του j

Για κάθε κόμβο πριν τον j

Ενημέρωσε το MaxShift

Τέλος Για

7.2.2 Βήμα Διαταραχής (Shake Step)

Προκειμένου ο αλγόριθμος να μην εγκλωβιστεί σε τοπικά μέγιστα εκτελείται το βήμα διαταραχής. Σε αυτό το βήμα αφαιρούνται ένας ή περισσότεροι κόμβοι από κάθε διαδρομή. Κάθε βήμα διαταραχής παίρνει ως είσοδο δύο ακεραίους. Ο ένας εκφράζει τον αριθμό των συνεχόμενων κόμβων που θα αφαιρεθούν (R) και ο άλλος τη θέση στη διαδρομή από την οποία θα ξεκινήσει η αφαίρεση (S). Αν φτάσουμε στον τελικό κόμβο η αφαίρεση συνεχίζεται μετά από τον αρχικό.

7.2.3 Ο ευρετικός αλγόριθμος

Ο αλγόριθμος ξεκινά αρχικοποιώντας όλες τις παραμέτρους του shake σε 1 και δημιουργεί ένα σύνολο άδειων διαδρομών. Ο αριθμός επαναλήψεων του insert και του shake χωρίς να έχουμε βελτίωση της λύσης είναι 150. Το μέγιστο πλήθος κόμβων που μπορούν να αφαιρεθούν από μια διαδρομή ορίζεται $N/3K$ όπου N το πλήθος των κόμβων της τοπολογίας και K ο αριθμός των διαδρομών. Αρχικά εφαρμόζεται το βήμα εισαγωγής μέχρι να βρεθεί ένα τοπικό μέγιστο. Αν η λύση είναι καλύτερη από αυτή που υπάρχει μέχρι εκείνη τη στιγμή, αυτή καταγράφεται και η μεταβλητή R γίνεται ίση με ένα για το βήμα διαταραχής. Στη συνέχεια εκτελείται το βήμα διαταραχής. Μετά από κάθε τέτοιο βήμα προστίθεται η τιμή της R στη μεταβλητή S και η R αυξάνεται κατά ένα για το επόμενο shake. Αν η S είναι μεγαλύτερη ή ίση από το μέγεθος της μικρότερης διαδρομής τότε μειώνεται όσο είναι το μέγεθος αυτό. Αν η R γίνει ίση με $N/3K$ τότε τίθεται ίση με 1.

Ακολουθεί ο ψευδοκώδικας του ILS :

Ψευδοκώδικας του ILS ευρετικού Αλγόριθμου

$S \leftarrow 1;$

$R \leftarrow 1;$

numberOfTimesNotImproved $\leftarrow 0;$

BestFoundSolution $\leftarrow \text{null};$

Ενόσω numberOfTimesNotImproved < 150 **Κάνε**

| **Ενόσω** η λύση δεν είναι τοπικά βέλτιστη **Κάνε**

| | **Insertion Step**

| **Τέλος Ενόσω**

| **Αν** η Λύση που βρέθηκε είναι καλύτερη από τη BestFoundSolution **Τότε**

BestFoundSolution \leftarrow Λύση;

$R \leftarrow 1;$

| numberOfTimesNotImproved $\leftarrow 0;$

Αλλιώς

numberOfTimesNotImproved ← numberOfTimesNotImproved + 1;

Τέλος Αν

Shake(R,S);

S ← S + R;

R ← R + 1;

Αν S > του μήκους της μικρότερης διαδρομής **Τότε**

S ← S - μήκος μικρότερης διαδρομής;

Τέλος Αν

| **Αν** R == N/3M **Τότε**

R ← 1;

| **Τέλος Αν**

Τέλος Ενώσω

Επέστρεψε την BestFoundSolution;

7.3 Ο αλγόριθμος Average Slack

Ο αλγόριθμος Average Slack βασίζεται στον Slack CSCRoutes για χρονοεξαρτώμενα (time Dependent)¹¹ στιγμιότυπα [3]. Ο Average Slack είναι προσαρμοσμένος σε στιγμιότυπα που δεν είναι χρονοεξαρτώμενα. Αυτό σημαίνει ότι ο χρόνος μεταφοράς από τον έναν κόμβο στον άλλο είναι σταθερός και δε μεταβάλλεται ανάλογα με το χρόνο αναχώρησης (leave) από τον πρώτο κόμβο. Ακόμα δε χρησιμοποιείται η τεχνική clustering, randomization και αντικατάστασης (replace).

Ο Average Slack χρησιμοποιεί ένα καθολικό κριτήριο για την εισαγωγή ενός κόμβου. Αυτό το κριτήριο λαμβάνει υπόψη την επίδραση μιας πιθανής εισαγωγής σε μια διαδρομή σε όλους τους κόμβους της διαδρομής. Ορίζεται μια επιπλέον μεταβλητή για κάθε κόμβο i σε μια διαδρομή ως εξής:

$slack_i = MaxStart_i - arrive_i$. Το $MaxStart_i$ συμβολίζει το μεγαλύτερο χρόνο στον οποίο μπορεί να αρχίσει η επίσκεψη στον κόμβο i χωρίς να παραβιάζονται τα χρονικά παράθυρα των κόμβων που έπονται του i .

Ισχύει $MaxStart_i = s_i + MaxShift_i$. Το $MaxStart_i$ ενός κόμβου i ορίζεται αναδρομικά ως εξής:

$MaxStart_i = \min\{D_i, MaxStart_{next(i)} - C_{inext(i)} - u_i\}$ όπου με $next(i)$ συμβολίζουμε τον επόμενο κόμβο από τον i στη διαδρομή. Για τον τελικό κόμβο της διαδρομής ισχύει $MaxStart_{n+1} = D_{n+1}$ δηλαδή το κλείσιμο του χρονικού παραθύρου στο συγκεκριμένο κόμβο.

Παρατηρούμε ότι αν η μεταβλητή $slack_i$ είναι κοντά στο μηδέν, είναι εξαιρετικά απίθανο να υπάρχει μια εφικτή εισαγωγή ενός σημείου μεταξύ των κόμβων $prev(i)$ και i (όπου με $prev(i)$ συμβολίζουμε τον προηγούμενο κόμβο του i στη διαδρομή).

Έστω v_1, v_2, \dots, v_n οι κόμβοι μιας διαδρομής r με v_1 αρχικό κόμβο της διαδρομής και v_n τον τελικό κόμβο της διαδρομής. Έστω επίσης v_k ένας υποψήφιος κόμβος προς εισαγωγή μεταξύ των σημείων v_i και v_{i+1} της διαδρομής r . Η εισαγωγή του v_k πιθανότατα θα καθυστερήσει το χρόνο άφιξης (arrive) στους κόμβους από v_{i+1} έως v_n γιατί προστίθεται ο χρόνος ταξιδιού στον v_k καθώς και ο χρόνος επίσκεψης στο συγκεκριμένο κόμβο που είναι u_k . Επίσης θα μεταβληθεί το $MaxStart$ των κόμβων που προηγούνται της εισαγωγής. Συνεπώς μεταβάλλονται οι τιμές $slack$ όλων των κόμβων της διαδρομής r .

¹¹ Τα χρονοεξαρτώμενα στιγμιότυπα είναι γραφήματα στα οποία το κόστος μετάβασης από τον ένα κόμβο στον άλλο μεταβάλλεται ανάλογα με το χρόνο αναχώρησης από τον πρώτο κόμβο. Αυτού του είδους τα στιγμιότυπα μοντελοποιούν την πολυπλοκότητα των μέσων μεταφοράς

Ορίζουμε την ποσότητα AvSlack ως το μέσο όρο των slack κάθε κόμβου μετά από μια πιθανή εισαγωγή του v_k στη διαδρομή r . Συγκεκριμένα έστω $slack_j^k$ $j = i + 1, \dots, n$ οι νέες τιμές των slack που μεταβάλλονται λόγω της αύξησης των χρόνων άφιξης στους κόμβους που έπονται της εισαγωγής. Επίσης μεταβάλλονται τα $slack_j^k$ $j = 1, \dots, i$ λόγω της μεταβολής του MaxStart στους κόμβους που προηγούνται της εισαγωγής. Ισχύει

$$AvSlack_k^i = \frac{\sum_{j=1}^i slack_j^k + \sum_{j=i+1}^n slack_j^k + slack_k}{n + 1}$$

Αν η ποσότητα AvSlack είναι μεγάλη, αυτό σημαίνει ότι ακόμα και μετά την εισαγωγή του κόμβου v_k υπάρχει μεγάλη πιθανότητα εισαγωγής και άλλων κόμβων στη διαδρομή τόσο πριν όσο και μετά από τον v_k . Στη συνέχεια για κάθε σημείο v_k που δεν περιλαμβάνεται στη λύση υπολογίζουμε το AvSlack για κάθε πιθανή θέση εισαγωγής. Έστω $maxAvSlack_k$ η μέγιστη τιμή AvSlack για το σημείο v_k ανάμεσα σε όλες τις πιθανές θέσεις. Για να αποφασίσουμε ποιο σημείο θα εισαχθεί τελικά υπολογίζουμε το

$slackWeight_k = profit_k^2 \cdot maxAvSlack_k$. Εισάγουμε το σημείο με τη μεγαλύτερη ποσότητα slackWeight στη θέση που προέκυψε το $maxAvSlack$.

Η ουσιαστική διαφορά του Average Slack από τον ILS είναι το διαφορετικό κριτήριο στο βήμα εισαγωγής.

7.4 Ο Υβριδικός αλγόριθμος GRASP-ELS για το TOPTW

Σε αυτήν την ενότητα παρουσιάζεται ένας ευρετικός αλγόριθμος για την επίλυση του Team Orienteering προβλήματος με time windows, που συνδυάζει την τεχνική Greedy Randomized Adaptive Search Procedure (GRASP) με την εξελικτική μέθοδο τοπικής αναζήτησης (ELS). Αυτή η μέθοδος παρουσιάστηκε από τους Labadie et al. [20]. Συγκρινόμενη με τον ILS των Vansteenwegen et. al., αυτή η μέθοδος παράγει λύσεις καλύτερης ποιότητας (δηλ. λύσεις με μεγαλύτερο συνολικό όφελος), έχοντας όμως μεγάλο υπολογιστικό χρόνο.

Για να παραχθούν αρχικές λύσεις προτείνονται πέντε κατασκευαστικοί ευρετικοί αλγόριθμοι. Οι τρεις από αυτούς (Insertion) προσπαθούν να εισάγουν μια κορυφή στη λύση σε κάθε επανάληψη. Διαφέρουν μεταξύ τους ως προς το κριτήριο με το οποίο επιλέγεται η καλύτερη εισαγωγή. Οι άλλοι δυο (sweep) δημιουργούν συστοιχίες (clusters) από κορυφές και κατασκευάζουν μια διαδρομή (tour) για κάθε συστοιχία. Μια κρίσιμη κίνηση και για τους πέντε κατασκευαστικούς αλγόριθμους είναι το βήμα εισαγωγής. Συνεπώς θα πρέπει σε κάθε βήμα εισαγωγής να ικανοποιούνται οι περιορισμοί από τα time windows και να μην παραβιάζεται το μέγιστο κόστος της διαδρομής (budget).

Οι περιορισμοί που πρέπει να ικανοποιούνται για να είναι εφικτή μια εισαγωγή αναφέρθηκαν στην παράγραφο 7.2.1 όταν περιγράφαμε το βήμα εισαγωγής για τον ILS. Αυτές είναι $Shift_j \leq Wait_k +$

$MaxShift_k$ (1) και $s_i + u_i + c_{ij} \leq D_j$ (2).

Στη συνέχεια περιγράφονται οι πέντε κατασκευαστικοί ευρετικοί αλγόριθμοι.

7.4.1 Insertion Heuristics

Οι ευρετικοί αλγόριθμοι εισαγωγής (Insertion Heuristics) κατασκευάζουν μια εφικτή λύση εισάγοντας μια κορυφή. Και σε αυτόν τον αλγόριθμο λαμβάνονται υπόψη το όφελος και η αύξηση του κόστους από την πιθανή εισαγωγή μιας κορυφής. Ωστόσο το κριτήριο εισαγωγής διαφοροποιείται από το κριτήριο του ILS στο ότι εξετάζεται και ο αριθμός των γειτονικών κόμβων κάθε κορυφής προς εισαγωγή. Όσο μεγαλύτερος είναι ο αριθμός των γειτονικών κόμβων μιας κορυφής τόσο αυξάνεται η πιθανότητα να εισαχθούν και νέοι κόμβοι στη λύση στην επόμενη επανάληψη.

Έστω ότι θέλουμε να εισάγουμε τον κόμβο v_j μετά τον κόμβο v_i σε μια διαδρομή και έστω $shift_j^i$ η χρονική επιβάρυνση στη διαδρομή από την εισαγωγή του v_j μετά τον κόμβο v_i όπως αυτή ορίστηκε στην παράγραφο 7.2.1. Ορίζουμε A_i το σύνολο των κορυφών που είναι προσβάσιμες από την κορυφή v_i δηλαδή το σύνολο των κορυφών v_j για τις οποίες ισχύει $s_i + u_i + c_{ij} \leq D_j$ και q_i η πληθικότητα του A_i (δηλ. q_i ο αριθμός των στοιχείων του A_i). Το κριτήριο (ratio) για την καλύτερη εισαγωγή υπολογίζεται από την ισότητα:

$$ratio_j^i = q_j \cdot \frac{p_j}{shift_j^i} \quad (3) \text{ όπου } p_j \text{ το όφελος του κόμβου.}$$

Η εισαγωγή κορυφής με τη μεγαλύτερη τιμή ratio έναντι κάθε πιθανής εισαγωγής εκτελείται. Στις πρώτες επαναλήψεις της μεθόδου ο όρος q_j επικρατεί του οφέλους. Η λύση αποτελείται από κορυφές με χαμηλότερη έναρξη στο time window R_i και από εκείνες τις κορυφές που έχουν μεγαλύτερη τιμή q_j . Όταν οι διαδρομές περιέχουν περισσότερες κορυφές, το σημαντικότερο κριτήριο γίνεται το profit. Η λύση θα μπορούσε να αρχικοποιηθεί με κάποιες από τις πιο κερδοφόρες κορυφές για να αποφευχθεί η επικράτηση του q_j .

Οι Labadie et al. [20] επέλεξαν το κριτήριο της σχέσης (3) έναντι του $\frac{p_j^2}{shift_j}$ του Vansteenwegen γιατί

παράγει λίγο καλύτερα αποτελέσματα σε κάποια δύσκολα στιγμιότυπα.

Οι τρεις κατασκευαστικοί αλγόριθμοι εισαγωγής περιγράφονται στη συνέχεια:

Heuristic Insert (I): Αυτή η μέθοδος κατασκευάζει μια εφικτή λύση με διαδοχικές εισαγωγές μιας κορυφής που δεν ανήκει στη λύση, στην τελευταία θέση της διαδρομής. Ξεκινώντας από την πρώτη διαδρομή $r=1$, ψάχνει για την καλύτερη εισαγωγή ανάμεσα σε όλες τις κορυφές που δεν έχουν συμπεριληφθεί στη λύση μέχρι να μην υπάρχει καμία εφικτή εισαγωγή. Μετά εκτελεί την ίδια διαδικασία στην επόμενη διαδρομή $r=2$, μέχρι να φτάσει στην τελευταία $r=m$. Η κορυφή v_j με το μεγαλύτερο $ratio_j$ της σχέσης (3) εισάγεται μετά από τον αντίστοιχο κόμβο v_i . Η πολυπλοκότητα του χώρου αναζήτησης είναι $O(n^2)$. Αρχικά κάθε διαδρομή περιέχει 1 κόμβο και κατά τη διάρκεια της διαδικασίας ο αριθμός των κόμβων στη διαδρομή γίνεται $O(n)$. Άρα έχουμε $O(n)$ βήματα. Σε κάθε βήμα, οι κορυφές που είναι εκτός διαδρομής είναι επίσης $O(n)$ και τις εξετάζουμε όλες για να αποφασίσουμε ποια θα μπει στην τελευταία θέση. Άρα η πολυπλοκότητα είναι $O(n^2)$.

Heuristic Insert (II): Αυτή η μέθοδος ευνοεί τις κορυφές με μεγαλύτερο profit για εισαγωγή στις διαδρομές. Αρχικά όλες οι κορυφές ταξινομούνται σε μη αύξουσα σειρά με βάση το profit. Μετά οι m διαδρομές αρχικοποιούνται με τις m κορυφές με το μεγαλύτερο profit. Στη συνέχεια κάθε διαδρομή συμπληρώνεται με άλλες κορυφές. Η διαδικασία είναι η εξής: για κάθε κορυφή που δεν περιλαμβάνεται στη λύση εξετάζονται όλες οι πιθανές θέσεις εισαγωγής σε όλες τις διαδρομές. Η καλύτερη εισαγωγή κρίνεται με βάση το λόγο της σχέσης (3). Η διαδικασία σταματά όταν δεν υπάρχει καμία εφικτή εισαγωγή.

Heuristic Insert (III): Σε αυτήν την μέθοδο γίνεται πάλι ταξινόμηση των κορυφών σε μη αύξουσα σειρά σύμφωνα με το profit. Σε κάθε επανάληψη εξετάζεται η κορυφή που βρίσκεται πρώτη στην ταξινομημένη λίστα και βρίσκεται η καλύτερη θέση εισαγωγής σε όλες τις διαδρομές με βάση την χρονική επιβάρυνση (shift) από την ενδεχόμενη εισαγωγή. Αν δεν υπάρχει κάποια εφικτή εισαγωγή για την τρέχουσα κορυφή, εξετάζεται η επόμενη κορυφή που είναι στη λίστα. Η διαδικασία εκτελείται μέχρι να εκτιμηθούν όλες οι κορυφές.

Για τα Heuristic Insert(II) και Heuristic Insert(III) το μέγεθος του χώρου αναζήτησης είναι $O(n^3)$. Αρχικά οι διαδρομές έχουν 1 κόμβο και φτάνουν σε $O(n)$. Άρα έχουμε $O(n)$ βήματα σε καθένα από τα οποία: για κάθε κορυφή γίνονται $O(n)$ συγκρίσεις για να καθοριστεί το καλύτερο σημείο εισαγωγής στη λύση και ο αριθμός των κορυφών εκτός της λύσης είναι $O(n)$. Έτσι προκύπτει η πολυπλοκότητα $O(n^3)$.

7.4.2 Sweep Heuristics

Σε αυτήν την κατηγορία κατασκευαστικών heuristics, οι κορυφές ταξινομούνται σε μη φθίνουσα σειρά σύμφωνα με την πολική γωνία. (Ο χώρος αναζήτησης είναι ο Ευκλείδειος χώρος). Η πολική γωνία θ ορίζεται βάσει των καρτεσιανών συντεταγμένων x και y . Συγκεκριμένα $\theta = \text{atan2}(y,x)$. Η συνάρτηση atan2 είναι παραλλαγή της συνάρτησης arctan που είναι η αντίστροφη της εφαπτομένης \tan . Η atan2 ορίζεται ως εξής:

- $\arctan\left(\frac{y}{x}\right)$ αν $x > 0$
- $\arctan\left(\frac{y}{x}\right) + \pi$ αν $x < 0$ και $y \geq 0$
- $\arctan\left(\frac{y}{x}\right) - \pi$ αν $x < 0$ και $y < 0$
- $\frac{\pi}{2}$ αν $x = 0$ και $y > 0$
- $-\frac{\pi}{2}$ αν $x = 0$ και $y < 0$
- Δεν ορίζεται αν $x = 0$ και $y = 0$

Στη συνέχεια η ταξινομημένη λίστα χωρίζεται σε m σταθερά διαστήματα (όσες και οι διαδρομές) I_r με $r = 1, 2, \dots, m$. Άρα κάθε διάστημα περιέχει τουλάχιστον $\left\lfloor \frac{n}{m} \right\rfloor$ κορυφές που μπορεί να επιλεγθούν για να δημιουργηθεί μια διαδρομή.

Heuristic Sweep (I): Αρχικά κάθε διαδρομή της λύσης αρχικοποιείται με την κορυφή που έχει το μεγαλύτερο όφελος στο αντίστοιχο διάστημα I_r . Τότε ο αλγόριθμος εξετάζει τις υπόλοιπες κορυφές στο τρέχον διάστημα I_r και αναζητά την καλύτερη εφικτή εισαγωγή στη διαδρομή. Για κάθε κορυφή που δεν συμπεριλαμβάνεται στη διαδρομή και ανήκει στο I_r , υπολογίζεται το $ratio_j^i$ της σχέσης (3) που ορίστηκε στην ενότητα 7.4.1. Η κορυφή με το μεγαλύτερο ratio εισάγεται στη διαδρομή. Αν δεν υπάρχει καμία εφικτή εισαγωγή για το τρέχον διάστημα, η διαδικασία αρχίζει να δημιουργεί την επόμενη διαδρομή μέχρι να παραχθούν m διαδρομές.

Heuristic Sweep (II): Η δεύτερη παραλλαγή του προτεινόμενου sweep heuristic προσπαθεί να εισάγει κορυφές με το μεγαλύτερο όφελος εντός του τρέχοντος διαστήματος I_r . Οι κορυφές κάθε διαστήματος I_r ταξινομούνται σε μη αύξουσα σειρά σύμφωνα με το όφελος. Αρχίζοντας από την κορυφή της ταξινομημένης λίστας, η διαδικασία εκτιμά την καλύτερη εφικτή εισαγωγή στην τρέχουσα διαδρομή βασιζόμενη στην αύξηση του κόστους (shift) από μια πιθανή εισαγωγή. Προφανώς η καλύτερη θέση εισαγωγής θα έχει το μικρότερο shift. Αν υπάρχει μια τέτοια εφικτή εισαγωγή, αυτή εκτελείται. Αλλιώς εξετάζεται η επόμενη κορυφή από τη λίστα.

Κάθε διάστημα I_r περιέχει το πολύ $\left\lceil \frac{n}{m} \right\rceil$ κορυφές. Συνεπώς για κάθε διάστημα I_r , και οι δυο

διαδικασίες Sweep πρέπει να εκτιμήσουν $O(n^3)$ εισαγωγές. Οι δυο διαδικασίες Sweep διαφέρουν μεταξύ τους μόνο ως προς το κριτήριο εισαγωγής. Αρχικά κάθε διαδρομή έχει 1 κόμβο και φτάνει σε $O(n)$. Άρα έχουμε $O(n)$ βήματα, σε καθένα από τα οποία: για κάθε κορυφή γίνονται $O(n)$ συγκρίσεις για να βρεθεί ποια είναι η πιο καλή θέση για να συμπεριληφθεί στη διαδρομή και ο αριθμός των κορυφών είναι $O(n)$. Έτσι προκύπτει η πολυπλοκότητα $O(n^3)$. Ο αριθμός των διαδρομών είναι σταθερός και δεν επηρεάζει την πολυπλοκότητα.

7.4.3 Ο αλγόριθμος GRASP-ELS

Οι πέντε κατασκευαστικοί αλγόριθμοι που παρουσιάστηκαν στις δυο προηγούμενες ενότητες χρησιμοποιούνται από το σχήμα GRASP προκειμένου να παραχθούν διαφορετικές λύσεις έναρξης για τη διαδικασία ELS. Υλοποιείται μια τυχαιοκρατική εκδοχή κάθε αλγορίθμου. Η τυχαιότητα καθορίζεται από μια παράμετρο p που συμβολίζει τον αριθμό εφικτών κινήσεων, οι οποίες αποθηκεύονται σε μια λίστα candidates σε κάθε επανάληψη. Η κίνηση που εκτελείται επιλέγεται τυχαία από τη λίστα candidates.

Ο ELS (Evolutionary Local Search) αποτελεί μια επέκταση του ILS. Ο ILS ξεκινά με μια λύση s που προκύπτει από κάποιο heuristic και δημιουργεί αρκετές προκύπτουσες (child) λύσεις εφαρμόζοντας διαταραχή στη λύση s . Οι προκύπτουσες λύσεις βελτιώνονται με τοπική αναζήτηση. Ο ELS επιπλέον δημιουργεί πολλαπλά αντίγραφα του s και εφαρμόζει τον ILS σε κάθε αντίγραφο.

Η διαδικασία διαταραχής είναι ένα κρίσιμο συστατικό στη μέθοδο. Όταν είναι ισχυρή, η μέθοδος συμπεριφέρεται ως ένας τυχαιοκρατικός ευρετικός αλγόριθμος. Αντίθετα όταν είναι αδύναμη οι λύσεις παγιδεύονται σε τοπικό βέλτιστο. Για να ελεγχθεί η διαταραχή χρησιμοποιείται μια παράμετρος p με εύρος τιμών στο διάστημα $[p_{\min}, p_{\max}]$.

Το γενικό πλαίσιο του GRASP-ELS όπως παρουσιάστηκε από τους Labadie et al. [20] είναι το εξής: Σε κάθε επανάληψη η GRASP παράγει μια αρχική λύση \bar{s} . Εφαρμόζεται στην \bar{s} μια διαδικασία τοπικής αναζήτησης και η λύση s που προκύπτει εισάγεται στο μηχανισμό ELS. Ο ELS δημιουργεί διάφορες προκύπτουσες (child) λύσεις για κάθε αντίγραφο της s , εφαρμόζοντας τυχαίες μεταλλάξεις και στη συνέχεια εκτελεί μια τοπική αναζήτηση. Η καλύτερη προκύπτουσα (child) λύση s' καταγράφεται και αν είναι καλύτερη από την s τότε η s' αντικαθιστά την s στην επόμενη επανάληψη του ELS. Αλλιώς ο ELS εκτελεί την επόμενη επανάληψη έχοντας ως αρχική λύση ξανά την s . Ο GRASP-ELS καθορίζεται από τρεις παραμέτρους που ορίζουν αντίστοιχα: τον αριθμό των προκυπτουσών λύσεων nc που παράγονται σε μια επανάληψη του ELS, τον μέγιστο αριθμό επαναλήψεων του ELS ni και τέλος τον αριθμό ns των λύσεων έναρξης που παράγονται από τον αλγόριθμο GRASP. Στη συνέχεια παρουσιάζεται η γενική δομή του GRASP-ELS των Labadie et al. [20].

GRASP-ELS meta-heuristic

Παράμετροι:

p : παράμετρος διαταραχής

k_{\max} : μέγιστο μέγεθος από γειτονίες στην τοπική αναζήτηση

ns : μέγιστος αριθμός επαναλήψεων GRASP

ni : μέγιστος αριθμός επαναλήψεων ELS

nc : μέγιστος αριθμός παραγόμενων child λύσεων

1 $f(s^*) \leftarrow 0$

2 **Για** i από 1 έως ns **Κάνε**

```

3   Αν  $i=1$  Τότε
4      $\bar{s} \leftarrow$  αρχική λύση (η καλύτερη που προκύπτει από τους 5 κατασκευαστικούς)
5   Αλλιώς
6      $\bar{s} \leftarrow$  τυχαία αρχική λύση
7   Τέλος Αν
8    $s \leftarrow$  Variable Neighborhood Descent ( $\bar{s}, k_{\max}$ )
9   Αν  $f(s) > f(s^*)$  Τότε  $s^* \leftarrow s$ 
10   $p \leftarrow p_{\min}$ 

11  Για  $j$  από 1 έως  $n_i$  Κάνε
12     $f(s') \leftarrow 0$ 

13  Για  $c$  από 1 έως  $n_c$  Κάνε

14     $s'' \leftarrow$  Perturbation( $s, p$ )
15     $\hat{s} \leftarrow$  Variable Neighborhood Descent ( $s'', k_{\max}$ )
16    Αν  $f(\hat{s}) > f(s')$  Τότε
17       $s' \leftarrow \hat{s}$ 
18    Τέλος Αν
19  Τέλος Για
20  Αν  $f(s') > f(s)$  Τότε
21     $s \leftarrow s'$ 
22     $p \leftarrow p_{\min}$ 
23  Αλλιώς
24     $p \leftarrow \min\{p_{\max}, p+1\}$ 
25  Τέλος Αν
26  Τέλος Για
27  Αν  $f(s) > f(s^*)$  Τότε  $s^* \leftarrow s$ 
28 Τέλος Για
29 Επέστρεψε τη λύση  $s^*$ 

```

Ο εξωτερικός βρόγχος (γραμμές 2-28) ελέγχει την GRASP. Η λύση έναρξης είναι η καλύτερη λύση που προκύπτει από τους πέντε κατασκευαστικούς αλγόριθμους (από τη δεύτερη επανάληψη υπάρχει τυχαιότητα). Ο μεσαίος βρόγχος (γραμμές 11-26) τρέχει το μηχανισμό ELS. Η παράμετρος διαταραχής p προσαρμόζεται μέσα σε αυτόν το βρόγχο ανάλογα με τα αποτελέσματα της αναζήτησης. Οι προκύπτουσες (child) λύσεις δημιουργούνται στον εσωτερικό βρόγχο στις γραμμές 13-19. Κάθε αντίγραφο της λύσης s (που είναι η λύση που μπαίνει στο μηχανισμό ELS από τη GRASP) αρχικά διαταράσσεται από την perturbation και μετά βελτιώνεται από τη Variable Neighborhood Descent, που είναι μια διαδικασία τοπικής αναζήτησης. Η καλύτερη από τις προκύπτουσες λύσεις καταγράφεται. Έξω από τον εσωτερικό βρόγχο, αν η καλύτερη child λύση s' βελτιώνει την τρέχουσα λύση s , η παράμετρος διαταραχής p παίρνει τη μικρότερη τιμή της και η λύση s αναβαθμίζεται. Διαφορετικά η p αυξάνεται και η s παραμένει αμετάβλητη. Ο αλγόριθμος σταματά μετά από n_s επαναλήψεις της GRASP.

7.4.3.1 Διαδικασία Variable Neighborhood Descent

Η διαδικασία Variable Neighborhood Descent (VND) είναι μια μέθοδος τοπικής αναζήτησης η οποία βασίζεται στην αρχή της συστηματικής αλλαγής της γειτονιάς κατά τη διάρκεια της αναζήτησης. Στη VND η εναλλαγή της γειτονιάς γίνεται με ντετερμινιστικό τρόπο. Κάθε γειτονιά έχει διαφορετική δομή από τις υπόλοιπες. Στον αλγόριθμο των Labadie et al. [20] η τοπική αναζήτηση γίνεται με VND καθώς γίνεται εναλλαγή μεταξύ δυο γειτονιών.

Η πρώτη γειτονιά περιέχει κλασικές κινήσεις που μπορεί να περιλαμβάνουν μια ή δυο διαφορετικές διαδρομές. Οι κινήσεις αυτές είναι:

1. 2-opt μετακινεί και αντικαθιστά δυο ακμές σε μια διαδρομή και αναδιατάσσει τις κορυφές.
2. Or-opt μετακινεί μια κορυφή σε άλλη θέση σε μια διαδρομή
3. 2-opt* εναλλάσσει δυο υπο-μονοπάτια μεταξύ δυο διαδρομών
4. Exchange εναλλάσσει δυο κορυφές

Αυτές οι κινήσεις έχουν στόχο να μειώσουν τα μήκη των διαδρομών. Όταν ο αριθμός των διαδρομών της λύσης m είναι μεγαλύτερος ή ίσος του 2 όλες οι παραπάνω κινήσεις εφαρμόζονται. Για $m=1$ εκτελούνται μόνο οι 2-opt, Or-opt και Exchange. Σε αυτή τη γειτονιά εξετάζονται οι εφικτές κινήσεις και η πρώτη που βελτιώνει τη λύση διατηρείται. Ο περιορισμός των time windows μπορεί να ελεγχθεί σε χρόνο $O(1)$.

Το δεύτερο σύνολο από γειτονιές καθιστά ικανή την εισαγωγή κορυφών που δεν ανήκουν ακόμα στη λύση. Μια γειτονιά $N_k(s)$, $k = 0, \dots, k_{\max}$ ορίζεται ως το πλήθος k συνεχόμενων κορυφών που αφαιρούνται από μια διαδρομή και αντικαθίστανται από μια αλυσίδα κορυφών που δεν ανήκουν σε κάποια διαδρομή. Η αναζήτηση ξεκινά με $k=0$ που σημαίνει ότι δε μετακινείται καμία κορυφή και διερευνά όλες τις διαδρομές για να βρεθεί μια κίνηση που μεγιστοποιεί τη διαφορά μεταξύ του συνολικού οφέλους των εισαγόμενων κορυφών και του συνολικού οφέλους των κορυφών που αφαιρούνται. Μόνο οι κινήσεις με θετική διαφορά μπορούν να γίνουν δεκτές. Η καλύτερη κίνηση για την τρέχουσα τιμή του k , που βρέθηκε ανάμεσα σε όλες τις διαδρομές, εκτελείται και η αναζήτηση συνεχίζεται στην ίδια γειτονιά μέχρι να μην μπορεί να βρεθεί κάποια βελτίωση. Αν δεν βρεθεί κάποια βελτίωση (θετική διαφορά) για την τρέχουσα τιμή του k , αυτή αυξάνεται κατά 1 και η αναζήτηση αρχίζει ξανά. Η διαδικασία σταματά για $k = k_{\max}$ ή όταν βρεθεί μια βελτίωση της λύσης νωρίτερα. Συνεπώς η διαδικασία δεν εξερευνά απαραίτητα όλες τις γειτονιές N_k για $k=0, \dots, k_{\max}$, καθώς αυτό θα απαιτούσε μεγάλη υπολογιστική προσπάθεια.

Η απόφαση για την εισαγόμενη ακολουθία κορυφών ανήκει στα NP-hard προβλήματα. Μπορεί να μοντελοποιηθεί ως πρόβλημα ως εξής: δεδομένου ενός χρονικού ορίου T_{ij} ανάμεσα σε δυο κορυφές v_i και v_j και ένα σύνολο διαθέσιμων κορυφών $U \subset V$, το ζητούμενο είναι να βρεθεί μια ακολουθία κορυφών $v_{i_1}, v_{i_2}, \dots, v_{i_q}$ από το U με μήκος μικρότερο ή ίσο από το T_{ij} που μεγιστοποιεί το συνολικό όφελος της εισαγόμενης ακολουθίας. Επιπλέον πρέπει να ικανοποιείται ο περιορισμός του time window κάθε κορυφής. Παρατηρούμε ότι αυτό το πρόβλημα είναι ισοδύναμο με το πρόβλημα του απλού (point-to-point) Orienteering με time windows. Τα v_i και v_j μπορούν να θεωρηθούν ως τα σημεία έναρξης και τερματισμού αντίστοιχα και το T_{ij} να είναι το όριο κόστους (budget). Πάρα την υπολογιστική πολυπλοκότητα, η ακριβής εκτίμηση (εύρεση) της ακολουθίας μπορεί να είναι εφικτή για μέτριες τιμές του k , γιατί ο αριθμός των εφικτών ακολουθιών ελαττώνεται λόγω των περιορισμών των time windows.

Η εκτίμηση της εισαγόμενης ακολουθίας θα μπορούσε να γίνει με χρήση δυναμικού προγραμματισμού, ωστόσο οι Labadie et al. [20] προτείνουν μια εναλλακτική ευρετική εκτίμηση βασισμένη στο κριτήριο (3) της παραγράφου 7.4.1. Η ακολουθία επεκτείνεται με εκείνη την κορυφή v_j που έχει τη μεγαλύτερη τιμή $ratio_j$ από τις διαθέσιμες κορυφές με την προϋπόθεση ότι η εισαγωγή είναι εφικτή. Αυτή η ευρετική εκτίμηση της ακολουθίας έχει πολυπλοκότητα $O(n^2)$.

7.4.3.2 Διαδικασία Διαταραχής (Perturbation)

Η διαταραχή μπορεί να θεωρηθεί ως μια τυχαία μετάλλαξη της τρέχουσας αρχικής λύσης. Η διαδικασία επιλέγει τυχαία μια ακολουθία από p συνεχόμενες κορυφές τις οποίες αφαιρεί από κάθε διαδρομή. Στη συνέχεια κατασκευάζει ακολουθίες από τις κορυφές που δεν ανήκουν στη λύση οι οποίες μπορούν να εισαχθούν σε κάποια διαδρομή λόγω της μείωσης του κόστους από την αφαίρεση των p κορυφών. Η ακολουθία με το μεγαλύτερο όφελος εισάγεται σε κάποια διαδρομή και αυτή η διαδρομή δεν εξετάζεται περαιτέρω. Η διαδικασία συνεχίζεται μέχρι όλες οι διαδρομές να ενσωματώσουν μια ακολουθία ή μέχρι να μην είναι εφικτή η εισαγωγή κάποιας ακολουθίας. Αν η παράμετρος p είναι πολύ μικρή ή οι διαθέσιμες κορυφές είναι λίγες, μπορεί κάποια διαδρομή να μην ενσωματώνει καμιά ακολουθία. Σε αυτήν την περίπτωση απλά αφαιρούνται οι p κορυφές.

7.5 Προτεινόμενος αλγόριθμος

Σε αυτήν την ενότητα προτείνουμε ένα νέο ευρετικό αλγόριθμο για το TOPTW και τον συγκρίνουμε με τον ILS για να εξετάσουμε την απόδοσή του.

Ο ευρετικός αλγόριθμος που θα παρουσιάσουμε δίνει μια καλή προσεγγιστική λύση στο TOPTW όπου κάθε διαδρομή ξεκινά και καταλήγει στο ίδιο σημείο και έχει την ίδια μέγιστη χρονική διάρκεια (budget) με τις υπόλοιπες. Συνδυάζει δυο ήδη δημοσιευμένους ευρετικούς αλγόριθμους. Ο ένας είναι ο αλγόριθμος Iterated Local Search (ILS) του Vansteenwegen et al. [2] και ο άλλος είναι ο Average Slack που βασίζεται στον αλγόριθμο Slack CSCRoutes [3] προσαρμοσμένο σε στιγμιότυπα όπου το κόστος διαδρομής μεταξύ των κόμβων είναι σταθερό και δεν εξαρτάται από το χρόνο αναχώρησης. Επιπλέον ο Average Slack που χρησιμοποιούμε διαφοροποιείται από τον Slack CSCRoutes [3] στο ότι δε χρησιμοποιεί την τεχνική της συσταδοποίησης (Clustering), της αντικατάστασης(replace) και της τυχαιότητας (randomization).

Το βήμα αντικατάστασης συμπεριλαμβάνεται στη διαδικασία τοπικής αναζήτησης και αντικαθιστά έναν κόμβο που ανήκει στη λύση με έναν κόμβο εκτός της λύσης. Στόχος είναι να γίνει η αντικατάσταση κόμβων με τη μεγαλύτερη διαφορά στο όφελος προκειμένου να βελτιωθεί η λύση.

Ο αλγόριθμος που προτείνουμε συνδυάζει τον ILS και τον Average Slack στο βήμα εισαγωγής. Η ιδέα είναι ότι σε διαδρομές με μεγάλους χρόνους αναμονής (waiting time) λειτουργεί καλύτερα ο Average Slack ενώ σε διαδρομές με μικρούς χρόνους αναμονής ο ILS. Όσο μεγαλύτερη είναι η μεταβλητή slack ενός κόμβου i σε μια διαδρομή r τόσο πιο πιθανή είναι η εφικτή εισαγωγή νέων κόμβων πριν από τον i στην r . Συνεπώς αν οι κόμβοι που έπονται του i στη διαδρομή r έχουν μεγάλους χρόνους αναμονής, οι χρόνοι αυτοί περιορίζονται από την εισαγωγή κόμβων πριν από τον κόμβο i .

Το πόσο μικροί θα είναι οι χρόνοι αναμονής των διαδρομών καθορίζεται από μια μεταβλητή θ (threshold) η οποία δίνεται ως είσοδος στον αλγόριθμο και κυμαίνεται από 0 έως 1.

Αναλυτικά το βήμα εισαγωγής έχει ως εξής: Χωρίζουμε τις m διαφορετικές διαδρομές σε δύο σύνολα R1 και R2. Το σύνολο R1 περιέχει τις διαδρομές με χαμηλό συνολικό άθροισμα χρόνων αναμονής ανά κόμβο (waiting time). Το R2 περιέχει τις διαδρομές που έχουν υψηλό συνολικό χρόνο αναμονής. Το όριο του χρόνου αναμονής των διαδρομών βάσει του οποίου θα γίνει ο διαχωρισμός στα δυο σύνολα R1 και R2 καθορίζεται από τη μεταβλητή θ (threshold) που είναι είσοδος στον αλγόριθμο. Συγκεκριμένα για κάθε διαδρομή υπολογίζεται το πηλίκο του συνολικού χρόνου αναμονής της διαδρομής προς τη μέγιστη επιτρεπόμενη χρονική διάρκεια (budget). Αν το πηλίκο είναι μικρότερο από θ , η διαδρομή ανήκει στο σύνολο R1. Αλλιώς ανήκει στο R2. Πρέπει να σημειωθεί ότι ο συνολικός χρόνος αναμονής κάθε διαδρομής, εκτός από το άθροισμα των χρόνων αναμονής των κόμβων της, περιλαμβάνει και τη διαφορά της μέγιστης χρονικής διάρκειας (budget) της διαδρομής μείον το συνολικό χρόνο της. Στη συνέχεια εφαρμόζεται το βήμα εισαγωγής του ILS στις διαδρομές του R1 και το βήμα του Average Slack στις διαδρομές του R2. Δηλαδή για τις διαδρομές του R1 αναζητείται ο κόμβος i που δεν ανήκει στη λύση με το υψηλότερο κριτήριο του ILS. Το κριτήριο ορίστηκε στην παρουσίαση του ILS και είναι η μέγιστη ποσότητα

$\frac{p_i^2}{shift_i}$. Για τις διαδρομές στο R2 αναζητείται ο κόμβος j με το υψηλότερο κριτήριο του Average Slack

όπως ορίστηκε στην ενότητα 7.3.

Αν το σύνολο R1 είναι κενό εισάγεται στη λύση ο κόμβος j στην αντίστοιχη διαδρομή και στην αντίστοιχη θέση. Αλλιώς αν το σύνολο R2 είναι κενό εισάγεται ο κόμβος i στην αντίστοιχη θέση. Αν κανένα από τα δύο σύνολα δεν είναι κενό, από τους κόμβους i και j επιλέγεται προς εισαγωγή ο κόμβος με το μεγαλύτερο όφελος. Η διαδικασία εισαγωγής τερματίζει όταν η λύση φτάσει σε τοπικό μέγιστο που σημαίνει ότι δεν προκύπτει κάποια εφικτή εισαγωγή κόμβου σε κάποια διαδρομή. Το βήμα διαταραχής είναι το ίδιο με αυτό των ILS και Average Slack.

Για να μειωθεί ο χρόνος εκτέλεσης του αλγορίθμου εφαρμόσαμε μια τεχνική επιτάχυνσης στον υπολογισμό του averageSlack και στον υπολογισμό των waiting time των διαδρομών. Η λογική είναι να αποθηκεύεται η τιμή του averageSlack και του waiting time των διαδρομών μετά από κάθε εισαγωγή κόμβου ώστε να μην υπολογίζονται συνεχώς στο βήμα εισαγωγής.

Έστω $sumSlack_r = \sum_{i=1}^n slack_i$ όπου οι τιμές των $slack_i$ είναι αυτές που ισχύουν πριν από τη νέα

εισαγωγή στη διαδρομή r . Θεωρούμε την εισαγωγή του κόμβου με δείκτη k μεταξύ των σημείων με δείκτη i και $i+1$. Έστω l ο δείκτης του τελευταίου σημείου στη διαδρομή, του οποίου ο χρόνος άφιξης (arrive) άλλαξε λόγω της εισαγωγής. Επίσης συμβολίζουμε με q το δείκτη του πρώτου σημείου στη διαδρομή του οποίου το MaxStart άλλαξε λόγω της εισαγωγής. Με $slack_j^k$ συμβολίζουμε τη νέα τιμή slack που θα έχει το σημείο j της διαδρομής μετά την εισαγωγή του σημείου k . Ο μέσος όρος των slack μετά την εισαγωγή του σημείου με δείκτη k μετά από το σημείο με δείκτη i στη διαδρομή θα είναι:

$$AvSlack_k^i = \frac{sumSlack_r + \sum_{j=q}^i (slack_j^k - slack_j) + \sum_{j=i+1}^l (slack_j^k - slack_j) + slack_k}{n+1}$$

Μετά την εισαγωγή κάποιου σημείου, αποθηκεύουμε τον αριθμητή του παραπάνω κλάσματος γιατί ισούται με το νέο $sumSlack_r$ και θα χρησιμοποιηθεί για να αποφασιστεί η επόμενη εισαγωγή. Με παρόμοιο τρόπο χρησιμοποιούμε τεχνική επιτάχυνσης για τον υπολογισμό του συνολικού χρόνου αναμονής των διαδρομών. Οι χρόνοι αναμονής αποθηκεύονται και ενημερώνονται μετά από κάθε εισαγωγή.

Η τεχνική αυτή είχε ως αποτέλεσμα να μειωθούν σημαντικά οι χρόνοι εκτέλεσης του συνδυαστικού αλγορίθμου.

Για $\theta=1$ ο συνδυαστικός αλγόριθμος αντιστοιχεί στον ILS.

Ο ψευδοκώδικας του συνδυαστικού αλγορίθμου παρουσιάζεται παρακάτω.

Ψευδοκώδικας του Συνδυαστικού Αλγορίθμου

bestSolution \leftarrow null

bestProfit \leftarrow 0

Αρχικοποίησε την currentSolution ως το σύνολο των K διαδρομών που αρχίζουν και τελειώνουν στο σημείο εκκίνησης (hotel)

maxIterations \leftarrow 150

maxNumberToRemove \leftarrow $N/3 * K$ (όπου N το πλήθος των σημείων και K ο αριθμός των διαδρομών)

notImproved \leftarrow 0

startNumber \leftarrow 1

removeNumber \leftarrow 1

Ενόςω (notImproved < maxIterations) **Κάνε**

Ενόςω η currentSolution δεν είναι τοπικά βέλτιστη **Κάνε**

Insert

Τέλος Ενόςω

```

currentProfit ← το κέρδος της currentSolution
Αν currentProfit > bestProfit Τότε
    bestSolution ← currentSolution
    bestProfit ← currentProfit
    removeNumber ← 1
    notImproved ← 0
Αλλιώς
    notImproved ← notImproved + 1
Τέλος Αν
maxSize ← το μέγεθος της μεγαλύτερης διαδρομής στη λύση
Αν removeNumber > maxSize/2 ή removeNumber > maxNumberToRemove Τότε
    removeNumber ← 1
Τέλος Αν
Shake(removeNumber, startNumber)
removeNumber ← removeNumber + 1
smallestSize ← το μέγεθος της μικρότερης διαδρομής στη λύση
startNumber ← startNumber – smallestSize
Τέλος Αν
Τέλος Ενόςω
Επέστρεψε bestSolution

```

Ψευδοκώδικας της συνάρτησης Εισαγωγής (Insert)

tb: Ο μέγιστος επιτρεπτός χρόνος κάθε διαδρομής

twt_i : ο συνολικός χρόνος αναμονής στη διαδρομή i (* σημείωση στο τέλος)

th: ένας πραγματικός αριθμός μεταξύ 0 και 1 (δίνεται από το χρήστη)

Έστω R1 το σύνολο των διαδρομών r_i για τις οποίες ισχύει $\frac{twt(r_i)}{tb} < th$ και R2 το σύνολο των

υπολοίπων διαδρομών

bestRatio ← 0

bestSlackWeight ← 0

Για κάθε κόμβο που δεν ανήκει στη λύση **Κάνε**

Για κάθε διαδρομή $r \in R1$ **Κάνε**

Εντόπισε τη θέση με το μικρότερο shift

Υπολόγισε το ratio (profit²/shift)

Αν ratio > bestRatio **Τότε**

bestRatio ← ratio

Έστω p_i το σημείο που αντιστοιχεί στο bestRatio και r_i η διαδρομή

Τέλος Αν

Τέλος Για

Για κάθε $r \in R2$ **Κάνε**

Βρες το μέγιστο μέσο slack (avgSlack)

Υπολόγισε την ποσότητα slackWeight = (profit² · avgSlack)

Αν slackWeight > bestSlackWeight **Τότε**

bestSlackWeight = slackWeight

Έστω p_j το σημείο με το μεγαλύτερο slackWeight και r_j η αντίστοιχη

διαδρομή

Τέλος Αν**Τέλος Για****Τέλος Για**

Αν το σύνολο R2 είναι κενό **Τότε**

Εισάγουμε το σημείο p_i στη διαδρομή r_i

Αφαιρούμε το σημείο p_i από τα διαθέσιμα σημεία

Αλλιώς Αν το σύνολο R1 είναι κενό **Τότε**

Εισάγουμε το σημείο p_j στη διαδρομή r_j

Αφαιρούμε το σημείο p_j από τα διαθέσιμα σημεία

Αλλιώς

Εισάγουμε το σημείο με τη μεγαλύτερο profit στην αντίστοιχη διαδρομή

Αφαιρούμε το σημείο από τα διαθέσιμα σημεία

Τέλος Αν

* Στο tw_i συμπεριλαμβάνεται και η διαφορά του budget με το χρόνο της διαδρομής και προστίθεται στο άθροισμα των χρόνων αναμονής των κόμβων που ανήκουν στη διαδρομή

7.5.1 Πειραματικά Αποτελέσματα

Στιγμιότυπα: Η αξιολόγηση του συνδυαστικού αλγορίθμου έγινε χρησιμοποιώντας κάποια δημοσιευμένα στιγμιότυπα καθώς και κάποια νέα τα οποία αναπαράγονται σε περισσότερο ρεαλιστικές συνθήκες. Οι Montemanni και Gambardella δημιούργησαν στιγμιότυπα για το TOPTW βασιζόμενοι στα στιγμιότυπα για το OPTW του Solomon και του Cordeau. Τα στιγμιότυπα του Solomon περιλαμβάνουν 100 κόμβους. Τα στιγμιότυπα $c1^*, r1^*, rc1^*$ του Solomon χαρακτηρίζονται από πολύ μικρότερο χρονικό όριο (budget) και μικρότερα χρονικά παράθυρα από τις επεκτάσεις τους που είναι τα στιγμιότυπα $c2^*, r2, rc2^*$. Τα στιγμιότυπα του Cordeau περιέχουν 48-288 κόμβους, σταθερό χρονικό όριο (1000 λεπτά). Για τα στιγμιότυπα $pr01-pr10$ το μέσο μήκος των χρονικών παραθύρων (time windows) είναι 135 λεπτά ενώ για τα $pr11-pr20$ είναι 269 λεπτά [21]. Όλα τα παραπάνω στιγμιότυπα εκτελούνται για αριθμό διαδρομών $m=1,2,3,4$.

Τα παραπάνω στιγμιότυπα ωστόσο δεν είναι κατάλληλα για να μοντελοποιήσουν ρεαλιστικά προβλήματα σχεδιασμού τουριστικών διαδρομών. Σε ρεαλιστικές συνθήκες τα σημεία ενδιαφέροντος που επισκεπτόμαστε σχετίζονται με μεγαλύτερα χρονικά παράθυρα τα οποία μπορεί να μην είναι ίδια σε κάθε μέρα επίσκεψης. Επιπλέον τα απομονωμένα σημεία είναι λίγα ενώ γενικά τα σημεία ενδιαφέροντος εμφανίζουν μεγαλύτερη πυκνότητα σε συγκεκριμένες περιοχές. Είναι επίσης αληθές ότι το όφελος κάθε σημείου συσχετίζεται με το χρόνο επίσκεψης ενώ σε ρεαλιστικές συνθήκες το χρονικό όριο είναι της τάξης μερικών ωρών ανά ημέρα επίσκεψης (σε αντίθεση με τα προαναφερθέντα στιγμιότυπα τα οποία ορίζουν αρκετά μεγάλο χρονικό όριο).

Για αυτούς τους λόγους δημιουργήθηκαν 100 νέα στιγμιότυπα (t^*) με τα εξής χαρακτηριστικά: ο αριθμός των διαδρομών κυμαίνεται από 1 έως 3, ο αριθμός των κόμβων είναι 100-200, το 80% των κόμβων τοποθετείται σε 1 από 1 έως 10 εικονικά κέντρα που δεν αποτελούν κόμβους του γραφήματος, ο χρόνος επίσκεψης σε κάθε σημείο είναι από 1 έως 120 λεπτά και είναι ανάλογος με το όφελος του σημείου. Σχετικά με τα χρονικά παράθυρα, το 50% των σημείων είναι ανοιχτά όλη την ημέρα, ενώ τα υπόλοιπα είναι κλειστά για μία ή δυο μέρες την εβδομάδα και τις υπόλοιπες μέρες είναι ανοιχτά 8:30-17:00. Το χρονικό όριο είναι 5 ώρες για τα $t2^*$ και κάτι παραπάνω από 10 ώρες στα $t1^*$.

Αποτελέσματα: Ο συνδυαστικός αλγόριθμος υλοποιήθηκε στη C++. Η εκτέλεση του αλγορίθμου έγινε σε υπολογιστή Intel(R) Xeon(R) με επεξεργαστή 2.5 GHz και 16 GB RAM.

Αυτό που εξετάζεται είναι το συνολικό όφελος που συλλέγει η λύση που επιστρέφεται από τον αλγόριθμο και ο χρόνος εκτέλεσης, που μετριέται σε millisecond. Στόχος είναι η μεγιστοποίηση του συνολικού οφέλους στο συντομότερο δυνατό χρόνο εκτέλεσης.

Ο αλγόριθμος συγκρίνεται με τον ILS σε όλα τα στιγμιότυπα του Solomon του Cordeau και τα νέα στιγμιότυπα t^* . Ο αριθμός των διαδρομών στα t^* στιγμιότυπα είναι καθορισμένος και κυμαίνεται από 1 έως 3^{12} . Τα στιγμιότυπα του Solomon και του Cordeau εκτελούνται για αριθμό διαδρομών $m=1,2,3,4$

Σε κάθε στιγμιότυπο εφαρμόστηκε ο συνδυαστικός αλγόριθμος για όλο το εύρος τιμών της παραμέτρου θ ξεκινώντας από 0 με αύξηση 0.01 έως το 1. Η τιμή $\theta=1$ αντιστοιχεί στον ILS. Μας ενδιαφέρει η αξιολόγηση του συνδυαστικού αλγορίθμου στο βήμα εισαγωγής σε σχέση με τον ILS. Για αυτό οι συγκρίσεις γίνονται θεωρώντας ότι ο ILS αντιστοιχεί στην τιμή $\theta=1$.

Ακολουθεί η μέση ποσοστιαία απόκλιση του συνδυαστικού αλγορίθμου σε σχέση με τον ILS ως προς το όφελος.

Μέση Ποσοστιαία Απόκλιση του Συνδυαστικού αλγορίθμου σε σχέση με τον ILS

Name	Κέρδος (%)			
	1	2	3	4
c1*	1,76	1,02	2,12	1,52
c2*	1,91	1,13	1,49	0
r1*	0,97	3,27	2,19	2,16
r2*	1,61	1,4	0,19	0
rc1*	9	5,74	3,76	2,68
rc2*	3,96	3,83	0,71	0,05
pr*	4,01	4,8	4,34	4,62
t1*		3,53		
t2*		6,92		

Οι συγκρίσεις στον παραπάνω πίνακα γίνονται με βάση τη μέγιστη τιμή ως προς το κέρδος του συνδυαστικού αλγορίθμου από όλα τα θ . Φαίνεται ότι ο αλγόριθμος υπερτερεί του ILS όσον αφορά το κέρδος για το κατάλληλο θ .

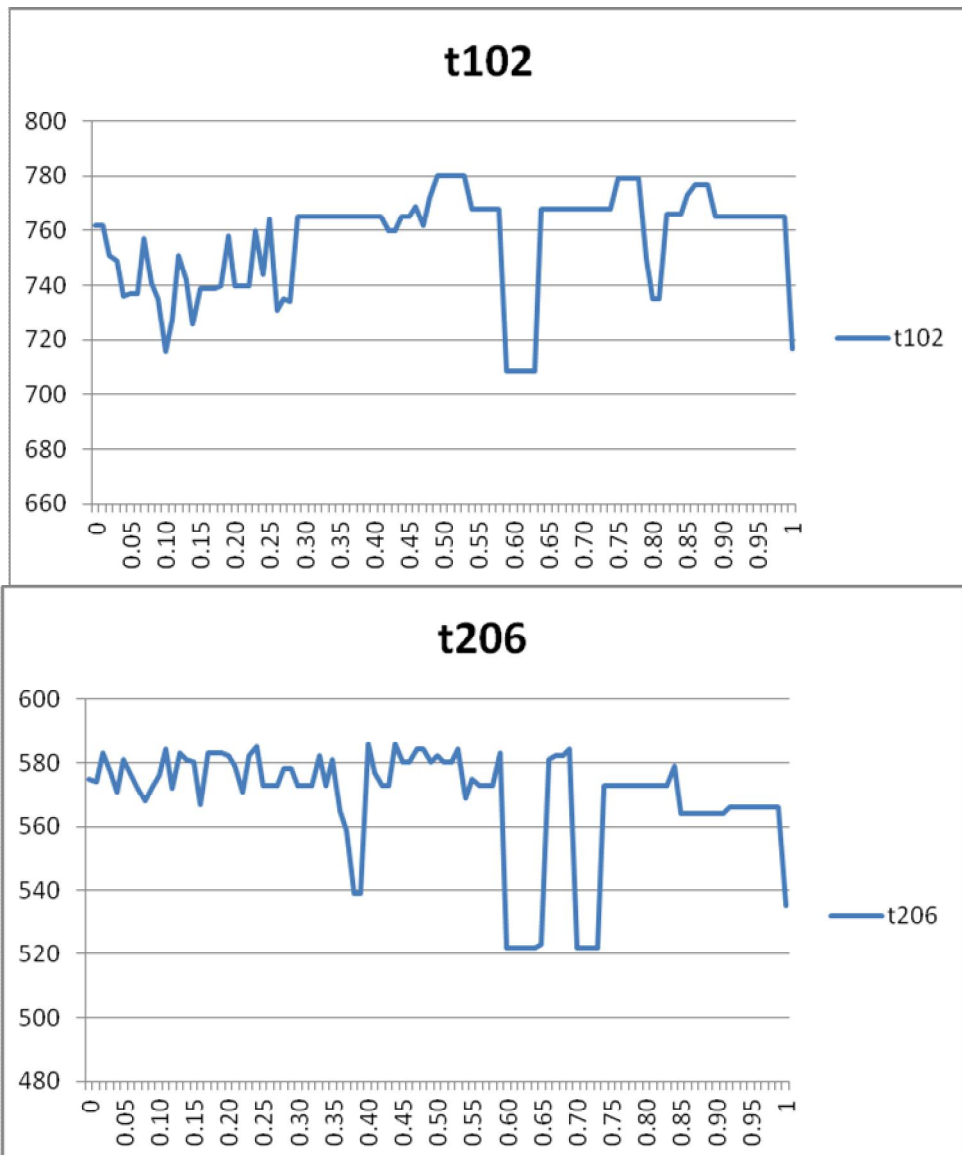
Στα $t2^*$ στιγμιότυπα η μέση απόκλιση είναι 6.92%. Τα $t2^*$ στιγμιότυπα έχουν σχετικά μικρό χρονικό όριο ανά διαδρομή που είναι 5 ώρες. Αυτό έχει ως αποτέλεσμα ο ILS να εγκλωβίζεται σε τοπικά μέγιστες περιοχές και να μην επισκέπτεται περιοχές με μεγάλη πυκνότητα κόμβων εξαιτίας του αυξημένου κόστους μετάβασης που ελαττώνει το κριτήριο εισαγωγής του ILS. Επίσης στα t^* στιγμιότυπα, ο αριθμός των κόμβων είναι αυξημένος σε σχέση με τα στιγμιότυπα του Solomon έχοντας ως αποτέλεσμα ο ILS να μειονεκτεί καθώς το κριτήριο εισαγωγής του είναι τοπικό και δεν εξερευνά όλο το χώρο αναζήτησης. Αντίθετα ο average Slack έχει καθολικό κριτήριο και ευνοεί την εισαγωγή πολλών νέων κόμβων.

Τα στιγμιότυπα rc1* έχουν χαμηλό μέγιστο χρονικό όριο ανά διαδρομή που εξηγεί την υψηλή μέση ποσοστιαία απόκλιση. Ο average Slack δημιουργεί μεγάλα slacks στους κόμβους που ανήκουν στη λύση ευνοώντας την εισαγωγή πολλών νέων κόμβων. Αυτό έχει ως αποτέλεσμα να παράγει λύσεις με μεγάλο πλήθος κόμβων, που τον κάνει πιο κερδοφόρο για μικρό αριθμό διαδρομών. Ο συνδυαστικός αλγόριθμος αρχικά τρέχει τον average Slack δημιουργώντας διαδρομές με μικρό συνολικό χρόνο αναμονής και στη συνέχεια εφαρμόζεται ο ILS σε αυτές τις διαδρομές βελτιώνοντας ακόμα περισσότερο τη λύση με τοπική αναζήτηση. Για τα rc1* στιγμιότυπα η μέση απόκλιση μειώνεται όσο αυξάνονται οι διαδρομές παραμένοντας ωστόσο υψηλή και για 4 διαδρομές που είναι 2.68%.

¹² Ο αριθμός των διαδρομών στα t^* στιγμιότυπα εμπεριέχεται ως δεδομένο και διαφέρει ανά στιγμιότυπο. Στα στιγμιότυπα του Solomon και του Cordeau ο αριθμός των διαδρομών καθορίζεται από το χρήστη. Στα πειράματα εκτελέστηκε ο αλγόριθμος με τον προκαθορισμένο αριθμό διαδρομών στα t^* στιγμιότυπα ενώ στα στιγμιότυπα του Solomon και του Cordeau για $m=1,2,3,4$

Στα στιγμιότυπα του Cordeau η απόκλιση είναι αρκετά υψηλή. Αυτό οφείλεται στον μεγάλο αριθμό υποψήφιων σημείων και στα μικρά χρονικά παράθυρα.

Οι τιμές με τις οποίες έγιναν οι συγκρίσεις είναι οι μεγαλύτερες τιμές του συνδυαστικού αλγορίθμου για θ από 0 έως 1. Ο συνδυαστικός αλγόριθμος στα t^* και pr^* στιγμιότυπα εμφανίζει συνολικό όφελος μεγαλύτερο του ILS σε αρκετά μεγάλο εύρος τιμών του θ . Το γεγονός ότι ο συνδυαστικός αλγόριθμος έχει μεγάλη μέση απόκλιση από τον ILS στα t^* είναι σημαντικό γιατί αυτά τα στιγμιότυπα ανταποκρίνονται σε ρεαλιστικές συνθήκες. Ενδεικτικά παρατίθενται τα στιγμιότυπα $t102$, $t206$ με γραφικές παραστάσεις σε όλο το εύρος τιμών του θ .



Συνδυαστικός Αλγόριθμος

Ο χρόνος εκτέλεσης του συνδυαστικού αλγορίθμου είναι συγκρίσιμος με το χρόνο εκτέλεσης του ILS. Σε κάποιες περιπτώσεις προκύπτει μεγαλύτερο όφελος από τον ILS με μικρότερο χρόνο εκτέλεσης. Στην ενότητα 10 παρατίθενται τα αναλυτικά αποτελέσματα για όλα τα στιγμιότυπα.

8. Συμπεράσματα

Σε αυτήν την εργασία ορίστηκε το πρόβλημα προσανατολισμού (Orienteering). Το πρόβλημα ανήκει στην κλάση των NP-hard προβλημάτων με αποτέλεσμα να έχουν αναπτυχθεί προσεγγιστικοί αλγόριθμοι για την επίλυσή του με πολυωνυμικό χρόνο εκτέλεσης. Το πρόβλημα ανήκει στην κατηγορία των APX-hard

προβλημάτων με παράγοντα προσέγγισης $\frac{1481}{1480}$. Αυτό σημαίνει ότι το πρόβλημα δεν μπορεί να

προσεγγιστεί σε πολυωνυμικό χρόνο με παράγοντα προσέγγισης μικρότερο του $\frac{1481}{1480}$.

Η προσέγγιση του προβλήματος στηρίζεται στα προβλήματα k-stroll και Min-Excess. Όταν το γράφημα είναι μη κατευθυνόμενο, η προσέγγιση του Orienteering προβλήματος ανάγεται στην προσέγγιση του Min-Excess προβλήματος το οποίο ανάγεται στην προσέγγιση του k-stroll. Χρησιμοποιώντας μια δικριτηριακή προσέγγιση στο k-stroll πρόβλημα προκύπτει μια $(2+\epsilon)$ -προσέγγιση στο OP βάσει των προαναφερθέντων αναγωγών. Το ϵ είναι οποιοσδήποτε θετικός αριθμός.

Όταν το γράφημα είναι κατευθυνόμενο, ορίζεται το πρόβλημα minimum ratio ATSP. Έτσι οι αναγωγές για την προσέγγιση είναι: η προσέγγιση του Orienteering προβλήματος ανάγεται στην προσέγγιση του Min-Excess προβλήματος το οποίο ανάγεται στην προσέγγιση του k-stroll το οποίο ανάγεται στην προσέγγιση

του minimum ratio ATSP. Έτσι προκύπτει μια $O\left(\frac{\log^2 n}{\log \log n}\right)$ -προσέγγιση για το κατευθυνόμενο

Orienteering πρόβλημα.

Το γράφημα μπορεί να έχει θετικό όφελος σε κάθε κόμβο ή να έχει χρονικά παράθυρα εντός των οποίων πρέπει να γίνει η επίσκεψη στους κόμβους. Τότε το πρόβλημα ονομάζεται βεβαρυμμένο (weighted) OP και OP με time windows (OPTW) αντίστοιχα. Το βεβαρυμμένο OP έχει προσεγγιστικό παράγοντα $\rho(1+\alpha(1))$ όπου ρ ο προσεγγιστικός παράγοντας του OP με μοναδιαία οφέλη στους κόμβους. Το OPTW έχει παράγοντα προσέγγισης $O(\alpha \cdot \max\{\log OPT, \log L\})$ όπου α η προσέγγιση του OP και L ο λόγος του μήκους του μεγαλύτερου χρονικού παραθύρου προς το μήκος του μικρότερου χρονικού παραθύρου.

Στη συνέχεια ορίζεται το πρόβλημα του ομαδικού προσανατολισμού (Team Orienteering) που αποτελεί γενίκευση του OP. Στο TOP υπάρχουν περισσότερες από μια διαδρομές. Κατά αναλογία με το OP ορίζεται το TOP με time windows (TOPTW) και παρουσιάζονται κάποιοι ευρετικοί αλγόριθμοι για το TOP και το TOPTW.

Οι ευρετικοί αλγόριθμοι για το TOP είναι ο GRASP και ο CGW. Για το TOPTW παρουσιάζονται οι ευρετικοί αλγόριθμοι ILS, Average Slack, GRASP-ELS.

Τέλος προτείνουμε ένα νέο ευρετικό αλγόριθμο για την επίλυση του TOPTW που συνδυάζει τον ILS και τον Average Slack. Ο συνδυαστικός αυτός αλγόριθμος συγκρίνεται με τον ILS και παρουσιάζει μεγαλύτερα συνολικά οφέλη σε συγκρίσιμο χρόνο εκτέλεσης με τον ILS. Τα αναλυτικά αποτελέσματα παρουσιάζονται στην ενότητα 10.

9. Αναφορές

- [1] Nagarajan, V., Ravy, R.: Poly-logarithmic Approximation Algorithms for Directed Vehicle Routing Problems. In: Workshop on Approximation Algorithms for Combinatorial Optimization Problems, pp. 257-270 (2007)
- [2] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden. Iterated Local Search for the team orienteering problem with time windows. Computers & Operations Research, 36 :3281-3290, 2009.
- [3] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, Grammati Pantziou and Nikolaos Vathis. Heuristics for the Time Dependent Team Orienteering Problem: Application to Tourist Route Planning. Comp. & Oper. Res. 62, 36-50, 2015
- [4] S. Boussier, D. Feillet and M. Gendreau. An exact algorithm for team orienteering problems. 4OR: A Quarterly Journal of Operations Research, 5:211-230.

- [5] T. Tsiligirides. Heuristic methods Applied To Orienteering. The Journal of the Operational Research Society, 35 (9) :pp . 797-809 , 1984
- [6] P. Vansteenwegen, W. Souffriau and D. Van Oudheusden . The Orienteering Problem: A Survey. European Journal Of Operational Research , 209 (1): 1-10, 2011
- [7] A. Blum ,S. Chawla, D. R. Karger, T. Lane, A. Meyerson , and M. Minkoff. Approximation Algorithms for Orienteering and Discounted-Reward TSP. In Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on, pages 46-55 , oct. 2003
- [8] N. J. Korula. Approximation Algorithms for network design and Orienteering. Phd Thesis, University of Illinois at Urbana-Campaign, 2010
- [9] B. Awerbuch, Y. Azar, A. Blum and S. Vempala . New Approximation Guarantees for minimum-weight k-trees and prize-collecting salesmen. SIAM J. Comput., 28 (1) : 254-262, 1998
- [10] C. Chekuri, N. Korula and M. Pal. Improved algorithms for orienteering and related problems. In Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete Algorithms , SODA '08 , pages 661-670, Philadelphia , PA, USA, 2008. Society for Industrial and Applied Mathematics.
- [11]K. Chaudhuri , B. Godfrey, S. Rao and K. Talwar. Paths, trees and minimum latency tours. In Foundations of Computer Science, 2003 . Proceedings. 44th Annual IEEE Symposium on , pages 36-45 , oct. 2003
- [12] N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation algorithms for deadline-tsp and vehicle routing with time windows. In Proceedings of the thirty-sixth annual ACM symposium on theory of Computing, STOC 04, pages 166-174, New York , NY, USA, 2004. ACM.
- [13] V. Nagarajan and R. Ravi. The directed orienteering problem. Algorithmika , 60:1017-1030, August 2011.
- [14] G. Laporte and S. Martello . The selective travelling salesman problem. Discrete Applied Mathematics , 26(2-3):193-207, 1990 .
- [15] S. Arora and G. Karakostas. A $2+\epsilon$ approximation algorithm for the k-mst problem . Mathematical Programming, 107:491-504
- [16] T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set cover problem. Operations Research Letters , 8:67-71, 1989
- [17] Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., Van Oudheusden, D.: A greedy randomized adaptive search procedure for the team orienteering problem. In:EU/Meeting 2008 on metaheuristics for logistics and vehicle routing (2008)
- [18] Chao, I.-M. , Golden, B.L. , Wasil, E.A. : The team orienteering problem. Eur. J. Oper. Res. 88(3), 464-474 (1996)
- [19] Lin, S. (1965) : "Computer solutions of the travelling salesman problem" , Bell System Technical Journal 44, 2245-2269.
- [20] Labadie, N., Melechovsky, J., Wolfler Calvo , R.: Hybridized evolutionary local search algorithm for the team orienteering problem with time windows. J. Heuristics 17,729-753 (2011)
- [21] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, Y. Tasoulas. Cluster-Based Heuristics for the Team Orienteering Problem with Time Windows. In Proceedings of 12th International Symposium on Experimental Algorithms (SEA 13), pages 390-401, 2013.
- [22] Lars Engebretsen and Marek Karpinski. Approximation hardness of TSP with bounded metrics. In Proceedings of the 28th International Colloquium on Automata , Languages and Programming, pages 201-212. Springer-Verlag, 2001

10.Αναλυτικά αποτελέσματα

Παραθέτουμε τα πειραματικά αποτελέσματα στα στιγμιότυπα του Solomon, του Cordeau και τα καινούρια στιγμιότυπα ($t1^*$, $t2^*$). Τα στιγμιότυπα του Solomon και του Cordeau παρουσιάζονται για αριθμό διαδρομών $m=1,2,3,4$. Σε κάθε στιγμιότυπο εμφανίζονται έως 3 διακριτές τιμές με το υψηλότερο όφελος για όλα τα θ από 0 έως 1 και ο χρόνος εκτέλεσης σε millisecond. Επίσης αναφέρεται η τιμή του θ με την

οποία προέκυψε το αντίστοιχο όφελος. Αριστερά παρατίθεται το όφελος (κέρδος) του ILS και ο χρόνος εκτέλεσής του.

Αποτελέσματα για στιγμιότυπα του Solomon για 1 διαδρομή

Τοπολογία	ILS			Συνδυαστικός		
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	Threshold (θ)	
c101	310	1	310	0	0,33	
c102	360	3	360	1	0,41	
			330	1	0,33	
c103	380	1	400	12	0,39	
			390	1	0,2	
			380	1	0,59	
c104	400	11	420	15	0,43	
			400	5	0,53	
			390	2	0,35	
c105	340	1	340	1	0,87	
			330	4	0,23	
c106	340	2	340	1	0,59	
c107	360	3	370	4	0,25	
			360	1	0,59	
			350	1	0,16	
c108	360	1	370	3	0,57	
			360	1	0,58	
c109	380	1	380	1	0,51	
c201	860	126	860	45	0,33	
			850	8	0,25	
			840	5	0,42	
c202	890	12	930	11	0,69	
			920	198	0,27	
			910	7	0,46	
c203	940	109	960	61	0,33	
			950	59	0,31	
			940	9	0,41	

c204	950	233	970	90	0,17
			960	13	0,62
			950	14	0,59
c205	900	30	900	9	0,96
			890	15	0,51
			880	37	0,06
c206	900	8	930	53	0,69
			920	100	0,37
			910	12	0,88
c207	920	253	930	173	0,33
			920	81	0,32
			910	14	0,9
c208	930	39	950	31	0,04
			940	13	0,64
			930	7	0,57
r101	182	1	198	4	0,34
			197	5	0,11
			182	0	0,8
r102	281	1	286	1	0,11
			281	1	0,76
			280	9	0,37
r103	286	1	286	1	0,58
r104	297	2	297	1	0,57
			288	2	0,22
r105	240	1	240	1	0,76
			238	10	0,22
			236	3	0,24
r106	281	1	293	1	0,49
			286	2	0,08
			281	1	0,76
r107	288	5	288	3	0,76
			286	1	0,59
r108	308	17	297	1	0,58
			292	3	0,76
			288	2	0,22
r109	277	32	277	23	0,8
			273	30	0,36
			259	2	0,18
r110	281	18	281	4	0,56
			280	24	0,09
			278	11	0,19
r111	294	26	295	5	0,76
			276	11	0,14
			275	1	0,59
r112	295	3	295	32	0,56
			292	16	0,81

			277	19	0,19
r201	769	31	786	232	0,39
			785	365	0,42
			784	61	0,38
r202	889	243	904	382	0,25
			901	434	0,32
			900	197	0,27
r203	995	270	1016	228	0,4
			1006	324	0,29
			1005	400	0,46
r204	1062	373	1081	309	0,59
			1080	77	0,78
			1077	209	0,96
r205	940	408	951	220	0,24
			946	491	0,64
			945	236	0,28
r206	1024	135	1015	479	0,59
			1012	162	0,72
			1007	381	0,96
r207	1030	79	1069	490	0,34
			1051	52	0,88
			1049	262	0,26
r208	1109	414	1104	467	0,31
			1102	143	0,9
			1101	858	0,47
r209	922	453	941	147	0,43
			936	36	0,24
			934	681	0,48
r210	952	238	971	429	0,43
			965	84	0,9
			964	237	0,42
r211	1013	29	1036	392	0,42
			1032	639	0,15
			1030	288	0,5
rc101	213	1	219	1	0,51
rc102	228	1	266	6	0,2
			259	5	0,65
			258	4	0,28
rc103	235	1	263	17	0,1
			257	28	0,04
			255	2	0,67
rc104	275	3	280	16	0,07
			278	3	0,24
			275	1	0,34
rc105	215	1	244	15	0,3
			236	6	0,15
			234	7	0,32
rc106	239	2	250	1	0,25

			246	2	0,09
			243	1	0,17
rc107	253	1	277	1	0,19
			276	1	0,06
			268	1	0,38
rc108	268	36	298	1	0,41
			288	2	0,59
			263	18	0,63
rc201	769	16	794	59	0,27
			784	55	0,26
			783	24	0,5
rc202	891	316	933	225	0,81
			925	195	0,37
			922	364	0,82
rc203	937	42	990	131	0,28
			978	204	0,33
			972	614	0,15
rc204	1057	371	1131	694	0,66
			1122	88	0,56
			1120	525	0,27
rc205	835	40	849	265	0,59
			846	173	0,42
			845	134	0,34
rc206	860	22	889	159	0,56
			877	35	0,26
			876	114	0,2
rc207	924	164	950	420	0,51
			944	385	0,3
			938	225	0,85
rc208	1012	91	1044	247	0,31
			1043	470	0,33
			1042	285	0,84

 Αποτελέσματα για σπιγμιότυπα του Solomon για 2 διαδρομές

2 Tours

Τοπολογία	ILS		Συνδυαστικός		Threshold (θ)
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	
c101	570	3	590	118	0,27
			580	4	0,43
			570	3	0,5
c102	650	93	660	75	0,26
			650	11	0,3
			640	7	0,81
c103	710	218	720	107	0,5
			710	17	0,2

			700	4	0,79
c104	740	91	760	46	0,43
			750	5	0,66
			740	14	0,8
c105	640	4	640	4	0,9
			630	8	0,14
			620	21	0,17
c106	620	4	620	7	0,86
			610	4	0,39
c107	670	15	670	9	0,83
			660	10	0,67
			650	22	0,15
c108	680	70	680	6	0,72
			670	12	0,63
			660	5	0,57
c109	720	70	720	21	0,25
			710	16	0,09
			700	30	0,18
c201	1420	130	1440	151	0,16
			1430	27	0,65
			1420	31	0,42
c202	1420	95	1450	406	0,08
			1440	95	0,42
			1430	37	0,94
c203	1440	65	1460	25	0,53
			1450	24	0,46
			1440	22	0,6
c204	1450	279	1460	80	0,68
			1450	29	0,5
			1440	26	0,57
c205	1440	264	1460	62	0,55
			1450	34	0,38
			1440	26	0,64
c206	1470	139	1480	54	0,42
			1470	42	0,6
			1460	38	0,63
c207	1460	938	1470	49	0,2
			1460	44	0,73
			1450	43	0,87
c208	1470	367	1480	51	0,42
			1470	22	0,53
			1460	19	0,6
r101	341	9	341	29	0,79
			339	31	0,48
			338	11	0,51
r102	498	4	508	223	0,25
			504	155	0,21
			503	24	0,71
r103	508	11	514	29	0,34

			513	77	0,25
			512	330	0,47
r104	524	36	547	335	0,31
			546	180	0,22
			543	221	0,27
r105	417	25	453	221	0,13
			442	73	0,16
			436	29	0,24
r106	513	12	529	102	0,25
			524	34	0,4
			522	169	0,17
r107	527	4	538	65	0,1
			535	224	0,23
			531	126	0,58
r108	542	35	557	496	0,11
			556	29	0,78
			555	342	0,29
r109	498	149	498	31	0,36
			497	52	0,38
			496	28	0,81
r110	474	12	523	240	0,11
			520	85	0,12
			517	124	0,31
r111	531	20	539	226	0,45
			538	30	0,81
			535	71	0,77
r112	521	4	538	59	0,52
			536	69	0,79
			535	157	0,28
r201	1208	1747	1223	1053	0,6
			1221	405	0,59
			1219	89	0,82
r202	1297	193	1316	474	0,95
			1312	175	0,58
			1308	1124	0,39
r203	1385	235	1399	927	0,63
			1397	91	0,71
			1390	740	0,62
r204	1432	501	1450	190	0,54
			1445	143	0,82
			1444	375	0,49
r205	1324	675	1355	178	0,23
			1354	115	0,94
			1351	126	0,59
r206	1399	53	1440	180	0,78
			1422	599	0,66
			1421	620	0,6
r207	1430	42	1452	581	0,3
			1449	130	0,5
			1446	198	0,59

r208	1458	57	1458	37	0,94
			1457	74	0,43
			1456	40	0,88
r209	1355	870	1382	92	0,71
			1375	518	0,77
			1373	83	0,38
r210	1389	151	1398	284	0,56
			1390	165	0,59
			1389	139	0,42
r211	1430	617	1444	738	0,84
			1443	526	0,77
			1442	70	0,6
rc101	417	5	419	18	0,31
			417	15	0,6
			410	18	0,15
rc102	497	44	503	32	0,4
			497	15	0,42
			493	18	0,58
rc103	486	60	523	23	0,21
			522	55	0,39
			521	17	0,59
rc104	541	47	569	128	0,46
			563	85	0,45
			560	48	0,09
rc105	445	33	468	19	0,15
			467	30	0,16
			450	6	0,67
rc106	450	9	477	23	0,2
			475	4	0,25
			474	87	0,28
rc107	489	6	516	58	0,17
			515	6	0,55
			506	7	0,16
rc108	480	24	551	232	0,24
			550	81	0,3
			546	28	0,29
rc201	1320	81	1360	854	0,15
			1340	134	0,38
			1337	319	0,4
rc202	1413	245	1477	658	0,01
			1470	247	0,52
			1465	193	0,66
rc203	1536	748	1582	1263	0,12
			1576	174	0,51
			1565	253	0,84
rc204	1665	53	1674	855	0,27
			1672	362	0,73
			1671	509	0,91
rc205	1360	239	1427	525	0,56
			1423	518	0,34

			1418	812	0,37
rc206	1438	56	1534	300	0,28
			1494	196	0,91
			1493	327	0,35
rc207	1505	221	1567	479	0,57
			1557	286	0,21
			1554	674	0,45
rc208	1596	52	1657	565	0,34
			1656	1192	0,37
			1651	1134	0,91

Αποτελέσματα για στιγμιότυπα του Solomon για 3 διαδρομές

3 Tours

Τοπολογία	ILS		Συνδυαστικός		Threshold (θ)
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	
c101	790	92	800	22	0,32
			790	8	0,8
			780	8	0,66
c102	870	7	910	505	0,7
			900	135	0,77
			890	12	0,29
c103	950	149	970	319	0,5
			960	52	0,44
			950	10	0,69
c104	990	111	1010	40	0,33
			1000	128	0,37
			990	9	0,68
c105	840	7	860	346	0,4
			850	29	0,86
			840	11	0,77
c106	850	77	860	54	0,45
			850	13	0,86
			840	14	0,49
c107	900	56	910	271	0,54
			900	64	0,56
			890	13	0,16
c108	890	161	920	264	0,32
			910	105	0,37
			900	8	0,72
c109	950	105	960	68	0,71
			950	8	0,79
			940	9	0,69
c201	1730	330	1790	395	0,19
			1780	39	0,92
			1770	310	0,33

c202	1740	155	1790	190	0,51
			1780	46	0,5
			1770	366	0,85
c203	1770	68	1780	319	0,51
			1770	66	0,67
			1760	53	0,57
c204	1770	237	1790	42	0,59
			1780	33	0,73
			1770	29	0,77
c205	1780	229	1810	680	0,9
			1790	36	0,79
			1780	38	0,63
c206	1810	181	1810	58	0,59
			1800	65	0,72
			1790	41	0,66
c207	1790	139	1810	94	0,34
			1800	63	0,45
			1790	35	0,84
c208	1790	54	1810	42	0,7
			1800	122	0,63
			1790	53	0,26
r101	470	73	478	8	0,18
			477	7	0,37
			476	8	0,3
r102	683	210	686	294	0,81
			683	253	0,64
			682	130	0,67
r103	720	1052	725	230	0,37
			724	44	0,53
			723	197	0,55
r104	767	180	773	866	0,55
			770	1218	0,42
			769	225	0,47
r105	589	88	611	50	0,76
			609	174	0,82
			608	273	0,43
r106	717	108	724	77	0,45
			723	291	0,32
			722	244	0,3
r107	736	117	753	1630	0,58
			752	269	0,37
			751	919	0,04
r108	785	66	797	279	0,32
			794	1218	0,36
			793	774	0,51
r109	685	47	702	515	0,37
			701	26	0,78
			699	10	0,74
r110	671	30	732	7	0,72
			730	191	0,06

			729		334	0,49
r111	766	225	772		112	0,69
			770		272	0,55
			769		85	0,42
r112	763	310	776		542	0,1
			775		253	0,25
			774		530	0,11
r201	1405	120	1423		431	0,43
			1420		333	0,88
			1419		222	0,58
r202	1446	609	1458		202	0,69
			1455		184	0,5
			1453		85	0,52
r203	1458	81	1458		25	0,92
r204	1458	22	1458		24	0,94
r205	1458	117	1458		33	0,63
			1449		64	0,28
r206	1458	26	1458		25	0,92
r207	1458	24	1458		24	0,9
r208	1458	22	1458		23	0,93
r209	1458	25	1458		27	0,96
r210	1458	32	1458		25	0,96
r211	1458	22	1458		22	0,96
rc101	597	38	616		220	0,18
			613		110	0,1
			611		106	0,73
rc102	682	47	704		98	0,19
			702		42	0,45
			701		89	0,13
rc103	742	96	763		39	0,58
			758		106	0,33
			755		55	0,21
rc104	778	45	822		677	0,24

			816	263	0,09
			813	52	0,67
rc105	664	25	679	227	0,45
			676	54	0,17
			673	61	0,22
rc106	639	77	688	231	0,4
			687	244	0,34
			686	213	0,3
rc107	734	25	762	67	0,62
			761	256	0,24
			757	409	0,55
rc108	779	203	790	10	0,32
			786	145	0,57
			785	52	0,71
rc201	1621	87	1666	189	0,33
			1659	728	0,17
			1652	625	0,38
rc202	1688	119	1706	538	0,55
			1705	248	0,42
			1703	210	0,39
rc203	1724	85	1724	43	0,55
			1721	48	0,3
			1719	41	0,54
rc204	1724	37	1724	25	0,86
rc205	1672	993	1700	423	0,41
			1685	764	0,48
			1683	84	0,82
rc206	1721	208	1724	104	0,05
			1722	458	0,42
			1721	49	0,76
rc207	1724	429	1724	50	0,68
			1722	58	0,8
			1721	590	0,09
rc208	1724	41	1724	24	0,88

Αποτελέσματα για στιγμιότυπα του Solomon για 4 διαδρομές

4 Tours

Τοπολογία	ILS		Συνδυαστικός		
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	Threshold (θ)
c101	990	277	1010	34	0,79
			1000	45	0,51
			990	14	0,49

c102	1100	623	1130	623	0,48
			1120	53	0,71
			1110	97	0,88
c103	1150	50	1180	375	0,34
			1170	47	0,59
			1160	35	0,58
c104	1220	133	1230	313	0,79
			1220	89	0,76
			1210	45	0,7
c105	1040	61	1060	188	0,39
			1050	15	0,91
			1040	28	0,64
c106	1060	338	1070	182	0,72
			1060	63	0,84
			1050	93	0,4
c107	1100	51	1110	337	0,64
			1100	15	0,55
			1090	19	0,58
c108	1110	758	1120	134	0,56
			1110	12	0,81
			1100	39	0,79
c109	1170	123	1180	441	0,74
			1170	15	0,65
			1160	43	0,37
c201	1810	36	1810	24	0,9
c202	1810	23	1810	24	0,93
c203	1810	24	1810	25	0,92
c204	1810	23	1810	24	0,98
c205	1810	22	1810	23	0,96
c206	1810	23	1810	24	0,93
c207	1810	23	1810	25	0,89
c208	1810	23	1810	23	0,97
r101	593	15	602	243	0,27
			601	158	0,37

			597	48	0,34
r102	798	21	826	245	0,49
			824	500	0,41
			823	217	0,23
r103	876	245	906	131	0,65
			900	161	0,77
			899	148	0,39
r104	939	470	956	311	0,61
			953	659	0,6
			951	278	0,73
r105	759	54	765	74	0,26
			761	491	0,63
			760	259	0,64
r106	876	62	899	379	0,49
			889	1653	0,56
			888	1515	0,54
r107	912	267	935	875	0,24
			932	544	0,32
			928	696	0,3
r108	972	258	979	651	0,59
			976	110	0,65
			975	415	0,49
r109	849	153	879	101	0,66
			873	90	0,71
			864	444	0,65
r110	870	462	896	160	0,41
			893	131	0,39
			892	415	0,46
r111	926	609	939	167	0,72
			935	256	0,77
			930	577	0,52
r112	946	782	956	498	0,65
			955	341	0,4
			954	760	0,05
r201	1458	62	1458	28	0,96
			1455	36	0,32
r202	1458	54	1458	23	0,97
r203	1458	23	1458	24	0,96
r204	1458	23	1458	24	0,93
r205	1458	23	1458	24	0,93
r206	1458	22	1458	24	0,95

r207	1458	23	1458	24	0,96
r208	1458	22	1458	24	0,93
r209	1458	23	1458	24	0,96
r210	1458	23	1458	24	0,96
r211	1458	23	1458	24	0,96
rc101	794	190	801	98	0,66
			797	549	0,18
			791	97	0,41
rc102	885	296	886	262	0,15
			883	407	0,19
			880	640	0,11
rc103	942	799	965	537	0,08
			964	488	0,1
			957	788	0,09
rc104	1023	309	1047	103	0,33
			1043	62	0,46
			1041	256	0,35
rc105	840	50	869	233	0,11
			861	247	0,29
			857	137	0,3
rc106	851	164	901	84	0,09
			884	192	0,75
			883	801	0,06
rc107	955	521	961	257	0,17
			957	451	0,29
			956	65	0,52
rc108	966	169	1021	571	0,02
			1019	388	0,19
			1017	232	0,07
rc201	1717	31	1724	32	0,56
			1722	69	0,49
			1721	34	0,85
rc202	1724	27	1724	23	0,9
rc203	1724	21	1724	23	0,89

rc204	1724	21	1724	22	0,92
rc205	1724	29	1724	33	0,92
			1721	321	0,36
			1719	46	0,25
rc206	1724	21	1724	23	0,88
rc207	1724	21	1724	23	0,92
rc208	1724	21	1724	22	0,94

Αποτελέσματα για στιγμιότυπα του Cordeau για 1 διαδρομή

1 Tour

Τοπολογία	ILS		Συνδυαστικός		
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	Threshold (θ)
pr01	304	28	308	6	0,62
			306	31	0,56
			305	5	0,47
pr02	384	22	390	39	0,66
			389	68	0,63
			388	99	0,77
pr03	387	17	394	38	0,41
			393	90	0,56
			392	17	0,89
pr04	445	88	479	338	0,49
			469	81	0,82
			466	47	0,63
pr05	574	218	585	715	0,54
			583	780	0,88
			582	754	0,59
pr06	531	143	567	650	0,66
			561	1541	0,61
			560	955	0,87
pr07	290	28	298	59	0,61
			293	7	0,6
			291	12	0,85
pr08	442	332	463	139	0,86
			460	191	0,63
			452	422	0,88
pr09	457	72	491	454	0,59
			488	470	0,69

			487	358	0,68
pr10	544	560	570	465	0,58
			561	1299	0,36
			557	978	0,6
pr11	331	20	335	3	0,45
			334	13	0,36
			332	15	0,63
pr12	425	5	434	26	0,46
			433	187	0,61
			432	155	0,7
pr13	443	389	454	39	0,83
			452	319	0,57
			451	173	0,8
pr14	536	103	526	340	0,51
			524	434	0,4
			522	481	0,38
pr15	650	956	677	334	0,64
			675	938	0,59
			674	976	0,62
pr16	599	1667	632	76	0,9
			616	132	0,5
			615	135	0,47
pr17	360	87	360	4	0,58
			358	17	0,33
			357	5	0,41
pr18	454	282	534	113	0,41
			531	97	0,57
			529	133	0,47
pr19	519	558	535	198	0,97
			530	274	0,57
			523	870	0,7
pr20	590	451	620	135	0,48
			617	543	0,53
			616	328	0,52

Αποτελέσματα για στιγμιότυπα του Cordeau για 2 διαδρομές

2 Tours

Τοπολογία	ILS		Συνδυαστικός		
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	Threshold (θ)
pr01	485	179	493	37	0,67
			485	128	0,93
pr02	667	32	484	21	0,45
			702	313	0,72

			701	606	0,39
			696	129	0,54
pr03	701	45	730	771	0,79
			726	478	0,34
			723	78	0,84
pr04	865	353	891	719	0,43
			889	1151	0,45
			886	927	0,3
pr05	1027	2004	1069	1993	0,63
			1053	542	0,76
			1051	195	0,73
pr06	1007	2508	1027	833	0,58
			1003	2389	0,76
			997	950	0,85
pr07	553	154	566	82	0,55
			564	154	0,59
			561	145	0,48
pr08	786	61	813	529	0,72
			810	639	0,79
			809	222	0,34
pr09	828	1312	879	1477	0,52
			870	1585	0,54
			869	1860	0,78
pr10	1022	634	1082	411	0,99
			1077	2988	0,6
			1069	3628	0,61
pr11	537	29	556	160	0,37
			550	307	0,35
			548	45	0,89
pr12	724	17	747	923	0,39
			745	430	0,42
			742	992	0,41
pr13	781	348	816	90	0,63
			811	604	0,75
			810	1059	0,49
pr14	904	362	949	355	0,76
			940	1400	0,7
			939	2682	0,8
pr15	1091	333	1175	423	0,73
			1164	5519	0,4
			1163	3762	0,29
pr16	1029	220	1122	10234	0,35
			1115	2267	0,46
			1114	3626	0,71
pr17	626	80	640	374	0,31
			634	187	0,76
			633	751	0,38
pr18	864	315	941	1918	0,35
			931	2722	0,37
			930	1439	0,34

pr19	911	404	1023	4350	0,33
			1011	2564	0,79
			988	399	0,66
pr20	1148	5988	1168	2564	0,57
			1167	1063	0,5
			1161	2014	0,46

Αποτελέσματα για στιγμιότυπα του Cordeau για 3 διαδρομές

3 Tours

Τοπολογία	ILS		Συνδυαστικός		
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	Threshold (θ)
pr01	598	5	619	93	0,48
			614	131	0,41
			611	108	0,79
pr02	893	724	930	86	0,38
			927	866	0,12
			922	114	0,42
pr03	954	514	984	596	0,92
			981	2532	0,56
			980	280	0,6
pr04	1189	1621	1246	727	0,65
			1245	1993	0,53
			1232	1229	0,9
pr05	1391	3899	1419	5310	0,87
			1413	1119	0,33
			1408	1812	0,81
pr06	1364	3957	1416	7109	0,73
			1415	2983	0,67
			1411	6030	0,94
pr07	708	236	721	42	0,42
			720	332	0,76
			718	86	0,51
pr08	1073	82	1092	937	0,82
			1084	591	0,89
			1080	374	0,96
pr09	1186	575	1228	1038	0,34
			1209	2902	0,42
			1193	1331	0,64
pr10	1460	6737	1515	1549	0,58
			1508	1307	0,56
			1499	2312	0,47
pr11	635	9	649	156	0,32
			645	39	0,35
			644	23	0,57
pr12	921	764	966	2512	0,39

			962	301	0,36
			956	1509	0,38
pr13	1078	80	1109	2370	0,53
			1103	396	0,49
			1099	263	0,69
pr14	1279	3122	1302	3360	0,33
			1293	666	0,79
			1291	1101	0,6
pr15	1469	1483	1605	1528	0,29
			1600	10297	0,38
			1594	7517	0,23
pr16	1467	1437	1555	7642	0,32
			1554	2918	0,38
			1543	7429	0,35
pr17	809	20	831	276	0,43
			819	840	0,31
			815	208	0,35
pr18	1149	87	1238	444	0,39
			1234	1290	0,3
			1232	962	0,85
pr19	1286	3268	1399	569	0,28
			1381	4580	0,3
			1371	1328	0,33
pr20	1522	5405	1646	902	0,53
			1608	755	0,37
			1606	6346	0,35

Αποτελέσματα για στιγμιότυπα του Cordeau για 4 διαδρομές

4 Tours

Τοπολογία	ILS		Συνδυαστικός		
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	Threshold (θ)
pr01	654	70	654	9	0,66
			652	27	0,43
			649	9	0,89
pr02	1011	59	1040	978	0,71
			1037	507	0,93
			1035	1295	0,8
pr03	1125	644	1174	1841	0,83
			1165	3150	0,44
			1164	2915	0,95
pr04	1430	340	1516	3426	0,75
			1515	6351	0,78
			1498	590	0,52
pr05	1691	7947	1740	3093	0,45
			1738	2658	0,76

			1737	3097	0,34
pr06	1704	3056	1800	1363	0,83
			1772	5917	0,87
			1765	4706	0,45
pr07	816	323	846	117	0,91
			837	41	0,86
			834	57	0,75
pr08	1264	92	1313	161	0,8
			1309	2202	0,95
			1307	153	0,89
pr09	1504	667	1516	513	0,47
			1509	2905	0,57
			1506	3194	0,75
pr10	1762	598	1841	7331	0,5
			1840	11747	0,25
			1839	9594	0,75
pr11	657	55	657	9	0,48
			654	5	0,9
pr12	1041	28	1088	145	0,44
			1086	944	0,14
			1084	631	0,42
pr13	1242	1048	1337	1686	0,3
			1336	1685	0,32
			1322	2208	0,6
pr14	1531	2152	1602	5786	0,8
			1597	1653	0,49
			1593	2127	0,01
pr15	1818	155	1975	1707	0,31
			1964	3506	0,41
			1957	1103	0,32
pr16	1805	1207	1937	5122	0,48
			1932	17993	0,38
			1917	18457	0,33
pr17	899	32	915	155	0,65
			911	441	0,39
			907	1268	0,31
pr18	1372	1624	1487	582	0,79
			1455	3435	0,36
			1454	2216	0,42
pr19	1550	631	1687	1326	0,64
			1676	5906	0,32
			1669	4726	0,27
pr20	1856	6563	1967	2225	0,33
			1965	1386	0,5
			1956	5248	0,37

Αποτελέσματα για στιγμιότυπα t1*

Τοπολογία	ILS		Συνδυαστικός		
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	Threshold (θ)
t101	377	61	395	23	0,04
			392	75	0,33
			389	69	0,02
t102	717	39	780	138	0,5
			779	478	0,75
			777	663	0,87
t103	803	797	822	410	0,46
			820	1689	0,02
			819	909	0,43
t104	756	122	789	71	0,27
			786	126	0,05
			781	1142	0,07
t105	423	41	431	293	0,01
			429	52	0,29
			428	30	0,31
t106	1167	175	1197	3194	0,45
			1191	818	0,59
			1189	632	0,99
t107	768	105	792	229	0,22
			789	1013	0,03
			788	1166	0,02
t108	712	523	724	703	0,39
			722	381	0,71
			721	196	0,41
t109	1095	581	1108	394	0,41
			1107	580	0,59
			1106	268	0,13
t110	783	324	827	376	0,93
			824	46	0,06
			823	231	0,23
t111	817	578	824	921	0,5
			822	453	0,78
			819	562	0,06
t112	800	61	822	537	0,32
			820	393	0,26
			819	690	0,37
t113	1104	464	1103	3084	0,03

			1102	678	0,19
			1099	1826	0,59
t114	450	6	491	148	0,02
			484	215	0,06
			483	148	0,01
t115	1042	379	1059	140	0,59
			1058	194	0,56
			1057	43	0,12
t116	840	27	868	1199	0,04
			866	777	0,4
			864	775	0,03
t117	447	203	463	449	0,01
			459	120	0,43
			458	134	0,02
t118	1152	1446	1173	1496	0,3
			1169	2133	0,45
			1168	1433	0,71
t119	1189	931	1194	206	0,21
			1192	1947	0,94
			1190	2096	0,32
t120	1008	408	1024	2485	0,31
			1019	1289	0,51
			1018	1755	0,34
t121	414	27	449	265	0,64
			445	411	0,72
			440	215	0,39
t122	471	117	481	17	0,04
			478	116	0,03
			476	207	0,48
t123	402	27	430	33	0,02
			418	39	0,04
			414	20	0,11
t124	435	2	477	157	0,39
			476	23	0,62
			472	59	0,02
t125	1182	528	1195	630	0,52
			1194	5552	0,19
			1193	1844	0,04
t126	383	6	430	331	0,04
			428	142	0,02
			421	71	0,03
t127	1013	252	1060	1503	0,07
			1056	621	0,63
			1054	200	0,44
t128	1079	53	1123	872	0,35
			1122	444	0,3
			1119	1698	0,23
t129	441	16	442	133	0,39
			441	12	0,86
			439	176	0,03

t130	812	50	848	1081	0,01
			841	953	0,14
			836	248	0,04
t131	401	36	424	255	0,01
			419	63	0,86
			418	37	0,04
t132	423	246	449	280	0,02
			437	142	0,07
			434	136	0,06
t133	778	34	826	572	0,43
			820	65	0,55
			819	116	0,31
t134	1226	1655	1253	276	0,02
			1252	1234	0,43
			1251	2116	0,04
t135	796	752	836	467	0,71
			829	2111	0,44
			828	609	0,74
t136	755	30	798	659	0,03
			790	555	0,08
			788	453	0,22
t137	1073	328	1127	3079	0,06
			1125	7968	0,03
			1124	6979	0,02
t138	1255	1490	1254	1478	0,89
			1245	2215	0,72
			1244	164	0,83
t139	1175	996	1182	271	0,59
			1180	300	0,37
			1179	1058	0,58
t140	1009	894	1044	415	0,36
			1041	521	0,8
			1039	466	0,5
t141	731	280	748	155	0,67
			746	682	0,11
			745	120	0,43
t142	1161	503	1232	2474	0,14
			1225	2131	0,77
			1219	833	0,42
t143	415	134	422	348	0,05
			416	45	0,02
			415	35	0,37
t144	768	769	770	699	0,41
			769	664	0,3
			768	387	0,01
t145	372	174	381	24	0,01
			379	379	0,03
			373	82	0,32
t146	781	1047	792	1424	0,4
			791	164	0,06

			789	480	0,01
t147	1103	728	1109	2745	0,06
			1104	1298	0,43
			1103	326	0,67
t148	468	6	475	172	0,34
			474	17	0,33
			473	173	0,05
t149	1077	613	1112	2112	0,76
			1106	1489	0,85
			1105	563	0,63
t150	479	141	487	21	0,03
			486	101	0,02
			485	11	0,04

 Αποτελέσματα για στιγμιότυπα t2*

Τοπολογία	ILS		Συνδυαστικός		
	Κέρδος	Χρόνος Εκτέλεσης	Κέρδος	Χρόνος Εκτέλεσης	Threshold (θ)
t201	178	8	190	10	0,04
			182	1	0,5
			180	9	0,03
t202	193	10	193	3	0,31
			191	8	0,46
			180	1	0,36
t203	174	6	179	5	0,03
			178	3	0,37
			174	5	0,8
t204	153	3	176	7	0,2
			173	28	0,04
			171	16	0,1
t205	435	13	453	138	0,38
			449	56	0,3
			447	26	0,66
t206	535	125	586	240	0,4
			585	53	0,24
			584	428	0,11
t207	174	1	200	18	0,14
			195	28	0,22
			182	8	0,15
t208	162	3	176	3	0,15
			175	2	0,22
			162	1	0,58
t209	426	19	467	443	0,04
			466	601	0,05
			465	293	0,26

t210	441	22	508	504	0,42
			507	1637	0,12
			506	343	0,41
t211	484	316	491	631	0,23
			489	216	0,34
			487	188	0,4
t212	458	3	477	255	0,28
			475	91	0,34
			472	107	0,24
t213	495	315	507	254	0,23
			506	110	0,26
			505	260	0,21
t214	316	36	324	76	0,28
			322	34	0,1
			321	34	0,29
t215	420	16	440	87	0,46
			439	116	0,33
			436	19	0,59
t216	463	61	485	498	0,02
			482	201	0,11
			480	316	0,07
t217	460	20	470	111	0,62
			467	609	0,23
			466	44	0,64
t218	155	1	155	0	0,74
			154	1	0,26
			153	1	0,69
t219	443	7	511	297	0,25
			509	1254	0,52
			508	159	0,24
t220	304	2	349	60	0,31
			348	101	0,25
			347	36	0,29
t221	279	127	311	649	0,1
			310	390	0,24
			306	231	0,12
t222	396	117	406	690	0,16
			401	69	0,04
			400	88	0,34
t223	183	1	229	26	0,21
			222	11	0,2
			221	34	0,08
t224	401	107	417	265	0,05
			416	715	0,08
			414	69	0,34
t225	545	519	564	789	0,44
			561	711	0,12
			560	546	0,21
t226	541	9	604	806	0,38
			600	667	0,03

			599	1198	0,18
t227	159	2	159	1	0,78
			156	6	0,17
			153	2	0,38
t228	518	22	547	200	0,28
			546	209	0,27
			545	271	0,37
t229	166	2	178	3	0,34
			169	6	0,15
			166	1	0,42
t230	290	84	296	67	0,34
			295	54	0,36
			294	12	0,4
t231	488	7	503	22	0,23
			501	195	0,26
			499	107	0,78
t232	501	215	561	806	0,21
			559	2120	0,11
			557	1086	0,19
t233	170	7	201	28	0,16
			192	34	0,18
			191	14	0,25
t234	467	25	519	717	0,55
			518	582	0,08
			516	318	0,27
t235	484	8	506	130	0,41
			502	89	0,38
			501	352	0,22
t236	159	2	180	6	0,08
			176	10	0,14
			174	6	0,13
t237	461	159	485	373	0,19
			484	199	0,54
			483	50	0,52
t238	527	68	532	946	0,26
			529	203	0,41
			528	768	0,29
t239	498	325	514	286	0,35
			512	434	0,25
			510	796	0,27
t240	269	4	335	56	0,1
			334	147	0,01
			332	48	0,16
t241	170	2	172	1	0,13
			171	5	0,15
			170	0	0,55
t242	171	5	175	3	0,12
			173	3	0,17
			171	4	0,59
t243	556	45	560	952	0,17

			558	423	0,24
			556	197	0,78
t244	331	14	345	75	0,25
			343	47	0,13
			342	44	0,24
t245	291	2	299	68	0,4
			294	10	0,25
			293	40	0,33
t246	451	132	486	354	0,09
			485	689	0,19
			478	299	0,33
t247	429	4	460	413	0,32
			459	151	0,38
			454	307	0,36
t248	458	1013	479	917	0,12
			477	604	0,08
			476	175	0,04
t249	431	43	447	307	0,23
			446	70	0,18
			444	438	0,08
t250	195	22	201	36	0,06
			200	6	0,32
			199	14	0,08
