Μελέτη και υλοποίηση ευρετικών μεθόδων για το πρόβλημα δρομολόγησης οχημάτων με περιορισμό χωρητικότητας

Αρβανίτης Συμεών

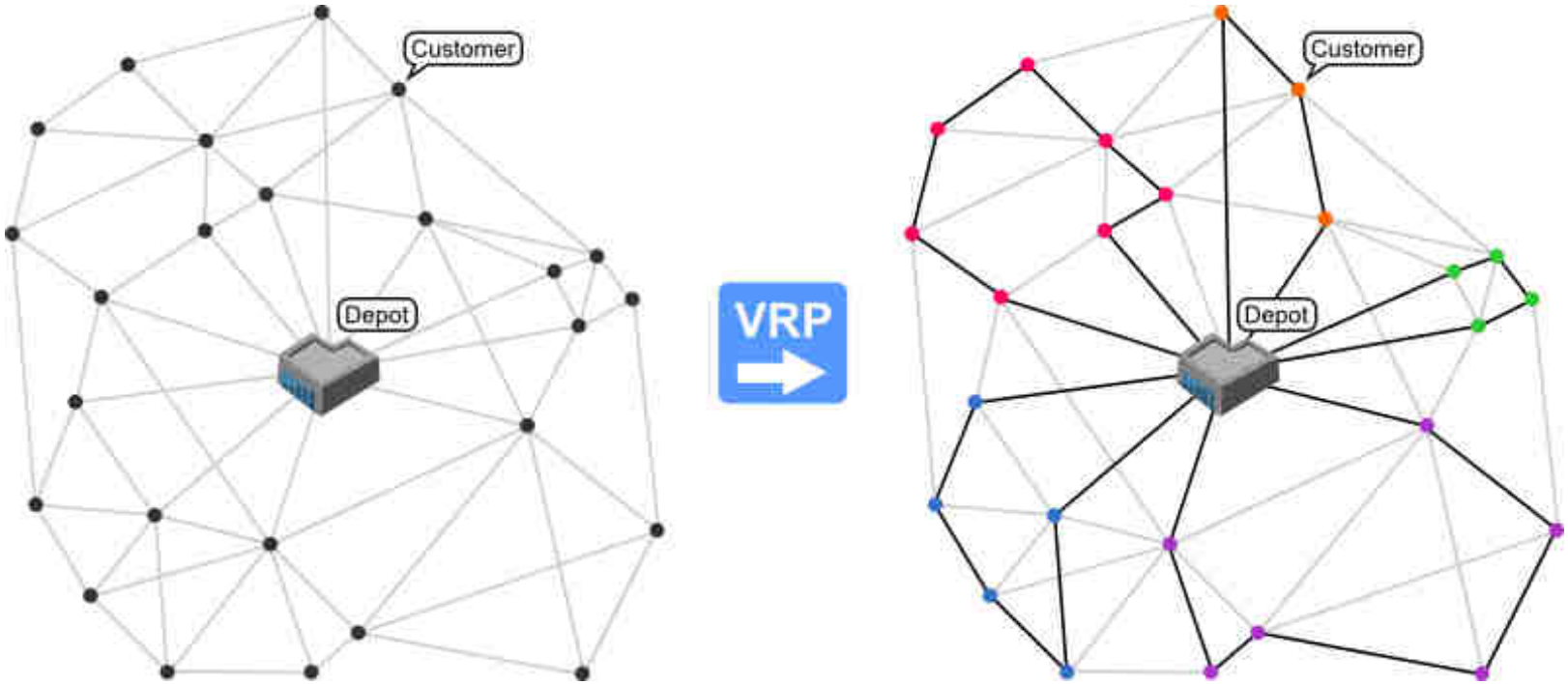
Πανεπιστήμιο Μακεδονίας, Σχολή Επιστημών Πληροφορίας
Τμήμα Εφαρμοσμένης Πληροφορικής, Εγνατία 156, 54636 Θεσσαλονίκη

25 Ιουνίου, 2019

VRP

- Πρόβλημα ελαχιστοποίησης κόστους.
- Δρομολόγηση οχημάτων σε ένα δίκτυο με σκοπό την εξυπηρέτηση όλων των πελάτών με το λιγότερο δυνατό κόστος.
- Συνάρτηση κόστους είναι η συνολική απόσταση που θα διανύσει ο στόλος των οχημάτων.
- Κάθε κόμβος ζήτησης πρέπει να εξυπηρετηθεί ακριβώς μια φορά από ένα όχημα.
- Τα οχήματα ξεκινούν από μια κεντρική αποθήκη και στο τέλος της διαδρομής τους επιστρέφουν σε αυτή.

VRP



Σχήμα: Παράδειγμα ενός vrp

CVRP, TSP

- Capacitated VRP: Κάθε πελάτης ζητά έναν αριθμό από αγαθά
- Όλα τα οχήματα έχουν συγκεκριμένη χωρητικότητα αγαθών

- TSP:Μόνο ένα όχημα στη διάθεση μας. Ολοι πρέπει να εξυπηρετηθούν από αυτό.
- Ψάχνουμε τον βέλτιστο κύκλο Hamilton

VRPTW, MDVRP

- VRP with Time Windows:
- Κάθε πελάτης πρέπει να εξυπηρετηθεί μέσα σε ένα προκαθορισμένο χρονικό περιθώριο

- Multi Depot VRP:
- Υπάρχουν πολλές αποθήκες. Κάθε κόμβος πρέπει να ανατεθεί σε μια αποθήκη και έτσι προκύπτουν πολλά μικρότερα CVRP

Nearest Neighbor

- Ο αλγόριθμος του εγγύτερου γείτονα επισκέπτεται τον κόμβο που βρίσκεται πιο κοντά στην τρέχουσα θέση αν μπορεί να τον εξυπηρετήσει.
- Η διαδρομή ολοκληρώνεται όταν δεν υπάρχουν κόμβοι που να μπορούμε να εξυπηρετήσουμε την υπολειπόμενη χωρητικότητα και αν υπάρχουν κόμβοι προς εξυπηρέτηση ξεκινά νέα διαδρομή.

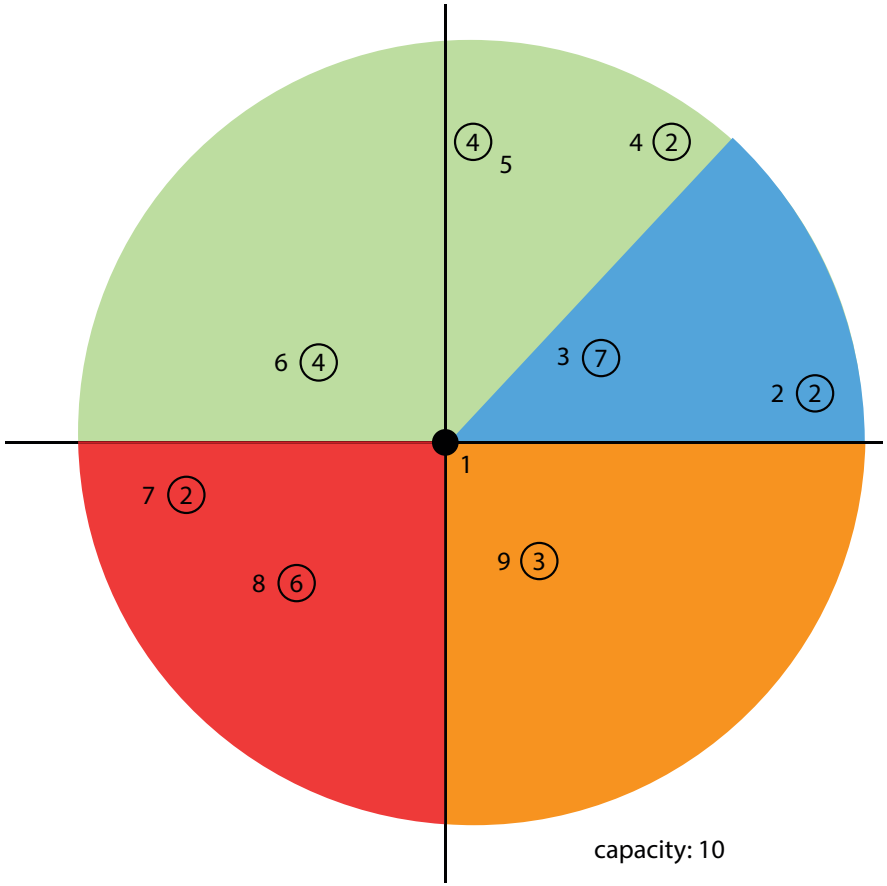
- Πολύ γρηγορότερος σε σχέση με τους άλλους αλγορίθμους κατασκευής που εξετάστηκαν.
- Δεν δίνει ποιοτικές λύσεις αφού το μέσο σφάλμα ξεπερνά το 35%.

Sweep

- Ο αλγόριθμος σάρωσης επιχειρεί να συσταδοποιήσει τους κόμβους ζήτησης. Αυτό γίνεται σαρώνοντας το επίπεδο με τον περιστρεφόμενο άξονα x.
- Υπολογίζεται η γωνία που σχηματίζει κάθε κόμβος με τον άξονα x.

- Πιο αργός από τον αλγόριθμο Nearest Neighbor αλλα με πολύ ποιοτικότερα αποτελέσματα.
- Πολλές φορές δεν προβλέπει σωστά τον αριθμό των οχημάτων που θα χρειαστούν.

Sweep



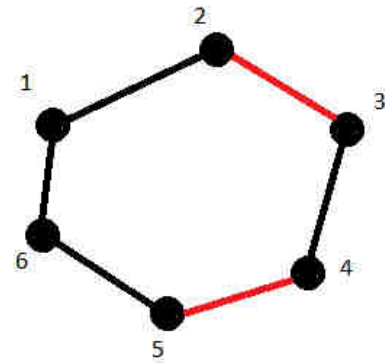
Σχήμα: Συσταδοποίηση με τον αλγόριθμο σάρωσης

Savings algorithm

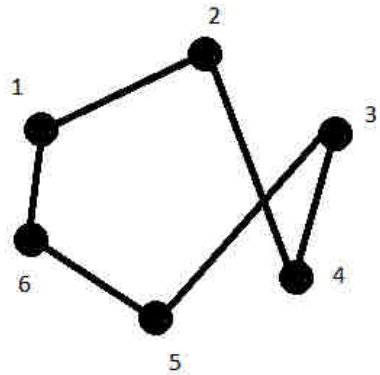
- Προτάθηκε από τους Clarke και Wright
- Για όλες τις πιθανές ακμές υπολογίζεται ο συντελεστής εξοικονόμησης
- $S_{ij} = D_{i0} + D_{j0} - D_{ij}$
- Ο συντελεστής μας δείχνει πόσο μας συμφέρει η εισαγωγή μιας ακμής
- Η λύση κατασκευάζεται βάζοντας τις όσο το δυνατον πιο συμφέρουσες ακμές
- Δίνει τα καλύτερα αποτελέσματα αλλα είναι η πιο αργή από τις 3 μεθόδους

2-opt

- Αφαιρούμε 2 ακμές και η διαδρομή χωρίζεται σε 2 κομμάτια.
- Επανασυνδέουμε χωρίς να χρησιμοποιηθούν οι ακμές που αφαιρέθηκαν.



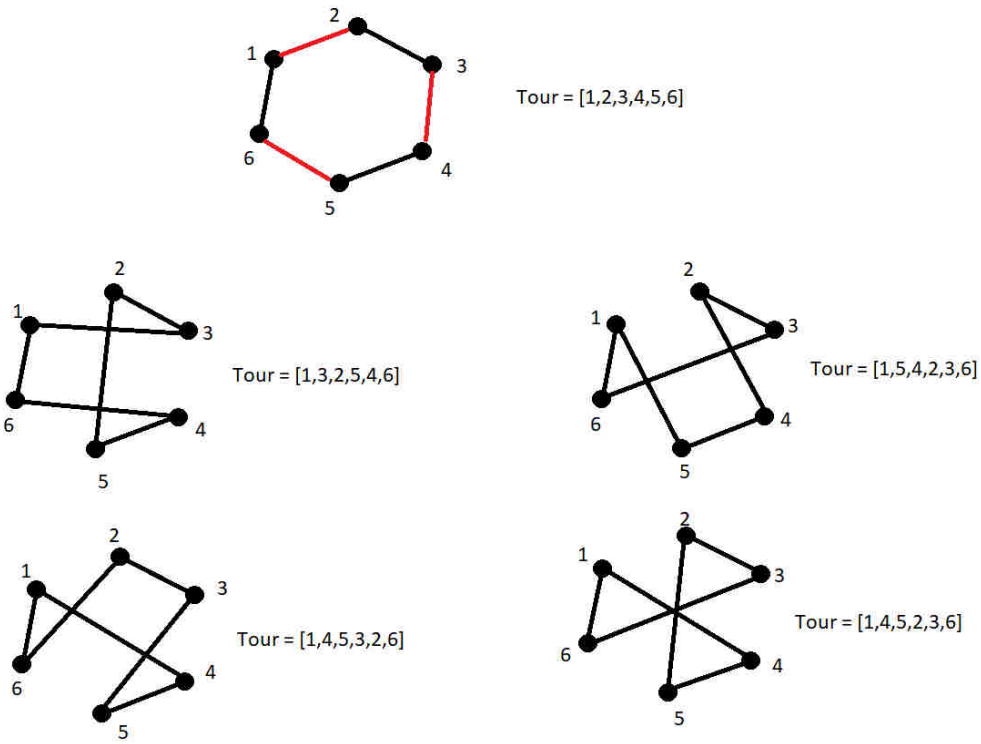
Tour = [1,2,3,4,5,6]



Tour = [1,2,4,3,5,6]

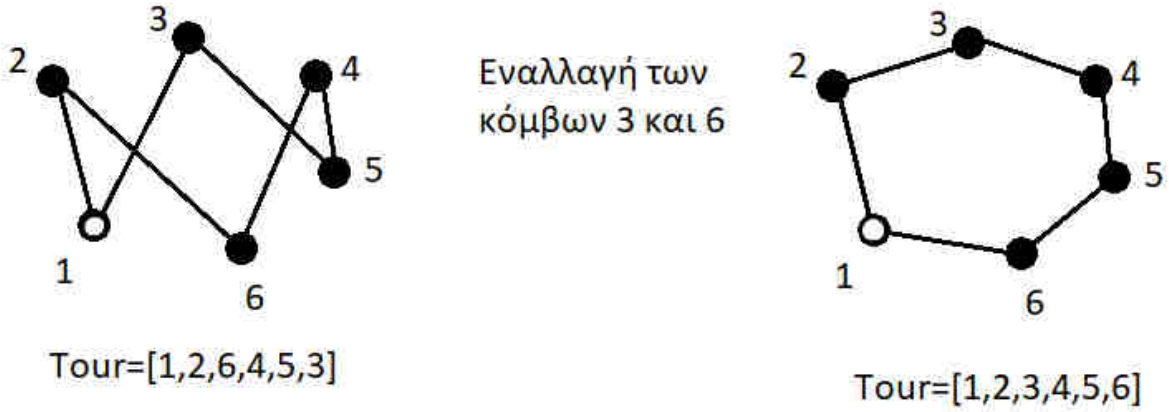
3-opt

- Αφαιρούμε 3 ακμές και η διαδρομή χωρίζεται σε 3 κομμάτια.
- Επανασυνδέουμε χωρίς να χρησιμοποιηθούν οι ακμές που αφαιρέθηκαν.
- Αν οι ακμές που αφαιρέσαμε δεν είναι συνεχόμενες έχουμε 4 τρόπους επανασύνδεσης



Swap

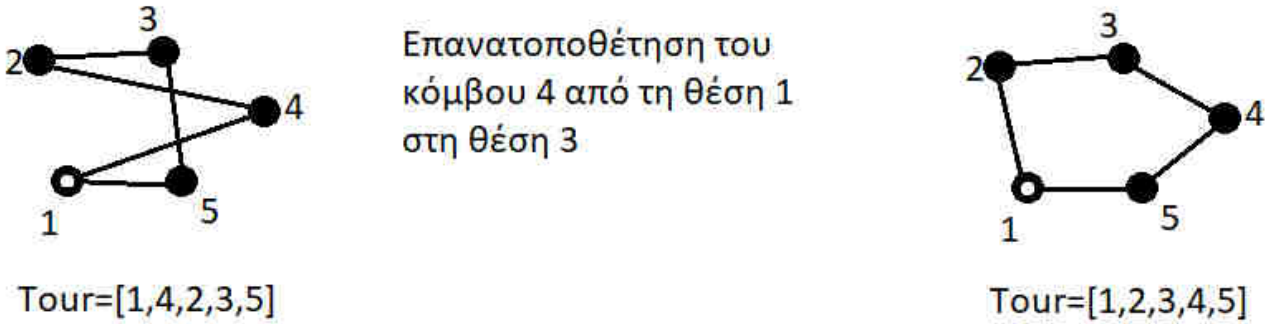
- Ανταλλαγή των θέσεων 2 κόμβων.
- Μπορεί να γίνει μέσα στην ίδια διαδρομή ή μεταξύ διαφορετικών διαδρομών όμως πρέπει να είναι εφικτό λόγω του περιορισμού χωρητικότητας



Σχήμα: Ανταλλαγή μέσα στην ίδια διαδρομή

Relocate

- Επανατοποθέτηση ενός κόμβου σε ένα άλλο σημείο.
- Μπορεί επίσης να γίνει μέσα στην ίδια διαδρομή ή μεταξύ διαφορετικών διαδρομών όμως πρέπει να είναι εφικτό λόγω του περιορισμού χωρητικότητας



Σχήμα: Επανατοποθέτηση κόμβου μέσα στην ίδια διαδρομή

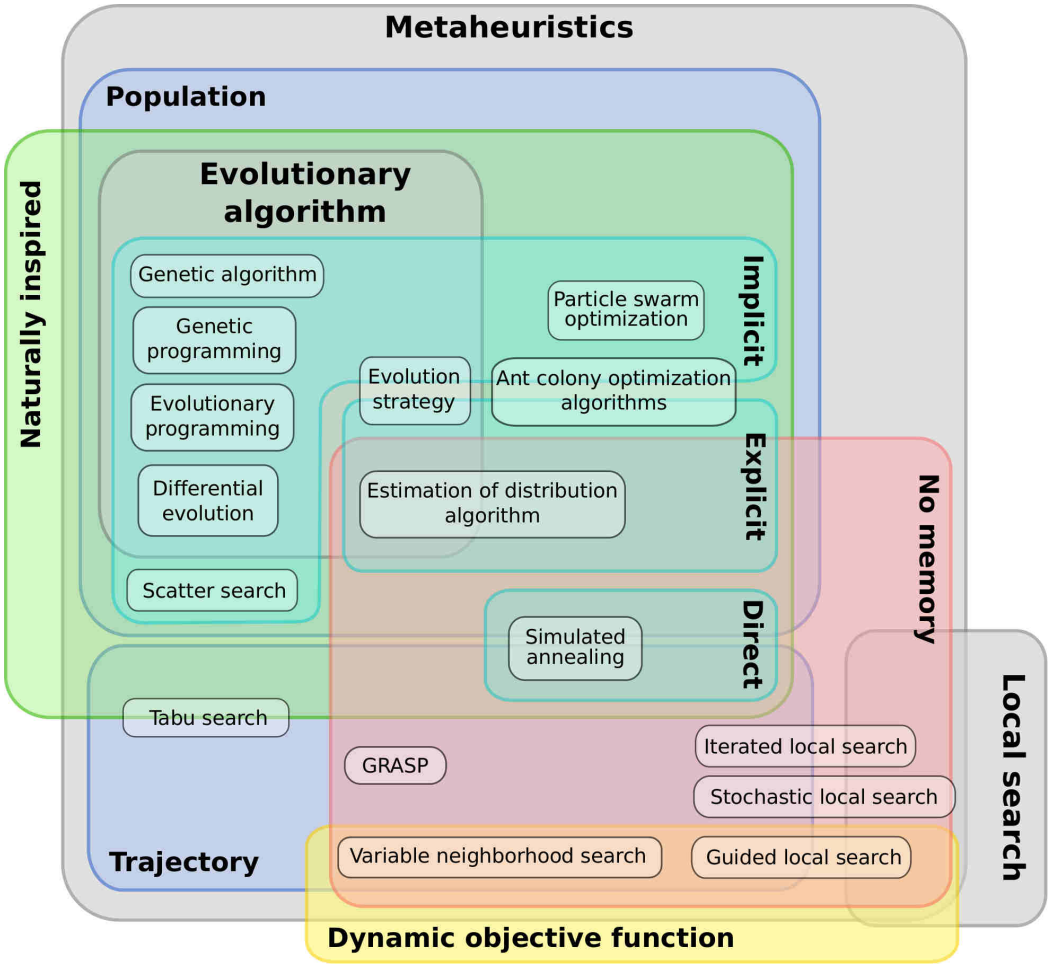
First vs Best improvement

- Οι μέθοδοι τοπικής αναζήτησης μπορούν να υλοποιηθούν είτε με την πρώτη βελτίωση είτε με την καλύτερη βελτίωση.
- Στην πρώτη βελτιώση επιστρέφεται η πρώτη βελτίωση που εντοπίζει η τοπική αναζήτηση.
- Στην καλύτερη βελτίωση η γειτονιά που επιλέγεται διερευνείται διεξοδικά και επιστρέφεται η καλύτερη από τις βελτιώσεις που βρέθηκαν.
- Ενδείκνυται να χρησιμοποιείται η πρώτη βελτίωση όταν ξεκινάμε από τυχαία λύση.
- Όταν ξεκινάμε με μια καλής ποιότητας λύση τότε η καλύτερη βελτίωση έχει καλύτερα αποτελέσματα.

About

- Οι μεθερευρετικές είναι γενικά πλαίσια έρευνας.
- Μπορούν με λίγες τροποποιήσεις να εφαρμοστούν σε ένα ευρύ φάσμα προβλημάτων βελτιστοποίησης.
- Δεν μας εξασφαλίζουν την εύρεση του ολικού βέλτιστου.
- Στόχος τους είναι η εξερεύνηση του χώρου των εφικτών λύσεων και η εύρεση μιας όσο το δυνατόν καλύτερης λύσης μέσα σε ένα λογικό χρονικό περιθώριο.
- Συνήθως είναι στοχαστικές δηλαδή περιέχουν τυχαία βήματα και κατευθύνσεις βελτιστοποίησης.

Classification



Σχήμα: Κατηγοριοποίηση μεθευρετικών μεθόδων

About

- Η VNS επιχειρεί την βελτιστοποίηση με χρήση των τελεστών τοπικής αναζήτησης που ορίζει ο χρήστης.
- Στη συνέχεια όταν εγκλωβιστεί επιχειρεί τυχαίες επιτρεπτές αλλαγές στην υπάρχουσα λύση.
- Έτσι μεταπηδά σε μια άλλη περιοχή λύσεων και επιχειρεί πάλι βελτιστοποίηση.
- Κριτήριο τερματισμού είναι συνήθως ο χρόνος εκτέλεσης που θα ορίσει ο χρήστης.

Variations

- Η BVNS χρησιμοποιεί μια γειτονιά κάθε φορά.
- Η GVNS βελτιστοποιεί χρησιμοποιώντας πολλές γειτονιές με συγκεκριμένη σειρά αιτιοκρατικά.
- Η RVNS χρησιμοποιεί μόνο τυχαίες αλλαγές και όχι βελτιστοποίηση.
- Η SVNS επιχειρεί να μεταπηδήσει σε πολύ μακρινές περιοχές.

- Το στιγμιότυπο διαβάζεται και τα δεδομένα του μετατρέπονται σε πίνακες.
- Ξεκινά η κατασκευή της αρχικής λύσης με τον αλγόριθμο εξοικονόμησης.
- Ο χρήστης ορίζει το χρονικό διάστημα που θα εκτελεστεί ο αλγόριθμος.

- Ο αλγόριθμος GVNS ξεκινά τη διαδικασία shaking με ένα τυχαίο βήμα.
- Στη συνέχεια επιχειρείται βελτιστοποίηση με VND
- Η VND βελτιστοποιεί εφαρμόζοντας με τη σειρά τοπική αναζήτηση 2-opt,swap,relocate. Εάν βρει βελτίωση ξεκινά από την αρχή στις γειτονιές αυτές με την ίδια σειρά.
- Εάν δεν βρεί τότε γίνεται ξανα shaking με ένα τυχαίο βήμα παραπάνω από την προηγούμενη φορά.

- Δοκιμάστηκε ένα πλήθος 84 στιγμιοτύπων.
- Εκτελέστηκαν όλοι οι αλγόριθμοι κατασκευής και καταγράφηκε η επίδοσή τους.
- Η GVNS δοκιμάστηκε για εκτέλεση 30' και 60' καταγράφηκαν οι επιδόσεις του αλγορίθμου για μέσο όρο δέκα εκτελέσεων σε κάθε περίπτωση.

- Διάσταση <40 : 30sec $\rightarrow <$ 1.5%
- Διάσταση 40-50 : 30sec $\rightarrow$ 3.2%, 60sec $\rightarrow$ 2.8%.
- Διάσταση 50-60 : 30sec $\rightarrow$ 4.5%, 60sec $\rightarrow$ 3.6%.
- Διάσταση 60-70 : 30sec $\rightarrow$ 5.6%, 60sec $\rightarrow$ 4.7%.
- Διάσταση >70 : 30sec $\rightarrow$ 6.4%, 60sec $\rightarrow$ 5.7%.

- Η τιμή του σχετικού σφάλματος για σταθερό χρόνο εκτέλεσης εξαρτάται από την διάσταση του προβλήματος.
- Όσο πιο μεγάλη διάσταση έχει το στιγμιότυπο τόσο περισσότερο χρόνο χρειάζεται για να λυθεί.
- Παρατηρούμε ότι το σφάλμα πέφτει αρκετά στα πρώτα 30 δευτερόλεπτα εκτέλεσης της GVNS.
- Αυξάνοντας τον χρόνο εκτέλεσης το σφάλμα συνεχίζει να μειώνεται αλλά με μικρότερο ρυθμό.
- Η εκτέλεση του αλγορίθμου για μεγάλο χρονικό διάστημα δεν εγγυάται την εύρεση του βέλτιστου.

Τέλος της παρουσίασης

Ευχαριστώ για την προσοχή σας!