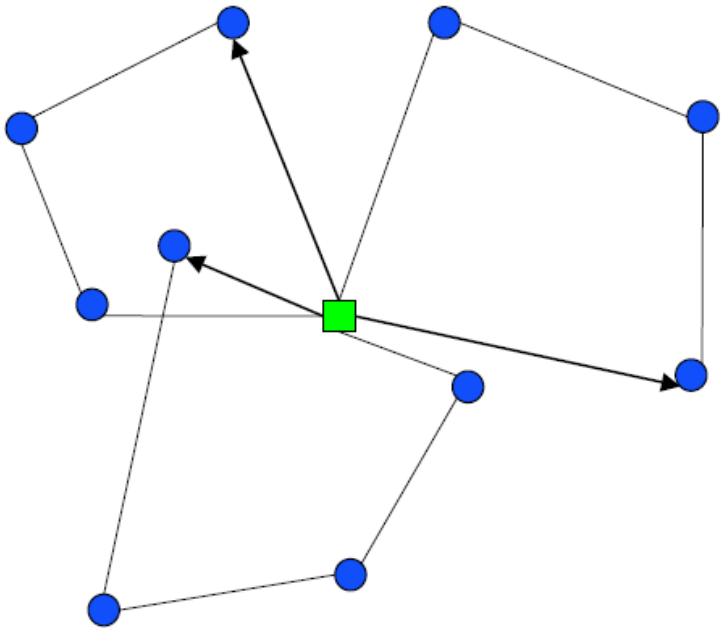
Ε Θ Ν Ι Κ Ο Μ Ε Τ Σ Ο Β Ι Ο Π Ο Λ Υ Τ Ε Χ Ν Ε Ι Ο

ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ ΦΟΡΤΗΓΩΝ ΟΧΗΜΑΤΩΝ ΜΕ ΧΡΟΝΙΚΑ ΠΑΡΑΘΥΡΑ ΚΑΙ ΧΩΡΗΤΙΚΟΤΗΤΑ (VRPTW)



Ιούλιος 2010

Γεώργιος Δ. Λιγνός
Α.Μ. 05104655

Υπεύθυνος καθηγητής:
Χρήστος Κυρανούδης

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία πραγματοποιήθηκε στα πλαίσια της Διπλωματικής Διατριβής. Το αντικείμενο της εργασίας αφορά στο πρόβλημα δρομολόγησης φορτηγών οχημάτων με χρονικά παράθυρα και χωρητικότητα, ένα σύγχρονο μοντέλο πραγματικών προβλημάτων με ευρεία εφαρμογή. Προκειμένου να βρεθεί η βέλτιστη λύση, απαιτείται μεγάλο υπολογιστικό κόστος, χρονικά ασύμφορο. Σκοπός είναι να παράγονται λύσεις με ευρετικούς ή μεταευρετικούς αλγορίθμους, τακτικές που παράγουν καλές αλλά όχι βέλτιστες λύσεις, σε πολύ μικρό σχετικά χρόνο. Οι ευερετικές προσεγγίσεις απαρτίζονται από δύο συνήθως μέρη. Το κατασκευαστικό μέρος παράγει την αρχική λύση και το βελτιωτικό εξετάζει τη γειτονιά της αρχικής λύσης αναζητώντας αλλαγές που επιφέρουν κέρδος. Η ποιότητα λύσης των ευρετικών είναι περιορισμένη. Οι μεταευρετικές προσεγγίσεις έχουν δύο χαρακτηριστικά: Λειτουργούν μέσα σε ένα λογικό πλαίσιο, έτσι ώστε να μαθαίνουν από τα αποτελέσματα που παράγουν, και στη συνέχεια να καθοδηγούν την έρευνα σε περιοχές καλής ποιότητας λύσης. Έχουν γενική εφαρμογή σε πολλών ειδών προβλήματα, χωρίς να χρειάζεται ιδιαίτερη προσαρμογή σε κάθε είδος προβλήματος. Για την αξιολόγηση των αλγορίθμων χρησιμοποιούνται συνήθως τα ακαδημαϊκά προβλήματα του Solomon. Τέλος παρουσιάζεται ένας νέος αλγόριθμος insertion (L0) και συγκρίνεται με έναν άλλον insertion τον I1. Η λύση που παράγεται, βελτιώνεται με τρεις χειριστές. Αποδεικνύεται πως ο κατασκευαστικός L0 παράγει αποτελέσματα αντίστοιχης ποιότητας, με αυτήν του I1, πιο γρήγορα. Επίσης συμπεραίνεται πως ανάλογα με τον τρόπο που γίνεται η εναλλαγή στους χειριστές: 1-relocate 2-change, επηρεάζεται η ταχύτητα παραγωγής της λύσης.

Περιεχόμενα

ΕΙΣΑΓΩΓΗ

Ο χημικός μηχανικός το 2002 στην Ελλάδα υστερούσε έναντι των υπολοίπων μηχανικών ως προς τον χρόνο που μεσολαβούσε μέχρι την πρώτη απασχόληση[i]. Η ελπίδα της «παραδοσιακής» πρόσληψης στη βιομηχανία ολοένα και ελαττώνεται, ιδιαίτερα αυτήν την εποχή της οικονομικής κρίσης, ταυτόχρονα όμως ανοίγονται νέοι, διαφορετικοί ορίζοντες, κυρίως τεχνικοοικονομικοί[ii]. Η τωρινή θεματολογία των μαθημάτων στο τμήμα Χημικής Μηχανικής στο ΕΜΠ, δίνει τη δυνατότητα και την αφορμή στον σπουδαστή, να ασχοληθεί με τομείς στους οποίους μπορεί να αποδώσει ακόμα και αν δεν συνδέονται άμεσα με το εξ ορισμού αντικείμενό του. Μέσα από την ευρεία και απαιτητική θεματολογία, αναπτύσσεται η δημιουργικότητά του. Χαρακτηριστικό παράδειγμα είναι η φήμη πως χημικοί μηχανικοί με MBA προτιμώνται σε σχέση με τους οικονομολόγους σε διοικητικές οικονομικές θέσεις, καθώς έχουν πιο πρακτικό τρόπο σκέψης[iii].

Εξετάζοντας τα υποψήφια αντικείμενα για την εκπόνηση της διπλωματικής μου εργασίας, οδηγήθηκα στην επιλογή υπολογιστικού θέματος ώστε να είναι δυνατή η εργασία από το σπίτι. Το εν λόγω αντικείμενο είναι κατανοητό, ενδιαφέρον και απαιτεί δημιουργικότητα. Θεώρησα λοιπόν πως το καταλληλότερο, για μένα, θέμα είναι το Πρόβλημα Δρομολόγησης Φορτηγών με Χρονικά Παράθυρα VRPTW (Vehicle Routing Problem with Time Windows). Η ενασχόληση με αυτό το αντικείμενο, καλλιέργησε το προγραμματιστικό μου ταλέντο, δίνοντας μου πλεονέκτημα έναντι συναδέλφων σε πολύπλοκα προβλήματα διακριτής βελτιστοποίησης.

Επί της ουσίας, πρόκειται για την επίλυση ενός μοντέλου σύγχρονων πραγματικών προβλημάτων δρομολόγησης, που παρουσιάζονται στην καθημερινότητα κυρίως όσον αφορά την εφοδιαστική διαχείριση. Μάλιστα όσο κι αν βελτιωθεί η τεχνολογία (π.χ. χωρητικότητα φορτηγών, καλύτερα οδικά δίκτυα, ζωντανή παρακολούθηση της κίνησης κλπ) ή η θέση της εκάστοτε «αποθήκης», το πρόβλημα θα παραμείνει το ίδιο, δηλαδή η δρομολόγηση. Οι σημερινές εφαρμογές των προβλημάτων αυτών αφορούν κυρίως ταχυδρομικές και τραπεζικές μεταφορές, συλλογή βιομηχανικών αποβλήτων, δρομολόγηση τηλεφωνημάτων, τροφίμων, πτήσεων, σχολικών λεωφορείων. Ιδιαίτερα οι μεταφορές εμπορευμάτων, είναι σήμερα μία από τις πιο σημαντικές δραστηριότητες, η οποία δεσμεύει μεγάλα ποσοστά του Ελληνικού και ξένου προϋπολογισμού. Η ανάπτυξη καλύτερων αλγορίθμων επίλυσης του προβλήματος VRPTW οδηγεί στην εξοικονόμηση πόρων για τη βιομηχανία και τις μεταφορές, άρα και στη μείωση του τελικού κόστους προϊόντος. Ενδεικτικά εκτιμάται ότι στον τομέα των ποτών και των τροφίμων, οι μεταφορές αντιστοιχούν στο 70% της επιπρόσθετης αξίας τους[iv]. Μία καλύτερη διαχείριση του διαθέσιμου στόλου φορτηγών οδηγεί επίσης στην αποσυμφόρηση των δρόμων και σε καθαρότερο περιβάλλον λόγω της μείωσης των εκπομπών καυσαερίων.

---

[i] ΕΡΕΥΝΑ ΓΙΑ ΤΗΝ ΑΠΑΣΧΟΛΗΣΗ ΤΩΝ ΔΙΠΛΩΜΑΤΟΥΧΩΝ ΜΗΧΑΝΙΚΩΝ, Επιστ. Ευθύνη: Εργαστήριο Βιομηχανικής & Ενεργειακής Οικονομίας ΕΜΠ ΑΘΗΝΑ 2005 σελ 77

[ii] Βλ. σημείωση (1) Σελ 35 ΠΙΝΑΚΑΣ 2

[iii] NTUA Seminar, P&G 31 Μαρτίου 2009

[iv] Golden, B. L., E. A. Wasil. 1987. Computerized vehicle routing in the soft drink industry. *Oper.Res.* 35 6–17

Ορισμός Προβλήματος

Από την εποχή του 1800 μαθηματικοί όπως ο Ιρλανδός W. R. Hamilton και ο Βρετανός Thomas Kirkman επιχείρησαν να λύσουν προβλήματα παρόμοια με αυτό του περιοδεύοντος πωλητή (travelling salesman problem, TSP). Τη δεκαετία του 1930 σε Βιέννη και Χάρβαρντ, δίνεται ο εξής ορισμός του προβλήματος: «δίνεται ένα σύνολο πόλεων και οι μεταξύ τους αποστάσεις. Να βρεθεί το δρομολόγιο ενός πωλητή, το οποίο ελαχιστοποιεί τη διανυθείσα απόσταση, έτσι ώστε να επισκεφτεί όλες τις πόλεις ακριβώς μία φορά». Επίσης ανακαλύπτεται πως η στρατηγική επίλυσης βάσει επιλογής του κοντινότερου πελάτη -πλεονεκτικός αλγόριθμος- «greedy» δεν οδηγεί πάντα σε βέλτιστη λύση. Το 1959 οι Dantzig και Ramser πρότειναν το πρόβλημα VRP (Vehicle Routing Problem)[1] το οποίο αποτελεί και άμεσο πρόγονο του υπό μελέτη VRPTW. Συγκεκριμένα στο VRP, προστίθεται ένας ακόμα παράγοντας, η ελαχιστοποίηση του αριθμού των χρησιμοποιούμενων φορτηγών. Όλα τα φορτηγά έχουν κοινή αφετηρία. Όλα τα προβλήματα παρόμοιου τύπου πραγματεύονται πολυπαραμετρική βελτιστοποίηση ενός μη γραμμικού προβλήματος[2].

Το VRPTW είναι μία παραλλαγή του αρχικού VRP, η οποία όμως είναι πιο κοντά σε πραγματικά προβλήματα, προσθέτοντας μία χρονική παράμετρο. Η νέα αυτή παράμετρος αφορά το χρονικό διάστημα για το οποίο αποθήκη και πελάτες είναι διαθέσιμοι. Σημειώνεται πως παλαιότερα συνηθιζόταν η χρήση του όρου VRSPTW (Vehicle Routing and Scheduling Problem with Time Windows)[3].

Τα δεδομένα του προβλήματος θεωρούνται κάθε φορά γνωστά με ακρίβεια και είναι τα εξής: μέγιστο φορτίο ανά φορτηγό (capacity), διαθέσιμος αριθμός φορτηγών (vehicle number), συντεταγμένες θέσης της αποθήκης και των πελατών (Xcoord, Ycoord), ζήτηση των πελατών (demand), χρονικά παράθυρα (ready time, due date) και διάρκεια εκφόρτωσης (service time). Κάθε χρησιμοποιούμενο φορτηγό πραγματοποιεί ακριβώς ένα δρομολόγιο, το οποίο αποτελείται από πολλές επιμέρους επισκέψεις. Κάθε δρομολόγιο ξεκινάει και τελειώνει στην αποθήκη. Κάθε φορτηγό ξεκινάει το χρόνο t=0 από την αποθήκη γεμάτο με το φορτίο του. Προκειμένου να εξυπηρετήσει κάποιον πελάτη, πρέπει να φτάσει στο σημείο που βρίσκεται αυτός πριν τη λήξη του χρονικού παραθύρου και έχοντας τουλάχιστον το απαραίτητο φορτίο διαθέσιμο. Αν ένα φορτηγό δεν διαθέτει αρκετό φορτίο για άλλες επισκέψεις, θα πρέπει να επιστρέψει στην αποθήκη και να τερματίσει το δρομολόγιό του, χωρίς να έχει τη δυνατότητα ανεφοδιασμού. Ο χρόνος που απαιτείται για να φτάσει στον πελάτη το φορτηγό ισούται με την Ευκλείδεια απόσταση μεταξύ των δύο σημείων. Εάν το φορτηγό φτάσει πριν την ώρα έναρξης του χρονικού παραθύρου του πελάτη, μπορεί να περιμένει εκεί, χωρίς μεταφορικό κόστος, μέχρι να είναι διαθέσιμος ο πελάτης. Κατά την εκφόρτωση το χρονικό κόστος είναι ίσο με τον χρόνο εκφόρτωσης κοκ. Σε περίπτωση που δεν υπάρχουν άλλοι μη δρομολογημένοι πελάτες τα φορτηγά γυρνάνε στην αποθήκη πριν τη λήξη του χρονικού της παραθύρου.

Αξιολόγηση Αλγορίθμων

Κριτήρια Αξιολόγησης Αλγορίθμων

Προκειμένου να συγκριθούν οι αλγόριθμοι μεταξύ τους, ορίζονται συγκεκριμένα κριτήρια. Τα πιο σημαντικά είναι: η ποιότητα λύσης, ο υπολογιστικός χρόνος, η επαναληψιμότητα και η δυναμική. Άλλα κριτήρια αναφέρονται στο επίπεδο δυσκολίας του κώδικα και είναι προφανώς πιο υποκειμενικά.

Ποιότητα Λύσης

Για να ελεγχθεί η ποιότητα των λύσεων των αλγορίθμων, θα πρέπει αυτοί να εξεταστούν σε μεγάλο φάσμα δυσκολίας. Για τον λόγο αυτό χρησιμοποιούνται τα προβλήματα του Solomon (βλ. σελ. 8). Η ποιότητα λύσης προκύπτει από τη σύγκριση των παρακάτω με ιεραρχική σειρά: συνολικός αριθμός χρησιμοποιούμενων φορτηγών (Cumulative Number of Vehicles «CNV») και ολικά διανυθείσα απόσταση (total travelled distance, «TTD»). Αυτό σημαίνει πως αρχικά εξετάζεται ο CNV· σε περίπτωση ισοβαθμίας, εξετάζεται και η TTD. Φαίνεται πως ισχύει ότι όσο μειώνεται ο CNV, αυξάνεται η TTD[4]. Η λογική που κρύβεται πίσω από την ιεράρχηση των κριτηρίων είναι προφανώς η ελαχιστοποίηση πρώτον των ημερομισθίων και δεύτερον της κατανάλωσης βενζίνης.

Υπολογιστικός Χρόνος

Σκοπός δεν είναι πάντα να βρεθεί ο παγκόσμιος άριστος αλλά μία εφικτή λύση. Γενικά ισχύει πως υπάρχει κάποια αναλογία μεταξύ του υπολογιστικού χρόνου και της ποιότητας της λύσης· όσο περισσότερο «τρέχει» ένας αλγόριθμος τόσο καλύτερης ποιότητας είναι η τελική λύση[v]. Προκειμένου να παραχθεί λύση σε συγκεκριμένο χρόνο, πρέπει να γίνουν συμβιβασμοί σχετικά με την ποιότητα της λύσης. Η σύγκριση υπολογιστικού χρόνου διαφορετικών αλγορίθμων πρέπει να γίνεται σε ίδια υπολογιστική ισχύ. Όταν δεν υπάρχουν δεδομένα υπολογιστικού χρόνου ή εσωτερικών επαναλήψεων, θεωρείται πως κάθε αλγόριθμος έχει όσο χρόνο χρειάζεται για να βρει την καλύτερη λύση που μπορεί. Η φιλοσοφία της ορθής σύγκρισης γίνεται: «πόσο καλά είναι τα καλύτερα αποτελέσματα που μπορούν να παραχθούν με τη συγκεκριμένη μέθοδο». Μερικές εφαρμογές απαιτούν όσο το δυνατόν μικρότερο χρόνο όπως η δρομολόγηση τηλεφωνημάτων και ο ανασχεδιασμός δρομολογίων φορτηγών σε πραγματικό χρόνο. Σε τέτοιες εφαρμογές, το κριτήριο είναι «η δυνατόν καλύτερη λύση στον συγκεκριμένο χρόνο» και η λύση αυτή ονομάζεται «pareto optimal».

Επαναληψιμότητα

Με τον όρο «Επαναληψιμότητα» ορίζεται η ικανότητα ενός αλγορίθμου να παράγει λύσεις ίδιας ποιότητας, όποτε χρησιμοποιείται στις ίδιες συνθήκες. Χρήσει τυχαίων μεταβλητών παράγονται τυχαία αποτελέσματα (στοχαστικοί αλγόριθμοι). Στην πράξη αυτό σημαίνει πως οι αλγόριθμοι που κάνουν χρήση γεννήτριας τυχαίων αριθμών, κάθε φορά που λύνουν το ίδιο πρόβλημα θα παράγουν διαφορετικές, και διαφορετικής ποιότητας, λύσεις, δυσχεραίνοντας την ανάλυση και σύγκριση αποτελεσμάτων [vi].

---

[v] Ισχύει εφόσον υπάρχει βελτιωτική μέθοδος μέσα στον αλγόριθμο

[vi] Για να εξαλειφθεί αυτό το πρόβλημα, η αξιολόγηση γίνεται με βάση τη μέση τιμή πολλών επαναλήψεων. Τέτοιοι αλγόριθμοι δεν μπορούν να θεωρηθούν αξιόπιστοι για μία ή λίγες επαναλήψεις.

Δυναμική

Η αξιοπιστία της κάθε μεθόδου μπορεί να ελεγχθεί εξετάζοντας την ποιότητα λύσης που παράγει σε διαφορετικές συνθήκες. Αν για κάποια μόνο προβλήματα η ποιότητα λύσης είναι καλή, τότε ο υπό εξέταση αλγόριθμος δεν παρουσιάζει ικανοποιητική δυναμική, απλά έτυχε να δώσει κάποιες καλές λύσεις.

Τα Προβλήματα Του Solomon

Προκειμένου η αξιολόγηση των αλγορίθμων να γίνεται «επί ίσοις όροις» χρησιμοποιούνται κυρίως τα τυποποιημένα χαρακτηριστικά προβλήματα (benchmarks) που εισήγαγε ο Solomon το 1987[5]. Η αξιολόγηση πραγματοποιείται συγκρίνοντας την ποιότητα της λύσης και τον απαιτούμενο υπολογιστικό χρόνο κάθε αλγορίθμου στο σύνολο των προβλημάτων. Ο Solomon προσάρμοσε κάποια προβλήματα που είχαν δημιουργηθεί για το κλασσικό VRP[6] για το VRPTW[3].

Υπάρχουν 6 ομάδες προβλημάτων και ονομάζονται R1, R2, C1, C2, RC1 και RC2 και η κάθε μία παρουσιάζει 8 – 12 διαφορετικές παραλλαγές (προβλήματα). Οι πελάτες βρίσκονται σε μια περιοχή $[0x100]^2$, οι θέσεις τους είναι ίδιες για κάθε πρόβλημα της ίδιας ομάδας (Εικόνα 1), αλλάζουν όμως τα χρονικά τους παράθυρα. Η θέση της αποθήκης είναι περίπου στο κέντρο των πελατών. Στις ομάδες R1 και R2 τα γεωγραφικά σημεία – πελάτες, είναι τυχαία διασκορπισμένα (Random). Στις ομάδες C1 και C2 οι πελάτες είναι διατεταγμένοι ανά περιοχές (Clusters). Τέλος στις ομάδες RC1 και RC2 κάποιοι πελάτες είναι διασκορπισμένοι και κάποιοι ομαδοποιημένοι. Στις ομάδες R1, C1 και RC1 τα προβλήματα έχουν σχεδιαστεί για μικρές διαδρομές όπου επιτρέπεται στα φορτηγά να εξυπηρετούν 5 έως 10 πελάτες περίπου. Αντίθετα τα προβλήματα R2, C2, RC2 έχουν σχεδιαστεί έτσι ώστε να μπορούν να εξυπηρετήσουν περισσότερους και από 30 πελάτες. Τα C1 και C2 έχουν δημιουργηθεί έτσι ώστε να επιτρέπονται καλές, ίσως και βέλτιστες, λύσεις ανά γεωγραφικές περιοχές. Στον παρακάτω Πίνακα φαίνεται η μορφή του αρχείου R101 (ομάδα R1 πρώτο πρόβλημα).

Πίνακας 1: Παράδειγμα αρχείου προβλήματος Solomon (R101)

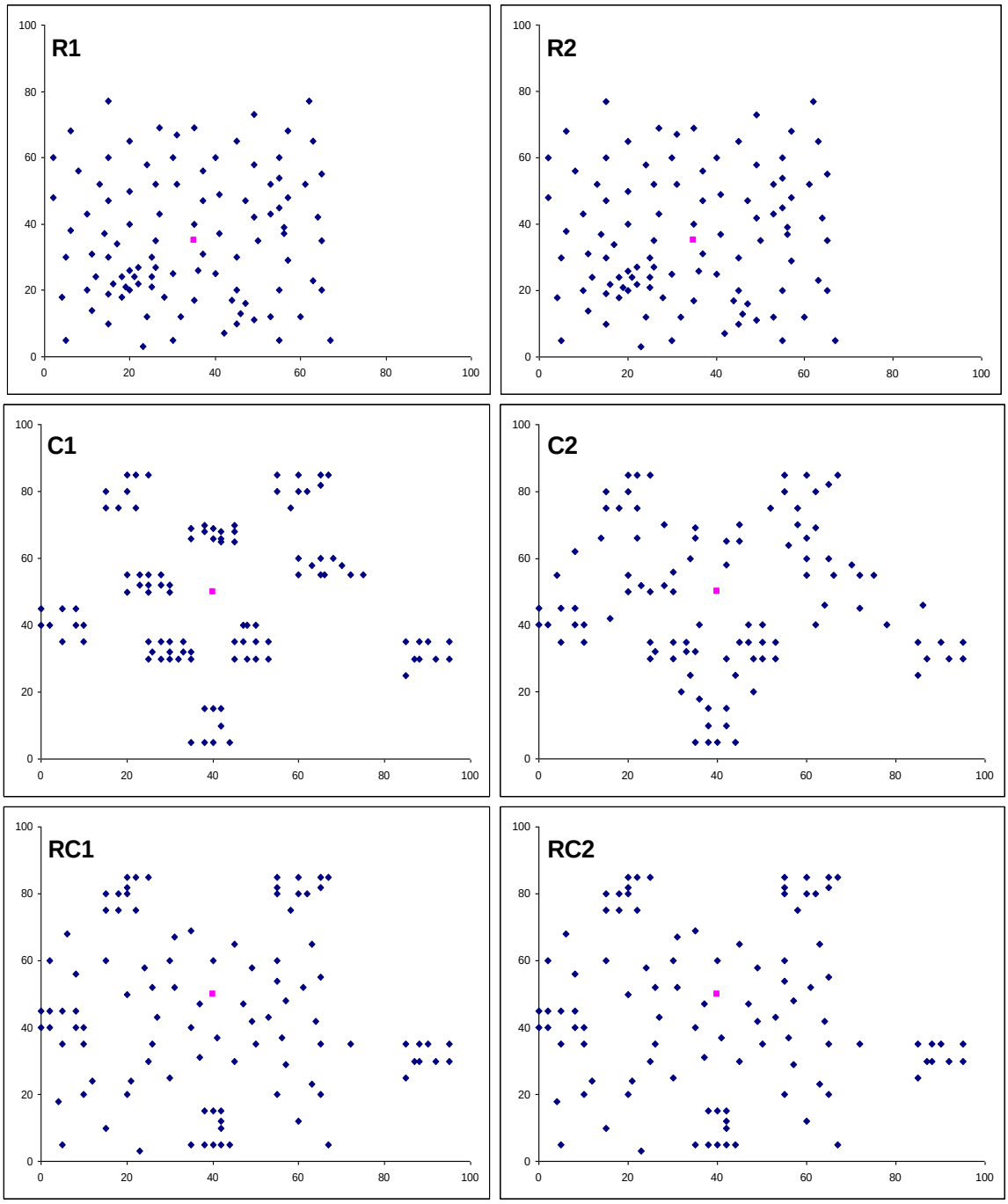
R101

VEHICLE
NUMBER CAPACITY
25 200

CUSTOMER
CUST NO. XCOORD. YCOORD. DEMAND READY TIME DUE DATE SERVICE TIME

0	35	35	0	0	230	0
1	41	49	10	161	171	10
2	35	17	7	50	60	10
..	..	..	..	..	..	..
99	20	26	9	83	93	10
100	18	18	17	185	195	10

Κατά την αξιολόγηση, η ποιότητα λύσης κάθε ομάδας ελέγχεται ξεχωριστά, ώστε να φαίνεται καλύτερα η συμπεριφορά του κάθε αλγορίθμου σε κάθε είδος προβλήματος. Με αυτόν τον τρόπο εξασφαλίζεται πως ο αλγόριθμος συμπεριφέρεται το ίδιο καλά σε κάθε περιβάλλον.



Εικόνα 1: Γεωγραφική κατανομή πελατών ανά ομάδα προβλήματος

Όπως φαίνεται στην εικόνα, τα προβλήματα R1 και R2 έχουν επίσης ίδια κατανομή των πελατών. Το ίδιο ισχύει και για τα RC1 και RC2. Αυτό δεν ισχύει όμως και για τα C1 και C2. Τα C2 είναι ελαφρώς πιο διασκορπισμένα.

Είδη αλγορίθμων

Ακριβείς αλγόριθμοι

Προκειμένου να παραχθεί η καλύτερη δυνατή λύση (ο παγκόσμιος άριστος), είναι απαραίτητο να χρησιμοποιηθεί αλγόριθμος, ο οποίος θα βρει όλες τις δυνατές λύσεις του προβλήματος, ώστε να εντοπιστεί αυτή που θα αριστοποιήσει την αντικειμενική συνάρτηση. Αν και οι αλγόριθμοι αυτοί είναι πολύ χρήσιμοι σε εύκολα προβλήματα, όταν τίθεται θέμα εφαρμογής σε προβλήματα αυξημένης πολυπλοκότητας, όπως το παρόν, είναι ασύμφοροι. Το υπολογιστικό κόστος, ο χρόνος δηλαδή που απαιτείται για να βρεθεί η βέλτιστη λύση, είναι τεράστιο, ενώ η απαιτούμενη μνήμη υπερβολική. Για τους παραπάνω λόγους οι ακριβείς αλγόριθμοι δεν χρησιμοποιούνται για τη λύση των VRPTW αυτούσιοι αλλά μόνο σε συνδυασμό με μεθόδους οι οποίες μειώνουν τον αριθμό των δυνατών τιμών κάθε μεταβλητής (βλ. σελ 18)

Ευρετικοί Αλγόριθμοι

Την αδυναμία των ακριβών αλγορίθμων έρχεται να καλύψει ένα άλλο είδος αλγορίθμων, οι ευρετικοί. Από τη φύση τους, καθώς είναι προσεγγιστικοί, δεν έχουν σαν στόχο να εντοπίσουν τη βέλτιστη λύση. Ακόμα κι αν αυτό γίνει, δεν μπορεί να αποδειχθεί. Περιορίζονται στη σάρωση «περιοχής λύσεων», ικανοποιητικής ποιότητας. Τέτοιες λύσεις απαιτούν πολύ μικρότερο υπολογιστικό χρόνο και μικρότερη μνήμη. Συνήθως αποτελούνται από δύο μέρη, το κατασκευαστικό και το τμήμα βελτίωσης[vii]. Στο κατασκευαστικό μέρος, δημιουργείται μια αρχική λύση S και στη συνέχεια στο τμήμα βελτίωσης, χρησιμοποιούνται διάφορες μέθοδοι-στρατηγικές με τις οποίες ανταλλάσοντας μέρη της αρχικής λύσης (π.χ. μεμονωμένοι πελάτες, ή/και κομμάτια από διαφορετικές αρχικές λύσεις), δημιουργώντας νέες βελτιωμένες λύσεις.

Αλγόριθμοι Τοπικής έρευνας

Οι τεχνικές τοπικής έρευνας αποτελούν μια ευρύτατη κλάση υπολογιστικών αλγορίθμων ευρετικής φύσης που επιλύουν προβλήματα διακριτής αριστοποίησης. Ο αριθμός των εφικτών λύσεων αν και πολύ μεγάλος είναι πεπερασμένος. Η τεχνική που χρησιμοποιείται είναι η έρευνα σε βάθος στο σύνολο των εφικτών λύσεων S ώστε να βρεθούν τοπικά ακρότατα της αντικειμενικής συνάρτησης. Στο τέλος κάθε επανάληψης δημιουργείται μια καινούρια λύση, η οποία προκύπτει από την προηγούμενη, και ανήκει στο γενικό σύνολο λύσεων S. Με άλλα λόγια οι νέες αυτές λύσεις ανήκουν στην *γειτονιά* των προηγούμενων λύσεων. Αυτό σημαίνει πως οι παραγόμενες λύσεις είναι συγγενικές με τις αρχικές και άρα μεταφέρουν την μυωπική συμπεριφορά τους. Για να τερματιστεί η επαναληπτική διαδικασία ορίζονται κριτήρια αποδοχής τελικής λύσης. Ένα σύνηθες φαινόμενο των αλγορίθμων τοπικής έρευνας είναι ο σύντομος τερματισμός τους λόγω της παγίδευσης τους σε περιοχές τοπικών ακρότατων.

[vii] Σε μερικές περιπτώσεις το τμήμα βελτίωσης είναι ενσωματωμένο στο κατασκευαστικό, χωρίς να γίνεται διαχωρισμός.

Μεταευρετικοί Αλγόριθμοι

Οι ευρετικοί αν και μειώνουν σημαντικά τον υπολογιστικό χρόνο επίλυσης, παρουσιάζουν δύο βασικά μειονεκτήματα: πρώτον: δεν εξάγουν ικανοποιητικά αποτελέσματα, εγκλωβίζονται σε τοπικά ελάχιστα και τερματίζονται επομένως οι λύσεις τους είναι σπανίως υψηλής ποιότητας. Δεύτερον: δεν έχουν γενική εφαρμογή σε προβλήματα διακριτής βελτιστοποίησης, επειδή κατασκευάζονται με βάση τους περιορισμούς κάθε προβλήματος.

Σε αυτά τα αδιέξοδα η λύση έρχεται από έναν νέο τύπο αλγορίθμων, ο οποίος επιχειρεί να συνδυάσει βασικές αρχές ευρετικών μεθόδων (κατασκευαστικών και βελτιωτικών) με ένα ψηλό επίπεδο πλαισίου λειτουργίας, με σκοπό την εξερεύνηση του χώρου των λύσεων. Ουσιαστικά πρόκειται για στρατηγικές που επιλέγονται με τέτοιο τρόπο, ώστε να επιτυγχάνεται μια δυναμική ισορροπία ανάμεσα στην εκμετάλλευση της συσσωρευμένης εμπειρίας της έρευνας και της εξερεύνησης του χώρου λύσεων. Ο χώρος λύσεων που επιλέγεται έχει διπλή σημασία: περιέχει το δυνατόν υψηλότερης ποιότητας λύσεις ενώ ταυτόχρονα αποκλείει περιοχές οι οποίες δεν έχουν ενδιαφέρον είτε ως αδύνατες είτε ως «κακές» λύσεις. Η έρευνα που διεξάγεται από ένα μεταευρετικό πρέπει να είναι αρκετά ευφυής ώστε να εντατικοποιείται αφενός στις περιοχές με υψηλής ποιότητας λύσεις (*μηχανισμός εντατικοποίησης*) και αφετέρου να μετακινείται σε ανεξερεύνητες περιοχές του χώρου των λύσεων όταν κρίνεται απαραίτητο (*μηχανισμός διαφοροποίησης*).

Κατασκευαστικοί Ευρετικοί

Ορισμός

Οι κατασκευαστικοί αλγόριθμοι δημιουργούν λύσεις από το μηδέν. Επιλέγουν σημεία - πελάτες, μέχρις ότου δημιουργηθεί μία εφικτή λύση. Η διαδρομή που κατασκευάζεται κάθε φορά προκύπτει από την ελαχιστοποίηση κριτηρίων. Οι διαδοχικές μέθοδοι κατασκευάζουν μία διαδρομή ανά φορά, από σημείο σε σημείο για κάθε φορτηγό, ενώ οι παράλληλες μέθοδοι πολλές διαδρομές ταυτόχρονα. Η παραγωγή λύσης υψηλής ποιότητας κατά τη διάρκεια εφαρμογής μιας κατασκευαστικής μεθόδου, είναι μεγάλης σημασίας, καθώς ακόμα κι αν χρησιμοποιείται βελτιωτική μέθοδος, η μυωπική συμπεριφορά της λύσης θα υπάρχει και στη γειτονιά της αρχικής λύσης.

Κατασκευαστικές μέθοδοι

Πλεονεκτικός

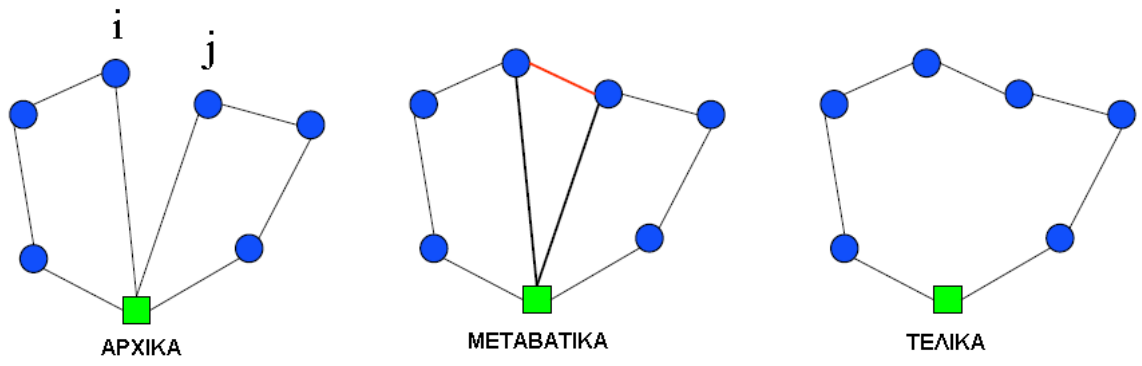
Η αρχή της μεθόδου είναι πολύ απλή. Κάθε πελάτης προστίθεται στο τέλος της διαδρομής, πριν δηλαδή επιστρέψει το φορτηγό στην αφετηρία. Το κριτήριο δρομολόγησης κάθε πελάτη είναι η ελαχιστοποίηση της απόστασης (ή/και του χρονικού ορίζοντα) μεταξύ του τελευταία δρομολογημένου και του υπό εξέταση πελάτη. Τα κριτήρια αποδοχής κάθε υπό εξέταση πελάτη είναι να μην παραβιάζονται τα χρονικά παράθυρα του πελάτη και της αφετηρίας και να υπάρχει αρκετό διαθέσιμο φορτίο στο φορτηγό. Κάθε διαδρομή τερματίζεται όταν δεν μπορεί να εισαχθεί άλλος πελάτης. Τα αποτελέσματα αυτού του αλγορίθμου είναι φτωχής ποιότητας (βλ. Πίνακα 2, NN nearest neighbor). Λόγω της απλότητάς του, οι λύσεις παράγονται ταχύτατα.

Route-first cluster-second

Η συνεισφορά του Solomon στα προβλήματα VRPTW είναι οπωσδήποτε μεγάλη. Εκτός από τα 56 προβλήματα που δημιούργησε και χρησιμοποιούνται για την αξιολόγηση και σύγκριση αλγορίθμων, δημιουργεί και τις πρώτες σοβαρές μεθόδους προσέγγισης της λύσης. Η συγκεκριμένη «route-first cluster-second» αποτελείται από δύο στάδια: αρχικά δημιουργείται μία μεγάλη διαδρομή, η οποία ενώνει όλους τους πελάτες. Στη συνέχεια διαμερίζεται σε άλλες μικρότερες. Για την κατασκευή της μεγάλης – αρχικής διαδρομής, χρησιμοποιούνται αλγόριθμοι περιοδεύοντος πωλητή (TSP). Επειδή τα αποτελέσματα από έναν τέτοιο αλγόριθμο διαφέρουν ανάλογα με τη μέθοδο κατασκευής της αρχικής διαδρομής, είναι αδύνατον να συγκριθεί με τους άλλους αλγορίθμους.

Savings

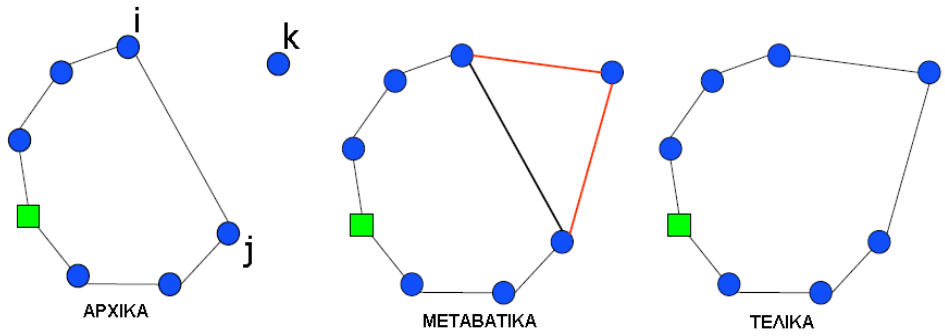
Άλλη μία μέθοδος προτείνεται από τον Solomon ως επέκταση του γνωστού "Savings" Εικόνα 2. Αρχικά αυτός ο αλγόριθμος σχεδιάστηκε για την επίλυση προβλημάτων VRP και μάλλον είναι ο γνωστότερος κατασκευαστικός αλγόριθμος. Η διαδικασία μοιάζει αντίστροφη της route-first cluster-second. Ξεκινάει με μία λύση κατά την οποία κάθε πελάτης εξυπηρετείται από ακριβώς ένα φορτηγό. Συνδυάζοντας στη συνέχεια δύο διαδρομές μικραίνει το κόστος εξυπηρέτησης $S_{ij}=d_{i0}+d_{0j}-d_{ij}$ (όπου με S_{ij} συμβολίζεται το κέρδος από τον συγκερασμό δύο διαδρομών d_{i0} και d_{0j}) ενώ με d_{ij} συμβολίζεται η μεταξύ των δύο πελατών απόσταση. Καθώς η διαδικασία επαναλαμβάνεται, ακυρώνονται κάποια δρομολόγια, ενώ πυκνώνουν κάποια άλλα, μέχρι να μην είναι δυνατή η συγχώνευση δύο διαδρομών. Ενώ αρχικά το μόνο κριτήριο που υπήρχε στα VRP ήταν η ελαχιστοποίηση του S_{ij} με την ένωση δύο σημείων[7], ο Solomon έρχεται να προσθέσει τον έλεγχο της εφικτότητας της λύσης. Αυτό συνεχίζεται μέχρι να εξαντληθεί το φορτίο του κάθε φορτηγού προκειμένου να βρεθεί ο κοντινότερος χρονικά ή/και γεωμετρικά πελάτης. Ο Solomon, θέλοντας να εξασφαλίσει μία ισορροπία ανάμεσα στο χρονικό και γεωγραφικό κριτήριο, εισάγει ένα άνω όριο χρόνου αναμονής σε κάθε διαδρομή. Η συγκεκριμένη μέθοδος μπορεί εύκολα να αναπτυχθεί είτε διαδοχικά είτε παράλληλα σε όλες τις διαδρομές.



Εικόνα 2: Αλγόριθμος Savings

Insertion

Οι σημαντικότεροι αλγόριθμοι του Solomon είναι οι insertion και ονομάζονται I1, I2 και I3. Ο πλέον αποτελεσματικός και πιο δημοφιλής είναι ο I1. Πολλές φορές χρησιμεύει ως βάση σύγκρισης με άλλους αλγορίθμους. Η βασική αρχή τους είναι κοινή. Κάθε διαδρομή αρχίζει με έναν πελάτη - κέντρο και οι υπόλοιποι μη δρομολογημένοι πελάτες εισάγονται σε αυτήν όσο αυτό είναι εφικτό (**Εικόνα 3**). Σε περίπτωση που υπάρχουν μη δρομολογημένοι πελάτες όταν τελειώσει μία διαδρομή, αρχίζει μια νέα.
Ο αρχικός πελάτης επιλέγεται είτε ως ο πιο απομακρυσμένος γεωγραφικά από την αποθήκη, είτε ως ο πελάτης του οποίου το χρονικό του παράθυρο ανοίγει νωρίτερα. Η εισαγωγή του νέου αδρομολόγητου πελάτη γίνεται ανάμεσα σε δύο δρομολογημένους. Σε αυτές τις μεθόδους υπάρχουν μεταβλητές που καθορίζουν πόσο σημαντική είναι για τη λύση η απόσταση του νέου πελάτη από την αποθήκη σε αντίθεση με την προστιθέμενη διανυθείσα απόσταση και τον προστιθέμενο απαιτούμενο χρόνο. Ο I2 φαίνεται να στοχεύει στην ελαχιστοποίηση της ολικά διανυθείσας απόστασης και του ολικού χρόνου, ενώ ο I3 στοχεύει στην εξυπηρέτηση των πελατών που επείγονται.



Εικόνα 3: Αλγόριθμος Insertion

Σε μεταγενέστερες μελέτες[8,9] διορθώνεται ένα κριτήριο του Solomon το οποίο υποτιμούσε το χρόνο που προστίθεται κατά την εισαγωγή ενός νέου πελάτη μεταξύ της αποθήκης και του πρώτου πελάτη. Με τα νέα κριτήρια σε σχετικά μικρά δρομολόγια μπορεί να επιτευχθεί βελτίωση έως και 50%.
Νωρίτερα έχει πραγματοποιηθεί από του Potvin και Rousseau μια διαφορετική χρήση του I1, η παράλληλη[10]. Αρχικά χρησιμοποιείται η αρχική μορφή του I1, η διαδοχική, προκειμένου να οριστεί ο αριθμός των χρησιμοποιούμενων φορτηγών ανά πρόβλημα και για να γίνει η επιλογή του αρχικού πελάτη. Στη συνέχεια χρησιμοποιούνται αυτά τα δεδομένα για να γίνει η ανάθεση των υπολοίπων πελατών. Το κριτήριο επιλογής του επόμενου πελάτη σχετίζεται με το κατά πόσο συμφέρει να δρομολογηθεί ένας πελάτης σε μια διαδρομή από ότι στις άλλες. Περεταίρω μελέτες[11] έδειξαν πως η χρήση πολλών επεξεργαστών παράλληλα, μειώνει γραμμικά τον υπολογιστικό χρόνο στα παράλληλα μέρη του αλγορίθμου.
Πρόσφατα ο Ioannou[12] χρησιμοποιώντας τη γενική μορφή του I1 όπως προτάθηκε από τον Solomon ρύθμισε τα κριτήρια έτσι ώστε να ελαχιστοποιείται η επίδραση της εισχώρησης νέων πελατών στους πελάτες που βρίσκονται στην υπό κατασκευή διαδρομή αλλά και στους αδρομολόγητους και στον υπό εξέταση πελάτη.

Sweep

Πρόκειται για συνδυασμό δύο αλγορίθμων, του αλγορίθμου σάρωσης (sweep heuristic)[13] και του I1. Η βάση της μεθόδου είναι αντίστοιχη με τη «Route-first cluster-second» όμως με ανάποδη σειρά. Αρχικά ομαδοποιούνται οι πελάτες κάνοντας χρήση του sweep heuristic, ανατίθεται δηλαδή σε κάθε φορτηγό ένας αριθμός πελατών. Η ομαδοποίηση γίνεται χωρίζοντας τους πελάτες σε περιοχές ανάλογα με τη γωνία που σχηματίζουν από το κέντρο βάρους του προβλήματος. Στη συνέχεια χρησιμοποιείται ο I1 για τη δρομολόγησή τους.

Τα αποτελέσματα των ευρετικών που βρέθηκαν στη βιβλιογραφία φαίνονται παρακάτω (Πίνακας 2). Ο πρώτος αριθμός δηλώνει το μέσο όρο των φορτηγών που χρησιμοποιήθηκαν ανά κλάση ενώ ο δεύτερος τον μέσο όρο της διανυθείσας απόστασης. Με CNV συμβολίζεται ο συνολικός αριθμός χρησιμοποιούμενων φορτηγών για όλα τα προβλήματα και αντίστοιχα με CTD η συνολική διανυθείσα απόσταση.

Πίνακας 2: Ευρετικοί κατασκευαστικοί αλγόριθμοι

Συγγραφέας / μέθοδος	R1	R2	C1	C2	RC1	RC2	CNV/ CTD	T[viii] (min)
Solomon	13,6	3,3	10,0	3,1	13,5	3,9	453	3,85
I1	1.436,7	1.402,4	951,9	692,7	1.596,5	1.682,1	73.004	
Solomon	14,5	3,3	10,1	3,4	14,2	4,1	474,8	4,08
I2	1638,7	1470,7	1049,8	921,5	1874,4	1797,6	82.038	
Solomon	14,1	3,4	10,0	3,5	14,0	4,0	468,6	4,85
I3	1651,7	1474,6	1103,3	1072,7	1849,7	1816,4	83.881	
Potvin et al.	13,33	3,09	10,67	3,38	13,38	3,63	453	19,6
I1 Παράλληλη	1.509,0	1.386,7	1.343,7	797,6	1.723,7	1.651,1	78.834	
Solomon	14,5	3,4	10,2	3,5	14,2	3,9	476,0	0,82
NN	1600,1	1472,3	1171,2	963,1	1800,0	1754,7	82.080	
Solomon	14,6	3,2	10,0	3,0	14,9	4,0	475,6	2,35
S	1499,7	1448,6	940,8	711,9	1804,5	1735,7	76415	
Ioannou	12,67	3,1	10,00	3,1	12,5	3,5	429	4
I1	1.370	1.310	865	662	1.512	1.483	67.891	

Δυστυχώς δεν ήταν δυνατόν να συμπεριληφθούν όλοι οι αλγόριθμοι που αναφέρθηκαν παραπάνω καθώς είτε δεν χρησιμοποίησαν τα προβλήματα του Solomon ή δεν δίνουν αρκετές πληροφορίες. Σχολιάζοντας τα αποτελέσματα φαίνεται πως ο Ioannou είχε καλύτερα αποτελέσματα με κάποιο σχετικό κόστος στον υπολογιστικό χρόνο. Οι υπολογιστικοί χρόνοι φαίνονται να είναι πολύ διαφορετικοί, αυτό όμως γίνεται γιατί διαφέρουν στην υπολογιστική δύναμη του κάθε επεξεργαστή. Ουσιαστικά θεωρούνται πάρα πολύ μικροί, συγκριτικά με τους αλγόριθμους τοπικής έρευνας είναι σαφώς γρηγορότεροι όμως υστερούν στην ποιότητα της λύσης συγκρινόμενοι με άλλους πιο εξεζητημένους αλγόριθμους. Η ταχύτητα επίλυσης των 56 προβλημάτων διαφέρει από μέθοδο σε μέθοδο και από υπολογιστή σε υπολογιστή.

[viii] Οι υπολογιστικοί χρόνοι αναφέρονται στο σύνολο των προβλημάτων και βρέθηκαν χρησιμοποιώντας τους εξής επεξεργαστές: Solomon: DEC 10, Potvin et al.: IBM PC, Ioannou et al.: Intel Pentium 133MHz

Βελτιωτικοί αλγόριθμοι

Ορισμός

Οι κλασσικοί αλγόριθμοι τοπικής έρευνας από τη φύση τους είναι βελτιωτικές μέθοδοι και σχηματίζουν μια γενική τάξη προσεγγιστικών ευρετικών αλγορίθμων. Στηρίζονται στη φιλοσοφία της συνεχούς βελτίωσης μιας λύσης, με επαναληπτικές μεθόδους εξερεύνησης των γειτόνων. Προκειμένου να σχεδιαστεί ένας αλγόριθμος τοπικής έρευνας, πρέπει να καθοριστεί από την αρχή: πώς θα δημιουργηθεί η αρχική λύση (κατασκευαστικό μέρος), ποια θα είναι η κίνηση (χειριστής/ές) με την οποία θα βελτιώνονται οι λύσεις, το κριτήριο αποδοχής της νέας λύσης(π.χ. ελαχιστοποίηση της διανυθείσας απόστασης), και πότε θα τερματίζεται η επαναληπτική διαδικασία (π.χ. έως ότου να μη βελτιώνεται περισσότερο η λύση). Ο μηχανισμός δημιουργίας της νέας λύσης, εξετάζει την επίδραση που έχει κάθε αλλαγή σε μία διαδρομή όταν οι αρχικοί πελάτες αντικαθίστανται με άλλους. Αν η αλλαγή είναι συμφέρουσα τότε αντικαθιστά την αρχική λύση.

Δύο στρατηγικές αποδοχής είναι γνωστές στα περιθώρια του VRPTW, η αποδοχή της πρώτης ικανής λύσης (FA first-accept) και η αποδοχή της καλύτερης δυνατής λύσης (BA best-accept). Κατά τη στρατηγική αποδοχής της πρώτης δυνατής λύσης, επιλέγεται ένας γείτονας που να τηρεί το προκαθορισμένο κριτήριο αποδοχής. Αντίθετα κατά τη στρατηγική αποδοχής της βέλτιστης λύσης, εξετάζονται όλοι οι γείτονες που τηρούν το προκαθορισμένο κριτήριο και επιλέγεται ο καλύτερος. Η τοπικά βέλτιστη λύση που παράγεται με αυτούς τους αλγόριθμους μπορεί να απέχει πολύ από την παγκόσμια βέλτιστη, αυτό οφείλεται στην αδυναμία των αλγορίθμων αυτών να προβλέψουν τα αποτελέσματα των αποφάσεών τους, γι' αυτό και χαρακτηρίζονται ως μυωπικοί. Πρακτικά αυτό σημαίνει πως η ποιότητα της λύσης εξαρτάται κατά πολύ από την αρχική λύση, τη λύση που παράγεται στο κατασκευαστικό κομμάτι, και από τον μηχανισμό που χρησιμοποιείται προκειμένου να παραχθούν οι νέες γειτονιές. Επίσης η ποιότητα λύσης είναι ανώτερη όταν συνδυάζονται σε ένα βήμα μια κατασκευαστική και μια βελτιωτική μέθοδος παρά σε δύο διαφορετικά.[14]

Οι περισσότεροι αλγόριθμοι που έχουν προταθεί σε προβλήματα δρομολόγησης φορτηγών είναι τύπου *ανταλλαγής άκρων*. Συνήθως χωρίζονται σε εσωτερικές ανταλλαγές (π.χ μετάθεση ενός πελάτη σε άλλη θέση) και σε εξωτερικές (ανταλλαγή πελατών μεταξύ δύο ή περισσοτέρων διαφορετικών διαδρομών). Οι γειτονιές ανταλλαγής άκρων για μία διαδρομή είναι το σύνολο των διαδρομών που παράγονται από μία αρχική διαδρομή, αντικαθιστώντας ένα σύνολο k άκρων του με k άλλα άκρα. Τέτοιες αντικαταστάσεις ονομάζονται k-ανταλλαγές (k-exchanges), και μία διαδρομή που δεν μπορεί να βελτιωθεί από μία k-ανταλλαγή ονομάζεται k-βέλτιστη (k-optimal).

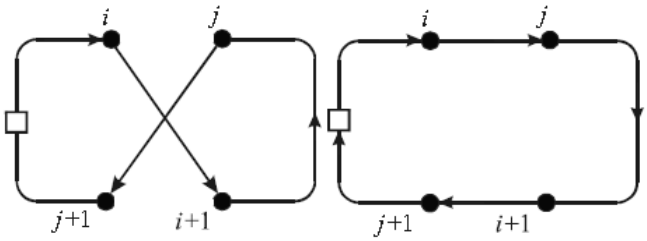
Ο χρόνος που απαιτείται για την παραγωγή του k-βέλτιστου είναι $O(n^k)$. Στην Εικόνα 4 φαίνεται ένας 2-opt χειριστής (2-exchange), ο οποίος προσπαθεί να βελτιώσει τη διαδρομή αντικαθιστώντας 2 από τα άκρα του με 2 άλλα άκρα. Η διαδικασία επαναλαμβάνεται μέχρι να μην βελτιώνεται περισσότερο η λύση.

Βελτιωτικές Μέθοδοι

Οι βελτιωτικές μέθοδοι χρησιμοποιούν αρχικές λύσεις για να παράξουν άλλες καλύτερες. Οι αρχικές λύσεις μπορεί να παράγονται με χρήση πλεονεκτικού, insertion, savings ή όποιουδήποτε άλλου κατασκευαστικού αλγορίθμου. Οι βελτιωτικές μέθοδοι έχουν σαν σκοπό να εντοπίσουν τα τοπικά ελάχιστα μιας περιοχής λύσεων. Πολλές φορές είναι δυνατόν να χρησιμοποιηθούν πολλές μέθοδοι μαζί, όπου η μία θα διαδέχεται την άλλη. Με αυτόν τον τρόπο εκμεταλλευόμαστε τα θετικά κάθε μεθόδου, παράγοντας υψηλότερης ποιότητας λύσεις.

2-opt και 3-opt

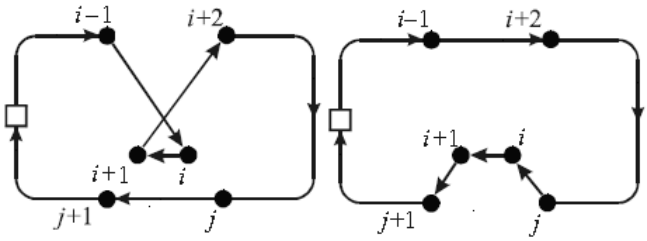
Οι χειριστές 2-opt και 3-opt προτάθηκαν από τον Lin[15]. Ο τρόπος που λειτουργούν είναι ο εξής: αφαιρούνται 2 ή 3 συνδέσεις αντίστοιχα και γίνεται προσπάθεια να βρεθεί ποια από τις άλλες δυνατές συνδέσεις είναι η καλύτερη. Στο σχήμα φαίνεται πως είναι δυνατόν να προκληθεί αλλαγή κατεύθυνσης κατά τη χρήση αυτών των χειριστών, όμως η πραγματικότητα είναι πως μόνο το 10% αντιστρέφονται[16]. Αυτές οι μέθοδοι βελτιώνουν εσωτερικά την τρέχουσα λύση.



Εικόνα 4: χειριστής 2-opt

Or-opt

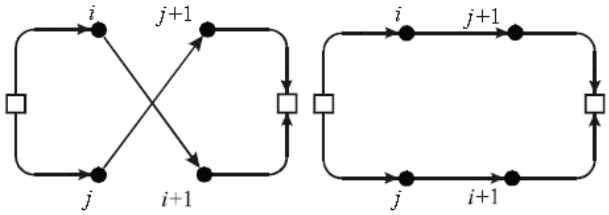
Ένας ακόμα εσωτερικός χειριστής είναι ο Or-opt Εικόνα 5, που παρουσιάστηκε το 1976 για το πρόβλημα του πλανόδιου πωλητή[17]. Η βασική ιδέα είναι να ανατοποθετηθούν δύο διαδοχικοί πελάτες της αρχικής διαδρομής σε άλλη θέση, χωρίς να αλλάζει η κατεύθυνση της διαδρομής όπως φαίνεται στο επόμενο σχήμα.



Εικόνα 5: Χειριστής Or-opt

2-opt*

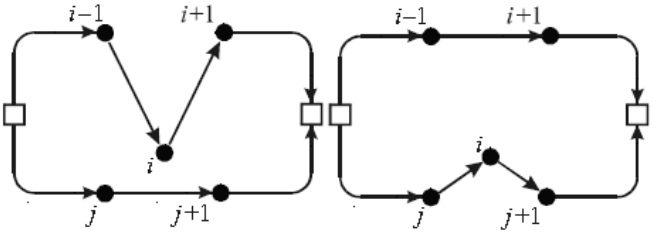
Οι Potvin και Rousseau στη συγκεκριμένη μελέτη[18] συγκρίνουν ευρετικούς αλγόριθμους ανταλλαγής άκρων για τα VRPTW προβλήματα (2-opt, 3-opt και Or-opt) ενώ παρουσιάζεται ένας καινούριος χειριστής, ο 2-opt*. Η βασική ιδέα είναι ο συνδυασμός δύο διαδρομών έτσι ώστε οι τελευταίοι πελάτες της μιας διαδρομής να τοποθετούνται μετά από τους αρχικούς πελάτες μιας άλλης διατηρώντας την κατεύθυνση της διαδρομής. Στην Εικόνα 6 φαίνεται σχηματικά η λειτουργία αυτού του χειριστή όπου τα άκρα (i,i+1), (j,j+1) αντικαθίστανται με τα (i,j+1), (j,i+1). Σε περίπτωση που ο i είναι ο πρώτος πελάτης της μίας διαδρομής και ο j+1 ο τελευταίος της άλλης, οι διαδρομές μπορούν να συγχωνευτούν σε μία.
Όταν οι Or-opt και 2-opt* συνδυάζονται, τα αποτελέσματα είναι ιδιαιτέρως καλής ποιότητας. Αυτό γίνεται ως εξής: στην αρχή χρησιμοποιείται ο ένας μέχρι να βρεθεί ένα τοπικό ελάχιστο. Όταν αυτό συμβεί, χρησιμοποιείται ο άλλος. Όταν αυτή η επαναλαμβανόμενη διαδικασία δεν βρίσκει άλλο τοπικό ελάχιστο σταματάει. Για το κατασκευαστικό κομμάτι χρησιμοποιείται ο I1 του Solomon.



Εικόνα 6: 2-opt*

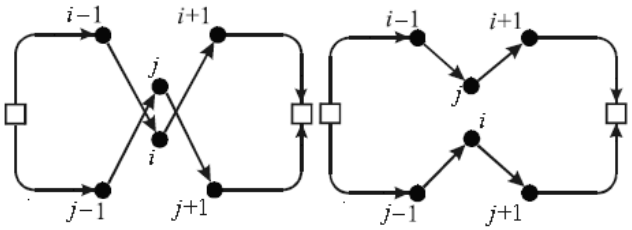
Relocate, exchange και cross

Οι παραπάνω χειριστές προτάθηκαν το 1992[19]. Η χρήση τους είναι τόσο διαδεδομένη ώστε χρησιμοποιούνται ακόμα και σήμερα σε ευρετικούς και μεταευρετικούς αλγορίθμους. Ο χειριστής ανατοποθέτησης «*relocate*» απλά μεταφέρει έναν δρομολογημένο πελάτη από μία διαδρομή σε μια άλλη χωρίς να αλλάζει ο προσανατολισμός των διαδρομών (**Εικόνα 7**). Τα οφέλη από αυτόν τον χειριστή είναι εμφανή, καθώς είναι δυνατόν όταν δρομολογήθηκε ο εν λόγω πελάτης, να μην μπήκε στην καλύτερη δυνατή για αυτόν θέση.



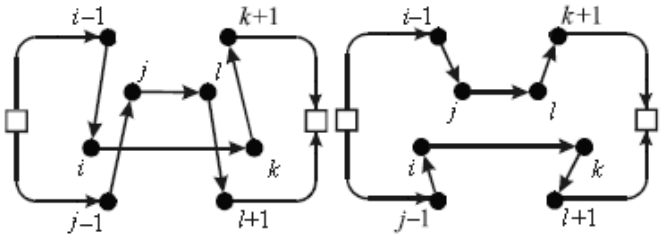
Εικόνα 7: Relocate operator χειριστής ανατοποθέτησης

Ο ανταλλακτικός χειριστής ανταλλάζει 2 πελάτες μεταξύ δύο διαδρομών όπως φαίνεται στην Εικόνα 8. Με αυτόν τον τρόπο επίσης δεν αλλάζει ο προσανατολισμός της λύσης.



Εικόνα 8: exchange operator χειριστής ανταλλαγής

Τέλος ο χειριστής διασταύρωσης «cross» (βλ. Εικόνα 9) ανταλλάσει ζευγάρια διαδοχικών πελατών. Ουσιαστικά οι δύο τελευταίοι χειριστές μοιάζουν πολύ μεταξύ τους. Η διαφορά τους είναι πως ανταλλάσσονται δύο πελάτες αντί για έναν. Κοινό χαρακτηριστικό είναι πως ούτε εδώ αλλάζει η κατεύθυνση της διαδρομής.



Εικόνα 9: cross exchange operator χειριστής ανταλλαγής διασταύρωσης

CP (περιοριστικός προγραμματισμός)

Σε προβλήματα τα οποία κάθε μεταβλητή μπορεί να πάρει πεπερασμένο αριθμό τιμών, η μείωση των διαθέσιμων τιμών μειώνει τις υπολογιστικές ανάγκες. Ο περιοριστικός προγραμματισμός «Constraint Programming» CP[20] αποτελεί αντιπροσωπευτικό παράδειγμα για τον τρόπο παρουσίασης και λύσης μιας μεγάλης ποικιλίας συνδυαστικών προβλημάτων. Αρχικά, κάθε μεταβλητή συσχετίζεται με έναν τομέα ο οποίος περιέχει τις δυνατές της τιμές. Προκειμένου να περιοριστεί ο αριθμός των τιμών κάθε τομέα, αξιοποιούνται οι φραγμοί, οι οποίοι συγκεντρώνουν πληροφορίες για συνδυασμούς που δημιουργούν κακής ποιότητας λύσης (ή αδύνατες λύσεις). Για κάθε τομέα αξιοποιούνται όλοι οι σχετικοί φραγμοί. Κατά τη μείωση ενός τομέα, προκύπτουν και μειώσεις άλλων. Αυτή η επαναλαμβανόμενη διαδικασία σταματάει όταν δεν είναι πλέον δυνατή η μείωση κάποιου τομέα, ή όταν κάποιος τομέας μηδενίζεται. Κατά τον περιοριστικό προγραμματισμό, ο υπολογισμός οδηγείται από τους φραγμούς γεγονός που τους καθιστά σημαντικούς. Ύστερα από την παραπάνω διαδικασία, τα προβλήματα λύνονται με τη χρήση ολοκληρωτικών τεχνικών όπως η «depth-first search και η branch and bound[30]».

λ-ανταλλαγή

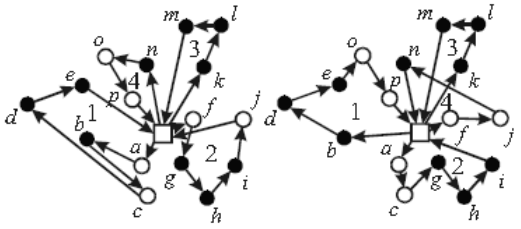
Το 1993[21] αναφέρεται ένας τρόπος ανταλλαγής δρομολογημένων πελατών ανά δύο διαδρομές της ίδιας λύσης που ονομάζεται λ-interchange. Το μέρος της λύσης που ανταλλάσσεται πρέπει να είναι μικρότερο ή ίσο με το μέγεθος λ για κάθε μία από τις συμμετέχουσες διαδρομές. Με αυτόν τον τρόπο παράγονται δύο νέες διαδρομές των οποίων οι πελάτες προκύπτουν από τις δύο αρχικές. Όταν αυτό πραγματοποιείται σε όλες τις διαδρομές της λύσης S προκύπτει μια γειτονιά λύσεων S’.

Εξαγωγικές αλυσίδες

Σκοπός της μεθόδου[22] είναι να δημιουργηθεί αρκετός χώρος για έναν αδρομολόγητο πελάτη σε μία διαδρομή, χωρίς να απαιτείται να αφαιρεθεί κάποιος άλλος. Η βασική ιδέα θυμίζει *αντιδράσεις με ενεργή ρίζα* και αποτελείται από τρία στάδια: έναρξη, διάδοση και τερματισμό. Η έναρξη της διαδικασίας γίνεται με την αφαίρεση ενός δρομολογημένου (i) πελάτη και την αντικατάστασή του με έναν άλλο αδρομολόγητο (u). Η διάδοση γίνεται με τη χρήση του νέου αδρομολόγητου πελάτη (πρώην δρομολογημένου, $u=i_{old}$) ο οποίος στην προσπάθειά του να δρομολογηθεί αναγκάζει έναν άλλο δρομολογημένο (i_{new}) να του δώσει τη θέση του. Ο τερματισμός της διαδικασίας γίνεται όταν επιτευχθεί ο σκοπός, δηλαδή δεν χρειάζεται να αφαιρεθεί δρομολογημένος πελάτης προκειμένου να δρομολογηθεί ο αδρομολόγητος. Τον ρόλο της ενεργής ρίζας παίζει ο αδρομολόγητος πελάτης. Η αφαίρεση και εισαγωγή τερματίζονται όταν δεν χρειάζεται να αφαιρεθεί πελάτης από τη διαδρομή δέκτη. Όσο προχωράει η έρευνα «αδειάζει» η διαδρομή δότης. Αν ένα δρομολόγιο αδειάσει εντελώς, πραγματοποιείται μείωση των διαδρομών.

Κυκλική μεταφορά

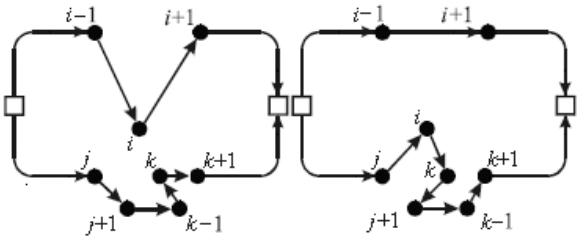
Οι Thomson και Psaraftis[23] προτείνουν μια μέθοδο που μοιάζει πολύ με τη «λ-ανταλλαγή». Η βασική διαφορά έγκειται στον αριθμό των διαδρομών που εμπλέκονται. Ουσιαστικά ο χειριστής ανταλλάσσει πελάτες «κυκλικά», δηλαδή κάθε διαδρομή δίνει πελάτες σε μια άλλη. Κάθε πελάτης έχει διαφορετική ζήτηση. Για αυτόν τον λόγο αποφασίζεται το όριο συνολικής ζήτησης που θα μεταφερθεί και ονομάζεται k. Η ακριβής αριστοποίηση είναι δύσκολη, γι' αυτό χρησιμοποιούνται ευρετικές προσεγγίσεις. Στην εικόνα που ακολουθεί ανταλλάσσονται οι πελάτες που απεικονίζονται με άσπρους κύκλους.



Εικόνα 10: κυκλική μεταφορά

Χειριστής GENI

Η μέθοδος GENI[24] (Εικόνα 11) είναι μία επέκταση της επανατοποθέτησης γειτονιών (σελ17), κατά την οποία ένας πελάτης i μπορεί επίσης να εισαχθεί μεταξύ των δύο πελατών της διαδρομής δέκτη j,k, οι οποίοι είναι πλησιέστερα σε αυτόν, ακόμα κι αν αυτοί οι κόμβοι δεν ήταν διαδοχικά δρομολογημένοι αρχικά.



Εικόνα 11: Ο χειριστής GENI- ανταλλαγής

Παράλληλη κατασκευαστική και βελτιωτική μέθοδος

Πρόκειται για μία μέθοδο, η οποία κατασκευάζει και βελτιώνει διάφορες διαδρομές ταυτόχρονα, η οποία προτάθηκε από τους Antes et al[25]. Η προσέγγιση βασίζεται στο σκεπτικό «διαπραγμάτευσης» μεταξύ πελατών και διαδρομών. Αρχικά κάθε μη δρομολογημένος πελάτης απαιτεί από κάθε διαδρομή το κόστος δρομολόγησης και στη συνέχεια στέλνει μια «προσφορά» της φθηνότερη τιμής. Κάθε διαδρομή διαλέγει τη συμφερότερη προσφορά. Οι τιμές υπολογίζονται όπως και στον I1. Όταν μία δυνατή λύση κατασκευαστεί, ο αριθμός των διαδρομών ελαττώνεται κατά μία και το πρόβλημα ξαναλύνεται. Οι συγγραφείς προτείνουν επίσης την ίδια προσέγγιση για βελτίωση λύσεων, όπου κάποιοι από τους «δύσκολους» πελάτες αφαιρούνται από τις διαδρομές και στη συνέχεια ξαναδρομολογούνται χρησιμοποιώντας τη διαδικασία που περιγράφεται παραπάνω.

LNS

Ο αλγόριθμος «έρευνα μεγάλης γειτονιάς» (large neighborhood search, LNS) βασίζεται στην αναδρομολόγηση επιλεγμένων πελατών χρησιμοποιώντας τεχνικές CP και προτάθηκε από τον Shaw το 1997[26]. Ο αλγόριθμος λειτουργεί επιλέγοντας με τυχαίο τρόπο ένα σύνολο δρομολογημένων πελατών. Οι επιλεγμένοι πελάτες αφαιρούνται από τα δρομολόγιά τους και στη συνέχεια επανεισέρχονται σε διαδρομές με το ελάχιστο κόστος. Προκειμένου να υπάρχουν αρκετές ευκαιρίες για ανταλλαγή πελατών μεταξύ διαφορετικών δρομολογίων, οι αφαιρούμενοι πελάτες επιλέγονται έτσι ώστε να έχουν κοινά στοιχεία. Τα κοινά στοιχεία μπορεί να είναι η γεωγραφική συγγένεια, η εξυπηρέτηση από το ίδιο όχημα, η παρόμοια ζήτηση αγαθών και άνοιγμα χρονικών παραθύρων την ίδια χρονική στιγμή. Για την αναδρομολόγηση των αφαιρεμένων επισκέψεων χρησιμοποιείται μία μέθοδος branch-and-bound[30] που συνδυάζεται με τη μέθοδο CP. Στην αρχική λύση, κάθε πελάτης εξυπηρετείται από ακριβώς ένα ξεχωριστό όχημα. Λόγω των υψηλών απαιτήσεων σε υπολογιστική ισχύ, αυτή η προσέγγιση μπορεί να εφαρμοστεί μόνο σε προβλήματα όπου ο αριθμός των πελατών ανά διαδρομή είναι σχετικά μικρός. Παρόμοια με την LNS αλλά με λίγο διαφορετικά κριτήρια κατά την αναδρομολόγηση είναι η LDS[27] που παρουσίασε την επόμενη χρονιά.

Ruin and recreate

Πρόκειται για μία μέθοδο παρόμοια με την LNS, την οποία προτείνει ο Schrimpf[28]. Αρχικά χρησιμοποιούνται τρεις στρατηγικές ώστε να αφαιρεθεί ένα μέρος των πελατών από μία λύση. Οι αφαιρεμένοι πελάτες δρομολογούνται πάλι με τυχαία σειρά χρησιμοποιώντας έναν πλεονεκτικό ευρετικό εισχώρησης. Για τη διαδικασία αφαίρεσης επιλέγεται στην τύχη ένας «πελάτης-κλειδί» από κάθε διαδρομή. Στη συνέχεια βάσει κάποιων κριτηρίων, που βασίζονται στην απόσταση των πελατών με τον «πελάτη-κλειδί», επιλέγονται οι πελάτες που θα αφαιρεθούν από τη διαδρομή. Κατά την έρευνα, λύσεις που χειροτερεύουν την τιμή της αντικειμενικής συνάρτησης, γίνονται αποδεκτές, εφόσον δεν ξεπερνιούνται κάποια όρια. Δεν υπάρχουν δεδομένα για τον υπολογιστικό χρόνο και η μέθοδος έχει στοχαστικά στοιχεία, όμως τα αποτελέσματα της μεθόδου είναι από τα καλύτερα όπως φαίνεται (Πίνακας 3).

LDS παραλλαγή

Η συγκεκριμένη παραλλαγή της μεθόδου LDS προτάθηκε από τους Caseau et al. [29] και είναι σχεδιασμένη για προβλήματα με πολλούς πελάτες. Οι βελτιωτικές μέθοδοι είναι ενσωματωμένες στο κατασκευαστικό μέρος. Στη βάση του ο αλγόριθμος χρησιμοποιεί παράλληλη εισαγωγή του φθηνότερου πελάτη, η οποία υπό προϋποθέσεις, μπορεί να δεχτεί τη δεύτερη καλύτερη εναλλακτική διαδρομής για κάθε πελάτη. Κατά την κατασκευή της λύσης, τρεις κινήσεις χρησιμοποιούνται μετά από κάθε εισχώρηση, η 2-opt*, η επαναεισχώρηση μιας αλυσίδας συνεχομένων πελατών από μία διαδρομή r σε μία άλλη r', όπως και η απλή μεταφορά πελάτη.

Όταν δεν είναι δυνατή η εισχώρηση πελάτη, χρησιμοποιούνται τρεις διαφορετικές κινήσεις προκειμένου να δημιουργήσουν χώρο για τον αδρομολόγητο πελάτη. Η πρώτη κίνηση, *swap (ανταλλαγή)*, αφαιρεί μία αλυσίδα συνεχομένων πελατών από την r και τους εισάγει σε μια άλλη r'. Η δεύτερη κίνηση, *ανατοποθέτηση*, αφαιρεί έναν πελάτη από μία διαδρομή r και τον εισάγει σε μια άλλη r', η οποία ίσως απαιτεί με τη σειρά της να αφαιρεθεί ένας δικός της πελάτης και να εισαχθεί σε τρίτη διαδρομή κλπ, ενώ ακολουθεί βελτίωση της κάθε διαδρομής στην οποία γίνεται η κίνηση. Η τελευταία κίνηση, *flush and relocate* (έκπλυση και ανατοποθέτηση), πρώτα αφαιρεί από την r όλους τους πελάτες οι οποίοι μπορούν να ανατοποθετηθούν απευθείας σε άλλη διαδρομή (χωρίς δηλαδή να γίνει κάποια άλλη ενέργεια στις διαδρομές-δέκτες), πρωτού προσπαθήσει να εισαχθεί αδρομολόγητος πελάτης. Σε περιπτώσεις όπου ο αριθμός των πελατών σε μια διαδρομή είναι μικρότερος από 30, η σειρά των πελατών μέσα στη διαδρομή βελτιώνεται χρήσει μιας CP με την branch-and-bound[30]. Διαφορετικά, σε μεγαλύτερες διαδρομές, χρησιμοποιείται ο 3-opt χειριστής προκειμένου να μετατραπεί η διαδρομή μετά από μία εισχώρηση. Οι συγγραφείς επίσης προσπαθούν να περιορίσουν τους πελάτες που περιλαμβάνονται σε κάθε διαδρομή σε μία γεωμετρική περιοχή.

Παράλληλη εισχώρηση 2 φάσεων

Ο αλγόριθμος αποτελείται από μία μέθοδο παράλληλης εισχώρησης για ομαδοποίηση και μία γραμμική επαναληπτική μέθοδο[31]. Το αρχικό κριτήριο για τον αλγόριθμο είναι η ελαχιστοποίηση της ολικά διανυθείσας απόστασης, και όχι ο ολικός αριθμός χρησιμοποιούμενων φορτηγών. Το δεύτερο κριτήριο είναι η ελαχιστοποίηση της ολικής αναμονής των πελατών. Οι τυχαίοι πελάτες επιλέγονται εντοπίζοντας τους πελάτες που δεν μπορούν να εξυπηρετηθούν στην ίδια διαδρομή λόγω φραγμών. Οι εναπομείναντες πελάτες εισχωρούνται σε αυτές τις αρχικές διαδρομές έτσι ώστε η αύξηση της διανυθείσας διαδρομής και του χρόνου αναμονής να ελαχιστοποιείται. Οι πελάτες με μικρό αριθμό δυνατών εισχωρήσεων και που η διαφορά ανάμεσα στην καλύτερη και τη δεύτερη καλύτερη θέση εισχώρησης είναι μικρή, προτιμώνται για ομαδοποιήσεις. Στο τέλος αυτού του σταδίου, της ομαδοποίησης, οι ομάδες αναμορφώνονται με τις βελτιωτικές μεθόδους Or-opt και 2-opt. Κατά το στάδιο δρομολόγησης, ο σκοπός τους προγραμματιστικού μοντέλου είναι να αναλύσει το πρόβλημα σε δύο γραμμικά υποπροβλήματα, όπου είτε η ολική απόσταση είτε ο χρόνος αναμονής ελαχιστοποιείται πρώτα. Οι συγγραφείς αναφέρουν ελαφρώς καλύτερα αποτελέσματα από τους Potvin και Rousseau (1993), αλλά απαιτείται μεγαλύτερος χρόνος υπολογισμού.

Cordone et al. 2001

Προτείνεται ένας ντετερμινιστικός ευρετικός αλγόριθμος ο οποίος βασίζεται στις κλασσικές k-opt ανταλλαγές, που συνδυάζονται με μία διαδικασία μείωσης του αριθμού των διαδρομών. Το ιδιαίτερο χαρακτηριστικό αυτού του αλγορίθμου είναι ότι εναλλάσσεται μεταξύ ελαχιστοποίησης της ολικής απόστασης και της ολικής διάρκειας της διαδρομής, προκειμένου να βγει από την περιοχή των τοπικών ελαχίστων[32]. Ο αλγόριθμος κατασκευάζει ένα σύνολο αρχικών λύσεων χρησιμοποιώντας τον αλγόριθμο του Solomon I1, και εφαρμόζει μια διαδικασία τοπικής έρευνας (ανταλλάζοντας 2 ή 3 τόξα) σε κάθε μία από αυτές και διαλέγει την καλύτερη. Η διαδικασία ελάττωσης της διαδρομής, προσπαθεί να εισάγει κάθε πελάτη μιας διαδρομής σε μια άλλη διαδρομή. Εάν η απλή εισχώρηση αποτύχει, χρησιμοποιείται η απλή εξαγωγή αλυσίδας (Glover 1991, 1992) όπου ένας πελάτης c_j, αφαιρείται από τη διαδρομή στόχο, r_n, και εισέρχεται σε άλλη διαδρομή, r_m, πριν εισαχθεί ο πελάτης c_i στην r_n. Η μέθοδος έχει πολύ καλά αποτελέσματα, αλλά μεγάλο υπολογιστικό κόστος (βλ. Εικόνα 12).

Braysy 2003

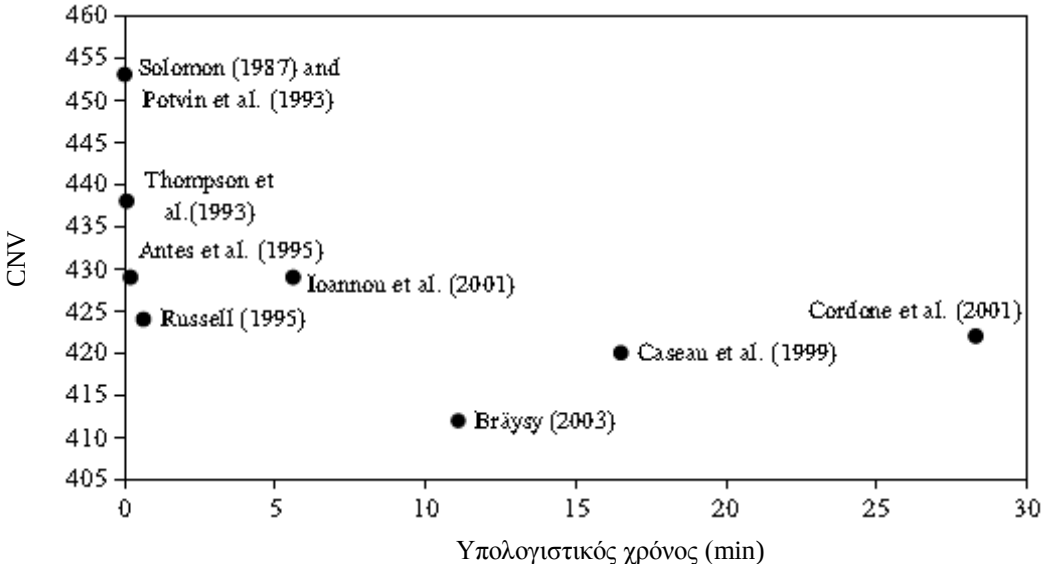
Προτείνεται μία μέθοδος με τρεις διαφορετικές προσεγγίσεις[33]. Δημιουργούνται αρχικές λύσεις χρησιμοποιώντας διάφορους συνδυασμούς παραμέτρων. Στη συνέχεια χρησιμοποιείται μια μέθοδος εξαγωγής αλυσίδων προκειμένου να μειωθεί ο αριθμός των φορτηγών. Στην τρίτη φάση, χρησιμοποιείται ο Or-opt χειριστής για να μειωθεί η ολικά διανυθείσα απόσταση.

Αποτελεσματικότητα μεθόδων

Όπως αναφέρθηκε (σελ.7) υπάρχουν διαφορετικά κριτήρια αξιολόγησης αλγορίθμων. Αρχικά εξετάζονται τα δύο πρώτα κριτήρια, τα οποία είναι και τα πιο σημαντικά, η ποιότητα λύσης και ο υπολογιστικός χρόνος[ix]. Για τον λόγο αυτόν στην Εικόνα *12* απεικονίζεται ο αριθμός των χρησιμοποιούμενων φορτηγών (CNV) στο σύνολο των 56 προβλημάτων του Solomon προς τον απαιτούμενο υπολογιστικό χρόνο. Δυστυχώς δεν υπάρχουν αρκετά δεδομένα ώστε να πραγματοποιηθεί σύγκριση με όλες τις προαναφερθείσες μεθόδους.

Ο υπολογιστικός χρόνος παρουσιάζεται υπολογισμένος με βάση την υπολογιστική δύναμη του Sun spark 10[34]. Σε συγκεκριμένο χρόνο η μέθοδος που δίνει το καλύτερο αποτέλεσμα ονομάζεται Pareto optimal. Ο απαιτούμενος υπολογιστικός χρόνος για τους Antes et al. (1995) και Thompson et al. (1993) είναι σχεδόν ίδιος, οπότε Pareto optimal ανάλογα με τον χρόνο που αποφασίζεται δίνουν οι μέθοδοι των Antes et al. (1995), Russell (1995) και Bräysy(2003).

[ix] Επειδή δεν έχουν όλοι οι αλγόριθμοι όνομα, χρησιμοποιούνται τα ονόματα των συγγραφέων τους.



Εικόνα 12: αποτελεσματικότητα των μεθόδων

Σε θέματα επαναληψιμότητας, δεν υπάρχουν δεδομένα που να υποδεικνύουν πως τα αποτελέσματα παράγονται κάθε φορά με διαφορετική ποιότητα. Είναι δηλαδή αλγόριθμοι ντετερμινιστικής φύσης.
Τέλος για τον έλεγχο δυναμικής των αλγορίθμων παρατίθεται ο Πίνακας 3, όπου φαίνεται πως ανά ομάδα έχουν παρόμοιας ποιότητας αποτελέσματα.

Πίνακας 3: Σύγκριση μεθόδων ανά ομάδα προβλημάτων

Συγγραφέας	R1	R2	C1	C2	RC1	RC2	CNV/CTD
Thompson et al. (1993)	13,00	3,18	10,00	3,00	13,00	3,71	438,00
	1356,92	1276,00	916,67	644,63	1514,29	1634,43	68916,00
Potvin et al. (1995)	13,33	3,27	10,00	3,13	13,25	3,88	448,00
	1381,90	1293,40	902,90	653,20	1545,30	1595,10	69285,00
Russell (1995)	12,66	2,91	10,00	3,00	12,38	3,38	424,00
	1317,00	1167,00	930,00	681,00	1523,00	1398,00	65827,00
Antes et al. (1995)	12,83	3,09	10,00	3,00	12,50	3,38	429,00
	1386,46	1366,48	955,39	717,31	1545,92	1598,06	71158,00
Prosser et al. (1996)	13,50	4,09	10,00	3,13	13,50	5,13	471,00
	1242,40	977,12	843,84	607,58	1408,76	1111,37	58273,00
Shaw (1997)	12,31		10,00		12,00		
	1205,06		8,28,38		1360,40		
Shaw (1998)	12,33		10,00		11,95		
	1201,79		828,38		1364,17		
Caseau et al. (1999)	12,42	3,09	10,00	3,00	12,00	3,38	420,00
	1233,34	990,99	828,38	596,63	1403,74	1220,99	58927,00
Schrimpf et al. (2000)	12,08	2,82	10,00	3,00	11,88	3,38	412,00
	1211,53	949,27	828,38	589,86	1361,76	1097,63	56830,00
Cordone et al. (2001)	12,50	2,91	10,00	3,00	12,38	3,38	422,00
	1241,89	995,39	834,05	591,78	1408,87	1139,70	58481,00
Bräysy (2003)	12,17	2,82	10,00	3,00	11,88	3,25	412,00
	1253,24	1039,56	832,88	593,49	1408,44	1244,96	59945,00

Μεταευρετικοί αλγόριθμοι

Ορισμός

Όπως αναφέρθηκε νωρίτερα (σελ.11) οι μεταευρετικοί αλγόριθμοι λειτουργούν με στρατηγικές που στηρίζονται στην εμπειρία λύσεων άλλων προβλημάτων. Χρησιμοποιούν πολλές τεχνικές από τους κατασκευαστικούς και τους βελτιωτικούς αλγόριθμους, ενώ κάνουν χρήση της γνώσης που αποκτούν εξερευνώντας την περιοχή λύσεων που επιλέγεται κάθε φορά.

Tabu Search[35]

Η Tabu Search (TS) είναι μια επαναληπτική μέθοδος τοπικής έρευνας. Σκοπός της είναι να εξερευνήσει περιοχές λύσης, συγγενικές με την αρχική, μέχρι να βρεθεί η επιθυμητή λύση. Αντίθετα με άλλες μεθόδους τοπικής έρευνας, η TS όταν συναντάει περιοχές που τις έχει εξερευνήσει ήδη, περιοχές Tabu, παράγει άλλες λύσεις πολλές φορές χειρότερες. Ο λόγος που γίνεται αυτό είναι για να αποφευχθούν τοπικά ελάχιστα. Η χρονική περίοδος που χαρακτηρίζεται μια περιοχή Tabu αλλάζει όσο τρέχει ο αλγόριθμος. Σε μερικές περιπτώσεις ο χαρακτηρισμός Tabu άρεται, αν για παράδειγμα όλες οι υπόλοιπες λύσεις ήταν χειρότερες.

Προκειμένου να παραχθούν λύσεις υψηλής ποιότητας χρησιμοποιούνται πολλές κατασκευαστικές και βελτιωτικές τεχνικές. Συνήθως χρησιμοποιείται η διαδοχική μέθοδος I1 ενώ δεν αποκλείονται και οι άλλες που αναφέρθηκαν νωρίτερα (σελ.11). Προκειμένου να παραχθούν γειτονιές της αρχικής λύσης, χρησιμοποιούνται χειριστές που αναφέρθηκαν προηγουμένως (σελ.16).

Η ανάγκη να μειωθεί η πολυπλοκότητα της έρευνας ωθεί στη χρήση κριτηρίων που περιορίζουν τον αριθμό των πελατών που εξετάζονται. Τέτοια κριτήρια σχετίζονται με την απόσταση των πελατών υπό έρευνα, τη γωνία που σχηματίζεται με το κέντρο βάρους μιας διαδρομής κ.α. Αναφέρεται[36] πως η χρήση παράλληλης μεθόδου παράγει ίδιας ποιότητας λύσεις, οπότε όταν χρησιμοποιούνται πολλοί επεξεργαστές, μειώνεται ο χρόνος επεξεργασίας. Μερικές φορές παράγονται αδύνατες λύσεις[37] προκειμένου να ξεπεραστούν οι φραγμοί και να διαφύγει η μέθοδος από τοπικά ελάχιστα. Η μείωση των χρησιμοποιούμενων φορτηγών πραγματοποιείται με χειριστές όπως ο Or-opt.

Επίκτητη μνήμη

Οι περιοχές των λύσεων που σαρώνονται πρέπει να προέρχονται από κάποια αρχική. Όταν η περιοχή αυτή εξαντλείται, ο αλγόριθμος μπορεί να επιλέξει μια διαδρομή που να ανήκει σε άλλη λύση, η οποία παράχθηκε σε προηγούμενο στάδιο. Το σύνολο των διαδρομών που μπορεί να χρησιμοποιήσει το πρόγραμμα αποκαλείται «επίκτητη μνήμη». Η επιλογή της διαδρομής πραγματοποιείται με βάση την ποιότητα της λύσης από την οποία προέρχεται η διαδρομή. Όταν ολοκληρωθεί η έρευνα στην συγκεκριμένη διαδρομή από τον αλγόριθμο Tabu, η βελτιωμένη έκδοση της διαδρομής επανέρχεται στην επίκτητη μνήμη ώστε να χρησιμοποιηθεί αργότερα. Παρακάτω (Πίνακας 4)[38] περιγράφονται οι αρχές που εφάρμοσαν οι διάφοροι συγγραφείς σε αλγορίθμους Tabu.

Πίνακας 4: αρχές των μεθόδων Tabu της βιβλιογραφίας

Συγγραφέας	Έτος	Κατασκευαστικός	Βελτιωτικός	CNV[x]	σχόλια
Garcia et al. (1)	1994	I1	2-opt, Or -opt	Ναι	Στενή γειτονιά
Rochat and Taillard (2)	1995	Παραλλαγή του I1, 2-opt,	2-opt, ανατοποθέτηση	Όχι	Adaptive memory
Carlton	1995	Insertion	ανατοποθέτηση	Όχι	Reactive tabu search
Potvin and Bengio (3)	1996	I1	2-opt*, Or -opt	Ναι	Στενή γειτονιά
Taillard et al. (4)	1997	I1	CROSS	Όχι	Soft time windows, επίκτητη μνήμη
Badeau et al.	1997	I1	CROSS	Όχι	Soft time windows, επίκτητη μνήμη
Chiang et al. (5)	1997	Παραλλαγή του Russell (1995)	λ-interchange	Όχι	Reactive tabu search
De Backer and Furnon (6)	1997	Savings	ανταλλαγή, ανατοποθέτηση, 2-opt* 2-opt,Or -opt	Όχι	Χρήση CP
Brandão (7)	1999	Insertion	ανατοποθέτηση, ανταλλαγή, GENI	Όχι	Στενή γειτονιά
Schulze and Fahle (8)	1999	I1,παράλληλη I1 και savings	Εξαγωγικές αλυσίδες, Or -opt	Ναι	Αποθήκευση παραγόμενων διαδρομών
Tan et al. (9)	2000	Insertion του Thangiah et al. (1994)	λ-ανταλλαγή, 2-opt*	Όχι	—
Lau et al. (10)	2001	Insertion	ανταλλαγή, ανατοποθέτηση	Όχι	Χρήση CP
Cordeau et al. (11)	2001	Παραλλαγή του sweep	ανατοποθέτηση, GENI	Όχι	—
Lau et al. (12)	2003	Ανατοποθέτηση από λίστα αναμονής	Ανταλλαγή, ανατοποθέτηση	Ναι	Χρήση λίστας αναμονής και ανώτερο όριο δρομολογίων

Πίνακας 5: αποτελεσματικότητα των μεθόδων του Πίνακα 4

Συγ/φέας	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
CNV	436	415	426	410	411	508	425	414	467	464	407	418
CTD	65.977	57.231	63.530	57.523	58.502	56.998	58.562	60.346	62.008	58.432	57.556	58.477

Ο παραπάνω συμπληρωματικός Πίνακας 5, δίνει τις τιμές των αντικειμενικών συναρτήσεων των μεθόδων που περιγράφονται στον Πίνακα 4. Με τη πρώτη ματιά ξεχωρίζουν η λύση των De Backer and Furnon (1997) ως κακή, αν και έχουν χρησιμοποιήσει πιο πολλούς χειριστές από τους άλλους συγγραφείς. Αντίθετα σαν πολύ καλή λύση φαίνεται αυτή των Cordeau et al. (2001) με 407 φορτηγά. Σύγκριση σε υπολογιστικό χρόνο δεν είναι δυνατόν να γίνει γιατί δεν δίνουν όλοι αρκετά στοιχεία για τη σύγκριση.

[x] Η στήλη CNV δείχνει το αν η αντικειμενική συνάρτηση είναι ο αριθμός των χρησιμοποιούμενων φορτηγών. Όταν η τιμή είναι «Όχι», σκοπός ήταν η ελαχιστοποίηση της διανυθείσας απόστασης, ενώ σαν δεύτερο κριτήριο αντί του διανυθείσας απόστασης είναι ο ταξιδιωτικός χρόνος.

Γενετικοί αλγόριθμοι

Οι γενετικοί αλγόριθμοι (Genetic Algorithms, GA) εμπνεύστηκαν από τη βιολογική μεταφορά πληροφορίας από γενιά σε γενιά. Χρειάζονται 2 γονείς που θα προσφέρουν χρωμοσώματα, τα οποία θα συνδυαστούν με κατάλληλο τρόπο έτσι ώστε να παραχθεί μία νέα γενιά. Σαν τρόπος σκέψης έχει ευρύ φάσμα εφαρμογής, ενώ η ιδέα είναι κατανοητή. Η γενετική έρευνα προϋποθέτει πως το πρόβλημα μπορεί να παρουσιαστεί με ακεραίους, δηλαδή αναφέρεται κυρίως σε διακριτή βελτιστοποίηση. Στην υπολογιστική γλώσσα, χρειάζονται παραπάνω από μία λύσεις και οι πληροφορίες που θα χρησιμοποιηθούν θα είναι από διαφορετικές λύσεις.

Η βάση του αλγορίθμου στηρίζεται στην εξέλιξη της πληροφορίας με τέτοιο τρόπο, ώστε να επιβιώσει η καλύτερη. Ο τερματισμός της μεθόδου πραγματοποιείται όταν πληρούνται κάποια κριτήρια όπως: αριθμός γενεών, ή η ομοιότητα της παραχθείσας λύσης με κάποιον γονέα. Στο τελευταίο στάδιο το καλύτερο χρωμόσωμα που παράγεται, αποκωδικοποιείται και δίνει την αντίστοιχη λύση.

Κάθε γενεά δημιουργείται σε τέσσερις φάσεις: *αντιπροσώπευση, επιλογή, ανασυνδυασμός και μετάλλαξη.* Κατά την αντιπροσώπευση μορφώνεται το χρωμόσωμα σύμφωνα με τα σημαντικά χαρακτηριστικά του κάθε γονέα. Κατά την επιλογή, διαλέγονται τα καλύτερα χρωμοσώματα. Τα κριτήρια επιλογής σχετίζονται με την ικανότητα του χρωμοσώματος να συνδυαστεί με άλλα, όμως γίνεται με στοχαστικό τρόπο βασισμένο συνήθως στις πιθανότητες. Στο στάδιο του ανασυνδυασμού, χρησιμοποιούνται τα χρωμοσώματα των επιλεγμένων γονέων για την παραγωγή της νέας γενιάς. Τέλος κατά τη μετάλλαξη πραγματοποιείται τυχαία μεταβολή κάποιων χρωμοσωμάτων προκειμένου να εξερευνηθεί ο χώρος λύσης και να διατηρηθεί η γενετική ποικιλία. Με την επανάληψη των τριών τελευταίων βημάτων δημιουργείται μια νέα γενιά. Κάθε γενιά αποτελείται από συγκεκριμένο αριθμό χρωμοσωμάτων, που έγκειται στις επιλογές του χρήστη. Ανάλογα με τον τύπο του GA που χρησιμοποιείται, αντικαθίσταται μέρος ή το σύνολο των αρχικών χρωμοσωμάτων.

Ο GA σαν μεταευρετικός χρειάζεται να συνδυάσει ευρετικούς προκειμένου να παράξει λύσεις. Η πρώτη προσέγγιση[39] στα VRPTW έγινε με τη χρήση του κατασκευαστικού «cluster-first, route second», όπου σκοπός ήταν να γίνει καλή ομαδοποίηση, ενώ στη συνέχεια η δρομολόγηση έγινε με αλγόριθμο insertion και τέλος η λύση βελτιώθηκε με ανταλλακτικούς χειριστές. Όλο και περισσότεροι ασχολούνται με τη χρήση υβριδικού GA με κατασκευαστικούς και με τοπικής έρευνας αλγορίθμους ακόμα και με άλλους μεταευρετικούς όπως ο Tabu[40] και ο αλγόριθμος μυρμηγκιών[41].

Παρόμοιες τεχνικές με αυτήν της γενετικής έρευνας είναι ο *εξελικτικός προγραμματισμός* και οι *εξελικτικές στρατηγικές*. Στο σύνολό τους τα τρία είδη σχηματίζουν τους *εξελικτικούς αλγορίθμους*. Οι διαφορές τους αφορούν το πρώτο και το τελευταίο στάδιο (αντιπροσώπευση και μετάλλαξη). Κατά την αντιπροσώπευση στις γενετικές στρατηγικές, προστίθενται νέες παράμετροι, οι λεγόμενες *παράμετροι στρατηγικής* «strategy parameters». Οι νέες αυτές παράμετροι μαζί με το σύνολο των λύσεων εξελίσσονται με τη βοήθεια των χειριστών ανασυνδυασμού και μετάλλαξης. Συγκεκριμένα στα VRPTW οι παράμετροι αναφέρονται στο πόσο συχνά θα επιλέγεται (τυχαία) κάθε χειριστής τοπικής έρευνας και στην εναλλαγή μεταξύ των δύο αντικειμενικών συναρτήσεων (CNV και CTD). Κατά τον ανασυνδυασμό των γονέων παράγεται μόνο ένας απόγονος. Με αυτόν τον τρόπο παράγονται λ>μ απογόνων, όπου μ

είναι το μέγεθος του πληθυσμού των γονέων που χρησιμοποιούνται. Στο τέλος επιλέγονται μ απόγονοι για τη δημιουργία της νέας γενιάς.
Ο ανασυνδυασμός είναι το πιο σημαντικό μέρος του αλγορίθμου και μπορεί να πραγματοποιηθεί με πληθώρα τεχνικών, όπως η γενετική διασταύρωση, η οποία ανταλλάζει τυχαία μέρη των δρομολογίων. Μία άλλη μέθοδος, η οποία χρησιμοποιείται πολύ, είναι η PMX. Αρχικά επιλέγονται δύο σημεία (πελάτες) ενός γονέα και αποκόβονται τα υπόλοιπα κομμάτια. Στη συνέχεια προστίθενται άλλοι πελάτες με τη μέθοδο της ανταλλαγής. Τέλος υπάρχει και η γενετική ανταλλαγή ενός σημείου, η οποία ανταλλάζει δύο σύνολα πελατών με σκοπό να εξυπηρετηθούν από δύο διαφορετικές διαδρομές. Η μετάλλαξη πραγματοποιείται χρησιμοποιώντας χειριστές που αφαιρούν και προσθέτουν πελάτες χρησιμοποιώντας τον χειριστή Or-opt.

Τα αποτελέσματα διαφόρων γενετικών μεθόδων συγκεντρωμένα

Πίνακας 6)[38] δείχνουν πως η ποιότητα λύσης μπορεί να είναι πολύ καλή έως μέτρια όσον αφορά τα CNV. Παρατηρείται μάλιστα πως τα C1 και C2 έχουν παρόμοια ποιότητα λύσης για όλους τους αλγορίθμους, η διαφορά όμως φαίνεται στις άλλες ομάδες.

Πίνακας 6: Αποτελεσματικότητα γενετικών μεθόδων

Authors	R1	R2	C1	C2	RC1	RC2	CNV/CTD
(1) Thangiah (1995a)	12.75	3.18	10.00	3.00	12.50	3.38	429
	1,300.25	1,124.28	892.11	749.13	1,474.13	1,411.13	65,074
(2) Potvin and Bengio (1996)	12.58	3.00	10.00	3.00	12.13	3.38	422
	1,296.83	1,117.64	838.11	590.00	1,446.25	1,368.13	62,634
(3) Berger et al. (1998)	12.58	3.09	10.00	3.00	12.13	3.50	424
	1,261.58	1,030.01	834.61	594.25	1,441.35	1,284.25	60,539
(4) Homberger and Gehring (1999)	11.92	2.73	10.00	3.00	11.63	3.25	406
	1,228.06	969.95	828.38	589.86	1,392.57	1,144.43	57,876
(5) Gehring and Homberger (1999)	12.42	2.82	10.00	3.00	11.88	3.25	415
	1,198	947	829	590	1,356	1,140	56,942
(6) Gehring and Homberger (2001)	12.00	2.73	10.00	3.00	11.50	3.25	406
	1,217.57	961.29	828.63	590.33	1,395.13	1,139.37	57,641
(7) Berger et al. (2003)	11.92	2.73	10.00	3.00	11.50	3.25	405
	1,221.10	975.43	828.48	589.93	1,389.89	1,159.37	57,952
(8) Tan et al. (2001a)	13.17	5.00	10.11	3.25	13.50	5.00	478
	1,227	980	861	619	1,427	1,123	58,605
(9) Tan et al. (2001b)	12.91	5.00	10.00	3.00	12.60	5.80	471
	1,205.0	929.6	841.96	611.2	1,392.3	1,080.1	56,931
(10) Wee Kit et al. (2001)	12.58	3.18	10.00	3.00	12.75	3.75	432
	1,203.32	951.17	833.32	593.00	1,382.06	1,132.79	57,265
(11) Mester (2002)	12.00	2.73	10.00	3.00	11.50	3.25	406
	1,208	954	829	590	1,387	1,119	57,219
(12) Jung and Moon (2002)	13.25	5.36	10.00	3.00	13.00	6.25	486
	1,179.95	878.41	828.38	589.86	1,343.64	1,004.21	54,779
(13) Le Bouthillier and Crainic (2005)	12.17	2.82	10.00	3.00	11.50	3.25	409
	1,209.27	965.91	828.38	589.86	1,389.22	1,143.70	57,574
(14) Homberger and Gehring (2005)	11.92	2.73	10.00	3.00	11.50	3.25	405
	1,212.73	955.03	828.38	589.86	1,386.44	1,123.17	57,309

Όλοι οι παραπάνω αλγόριθμοι είναι στοχαστικοί. Όλοι τους συμφωνούν με τα κριτήρια που τέθηκαν στην αρχή (Ποιότητα Λύσης σελ.7) εκτός από τους 8,9 και 12, που είχαν σαν σκοπό την ελαχιστοποίηση του CTD και μάλιστα οι Jung and Moon (2002) τα κατάφεραν πολύ καλά. Pareto Optimal για τις μεθόδους που παρέχουν αρκετά δεδομένα είναι οι 4,5,6 και 12, γιατί είναι και οι μόνες που αναφέρουν πόσες φορές έτρεξε ο αλγόριθμος ώστε να παραχθούν τα αναφερόμενα αποτελέσματα. Παρατηρείται πως η βέλτιστη λύση για την κάθε ομάδα είναι:

Ομάδα	R1	R2	C1	C2	RC1	RC2
μ.ο. φορτηγών	11,92	2,73	10,00	3,00	11,50	3,25

Επίσης παρατηρείται πως οι δύο βέλτιστες λύσεις για τους (7) και (9) βρίσκουν αυτές τις τιμές και διαφέρουν μεταξύ τους στη διανυθείσα απόσταση ελάχιστα.

Άλλοι μεταευρετικοί

Εκτός από τους αλγόριθμους Tabu και γενετικής έρευνας, υπάρχει πληθώρα άλλων μεταευρετικών. Στη συνέχεια θα αναφερθούν οι σημαντικότεροι.

GRASP

Το όνομα αυτού του αλγορίθμου προέρχεται από τα αρχικά των λέξεων «πλεονεκτική τυχαία προσαρμόσιμη διαδικασία έρευνας» *Greedy randomized adaptive search procedure* και προτάθηκε από τους Kontoravdis et al[42]. Οι αρχικοί πελάτες τοποθετούνται σε ένα αριθμό διαδρομών, με κριτήριο το κατά πόσο είναι διασκορπισμένοι γεωγραφικά και πιο παρεμποδισμένοι χρονικά. Στη συνέχεια ο αλγόριθμος βρίσκει την καλύτερη θέση για εισαγωγή κάθε πελάτη σε κάθε διαδρομή και υπολογίζει το κόστος εισαγωγής χρήσει του Solomon I1. Σε κάθε πελάτη δίνεται «πρόστιμο» ανάλογα με το πόσο δύσκολα μπορεί να δρομολογηθεί. Οι L πιο «δύσκολοι» πελάτες μπαίνουν σε μια λίστα και από αυτήν επιλέγεται τυχαία ποιος θα δρομολογηθεί. Μετά από 5 επαναλήψεις πραγματοποιείται τοπική έρευνα ώστε να παραχθεί η καλύτερη δυνατή λύση. Γίνεται προσπάθεια οι διαδρομές με λίγους πελάτες να αφαιρούνται.
Η εκτίμηση του αριθμού των διαδρομών γίνεται με τρία κριτήρια. Πρώτον: βάσει του αριθμού των φορτηγών που χρειάζεται για να καλυφθεί η ζήτηση του προβλήματος, δεύτερον: βάσει των αδύνατων συνδυασμών λόγω χρονικών παραθύρων ή/και λόγω ζήτησης και τρίτον: ο χρόνος που χρειάζεται κάθε φορτηγό για να φτάσει στον κοντινότερο πελάτη ή ο χρόνος χρειάζεται για να πάει ένα φορτηγό από την αποθήκη στον πρώτο πελάτη. Σαν μέθοδος θεωρείται πολύ γρήγορη, όμως τα αποτελέσματά της δεν είναι τα καλύτερα (βλ. Εικόνα 13)

Νευρικό δίκτυο

Η μέθοδος[43] του νευρικού δικτύου χρησιμοποιείται για την επιλογή των αρχικών πελατών και συνδυάζεται με μια παράλληλη μέθοδο φθηνότερης εισχώρησης (cheapest insertion). Κάθε όχημα αντιστοιχίζεται σε ένα άνυσμα - «νευρώνα». Στην αρχή όλα τα ανύσματα τοποθετούνται κοντά στην αφετηρία στην τύχη. Στη συνέχεια υπολογίζεται η απόσταση κάθε πελάτη από κάθε άνυσμα. Το άνυσμα που βρίσκεται κοντύτερα στον κάθε πελάτη ενημερώνεται και μεταφέρεται πιο κοντά στον πελάτη. Αυτό επαναλαμβάνεται για όλους τους πελάτες μερικές φορές. Κάθε φορά που ξαναρχίζει η διαδικασία, το εύρος κίνησης του ανύσματος μικραίνει. Στο τέλος της φάσης αυτής, οι αρχικοί πελάτες επιλέγονται ως οι κοντινότεροι των απολήξεων των ανυσμάτων.

GLS καθοδηγούμενη τοπική έρευνα

Η καθοδηγούμενη τοπική έρευνα προτάθηκε βασίζεται στην επίκτητη μνήμη[44]. Όταν η μέθοδος παράγει λύσεις που πλησιάζουν προηγούμενα τοπικά ελάχιστα, στο κόστος της λύσης προστίθεται ποινή, πραγματοποιώντας διαφοροποίηση. Προφανώς όσες περισσότερες ποινές έχει δεχτεί μία λύση, τόσο δυσκολότερο γίνεται να ξαναπροστεθεί ποινή σε αυτήν. Η αρχική λύση δεν περιέχει κανένα δρομολογημένο πελάτη, γεγονός που οδηγεί τον αλγόριθμό να παράξει λύσεις. Ο αλγόριθμος κάνει χρήση τεσσάρων τεχνικών τοπικής έρευνας (2-opt, ανατοποθέτηση, ανταλλαγή και 2-opt* με τη μέθοδο καλύτερη αποδοχής (BA). Η μέθοδος είναι ολοκληρωτικά ντετερμινιστική.

SA Προσομοίωση σκλήρυνσης

Η ιδέα προέρχεται από τη στατιστική μηχανική και σχετίζεται με το φαινόμενο σκλήρυνσης στερεών μέσω θέρμανσης και αργής ψύξης ώστε να επιτευχθεί κρυσταλλοποίηση χαμηλής ενέργειας *Simulated annealing (SA)* [45]. Η μέθοδος είναι στοχαστική. Η SA οδηγεί την τοπική έρευνα ώστε να γίνει αποδεκτή η λύση S' αν $\Delta \leq 0$, όπου $\Delta = C(S') - C(S)$. Προκειμένου να μην παγιδευτεί η μέθοδος σε τοπικά ελάχιστα, υπάρχει η πιθανότητα κάποιες λύσεις που αυξάνουν την τιμή της αντικειμενικής συνάρτησης κατά $e^{-\Delta/T}$, να γίνουν δεκτές. Όπου $\Delta>0$ και T μία παράμετρος που ονομάζεται «θερμοκρασία». Η τιμή της T ποικίλει από πολύ μεγάλη έως σχεδόν 0. Αυτές οι τιμές ρυθμίζονται από το πρόγραμμα ψύξης.

Η μέθοδος έχει συνδυαστεί με πολλές άλλες μεθόδους. Μία πολύ ενδιαφέρουσα είναι μία που συνδυάζει Tabu, insertion και sweep ευρετικούς. Οι αρχικές λύσεις δημιουργούνται με τους κατασκευαστικούς insertion και sweep και στη συνέχεια χρησιμοποιούνται τρεις χειριστές γειτονιών που βασίζονται στην ανταλλαγή ή ανατοποθέτηση τμήματος της λύσης. Ο αλγόριθμος επαναλαμβάνεται πολλές φορές θέτοντας T= η καλύτερη θερμοκρασία. Ο insertion του Solomon χρησιμοποιείται προκειμένου να μειωθεί ο αριθμός των διαδρομών και για να εντατικοποιηθεί η έρευνα στις τρέχουσες λύσεις. Στο τέλος πραγματοποιείται τυχαία ανταλλαγή πελατών για να πραγματοποιηθεί η διαφοροποίηση.

ACO Αποικία μυρμηγκιών

Η μέθοδος ACO[46] μιμείται τη συμπεριφορά των πραγματικών μυρμηγκιών που ψάχνουν για φαγητό. Κατά την αναζήτησή τους, τα μυρμήγκια αφήνουν στο πέρασμά τους μία ορμόνη, τη φερομόνη. Η ποσότητα της φερομόνης που υπάρχει σε κάθε διαδρομή εξαρτάται από το μήκος της διαδρομής και από την ποιότητα της τροφής. Η φερμομόνη που υπάρχει στις διαδρομές, δίνει πληροφορίες στα υπόλοιπα μυρμήγκια, καθώς τα προσελκύει. Με τον καιρό, τα μονοπάτια που οδηγούν σε τροφές μεγάλης ποσότητας και κοντά στην φωλιά, γίνονται πιο πολυσύχναστες και σημαδεύονται με περισσότερη φερομόνη. Κάθε νοητό μυρμήγκι κατασκευάζει μία δυνατή λύση, με τη μέθοδο του πλησιέστερου γείτονα. Σαν κριτήρια εισαγωγής κάθε πελάτη χρησιμοποιούνται: η απόσταση μεταξύ των πελατών, πόσες φορές δεν έχει εισαχθεί ένας πελάτης σε λύση, τα χρονικά παράθυρα. Οι λύσεις βελτιώνονται με χειριστές cross-ανταλλαγής.

Χρησιμοποιούνται δύο αποικίες μυρμηγκιών με ιεραρχικό τρόπο, η μία προσπαθεί να ελαχιστοποιήσει τον αριθμό των χρησιμοποιούμενων φορτηγών ενώ η άλλη την ολικά διανυθείσα απόσταση. Οι δύο αποικίες συνεργάζονται βελτιώνοντας την καλύτερη λύση που έχει βρεθεί. Σε περίπτωση που η νέα λύση περιέχει λιγότερα φορτηγά, ενεργοποιούνται και οι δύο αποικίες και εξερευνούν την γειτονιά της λύσης.

Ανάβαση λόφου (hill climbing)

Πρόκειται για μία μέθοδο αλληλεπίδρασης, κατά την οποία παράγεται μια εικόνα και ο χρήστης καλείται να αποφασίσει προς τα πού θα κινηθεί η έρευνα[47]. Στο κατασκευαστικό κομμάτι εκτός από τον αλγόριθμο hill climbing, πραγματοποιείται ανατοποθέτηση n πελατών προκειμένου να βρεθεί ένα τοπικό ελάχιστο. Οι επιλογές του χρήστη σχετίζονται με τις περιοχές που πιστεύει πως έχουν καλές λύσεις και με το ποια κριτήρια θα επιλεχθούν BA ή FA κατά την βελτίωση της λύσης. Τελευταίο βήμα είναι η χρήση ενός branch-and-bound αλγόριθμου, προκειμένου να βελτιστοποιηθεί κάθε διαδρομή.

VNS 4 φάσεων

Πρόκειται για μια έκδοση του VNS η οποία συνδυάζει πολλούς αλγόριθμους μαζί[33]. Στην Εικόνα 13 είναι ο αλγόριθμος με την καλύτερη ποιότητα λύσης. Η αρχική λύση δημιουργείται χρησιμοποιώντας έναν κατασκευαστικό ευρετικό που δανείζεται τις βασικές ιδέες παλαιότερων μεθόδων[14,5] . Οι διαδρομές κατασκευάζονται διαδοχικά. Όταν ο αριθμός μιας διαδρομής φτάσει μια σταθερά k, οι πελάτες ξαναδρομολογούνται χρήσει του Or-opt. Στη δεύτερη φάση χρησιμοποιείται ένας τύπος εξαγωγικών αλυσίδων για να ελαχιστοποιηθεί ο αριθμός των δρομολογίων. Στην τρίτη φάση οι παραγόμενες λύσεις βελτιώνονται ως προς τη διανυθείσα απόσταση χρησιμοποιώντας τον VNS ανάμεσα σε τέσσερεις νέες βελτιωτικές μεθόδους. Αυτές οι διαδικασίες βασίζονται σε παραλλαγές του CROSS-ανταλλακτικού και σε έναν τύπο insertion. Στην τέταρτη φάση, η αντικειμενική συνάρτηση που χρησιμοποιήθηκε από τους χειριστές τοπικής έρευνας αλλάζει έτσι ώστε να περιλαμβάνει τον χρόνο αναμονής, με σκοπό να ξεφύγει από τα τοπικά ελάχιστα.

Την επόμενη χρονιά βελτιώνεται ο αλγόριθμος[48] στο κατασκευαστικό και βελτιωτικό του κομμάτι και εφαρμόζοντας μια νέα μετα-βελτιωτική τεχνική που βασίζεται στην τεχνική *threshold acceptance*, μία ντετερμινιστική έκδοση του SA. Πιο συγκεκριμένα η ανακατανομή των πελατών δεν γίνεται στην κατασκευαστική φάση ενώ χρησιμοποιείται μια έκδοση των εξαγωγικών αλυσίδων η οποία επιτρέπει αδύνατες λύσεις και την εξαγωγή κάποιων πελατών από κάθε διαδρομή στη δεύτερη φάση. Στη φάση βελτιστοποίησης της απόστασης, χρησιμοποιούνται μόνο διάφορες εκδόσεις του CROSS-ανταλλακτικού χειριστή. Μερικές φορές πραγματοποιείται αναστροφή της κατεύθυνσης σε κάποια τμήματα της διαδρομής δημιουργώντας περισσότερες θέσεις εισχώρησης. Στην τελευταία φάση, χρησιμοποιείται ένας ευρετικός που συνδυάζει τις ιδέες του CROSS και του GENIUS ανταλλακτικού.

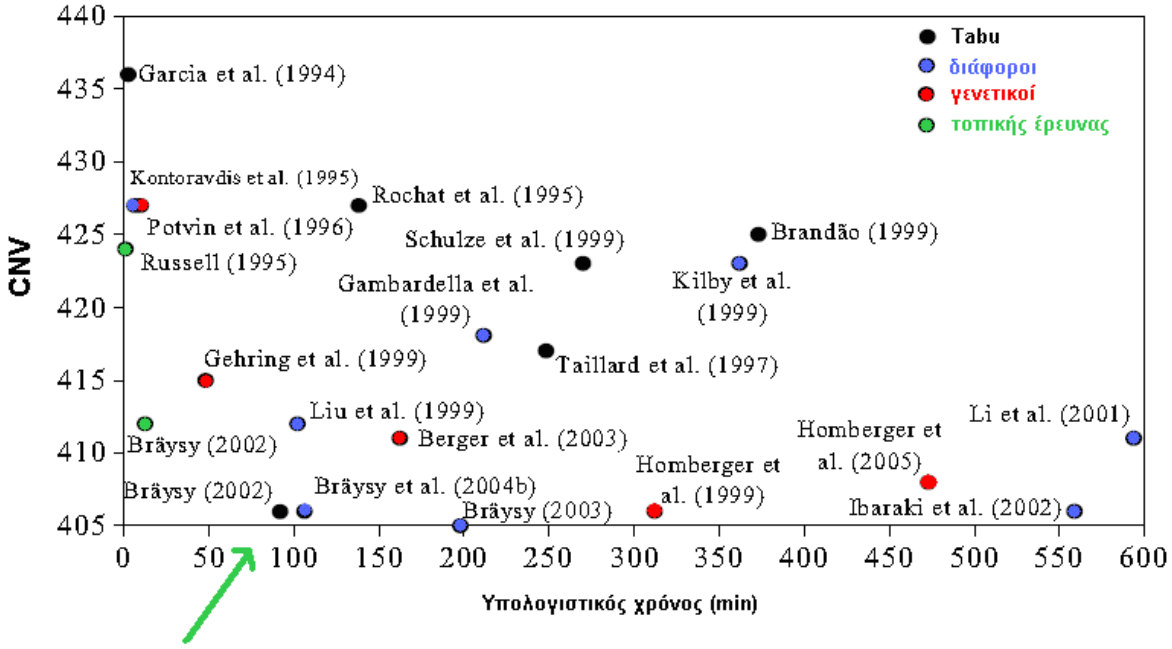
Αποτελεσματικότητα διαφόρων μεταευρετικών

Όπως φαίνεται (Πίνακας 7) κάποιες μέθοδοι παράγουν λύσεις το ίδιο ικανοποιητικές με αυτές των γενετικών αλγορίθμων. Η επίδοση 405 φορτηγά συναντιέται δύο φορές ενώ υπάρχουν και επιδόσεις με 406 και 407 φορτηγά. Όλες σχεδόν οι μέθοδοι καταφέρνουν να βρουν τη «βέλτιστη λύση» για τις ομάδες C1 και C2, πολλές για την ομάδα RC2, οι μισές περίπου για την R2 ενώ για τις RC1 και R1 φαίνεται πως είναι πολύ δύσκολο να βρεθούν. Επίσης φαίνεται πως επιτυγχάνονται πολύ καλά αποτελέσματα όσον αφορά τη διανυθείσα απόσταση. Οι ιδέες που βρίσκονται πίσω από τους μεταευρετικούς είναι ιδιαίτερα φιλοσοφημένες και δημιουργικές και ξεπερνάνε κατά πολύ τις επιδόσεις, ως προς την ποιότητα των λύσεων, των ευρετικών αλγορίθμων.
Κάποιες μέθοδοι δεν παρουσιάζουν αποτελέσματα για όλες τις ομάδες, καθώς δημιουργήθηκαν με σκοπό να λύνουν συγκεκριμένα προβλήματα.

Πίνακας 7: Αποτελεσματικότητα διαφόρων μεταευρετικών

Authors	R1	R2	C1	C2	RC1	RC2	CNV/CTD
(1) Thangiah et al. (1994)	12.33	3.00	10.00	3.00	12.00	3.38	418
	1,227.42	1,005.00	830.89	640.86	1,391.13	1,173.38	58,905
(2) Kontoravdis and Bard (1995)	12.58	3.09	10.00	3.00	12.63	3.50	427
	1,325.44	1,164.27	827.3	589.65	1,500.94	1,414.21	64,196
(3) Potvin and Robillard (1995)	13.58	3.09	10.56	3.38	13.63	3.63	457
	1,539.4	1,325.1	1,237.2	875.6	1,828.9	1,578.9	78,451
(4) Bachem et al. (1996)	12.58	3.00	—	—	12.13	3.38	—
	1,392.0	1,199.6			1,501.6	1,500.1	
(5) Chiang and Russell (1996)	12.50	2.91	10.00	3.00	12.38	3.38	422
	1,308.82	1,166.42	909.80	666.30	1,473.90	1,393.70	64,996
(6) Liu and Shen (1999)	12.17	2.82	10.00	3.00	11.88	3.25	412
	1,249.57	1,016.58	830.06	591.03	1,412.87	1,204.87	59,318
(7) Kilby et al. (1999)	12.67	3.00	10.00	3.00	12.13	3.38	423
	1,200.33	966.56	830.75	592.24	1,388.15	1,133.42	57,423
(8) Caseau et al. (1999b)	12.17	—	—	—	12.00	—	—
	1,207.27				1,356.62		
(9) Gambardella et al. (1999)	12.00	2.73	10.00	3.00	11.63	3.25	407
	1,217.73	967.75	828.38	589.86	1,382.42	1,129.19	57,525
(10) Tan et al. (2000)	14.50	3.64	10.11	3.25	14.75	4.25	483
	1,420.12	1,278.97	958.57	766.46	1,648.77	1,641.89	72,194
(11) Anderson et al. (2000)	—	—	—	—	11.63	—	—
					1,397		
(12) Tan et al. (2001c)	13.10	4.60	10.00	3.30	12.70	5.60	470
	1,213.16	952.30	841.92	612.75	1,415.62	1,120.37	57,799
(13) Bräysy (2003)	11.92	2.73	10.00	3.00	11.50	3.25	405
	1,222.12	975.12	828.38	589.86	1,389.58	1,128.38	57,710
(14) Li and Lim (2001)	12.08	2.91	10.00	3.00	11.75	3.25	411
	1,215.14	953.43	828.38	589.86	1,385.47	1,142.48	57,467
(15) Bent and Van Hentenryck (2004)	11.92	2.73	10.00	3.00	11.50	3.25	405
	1,211.10	954.27	828.38	589.86	1,384.17	1,124.46	57,273
(16) Rousseau et al. (2002)	12.08	3.00	10.00	3.00	11.63	3.38	412
	1,210.21	941.08	828.38	589.86	1,382.78	1,105.22	56,953
(17) Czech and Czarnas (2002)	—	—	—	—	11.50	3.25	—
					1,384.17	1,119.49	
(18) Bräysy et al. (2004b)	12.00	2.73	10.00	3.00	11.50	3.25	406
	1,214.69	960.44	828.38	589.86	1,389.20	1,124.14	57,422
(19) Ibaraki et al. (2002)	11.92	2.73	10.00	3.00	11.50	3.25	405
	1,217.40	959.11	828.38	589.86	1,391.03	1,122.79	57,444

Στην Εικόνα 13 δίνονται τα αποτελέσματα από μεταευρετικούς που αναφερθήκαν νωρίτερα. Στον άξονα (X) δίνεται ο χρόνος σε λεπτά, υπολογισμένος για Sun Sparc 10, και στον άξονα (Y) τα χρησιμοποιούμενα φορτηγά. Με μαύρο χρώμα σημαδεύονται οι αλγόριθμοι Tabu, με κόκκινο οι γενετικής έρευνας, με μπλε οι υπόλοιποι «διάφοροι» αλγόριθμοι, με πράσινο μέθοδοι που χρησιμοποιούν τοπική έρευνα ενώ για τον αλγόριθμο που είναι σημαδεμένος με βελάκι ίσως πρόκειται για τον ίδιο αλγόριθμο με τον άλλο Bräysy 2002 αλλά με διαφορετικές παραμέτρους. Είναι προφανές πως τα pareto optimal δίνονται από τον Russell 1995 και όλα τα υπόλοιπα από τον Bräysy (2002,2002,2003)



Εικόνα 13: σύγκριση μεταευρετικών

Μία ακόμα παρατήρηση που προκύπτει από την εικόνα είναι πως η έρευνα με το πέρασμα του χρόνου γίνεται πιο απαιτητική απαιτώντας μικρότερους χρόνους και λιγότερα φορτηγά. Εντυπωσιακό είναι το αποτέλεσμα του Russell (1995) ο οποίος με ευρετικές μεθόδους, μέσα σε λίγο υπολογιστικό χρόνο κατάφερε να ξεπερνάει μεταγενέστερες μεταευρετικές μεθόδους. Επίσης όπως φαίνεται και στους προηγούμενους πίνακες, η καλύτερη λύση που έχει παραχθεί είναι αυτή των 405 φορτηγών και μάλλον δεν υπάρχουν πολλά περιθώρια βελτίωσης.
Συμπερασματικά, όταν οι εφαρμογές απαιτούν ταχύτατα αποτελέσματα, είναι λογικό να χρησιμοποιούνται ευρετικές μέθοδοι, οι οποίες έχουν μικρότερη ζήτηση σε υπολογιστική ισχύ, όμως δεν παράγουν ιδιαίτερα καλά αποτελέσματα. Αν πάλι οι εφαρμογές δίνουν κάποιο χρονικό περιθώριο (τα 100 λεπτά σε Sun Sparc του Bräysy αντιστοιχούν σε λιγότερο από 1 λεπτό για έναν μέτριο σημερινό υπολογιστή και αφορούν τη λύση 56 προβλημάτων 100 πελατών!) τότε μπορούν να χρησιμοποιηθούν μεταευρετικές μέθοδοι.
Τέλος, για εφαρμογές όπου η ζήτηση είναι υπερβάλλουσα του διαθέσιμου φορτίου, είναι δυνατόν ο χρήστης να διαλέξει ποιες περιοχές θεωρεί σημαντικότερες προς λύση, χρησιμοποιώντας μεθόδους *interactive* όπως αυτή του hill climbing (βλ. σελ. 29).

ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ

Σκοπός

Σκοπός του πειραματικού μέρους ήταν η δημιουργία ενός αλγορίθμου insertion (ο οποίος ονομάστηκε L0), που θα παρήγαγε αποτελέσματα συγκρίσιμα με αυτά των ευρετικών της βιβλιογραφίας (Πίνακας 2: Ευρετικοί κατασκευαστικοί αλγόριθμοι). Η γλώσσα που χρησιμοποιήθηκε για όλους τους κώδικες ήταν η Fortran 95. Αναφέρεται πως η Fortran και η C συμπεριφέρονται 5 -10 φορές πιο γρήγορα από την Java[38].

Ο αλγόριθμός I1

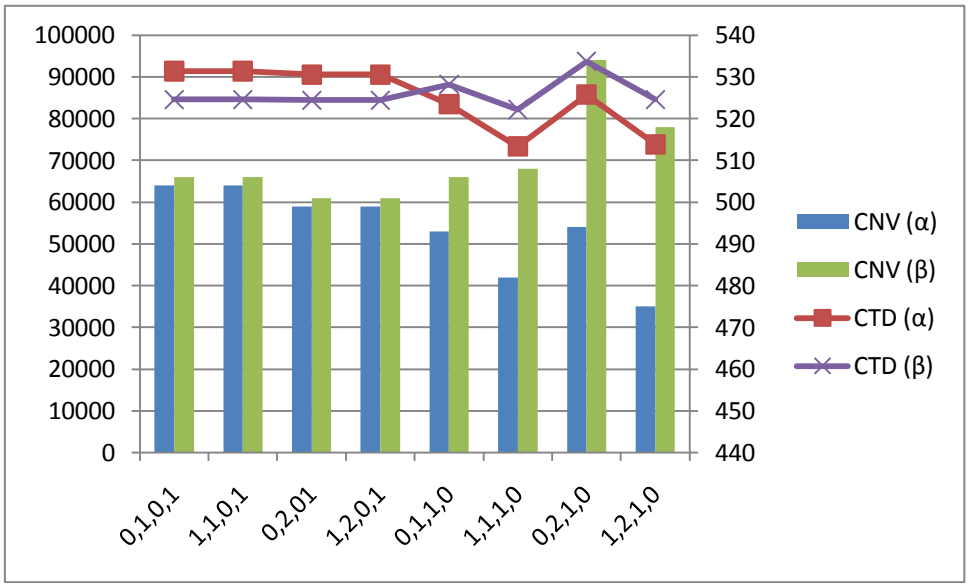
Για λόγους σύγκρισης, επιλέχθηκε ο πλέον ικανός ευρετικός κατασκευαστικός I1 του Solomon, λόγω συγγένειας με τον L0. Ο I1 θεωρείται αποτελεσματικός και χρησιμοποιείται ακόμα και σε μεταευρετικούς για λόγους εντατικοποίησης. Αυτόνομος απαιτεί λίγο χρόνο προκειμένου να παράξει ικανοποιητικής ποιότητας αποτελέσματα. Τέλος όντας ντετερμινιστικός είναι επαναλήψιμος. Αυτοί ήταν οι κύριοι λόγοι που στην επιλογή του I1 σαν μέτρο σύγκρισης.

Κατά την έρευνα περί του I1, βρέθηκε πως τα δημοσιευμένα αποτελέσματα είχαν δύο προβλήματα. *Πρώτον* δεν ήταν αποτέλεσμα ενός «τρεξίματος». Χρησιμοποίησε δηλαδή διαφορετικές τιμές για τις μεταβλητές του μ,λ,α1,α2 και 2 διαφορετικά κριτήρια για την εύρεση του πρώτου προς δρομολόγηση πελάτη. Η μεταβλητή «μ» αναφέρεται στο πόσο σημαντική είναι η προστιθέμενη απόσταση, η «λ» πόσο σημαντική είναι η απόσταση του νέου πελάτη από την αποθήκη ενώ οι «α1»,»α2» πόση βαρύτητα δίνεται στην προστιθέμενη απόσταση σε σχέση με τον προστιθέμενο χρόνο. Οι συγγραφείς χρησιμοποίησαν τις εξής τιμές για τις μεταβλητές απόφασης (1,1,1,0), (1,2,1,0), (1,1,0,1) και (1,2,0,1). Τα κριτήρια επιλογής του πρώτου προς δρομολόγηση πελάτη είναι α) ο πιο απομακρυσμένος και β) ο πελάτης του οποίου το χρονικό παράθυρο ανοίγει πρώτο. Από τις λύσεις που παρήχθησαν, επιλέχθηκαν οι καλύτερες για να τον εκπροσωπήσουν. Δεύτερον, τα αποτελέσματα του αλγορίθμου δεν παράγονται μόνο από τον κατασκευαστικό I1 αλλά βελτιώνονται και από τον 2-opt.

Οι παραπάνω λόγοι οδήγησαν στην αναπαραγωγή του I1. Μελετώντας λοιπόν τη δημοσίευσή του, προκειμένου να γίνει σύγκριση των δύο κατασκευαστικών μεθόδων (του I1 και του δικού μου), παρήχθησαν τα αποτελέσματα που φαίνονται στην Εικόνα 14. Ο υπολογιστικός χρόνος ανά λύση (για κάθε συνδυασμό) για όλους τους αλγόριθμους που ακολουθούν υπολογίστηκε για επεξεργαστή Sun Sparc 10 χρησιμοποιώντας τους συντελεστές Dongarra 2009 για PentiumIII @ 750MHz ενώ οι αλγόριθμοι έτρεξαν σε PentiumIII @ 747MHz.

Ο I1 παρουσιάζει δύο ιδιαιτερότητες. *Πρώτον:* εξετάζει κάθε πελάτη και κάθε δυνατή θέση εισχώρησης και επιλέγεται ο πελάτης με το μικρότερο κόστος εισχώρησης. Αυτή η μέθοδος όπως θα φανεί αργότερα κοστίζει σε υπολογιστικό χρόνο. *Δεύτερον:* προκειμένου να κερδίσει χρόνο, αντί να εξετάζει ολόκληρη τη διαδρομή για το κατά πόσο είναι εφικτή η λύση, χρησιμοποιεί τη μέθοδο PF (push forward) με την οποία υπολογίζεται πόση είναι η μετάθεση χρόνου που λαμβάνει κάθε δρομολογημένος πελάτης, αν πραγματοποιηθεί η εισχώρηση του νέου πελάτη.

Για λόγους πληρότητας, εκτός από τις τιμές που δημοσιεύτηκαν, δοκιμάστηκαν και άλλες τιμές. Φαίνεται πως η καλύτερη των περιπτώσεων είναι ο συνδυασμός (1,2,1,0) για το κριτήριο επιλογής του πρώτου πελάτη «η μεγαλύτερη απόσταση από την αποθήκη» (μπλε μπάρες). Σε περίπτωση που επιλεχθεί το κριτήριο, «ο πελάτης που ανοίγει πρώτος», ο καλύτερος συνδυασμός των μεταβλητών απόφασης είναι (0,2,0,1) και ο (1,2,0,1) (πράσινες μπάρες).



Εικόνα 14: σύγκριση αποτελεσμάτων CTD και CNV του I1, για διαφορετικές τιμές μ,λ,α1,α2, χωρίς βελτιωτική μέθοδο. Κριτήριο πρώτου πελάτη: (α) ο πιο απομακρυσμένος, (β) ο πρώτος που ανοίγει.

Ο χώρος μνήμης που δεσμεύτηκε κατά την εκτέλεση του προγράμματος ήταν πολύ μικρός 4.580KB. Ο Υπολογιστικός χρόνος ήταν 210 δευτερόλεπτα ανά λύση (μέσος όρος 4 επαναλήψεων) δηλαδή σχεδόν 3,5 λεπτά για επεξεργαστή Sun Sparc 10.

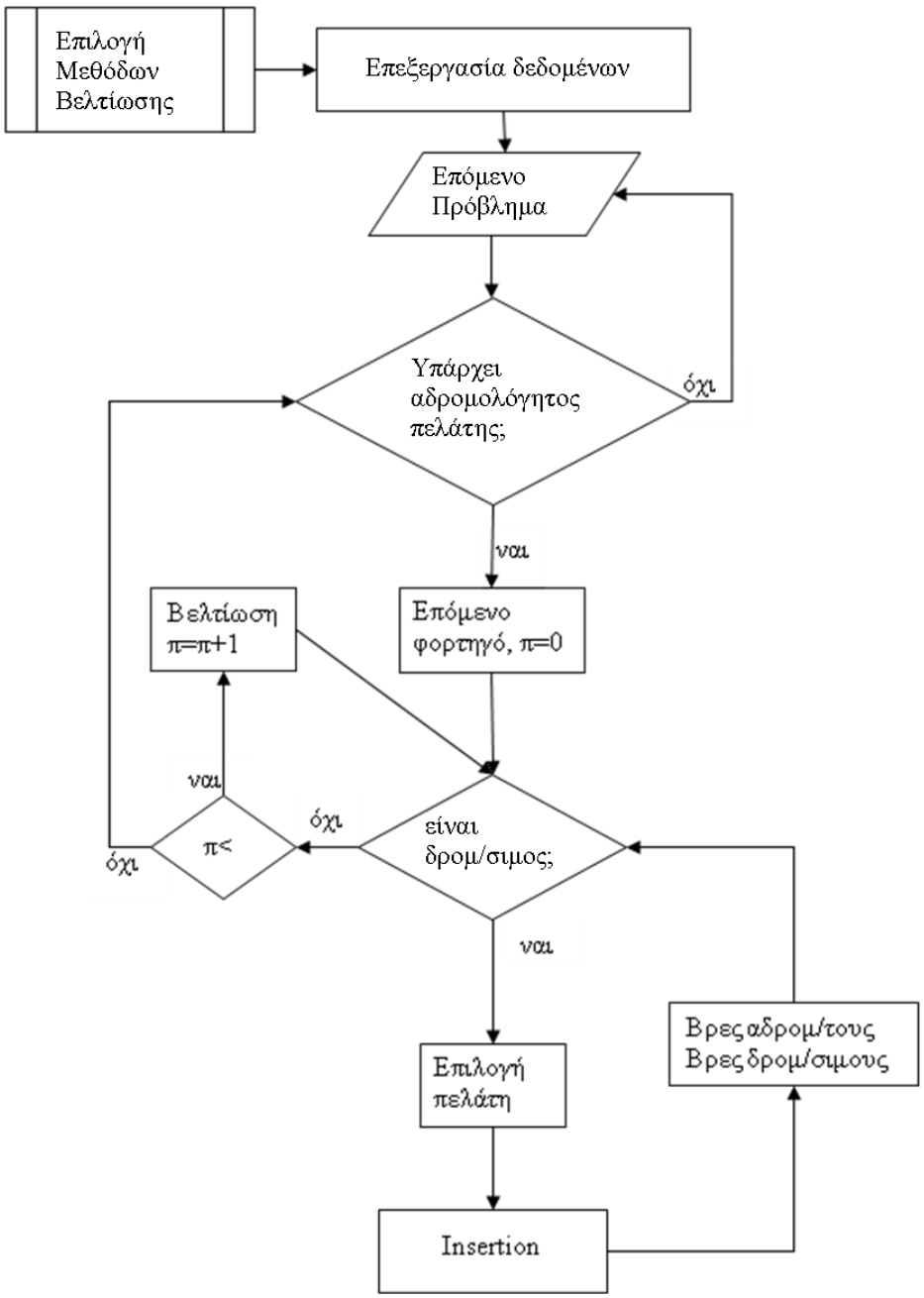
L0 insertion

Για λόγους εξοικείωσης με το πρόβλημα και με τις εντολές που θα χρησιμοποιούσα για πρώτη φορά, αρχικά δημιουργήθηκαν άλλοι απλούστεροι αλγόριθμοι πλεονεκτικού τύπου[xi]. Στη συνέχεια δημιουργήθηκε ένας αλγόριθμος insertion (L0) ο οποίος και συγκρίνεται με τον I1. Όλοι οι αλγόριθμοι που χρησιμοποιήθηκαν είναι διαδοχικής φύσεως. Καθώς τα αποτελέσματα δεν ήταν αρκετά καλά, χρησιμοποιήθηκαν 3 χειριστές οι οποίοι βελτιώνουν τις παραγόμενες λύσεις. Οι χειριστές εφαρμόστηκαν στον (L0) μόνο, και στη συνέχεια συγκρίθηκαν με τον βελτιωμένο I1 της βιβλιογραφίας.

Βασική πορεία του αλγορίθμου

Κατά την εκτέλεση του προγράμματος ο αλγόριθμος ζητάει από τον χρήστη πόσες φορές θα χρησιμοποιηθούν οι χειριστές για το τμήμα βελτίωσης: Στη συνέχεια υπολογίζει τις αποστάσεις μεταξύ πελατών και αρχίζει να λύνει τα προβλήματα όπως φαίνεται στην Εικόνα 15.

[xi] Τα αποτελέσματα αυτών των αλγορίθμων δεν αναφέρονται γιατί υπήρχαν προγραμματιστικά λάθη.



Εικόνα 15: διάγραμμα ροής L0

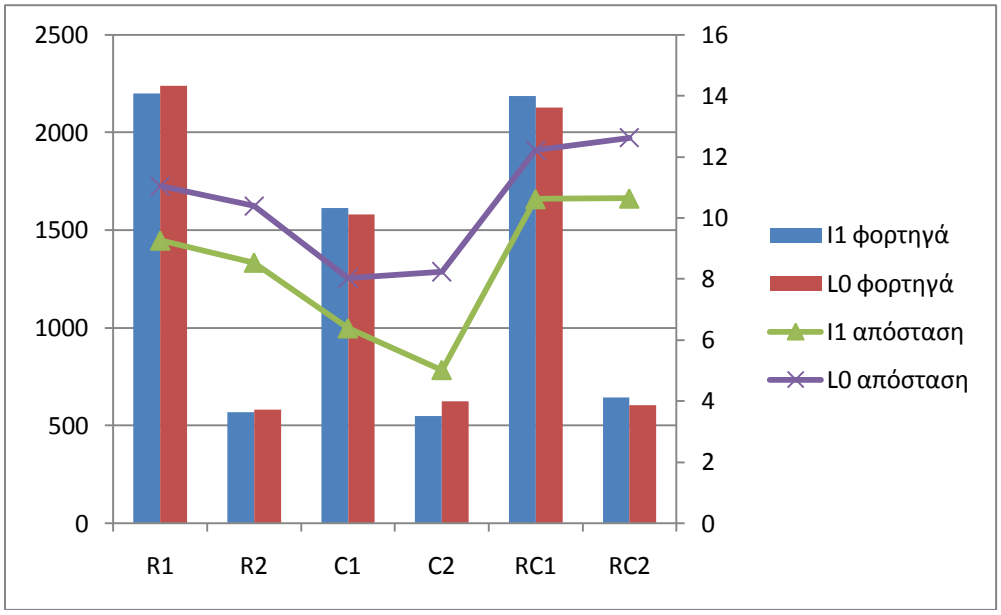
Καθώς είναι διαδοχικής φύσεως, δρομολογεί τους πελάτες έναν-έναν σε ένα δρομολόγιο τη φορά, έως ότου να μην γίνεται να δρομολογηθούν άλλοι πελάτες. Στη συνέχεια, αν έχει ζητηθεί από τον χρήστη, τα αποτελέσματα βελτιώνονται. Υπάρχουν δύο σημαντικές διαφορές σε σύγκριση με τον I1. Η πρώτη είναι το κριτήριο επιλογής του επόμενου προς δρομολόγηση πελάτη. Αντί να εξετάζονται όλες οι θέσεις για όλους τους υποψήφιους πελάτες, αναζητείται απλά ο διαθέσιμος πελάτης που είναι πιο εγγύς στο κέντρο βάρους

της τρέχουσας διαδρομής και δεν έχει απαγορευτική ζήτηση. Σαν *εγγύτητα* εδώ θεωρείται το άθροισμα της ανηγμένης απόστασης από το κέντρο βάρους προς τη μέγιστη απόσταση από την αποθήκη με τον ανηγμένο χρόνο λήξης του χρονικού παραθύρου προς τον αργότερο χρόνο λήξης των χρονικών παραθύρων.

Cost(k)=α*dist(k)/maxdist+(1-α)*LS(k)/maxLS(k).

Στη συνέχεια εξετάζεται ποια είναι η καλύτερη θέση για να δρομολογηθεί ο επιλεγμένος πελάτης. Το κόστος προκύπτει από το άθροισμα της προστιθέμενης απόστασης με τον χρόνο αναμονής που θα είχε το φορτηγό σε αυτόν τον πελάτη. Αν δεν βρίσκεται κάποια δυνατή θέση για να δρομολογηθεί ο υπό εξέταση πελάτης, βγαίνει από τη λίστα των δρομολογήσιμων πελατών για την τρέχουσα διαδρομή. Αρχικά το κέντρο βάρους είναι οι συντεταγμένες της αποθήκης. Στη συνέχεια προσαρμόζεται μόνο στους δρομολογημένους πελάτες. Το κέντρο βάρους ενημερώνεται κάθε φορά που προστίθεται (ή αφαιρείται) ένας πελάτης. Το α είναι μία μεταβλητή βαρύτητας. Όταν α=0,5 δίνεται η ίδια βαρύτητα σε χρόνο και απόσταση. Εμπειρικά δίνεται ένα μικρό προβάδισμα στην απόσταση και το α γίνεται 0,54.

Αποτελέσματα κατασκευαστικής μεθόδου



Εικόνα 16: Σύγκριση I1 με L0 χωρίς βελτιωτικές μεθόδους

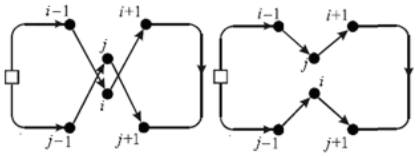
Η σύγκριση των μεθόδων ως προς τον αριθμό φορτηγών ανά ομάδα δεν δείχνει υπεροχή σε κάποια από τις δύο μεθόδους. Ο I1 φαίνεται καλύτερος στις ομάδες R1, R2 και C2 ενώ ο L0 φαίνεται καλύτερος στις ομάδες C1, RC1 και RC2. Σε θέμα όμως διανυθείσας απόστασης υπάρχει σίγουρα υπεροχή της I1. Συνολικά τα φορτηγά που χρησιμοποιήθηκαν στον L0 ήταν 476 έναντι 475 για τον I1. Η ολικά διανυθείσα απόσταση για τον L0 ήταν 91.205 ενώ για τον I1 73.880. Η διαφορά εδώ είναι αισθητή. Τέλος σε θέμα υπολογιστικού χρόνου, ο L0 χρειάστηκε μόλις 131 δευτερόλεπτα ή περίπου 2,2 λεπτά δηλαδή ήταν 1,6 φορές πιο γρήγορος από τον I1. Αυτό οφείλεται στο ότι δεν εξετάζει όλες τις πιθανές θέσεις κάθε πελάτη. Επίσης επειδή δεν χρησιμοποιείται η τεχνική PF, πιθανόν ο L0 να ήταν ακόμα πιο γρήγορος.

Βελτιωτικές μέθοδοι

Τα αποτελέσματα από την κατασκευαστική μέθοδο χρειάζονται βελτίωση προκειμένου να ανταγωνιστούν τις άλλες ευρετικές μεθόδους. Για τον λόγο αυτόν κατασκευάστηκαν εσωτερικοί χειριστές βελτίωσης οι οποίοι ενεργοποιούνται όταν δεν είναι δυνατόν να βρεθεί άλλος δρομολογήσιμος πελάτης, υπάρχουν μερικοί ακόμα διαθέσιμοι προς δρομολόγηση και οι δρομολογημένοι στην τρέχουσα διαδρομή είναι πάνω από 3.

E-opt

Ο πρώτος χειριστής, στη βιβλιογραφία ονομάζεται και 2-ανταλλακτικός αλλά για λόγους συντομίας και διαφορετικών κριτηρίων που χρησιμοποιούνται θα ονομάζεται στο εξής E-opt (Exchange-operator) (Εικόνα 17). Ο E-opt εξετάζει με τη σειρά όλα τα ζευγάρια πελατών μίας διαδρομής και σε περίπτωση που μία εφικτή ανταλλαγή μειώνει *το κόστος δρομολόγησης* (το άθροισμα της συνολικής αναμονής με τη συνολική διανυθείσα απόσταση της διαδρομής) πραγματοποιείται η ανταλλαγή. Οι ανταλλαγές πραγματοποιούνται «*E*» φορές κάθε φορά που τρέχει το βελτιωτικό τμήμα. Στη συνέχεια όλοι οι αδρομολόγητοι πελάτες που είχαν βγει από τη λίστα «δρομολογήσιμοι» επανέρχονται και επανεξετάζονται με τον βασικό αλγόριθμο L0 για να δρομολογηθούν.



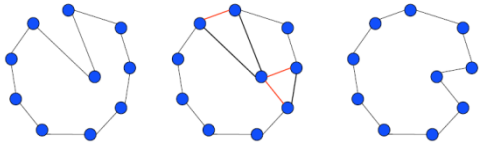
Εικόνα 17: E-opt

2-opt

Αν και συνήθως χρησιμοποιείται με τη μέθοδο BA (σελ 16), εδώ για λόγους ευκολίας και ομοιογένειας ως προς τη σύγκριση με τους άλλους χειριστές, χρησιμοποιείται με τη μέθοδο FA. Μία εφικτή ανταλλαγή πραγματοποιείται όταν ελαττώνεται το κόστος δρομολόγησης. Ο χειριστής εκτελείται «*dyo*» φορές.

1-relocate + D-opt

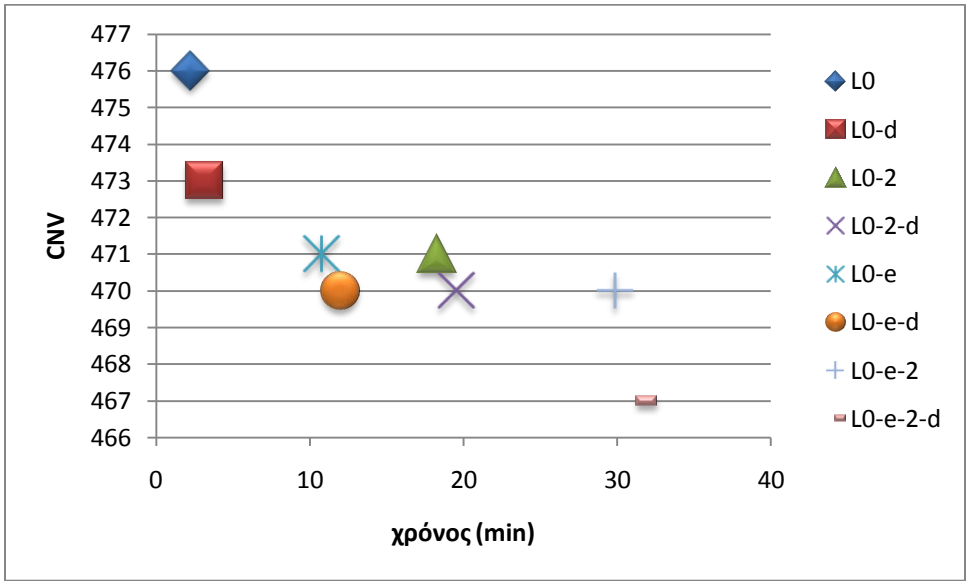
Πρόκειται για μία παραλλαγή του απλού χειριστή 1-ανατοποθέτησης (Εικόνα 18). Αρχικά υπολογίζεται το πόσο προβληματικός είναι ο κάθε δρομολογημένος πελάτης με συνάρτηση κόστους ***Costos(m)= 3*wait(v,m)+apostasi(i,j),*** όπου m η θέση δρομολόγησης, wait(v,m) η αναμονή της θέσης m στο δρομολόγιο v, και apostasι(i,j) η απόσταση του υπό εξέταση πελάτη από τον προηγούμενο από αυτόν στο δρομολόγιο. Αν υπάρχει πελάτης που είναι πάνω από 2 φορές πιο προβληματικός από τον μέσο όρο, αφαιρείται από το δρομολόγιο και ενεργοποιείται ο L0 για να τον ξαναδρομολογήσει. Ο χειριστής εκτελείται «D» φορές, ενώ την τελευταία φορά του δίνεται η δυνατότητα να αντικαταστήσει τον προβληματικό πελάτη με άλλον αδρομολόγητο. Ένας πελάτης όσο προβληματικός και να είναι δεν μπορεί να αφαιρεθεί δύο συνεχόμενες φορές.



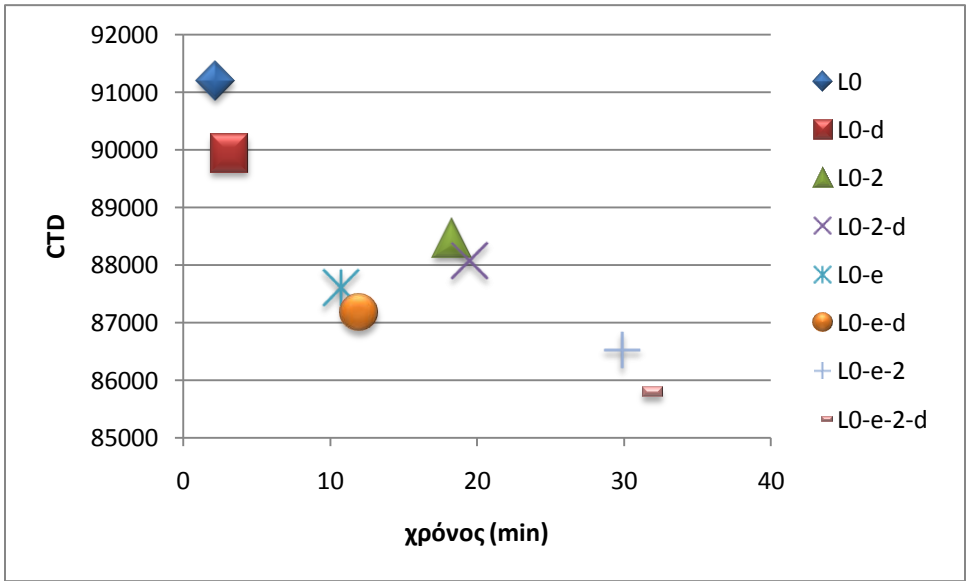
Εικόνα 18:1-ανατοποθέτηση (1-relocate)

Αποτελέσματα βελτιώσεων L0

Στην Εικόνα 19 φαίνεται η επίδραση των προαναφερθέντων βελτιωτικών μεθόδων στον CNV σε όλους τους πιθανούς συνδυασμούς για 1*λ φορές, όπου ο πρώτος αριθμός φανερώνει τις συνολικές εναλλαγές ενώ ο δεύτερος τις επαναλήψεις ανά μέθοδο. Με λ θα συμβολίζεται ο αριθμός των δρομολογημένων πελατών στην τρέχουσα διαδρομή. Όπως φαίνεται ο d-opt είναι φανερά πιο γρήγορος από τους άλλους δύο, ενώ ακολουθεί ο e-opt και στη συνέχεια ο 2-opt. Σε αποτελεσματικότητα ο d-opt φαίνεται να υστερεί αν και πολύ ταχύτερος. Στη συνέχεια, όσον αφορά τον CNV ο e-opt έχει ίδια αποτελέσματα με αυτά του 2-opt σε μικρότερο χρόνο, ενώ στη CTD είναι κατά 1000 μονάδες καλύτερος (Εικόνα 20). Τέλος ο συνδυασμός και των τριών μεθόδων φέρνει πολύ ενθαρρυντικά αποτελέσματα, όμως φαίνεται πως καθυστερούν σχετικά να παραχθούν.

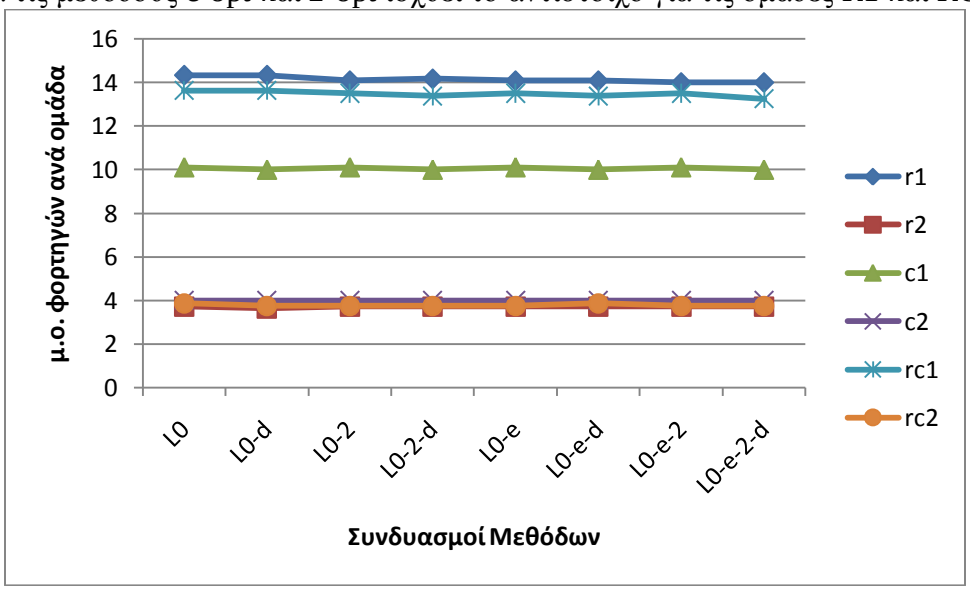


Εικόνα 19: Συνδυασμοί βελτιωτικών μεθόδων (CNV-χρόνος)



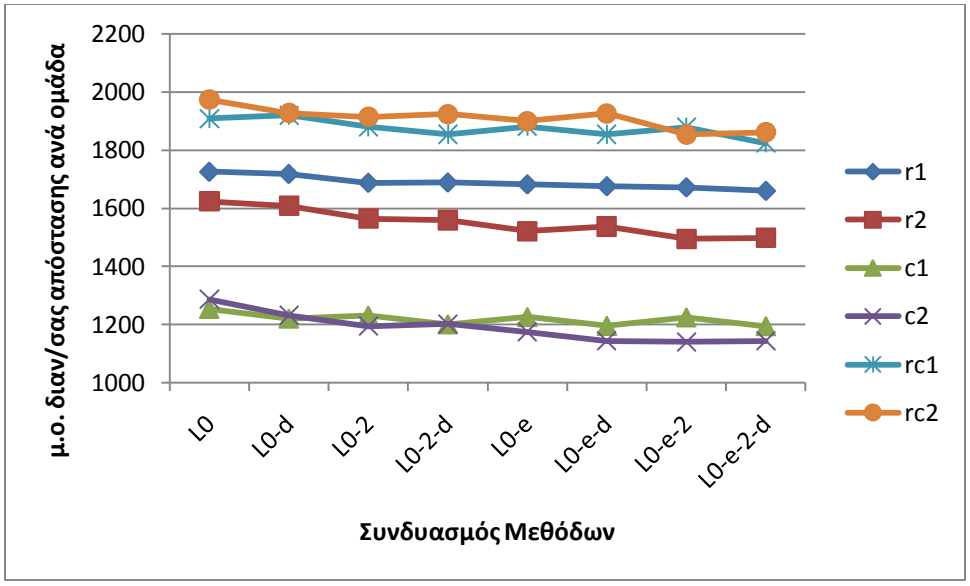
Εικόνα 20: Συνδυασμοί βελτιωτικών μεθόδων (CTD-χρόνος)

Στις επόμενες εικόνες φαίνεται πως η επίδραση του D-opt στα προβλήματα των ομάδων R1 και C1 είναι θετική όσον αφορά τον μέσο όρο των χρησιμοποιούμενων φορτηγών, ενώ για τις μεθόδους e-opt και 2-opt ισχύει το αντίστοιχο για τις ομάδες R1 και RC1.



Εικόνα 21: Επίδραση μεθόδων ανά ομάδα προβλημάτων σε αριθμό φορτηγών

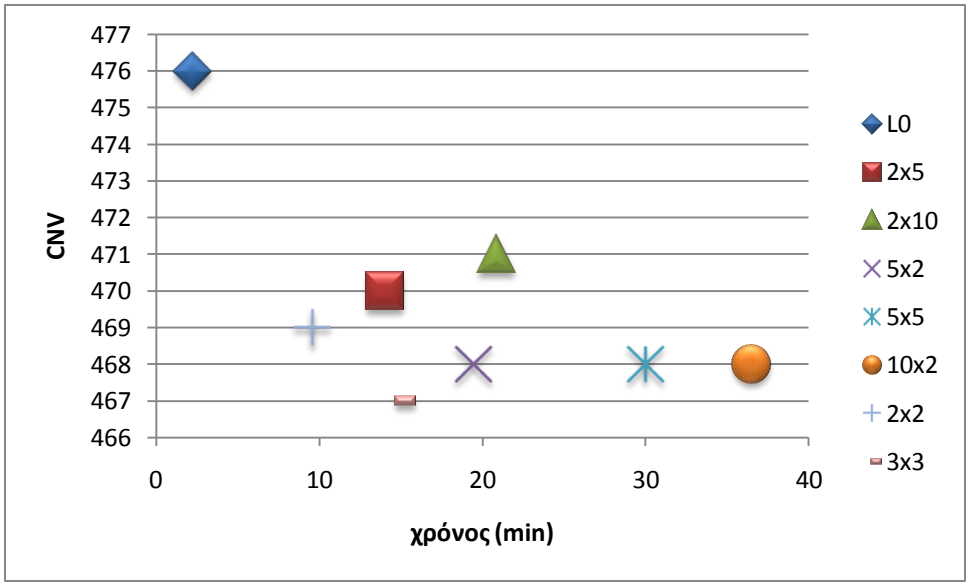
Όσον αφορά την μέση διανυθείσα απόσταση ανά ομάδα προβλήματος, είναι φανερό πως όσο πιο πολλές βελτιωτικές μέθοδοι χρησιμοποιούνται τόσο πιο καλά είναι τα παραγόμενα αποτελέσματα.



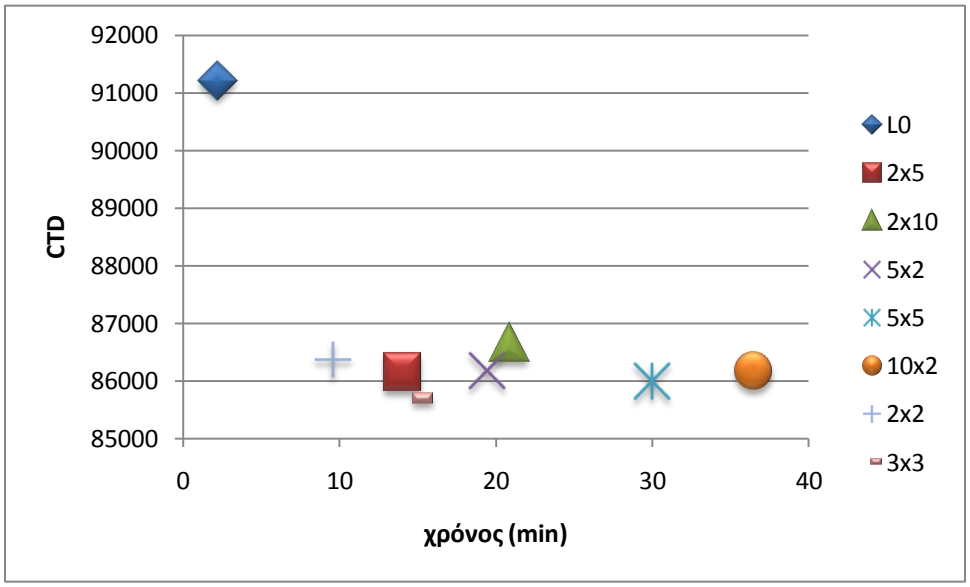
Εικόνα 22: Επίδραση μεθόδων ανά ομάδα προβλημάτων σε απόσταση

Όπως φαίνεται από τους παραπάνω συνδυασμούς μεθόδων, ο 2-opt και ο e-opt καθυστερούν αρκετά να βελτιώσουν τη λύση, γι' αυτόν το λόγο στη συνέχεια θα χρησιμοποιηθούν νέοι συνδυασμοί.

Δοκιμάστηκαν διάφοροι συνδυασμοί και από τα αποτελέσματα (Εικόνα 23 και Εικόνα 24) φαίνεται πως οι καλύτεροι είναι ο (2x2) και ο (3x3). Ο δεύτερος μάλιστα έχει αποτελέσματα αντίστοιχα με τον (L0 e-2-d) δηλαδή του (1xλ) σε χρόνο μικρότερο από 14,8 min για έναν Sun Sparc 10 σε αντίθεση με τον (1xλ) του οποίου ο υπολογιστικός χρόνος αντιστοιχεί σε πάνω από 31 min.



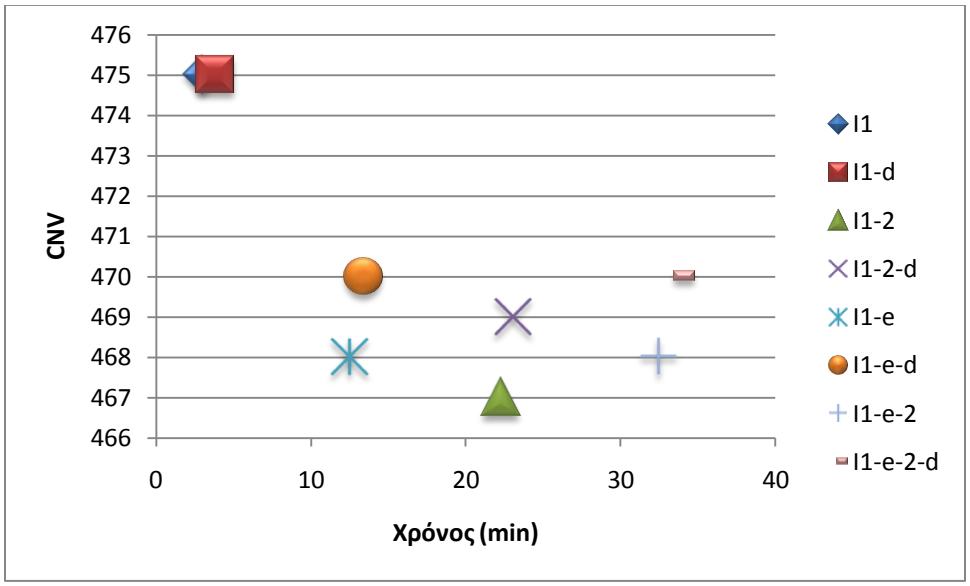
Εικόνα 23: αποτελεσματικότητα άλλων συνδυασμών σε CNV



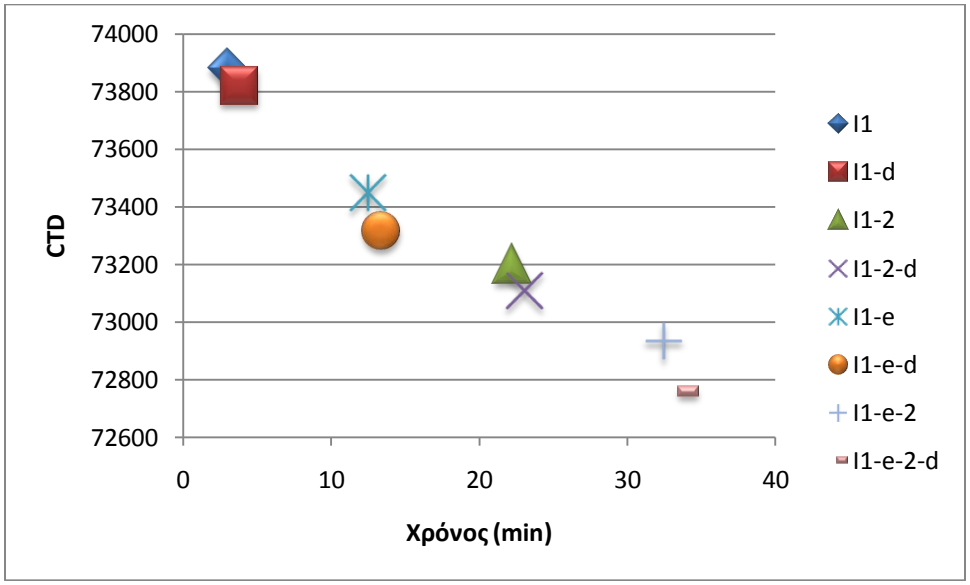
Εικόνα 24: Αποτελεσματικότητα άλλων συνδυασμών σε CTD

Αποτελεσματικότητα Βελτιώσεων I1

Ακολουθώντας την ίδια πειραματική διαδικασία που ακολουθήθηκε για τον L0 προκύπτουν τα παρακάτω δεδομένα. Όπως φαίνεται (Εικόνα 25 και Εικόνα 26) ο χειριστής D-opt εδώ δεν είναι ιδιαίτερα αποτελεσματικός. Αντίθετα ο E-opt είναι πολύ αποτελεσματικός ενώ ο 2-opt επιτυγχάνει τα καλύτερα αποτελέσματα μόνος του αντί σε συνδυασμό με άλλους χειριστές. Ίσως λοιπόν να μην είναι τυχαίο που ο Solomon χρησιμοποίησε μόνο τον 2-opt για να βελτιώσει τα αποτελέσματά του. Ο λόγος που υπάρχει ασυμφωνία με τα δημοσιευμένα αποτελέσματα προφανώς σχετίζεται με το γεγονός ότι ο Solomon συνδύασε τις καλύτερες από τις 8 παραχθήσες λύσεις, για διαφορετικές τιμές μ,λ,α1,α2.



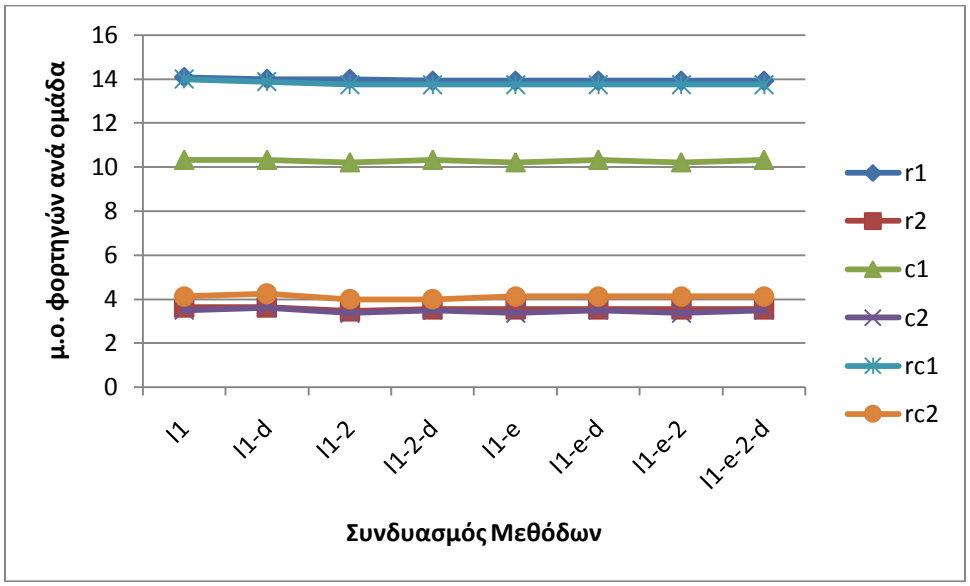
Εικόνα 25: Συνδυασμοί βελτιωτικών μεθόδων (CNV-χρόνος)



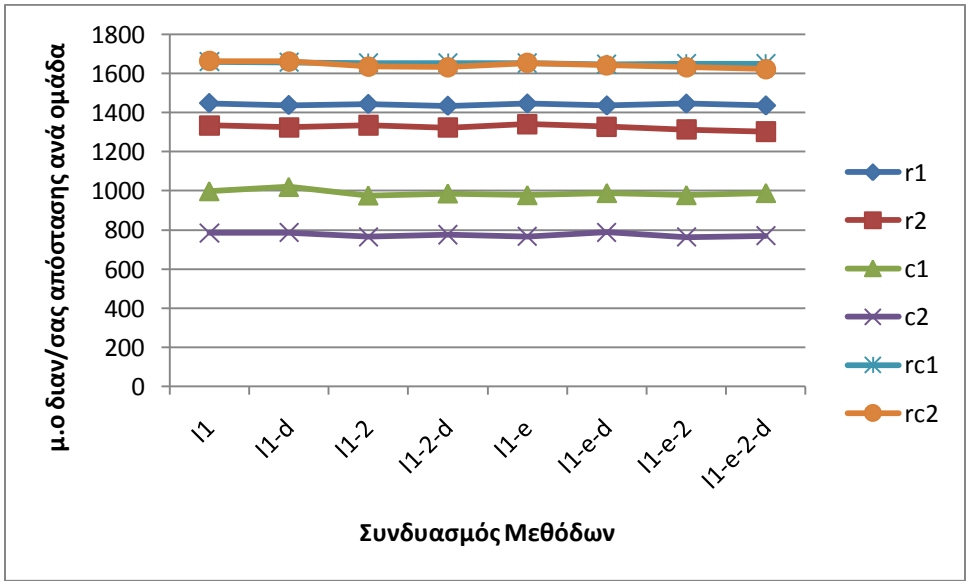
Εικόνα 26: Συνδυασμοί βελτιωτικών μεθόδων (CTD-χρόνος)

Ιδιαίτερη εντύπωση προκαλεί η Εικόνα 26 καθώς η CTD φαίνεται να μειώνεται γραμμικά με τον υπολογιστικό χρόνο.

Όπως φάνηκε και στις προηγούμενες εικόνες, ο 2-opt συμπεριφέρεται πολύ καλύτερα από τους άλλους χειριστές και ιδιαίτερα όταν είναι μόνος του και ειδικά όταν πρόκειται για τις ομάδες R2,C1,C2, RC2 (Εικόνα 27). Αντίστοιχα καλή συμπεριφορά δείχνει και ο E-opt.



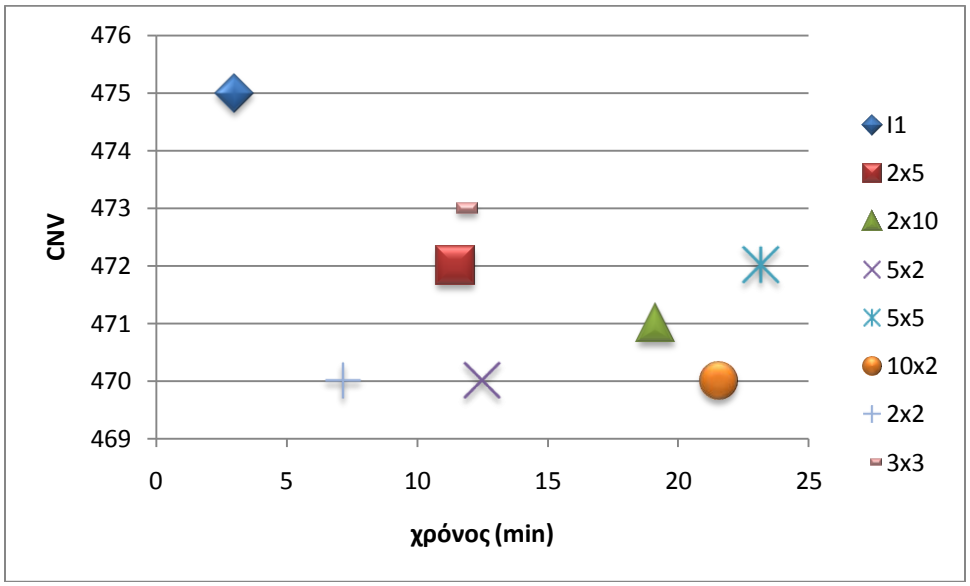
Εικόνα 27: Επίδραση μεθόδων ανά ομάδα προβλημάτων σε αριθμό φορτηγών



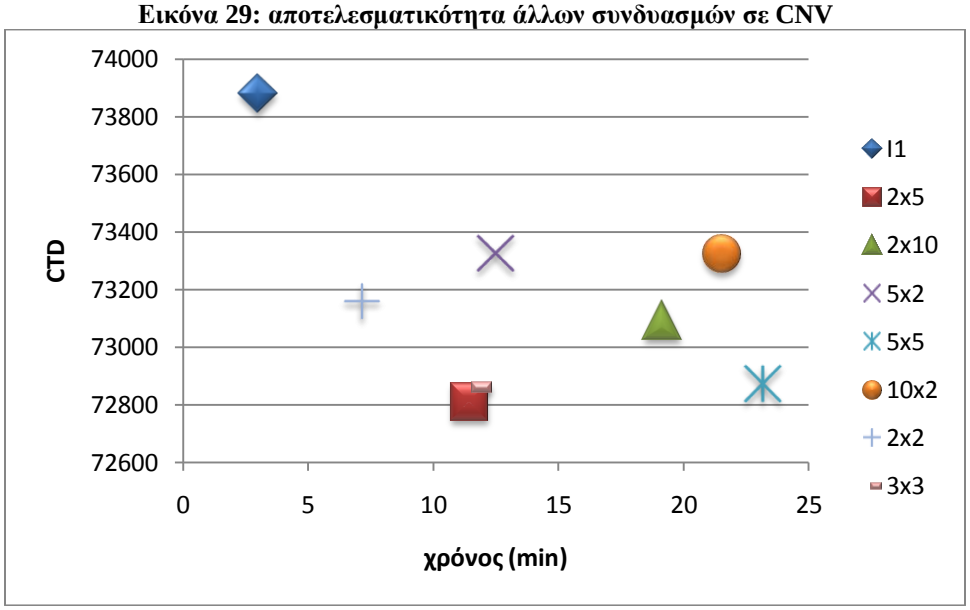
Εικόνα 28: Επίδραση μεθόδων ανά ομάδα προβλημάτων σε απόσταση

Όσο επενδύεται χρόνος σε ρυθμιστές μειώνεται και η διανυθείσα απόσταση. Η ελάχιστη διανυθείσα απόσταση παρατηρείται στο συνδυασμό των μεθόδων 2-opt - d-opt, ενώ οι διαφορές ανά ομάδα είναι μικρές.

Αντίθετα με το τι ισχύει για τον L0 ο I1 δείχνει τη βέλτιστη συμπεριφορά του μόνο κατά τη χρήση του 2-opt. Στις συνθήκες που εξετάζονται εδώ, η καλύτερη λύση είναι η 2x2 η οποία και παράγεται ταχύτατα. Γενικά ισχύει πως ο I1 παρουσιάζει σε αυτού του είδους τους συνδυασμούς μικρότερο υπολογιστικό χρόνο από τον L0 αλλά με μικρότερη ποιότητα.



Εικόνα 29: αποτελεσματικότητα άλλων συνδυασμών σε CNV



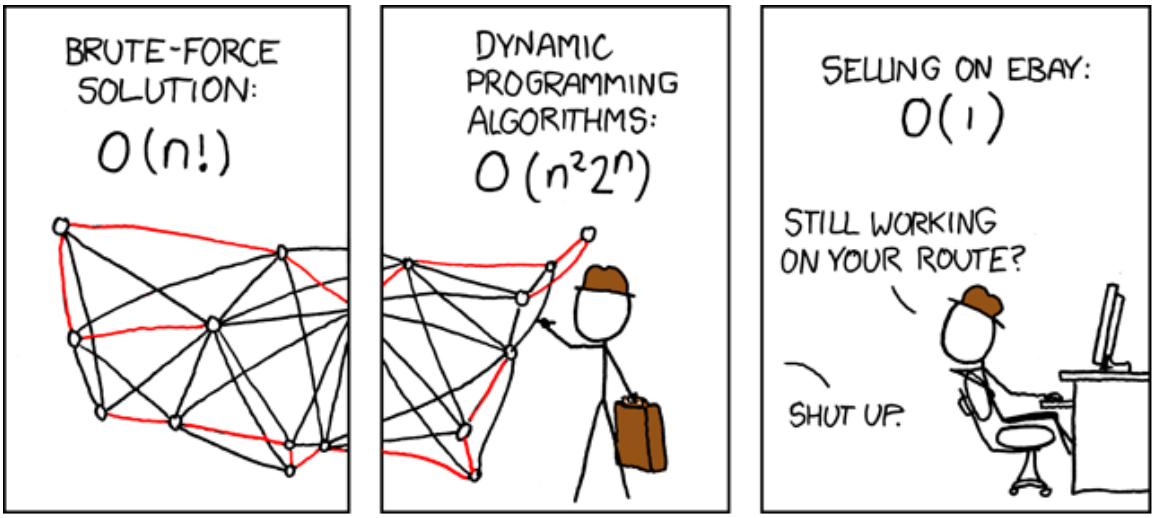
Εικόνα 30: αποτελεσματικότητα άλλων συνδυασμών σε CTD

Ο συνδυασμός 2x2 συμπεριφέρεται καλά και στη διανυθείσα απόσταση, καλύτερα όμως συμπεριφέρεται ο 2x5. Αυτό φαίνεται λογικό, καθώς χρησιμοποιούνται πιο εντατικά οι χειριστές 2-opt και e-opt που σκοπό έχουν να ελαττώσουν την απόσταση. Σκόπιμα ο D-opt σε όλες τις περιπτώσεις χρησιμοποιείται τελευταίος. Ο λόγος είναι για να απομακρύνει τους πελάτες οι οποίοι πραγματικά είναι προβληματικοί. Υπάρχει δηλαδή κίνδυνος διαφορετικά να αφαιρέσει έναν πελάτη, ο οποίος απλά δεν δρομολογήθηκε στη σωστή θέση αρχικά, αλλά θα μπορούσε να βρεθεί σε καλύτερη.

Συμπεράσματα

Αρχικά η κατασκευαστική μέθοδος που δημιουργήθηκε L0 υστερούσε ως προς την ολική διανυθείσα απόσταση κατά πολύ σε σχέση με τον I1. Ο CNV όμως ήταν σχεδόν ίσος για τις δύο μεθόδους πριν τη βελτίωση. Τα πράγματα δεν άλλαξαν πολύ μετά την εφαρμογή των βελτιώσων. Χρησιμοποιήθηκε και για τους δύο αλγορίθμους μόνο μία λύση προς βελτίωση, η καλύτερη που παράγεται κατά το κατασκευαστικό τους μέρος. Στη συνέχεια οι βελτιωμένες λύσεις που παρήχθησαν ήταν παρόμοιας ποιότητας, ενώ ο υπολογιστικός χρόνος ήταν πολλαπλάσιος. Ο L0 συμπεριφέρθηκε αναμενόμενα όσον αφορά τον CNV ενώ ο I1 όσον αφορά τη CTD. Ο 2-opt καταναλώνει πολύ περισσότερο υπολογιστικό χρόνο από ότι οι άλλοι 2 χειριστές. Για κάποιον άγνωστο λόγο ο 2-opt λειτουργεί καλύτερα στον I1. Αρχικά ο υπολογιστικός χρόνος για το κατασκευαστικό μέρος του I1 ήταν μεγαλύτερος από τον αντίστοιχο για τον L0 αλλά σε μερικές περιπτώσεις βελτιώσεων, γίνεται ταχύτερος, καθώς δεν υπολογίζει από την αρχή όλο το δρομολόγιο όπως ο L0 αλλά εντοπίζει μόνο τις αλλαγές που δημιουργούνται κατά το βελτιωτικό μέρος με τη μέθοδο PF (Push Forward). Ο L0 θα ήταν πιο γρήγορος αν χρησιμοποιούσε τη μέθοδο PF και θα μπορούσε να βρει εφαρμογές σε πολύ απαιτητικά αντικείμενα όσον αφορά τον υπολογιστικό χρόνο.

Η καλύτερη λύση που παράγεται με τις βελτιωτικές μεθόδους (467 CNV) για τον L0 είναι ταχύτερη από αυτήν του I1 και γι'αυτόν το λόγο, αν και διανύει μεγαλύτερη απόσταση, θεωρείται καλύτερη λύση και pareto optimal μεταξύ των δύο.



Βιβλιογραφία

[1] G. B. Dantzig and J. H. Ramser 1959. The Truck Dispatching Problem INFORMS.

[2] http://en.wikipedia.org/wiki/VRP

[3] http://www2.imm.dtu.dk/~jla/solomon.html

[4] Bräysy O. 2001. Local search and variable neighborhood search algorithms for the vehicle routing problem with time windows. Doctoral dissertation, University of Vaasa, Finland.

[5] Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *per.Res.* 35 254–265.

[6] Christofides, N., J. Beasley. 1984. The period routing problem. *Networks* 14 237–246.

[7] Clarke, G., J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Oper.Res.* 12 568–581.

[8] Dullaert, W. 2000. Impact of relative route length on the choice of time insertion criteria for insertion heuristics for the vehiclerouting problem with time windows. B. Maurizio, ed. *Proc. Rome Jubilee 2000 Conf.Impr oving Knowledge Tools Transportation Logist.*, Faculty of Engineering, University of Rome, Italy, 153–156.

[9] Dullaert, W. and O. Bräysy. 2003. Routing with relatively few customers per route. *TOP* 11 325–336.

[10] Potvin, J.-Y., J.-M. Rousseau. 1993. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eur.J.Oper .Res.* 66 331–340.

[11] Foisy, C., J.-Y. Potvin. 1993. Implementing an insertion heuristic for vehicle routing on parallel hardware. *Comput.Oper .Res.* 20 737–745.

[12] Ioannou, G., M. Kritikos, G. Prastacos. 2001. A greedy look-ahead heuristic for the vehicle routing problem with time windows. *J.Oper .Res.Soc.* 52 523–537.

[13] Gillett, B., L. R. Miller. 1974. A heuristic algorithm for the vehicle dispatch problem. *Oper.Res.* 22 340–349.

[14] Russell, R. A. 1995. Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Sci.* 29 156–166.

[15] Lin, S. 1965. Computer solutions of the traveling salesman problem. *Bell System Tech.J.* 44 2245–2269.

[16] Baker, E. K., J. R. Schaffer. 1986. Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints. *Amer.J.Math.Management Sci.* 6 261–300.

[17] Or, I. 1976. Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. Ph.D. thesis, Northwestern University, Evanston, IL.

[18] Potvin, J.-Y., J.-M. Rousseau. 1993. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eur.J.Oper .Res.* 66 331–340.

[19] Savelsbergh, M. W. P. 1992. The vehicle routing problem with time windows: Minimizing route duration. *J.Comput.* 4 146–154.

[20] De Backer B., V. Furnon, P. Prosser, P. Kilby, P. Shaw. 1997. Local search in constraint programming: Application to the vehicle routing problem. *Proc.CP-97 Workshop Indust.Constraint- Directed Scheduling*, Schloss Hagenberg, Austria, 1–15.

[21] Osman, I. H. 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems. *Ann.Oper . Res.* 41 421–452.

[22] Glover, F. 1991. Multilevel tabu search and embedded search neighborhoods for the traveling salesman problem. Working paper, College of Business and Administration, University of Colorado, Boulder, CO.

[23] Thompson, P. M., H. N. Psaraftis. 1993. Cyclic transfer algorithms for multivehicle routing and scheduling problems. *Oper.Res.* 41 935–946.

[24] Gendreau, M., A. Hertz, G. Laporte. 1992. A new insertion and postoptimization procedures for the traveling salesman problem. *Oper.Res.* 40 1086–1093.

[25] Antes, J., U. Derigs. 1995. A new parallel tour construction algorithm for the vehicle routing problem with time windows. Working paper, Department of Economics and Computer Science, University of Kφln, Germany.

[26] Shaw, P. 1997. A new local search algorithm providing high quality solutions to vehicle routing problems. Working paper, Department of Computer Science, University of Strathclyde, Glasgow, Scotland.
[27] Shaw, P. 1998. Using constraint programming and local search methods to solve vehicle routing problems. M. Maher, J.-F. Puget, eds. *Principles and Practice of Constraint Programming— CP98, Lecture Notes in Computer Science*. Springer-Verlag, New York, 417–431.
[28] Schrimpf, G., J. Schneider, H. Stamm-Wilbrandt, G. Dueck. 2000. Record breaking optimization results using the ruin and recreate principle. *J.Comput.Phys.* 159 139–171.
[29] Caseau, Y., F. Laburthe. 1999. Heuristics for large constrained vehicle routing problems. *J.Heuristics* 5 281–303.
[30] Caseau, Y., F. Laburthe. 1997. Solving small TSPs with constraints. L. Naish, ed. *Proc.14th Internat.Conf.Logic Programming*. MI T Press, Cambridge, MA, 316–330.
[31] Hong, S.-C., Y.-B. Park. 1999. A heuristic for bi-objective vehicle routing with time window constraints. *Internat.J.Pr oduction Econom.* 62 249–258.
[32] Cordone, R., R. Wolfler-Calvo. 2001. A heuristic for the vehicle routing problem with time windows. *J.Heuristics* 7 107–129.
[33] Bräysy, O. 2003. Fast local searches for the vehicle routing problem with time windows. *Inform.Systems Oper.Res.* 41 179–194.
[34] Dongarra, J. 1998. Performance of various computers using standard linear equations software. Report CS-89-85, Department of Computer Science, University of Tennessee, Knoxville, TN.
[35] Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* 13 533–549.
[36] Badeau, P., M. Gendreau, F. Guertin, J.-Y. Potvin, E. Taillard. 1997. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Res. C* 5 109–122.
[37] Cordeau, J.-F., G. Laporte, A. Mercier. 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *J. Oper. Res. Soc.* 52 928–936.
[38] Bräysy O. Gendreau M. 2005. Vehicle Routing Problem with Time Windows, Part II: Metaheurestics Transportation Science Vol. 39, No. 1, February 2005, pp. 119-139
[39] Thangiah, S. R., K. E. Nygard, P. L. Juell. 1991. GIDEON: A genetic algorithm system for vehicle routing with time windows. *Proc 7th IEEE Conf. Artificial Intelligence Appl.*, IEEE Computer Society Press, Los Alamitos, 322–328.
[40] Wee Kit, H., J. Chin, A. Lim. 2001. A hybrid search algorithm for the vehicle routing problem with time windows. *Internat. J. Artificial Intelligence Tools* 10 431–449.
[41] Berger, J., M. Barkaoui, O. Bräysy. 2003. A route-directed hybrid genetic approach for the vehicle routing problem with time windows. *Inform. Systems Oper. Res.* 41 179–194.
[42] Kontoravdis, G. A., J. F. Bard. 1995. A GRASP for the vehicle routing problem with time windows. *INFORMS J. Comput.* 7 10–23.
[43] Potvin, J.-Y., C. Robillard. 1995. Clustering for vehicle routing with a competitive neural network. *Neurocomputing* 8 125–139.
[44] Kilby, P., P. Prosser, P. Shaw. 1999. Guided local search for the vehicle routing problem with time windows. S. Voss, S. Martello, I. H. Osman, C. Roucairol, eds. *META-HEURISTICS Advanced Trends Local Search Paradigms for Optimization*, Kluwer Academic Publishers, Boston, MA, 473–486.
[45] Chiang, W. C., R. A. Russell. 1996. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Ann. Oper. Res.* 63 3–27.
[46] Gambardella, L. M., E. Taillard, G. Agazzi. 1999. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. D. Corne, M. Dorigo, F. Glover, eds. *New Ideas in Optimization.* McGraw-Hill, London, UK, 63–76.
[47] Anderson, D., E. Anderson, N. Lesh, J. Marks, B. Mirtich, D. Ratajczak, K. Ryall. 2000. Human-guided simple search. Working paper, Mitsubishi Electric Research Laboratory, Cambridge, MA.
[48] Bräysy, O., G. Hasle, W. Dullaert. 2004b. A multi-start local search algorithm for the vehicle routing problem with time windows. *Eur. J. Oper. Res.* 159 586–605.