



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ II:

ΑΝΑΛΥΣΗΣ, ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΔΙΕΡΓΑΣΙΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ

ΑΝΑΠΤΥΞΗ ΥΒΡΙΔΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ ΓΙΑ ΤΗΝ
ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΗΣ ΧΩΡΟΘΕΤΗΣΗΣ ΠΡΟΪΟΝΤΩΝ ΣΕ
ΑΠΟΘΗΚΕΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΤΟΥ ΣΠΟΥΔΑΣΤΗ

ΣΥΚΩΚΗ ΓΕΩΡΓΙΟΥ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΣΑΡΙΜΒΕΗΣ
ΧΑΡΑΛΑΜΠΟΣ

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ II: ΑΝΑΛΥΣΗΣ, ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΔΙΕΡΓΑΣΙΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ

**Ανάπτυξη Υβριδικού Αλγόριθμου για την Βελτιστοποίηση της
Χωροθέτησης Προϊόντων σε Αποθήκες**

Διπλωματική Εργασία του Σπουδαστή: Συκώκη Γεώργιου

Επιβλέπων Καθηγητής: Σαρίμβεης Χαράλαμπος, Αναπληρωτής Καθηγητής

Συνεπίβλεψη: Νικολακόπουλος Αθανάσιος, Ε.ΔΙ.Π.

Τριμελής Επιτροπή:

Κυρανούδης Χρήστος, Καθηγητής
Μαυρωτάς Γεώργιος, Αναπληρωτής Καθηγητής
Σαρίμβεης Χαράλαμπος, Αναπληρωτής Καθηγητής

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2014

Περιεχόμενα

Λίστα σχημάτων.....	5
Λίστα Πινάκων	5
Περίληψη	7
Abstract.....	9
Εισαγωγή.....	11
1. Μαθηματικός προγραμματισμός.....	16
Εφαρμογές – Χρήση	17
Γενικό Μοντέλο	17
Είδη μοντέλων	17
Μεταβλητές ενός μοντέλου μαθηματικού προγραμματισμού	18
Διεπίπεδος Προγραμματισμός.....	18
Γενικό Μοντέλο Προβλημάτων Διεπίπεδου Προγραμματισμού	19
Γενικό Μοντέλο Προβλημάτων Γραμμικού Διεπίπεδου Προγραμματισμού	20
Γενικό Μοντέλο Προβλημάτων Μικτού Γραμμικού Ακέραιου Διεπίπεδου Προγραμματισμού	20
2. Συλλογή Προϊόντων.....	21
3. Διάταξη των προϊόντων στην αποθήκη	25
Στρατηγικές Αποθήκευσης.....	25
4. Πρόβλημα Περιοδεύοντος Πωλητή.....	27
5. Αλγόριθμοι επίλυσης TSP προβλημάτων	29
Αναλυτικοί Αλγόριθμοι	29
Κλάδου και Φράγματος	29
Κλάδου και Τομής (Branch and Cut)	32
Κλάδου και Αξιολόγησης	33
Ευρετικοί αλγόριθμοι	34
Γενετικοί Αλγόριθμοι.....	35
Μέθοδος Προσομοιωμένης Ανόπτωσης	37
Μέθοδος Απαγορευμένης Έρευνας.....	38
Εξελικτικοί Αλγόριθμοι	41
Αλγόριθμος Επίλυσης Προβλημάτων Περιοδεύοντος Πωλητή σε Αποθήκες Παράλληλογράμμου Σχήματος	43

6. Περιγραφή του γενικού προβλήματος εύρεσης της βέλτιστης διάταξης προϊόντων σε αποθήκη.....	44
7. Γενική περιγραφή του αλγορίθμου επίλυσης.....	46
8. Υπόθεση Εργασίας.....	53
9. Αποτελέσματα.....	56
10. Συμπεράσματα.....	61
Παράρτημα	62
Αναλυτική περιγραφή του αλγόριθμου επίλυσης του προβλήματος	62
Περιγραφή του αλγόριθμου με τη βοήθεια του οποίου υπολογίζονται οι ελάχιστες αποστάσεις.....	65
Τμήματα του κώδικα που χρησιμοποιήθηκαν σαν ορολογίες παραπάνω ..	67
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	83

Λίστα σχημάτων

Σχήμα 1 Τα στοιχεία μιας εφοδιαστικής αλυσίδας.....	11
Σχήμα 2 Άμεση αποστολή προϊόντων από προμηθευτές σε καταστήματα....	13
Σχήμα 3 Μεταφορά προϊόντων από προμηθευτές σε καταστήματα με την βοήθεια κέντρου διανομής.	14
Σχήμα 4 Τρόπος λειτουργίας του μαθηματικού προγραμματισμού για επίλυση προβλημάτων βελτιστοποίησης.....	16
Σχήμα 5 Ο τρόπος λειτουργίας και δημιουργίας κλαδιών του B&B αλγόριθμου.	31
Σχήμα 6 Οι διάφορες κατηγορίες ευρετικών αλγορίθμων.....	35
Σχήμα 7 Τρόπος λειτουργίας του Crossover Operator.....	36
Σχήμα 8 Τρόπος επιλογής της βέλτιστης αποδεκτής υποψήφιας λύσης. (Αλγοριθμική απεικόνιση).....	40
Σχήμα 9 Τρόπος επιλογής της βέλτιστης αποδεκτής υποψήφιας λύσης. (Διαγραμματική απεικόνιση).....	41
Σχήμα 10 Τρόπος λειτουργίας των EAs.....	43
Σχήμα 11 Λογικό διάγραμμα απεικόνισης του αλγορίθμου που κατασκευάστηκε.....	46
Σχήμα 12 Διαδρομή συλλογής των προϊόντων μιας παραγγελία διανύοντας της ελάχιστης απόστασης.	47
Σχήμα 13 Πραγματική διάταξη των προϊόντων στην αποθήκη.	48
Σχήμα 14 Διάταξη των προϊόντων μετά την 1 ^η αλλαγή.	49
Σχήμα 15 Διάταξη των προϊόντων μετά την 2 ^η αλλαγή.	50
Σχήμα 16 Διάταξη των προϊόντων μετά την 3 ^η αλλαγή.	51
Σχήμα 17 Διάταξη των προϊόντων μετά την 4 ^η αλλαγή.	52
Σχήμα 18 Δείγμα δελτίου παραγγελίας.....	54
Σχήμα 19 Η κάτοψη της αποθήκης.	55
Σχήμα 20 Διάταξη των προϊόντων μετά την 5 ^η αλλαγή.	59
Σχήμα 21 Η πορεία του Total_distance με τις αλλαγές στη διάταξη των προϊόντων της αποθήκης.....	60
Σχήμα 22 Προτιμώμενο σχήμα αποθήκης για εφαρμογή του αλγορίθμου.	65
Σχήμα 23 Απεικόνιση της αποθήκης με τα σημεία κουκίδες που αντιπροσωπεύουν κόμβους και προϊόντα.....	66
Σχήμα 24 Μια διαδρομή για τη συλλογή των προϊόντων μιας παραγγελίας...	66
Σχήμα 25 Διαφορετικές διαδρομές για την ένωση των σημείων a,b.....	67

Λίστα Πινάκων

Πίνακας 1 Οι συνολικές αποστάσεις για τις 5 πρώτες διατάξεις των προϊόντων στην αποθήκη.....	56
Πίνακας 2 Οι τιμές του Total_distance κατά τις διαδοχικές αλλαγές που λαμβάνουν χώρα στην 5η μεθοδολογία.	57

Περίληψη

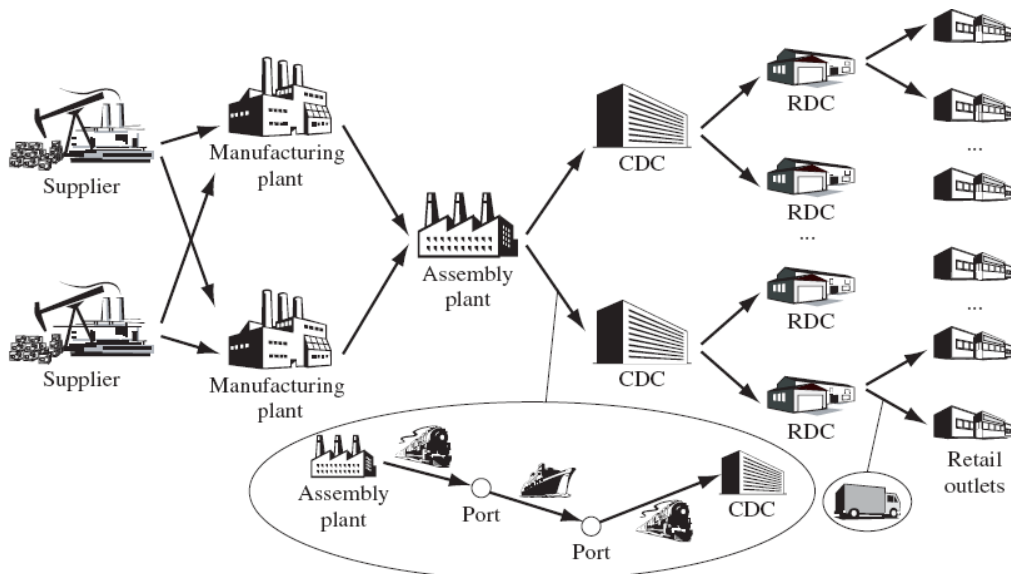
Η παρακάτω εργασία αφορά την ανάπτυξη ενός υβριδικού αλγορίθμου, με σκοπό την βελτιστοποίηση της χωροθέτησης προϊόντων σε αποθήκες. Το πρόβλημα που αναλύεται ανήκει στην κατηγορία των προβλημάτων διεπίπεδου γραμμικού ακέραιου προγραμματισμού. Το πρωτεύον πρόβλημα αφορά την εύρεση της βέλτιστης διάταξης των προϊόντων στον χώρο της αποθήκης, με σκοπό την ελάττωση της συνολικής απόστασης που διανύεται από τους συλλέκτες για την συλλογή όλων των παραγγελιών. Το δευτερεύον πρόβλημα είναι μια παραλλαγή του προβλήματος του περιοδεύοντος πωλητή και αφορά την επιλογή της βέλτιστης διαδρομής για την συλλογή των προϊόντων μιας παραγγελίας, ώστε να ελαχιστοποιείται ο χρόνος συλλογής. Το σύνολο των δυνατών λύσεων στο πρωτεύον πρόβλημα είναι $(n)_k = n!/(n-k)!$, όπου n είναι ο αριθμός των θέσεων στην αποθήκη και k είναι οι κωδικοί των προϊόντων. Έτσι για την επίλυση του αναπτύχθηκε μεταερευνητική μεθοδολογία, καθώς η χρήση κάποιας αναλυτικής μεθοδολογίας είναι αδύνατη. Για την επίλυση του δευτερεύοντος προβλήματος χρησιμοποιήθηκε αναλυτική μέθοδος επίλυσης. Πιο συγκεκριμένα, για την επίλυση του πρωτεύοντος προβλήματος, χρησιμοποιήθηκαν μεθοδολογίες παραγωγής νέων λύσεων που στηρίζονται στην συχνότητα εμφάνισης των προϊόντων στο σύνολο των παραγγελιών, αλλά και την σχετική συχνότητα εμφάνισης ζευγαριών προϊόντων, ενώ για την επίλυση του δευτερεύοντος χρησιμοποιήθηκε ένας καθιερωμένος, στη διεθνή βιβλιογραφία, αλγόριθμος δυναμικού προγραμματισμού. Για την αποδοχή/απόρριψη των διάφορων λύσεων που παράγονται, ακολουθήθηκε η λογική μιας παραλλαγής της μεθόδου της Προσομοιωμένης Ανόπτησης. Η υπόθεση εργασίας, στην οποία εφαρμόστηκε ο αλγόριθμος, αφορά την αποθήκη της εταιρίας Μπαρμπαστάθης στην Αθήνα, που λειτουργεί ως κέντρο διανομής για τις κεντρικές αποθήκες στην Θεσσαλονίκη. Η αποθήκη είναι παραλληλόγραμμη με διαδρόμους μεταξύ των ραφιών αποθήκευσης, που επικοινωνούν μόνο μέσω ενός οριζόντιου διαδρόμου. Τα δεδομένα που χρησιμοποιήθηκαν για την αξιολόγηση του αλγόριθμου αφορούν ιστορικά δεδομένα παραγγελιών, αλλά και διάταξης των προϊόντων στον χώρο της αποθήκης. Τα αποτελέσματα που παράγονται από τον αλγόριθμο είναι ιδιαίτερα ενθαρρυντικά, αφού δείχνουν ότι μπορεί να εξοικονομηθεί μέχρι και 46% της συνολικής απόστασης που διανύουν αυτή τη στιγμή οι συλλέκτες για το σύνολο των παραγγελιών, αν γίνουν οι απαραίτητες αλλαγές στη διάταξη των προϊόντων στον χώρο.

Abstract

This thesis reports the development of a hybrid algorithm utilized for the optimization of the Stock Keeping Units (SKUs) assignment in rectangular warehouses. The analyzed problem belongs to the class of linear integer bi-level programming problems. The Leader problem pertains to the search of the optimal assignment of the SKUs in the storage space in order to minimize the accumulated distance traveled by the order pickers for all orders. The Follower problem, is a variation of the Traveling Salesman Problem, and concerns the selection of the route that minimizes order collection times. The set of possible solutions for the Leader problem has a cardinality $(n)_k = n!/(n-k)!$, where n is the number of SKU positions and k is the number of product codes. For the solution, a metaheuristic methodology was developed, as conventional analytical methodologies were inapplicable. This is not the case for the solution of the secondary problem, where an analytical methodology was used. More specifically, for the solution of the Leader problem, heuristic solution generating methodologies that are based on the frequency by which a product appears in the order set and the relative frequency by which coupled products appear simultaneously in different orders were applied. For the Follower problem a well established in the international literature dynamic programming algorithm, was used. For the acceptance or rejection of the generated solutions, the rationale of a simulated annealing method variation was followed. The case study for which the algorithm was applied, concerns a storage warehouse of the Barbastathis Company in Athens that serves as a distribution center for the central warehouses of Thessaloniki. The warehouse is rectangular with aisles between the storage racks that interconnect through a single horizontal aisle. The data used for the assessment of the algorithm consist of historical data of orders and of the initial SKUs assignment. The results generated by the algorithm are particularly promising, showing that the company can reduce up to 46% of the total distance travelled by the order pickers, if the SKU assignment in the warehouse is adjusted accordingly.

Εισαγωγή

Ο κλάδος των Logistics έχει να κάνει με τον προγραμματισμό και την ρύθμιση της διακίνησης διαφόρων προϊόντων και σχετικών με αυτά πληροφοριών, σε πολλών ειδών εταιρείες, ή οργανώσεις στον ιδιωτικό αλλά και στο δημόσιο τομέα. Γενικά, η αποστολή του κλάδου αυτού και των σχετικών με αυτόν εργασιών, είναι να φέρει το σωστό προϊόν στην κατάλληλη θέση, την κατάλληλη στιγμή, καθώς θα βελτιστοποιεί κάποιο μέτρο απόδοσης(π.χ. ελαχιστοποίηση κόστους λειτουργίας) και θα ικανοποιεί ένα σύνολο περιορισμών (π.χ. περιορισμός στον προϋπολογισμό). Ένα σύστημα Logistics αποτελείται από ένα σύνολο τοποθεσιών που διευκολύνουν την εξυπηρέτηση των διαφόρων μεταφορών που γίνονται. Σε αυτές τις τοποθεσίες τα διάφορα προϊόντα υπόκεινται σε επεξεργασία. Κάποιες από αυτές είναι : Κέντρα Διανομής (ΚΔ), αποθήκες, σημεία μεταμόρφωσης-επεξεργασίας, κέντρα παραγωγής, χωματερές, αποτεφρωτήρες απορριμμάτων κ.α. Η εφοδιαστική αλυσίδα (Supply chain) είναι ένα περίπλοκο σύστημα Logistics, στο οποίο κάποιες πρώτες ύλες επεξεργάζονται και μετατρέπονται σε τελικά προϊόντα που διανέμονται στους καταναλωτές ή τις εταιρείες που τελικά τα αγοράζουν. Αυτή η αλυσίδα προκειμένου να είναι λειτουργική χρειάζεται: αποθήκη, κέντρα διανομής, προμηθευτές, κέντρα παραγωγής και καταστήματα λιανικής πώλησης. (Σχήμα 1)



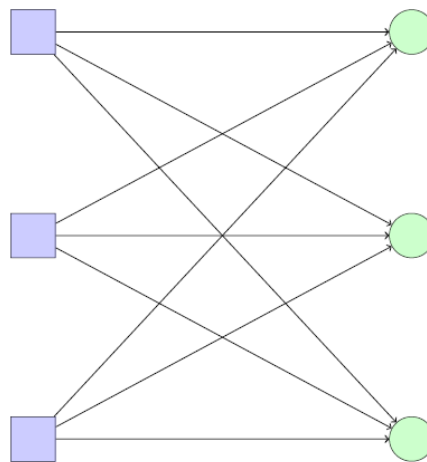
Σχήμα 1 Τα στοιχεία μιας εφοδιαστικής αλυσίδας.

Οι αποθήκες είναι τα σημεία μιας εφοδιαστικής αλυσίδας στα οποία το προϊόν «σταματάει» για κάποιο χρονικό διάστημα, πράγμα το οποίο σημαίνει πως καταλαμβάνει και χώρο, αλλά και χρόνο . Έτσι, γίνεται εύκολα αντιληπτό πως η διαχείριση μιας αποθήκης παίζει σημαντικό ρόλο για ολόκληρη την εφοδιαστική αλυσίδα κάποιου συνόλου προϊόντων. Η αποθήκη και πιο συγκεκριμένα η σωστή διαχείρισή της προσφέρει μερικές σημαντικές δυνατότητες που της δίνουν κάποια ιδιαίτερη αξία, παρά τον χώρο, τον χρόνο

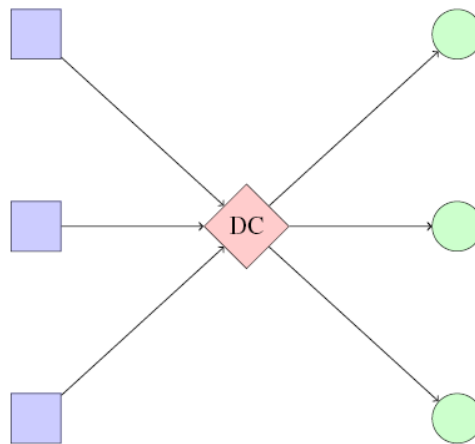
και το χρήμα που ξοδεύεται για να υπάρχει και να συντηρείται. Οι σημαντικότερες από αυτές είναι:

- **Ο προγραμματισμός των προμηθειών για την ικανοποίηση της ζήτησης των πελατών.** Μία απ' τις σημαντικότερες δοκιμασίες που πρέπει συχνά να αντιμετωπιστεί για τη σωστή διαχείριση μιας εφοδιαστικής αλυσίδας, είναι η διαφορά χρόνου με την οποία αλλάζει η ζήτηση των πελατών και ο ανεφοδιασμός της αλυσίδας (που αλλάζει πιο αργά). Δεν είναι λίγες οι φορές που υπάρχει αναπάντεχα αυξημένη ζήτηση κάποιου προϊόντος ή κάποιου συνόλου προϊόντων (π.χ. λόγω εποχικότητας). Τα καταστήματα λιανικής πώλησης σε τέτοιες περιπτώσεις δυσκολεύονται να ανταπεξέλθουν στις απαιτήσεις των πελατών σε αντίθεση με κάποια εταιρεία που έχει την υποστήριξη κάποιας αποθήκης. Με τα αποθέματα που υπάρχουν στην αποθήκη, λοιπόν, δίνεται η δυνατότητα για γρήγορη ανταπόκριση στην αυξημένη ζήτηση. Ένα άλλο πρόβλημα που λύνεται με την βοήθεια της αποθήκης και της σωστής διαχείρισης της είναι η καθυστέρηση που υπάρχει στις μεταφορές από κάποια πόλη σε κάποια άλλη, προκειμένου να φτάσει το προϊόν στο σημείο όπου θα πωλείται. Αυτές οι μεταφορές μπορεί να οφείλονται είτε λόγω κυκλοφοριακού είτε γραφειοκρατικού είτε καιρικού κ.α. προβλήματος. Με την ύπαρξη αποθήκης κοντά στο σημείο πώλησης, λοιπόν, αποφεύγονται οι μεγάλες αποστάσεις και οι μεταφορές, με αποτέλεσμα την αποφυγή μη ανταπόκρισης στις ανάγκες των πελατών. Τέλος με το σωστό σχεδιασμό της ποσότητας των αποθεμάτων σε μια αποθήκη μπορούν να αποφευχθούν και προβλήματα που έχουν να κάνουν με μειωμένη αγορά προμηθειών (π.χ. λόγω αύξηση των τιμών από τους προμηθευτές). Σε αυτή την περίπτωση με το απόθεμα που υπάρχει στην αποθήκη μπορούν να καλυφθούν οι ανάγκες των πελατών για ένα ικανοποιητικό διάστημα, τουλάχιστον , μέχρι να γίνει κάποια νέα συμφωνία με τους προμηθευτές.
- **Η συγκέντρωση και η ομαδοποίηση των διάφορων προϊόντων για την μείωση του κόστους μεταφοράς και την γρηγορότερη εξυπηρέτηση των πελατών.** Υπάρχει ένα σταθερό κόστος κάθε φορά που μεταφέρεται ένα προϊόν. Αυτό το κόστος, μάλιστα, αυξάνεται αισθητά όταν το μέσο μεταφοράς είναι αεροπλάνο, πλοίο ή τρένο, για αυτό και πολλές φορές τα μέσα αυτά φορτώνονται μέχρι και τη μέγιστη χωρητικότητά τους. Κατά συνέπεια, μερικές φορές οι διανομείς συγκεντρώνουν πολλές παραγγελίες μαζί στις αποθήκες τους και τις κάνουν μία προκειμένου να γλιτώσουν και χρόνο, αλλά και έξοδα. Με αυτό τον τρόπο τα φορτηγά που χρησιμοποιούνται για την μεταφορά, λειτουργούν πιο εύκολα και πιο γρήγορα, καθώς γίνεται ευκολότερος ο προγραμματισμός της αποβάθρας που χρησιμοποιεί το κάθε φορτηγό

και επιπλέον δε χρειάζεται να περιμένει αφού οι παραγγελίες είναι ήδη συγκεντρωμένες. Ας πάρουμε για παράδειγμα μια εταιρεία, η οποία έχει εκατοντάδες καταστήματα τα οποία προμηθεύονται τα προϊόντα τους από εκατοντάδες προμηθευτές. Επειδή λοιπόν, οι αποστολές είναι συχνές κανένας προμηθευτής δεν στέλνει μεγάλες ποσότητες προϊόντων στα μαγαζιά αυτά. Αν οι αποστολές αυτές γίνονταν απ'ευθείας στα καταστήματα ο κάθε προμηθευτής θα έπρεπε να στέλνει εκατοντάδες φορτηγά καθημερινά μισοάδεια, πράγμα το οποίο θα κόστιζε αρκετά χρήματα σε μια εταιρεία (Σχήμα 2) Στην περίπτωση ύπαρξης μιας ενδιάμεσης αποθήκης (Cross Dock ή Distribution Centre-ΚΔ -), αντίθετα, θα επιτρεπόταν στον προμηθευτή να στέλνει φορτηγά με μεγαλύτερους όγκους προμηθειών τα οποία θα συγκεντρώνονται και θα ομαδοποιούνται στην αποθήκη με βάση το μαγαζί στο οποίο θα αποσταλούν. Με αυτό τον τρόπο και οι προμηθευτές θα στέλνουν λιγότερες παραγγελίες , αλλά και τα μαγαζιά θα λαμβάνουν λιγότερες παραγγελίες. Αυτό θα γίνεται χρησιμοποιώντας φορτηγά τα οποία θα μεταφέρουν ποσότητα προϊόντων κοντά στην μέγιστη χωρητικότητά τους. Έτσι λοιπόν χρησιμοποιούνται και λιγότερα μέσα μεταφοράς με αποτέλεσμα να μειώνεται το κόστος μεταφοράς (Σχήμα 3).



Σχήμα 2 Άμεση αποστολή προϊόντων από προμηθευτές σε καταστήματα.



Σχήμα 3 Μεταφορά προϊόντων από προμηθευτές σε καταστήματα με την βοήθεια κέντρου διανομής.

- **Διαθεσιμότητα σε εποχιακά προϊόντα καθ'όλη τη διάρκεια του χρόνου.** Τα εποχιακά προϊόντα μπορούν να παράγονται κανονικά και να αποθηκεύονται στην αποθήκη και να πωλούνται αρκετούς μήνες μετά εφόσον υπάρχει η κατάλληλη διαθέσιμη ποσότητα.

Το πρόβλημα που θα αναλυθεί παρακάτω κατατάσσεται στα Προβλήματα Ακέραιου Διεπίπεδου Προγραμματισμού [33] καθώς επιδιώκεται η βελτιστοποίηση δύο αντικειμενικών συναρτήσεων που έχουν να κάνουν με τις αποστάσεις που διανύουν οι συλλέκτες των προϊόντων μιας αποθήκης και οι μεταβλητές που συμμετέχουν σε αυτές είναι ακέραιες. Οι μεταβλητές αυτές στο συγκεκριμένο πρόβλημα είναι η διάταξη των προϊόντων στον χώρο της αποθήκης και η διαδρομή που επιλέγεται να κάνουν οι συλλέκτες για να ολοκληρώσουν μια παραγγελία. Βελτιστοποιώντας, λοιπόν, αυτές τις συναρτήσεις με αυτές τις μεταβλητές επιτυγχάνεται η ελαχιστοποίηση των αποστάσεων που διανύονται από τους εργαζόμενους. Συνεπώς ελαχιστοποιείται ο χρόνος, άρα και το κόστος λειτουργίας του αποθηκευτικού χώρου. Και τα δύο αυτά στοιχεία είναι πολύ σημαντικά για τις εφοδιαστικές αλυσίδες γενικότερα, αφού αυτός είναι ο κύριος σκοπός των ανθρώπων που τις διαχειρίζονται, δηλαδή να μπορούν να εξυπηρετούν τους πελάτες τους γρήγορα και με όσο λιγότερο κόστος μπορούν. Για την επίλυση του μοντέλου που φτιάχτηκε δημιουργήθηκαν ευρετικοί αλγόριθμοι, καθώς δεν υπήρχαν οι κατάλληλες προϋποθέσεις για την χρήση αναλυτικών αλγορίθμων και συνεπώς δε θα είχαμε καλά αποτελέσματα. Τα αποτελέσματα που παράγονται είναι ιδιαίτερα ενθαρρυντικά, αφού φαίνεται ότι επιδέχεται μεγάλη βελτίωση η λειτουργία της αποθήκης. Αυτό μπορεί να τεκμηριωθεί από το αποτέλεσμα της συνολικής απόστασης που βελτιώνεται περίπου κατά 43%.

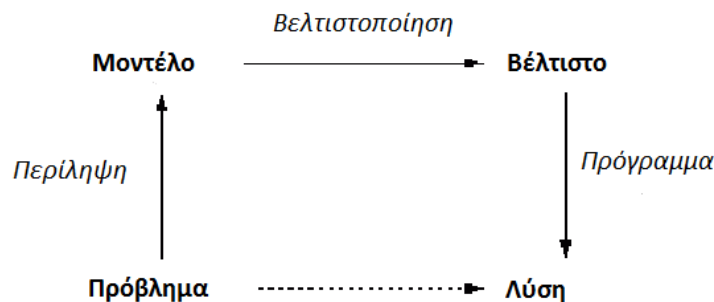
Στις παρακάτω ενότητες περιγράφεται αναλυτικά το πρόβλημα, παρουσιάζονται ορισμένες έννοιες και θεωρίες που θα πρέπει να

κατανοηθούν, ώστε να γίνει σαφής και εύκολα αντιληπτή η ενότητα που επιλύεται το πρόβλημα και παρουσιάζονται τα αποτελέσματα.

Πιο συγκεκριμένα, στην 1^η ενότητα γίνεται μια σύντομη αναφορά στον μαθηματικό προγραμματισμό γενικά, τις χρήσεις του και γίνεται μια πιο εκτενής ανάλυση για το Διεπίπεδο Προγραμματισμό και τα μοντέλα του. Ακολουθεί η 2^η ενότητα που αναφέρεται γενικά στον τρόπο με τον οποίο συλλέγονται τα προϊόντα μέσα σε μια αποθήκη (Order Picking), ενώ περιγράφονται εκτενέστερα μέσα στο κείμενο διάφορες μέθοδοι συλλογής προϊόντων με βάση διαφορετικά κριτήρια. Στην 3^η ενότητα γίνεται περιγραφή των τρόπων με τους οποίους μπορούν να διαταχθούν τα διάφορα προϊόντα μέσα στο χώρο μιας αποθήκης (SKU assignment). Στην 4^η ενότητα γίνεται μια σύντομη αναφορά στο Πρόβλημα Περιοδεύοντος Πωλητή (TSP). Στην 5^η ενότητα, η οποία περιλαμβάνει αλγόριθμους επίλυσης των TSP, αλλά και γενικότερα περίπλοκων προβλημάτων, έχουμε τρεις υποενότητες: τους αναλυτικούς αλγόριθμους, όπου περιγράφονται τρεις από αυτούς που χρησιμοποιούνται πιο συχνά, τους ευρετικούς αλγόριθμους, όπου περιγράφονται τέσσερις κλασσικές, επίσης, περιπτώσεις και μια ειδική κατηγορία αλγορίθμων χρήσιμη για την επίλυση του προβλήματος που θα ασχοληθούμε. Στην 6^η ενότητα γίνεται αναφορά στο γενικότερο πρόβλημα στο οποίο ανήκει η περίπτωση μας και στο μαθηματικό μοντέλο που ακολουθεί. Στην 7^η ενότητα παρουσιάζεται αναλυτικά ο γενικός αλγόριθμος επίλυσης του προβλήματος. Στην 8^η ενότητα γίνεται περιγραφή της περίπτωσης με την οποία θα ασχοληθούμε και στην οποία θα εφαρμοστεί ο αλγόριθμος. Τέλος, στην 9^η ενότητα παρουσιάζονται και σχολιάζονται τα αποτελέσματα που παράγει ο αλγόριθμος.

1. Μαθηματικός προγραμματισμός

Ένας ορισμός που θα μπορούσε να δοθεί στην παραπάνω έννοια είναι ο εξής: Η χρήση μαθηματικών μοντέλων, ιδιαίτερα μοντέλων βελτιστοποίησης, τα οποία βοηθούν στη λήψη διαφόρων αποφάσεων. Ένα ιδιαίτερο χαρακτηριστικό του είναι ότι η βέλτιστη λύση για ένα μοντέλο δίνεται αυτόματα μέσω λογισμικού που εκτελεί βελτιστοποίηση και είναι εγκατεστημένο στον υπολογιστή, όπου γίνεται ο προγραμματισμός. Απαντάει στην ερώτηση «Τι είναι καλύτερο;» , γι'αυτό κιόλας είναι πιο περιορισμένη η εμβέλεια χρήσης του σε σχέση με άλλες μεθόδους. Η αλήθεια είναι ότι δεν βρίσκει πάντα τη βέλτιστη λύση σε ένα πραγματικό πρόβλημα, αλλά βρίσκει τη βέλτιστη λύση στο μοντέλο που έχει δημιουργηθεί και περιγράφει κατά κάποιον τρόπο το πραγματικό πρόβλημα. Αν, λοιπόν, το μοντέλο έχει στηθεί σωστά τότε η λύση που θα προταθεί θα είναι και η πραγματικά βέλτιστη, αν όμως δεν έχει στηθεί απόλυτα σωστά, τότε δε θα είναι η πραγματικά βέλτιστη λύση. Η ανάλυση όμως του γιατί αυτή η λύση δεν είναι η βέλτιστη θα μπορούσε να οδηγήσει στην καλύτερη κατανόηση του πραγματικού προβλήματος τελικά (Σχήμα 4) [3].



Σχήμα 4 Τρόπος λειτουργίας του μαθηματικού προγραμματισμού για επίλυση προβλημάτων βελτιστοποίησης.

Τα βασικά χαρακτηριστικά που πρέπει να έχει ένα πραγματικό πρόβλημα για να λυθεί με Μαθηματικό Προγραμματισμό είναι:

- Πολλές πιθανές αποδεκτές λύσεις.
- Ορισμένα μέσα για την αξιολόγηση της ποιότητας των εναλλακτικών λύσεων.
- Κάποια διασύνδεση μεταξύ των μεταβλητών στοιχείων (μεταβλητών) του συστήματος [3].

Αυτά τα απαραίτητα χαρακτηριστικά αντικατοπτρίζονται και στα βασικά συστατικά ενός μαθηματικού μοντέλου τα οποία είναι:

- Οι τιμές των μεταβλητών απόφασης που εκφράζουν ουσιαστικά τους αγνώστους του προβλήματος και είναι οι μεταβλητές που ελέγχει ο

αποφασίζων, δηλ. εκείνες των οποίων τις τιμές μπορεί να καθορίσει. Το σύνολο των μεταβλητών απόφασης αποτελεί ουσιαστικά το αντικείμενο της διαδικασίας λήψης απόφασης. Η διαδικασία αριστοποίησης αποσκοπεί στο να βρεθούν οι τιμές εκείνες για τις μεταβλητές απόφασης οι οποίες βελτιστοποιούν την αντικειμενική συνάρτηση.

- Η αντικειμενική συνάρτηση που αποτελεί τη μαθηματική σχέση των μεταβλητών απόφασης που εκφράζει το κριτήριο βελτιστοποίησης. Επιδιώκεται είτε η ελαχιστοποίηση είτε η μεγιστοποίησή της τιμής της αντικειμενικής συνάρτησης.
- Οι σχέσεις μεταξύ των μεταβλητών απόφασης, ή αλλιώς οι περιορισμοί. Περιορισμοί είναι οι μαθηματικές σχέσεις που καθορίζουν τις τιμές που μπορούν να πάρουν οι μεταβλητές απόφασης στη διαδικασία της βελτιστοποίησης. Καθορίζουν δηλαδή το πεδίο ορισμού (εφικτό χωρίο) του προβλήματος. Οι περιορισμοί μπορεί να είναι ισότητες ή/και ανισότητες.
- Εξωτερικές μεταβλητές εισόδου (παράμετροι) και δεδομένα. Παράμετροι είναι τα εξωγενώς οριζόμενα (εκτός του ελέγχου του αποφασίζοντα) μεγέθη του προβλήματος. Πρόκειται ουσιαστικά για τους γνωστούς όρους του προβλήματος οι οποίοι έχουν σταθερή τιμή στη διαδικασία βελτιστοποίησης. Συνήθως είναι συντελεστές των μεταβλητών απόφασης ή εκφράζουν ποσότητες απαραίτητες στη διαμόρφωση των περιορισμών (π.χ. την απαιτούμενη ζήτηση μιας δραστηριότητας) [3, 6].

Εφαρμογές – Χρήση

1. **Περιοριστική.** Προτείνει πρακτικές λύσεις για την επίλυση προβλημάτων
2. **Πρόβλεψη.** Προβλέπει τις συνέπειες που θα επιφέρει μια στρατηγική.
3. **Ανάλυση Ευαισθησίας.** Αξιολογεί τα μοντέλο και την πραγματικότητά του και δίνει απαντήσεις για τα “what if” σενάρια.
4. **Οικονομετρική / Στατιστική Ανάλυση** [5].

Γενικό Μοντέλο

$$\text{Optimize } F(x)$$

$$\text{Υπό συνθήκες (υ.σ.) } G(x) \leq S_1$$

$$\text{και } x \in S_2$$

Όπου $F(x)$, $G(x)$ συναρτήσεις του x , όπου x διάνυσμα μεταβλητών απόφασης και S_1 , S_2 σύνολα που καθορίζουν τους περιορισμούς του εκάστοτε προβλήματος.

Είδη μοντέλων

- **Γραμμικός Προγραμματισμός (LP).** $F(x)$, $G(x)$ είναι γραμμικές και το x μη αρνητικό και συνεχές.

Η γενική μορφή του LP είναι η παρακάτω:

$$\text{Maximize (minimize) } z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

$$\text{s.t. : } a_{11}x_1 + a_{12}x_2 + a_{1n}x_n (\leq, \geq, =) b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{2n}x_n (\leq, \geq, =) b_2$$

$$a_{m1}x_1 + a_{m2}x_2 + a_{mn}x_n (\leq, \geq, =) b_m$$

Όπου τα x_i είναι μη αρνητικά.

- **Μη Γραμμικός Προγραμματισμός (NLP).** $F(x)$ και/ή $G(x)$ μη γραμμικές και x μη αρνητικό και συνεχές.
- **Ακέραιος Προγραμματισμός (IP).** $F(x)$ και $G(x)$ γραμμικές ή/και μη γραμμικές και x ακέραιο.
- **Μικτός Ακέραιος Προγραμματισμός (MIP).** $F(x)$ και $G(x)$ γραμμικές ή/και μη γραμμικές και x ακέραιο και συνεχές.
- **Τετραγωνικός Προγραμματισμός (QP).** $F(x)$ τετραγωνική και $G(x)$ γραμμική και x μη αρνητικό [5, 6].

Μεταβλητές ενός μοντέλου μαθηματικού προγραμματισμού

Οι μεταβλητές αντιπροσωπεύουν αποφάσεις που πρέπει να παρθούν προκειμένου να λειτουργήσει το σύστημα.

Υπάρχουν έξι βασικές ομάδες μεταβλητών: μεταβλητές απόφασης, οι μεταβλητές εισόδου, μεταβλητές κατάσταση, εξωγενείς μεταβλητές, τυχαίες μεταβλητές και οι μεταβλητές εξόδου [4].

Διεπίπεδος Προγραμματισμός

Τα προβλήματα Διεπίπεδου Προγραμματισμού παρουσιάστηκαν για πρώτη φορά από τον von Stackelberg το 1934 [25]. Μπορούν να παρομοιαστούν με ιεραρχικά παιχνίδια στο οποίο συμμετέχουν δύο φορείς λήψης αποφάσεων, όπου ο πρώτος (που ονομάζεται Leader-αρχηγός) έχει τον πρώτο λόγο ή αλλιώς κάνει την πρώτη επιλογή και ο δεύτερος (που ονομάζεται Follower-ακόλουθος) έχει την βέλτιστη αντίδραση με βάση ήδη υφιστάμενη επιλογή του Αρχηγού. Στόχος του Αρχηγού είναι να βρει μια επιλογή, η οποία σε συνδυασμό με την βέλτιστη αντίδραση του Ακόλουθου θα βελτιστοποιήσει την αντικειμενική συνάρτηση του Αρχηγού [25].

Είναι ένα είδος μαθηματικού προγραμματισμού το οποίο έχει ένα ιδιαίτερο χαρακτηριστικό: Περιέχεται πρόβλημα βελτιστοποίησης μέσα στους περιορισμούς [26]. Είναι ένα είδος πολυεπίπεδου προγραμματισμού και πιο συγκεκριμένα δύο επιπέδων. Τα προβλήματα που επιλύονται με τη χρήση του συγκεκριμένου προγραμματισμού δημιουργήθηκαν λόγω της ανάγκης εξέλιξης δύο άλλων σχετικών με αυτόν επιστημονικών κλάδων:

1. Λογική επέκταση των προβλημάτων του μαθηματικού προγραμματισμού

2. Γενίκευση ενός συγκεκριμένου προβλήματος στην θεωρία παιγνίων (Stackelberg Game)

Βρίσκει εφαρμογή σε πολλούς τομείς, όπως οικονομικά, διάφορους τομείς της μηχανικής γενικότερα, κατανομή των πόρων, σχεδιασμός δικτύων μεταφοράς, διαχείριση συμφόρησης κ.α. και κυρίως σε πολυεπίπεδα συστήματα (multilevel systems) αυτών των τομέων. Για παράδειγμα, η αντικειμενική συνάρτηση που έχει να κάνει με ένα τμήμα μιας εταιρείας είναι μερικώς ορισμένη και επηρεασμένη από μεταβλητές που ελέγχονται από άλλα τμήματα της [7, 8].

Γενικό Μοντέλο Προβλημάτων Διεπίπεδου Προγραμματισμού

$$\begin{aligned} & \min_{x \in X} F(x, y) \\ \text{s.t.} & \\ & G(x, y) \leq 0 \\ & g(x, y) \leq 0 \\ & x, y \leq 0 \end{aligned}$$

όπου $x \in R^{n1}$ και $y \in R^{n2}$. Οι μεταβλητές του προβλήματος (2.1) διαιρούνται σε δύο κατηγορίες, δηλαδή οι μεταβλητές υψηλότερου επιπέδου $x \in R^{n1}$ και οι μεταβλητές χαμηλότερου επιπέδου $y \in R^{n2}$. Ομοίως, οι συναρτήσεις $F: R^{n1} \times R^{n2} \rightarrow R$ και $f: R^{n1} \times R^{n2} \rightarrow R$ είναι οι αντικειμενικές συναρτήσεις του ανώτερου και του κατώτερου επιπέδου, αντίστοιχα, ενώ οι συναρτήσεις $G: R^{n1} \times R^{n2} \rightarrow R^{m1}$ και $g: R^{n1} \times R^{n2} \rightarrow R^{m2}$ ονομάζονται ανώτερου επιπέδου και κατώτερου επιπέδου περιορισμοί αντίστοιχα. Περιορισμούς ανώτερου επιπέδου περιλαμβάνουν οι μεταβλητές και των δύο επιπέδων και διαδραματίζουν έναν πολύ συγκεκριμένο ρόλο [9].

Τα δύο παρακάτω γενικά μοντέλα παρουσιάζονται όπως παρουσιάζονται στην βιβλιογραφία [8]:

Γενικό Μοντέλο Προβλημάτων Γραμμικού Διεπίπεδου Προγραμματισμού

$$\min_{x \in X} F(x, y) = c_1 x + d_1 y$$

s.t.

$$A_1 x + B_1 y \leq b_1$$

$$\min_{y \in Y} f(x, y) = c_2 x + d_2 y$$

s.t.

$$A_2 x + B_2 y \leq b_2$$

Γενικό Μοντέλο Προβλημάτων Μικτού Γραμμικού Ακέραιου Διεπίπεδου Προγραμματισμού

$$\min_{x \in X} F(x, y) = c_1 x + d_1 y$$

s.t.

$$A_1 x + B_1 y \leq b_1$$

$$\min_{y \in Y} f(x, y) = c_2 x + d_2 y$$

s.t.

$$A_2 x + B_2 y \leq b_2$$

$$x \in \mathbb{R},$$

$$y \in \mathbb{Z}$$

2. Συλλογή Προϊόντων

Η συλλογή προϊόντων (order picking) είναι η διαδικασία της συλλογής διαφόρων προϊόντων από το χώρο στον οποίο βρίσκονται αποθηκευμένα, με σκοπό την ομαδοποίησή τους σε παραγγελίες και την αποστολή τους στους πελάτες. Η δραστηριότητα της συλλογής προϊόντων είναι με διαφορά η πιο χρονοβόρα και κοστοβόρα μέσα σε μία αποθήκη ή κέντρο διανομής [27]. Παρόλο που τα ποσοστά διαφέρουν από εταιρεία σε εταιρεία, ανάλογα με τη φύση των προϊόντων και το σύστημα συλλογής των παραγγελιών που εφαρμόζεται, στη συλλογή προϊόντων οφείλεται κατά μέσο όρο περίπου το 60% του συνολικού κόστους εργασίας.

Η σύγχρονη τάση που επικρατεί διεθνώς είναι αυτή των μικρών παραγγελιών με μεγάλη συχνότητα σε αντίθεση με ό,τι συνέβαινε παλαιότερα όπου οι παραγγελίες ήταν μεγαλύτερες αλλά γίνονταν σε αραιότερα χρονικά διαστήματα. Όπως είναι φυσικό η τάση αυτή επηρεάζει καθοριστικά τη δραστηριότητα του συλλογή προϊόντων θέτοντας σε δοκιμασία την αποτελεσματικότητά του. Πριν από κάθε άλλη ενέργεια, λοιπόν, η επιχείρηση καλείται να αποφασίσει τον τρόπο με τον οποίο θα εξυπηρετεί τους πελάτες της αλλά και το επίπεδο της εξυπηρέτησης αυτής. Ζητήματα όπως το μέγεθος της παραγγελίας και η ταχύτητα εκτέλεσης, που είναι καθοριστικά για την ανταγωνιστικότητα της, αποτελούν βασικούς παράγοντες για το (ανα)σχεδιασμό ενός συστήματος συλλογής προϊόντων.

Διάφοροι εξωτερικοί παράγοντες, όπως η εποχικότητα της ζήτησης ή/και της παραγωγής, η ανάπτυξη νέων προϊόντων και οι προωθητικές ενέργειες επηρεάζουν τις αποφάσεις του (ανα)σχεδιασμού της συλλογής προϊόντων και πρέπει οπωσδήποτε να ληφθεί μέριμνα ώστε το σύστημα να ανταποκρίνεται σε αυτές τις αυξημένες ανάγκες.

Επιπλέον βασικά σημεία που πρέπει να εξετασθούν είναι η εργονομία και οι κανόνες ασφαλείας των εργαζομένων. Σε πολλές περιπτώσεις η βελτίωση της εργονομίας επιφέρει από μόνη της σημαντική αύξηση της παραγωγικότητας της συλλογής προϊόντων. Έτσι για παράδειγμα δεν πρέπει τα ογκώδη και βαριά προϊόντα να τοποθετούνται πολύ χαμηλά ή πολύ ψηλά, η κωδικοποίηση των θέσεων πρέπει να είναι ακριβής και να μη δημιουργεί σύγχυση ή παρανοήσεις, η διαδικασία αναπλήρωσης των θέσεων συλλογής πρέπει να γίνεται σε διαφορετικό χρόνο από το picking ώστε να αποφεύγονται λάθη, καθυστερήσεις και ατυχήματα κ.α. [10, 11].

Ανάλογα με το είδος των προϊόντων και τη φύση της επιχείρησης υπάρχουν τρεις διαφορετικές μέθοδοι συλλογής προϊόντων:

- τη συλλογή ανά παραγγελία
- τη συγκεντρωτική συλλογή
- τη συλλογή κατά ζώνη

Στη **συλλογή ανά παραγγελία**, συλλέγονται κάθε φορά τα προϊόντα που αφορούν μία μόνο παραγγελία τα οποία τοποθετούνται πάνω σε παλέτα, κυλιόμενοι κλωβοί, καρότσι ή άλλο μέσο μεταφοράς και οδηγούνται στο χώρο συγκέντρωσης παραγγελιών για την αποστολή τους. Το βασικό πλεονέκτημα αυτής της μεθόδου είναι ότι δεν απαιτείται διαχωρισμός των προϊόντων αφού όλα ανήκουν στην ίδια παραγγελία. Μειονέκτημα αποτελούν οι μεγάλες αποστάσεις που διανύονται από τους εργαζομένους, αφού για κάθε παραγγελία ο εργαζόμενος ξεκινά νέο "κύκλο" μέσα στην αποθήκη, γεγονός που επηρεάζει αρνητικά το ρυθμό συλλογής ο οποίος είναι πολύ χαμηλός. Η μέθοδος αυτή της συλλογής βρίσκει εφαρμογή στην περίπτωση όπου ο αριθμός των παραγγελιών είναι μικρός ή ο αριθμός των γραμμών των παραγγελιών περιορισμένος.

Η **συγκεντρωτική συλλογή** είναι από τις πιο διαδεδομένες μεθόδους. Εδώ οι παραγγελίες ομαδοποιούνται ανά προϊόν και η συλλογή γίνεται για κάθε κωδικό ξεχωριστά. Σε δεύτερο στάδιο γίνεται ο διαχωρισμός των προϊόντων σε κάθε παραγγελία. Είναι προφανές ότι ο τρόπος αυτός πλεονεκτεί σε ταχύτητα αφού ο συλλέκτης-εργαζόμενος συλλέγει όλα τα προϊόντα σε ένα μόνο "κύκλο", αλλά απαιτεί χρόνο, χώρο και προσωπικό για το διαχωρισμό των κιβωτίων. Η μέθοδος αυτή βρίσκει εφαρμογή όταν το μέγεθος των παραγγελιών είναι σχετικά μικρό και το πλήθος των κωδικών αρκετά μεγάλο.

Στη **συλλογή κατά ζώνη** τα προϊόντα ομαδοποιούνται σε ζώνες με βάση τη θέση τους στην αποθήκη. Από κάθε ζώνη συλλέγονται τα απαιτούμενα προϊόντα τα οποία διαχωρίζονται σε δεύτερο στάδιο όπως και στη συγκεντρωτική συλλογή σε κάθε παραγγελία. Το πλεονέκτημα της μεθόδου αυτής είναι ότι μειώνει σημαντικά τις διανυόμενες αποστάσεις, αν και χρειάζεται προσοχή ώστε να μην εξαναγκάζονται οι εργαζόμενοι να συλλέξουν προϊόντα που βρίσκονται εκτός της περιοχής δικαιοδοσίας τους. Η συλλογή κατά ζώνη εφαρμόζεται στις περιπτώσεις εκείνες που το πλήθος των κωδικών είναι πολύ μεγάλο [10].

Ανάλογα με το βαθμό αυτοματοποίησης με τον οποίο εκτελείται η δραστηριότητα της συλλογής μπορούμε να διαχωρίσουμε τα συστήματα του συλλογή προϊόντων σε τρεις βασικές κατηγορίες:

- χειροκίνητα (manual)
- ημιαυτοματοποιημένα (mechanically assisted)
- πλήρως αυτοματοποιημένα (automated)

Στο χειροκίνητο σύστημα συλλογής προϊόντων, το οποίο είναι με διάφορες παραλλαγές το πλέον διαδεδομένο σύστημα, ο εργαζόμενος περπατά μέσα στην αποθήκη και με τη βοήθεια της λίστας που έχει στα χέρια του (picking list) συλλέγει τα προϊόντα και τα τοποθετεί πάνω στην παλέτα ή το καρότσι που έχει μαζί του. Η συλλογή γίνεται από ένα επίπεδο και στις περισσότερες

περιπτώσεις τα προς συλλογή προϊόντα βρίσκονται τοποθετημένα πάνω σε παλέτες, αν και στην περίπτωση που αυτά είναι μικρών διαστάσεων και βρίσκονται τοποθετημένα σε ράφια θυρίδας, τα επίπεδα μπορεί να είναι περισσότερο με τη χρήση παταριού ή μικρού ύψους ραφιού παλέτας στο δεύτερο επίπεδο. Το σύστημα αυτό όπως είναι φανερό δεν απαιτεί ιδιαίτερο εξοπλισμό, είναι πολύ ευέλικτο αλλά υστερεί σημαντικά στον έλεγχο και στην αποδοτικότητα. Με βάση στατιστικά στοιχεία και μελέτες αποδεικνύεται ότι το μεγαλύτερο ποσοστό του χρόνου σε ένα τέτοιο σύστημα αναλώνεται στη μετακίνηση του εργαζομένου μέσα στην αποθήκη, ενώ ο πραγματικός χρόνος για τη συλλογή των προϊόντων δεν ξεπερνά το 25% του συνολικού χρόνου.

Ο ρυθμός συλλογής για το σύστημα αυτό κυμαίνεται από 90 έως 130 χαρτοκιβώτια ανά ώρα και εργαζόμενο στην περίπτωση της συλλογής από παλέτα και τοποθέτησης σε καρότσι ή παλετοφόρο, αν και οι τιμές αυτές μπορούν να διαφοροποιηθούν σημαντικά ανάλογα με το μέγεθος και το βάρος των κιβωτίων και τον αριθμό των κωδικών των προϊόντων. Για την αύξηση του ρυθμού συλλογής μπορούν να τοποθετηθούν τα ταχέως κινούμενα προϊόντα πολύ κοντά στο χώρο συγκέντρωσης των παραγγελιών ώστε να μειωθούν οι διανυόμενες αποστάσεις.

Ιδιαίτερη προσοχή απαιτεί ο δρόμος που θα ακολουθεί ο εργαζόμενος μέσα στην αποθήκη. Το σύστημα Μονοδρόμησης (μορφή "Z") όπου ο εργαζόμενος συλλέγει και από τις δύο πλευρές είναι το πλέον συνηθισμένο εφόσον υπάρχει η δυνατότητα και ο αριθμός των προϊόντων (τεμαχίων ή κιβωτίων) που συλλέγονται από κάθε πλευρά είναι μικρός. Το σύστημα Διπλής Κατεύθυνσης (μορφή "U") είναι προτιμότερο σε φαρδύτερους διαδρόμους (πάνω από 3,0 μ.) και όταν ο αριθμός των προϊόντων που συλλέγονται από κάθε θέση είναι μεγάλος.

Τα ημιαυτοματοποιημένα συστήματα συλλογής προϊόντων ανάλογα με την τεχνική που χρησιμοποιούν διευκολύνουν την πρόσβαση του εργαζομένου σε περισσότερα από ένα επίπεδα ή/και παρέχουν πρόσβαση των προϊόντων στον εργαζόμενο χωρίς αυτός να μετακινηθεί από τη θέση του.

Στην πρώτη περίπτωση ο εργαζόμενος επιβαίνει σε ειδικό ανυψωτικό το οποίο μπορεί να έχει μόνο μία απλή ανακλινόμενη πλατφόρμα ή να ανυψώνει τον χειριστή ο οποίος βρίσκεται μέσα σε καμπίνα σε μεγάλο ύψος. Τα μηχανήματα αυτά μπορούν να κινούνται ελεύθερα (free path order picking trucks) ή σε οδηγούς (rail mounted order picking cranes) πάντα με βάση τους χειρισμούς του εργαζομένου.

Καθοριστικό σημείο για την απόδοση τέτοιων συστημάτων είναι η μείωση των διανυόμενων αποστάσεων, τόσο στο οριζόντιο όσο κυρίως στο κατακόρυφο επίπεδο. Απαιτείται δηλαδή προσεκτικός σχεδιασμός της σειράς με την οποία

θα συλλεχθούν οι κωδικοί ώστε να ελαχιστοποιηθούν οι κινήσεις. Ο ρυθμός συλλογής κυμαίνεται στα επίπεδα του προηγούμενου συστήματος, 90 έως 130 χαρτοκιβώτια ανά ώρα, αλλά το πλεονέκτημα βρίσκεται στην πρόσβαση μεγαλύτερου αριθμού θέσεων και συχνά στη μειωμένη ανάγκη για αναπλήρωση των θέσεων αυτών. Τα μειονεκτήματα του συστήματος είναι το υψηλό κόστος και η μειωμένη ευελιξία.

Άλλη μορφή ημιαυτοματοποιημένου συστήματος χρησιμοποιεί οριζόντια ή κατακόρυφα στροβιλοδρόμια, συνεχή αποθήκευση και μεταφορικές ταινίες. Τα συστήματα αυτά καθοδηγούνται από τον εργαζόμενο με χρήση πληκτρολογίου ή αναγνώστη bar code ή σε πιο εξελιγμένη μορφή από ένα κεντρικό Η/Υ. Η θέση από την οποία πρέπει να συλλεχθούν προϊόντα παρουσιάζεται μπροστά στον χειριστή ο οποίος δεν μετακινείται. Η κίνηση του συστήματος ρυθμίζεται από το σήμα που δίνει ο χειριστής κάθε φορά που ολοκληρώνει τη συλλογή από τη συγκεκριμένη θέση.

Τα πλήρως αυτοματοποιημένα συστήματα συλλογής προϊόντων μπορούν να λειτουργήσουν αποδοτικά όταν είναι δυνατό να προσδιοριστούν επακριβώς τα βασικά χαρακτηριστικά του προϊόντος όπως το μέγεθος, το σχήμα και το βάρος. Σε αρκετές περιπτώσεις τα συστήματα αυτά εφαρμόζονται σε εργοστάσια όπου ο απόλυτος έλεγχος των προϊόντων είναι δυνατόν να επιτευχθεί [10, 11, 27].

Η κύρια λειτουργία της αποθήκης αφορά στην αδιάλειπτη προμήθεια και διανομή προϊόντων. Οι εταιρείες διαθέτουν αρκετές επιλογές για την αποθήκευση των προϊόντων τους. Τα βασικά είδη αποθηκών που αποτελούν εναλλακτικές επιλογές είναι:

- **Cross – Docking.** Η αποθήκη έχει το ρόλο του κέντρου διανομής και ανάμιξης προϊόντων (distribution mixing center). Τα προϊόντα φθάνουν σε μεγάλες παρτίδες και αμέσως κατακερματίζονται σε μικρότερες, αναμειγνύονται με άλλα και δημιουργούνται οι παραγγελίες, οι οποίες αποστέλλονται άμεσα στους πελάτες.
- **3PL.** Η 3PL εταιρεία αναλαμβάνει την φύλαξη και την διανομή των προϊόντων. Με αυτό τον τρόπο η 3PL εταιρεία επωμίζεται την υποχρέωση της σωστής διακίνησης των προϊόντων με την χρήση ειδικού εξοπλισμού, π.χ περονοφόρα, στόλο οχημάτων διανομής, πληροφοριακό σύστημα, κ.λπ.
- **Μισθωμένη ή ιδιόκτητη αποθήκη.** Η ίδια η εταιρεία πρέπει να δαπανήσει ένα ποσό για να εξοπλίσει ένα χώρο ή να ενοικιάσει έτοιμο αποθηκευτικό χώρο, έτσι ώστε να στεγάσει τα αποθέματα των προϊόντων που εμπορεύεται [25].

3. Διάταξη των προϊόντων στην αποθήκη

Η νέα παγκόσμια οικονομία έχει αυξήσει την πίεση στους διανομείς να προμηθεύουν με ταχείς ρυθμούς προϊόντα στους πελάτες. Σε μια τέτοια ανταγωνιστική αγορά, η εκπλήρωση των παραγγελιών των πελατών μέσα σε ένα διάστημα 24 ωρών τείνει να γίνει το νέο πρότυπο σε πολλές βιομηχανίες, συμπεριλαμβανομένων των φαρμακευτικών, των τροφίμων & ποτών, ειδών γραφείου και επίπλων. Εν τω μεταξύ, ο αριθμός των διαφορετικών Μονάδων Αποθήκευσης Προϊόντων (ΜΑΠ) που πρέπει να παραδοθούν αυξάνεται με ταχύτατους ρυθμούς, πράγμα που σημαίνει ότι ο χαμηλός όγκος των όλο και περισσότερων ΜΑΠ πρέπει να παραδοθεί πιο συχνά και πιο γρήγορα.

Περιττό να πούμε ότι τα κέντρα διανομής (ΚΔ) δεν έχουν άλλη επιλογή, παρά να βελτιώσουν τις μεθόδους εκπλήρωσης των παραγγελιών τους μέσω της καλύτερης αποθήκευσης, συλλογής και στρατηγικής δρομολόγησης.

Όταν ένας μεγάλος αριθμός SKUs πρέπει να συλλεχθεί σε μικρές ποσότητες (case picking), τα ΚΔ συνηθίζουν να χωρίζουν το χώρο αποθήκευσης σε δύο μέρη: την «αποθεματική ζώνη» (ή reserve area), όπου τα προϊόντα είναι αποθηκευμένα σε παλέτες, και η «ζώνη προώθησης» (ή fast pick area), όπου είναι αποθηκευμένα και ταξινομημένα τα προϊόντα έτσι ώστε να είναι εύκολη η συλλογή τους από μηχανήματα συλλογής παραγγελιών ή τους ανθρώπους που κάνουν την συλλογή των παραγγελιών. Λαμβάνοντας υπόψη τα παραπάνω, η δουλειά των διαχειριστών των ΚΔ είναι να αποφασίσουν πού να τοποθετήσουν τις διάφορες ΜΑΠ (SKU assignment), με ποιόν τρόπο θα γίνει η συλλογή τους και πόσος χώρος θα πρέπει να χορηγηθεί σε κάθε μία στη περιοχή ταχείας συλλογής, προκειμένου να βελτιστοποιηθεί η εκπλήρωση των παραγγελιών των πελατών. Επιπλέον, αυτά τα στελέχη πρέπει να καθορίσουν στρατηγικές αναπλήρωσης (replenishment) που θα εγγυώνται τη διαθεσιμότητα του εκάστοτε προϊόντος οποιαδήποτε στιγμή χρειάζεται στην «ζώνη προώθησης». Το να πάρει κανείς τις σωστές αποφάσεις είναι δύσκολο, τόσο λόγω του επιπέδου της πολυπλοκότητας, όσο και του αντίκτυπου που μπορεί να έχουν οι αποφάσεις αυτές στην απόδοση του ΚΔ και των λειτουργικών του εξόδων [11].

Στρατηγικές Αποθήκευσης

Οι στρατηγικές αποθήκευσης που ακολουθούνται σε ένα ΚΔ ή μια αποθήκη αφορούν, κυρίως, τον τρόπο με τον οποίο τοποθετούνται οι διάφορες SKU στις διαθέσιμες κενές θέσεις. Μερικές κλασσικές στρατηγικές είναι:

- ❖ **Προκαθορισμένη Αποθήκευση (Dedicated storage).** Τα προϊόντα τοποθετούνται σε προκαθορισμένες θέσεις. Ένα μειονέκτημα της στρατηγικής αυτής είναι ότι ακόμα και αν κάποιο προϊόν δεν είναι διαθέσιμο καταλαμβάνει θεωρητικά μια θέση στην αποθήκη. Επιπλέον, για κάθε προϊόν πρέπει να διατηρείται επαρκής χώρος, τόσο ώστε κάθε στιγμή να μπορεί να αποθηκευτεί η μέγιστη ποσότητα

αποθέματος του προϊόντος αυτού. Τέλος, η συγκεκριμένη στρατηγική μπορεί να φανεί ιδιαίτερα χρήσιμη για την αποθήκευση προϊόντων με διαφορετικά βάρη. Τα βαριά προϊόντα πρέπει να βρίσκονται στο κάτω μέρος της παλέτας και τα ελαφριά στην κορυφή. Με την αποθήκευση των προϊόντων κατά αυτόν τον τρόπο και την ανάλογη δρομολόγηση των συλλεκτών των παραγγελιών επιτυγχάνεται μια καλή σειρά στοίβαξης και κατά συνέπεια διευκολύνεται η συλλογή, χωρίς κάποια ιδιαίτερη διαδικασία.

- ❖ **Τυχαία αποθήκευση (Random storage).** Τα προϊόντα τοποθετούνται σε διαφορετικές, τυχαίες θέσεις ανάλογα με το που υπάρχει διαθέσιμος χώρος. Αυτή η στρατηγική θα μπορούσε μόνο να εφαρμοστεί με την ύπαρξη και την υποστήριξη της απαραίτητης τεχνολογίας από την αποθήκη.
- ❖ **Ομαδοποιημένη αποθήκευση (Class-based storage).** Τα προϊόντα τοποθετούνται σε συγκεκριμένες ζώνες της αποθήκης. Τα προϊόντα τοποθετούνται τυχαία μέσα στην κατάλληλη για αυτά ζώνη. Οι ζώνες συχνά δημιουργούνται με βάση την συχνότητα ζήτησης των προϊόντων. Τα συχνά κινούμενα προϊόντα συνήθως ονομάζονται A-items. Η αμέσως επόμενη κατηγορία με βάση την ζήτηση των προϊόντων ονομάζεται B-items κλπ. Συνήθως οι ζώνες που χωρίζονται τα προϊόντα είναι τρεις, αλλά μερικές φορές ταξινομούνται και σε περισσότερες, στρατηγική η οποία αν έχει σωστή διαχείριση δίνει και καλύτερα αποτελέσματα.
- ❖ **Αποθήκευση στην κοντινότερη διαθέσιμη θέση (Closest open location storage).** Είναι η λεγόμενη στρατηγική των συλλεκτών. Αυτή που θα ακολουθούσαν αν διάλεγαν μόνοι τους το που θα τοποθετήσουν τα προϊόντα. Η πρώτη κενή θέση που θα έβρισκαν θα ήταν και αυτή που θα επέλεγαν για να τοποθετήσουν το προϊόν. Με αυτό τον τρόπο η αποθήκη θα ήταν γεμάτη προϊόντα στα ράφια που βρίσκονται κοντά στις ράμπες εισόδου-εξόδου και άδεια στα ράφια μακριά από αυτές.
- ❖ **Full-turnover storage.** Τα προϊόντα τοποθετούνται σε θέσεις ανάλογα με την συχνότητα ζήτησης τους. Δηλαδή τα συχνά κινούμενα προϊόντα είναι αυτά που καταλαμβάνουν τις θέσεις κοντά στις ράμπες εισόδου-εξόδου και τα αργά κινούμενα τις θέσεις μακριά από αυτές. Αυτή η μέθοδος αποθήκευσης διατρέχει τον εξής κίνδυνο. Σε περίπτωση αλλαγής της ζήτησης ή ύπαρξης κάποιου διαστήματος εποχικότητας κλπ δεν είναι λειτουργική και συνεπώς πρέπει να ανασυνταχθεί πράγμα που κοστίζει χρόνο και χρήμα. Με άλλα λόγια το μειονέκτημα αυτής της στρατηγικής είναι η έλλειψη εύκολης προσαρμογής σε ιδιαίτερες καταστάσεις και συνθήκες με αποτέλεσμα την έλλειψη αποτελεσματικότητας. Με τη χρήση όμως κατάλληλων τεχνολογιών και μεθόδων πρόβλεψης, θα μπορούσε να είναι μία από τις πιο λειτουργικές στρατηγικές [12].

4. Πρόβλημα Περιοδεύοντος Πωλητή

Το πρόβλημα περιοδεύοντος πωλητή (Traveling Salesperson Problem-TSP), όπως θα αναφερθεί και στη συνέχεια, είναι ένα κομμάτι του ευρύτερου προβλήματος της παρούσας εργασίας με το οποίο θα ασχοληθούμε εκτενώς και για αυτό τον λόγο είναι χρήσιμο να γίνει μια σύντομη περιγραφή του.

Το TSP, λοιπόν, είναι ένα πολύ γνωστό και σημαντικό πρόβλημα συνδυαστικής βελτιστοποίησης. Ο στόχος είναι να βρεθεί η συντομότερη διαδρομή που κάποιος επισκέπτεται κάθε σημείο μιας συγκεκριμένης λίστας ακριβώς μια φορά και στη συνέχεια επιστρέφει στο σημείο από το οποίο ξεκίνησε. Επισήμως, το TSP μπορεί να διατυπωθεί ως εξής: οι αποστάσεις μεταξύ n πόλεων αναπαρίστανται σε έναν πίνακα D με στοιχεία d_{ij} όπου $i, j = 1, \dots, n$ και τα διαγώνια στοιχεία d_{ii} είναι μηδέν. Μια διαδρομή μπορεί να αναπαρασταθεί σαν ένα κυκλικό σύνολο σημείων π από το $\{1, 2, \dots, n\}$ όπου το $\pi(i)$ αντιπροσωπεύει το σημείο που ακολουθεί το σημείο i στην διαδρομή. Έτσι, το TSP σε αυτή την περίπτωση είναι το πρόβλημα βελτιστοποίησης που έχει να κάνει με την εύρεση του συνόλου σημείων π που ελαχιστοποιεί το μήκος της διαδρομής που ορίζεται ως το παρακάτω άθροισμα:

$$\sum_{i=1}^n d_{i\pi(i)}$$

Για να λυθεί αυτό το πρόβλημα βελτιστοποίησης πρέπει να υπολογισθεί το μήκος της διαδρομής $(n-1)!$ διανυσμάτων του συνόλου σημείων. Έτσι γίνεται εύκολα αντιληπτό πως ότι καταλήγουμε σε ένα πρόβλημα πολύ δύσκολο επιλύσιμο και που είναι γνωστό NP-complete πρόβλημα. Παρ'όλα αυτά η επίλυση ενός τέτοιου είδους προβλήματος είναι σημαντικό μέρος εφαρμογών σε διάφορες θεματικές περιοχές όπως η δρομολόγηση οχημάτων, ο προγραμματισμός μηχανημάτων, η δρομολόγηση των συλλεκτών των προϊόντων σε μια αποθήκη, η εκχώρηση συχνοτήτων σε δίκτυα επικοινωνιών κ.α. [13, 14].

Το TSP μπορεί να αναπαρασταθεί σε ένα γράφημα $G = (V, E)$, όπου V είναι το σύνολο των κορυφών (ή πόλεων) και το E είναι το σύνολο των ακμών (ή των συνδέσμων μεταξύ των πόλεων). Κάθε ακμή έχει ένα σχετικό κόστος (ή μήκος) από το i στο j , που συμβολίζεται c_{ij} . Αν το διάνυσμα x ορίζεται ως

$$x_{ij} = \begin{cases} 1 & \text{η διαδρομή πάει από το } i \text{ στο } j \\ 0 & \text{σε κάθε άλλη περίπτωση} \end{cases}$$

τότε το TSP μπορεί να διαμορφωθεί, γενικά, ως εξής [31]:

$$\begin{aligned}
& \min \sum_{i=0}^n \sum_{j \neq i, j=0}^n c_{ij} x_{ij} \\
& 0 \leq x_{ij} \leq 1 \quad i, j = 0, \dots, n \\
& u_i \in \mathbb{Z} \quad i = 0, \dots, n \\
& \sum_{i=0, i \neq j}^n x_{ij} = 1 \quad j = 0, \dots, n \\
& \sum_{j=0, i \neq j}^n x_{ij} = 1 \quad i = 0, \dots, n \\
& u_i - u_j + nx_{ij} \leq n-1 \quad 1 \leq i \neq j \leq n
\end{aligned}$$

5. Αλγόριθμοι επίλυσης TSP προβλημάτων

Επειδή, λοιπόν, όπως αναφέρθηκε και παραπάνω αν και δύσκολα τα προβλήματα αυτού του είδους είναι ιδιαίτερα σημαντικά για πολλούς λόγους, έχουν αναπτυχθεί διάφοροι τύποι αλγόριθμων που προσεγγίζουν μια ικανοποιητική αν όχι τη βέλτιστη λύση κάποιων περιορισμένων περιπτώσεων του συγκεκριμένου προβλήματος.

Αναλυτικοί Αλγόριθμοι

Κλάδου και Φράγματος

Ο αλγόριθμος Κλάδου και Φράγματος (Branch and Bound – B&B) ψάχνει ολόκληρο το σύνολο των λύσεων για ένα πρόβλημα και προσπαθεί να βρει την καλύτερη λύση. Παρόλα αυτά πολλές φορές, λόγω της εκθετικής αύξησης των πιθανών λύσεων του προβλήματος, ο αλγόριθμος δυσκολεύεται να προσδιορίσει την καλύτερη λύση, καθώς του είναι δύσκολο να διατρέξει σε κάθε μία από τις πιθανές. Η χρησιμοποίηση ορίων για τη συνάρτηση προς βελτιστοποίηση σε συνδυασμό με την τιμή της τρέχουσας βέλτιστης λύσης επιτρέπει στον αλγόριθμο να ψάχνει έμμεσα κάποια μέρη του συνόλου (υποσύνολα) στο οποίο βρίσκεται η λύση.

Αρχικά, ξεκινώντας την επίλυση, μόνο ένα υποσύνολο υπάρχει και αυτό είναι το γενικότερο σύνολο και η τρέχουσα βέλτιστη λύση σε αυτή την περίπτωση είναι το ∞ . Τα σύνολα που δεν έχουν διερευνηθεί μπορούν να παρομοιαστούν με κόμβους σε ένα δυναμικά γεννώμενο δέντρο ανίχνευσης λύσεων, το οποίο αρχικά περιέχει μόνο τη ρίζα, και κάθε επανάληψη ενός κλασσικού B&B αλγόριθμου του προσδίδει ένα παραπάνω κόμβο. Η λογική της κάθε επανάληψης απαρτίζεται από τρία βασικά στοιχεία: τον υπολογισμό των ορίων, τις διακλαδώσεις και την επιλογή του κόμβου που θα λάβει μέρος στη διαδικασία. Στο Σχήμα φαίνεται η αρχική κατάσταση και το πρώτο βήμα της διαδικασίας που περιγράφεται παραπάνω.

Η σειρά τους μπορεί να ποικίλει ανάλογα με την στρατηγική επιλογής των κόμβων που θα συμμετέχουν στη διαδικασία. Αν η επιλογή του επόμενου υποπροβλήματος βασίζεται στην περιορισμένη τιμή των υποπροβλημάτων, τότε η πρώτη ενέργεια που θα γίνει στην επανάληψη μετά την επιλογή του κόμβου θα είναι η διακλάδωση δηλαδή η υποδιαίρεση του υποσυνόλου σε δύο ή περισσότερα μικρότερα υποσύνολα που θα διερευνηθούν στις επαναλήψεις που ακολουθούν. Για κάθε ένα από αυτά, διερευνάται το αν το υποσύνολο αποτελείται από μία μοναδική λύση και αν ισχύει αυτό συγκρίνεται με την τρέχουσα βέλτιστη λύση και ο αλγόριθμος κρατάει την καλύτερη από τις δύο. Σε άλλες περιπτώσεις αν δεν ισχύουν τα παραπάνω υπολογίζεται η οριοθετημένη συνάρτηση για το υποσύνολο και συγκρίνεται με την τρέχουσα βέλτιστη λύση. Αν μπορεί να αποδειχθεί ότι το επιλεγμένο υποσύνολο δεν περιέχει την βέλτιστη λύση τότε απορρίπτεται, αλλιώς αποθηκεύεται μαζί με τους άλλους ενεργούς κόμβους μαζί με τα όριά του. Αυτή ονομάζεται “eager

strategy” για την ανάπτυξη κόμβων, καθώς τα όρια υπολογίζονται αν υπάρχουν διαθέσιμοι κόμβοι μόνο. Υπάρχουν και πολλές ακόμα στρατηγικές.

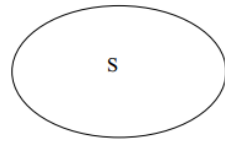
Η διερεύνηση σταματά όταν δεν υπάρχουν κομμάτια του συνόλου στο οποίο βρίσκεται η λύση που να μην έχουν διερευνηθεί και η βέλτιστη λύση είναι τελικά η τελευταία «τρέχουσα καλύτερη λύση».

Όπως γίνεται κατανοητό ένας B&B αλγόριθμος «σπάει» το κυρίως πρόβλημα σε υποπροβλήματα και προσπαθεί να τα εξερευνήσει μέχρι να μην υποδιαιρούνται άλλο. Αυτό συμβαίνει σε μία από τις παρακάτω περιπτώσεις :

- Η βέλτιστη λύση του χαλαρωμένου υποπροβλήματος ικανοποιεί τους περιορισμούς του μη χαλαρωμένου υποπροβλήματος και οι τιμές των αντίστοιχων αντικειμενικών συναρτήσεων των δύο περιπτώσεων είναι ίσες.
- Η αδυναμία εύρεσης εφικτής λύσης για το χαλαρωμένο υποπρόβλημα σημαίνει το ίδιο και για το μη χαλαρωμένο υποπρόβλημα.
- Το ανώτατο όριο που δίνεται από το χαλαρωμένο υποπρόβλημα είναι μικρότερο ή ίσο με την τιμή της αντικειμενικής συνάρτησης που αντιστοιχεί στην καλύτερη γνωστή εφικτή λύση.

Σε μια τυπική επανάληψη ο αλγόριθμος κάνει τα παρακάτω βήματα (Σχήμα 5):

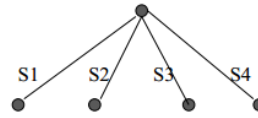
- Επιλέγει ένα φύλλο από το διακλαδωμένο δέντρο, δηλαδή ένα υποπρόβλημα που δεν έχει υποδιαιρεθεί σε άλλα μικρότερα.
- Το υποπρόβλημα χωρίζεται σε άλλα μικρότερα και προσδιορίζονται οι διάφορες χαλαρώσεις τους.
- Κάθε νέο πιο χαλαρό υποπρόβλημα επιλύεται και ελέγχεται αν ανήκει σε μία από τις προαναφερθέντες περιπτώσεις. Αν ανήκει τότε δεν μπορεί και δεν χρειάζεται να ερευνηθεί περαιτέρω. Αν πάλι δεν ανήκει αποθηκεύεται για περαιτέρω επεξεργασία και υποδιαιρέσεις σε επόμενες επαναλήψεις.
- Αν μια νέα εφικτή λύση βρεθεί και είναι καλύτερη από την τρέχουσα, τότε ακόμα και τα αποθηκευμένα κλαδιά που έχουν ανώτατο όριο μικρότερο από την τιμή της νέας εφικτής λύσης, μπορούν να διαγραφούν χωρίς παραπάνω έρευνα [16].



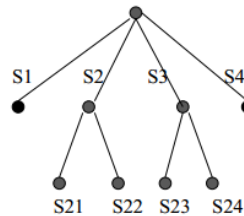
(a)



(b)



* = δεν περιέχεται βέλτιση λύση (c)



Σχήμα 5 Ο τρόπος λειτουργίας και δημιουργίας κλαδιών του B&B αλγόριθμου.

Ένας B&B αλγόριθμος για ένα πρόβλημα ελαχιστοποίησης (αντίστοιχα και για μεγιστοποίηση) αποτελείται από τρία βασικά στοιχεία:

- i. Μία οριοθετημένη συνάρτηση η οποία, για ένα υποσύνολο του συνόλου στο οποίο βρίσκεται η λύση, δίνει τη δυνατότητα εύρεσης ενός κατώτατου ορίου της καλύτερης δυνατής λύσης που μπορεί να παρθεί χρησιμοποιώντας το συγκεκριμένο υποσύνολο.
- ii. Μία στρατηγική επιλογής του υποσυνόλου που θα διερευνηθεί στην τρέχουσα επανάληψη και
- iii. Έναν κανόνα «διακλάδωσης» που μπορεί να εφαρμοστεί αν ένα υποσύνολο μετά τη διερεύνησή του δεν μπορεί να αποκλειστεί, υποδιαιρώντας αυτό το υποσύνολο σε δύο ή περισσότερα υποσύνολα και εξετάζοντας τα στις επόμενες επαναλήψεις [16, 17].

Χαλάρωση

Η γενική ιδέα της χαλάρωσης (relaxation) είναι η αντικατάσταση ενός περιορισμού από έναν περιορισμό λιγότερο αυστηρό. Πιο επίσημα η έννοια της χαλάρωσης είναι η εξής: Ας υποθέσουμε ότι κάποιο σύνολο μεταβλητών παίρνει τιμές από ένα σύνολο X , τότε αυτό θα αντικατασταθεί από ένα σύνολο Y για το οποίο θα ισχύει $X \subseteq Y$ [17].

Κλάδου και Τομής (Branch and Cut)

Αυτή η μέθοδος είναι ένας τρόπος μετασχηματισμού της Branch and Bound στρατηγικής, η οποία επιδιώκει την ενίσχυση της χαλάρωσης του γραμμικού προγραμματισμού (LPR) ενός IP (integer problem) και αυτό το επιτυγχάνει με την εισαγωγή κάποιου είδους νέων «περιορισμών» λεγόμενους και ως επίπεδα τομής (cutting planes). Και γενικότερα ισχύει

$$\text{Branch \& Cut} = \text{Branch \& Bound} + \text{Cutting Planes}$$

Η μέθοδος Branch and Cut αλγορίθμων, λοιπόν, είναι μια μέθοδος επίλυσης που χρησιμοποιείται κυρίως για την επίλυση προβλημάτων ακέραιου(και συνδυασμένου με γραμμικό) προγραμματισμού και εξασφαλίζουν την εύρεση της βέλτιστης λύσης του εκάστοτε προβλήματος. Υπάρχουν περιπτώσεις στις οποίες, βέβαια, δεν μπορεί να δώσει τη βέλτιστη λύση, αλλά σε συνδυασμό με κάποιους ευρετικούς αλγόριθμους προσεγγίζει αρκετά την λύση και στη περίπτωση που δεν επιτευχθεί η βέλτιστη, μπορεί και αναγνωρίζει το πόσο μακριά βρισκόμαστε από αυτή [15, 18].

Αυτή η μέθοδος μπορεί να περιγραφεί πολύ απλά στη γενική της μορφή με 7 απλά βήματα :

- i. **Εκκίνηση.** Συμβολίζουμε το αρχικό πρόβλημα ακέραιου προγραμματισμού ως ILP^0 και θέτουμε τους ενεργούς κόμβους $L = \{ILP^0\}$. Θέτουμε, επίσης, το άνω όριο $z = +\infty$. Ορισμός $z = -\infty$ για το ένα πρόβλημα με $L = \emptyset$.
- ii. **Λύση.** Αν $L = \emptyset$, τότε η λύση x^* που έδωσε η επικείμενη αντικειμενική τιμή του z είναι η βέλτιστη. Αν δεν υπάρχει τέτοιο x (δηλαδή $z = +\infty$), τότε το ILP δεν μπορεί να λυθεί.
- iii. **Επιλογή προβλήματος.** Επιλογή και διαγραφή ενός προβλήματος ILP^l από το L .
- iv. **Χαλάρωση.** Εύρεση της χαλάρωσης του γραμμικού προγραμματισμού ILP^l . Αν είναι αδύνατο, θέτουμε $z = +\infty$ και πάμε στο βήμα 6. Ας δεχτούμε ότι το z συμβολίζει την βέλτιστη αντικειμενική τιμή της χαλάρωσης αν είναι πεπερασμένη και το x^R μια βέλτιστη λύση, σε κάθε άλλη περίπτωση θέτουμε $z = -\infty$.
- v. **Πρόσθεση επιπέδων cutting.** Αν είναι επιθυμητό ψάχνουμε για επίπεδα cutting που παραβιάζονται από το x^R . Αν βρεθεί κάποιος, το προσθέτουμε στη χαλάρωση και επιστρέφουμε στο βήμα 4.
- vi. **Fathoming and Pruning (εμβάθυνση και ξεσκαρτάρισμα).**

Αν ισχύει $z \geq z^*$ τότε επιστρέφουμε στο βήμα ii.

Αν ισχύει $\underline{z} < z^R$ και το x^R είναι εφικτή ακέραια λύση, τότε \underline{z} και διαγράφουμε από το L όλα τα προβλήματα με $\underline{z} \geq z^R$ και γυρνάμε πίσω στο βήμα ii.

Διαχωρισμός. Ας δεχτούμε ότι $\{S^j\}_{j=1}^{j=k}$ είναι ένα μέρος του συνόλου των περιορισμών S^j του προβλήματος ILP^j . Προσθέτουμε τα προβλήματα $\{ILP^j\}_{j=1}^{j=k}$ στο L , όπου το ILP^j είναι το ILP^j στην εφικτή περιοχή περιορισμένη στο S^j και το \underline{z}^j για $j = 1, 2, \dots, k$ το θέτουμε ίδιο με την τιμή του \underline{z} για το αρχικό πρόβλημα I . Ύστερα γυρνάμε και πάλι στο βήμα ii [18].

Κλάδου και Αξιολόγησης

Οι αλγόριθμοι αυτού του τύπου (Branch and Price) είναι, επίσης, συμπληρωματικοί του Branch and Bound και είναι ουσιαστικά ένας μετασχηματισμός του, εισάγοντας νέες κολώνες ή αλλιώς μεταβλητές, ώστε το πρόβλημα προς επίλυση να γίνει απλούστερο και επιλύσιμο.

Ας υποθέσουμε, τώρα, ως κύριο/γενικό πρόβλημα ακέραιου προγραμματισμού με το οποίο θα ασχοληθούμε το εξής:

$$\begin{aligned} & \text{Min } \tilde{c}^* \lambda \\ \text{s.t.} & \\ & \tilde{C}^* \lambda \geq \delta \\ & \sum_{k=1}^t \lambda_k = 1, \\ & \lambda \in \{0, 1\}^t \end{aligned}$$

Άρα το συγκεκριμένο πρόβλημα με χαλάρωση διαμορφώνεται ως εξής [19, 20]:

$$\begin{aligned} & \text{Min } \tilde{c}^* \lambda \\ \text{s.t.} & \\ & \tilde{C}^* \lambda \geq \delta \\ & \sum_{k=1}^t \lambda_k = 1, \\ & \lambda \in \mathbb{R}_+^t \end{aligned}$$

όπου γίνεται προφανής η πολυπλοκότητα του και επομένως η αδυναμία επίλυσής του με μια απλή μέθοδο επίλυσης. Άρα χρειάζεται κάτι πιο εξειδικευμένο και αυτό μας το δίνει η μέθοδος Branch and Bound σε συνδυασμό με την Branch and Price, η οποία μπορεί να εφαρμοστεί σε ένα γενικό πρόβλημα όπως αυτό που περιγράφηκε παραπάνω με τον παρακάτω τρόπο [20]:

- i. Ξεκινάμε με ένα περιορισμένο κύριο πρόβλημα (RMP), το οποίο είναι το αρχικό μας πρόβλημα που τώρα θα περιλαμβάνει μικρότερο αριθμό στηλών.
- ii. Επιλύουμε το πιο απλό πρόβλημα που έχει δημιουργηθεί με κάποια απλή μέθοδο (π.χ. Simplex).
- iii. Ελέγχουμε αν κάποια από τις στήλες που δεν περιέχεται στο RMP έχει αρνητικό μειωμένο κόστος.
- iv. Αν δεν υπάρχει κάποια τέτοια στήλη σταματάμε. Έχει βρεθεί η βέλτιστη λύση του κύριου προβλήματος LP με την χαλάρωση του.
- v. Σε κάθε άλλη περίπτωση προσθέτουμε μία παραπάνω κολώνα στο RMP.
- vi. Ξαναβρίσκουμε μια βέλτιστη λύση στο πρόβλημα με κάποια απλή μέθοδο και γυρνάμε στο βήμα iii.

Το πρόβλημα που έχει να κάνει με τον έλεγχο του αν πρέπει να προστεθεί μια παραπάνω στήλη ή όχι ονομάζεται **pricing problem**.

Από τη θεωρία της δυικότητας μπορεί να επιλυθεί αυτό το πρόβλημα, λύνοντας το 0-1 LP:

$$\text{Min}\{(c-C^T*\pi)*x : x \text{ Min}\{(c-C^T*\pi)*x : xP_i^l\},$$

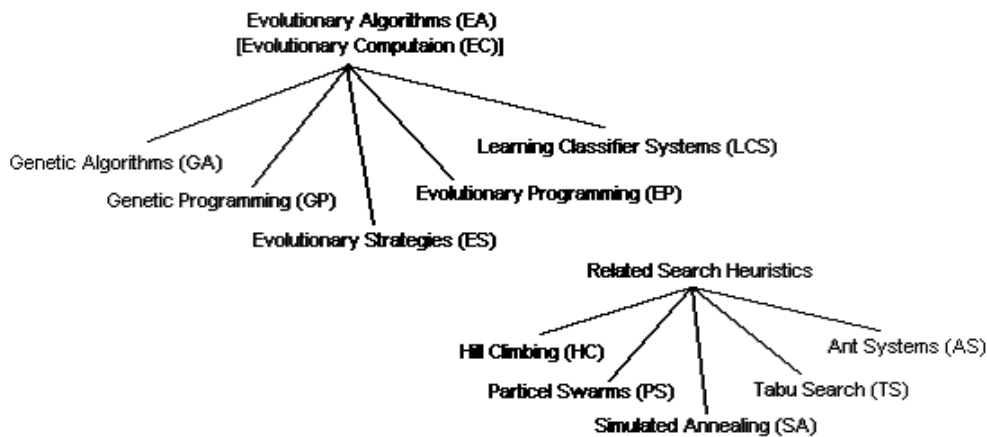
Όπου π είναι το διάνυσμα των δυαδικών τιμών για τους περιορισμούς [19]:

$$\tilde{C} * \lambda \geq \delta$$

Ευρετικοί αλγόριθμοι

Είναι φανερό από την ανάλυση που έγινε παραπάνω ότι οι αναλυτικοί αλγόριθμοι δίνουν λύση σε ορισμένες περιπτώσεις TSP. Τι γίνεται όμως όταν ακόμα και ο συνδυασμός κάποιων αλγόριθμων δεν μπορεί να δώσει κάποια καλή προσέγγιση της βέλτιστης λύσης; Τι γίνεται δηλαδή όταν το πρόβλημα που έχουμε να αντιμετωπίσουμε δεν είναι μέχρι και μεσαίας, αλλά μεγαλύτερης κλίμακας; Τι γίνεται όταν όλες αυτές οι μέθοδοι για να εφαρμοστούν χρειάζονται υπερβολικά μεγάλο χρόνο και υπερβολικά μεγάλη μνήμη υπολογιστή; Τότε, λοιπόν, όταν συμβαίνουν όλα αυτά καταφεύγουμε στους ευρετικούς αλγόριθμους, οι οποίοι μπορεί να μην εγγυώνται την εύρεση της βέλτιστης λύσης, αλλά μπορούν να δώσουν μια ικανοποιητική λύση σε

πολλά προβλήματα μεγάλης κλίμακας. Οι ευρετικές μέθοδοι επίλυσης προβλήματος κατά γενική ομολογία είναι σχετικά καλές και σχετικά γρήγορες λαμβάνοντας υπ' όψιν τα προβλήματα που αντιμετωπίζονται με αυτές. Παρακάτω θα ασχοληθούμε με κάποιες σημαντικές ευρετικές μεθόδους ξεχωριστά (Σχήμα 6) [21].



Σχήμα 6 Οι διάφορες κατηγορίες ευρετικών αλγορίθμων.

Γενετικοί Αλγόριθμοι

Το όνομα αυτής της οικογένειας αλγορίθμων (Genetic Algorithms) προέρχεται από τον τρόπο με τον οποίο μιμείται την διαδικασία ανάπτυξης/εξέλιξης των οργανισμών και η λύση που προκύπτει από έναν τέτοιο αλγόριθμο παρομοιάζεται με μια γενετική αλυσίδα. Χαρακτηριστικά της μεθόδου αυτής είναι η επιβίωση του ισχυρότερου μηχανισμού στον οποίο οι δυνητικές λύσεις ενός πληθυσμού ανταγωνίζονται η μία την άλλη, καθώς και ο ανασυνδυασμός λύσεων που προκύπτουν από μια διαδικασία ζευγαρώματος και τυχαίων αλλαγών. Το συναρπαστικό χαρακτηριστικό αυτού του αλγόριθμου είναι ότι μπορεί να παράγει όλο και καλύτερες λύσεις, χωρίς την προϋπόθεση της βαριάς κατανόησης του κυρίως προβλήματος! Τα προβλήματα στα οποία μπορούν να εφαρμοστούν γενετικοί αλγόριθμοι πρέπει να έχουν τα δύο παρακάτω χαρακτηριστικά:

- Μια λύση μπορεί να εκφραστεί σαν «αλυσίδα» ή αλλιώς γραμμή (string).
- Υπάρχει μια τιμή που εκφράζει την αξία αυτής της γραμμής.

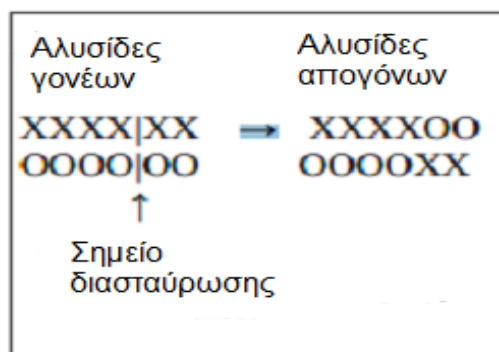
Οι γενετικοί αλγόριθμοι έχουν κάποια πλεονεκτήματα που τους κάνουν να ξεχωρίζουν:

- Είναι εύκολο το κομμάτι του προγραμματισμού τους.

- Δουλεύουν άμεσα με ολοκληρωμένες λύσεις. Δεν χρειάζεται, δηλαδή, όπως στην branch and bound να γίνονται εκτιμήσεις ή να υπάρχουν περιορισμένες συναρτήσεις.

Υπάρχουν τρεις βασικοί τελεστές αυτής της μεθόδου οι οποίοι συμμετέχουν και επηρεάζουν πολύ την όλη διαδικασία επίλυσης:

- I. **Ο Τελεστής Αναπαραγωγής (The Reproduction Operator).** Η αναπαραγωγή είναι ισοδύναμη με τον «διαγωνισμό» της επιβίωσης του ισχυρότερου. Καθορίζει, όχι μόνο, τις λύσεις που επιβιώνουν, αλλά και πόσα αντίγραφα της κάθε λύσης θα φτιαχτούν. Αυτό βοηθάει, όπως θα δούμε και αργότερα, την διαδικασία διασταύρωσης. Η πιθανότητα επιβίωσης μιας λύσης είναι ανάλογη με την αξία της λύσης αυτής, γνωστή και ως fitness. Με αυτό τον τρόπο οι πιο ισχυρές λύσεις (most fit) είναι αυτές που διαλέγονται (επιβιώνουν) τις περισσότερες φορές και οι αδύναμες λύσεις (most unfit) είναι αυτές που δεν λαμβάνονται υπ'όψιν συνήθως (πεθαίνουν) και έτσι το πρόβλημα δεν καταναλώνει πολύ χρόνο και μνήμη του PC. Μετά το πέρας αυτής της διαδικασίας, λοιπόν, υπάρχει ένας ενδιάμεσος πληθυσμός, γνωστός και ως Mating pool (χώρος ζευγαρώματος), ο οποίος είναι έτοιμος να αναμιχθεί και να σμίξει, σύμφωνα με τη διαδικασία αναπαραγωγής των παιδιών όπου μοιράζονται κάποια γενετικά χαρακτηριστικά των δύο γονέων. Αυτή την λειτουργία την υπηρετεί ο crossover operator.
- II. **Ο Τελεστής Διασταύρωσης (The Crossover Operator).** Κατά την διαδικασία αυτή, δύο αρχικές(γονείς) λύσεις-γραμμές από το mating pool συνδυάζονται για να δημιουργήσουν δύο νέες(παιδιά) λύσεις. Έτσι, λοιπόν, ένας νέος πληθυσμός είναι σχεδόν έτοιμος (Σχήμα 7).



Σχήμα 7 Τρόπος λειτουργίας του Crossover Operator.

- III. **Ο Τελεστής Μετάλλαξης (The Mutation Operator).** Είναι ο τελευταίος τελεστής που θα διαμορφώσει τελικά τον νέο πληθυσμό. Χρησιμοποιείται για να αλλάζει τυχαία τις τιμές κάποιων θέσεων σε κάποιες γραμμές και βασίζεται σε μια παράμετρο που καθορίζει το επίπεδο ομοιότητας. Μία συνήθης περίπτωση είναι η αλλαγή σε 1 στις

1000 περιπτώσεις ομοιότητας. Η ομοιότητα είναι ένα πολύ σημαντικό κομμάτι του αλγόριθμου και η τιμή της παραμέτρου που την εκφράζει ασκεί μεγάλη επιρροή στην ποιότητα της τελικής λύσης.

Η γενική διαδικασία επίλυσης ενός προβλήματος με γενετικό αλγόριθμο μπορεί να συνοψιστεί στα παρακάτω :

- i. Προετοιμασία για εκκίνηση. Επιλογή του μεγέθους n του πληθυσμού και της τιμής της παραμέτρου ομοιότητας. Επιλογή των operators και των συνθηκών τερματισμού.
- ii. Παραγωγή ενός αρχικού πληθυσμού με τυχαίο τρόπο και υπολογισμός της τιμής της ισχύος (fitness) κάθε γραμμής. Θέτουμε ως τρέχουσα λύση την λύση με την μεγαλύτερη ισχύ από τον αρχικό πληθυσμό.
- iii. Εφαρμογή του Τελεστή Αναπαραγωγής στον πληθυσμό για την παραγωγή ενός mating pool μεγέθους n .
- iv. Εφαρμογή του Τελεστή Διασταύρωσης στις γραμμές που προκύπτουν στο mating pool για την παραγωγή ενός νέου δοκιμαστικού πληθυσμού μεγέθους n .
- v. Εφαρμογή του Τελεστή Μετάλλαξης στον δοκιμαστικό πληθυσμό για την παραγωγή του νέου τελικού πληθυσμού. Υπολογισμός των τιμών ισχύων των διαφόρων λύσεων-γραμμών και ενημέρωση της τιμής της τρέχουσας λύσης αν βρεθεί κάποια λύση με μεγαλύτερη ισχύ από αυτήν της αρχικής.
- vi. Αν συναντώνται οι συνθήκες τερματισμού τότε η τελική λύση είναι η τρέχουσα, αλλιώς επιστρέφουμε στο βήμα iii.

Υπάρχουν διάφορες συνθήκες τερματισμών με βάση τις οποίες μπορούμε να τρέξουμε τον αλγόριθμο. Μία από αυτές είναι ο τερματισμός του μετά από κάποιο προκαθορισμένο αριθμό δημιουργηθέντων πληθυσμών. Μία άλλη είναι ο τερματισμός του μετά την παρατήρηση μηδανικών αλλαγών μεταξύ των γενεών [21].

Μέθοδος Προσομοιωμένης Ανόπτωσης

Αυτή η μέθοδος (Simulated Annealing - SA) είναι μια γνωστή ευρετική μέθοδος που χρησιμοποιείται για την επίλυση διακριτών , αλλά και συνεχών προβλημάτων. Αναπτύχθηκε πριν τους γενετικούς αλγόριθμους και σταδιακά αντικαθίσταται από αυτούς , αλλά ακόμα παραμένει ευρεία η χρήση της. Είναι βασισμένη σε μια αναλογία της θερμικής επεξεργασίας των μετάλλων (γνωστή και ως annealing). Όταν τα μέταλλα περνάνε προσεκτικά από την διαδικασία του annealing, συνήθως με την ακριβή ρύθμιση και έλεγχο της διαδικασίας ψύξης, κάποιες πολύ επιθυμητές ιδιότητες (όπως σκληρότητα και ελαστικότητα) μπορούν να επιτευχθούν.

Στη βελτιστοποίηση με χρήση της μεθόδου SA, όταν η παράμετρος της «θερμοκρασίας» είναι υψηλή, υπάρχει ένα μεγάλο εύρος πιθανών λύσεων

στο οποίο ταλαντεύεται η λύση του προβλήματος, ενώ όσο η παράμετρος αυτή κατεβαίνει, τόσο μικρότερο γίνεται και το εύρος και τελικά βρίσκεται η τελική λύση (που καταλήγει να παραμένει σε μια «παγωμένη» κατάσταση). Αυτό έχει σαν αποτέλεσμα την δυνατότητα να ξεφεύγει η λύση από τοπικά βέλτιστα σημεία κατά τη διάρκεια της διαδικασίας όπου επικρατεί υψηλή θερμοκρασία.

Παρακάτω παρουσιάζεται στη γενική του μορφή (με βασικά βήματα) η μεθοδολογία για την επίλυση ενός προβλήματος ελαχιστοποίησης κόστους:

- i. Προετοιμασία για έναρξη. Εύρεση αρχικής λύσης S , πιθανότατα τυχαία . Επιλογή μιας αρχικής υψηλής θερμοκρασίας $T > 0$. Επιλογή μιας τιμής r , που είναι η παράμετρος που εκφράζει τον ρυθμό ψύξης.
- ii. Επιλογή μιας κοντινής τιμής του S που θα την ονομάζουμε S' .
- iii. Υπολογισμός της διαφοράς των κοστών $\Delta = cost(S') - cost(S)$
- iv. Απόφαση για την αποδοχή ή μη της νέας λύσης. Αν $\Delta \leq 0$ τότε η S' είναι καλύτερη η ίση της S και θέτουμε $S = S'$, αλλιώς η S' είναι χειρότερη της S και θέτουμε $S = S'$ με πιθανότητα $e^{-\Delta/T}$.
- v. Αν οι συνθήκες τερματισμού εμφανιστούν τότε βγαίνουμε από τον αλγόριθμο με την S να είναι η τελική λύση, αλλιώς μειώνουμε την αρχική θερμοκρασία θέτοντας $T = r * T$ και γυρνάμε πίσω στο βήμα ii.

Μια απλή συνθήκη τερματισμού είναι το πάγωμα της S , δηλαδή για μεγάλο αριθμό επαναλήψεων να μην αλλάζει η τιμή της [21, 28].

Αυτή η μέθοδος επίλυσης αλγορίθμων είναι ιδιαίτερα σημαντική, καθώς χρησιμοποιείται παραλλαγή τους για την επίλυση του δικού μας προβλήματος.

Μέθοδος Απαγορευμένης Έρευνας

Η Μέθοδος Απαγορευμένης Έρευνας (Tabu Search - TS) είναι μια μετα-ευρετική μέθοδος που οδηγεί μια τοπική ευρετική διαδικασία έρευνας στην εξερεύνηση του διαστήματος των λύσεων πέρα των τοπικών βέλτιστων. Ένα από τα κύρια χαρακτηριστικά της είναι η χρήση της προσαρμοστικής μνήμης (adaptive memory) η οποία δημιουργεί μια πιο ευέλικτη συμπεριφορά έρευνας. Η TS χρησιμοποιεί την "Attributive memory", κυρίως, για λόγους καθοδήγησης. Χωρίζεται σε δύο κατηγορίες την Frequency based και την Recency based memory, των οποίων οι λειτουργία είναι προφανής από το όνομα που τους έχει δοθεί. Η μνήμη αυτή αντί να καταγράφει ολόκληρες λύσεις, καταγράφει μόνο μερικά σημαντικά χαρακτηριστικά. Έτσι εντοπίζονται οι αλλαγές που γίνονται σε κάποια σημαντικά για τον αλγόριθμο χαρακτηριστικά από λύση σε λύση. Αυτά όλα γίνονται με σκοπό την καθοδήγηση όπως προαναφέρθηκε, δηλαδή την επιλογή κινήσεων για την εύρεση τελικά της βέλτιστης λύσης. Το δεύτερο κύριο χαρακτηριστικό της που μαζί με το πρώτο την κάνει ξεχωριστή είναι οι ευαίσθητες στρατηγικές

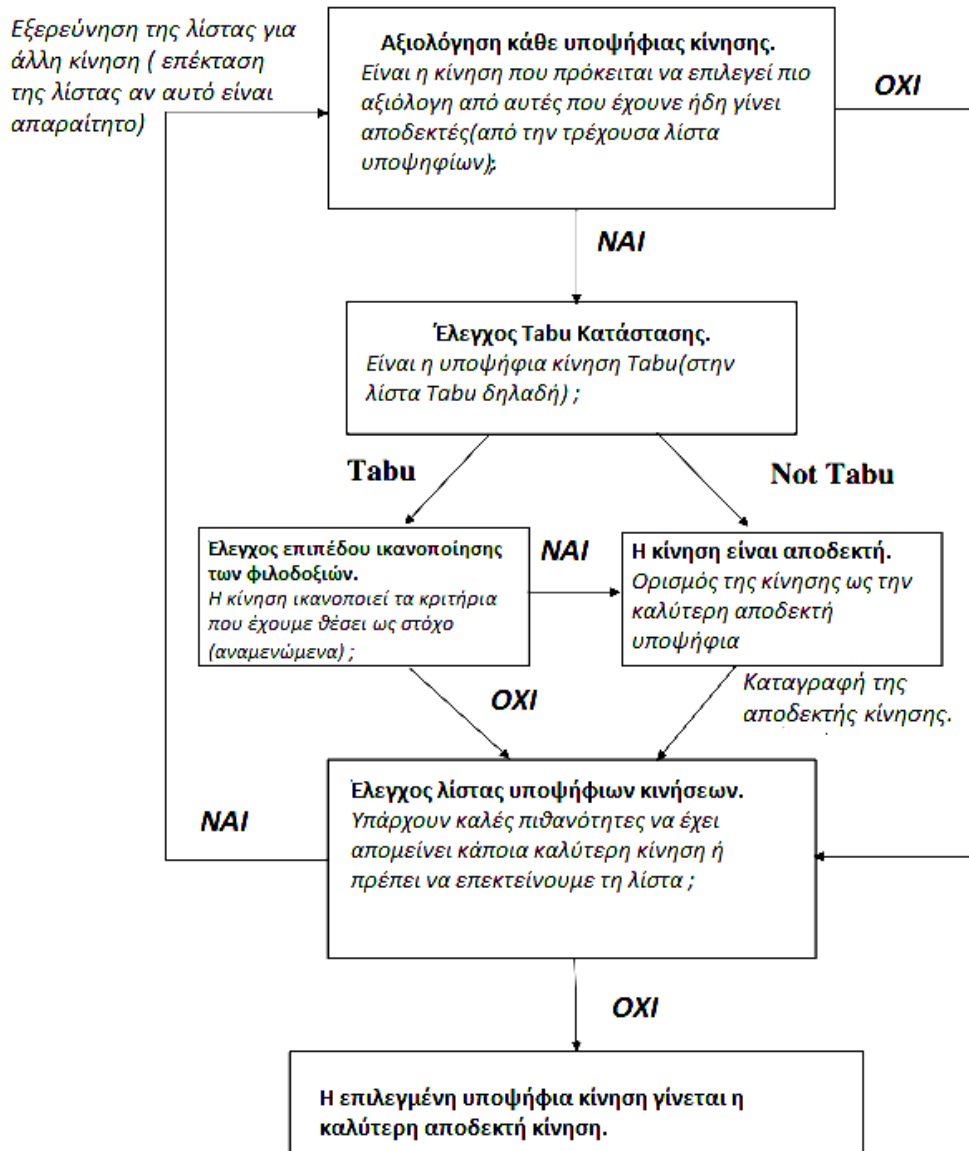
εξερεύνησης. Αυτή η μέθοδος συνδυάζεται με στρατηγικές που βασίζονται στην μνήμη και ο συνδυασμός τους δίνει σημαντικά αποτελέσματα [22].

Tabu List είναι η λίστα στην οποία περιέχονται σημεία-λύσεις ή κινήσεις που πρέπει να αποφευχθούν κατά τη διαδικασία επίλυσης με τη βοήθεια του αλγόριθμου. Αυτή η λίστα ενημερώνεται διαρκώς από στοιχεία της βραχυπρόθεσμης μνήμης του υπολογιστή.

Η γενική μεθοδολογία επίλυσης με την Tabu Search μπορεί να περιγραφεί ως εξής:

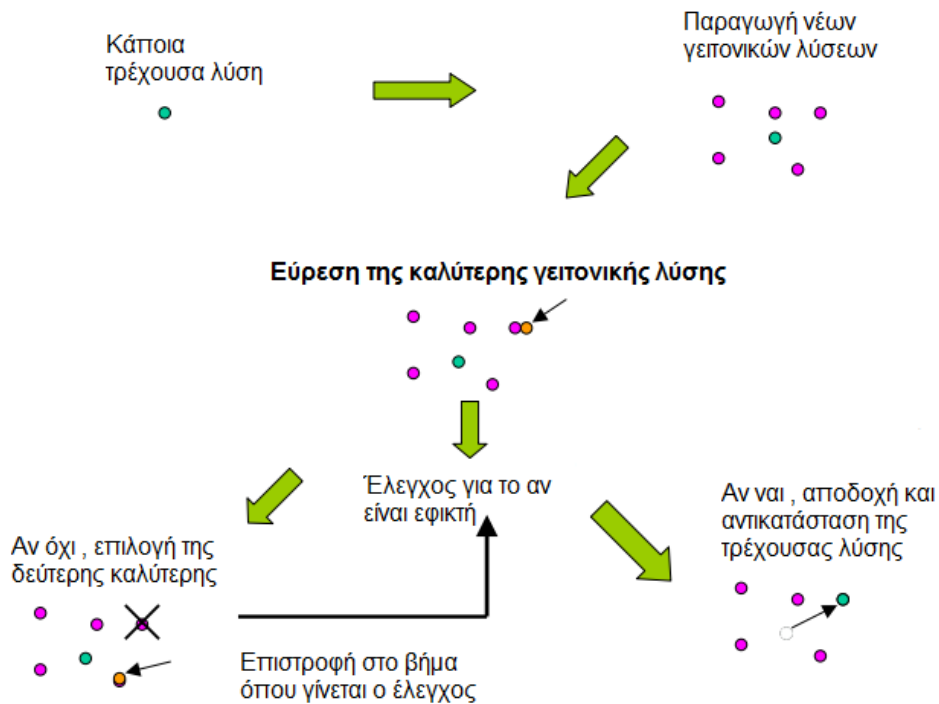
- i. Αρχίζουμε με μια αρχική τρέχουσα λύση. Αυτή μπορεί να παρθεί από την εκκίνηση ή κάποιο ενδιάμεσο ή μακροπρόθεσμο δεδομένο μνήμης.
- ii. Δημιουργία μιας λίστας με υποψήφιες κινήσεις. Αν εφαρμοστούν τότε γεννούν μια νέα γειτονική λύση από την τρέχουσα λύση.
- iii. Επιλογή της καλύτερης αποδεκτής υποψήφιας. Ορισμός της λύσης που παίρνεται ως την νέα τρέχουσα λύση. Καταγραφή της λύσης αυτής σαν νέα βέλτιστη λύση αν είναι καλύτερη από την προηγούμενη.
- iv. Συνθήκες τερματισμού. Τερματισμός αν ένας συγκεκριμένος αριθμός επαναλήψεων έχει γίνει συνολικά ή από την στιγμή που η τελευταία βέλτιστη λύση έχει βρεθεί.
- v. Αν δεν συναντώνται συνθήκες τερματισμού τότε ενημερώνουμε τις συνθήκες αποδοχής και επιστρέφουμε στο Βήμα ii.

Όσον αφορά το Βήμα iii αυτής της γενικής μεθοδολογίας περιγράφεται παρακάτω με διάγραμμα αναλυτικά το πώς γίνεται η επιλογή της καλύτερης αποδεκτής υποψήφιας λύσης (Σχήματα 8, 9) [23]:



Σχήμα 8 Τρόπος επιλογής της βέλτιστης αποδεκτής υποψήφιας λύσης. (Αλγοριθμική απεικόνιση)

Η σχηματικά μπορεί να αναπαρασταθεί ως εξής :



Σχήμα 9 Τρόπος επιλογής της βέλτιστης αποδεκτής υποψήφιας λύσης. (Διαγραμματική απεικόνιση)

Εξελικτικοί Αλγόριθμοι

Οι Εξελικτικοί Αλγόριθμοι (Evolutionary algorithms-EAs) είναι ένα σύνολο ευρετικών αλγόριθμων που δίνουν λύσεις σε προβλήματα βελτιστοποίησης χρησιμοποιώντας κάποιες πτυχές της φυσικής εξέλιξης. Ένα από τα θετικά τους είναι ότι είναι εύχρηστοι, καθώς μπορούν να εφαρμοστούν σε διάφορα είδη προβλημάτων, χωρίς κάποιες ιδιαίτερες απαιτήσεις για εφαρμογή και αυτό συμβαίνει λόγω της μεγάλης ποικιλίας που υπάρχει στο σύνολο αυτό των αλγόριθμων. Έτσι, για κάθε είδος προβλήματος βρίσκεται κάποιος αλγόριθμος που να ταιριάζει και να το επιλύει σύμφωνα με το είδος των δεδομένων που πρόκειται να επεξεργαστούν, την αναπαράσταση της λύσης και το εύρος του διαστήματος της έρευνας.

Η βασική ιδέα των EAs είναι η εξής: αν μόνο τα στοιχεία ενός πληθυσμού που υπακούν σε κάποια κριτήρια επιλογής αναπαράγονται και τα άλλα στοιχεία του πληθυσμού απορρίπτονται, τότε ο πληθυσμός θα συγκλίνει στα στοιχεία που έχουν κάποια συγκεκριμένα επιθυμητά χαρακτηριστικά. Ισχύει η λογική

της «επιβίωσης του ισχυρότερου» (ή Survival of the fittest) στην οποία βασίζεται η ανάπτυξη των πληθυσμών, των οποίων τα στοιχεία θα πρέπει να αντιπροσωπεύουν μια λύση του προβλήματος βελτιστοποίησης προς επίλυση. “Survival of the fittest” σημαίνει ότι μεγαλύτερη πιθανότητα επιβίωσης-επιλογής κατά τη δημιουργία ενός νέου πληθυσμού έχουν τα στοιχεία αυτά που φέρουν κάποιο ζητούμενο χαρακτηριστικό και μάλιστα το μεταφέρουν και στα νέα στοιχεία του νέου πληθυσμού που δημιουργείται. Η πιθανότητα επιλογής με τον τρόπο δημιουργίας αυτών των αλγόριθμων είναι πάντα ανάλογη της ποιότητας (ή fitness) της λύσης που εκπροσωπεί [29].

Ένας γενικός τρόπος περιγραφής των EAs είναι ο εξής :

Ξεκινώντας, δίνεται τυχαίος πληθυσμός $A(s=0)$

επανάληψη

Αξιολόγηση της ισχυρότητας όλων των a_i του $A(s)$.

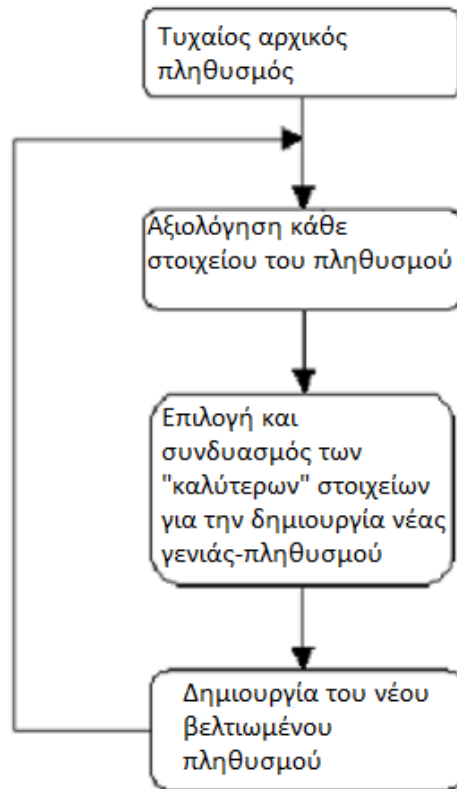
Επιλογή των ισχυρότερων a_i ως προγόνους $B(s)$ από τον $A(s)$

Παραγωγή νέων απογόνων $C(s)$ από τον $B(s)$

$A(s+1) = C(s)$

Μέχρι να εμφανιστεί συνθήκη τερματισμού

Ή αλλιώς σε μορφή διαγράμματος (Σχήμα 10) [24]:



Σχήμα 10 Τρόπος λειτουργίας των EAs.

Αλγόριθμος Επίλυσης Προβλημάτων Περιοδούντος Πωλητή σε Αποθήκες Παραλληλόγραμμου Σχήματος

Αντίθετα με όλους τους αλγόριθμους που περιγράφηκαν παραπάνω, υπάρχει ένας αλγόριθμος, ο οποίος δίνει πάντα την βέλτιστη λύση για συγκεκριμένη κατηγορία TSP. Ο αλγόριθμος αυτός είναι ο “Ratliff and Rosenthal”(και περιγράφεται αναλυτικά στην 2^η ενότητα του παραρτήματος) [32] και δίνει λύση για αποθήκες παραλληλόγραμμου σχήματος που έχουν μόνο 1 block, όπως αυτή του Σχήματος 19. Παρακάτω αυτό θα γίνει φανερό και στην πράξη, αφού θα εφαρμοστεί στην επίλυση του προβλήματος της παρούσας εργασίας.

6. Περιγραφή του γενικού προβλήματος εύρεσης της βέλτιστης διάταξης προϊόντων σε αποθήκη

Το πρόβλημα με το οποίο ασχολούμαστε είναι μια κλασική περίπτωση Προβλήματος Διεπίπεδου Προγραμματισμού, όπου οι αντικειμενικές συναρτήσεις είναι δύο και αντίστοιχα και οι μεταβλητές που συμμετέχουν ανήκουν σε δύο σύνολα. Πιο συγκεκριμένα, επειδή όλες οι μεταβλητές είναι ακέραιες και κυρίως δυαδικές, το πρόβλημα κατατάσσεται στην κατηγορία του Προβλημάτων Ακέραιου Διεπίπεδου Προγραμματισμού. Η κύρια αντικειμενική συνάρτηση είναι η συνολική απόσταση που διανύεται για να ολοκληρωθούν όλες οι παραγγελίες και η μεταβλητή που διαφοροποιείται για να επιτευχθεί η βελτιστοποίηση της είναι η διάταξη των προϊόντων στην αποθήκη. Η δευτερεύουσα αντικειμενική συνάρτηση του προβλήματος είναι η απόσταση που διανύεται για την ολοκλήρωση της κάθε παραγγελίας. Η μεταβλητή που επηρεάζει αυτή την απόσταση είναι η διαδρομή η οποία επιλέγεται από τους συλλέκτες για να ολοκληρώσουν την κάθε παραγγελία. Συνεπώς, τα αποτελέσματα που θα παραχθούν από τον αλγόριθμο θα είναι η προτεινόμενη βέλτιστη διαδρομή για τους συλλέκτες και μια προτεινόμενη διάταξη των προϊόντων στον χώρο της αποθήκης.

Παρακάτω περιγράφεται το γενικό μαθηματικό μοντέλο με το οποίο μπορεί να περιγραφεί το πρόβλημα μας:

$$\min D_{total} = \sum_p D_p,$$

$$s.t.: d_{i,j} = f(x_{k,n})$$

$$\min D_p = \sum_{i,j} y_{ijp} d_{i,j}$$

$$s.t.: \text{περιορισμοί TSP}$$

p : παραγγελία = 1,...,110

$x_{k,n}$: κωδικός k στην θέση n

$y_{i,j,p}$: για την παραγγελία p αν πάμε από τον κωδικό i στον κωδικό j

$d_{i,j}$: η απόσταση από τον κωδικό i στον κωδικό j

- Περιορισμοί TSP: όπως αυτοί περιγράφονται στην 4^η ενότητα
- Η συνάρτηση f που λαμβάνει μέρος στο παραπάνω μοντέλο είναι γραμμική
- $x, y \in \{0,1\}$

Για το πρωτεύον πρόβλημα, το σύνολο των δυνατών λύσεων είναι n ανά k , διατάξεις, όπου n ο αριθμός των θέσεων στην αποθήκη και k είναι οι κωδικοί των προϊόντων. Οι k -άδες αποτελούνται από διατεταγμένα, διαφορετικά στοιχεία. Το σύνολο, λοιπόν, των δυνατών λύσεων στο πρόβλημα είναι:

$${}_{(n)}_k = \frac{n!}{(n-k)!}.$$

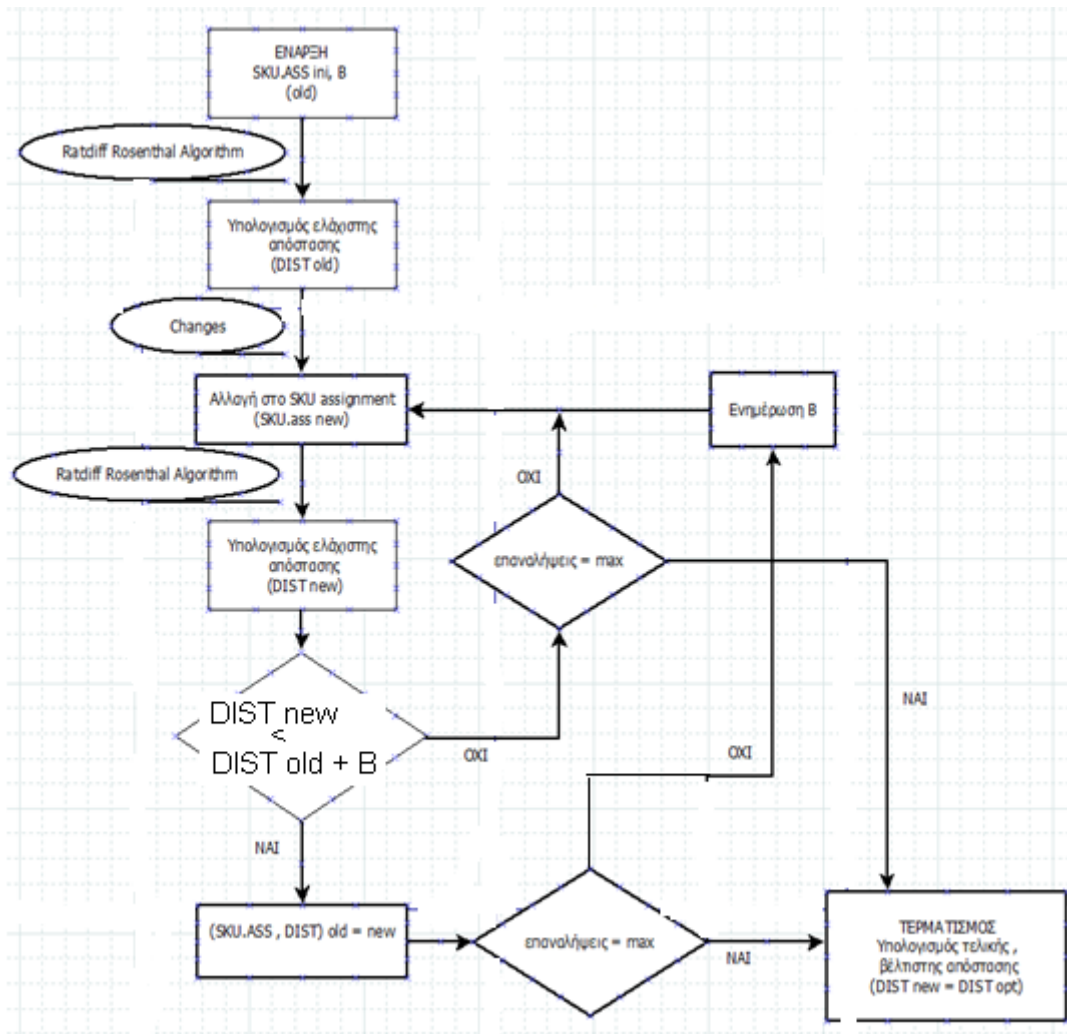
Αυτό δικαιολογεί την επιλογή, που έγινε, για χρήση ευρετικού αντί για αναλυτικού (π.χ. Branch and Bound) αλγόριθμου για την επίλυση του προβλήματος.

7. Γενική περιγραφή του αλγορίθμου επίλυσης

Ο αλγόριθμος που χρησιμοποιήθηκε αποτελείται από δύο μέρη:

- Ελαχιστοποίηση της συνολικής απόστασης που διανύεται για τις παραγγελίες, με αλλαγή στη διάταξη των προϊόντων στο χώρο.
- Ελαχιστοποίηση της απόστασης που διανύουν οι συλλέκτες των προϊόντων για κάθε παραγγελία, με επιλογή της κατάλληλης διαδρομής (Βελτιστοποίηση Συλλογής Προϊόντων).

Το λογικό διάγραμμα με το οποίο μπορούν συνοπτικά να περιγραφούν τα βήματα του αλγορίθμου φαίνεται στο Σχήμα 11:

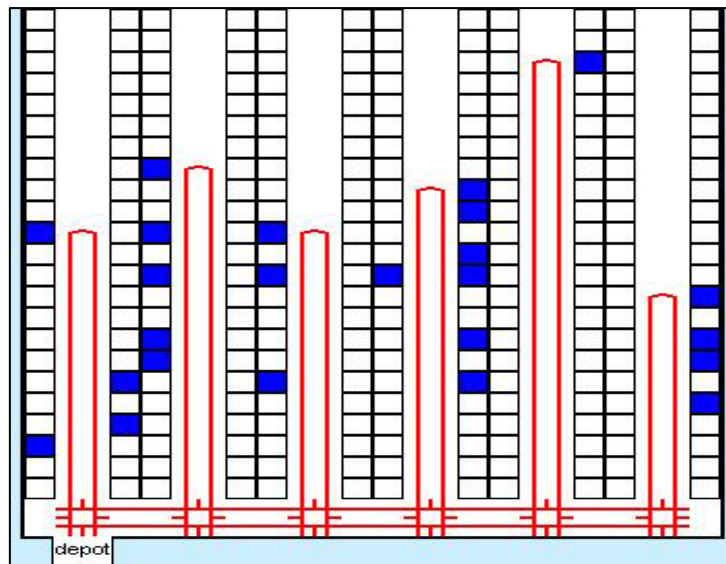


Σχήμα 11 Λογικό διάγραμμα απεικόνισης του αλγορίθμου που κατασκευάστηκε.

Για την εξαγωγή των αποτελεσμάτων χρησιμοποιήθηκε η μέθοδος της Προσομοιωμένης Ανόπτησης. Με τη μέθοδο αυτή ελέγχεται το κατά πόσο είναι αποδεκτή ή όχι η κάθε λύση που παράγεται από τον αλγόριθμο. Ο έλεγχος αυτός γίνεται με μια μεταβλητή B (βλ. Σχήμα 11), η οποία συμβολίζει

τη διαφορά της συνολικής απόστασης που προκύπτει από δύο διαδοχικές λύσεις-διατάξεις των προϊόντων. Η μεταβλητή B ξεκινάει στην συγκεκριμένη περίπτωση από 200 και όσο προχωρούν οι επαναλήψεις και οι προτεινόμενες λύσεις ελαττώνεται με αποτέλεσμα να φτάσει μέχρι και το 20.

Στο πρώτο μέρος του αλγορίθμου, γίνεται μετάφραση των δεδομένων που έχουν παρθεί από την εταιρία σε μορφή τέτοια ώστε να είναι εύκολα διαχειρίσιμα από τον αλγόριθμο. Στην συνέχεια, με τη βοήθεια των δεδομένων σχηματίζονται πίνακες για την εκάστοτε παραγγελία και ένας πίνακας που είναι το αποτύπωμα της αποθήκης (Σχήμα 13), αφού έχει όλους τους κωδικούς τοποθετημένους στην πραγματικής τους θέση. Έτσι, λοιπόν, υπολογίζεται η διαδρομή για την οποία προκύπτει η ελάχιστη απόσταση (1^ο βήμα του Σχήματος 11), με τη βοήθεια σχετικού αλγορίθμου [32] (βλ. Παράρτημα), που διανύεται για κάθε παραγγελία ξεχωριστά και στο τέλος προστίθενται και δίνουν την συνολική (Σχήμα 12).



Σχήμα 12 Διαδρομή συλλογής των προϊόντων μιας παραγγελία διανύοντας της ελάχιστης απόστασης.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101002	0	150026	103004	0	130004	101220	0	104600	100112	0	100934	100173	0	100510	0	0	101009	0
0	0	150028	0	0	0	0	0	101100	0	0	0	100175	0	100511	0	0	0	0
0	0	150029	111001	0	120002	122220	0	109100	100109	0	100932	100172	0	100512	0	0	0	0
103002	0	150019	0	0	103005	103220	0	107100	100101	0	100935	100174	0	100513	0	0	109011	0
0	0	124006	124001	0	150023	0	0	104300	100102	0	100292	100116	0	100514	100644	0	103015	0
109002	0	122005	0	0	150025	109220	0	124100	0	0	100937	100170	0	100515	0	0	107005	0
0	0	0	122004	0	150011	0	0	0	100110	0	100295	100142	0	100516	0	0	0	0
107002	0	0	0	0	0	111220	0	130100	0	0	100296	100143	0	0	0	0	101053	0
0	0	0	123001	0	150012	107220	0	0	100223	0	100297	100907	0	100800	0	0	0	0
104002	0	0	0	0	150013	124220	0	134001	0	0	100290	100908	0	0	0	0	150017	0
0	0	0	130001	0	150009	120220	0	134002	100225	0	100291	100130	0	100802	0	0	0	0
120001	0	0	0	0	150001	0	0	127100	100956	0	100168	100132	0	100803	0	0	0	0
0	0	0	127001	0	150004	130220	0	101100	100030	0	0	100134	0	0	0	0	0	0
150022	0	0	0	0	104008	0	0	137001	100031	0	100954	100135	0	100808	0	0	107012	0
150024	0	0	131002	0	130005	0	0	130015	100032	0	100955	100909	0	100807	0	0	109012	0
150005	0	0	0	0	150008	0	0	120100	102621	0	0	102634	0	0	0	0	122005	0
0	0	0	131005	0	101001	0	0	132007	0	0	100957	0	0	100810	0	0	0	0
0	0	0	0	0	129001	0	0	132005	102623	0	100958	0	0	100805	0	0	0	0
0	0	0	131003	0	129002	0	0	130100	100906	0	100959	0	0	100806	0	0	0	0
0	0	0	131004	0	130003	171001	0	134002	0	0	100757	100558	0	100123	0	0	0	0
0	0	0	123002	0	104006	0	0	0	100107	0	100760	100559	0	100124	0	0	0	0
0	0	0	125001	0	104001	0	0	134001	100113	0	102618	100560	0	100125	0	0	0	0
0	0	0	0	0	104003	0	0	132006	100117	0	0	100561	0	0	0	0	0	0
0	0	0	133001	0	160010	0	0	0	100115	0	102622	0	0	0	0	0	0	0
0	0	0	0	0	160011	0	0	0	0	0	0	0	0	100124	0	0	0	0
0	0	0	130030	0	101012	0	0	126100	100132	0	0	0	0	0	0	0	0	0
0	0	0	130031	0	0	0	0	120101	100134	0	100400	0	0	0	0	0	0	0
0	0	0	138001	0	0	0	0	0	0	0	100953	0	0	0	0	0	0	0
0	0	0	0	0	109013	150018	0	0	0	0	100955	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Σχήμα 13 Πραγματική διάταξη των προϊόντων στην αποθήκη.

Αφού λοιπόν υπολογίζεται η ελάχιστη απόσταση με τα προϊόντα διατεταγμένα όπως είναι στην πραγματικότητα, στη συνέχεια γίνεται μια προσπάθεια εύρεσης διαφορετικών θέσεων στα προϊόντα προκειμένου να ελαττωθεί το κόστος λειτουργίας της αποθήκης. Έτσι, σε αυτό το σκέλος περιέχονται διάφορες ευρετικές μέθοδοι αλλαγής των θέσεων των κωδικών στον χώρο της αποθήκης (2^ο βήμα του Σχήματος 11).

- I. Αρχικά, επιλέγεται να τοποθετηθούν τα προϊόντα στις διάφορες θέσεις, κατά σειρά μειούμενης συχνότητας εμφάνισης σε παραγγελίες σε κάθε στήλη. Για το σκοπό αυτό υπολογίζονται οι συχνότητες των προϊόντων και τοποθετούνται ανάλογα στις θέσεις της αποθήκης. Έτσι παράγεται μια νέα, 2^η διάταξη των προϊόντων στον χώρο (βλ. Σχήμα 14).

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
109002	0	150028	131002	0	150011	171001	0	109100	100225	0	100934	100130	0	100511	100644	0	101009
120001	0	124006	130001	0	129001	109220	0	101100	100112	0	100296	100175	0	100512	0	0	109011
107002	0	150019	131005	0	130004	107220	0	104600	100109	0	100295	100173	0	100513	0	0	107005
104002	0	122005	111001	0	103005	101220	0	132005	102621	0	100297	100143	0	100514	0	0	150017
101002	0	150026	123001	0	150025	130220	0	101100	100906	0	100291	100116	0	100802	0	0	103015
150022	0	150029	131004	0	130003	124220	0	107100	100110	0	102618	100132	0	100805	0	0	101053
150005	0	0	127001	0	130005	103220	0	124100	102623	0	100957	100170	0	100806	0	0	122005
150024	0	0	130031	0	150012	120220	0	120100	100223	0	100958	100134	0	100124	0	0	107012
103002	0	0	103004	0	150001	122220	0	132006	100117	0	100955	100135	0	100510	0	0	109012
0	0	0	122004	0	104008	111220	0	130015	100115	0	100292	100558	0	100516	0	0	0
0	0	0	123002	0	104001	150018	0	134002	100101	0	100400	100142	0	100800	0	0	0
0	0	0	124001	0	150023	0	0	132007	100113	0	100954	100561	0	100803	0	0	0
0	0	0	131003	0	101001	0	0	134002	100107	0	100290	100174	0	100807	0	0	0
0	0	0	133001	0	104006	0	0	130100	100134	0	100959	100909	0	100125	0	0	0
0	0	0	125001	0	104003	0	0	120101	100102	0	100760	100560	0	100515	0	0	0
0	0	0	138001	0	120002	0	0	130100	100030	0	100954	100172	0	100808	0	0	0
0	0	0	130030	0	129002	0	0	126100	100031	0	100935	100559	0	100810	0	0	0
0	0	0	0	0	150013	0	0	104300	100132	0	100757	100908	0	100123	0	0	0
0	0	0	0	0	150009	0	0	134001	100956	0	100932	102634	0	100124	0	0	0
0	0	0	0	0	150004	0	0	127100	100032	0	100937	100907	0	0	0	0	0
0	0	0	0	0	150008	0	0	137001	0	0	100168	0	0	0	0	0	0
0	0	0	0	0	160010	0	0	134001	0	0	102622	0	0	0	0	0	0
0	0	0	0	0	160011	0	0	0	0	0	100953	0	0	0	0	0	0
0	0	0	0	0	101012	0	0	0	0	0	100955	0	0	0	0	0	0
0	0	0	0	0	109013	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Σχήμα 14 Διάταξη των προϊόντων μετά την 1^η αλλαγή.

II. Η δεύτερη ευρετική μέθοδος που χρησιμοποιήθηκε ακολουθεί την ίδια λογική με την πρώτη με την διαφορά ότι εφαρμόζεται σε κάθε γραμμή της αποθήκης και παράγεται ένας 3^{ος} τρόπος τοποθέτησης των προϊόντων. Να σημειωθεί εδώ ότι η αλλαγή γίνεται παίρνοντας ως βάση την ήδη τροποποιημένη διάταξη προϊόντων με τον πρώτο τρόπο (βλ. Σχήμα 15).

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101009	0	100225	131002	0	150011	100130	0	109002	100934	0	109100	150028	0	171001	100511	0	0	0	0
109011	0	129001	120001	0	100112	100296	0	100175	130001	0	124006	101100	0	109220	100512	0	0	0	0
130004	0	107005	107002	0	100295	131005	0	100109	100173	0	104600	150019	0	107220	100513	0	0	0	0
103005	0	104002	102621	0	111001	100143	0	100297	150017	0	132005	122005	0	101220	100514	0	0	0	0
123001	0	100906	101002	0	150025	100116	0	100291	103015	0	101100	150026	0	130220	100802	0	0	0	0
100110	0	130003	131004	0	150022	100132	0	102618	107100	0	150029	124220	0	101053	100805	0	0	0	0
102623	0	127001	130005	0	100170	150005	0	100957	124100	0	122005	103220	0	100806	0	0	0	0	0
130031	0	100223	150012	0	150024	100958	0	120100	100134	0	120220	100124	0	107012	0	0	0	0	0
103004	0	100117	150001	0	100955	103002	0	132006	100135	0	122220	100510	0	109012	0	0	0	0	0
100115	0	122004	104008	0	100558	100292	0	130015	111220	0	100516	0	0	0	0	0	0	0	0
123002	0	100101	104001	0	100142	100400	0	134002	150018	0	100800	0	0	0	0	0	0	0	0
124001	0	150023	100113	0	100561	100954	0	132007	100803	0	0	0	0	0	0	0	0	0	0
101001	0	131003	100107	0	100174	100290	0	134002	100807	0	0	0	0	0	0	0	0	0	0
104006	0	100909	100134	0	100959	133001	0	130100	100125	0	0	0	0	0	0	0	0	0	0
104003	0	100560	100760	0	100102	125001	0	120101	100515	0	0	0	0	0	0	0	0	0	0
120002	0	100172	100030	0	138001	100954	0	130100	100808	0	0	0	0	0	0	0	0	0	0
129002	0	100559	100031	0	126100	100935	0	130030	100810	0	0	0	0	0	0	0	0	0	0
150013	0	100132	100908	0	100757	104300	0	100123	0	0	0	0	0	0	0	0	0	0	0
150009	0	100956	102634	0	134001	100932	0	100124	0	0	0	0	0	0	0	0	0	0	0
150004	0	100032	100907	0	127100	100937	0	0	0	0	0	0	0	0	0	0	0	0	0
150008	0	137001	100168	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160010	0	134001	102622	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160011	0	100953	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101012	0	100955	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
109013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Σχήμα 15 Διάταξη των προϊόντων μετά την 2^η αλλαγή.

III. Η τρίτη μέθοδος αλλαγής των θέσεων των προϊόντων στηρίζεται στην σχετική συχνότητα εμφάνισης διάφορων ζευγαριών από αυτά στις ίδιες παραγγελίες. Το ερέθισμα για την ανάπτυξη αυτής της μεθόδου δόθηκε από τους εργαζομένους της αποθήκης, οι οποίοι περιέγραψαν με λόγια το τι θα επιθυμούσαν να κάνουν για να διευκολύνουν τη δουλειά τους. Έτσι, αυτή η μέθοδος είναι, ουσιαστικά, τεχνητή νοημοσύνη που αξιοποιεί την εμπειρία και μαθηματικοποιεί τις προτάσεις βελτίωσης των συλλεκτών. Επιλέγονται για τον σκοπό αυτό οι πιο συχνά εμφανιζόμενοι κωδικοί, ώστε να έχουν νόημα οι μετακινήσεις (π.χ. με συχνότητα εμφάνισης στο σύνολο των παραγγελιών > 15). Η επιλογή του αριθμού 15 και η εξέταση ζευγαριών κωδικών (και όχι π.χ. τριάδων ή τετράδων) είναι παράμετροι του προβλήματος και επιδέχονται βελτιστοποίηση. Για το σκοπό αυτό οι κωδικοί αυτοί τοποθετούνται με τυχαία σειρά σε ένα διάγραμμα γραμμή. Για το κάθε ζευγάρι που μπορεί να δημιουργηθεί από τους παραπάνω κωδικούς υπολογίζεται σχετική συχνότητα εμφάνισης τους στις διάφορες παραγγελίες και συνολικά στο τέλος. Τελειώνοντας, βρίσκεται για τον κάθε κωδικό το ζεύγος του με την μεγαλύτερη συνολική σχετική συχνότητα και τοποθετούνται διαδοχικά σε δύο θέσεις της αποθήκης. Έτσι, δημιουργείται μια νέα, 4^η διάταξη των κωδικών στον χώρο (βλ. Σχήμα 16). Να σημειωθεί ότι η αλλαγή αυτή γίνεται λαμβάνοντας ως βάση την αλλαγμένη ήδη δύο φορές από τις προηγούμενες μεθόδους διάταξη των προϊόντων.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
130005	0	150017	103004	0	100132	150025	0	109002	100934	0	109100	150028	0	171001	100511	0	0
100173	0	100116	120001	0	100112	102623	0	130031	101009	0	123001	101100	0	109220	100512	0	0
129001	0	100110	131002	0	130001	100906	0	124006	100109	0	100291	150019	0	107220	100513	0	0
103005	0	150012	150001	0	100225	131004	0	100297	100175	0	132005	122005	0	101220	100514	0	0
109011	0	100130	101002	0	107002	130003	0	111001	103015	0	101100	150026	0	130220	100802	0	0
107005	0	131005	130004	0	150022	150005	0	102618	107100	0	150029	124220	0	101053	100805	0	0
150011	0	127001	100295	0	100170	104600	0	100957	124100	0	122005	103220	0	100806	0	0	0
104002	0	100223	102621	0	150024	100958	0	120100	100134	0	120220	100124	0	107012	0	0	0
100296	0	100117	100143	0	100955	103002	0	132006	100135	0	122220	100510	0	109012	0	0	0
100115	0	122004	104008	0	100558	100292	0	130015	111220	0	100516	0	0	0	0	0	0
123002	0	100101	104001	0	100142	100400	0	134002	150018	0	100800	0	0	0	0	0	0
124001	0	150023	100113	0	100561	100954	0	132007	100803	0	0	0	0	0	0	0	0
101001	0	131003	100107	0	100174	100290	0	134002	100807	0	0	0	0	0	0	0	0
104006	0	100909	100134	0	100959	133001	0	130100	100125	0	0	0	0	0	0	0	0
104003	0	100560	100760	0	100102	125001	0	120101	100515	0	0	0	0	0	0	0	0
120002	0	100172	100030	0	138001	100954	0	130100	100808	0	0	0	0	0	0	0	0
129002	0	100559	100031	0	126100	100935	0	130030	100810	0	0	0	0	0	0	0	0
150013	0	100132	100908	0	100757	104300	0	100123	0	0	0	0	0	0	0	0	0
150009	0	100956	102634	0	134001	100932	0	100124	0	0	0	0	0	0	0	0	0
150004	0	100032	100907	0	127100	100937	0	0	0	0	0	0	0	0	0	0	0
150008	0	137001	100168	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160010	0	134001	102622	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160011	0	100953	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101012	0	100955	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
109013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Σχήμα 16 Διάταξη των προϊόντων μετά την 3^η αλλαγή.

IV. Τα ζευγάρια που δημιουργούνται στην περίπτωση III, είναι σχηματισμένα με τυχαίο τρόπο και υπάρχουν πολλοί άλλοι συνδυασμοί που δεν λαμβάνονται υπόψη. Ένας τρόπος για να παραχθεί μια νέα σειρά ζευγαριών είναι να ανακαταταγούν στο διάλυσμα γραμμή οι συχνά εμφανιζόμενοι κωδικοί κατά σειρά μειούμενης συχνότητας. Έτσι, δημιουργούνται νέα ζεύγη που οδηγούν και σε νέα αποτελέσματα κατόπιν εφαρμογής της διαδικασίας που περιγράφηκε στην μεθοδολογία 3 και παράγεται ο 5^{ος} τρόπος διάταξης των προϊόντων (βλ. Σχήμα 17).

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101009	0	111001	100170	0	150011	100130	0	150022	100934	0	150025	150028	0	171001	100511	0	0	0
109011	0	123001	150024	0	131002	150012	0	100175	100143	0	150017	101100	0	109220	100512	0	0	0
100225	0	109100	107002	0	129001	131005	0	100295	102623	0	130003	150019	0	107220	100513	0	0	0
100112	0	130005	100296	0	130004	101002	0	103015	100173	0	131004	122005	0	101220	100514	0	0	0
100906	0	130001	100109	0	107005	104002	0	100291	100955	0	101100	150026	0	130220	100802	0	0	0
100116	0	124006	100110	0	109002	100132	0	102618	107100	0	150029	124220	0	101053	100805	0	0	0
104600	0	127001	100297	0	103005	150005	0	100957	124100	0	122005	103220	0	100806	0	0	0	0
130031	0	100223	132005	0	120001	100958	0	120100	100134	0	120220	100124	0	107012	0	0	0	0
103004	0	100117	150001	0	102621	103002	0	132006	100135	0	122220	100510	0	109012	0	0	0	0
100115	0	122004	104008	0	100558	100292	0	130015	111220	0	100516	0	0	0	0	0	0	0
123002	0	100101	104001	0	100142	100400	0	134002	150018	0	100800	0	0	0	0	0	0	0
124001	0	150023	100113	0	100561	100954	0	132007	100803	0	0	0	0	0	0	0	0	0
101001	0	131003	100107	0	100174	100290	0	134002	100807	0	0	0	0	0	0	0	0	0
104006	0	100909	100134	0	100959	133001	0	130100	100125	0	0	0	0	0	0	0	0	0
104003	0	100560	100760	0	100102	125001	0	120101	100515	0	0	0	0	0	0	0	0	0
120002	0	100172	100030	0	138001	100954	0	130100	100808	0	0	0	0	0	0	0	0	0
129002	0	100559	100031	0	126100	100935	0	130030	100810	0	0	0	0	0	0	0	0	0
150013	0	100132	100908	0	100757	104300	0	100123	0	0	0	0	0	0	0	0	0	0
150009	0	100956	102634	0	134001	100932	0	100124	0	0	0	0	0	0	0	0	0	0
150004	0	100032	100907	0	127100	100937	0	0	0	0	0	0	0	0	0	0	0	0
150008	0	137001	100168	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160010	0	134001	102622	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160011	0	100953	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101012	0	100955	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
109013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Σχήμα 17 Διάταξη των προϊόντων μετά την 4^η αλλαγή.

V. Ένας επιπλέον τρόπος δημιουργίας καινούργιων ζευγαριών είναι ο εξής: Εφαρμόζεται μια επαναληπτική διαδικασία στην οποία για κάθε καινούργια επανάληψη οι συχνά εμφανιζόμενοι κωδικοί στο διάνυσμα γραμμή μετατοπίζονται κατά μια θέση δεξιά. Έτσι μετά από πολλές επαναλήψεις οι κωδικοί έχουν πάρει όλες τις πιθανές θέσεις και έχουν εξεταστεί διαφορετικοί συνδυασμοί ζευγαριών και ταυτόχρονα έχουν παραχθεί και διαφορετικοί τρόποι διάταξης των προϊόντων στην αποθήκη. Με αυτό τον τρόπο παράγονται πολλά καινούργια αποτελέσματα με την επαναλαμβανόμενη χρήση της μεθοδολογίας III.

Αφού λοιπόν, δημιουργούνται οι καινούργιοι τρόποι διάταξης κωδικών προϊόντων με τις διάφορες ευρετικές μεθόδους που περιγράφηκαν παραπάνω, πρέπει να υπολογιστεί η συνολική απόσταση (3^ο βήμα του Σχήματος 11) που θα διανυθεί από τους συλλέκτες των προϊόντων για το σύνολο των παραγγελιών, ώστε να ελεγχθεί κατά πόσο βελτιώνουν ή χειροτερεύουν το αποτέλεσμα για το κόστος λειτουργίας της αποθήκης. Για το σκοπό αυτό θα πρέπει για την κάθε καινούργια διάταξη που σχηματίζεται με τις διάφορες μεθοδολογίες να αναδιαμορφώνονται και οι αντίστοιχες παραγγελίες που θα περιέχουν τα προϊόντα σε στις νέες τους θέσεις, και με βάση τις οποίες θα υπολογιστεί και η συνολική απόσταση. Δηλαδή πρέπει να γίνει η αντίστροφη διαδικασία από αυτή που περιγράφηκε στο πρώτο μέρος του αλγορίθμου και έτσι με δεδομένη την διάταξη των προϊόντων να δημιουργηθούν οι παραγγελίες με τα προϊόντα που περιλαμβάνουν στις νέες θέσεις τους και να υπολογιστεί η απόσταση που διανύεται από τους συλλέκτες.

8. Υπόθεση Εργασίας

Ο αλγόριθμος που δημιουργήθηκε, έχει στόχο την μείωση του χρόνου που καταναλώνεται για την συλλογή των διαφόρων παραγγελιών της αποθήκης του Μπαρμπαστάθη στην Αθήνα, που οδηγεί κατά συνέπεια και στην ελάττωση του κόστους λειτουργίας της. Για τον σκοπό αυτό, σε συνεργασία με τους υπεύθυνους συλλέχθηκαν δεδομένα, τα οποία αφορούν δελτία παραγγελιών (βλ. Σχήμα 18), μέσω των οποίων μπορεί να σχηματιστεί μια εικόνα της αποθήκης που προσεγγίζει ικανοποιητικά την πραγματικότητα. Τα δεδομένα αυτά, με κατάλληλη επεξεργασία, μπορούν να παράγουν αποτελέσματα σχετικά με την διάταξη των προϊόντων στην αποθήκη, αλλά και τις διαδρομές που γίνονται για να συλλεχθούν αυτά, ικανοποιώντας τους στόχους που έχουν τεθεί.

Ο αποθηκευτικός χώρος, για τον οποίο έγινε η παρούσα εργασία είναι το παράρτημα της εταιρίας Μπαρμπαστάθη στην Αθήνα. Η κεντρική αποθήκη βρίσκεται στη Θεσσαλονίκη και ο χώρος της Αθήνας χρησιμοποιείται ως κέντρο διανομής (ΚΔ) για να διευκολύνει τους πελάτες της Αττικής, αλλά και γενικότερα της Στερεάς Ελλάδας και της Πελοποννήσου. Ο αποθηκευτικός αυτός χώρος, λοιπόν, αποτελείται από ένα μπλοκ που έχει 6 διαδρόμους και 30 θέσεις τοποθέτησης των προϊόντων σε κάθε σειρά ραφιών (βλ. Σχήμα 19). Ένα από τα ιδιαίτερα χαρακτηριστικά του είναι ότι καταψύχεται συνεχώς λόγω της φύσης των προϊόντων που φυλάσσονται μέσα. Ένα άλλο χαρακτηριστικό που αξίζει να σημειωθεί είναι ότι τα ράφια είναι κινούμενα, δηλαδή κάθε στιγμή υπάρχει ένας διάδρομος ανοιχτός, μέχρι να δοθεί η εντολή να μετακινηθούν τα ράφια και να ανοίξει κάποιος άλλος. Αυτό από τη μια εξοικονομεί χώρο στην αποθήκη, ενώ από την άλλη αναγκάζει τους συλλέκτες των προϊόντων να έχουν περισσότερο χρόνο παραμονής μέσα στον χώρο που καταψύχεται. Όσον αφορά τον τρόπο με τον οποίο συλλέγονται τα προϊόντα του ΚΔ εφαρμόζεται η συλλογή ανά παραγγελία που προσφέρει ταχύτητα, ευελιξία και λιγότερες διαδρομές και το χειροκίνητο σύστημα συλλογής παραγγελιών λόγω περιορισμού που θέτουν οι ήδη υπάρχουσες εγκαταστάσεις της εταιρίας. Στο Σχήμα 12 φαίνεται ο τρόπος που έχει επιλεγεί για την συλλογή των προϊόντων σε κάθε παραγγελία. Ο τρόπος με τον οποίο είναι διατεταγμένα τα προϊόντα στις θέσεις του ΚΔ στηρίζεται κυρίως στην εμπειρία και τις ανάγκες των συλλεκτών.

Picking List Ανά Δρομολόγιο

Ημερ/νία Εκτύπωση

Δρομολόγιο: 0115

Φορτηγό: 2187

Τακτ.Δρομ.: 0115

ΝΙΚΟΣ



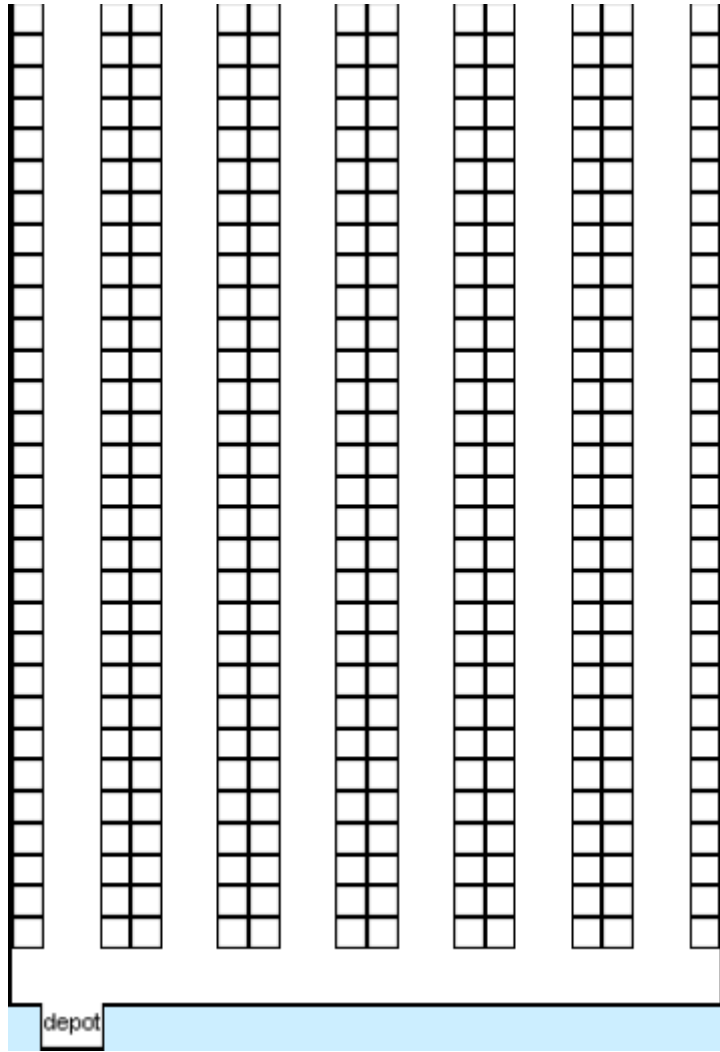
0000000018

ΡΑΜΠΑ: f65

PickingList LU

Κωδ.Είδους	Περιγραφή	Θέση *
165002	15 FISH STICKS 6X450	B10 - 27 - 0 -
165008	ΚΑΛΑΜΑΡΑΚΙΑ A LA ROMANA 10X400	BB - 8 - 0 -
166000	11CHICKEN NUGGETS 10X230	BB - 10 - 0 -
120001	ΣΠΑΝΑΚΙ ΦΥΛΛΑ IQF ΜΕΡΙΔΕΣ 10X1000	CA - 12 - 0 -
150026	ΑΓΚΙΝΑΡΕΣ ΑΛΑ ΠΟΛΙΤΑ(-1,5Ε) 8X1200 ΜΣ	C01 - 1 - 0 -
150028	ΑΡΑΚΑΣ ΜΕ ΑΓΚΙΝΑΡΕΣ 8X1200 (-1,5 Ε) ΜΣ	C01 - 2 - 0 -
150019	ΦΑΣΟΛΑΚΙΑ ΣΤΡΟΓ.ΛΑΔΕΡΑ(-1 Ε)10X1000	C01 - 4 - 0 -
124006	ΚΑΛΑΜΠΟΚΙ (450+150 ΔΩΡΟ) 15X600 ΜΣ	C01 - 5 - 0 -
122005	ΑΓΚΙΝΑΡΕΣ 12 X 600 (-1Ε) ΜΣ	C01 - 6 - 0 -
104002	ΠΑΤΑΤΕΣ ΚΛΑΣΣΙΚΕΣ 10 X 1000 ΜΣ	CA - 10 - 0 -
107002	ΦΑΣΟΛΑΚΙΑ ΠΛΑΤΙΑ 20 X 450 ΜΣ	CA - 8 - 0 -
109002	ΦΑΣΟΛΑΚΙΑ ΣΤΡΟΓΓΥΛΑ 20 X 450 ΜΣ	CA - 6 - 0 -
101002	ΑΡΑΚΑΣ 24 X 450 ΜΣ	CA - 1 - 0 -
130004	ΣΑΛΑΤΑ ΚΑΛΑΜΠΟΚΙΟΥ 16 X 450 ΜΣ	C03 - 1 - 0 -
103005	ΜΠΑΜΙΑ EXTRA 10X1000γρ	C03 - 4 - 0 -
150025	ΑΡΑΚΑΣ ΜΕ ΑΓΚΙΝΑΡΕΣ 10X600 ΜΣ	C03 - 6 - 0 -
150011	ΑΡΑΚΑΣ ΛΑΔΕΡΟΣ 10 X 1000 ΜΣ	C03 - 7 - 0 -
150004	ΦΑΣΟΛΑΚΙΑ ΓΙΑΧΝΗ 10 X 1000 ΜΣ	C03 - 13 - 0 -

Σχήμα 18 Δείγμα δελτίου παραγγελίας



Σχήμα 19 Η κάτοψη της αποθήκης.

9. Αποτελέσματα

Τα αποτελέσματα που παράχθηκαν για τις παραπάνω περιπτώσεις έχουν καταγραφεί στον παρακάτω πίνακα:

Πίνακας 1 Οι συνολικές αποστάσεις για τις 5 πρώτες διατάξεις των προϊόντων στην αποθήκη

Total_distance 1	Total_distance 2	Total_distance 3	Total_distance 4	Total_distance 5
6312	3825	3656	3775	3696

Πίνακας 2 Οι τιμές του Total_distance κατά τις διαδοχικές αλλαγές που λαμβάνουν χώρα στην 5η μεθοδολογία.

Total_Distance 6	
1	3775
2	3678
3	3670
4	3668
5	3647
6	3668
7	3666
8	3641
9	3692
10	3682
11	3711
12	3713
13	3664
14	3660
15	3668
16	3684
17	3691
18	3637
19	3609
20	3615
21	3654
22	3805
23	3779
24	3813
25	3781
26	3645
27	3634
28	3660
29	3670
30	3651
31	3678
32	3730
33	3922
34	3647
35	3655
36	3633
37	3663

Total_distance1: Η συνολική απόσταση που διανύεται για όλες τις παραγγελίες με την πραγματική διάταξη των προϊόντων στο χώρο της αποθήκης που έχει γίνει από την εταιρία.

Total_distance2: Η συνολική απόσταση που διανύεται για όλες τις παραγγελίες έχοντας αλλάξει τη διάταξη των προϊόντων στο χώρο της αποθήκης με την ευρετική μέθοδο 1.

Total_distance3: Η συνολική απόσταση που διανύεται για όλες τις παραγγελίες έχοντας αλλάξει τη διάταξη των προϊόντων στο χώρο της αποθήκης με την ευρετική μέθοδο 2.

Total_distance4: Η συνολική απόσταση που διανύεται για όλες τις παραγγελίες έχοντας αλλάξει τη διάταξη των προϊόντων στο χώρο της αποθήκης με την ευρετική μέθοδο 3.

Total_distance5: Η συνολική απόσταση που διανύεται για όλες τις παραγγελίες έχοντας αλλάξει τη διάταξη των προϊόντων στο χώρο της αποθήκης με την ευρετική μέθοδο 4.

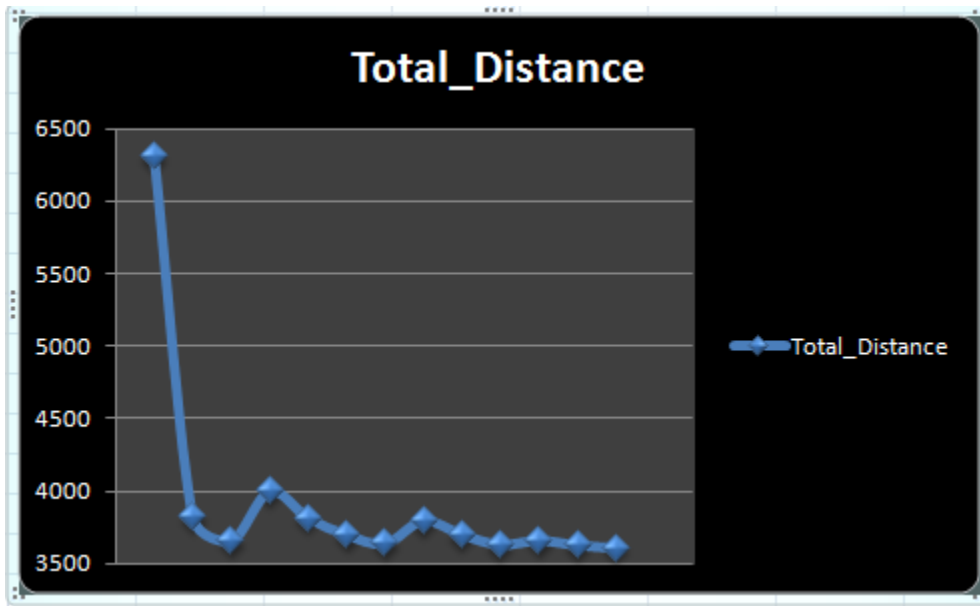
Total_distance6: Το διάλυμα γραμμή που περιέχει τις συνολικές αποστάσεις για τις διάφορες θέσεις των κωδικών στον πίνακα Frequent_Codes, όπως έχει περιγραφεί στην ευρετική μέθοδο 5.

Τα αποτελέσματα μοιάζουν ιδιαίτερα ενθαρρυντικά αφού ακόμα και με την πρώτη αλλαγή που γίνεται, η συνολική απόσταση που διανύεται από τους συλλέκτες της αποθήκης και συνεπώς το κόστος λειτουργίας της μειώνεται αισθητά (40%). Αυτό φαίνεται συγκρίνοντας τα Total_distance1 και Total_distance2. Συνεπώς το SKU_new που προκύπτει με την πρώτη αλλαγή είναι αποδεκτό σαν λύση. Προχωρώντας και σε επόμενη αλλαγή βλέπουμε ότι τα αποτελέσματα βελτιώνονται παραπάνω (μείωση 5%), όχι βέβαια όσο και με την πρώτη αλλαγή (Total_distance3). Έτσι, γίνεται και το SKU_new2 αποδεκτό και προτεινόμενο για την εταιρία. Ακόμα και μερικές περιπτώσεις από την τρίτη μεθοδολογία αλλαγών φαίνεται να έχουν μικρή μεν (1%), αλλά θετική επίδραση δε, στο τελικό αποτέλεσμα (Total_distance6). Συνεπώς, τα αποτελέσματα που γίνονται αποδεκτά είναι αυτά με τους αύξοντες αριθμούς 5, 8, 18, 19, 20, 21, 26, 27, 30, 34, 35 και 36 του πίνακα. Το κριτήριο με το οποίο γίνονται αποδεκτά τα παραπάνω αποτελέσματα είναι ότι τηρούν την συνθήκη που περιγράφεται στο 4^ο βήμα του Σχήματος 11, δηλαδή $Total_distance_{new} < Total_distance_{old} + B$. Η τελική λύση που θα γίνει αποδεκτή και θα προταθεί στην εταιρεία είναι αυτή βέβαια που δίνει το μικρότερο συνολικό Total_distance και είναι η περίπτωση 19 του Total_distance6, όπου και σταματάει να ισχύει η παραπάνω συνθήκη και έτσι τερματίζεται ο αλγόριθμος, όπως φαίνεται και στο Σχήμα 11. Παρακάτω φαίνεται και η διάταξη με την οποία πρέπει να τοποθετηθούν τα προϊόντα προκειμένου να επιτευχθεί αυτή η λύση (Σχήμα 20):

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
130005	0	127001	103004	0	100132	150025	0	109002	100909	0	109100	150028	0	171001	100511	0	0	0
100173	0	150001	107100	0	100112	120100	0	107005	150024	0	104600	101100	0	109220	100512	0	0	0
101009	0	101002	100935	0	104300	100101	0	150011	100175	0	100954	150019	0	107220	100513	0	0	0
104002	0	130004	150012	0	102621	103005	0	102623	100143	0	100223	122005	0	101220	100514	0	0	0
150023	0	100297	123001	0	100170	109011	0	100296	131002	0	102618	150026	0	130220	100802	0	0	0
125001	0	100958	100110	0	107002	120001	0	100954	100906	0	150029	124220	0	101053	100805	0	0	0
130003	0	131004	100295	0	100932	100291	0	124100	133001	0	122005	103220	0	100806	0	0	0	0
150005	0	100934	100292	0	104001	100116	0	100957	122004	0	120220	100124	0	107012	0	0	0	0
100135	0	111001	131003	0	100955	101100	0	100400	124006	0	122220	100510	0	109012	0	0	0	0
100115	0	150022	132005	0	100558	130001	0	130015	111220	0	100516	0	0	0	0	0	0	0
123002	0	100130	100109	0	100142	100225	0	134002	150018	0	100800	0	0	0	0	0	0	0
124001	0	100117	104008	0	100561	132006	0	132007	100803	0	0	0	0	0	0	0	0	0
101001	0	100134	100107	0	100174	100290	0	134002	100807	0	0	0	0	0	0	0	0	0
104006	0	130031	100134	0	100959	129001	0	130100	100125	0	0	0	0	0	0	0	0	0
104003	0	100560	100760	0	100102	131005	0	120101	100515	0	0	0	0	0	0	0	0	0
120002	0	100172	100030	0	138001	103015	0	130100	100808	0	0	0	0	0	0	0	0	0
129002	0	100559	100031	0	126100	150017	0	130030	100810	0	0	0	0	0	0	0	0	0
150013	0	100132	100908	0	100757	100113	0	100123	0	0	0	0	0	0	0	0	0	0
150009	0	100956	102634	0	134001	103002	0	100124	0	0	0	0	0	0	0	0	0	0
150004	0	100032	100907	0	127100	100937	0	0	0	0	0	0	0	0	0	0	0	0
150008	0	137001	100168	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160010	0	134001	102622	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160011	0	100953	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
101012	0	100955	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
109013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Σχήμα 20 Διάταξη των προϊόντων μετά την 5^η αλλαγή.

Αξίζει να σημειωθεί ότι πολλές φορές σε τέτοιου είδους προβλήματα πρέπει να γίνονται αποδεκτές και λύσεις οι οποίες δείχνουν αρχικά να μην βελτιώνουν το αποτέλεσμα, αλλά μακροπρόθεσμα και σε συνδυασμό με άλλες μεθόδους μπορούν να οδηγήσουν σε κάτι καλύτερο, αφού βοηθούν τον αλγόριθμο να ξεφεύγει από τα τοπικά ελάχιστα που μπορεί να συναντήσει κατά την παραγωγή λύσεων(βλ. Σχήμα 21). Αυτό συνέβη και στο συγκεκριμένο πρόβλημα αφού π.χ. το SKU_new3 που δημιουργείται μας δίνει Total_distance4 μεγαλύτερο από το προηγούμενο του. Παρ' όλα αυτά με κάποιες μορφοποιήσεις αργότερα όπως έχει περιγραφεί στις μεθοδολογίες 4, 5, επιτυγχάνεται καλύτερο αποτέλεσμα και από το Total_distance3 όπως φαίνεται σε περιπτώσεις του Total_distance6 που αναφέρθηκαν και παραπάνω. Ένα ακόμα παράδειγμα που επιβεβαιώνει την περίπτωση στην οποία αναφερόμαστε είναι η 16^η, 17^η και 18^η γραμμή στον πίνακα Total_distance6. Στην περίπτωση αυτή η διάταξη των προϊόντων στον χώρο αλλάζει μια φορά από την 16^η στην 17^η γραμμή και δίνει το αποτέλεσμα που αναγράφεται όπου μοιάζει να γίνεται χειρότερο από το ήδη υπάρχον. Παρ' όλα αυτά αν δεν είχε γίνει αυτή η αλλαγή δε θα οδηγούμασταν στην λύση της γραμμής 18, η οποία είναι σαφώς καλύτερη και από την 16^η και έχει προκύψει με αλλαγή στην ήδη αλλαγμένη διάταξη από την οποία προκύπτει το φαινομενικά μη αποδεκτό αποτέλεσμα της γραμμής 17.



Σχήμα 21 Η πορεία του Total_distance με τις αλλαγές στη διάταξη των προϊόντων της αποθήκης

10. Συμπεράσματα

Ο συνολικός χρόνος διεκπεραίωσης των παραγγελιών στις μεγάλες αποθήκες περιλαμβάνει την συλλογή των προϊόντων από τις θέσεις αποθήκευσης και στην συνέχεια την αποστολή τους. Η διαδικασία αυτή καταλαμβάνει περίπου το 60% του συνολικού χρόνου [28] και είναι συνάρτηση της επιλογής των θέσεων στις οποίες αποθηκεύονται τα προϊόντα καθώς και τις επιλογής της σειράς συλλογής των προϊόντων της κάθε παραγγελίας. Το πρόβλημα της βέλτιστης χωροθέτησης των προϊόντων στην αποθήκη διαμορφώνεται ως πρόβλημα διεπίπεδου ακέραιου προγραμματισμού, με κύριο αντικειμενικό στόχο την ελαχιστοποίηση του συνολικού χρόνου συλλογής των προϊόντων όλων των παραγγελιών. Ο δευτερεύον στόχος είναι η ελαχιστοποίηση του χρόνου συλλογής των προϊόντων της κάθε παραγγελίας ξεχωριστά. Στην εργασία αυτή αναπτύχθηκε υβριδικός αλγόριθμος που περιλαμβάνει έναν μεταερευτικό αλγόριθμο για την επίλυση του πρωτεύοντος προβλήματος που λειτουργεί συνεργατικά με έναν αλγόριθμο δυναμικού προγραμματισμού για την επίλυση του δευτερεύοντος προβλήματος. Ο αλγόριθμος χρησιμοποιήθηκε για την επίλυση προβλήματος υπόθεσης εργασίας της αποθήκης κατεψυγμένων προϊόντων του Μπαρμπαστάθης Α.Ε. με χρήση ιστορικών δεδομένων παραγγελιών και πληροφορίες για την διάταξη της αποθήκης αλλά και τις συνήθειες πρακτικές συλλογής των προϊόντων. Τα αποτελέσματα της βελτιστοποίησης δείχνουν ότι υπάρχει περιθώριο μείωσης του συνολικού χρόνου συλλογής των παραγγελιών έως και 46% αν μεταβληθεί η υπάρχουσα διάταξη των προϊόντων στην αποθήκη σύμφωνα με την διάταξη που υποδεικνύει η βέλτιστη λύση. Το ποσοστό αυτό δείχνει ότι ακόμα και αν για λόγους ύπαρξης επιπλέον περιορισμών που δεν έχουν ληφθεί υπόψη κατά την επίλυση του προβλήματος η προτεινόμενη λύση δεν μπορεί να υλοποιηθεί με ακρίβεια, υπάρχει σαφές περιθώριο μείωσης του συνολικού χρόνου συλλογής και άρα του κόστους λειτουργίας της αποθήκης. Μια σημαντική παράμετρος που μπορεί να ληφθεί υπόψη σε μελλοντική έρευνα της χωροθέτησης προϊόντων σε αποθήκες είναι η διερεύνηση και αξιολόγηση της δυνατότητας συγκέντρωσης των παραγγελιών σε κοινά δρομολόγια συλλογής και στην συνέχεια διαχωρισμός των προϊόντων σε φορτία της κάθε παραγγελίας πριν την αποστολή.

Παράρτημα

Αναλυτική περιγραφή του αλγόριθμου επίλυσης του προβλήματος

Στο πρώτο μέρος του αλγορίθμου, που γράφτηκε σε Matlab, γίνεται μετάφραση των δεδομένων που έχουν παρθεί από την εταιρία σε μορφή τέτοια ώστε να είναι εύκολα διαχειρίσιμα από τον αλγόριθμο. Πιο συγκεκριμένα με το script "Position"(βλ. 3^ο μέρος παραρτήματος) ο αλγόριθμος διαβάζει τα δεδομένα που έχουν καταγραφεί στο excel και με μια επαναληπτική διαδικασία τα μετατρέπει σε δύο μορφές πινάκων. Στον πρώτο πίνακα (Codes1) περιέχονται οι κωδικοί των προϊόντων στις αντίστοιχες θέσεις τους μέσα στην αποθήκη για την εκάστοτε παραγγελία, ενώ στον δεύτερο πίνακα, που είναι δυαδικός (Orders1), σημειώνεται με 1 το προϊόν, στην αντίστοιχη θέση του μέσα στην αποθήκη, που περιέχεται στην εκάστοτε παραγγελία. Στην συνέχεια σχηματίζεται ο πίνακας ddd που είναι πολυεπίπεδος και κάθε του επίπεδο είναι μια διαφορετική παραγγελία και οι αντίστοιχοι κωδικοί που περιέχονται σε αυτήν. Έτσι, λοιπόν επιλέγοντας από όλα τα επίπεδα του πίνακα το μέγιστο για το κάθε του στοιχείο (i,j), δημιουργείται ο πίνακας SKUs που είναι ουσιαστικά το αποτύπωμα της αποθήκης αφού περιέχει όλους τους κωδικούς και είναι τοποθετημένοι στην πραγματική τους θέση. Στο τέλος του πρώτου αυτού μέρους του αλγορίθμου, για το κάθε Orders1 που αντιπροσωπεύει μια διαφορετική παραγγελία υπολογίζεται με τη βοήθεια του "Ratliff Rosenthal Algorithm" η ελάχιστη απόσταση για την συλλογή των προϊόντων της κάθε παραγγελίας, η διαδρομή που πρέπει να γίνει για να επιτευχθεί αυτή με το "Ratliff Rosenthal Plot Picking tour", καθώς και η τελική συνολική απόσταση για την εκτέλεση όλων των δρομολογίων των παραγγελιών.

Οι ευρετικές μέθοδοι που χρησιμοποιούνται για την αλλαγή της διάταξης των προϊόντων περιγράφονται αναλυτικά παρακάτω:

1. Αρχικά, επιλέγεται να τοποθετηθούν τα προϊόντα στις διάφορες θέσεις, κατά σειρά μειούμενης συχνότητας σε κάθε στήλη. Για αυτό το σκοπό, λοιπόν, δημιουργείται ο πίνακας Frequency που περιέχει τις διάφορες συχνότητες των προϊόντων στις αντίστοιχες θέσεις τους. Η δημιουργία του πίνακα αυτού γίνεται από την πρόσθεση των επιπέδων του πίνακα ddd και κατόπιν την διαίρεση με το max του κάθε (i,j) στοιχείου. Στη συνέχεια γίνεται κατάταξη αυτών των συχνοτήτων ανά στήλη κατά μειούμενη σειρά και κρατώντας το index με το οποίο μετακινούνται τα διάφορα στοιχεία του Frequency, το εφαρμόζουμε και στον πίνακα SKUs και το αποτέλεσμα μας δίνει έναν καινούργιο πίνακα (SKU_new) ο οποίος έχει τα προϊόντα τοποθετημένα στις στήλες τους κατά σειρά μειούμενης συχνότητας.

2. Η δεύτερη ευρετική μέθοδος που χρησιμοποιήθηκε ακολουθεί την ίδια λογική με την πρώτη με την διαφορά ότι εφαρμόζεται σε κάθε γραμμή της αποθήκης. Έτσι, λοιπόν, κατά τρόπο αντίστοιχο και χρησιμοποιώντας πάλι τον πίνακα Frequency, κάνοντας μια κατανομή των προϊόντων στις διάφορες θέσεις κατά σειρά μειούμενης συχνότητας ανά γραμμή και χρησιμοποιώντας το κατάλληλο index , δημιουργείται ο πίνακας SKU_new2. Να σημειωθεί εδώ ότι η αλλαγή γίνεται παίρνοντας ως βάση την ήδη τροποποιημένη διάταξη προϊόντων με τον πρώτο τρόπο.
3. Η τρίτη μέθοδος αλλαγής των θέσεων των προϊόντων στηρίζεται στην συχνότητα σχετικής εμφάνισης διάφορων ζευγαριών από αυτά στις διαθέσιμες παραγγελίες . Επιλέγονται για τον σκοπό αυτό οι πιο συχνά εμφανιζόμενοι κωδικοί , ώστε να έχουν νόημα οι μετακινήσεις (π.χ. με συχνότητα εμφάνισης > 15) . Το διάνυσμα που περιέχει τους κωδικούς αυτούς είναι το Frequent_Codes και σχηματίζεται επιλέγοντας από τον πίνακα Frequency τους κωδικούς με τιμή μεγαλύτερη του 15. Στη συνέχεια, με μια επαναληπτική διαδικασία εντοπίζονται οι παραγγελίες στις οποίες εμφανίζονται οι Frequent_Codes και σημειώνεται 1 στην αντίστοιχη γραμμή της παραγγελίας που εμφανίζεται ο εκάστοτε κωδικός . Έτσι δημιουργείται ο δυαδικός πίνακας Table_new που έχει γραμμές τις παραγγελίες και κάθε στήλη αντιστοιχεί σε έναν Frequent_Code και εμφανίζει για κάθε κωδικό τις παραγγελίες στις οποίες εμφανίζεται(με την ύπαρξη ή μη του στοιχείου 1 στην κάθε γραμμή). Κατόπιν σχηματίζεται ο πολυεπίπεδος πίνακας Sums με τη βοήθεια του προηγούμενου πίνακα . Κάθε επίπεδο του αντιστοιχεί σε ένα από τα στοιχεία του Frequent_Codes και κάθε στήλη αντιστοιχεί στους διάφορους κωδικούς με τους οποίους “ζευγαρώνει” κάθε φορά το εκάστοτε στοιχείο, ενώ οι γραμμές συνεχίζουν να αντιστοιχούν στις παραγγελίες. Έτσι , λοιπόν τα ζευγάρια που εμφανίζονται ταυτόχρονα στις παραγγελίες εμφανίζουν τον αριθμό 2 στον πίνακα Sums και αυτά είναι που μας ενδιαφέρουν για παρακάτω. Εύκολα έτσι προκύπτει και ο πίνακας Condition που είναι ακριβώς ίδιος με τον Sums με την μόνη διαφορά ότι όπου υπάρχει 2 σε στοιχείο του Sums αντικαθίσταται με 1 στον Condition. Έπειτα δημιουργείται το πολυεπίπεδο διάνυσμα γραμμή Relativity που περιέχει τα αθροίσματα κάθε στήλης του Condition και έτσι τελικά έχουμε έναν αριθμό για κάθε ζευγάρι των Frequent_Codes που αντιπροσωπεύει την ταυτόχρονη συχνότητα εμφάνισης τους στις παραγγελίες. Συνεχίζοντας , βρίσκεται για τον κάθε κωδικό το ζευγάρι του με το οποίο έχει το μεγαλύτερο Relativity και αυτό αποτυπώνεται στον πίνακα Max_Relativity. Τελειώνοντας, λοιπόν, για να σχηματιστεί το καινούργιο SKU assignment χρησιμοποιείται τελικά το Max_Relativity που υπολογίστηκε και τα ζευγάρια στα οποία αντιστοιχεί το Max_Relativity έρχονται δίπλα – δίπλα, δηλαδή οι κωδικοί στους οποίους αντιστοιχεί αυτό είναι σε

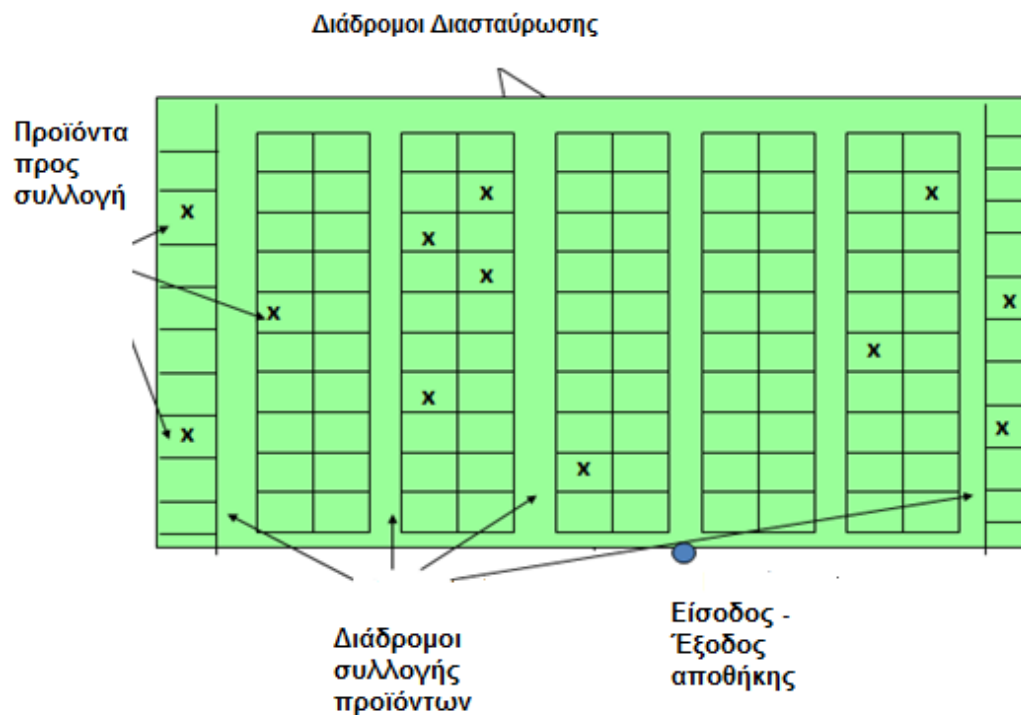
διπλανές θέσεις στον χώρο της αποθήκης. Έτσι, δημιουργείται μια νέα διάταξη των κωδικών στον χώρο η οποία ονομάζεται SKU_new3. Να σημειωθεί ότι η αλλαγή αυτή γίνεται λαμβάνοντας ως βάση το αλλαγμένο ήδη 2 φορές από τις προηγούμενες μεθόδους SKU assignment.

4. Τα ζευγάρια που δημιουργούνται στην περίπτωση 3 , είναι σχηματισμένα με τυχαίο τρόπο και υπάρχουν πολλοί άλλοι συνδυασμοί που δεν λαμβάνονται υπόψη . Ένας τρόπος για να παραχθεί μια νέα σειρά ζευγαριών και να τοποθετηθούν αυτά με το μεγαλύτερο Relativity διαδοχικά είναι να γίνει μια ανακατανομή των Frequent_Codes και να τοποθετηθούν στο διάνυσμα γραμμή κατά μειούμενη τιμή Frequency. Έτσι , δημιουργούνται νέα ζεύγη που οδηγούν και σε νέα αποτελέσματα κατόπιν εφαρμογής της διαδικασίας που περιγράφηκε στην μεθοδολογία 3 και παράγεται το SKU_new4.
5. Ένας επιπλέον τρόπος δημιουργίας καινούργιων ζευγαριών είναι ο εξής : Εφαρμόζεται μια επαναληπτική διαδικασία στην οποία για κάθε καινούργια επανάληψη οι κωδικοί στο διάνυσμα γραμμή Frequent_Codes μετατοπίζονται κατά μια θέση δεξιά . Έτσι μετά από 37 επαναλήψεις οι κωδικοί έχουν πάρει όλες τις πιθανές θέσεις και έχουν εξεταστεί 37 διαφορετικοί συνδυασμοί ζευγαριών και ταυτόχρονα έχουν παραχθεί 37 διαφορετικά SKU assignments τα οποία είναι τοποθετημένα σε κάθε μία από τις στήλες του διανύσματος γραμμής SKU_new5 . Με αυτό τον τρόπο παράγονται πολλά καινούργια αποτελέσματα με την επαναλαμβανόμενη χρήση της μεθοδολογίας 3.

Στη συνέχεια, δημιουργούνται για την κάθε περίπτωση νέοι διαφορετικοί πίνακες ddd_new. Αυτοί αντιπροσωπεύουν ότι και το ddd που προαναφέρθηκε , με τη διαφορά όμως ότι οι παραγγελίες που περιέχονται σε κάθε επίπεδο τους είναι φτιαγμένες με βάση την καινούργια διάταξη στο χώρο για την κάθε περίπτωση. Μετά την διαδικασία αυτή είναι αναγκαία η δημιουργία της κάθε μιας παραγγελίας ξεχωριστά , ώστε να υπολογιστούν οι επιμέρους αποστάσεις και στο τέλος να προστεθούν για να βγει η συνολική. Αυτό επιτυγχάνεται με τον πίνακα Order_new που είναι ο αντίστοιχος του Order1 στο πρώτο μέρος του αλγορίθμου και ουσιαστικά . Αυτή η διαδικασία επαναλαμβάνεται πολλές φορές , για όλες τις παραγγελίες και ουσιαστικά κάθε Order_new της κάθε διαφορετικής επανάληψης είναι ένα επίπεδο του ddd_new. Μετά, λοιπόν, με τη βοήθεια του “Ratliff Rosenthal Algorithm” υπολογίζεται η επιμέρους, αλλά και η συνολική απόσταση για το κάθε SKU assignment ξεχωριστά.

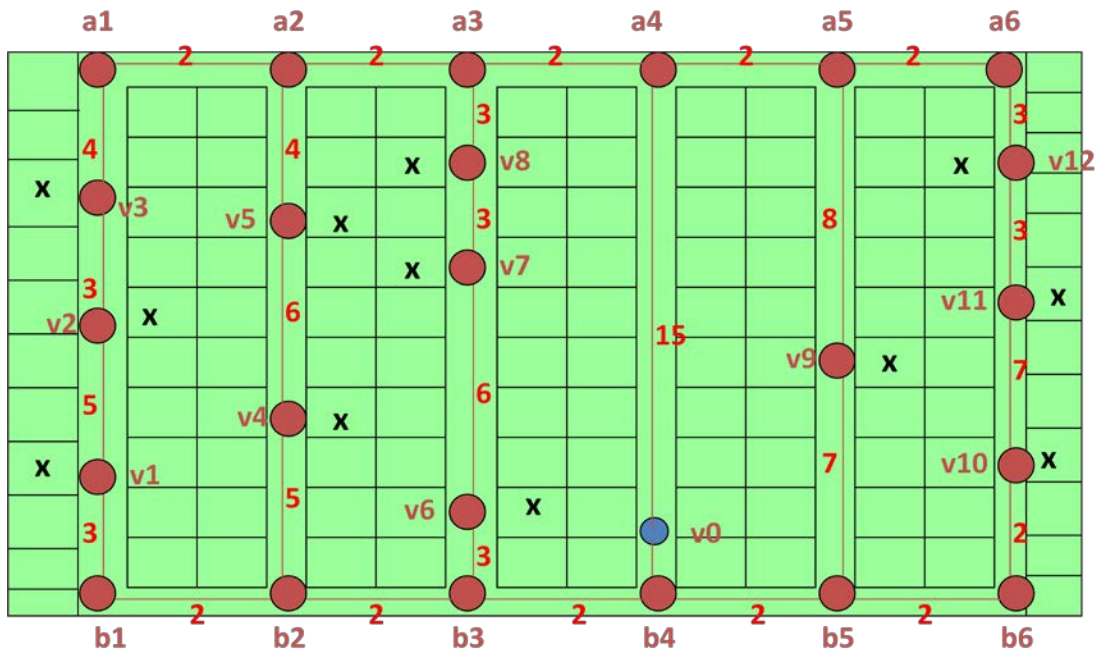
Περιγραφή του αλγόριθμου με τη βοήθεια του οποίου υπολογίζονται οι ελάχιστες αποστάσεις

Ο “Ratliff and Rosenthal Algorithm” είναι ο αλγόριθμος που υπάρχει στη βιβλιογραφία [32] και χρησιμοποιήθηκε στην παρούσα εργασία για τον υπολογισμό των βέλτιστων αποστάσεων που μπορούν να διανύσουν οι συλλέκτες των προϊόντων μιας παραγγελίας. Ο αλγόριθμος αυτός(όπως φαίνεται και στο Σχήμα 21) μπορεί να εφαρμοστεί στις περισσότερες από τις κλασσικές αποθήκες για να επιλύσει το TSP.

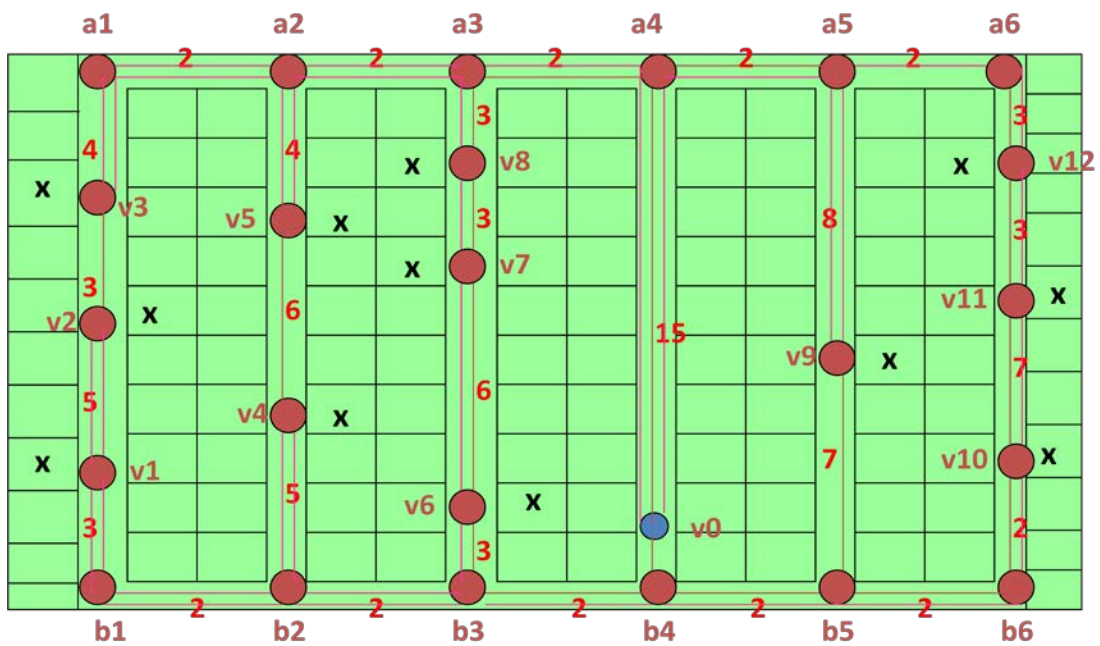


Σχήμα 22 Προτιμώμενο σχήμα αποθήκης για εφαρμογή του αλγορίθμου.

Η μέθοδος αυτή στηρίζεται στο ότι ο συλλέκτης των προϊόντων πρέπει να περάσει από όλα τα σημεία που φαίνονται σαν κουκίδες στο Σχήμα 22 και αντιπροσωπεύουν τα προϊόντα που πρέπει να συλλεχθούν στην εκάστοτε παραγγελία. Σαν κουκίδες όμως φαίνονται και οι κόμβοι από τους οποίους έχει τη δυνατότητα να περάσει ο συλλέκτης για να μαζέψει τα επόμενα προϊόντα. Ενώνοντας, λοιπόν, τα σημεία των προϊόντων μέσω κόμβων δημιουργείται μια διαδρομή για την κάθε παραγγελία(βλ. Σχήμα 23).

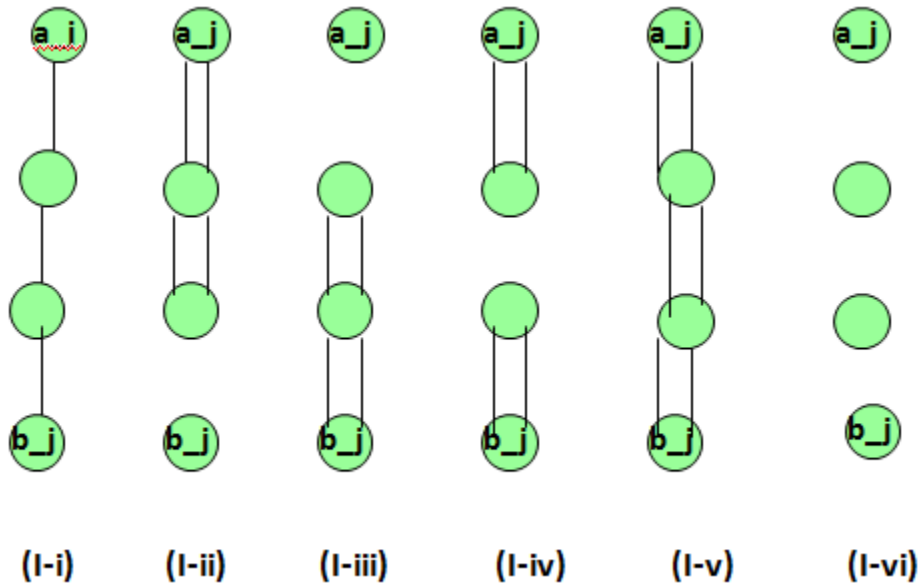


Σχήμα 23 Απεικόνιση της αποθήκης με τα σημεία κουκίδες που αντιπροσωπεύουν κόμβους και προϊόντα.



Σχήμα 24 Μια διαδρομή για τη συλλογή των προϊόντων μιας παραγγελίας.

Επειδή, όμως, οι συλλέκτες μπορούν να διαλέξουν διάφορους τρόπους με τους οποίους θα συλλέξουν τα προϊόντα ή να διαφέρουν οι κόμβοι από τους οποίους περνάνε, δημιουργούνται πολλές πιθανές διαδρομές για μια παραγγελία(βλ. Σχήμα 24). Αυτό που κάνει ο αλγόριθμος, λοιπόν, είναι να υπολογίζει τις συνολικές αποστάσεις για κάθε εναλλακτική και να επιλέγει αυτή με την μικρότερη.



Σχήμα 25 Διαφορετικές διαδρομές για την ένωση των σημείων a,b.

Ειδικότερα για την επίλυση του προβλήματος μας, λόγω μιας ιδιαιτερότητας της αποθήκης, η μοναδική διαδρομή που μπορεί να γίνει για την συλλογή των προϊόντων στις διάφορες παραγγελίας μοιάζει με την περίπτωση (I-v) του Σχήματος. Αυτό συμβαίνει, διότι το σχήμα της αποθήκης είναι αυτό που έχει παρουσιαστεί και παραπάνω στο Σχήμα 25 [32].

Τμήματα του κώδικα που χρησιμοποιήθηκαν σαν ορολογίες παραπάνω

Παρακάτω παρουσιάζεται σε μορφή κώδικα το script “Changes” που αναφέρεται στο Σχήμα 11 και είναι η 1^η μέθοδος με βάση την οποία αλλάζει η διάταξη των προϊόντων στον χώρο:

```

Fr=zeros(32,18)
for i=1:size(SKUs,1)
    for j=1:size(SKUs,2)
        for k=1:size(ddd,3)

            Fr(i,j)=Fr(i,j)+ddd(i,j,k);
        end

        if SKUs(i,j)>0
            Frequency(i,j)=Fr(i,j)/SKUs(i,j);
        else
            Frequency(i,j)=0;
        end

    end
end

[temp,new] = sort(Frequency,'descend')

for j =1:size(SKUs,2)
    SKUs_new(:,j) = SKUs(new(:,j),j);
end
B=zeros(1,size(SKUs,2));

SKU_new=[B ; SKUs_new(1:31,:)];
ddd_new=ddd;
for i=1:size(SKUs,1)
    for j=1:size(SKUs,2)
        for k=1:3:size(ddd,3)
            for l=1:size(SKUs,1)
                if ddd(i,j,k)>0 & ddd(i,j,k)==SKU_new(l,j)
                    ddd_new(l,j,k)=SKU_new(l,j);
                    ddd_new(i,j,k)=0;
                end
            end
        end
    end

end
end
end
Order_new=SKUs*0;

```

```

for i=1:size(SKUs,1)
    for j=1:size(SKUs,2)
        for k=1:3:size(ddd,3)

            if ddd_new(i,j,k)>0
                Order_new(i,j,k)=1;
            end
        end
    end
end
end
end

```

Παρακάτω παρουσιάζεται η 2^η μέθοδος που χρησιμοποιήθηκε για την αλλαγή της διάταξης των προϊόντων στον χώρο :

```
[SKUs_1,new1]=sort(Frequency,2,'descend');
```

```

SKUs_2=[SKUs_1(:,1) zeros(size(SKUs_1,1),1) SKUs_1(:,2:3)
zeros(size(SKUs_1,1),1) SKUs_1(:,4:5) zeros(size(SKUs_1,1),1)
SKUs_1(:,6:7) zeros(size(SKUs_1,1),1) SKUs_1(:,8:9)
zeros(size(SKUs_1,1),1) SKUs_1(:,10:11)
zeros(size(SKUs_1,1),2)];

```

```

temp1=[zeros(1,size(temp,2)); temp];
[SKUs_3,new2]=sort(temp1,2,'descend');
```

```
for j =1:size(SKUs,1)
```

```

    SKUs_new2(j,:) = SKU_new(j,new2(j,:));
end

```

```

SKU_new2=[SKUs_new2(:,1) zeros(size(SKUs_1,1),1)
SKUs_new2(:,2:3) zeros(size(SKUs_1,1),1) SKUs_new2(:,4:5)
zeros(size(SKUs_1,1),1) SKUs_new2(:,6:7)
zeros(size(SKUs_1,1),1) SKUs_new2(:,8:9)
zeros(size(SKUs_1,1),1) SKUs_new2(:,10:11)
zeros(size(SKUs_1,1),2)];

```

Συνεχίζοντας παρουσιάζεται ο 3^{ος} τρόπος αλλαγής της διάταξης των προϊόντων στον χώρο της αποθήκης:

```
k=1;
for i=1:size(SKUs,1)
    for j=1:size(SKUs,2)
        if Frequency(i,j)>15
            Frequent_Codes(1,k)=SKUs(i,j);
            Frequencies(1,k)=Frequency(i,j);
            k=k+1;
        end
    end
end
Table_new=zeros(size(ddd,3),size(Frequent_Codes,2));
for k=1:size(Frequent_Codes,2)
    for l=1:3:size(ddd,3)
        for i=1:size(SKUs,1)
            for j=1:size(SKUs,2)
                if ddd(i,j,l)==Frequent_Codes(1,k)
                    Table_new(l,k)=1;
                end
            end
        end
    end
end
Orders_Codes=[Frequent_Codes(1,:); Table_new];
Sums=zeros(size(ddd,3),size(Frequent_Codes,2),size(Frequent_Codes,2)-1);
for k=1:size(Frequent_Codes,2)-1
    for l=1:size(Frequent_Codes,2)-k
        for i=1:size(ddd,3)
            Sums(i,l,k)=Table_new(i,k)+Table_new(i,k+l);
        end
    end
end
```

```
Condition=Sums*0;
for i=1:size(ddd,3)
    for j=1:size(Frequent_Codes,2)
        for k=1:size(Frequent_Codes,2)-1
            if Sums(i,j,k)==2
                Condition(i,j,k)=1;
            end
        end
    end
end
end
```

```
Relativity=sum(Condition);
Final_Condition=[Condition ; Relativity];
Max_Relativity=max(Relativity,[],2);
```



```

for j =1:size(Frequent_Codes,1)

    Frequent_Codes2(j,:) = Frequent_Codes(j,new4(j,:));
end

Table_new=zeros(size(ddd,3),size(Frequent_Codes,2));
for k=1:size(Frequent_Codes,2)
    for l=1:3:size(ddd,3)
        for i=1:size(SKUs,1)
            for j=1:size(SKUs,2)
                if ddd(i,j,l)==Frequent_Codes2(1,k)
                    Table_new(l,k)=1;
                end
            end
        end
    end
end
Orders_Codes=[Frequent_Codes2(1,:) ; Table_new] ;
Sums=zeros(size(ddd,3),size(Frequent_Codes,2),size(Frequent_C
odes,2)-1);
for k=1:size(Frequent_Codes,2)-1
    for l=1:size(Frequent_Codes,2)-k
        for i=1:size(ddd,3)
            Sums(i,l,k)=Table_new(i,k)+Table_new(i,k+l);
        end
    end
end

Condition=Sums*0;
for i=1:size(ddd,3)
    for j=1:size(Frequent_Codes,2)
        for k=1:size(Frequent_Codes,2)-1
            if Sums(i,j,k)==2
                Condition(i,j,k)=1;
            end
        end
    end
end

Relativity=sum(Condition);
Final_Condition=[Condition ; Relativity];
Max_Relativity=max(Relativity,[],2);

SKU_new4=SKU_new2;

```


Παρακάτω παρουσιάζεται και η τελευταία μεθοδολογία αλλαγής της διάταξης των προϊόντων στον χώρο της αποθήκης που δίνει και το τελικό αποτέλεσμα:

```
Frequent_Codes3=zeros(1,size(Frequent_Codes,2));
SKU_newX=SKU_new2
for jj=1:size(Frequent_Codes,2)-1
for i=1:size(Frequent_Codes,2)-1
Frequent_Codes3(1,i+1)=Frequent_Codes(1,i);
Frequent_Codes3(1,1)=Frequent_Codes(1,size(Frequent_Codes,2)
));
end
```

```
Table_new=zeros(size(ddd,3),size(Frequent_Codes,2));
for k=1:size(Frequent_Codes,2)
for l=1:3:size(ddd,3)
for i=1:size(SKUs,1)
for j=1:size(SKUs,2)
if ddd(i,j,l)==Frequent_Codes3(1,k)
Table_new(l,k)=1;
end
end
end
end
end
Orders_Codes=[Frequent_Codes3(1,:) ; Table_new] ;
Sums=zeros(size(ddd,3),size(Frequent_Codes,2),size(Frequent_C
odes,2)-1);
for k=1:size(Frequent_Codes,2)-1
for l=1:size(Frequent_Codes,2)-k
for i=1:size(ddd,3)
Sums(i,l,k)=Table_new(i,k)+Table_new(i,k+l);
end
end
end
```

```
Condition=Sums*0;
for i=1:size(ddd,3)
    for j=1:size(Frequent_Codes,2)
        for k=1:size(Frequent_Codes,2)-1
            if Sums(i,j,k)==2
                Condition(i,j,k)=1;
            end
        end
    end
end
end
```

```
Relativity=sum(Condition);
Final_Condition=[Condition ; Relativity];
Max_Relativity=max(Relativity,[],2);
```



```

ddd_new5=ddd_new2;
for i=1:size(SKUs,1)
    for j=1:size(SKUs,2)
        for k=1:3:size(ddd,3)
            for l=1:size(SKUs,1)
                if ddd_new2(i,j,k)>0 & ddd_new2(i,j,k)==SKU_newX(l,j)
                    ddd_new5(l,j,k)=SKU_newX(l,j);
                    ddd_new5(i,j,k)=0;
                end
            end
        end
    end
end
end
Order_new=SKUs*0;
for i=1:size(SKUs,1)
    for j=1:size(SKUs,2)
        for k=1:3:size(ddd,3)

            if ddd_new5(i,j,k)>0
                Order_new(i,j,k)=1;
            end
        end
    end
end
end

Total_distanceX=0;

for k=1:3:size(allorders1-2,2)

    for i=1:size(Orders1,1)
        for j=1:size(Orders1,2)

            Orders1(i,j)=Order_new(i,j,k);

        end
    end
end
Ratliff_Rosenthal_Algorithm_New

    Total_distanceX=Total_distanceX+Total_distance
    Total_distance
end
Total_distanceX
Total_distance6(1,jj)=Total_distanceX

```

```
for i=1:size(Frequent_Codes,2)
Frequent_Codes_Table(jj,i)=Frequent_Codes3(1,i);
end
Frequent_Codes=Frequent_Codes3;
end

[Best_solutions,IX]=sort(Total_distance6,'descend');
```

Χρήσιμο, τέλος, είναι να παρουσιαστεί και το script “Position” το οποίο αναφέρεται παραπάνω στο παράρτημα στην εκτενής περιγραφή του αλγορίθμου:

```
allorders1=xlsread('allorders1')

all=allorders1(:,k:k+2);

E1=all(:,2)/100;
v=floor(E1);

F1=E1-v;
```



```

Pos1=round(Pos1a);

for i=1:size(Pos1,1)
    L1=Pos1(i,1)+1;
    L2=Pos1(i,2);

    Codes1(L1,L2)=all(i,1);
    Orders1(L1,L2)=1;
end

for i=1:32;
    for j=1:18;

        ddd(i,j,k)=Codes1(i,j);
    end
end

SUM = 0;
for i=1:32
    for j=1:18

        SKUs(i,j)=max(ddd(i,j,:));

        if SKUs(i,j)==0
            SUM=SUM;
        else
            SUM=SUM+1;
        end
    end
end
end

```

```
SKUx=zeros(32,18);
for i=1:31
    for j=1:17
        for k=i:18
            for l=j+1:18
                if SKUs(i,j)==SKUs(k,l) & SKUs(i,j)~=0
                    SKUx(i,j)=5;
                end
            end
        end
    end
end
```

BIBΛΙΟΓΡΑΦΙΑ

- 1) John J. Bartholdi, III, Steven T. Hackman, "Warehouse and Distribution Science" Release 0.95, The Supply Chain and Logistics Institute, School of System and Industrial Engineering, Georgia Tech.
- 2) Gianpaolo Ghiani, Gilbert Laporte, Roberto Musmanno, "Introduction to Logistics Systems Planning and Control".
- 3) Eudoxus Systems Ltd. , LP Training , "What is Mathematical Programming ?" (<http://www.eudoxus.com/>)
- 4) Article by ScienceDaily , "Mathematical Model"
- 5) G.Cornevelis van Kooten, "Introduction to Mathematical and Linear Programming".
- 6) Auburn University Academic Classes , "Introduction to Mathematical Programming", AGEC 7100(Notes, Set 1).
- 7) S. Dempe, S. Franke, "Bilevel Optimization Problems with Vectorvalued Objective Functions in Both Levels" , Paper by TU Bergakademie Freiberg, Germany.
- 8) Chris Fricke, "An Introduction to Bilevel Programming", Department of Mathematics and Statistics, University of Melbourne.
- 9) Benoit Colson, Patrice Marcotte, Gilles Savard, Revised version: May 2005 , "An Overview of Bilevel Optimization" , Department of Mathematics, The University of Namur , Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Département de mathématiques et de génie industriel, Ecole Polytechnique de Montréal.
- 10) Dave Piasecki , "Order Picking : Methods and Equipment for Piece Pick, Case Pick and Pallet Pick Operations", Article , Inventory Operations Consulting LLC.
- 11) De Koster, R., Le-Duc, T., and Roodbergen, K.J. ,2007, "Design and control of warehouse order picking: a literature review", European Journal of Operational Research 182(2), 481-501.
- 12) Jean-Philippe Gagliardi, Angel Ruiz, Jacques Renaud, April 2008, "Space Allocation and Stock Replenishment Synchronization in a Distribution Center", Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Faculté des Sciences de l'administration, Laval University, Canada.
- 13) Michael Hahsler, Kurt Hornik , "TSP- Infrastructure for the Traveling Salesperson Problem", Southern Methodist University, Wien.
- 14) Adam N. Letchford , January 2013, "The Traveling Salesman Problem", Publications ,Lancaster University.
- 15) John E. Mitchell , April 19 - 1999, revised September 7 – 1999, "Branch and Cut Algorithms for Combinatorial Optimization Problems", Mathematical Sciences Rensselaer Polytechnic Institute Troy, NY, USA

- 16) Jens Clausen , March 12 – 1999, “Branch and Bound Algorithms Principles and Examples” , Department of Computer Science, University of Copenhagen.
- 17) Tony Oktatas, 6 April 2011, “The Branch and Bound Method” , SecondExpert, Chapter 24.
- 18) Mohan Akella, Sharad Gupta, Avijit Sarkar, “Branch and Price , Column Generation for Solving Huge Integer Programs”, University at Buffalo , Department of Industrial Engineering.
- 19) Jacques Desrosiers, Marco E.Lubbecke, April 8-2010 , revised July 7-2010, “Branch Price and Cut Algorithms”, HEC Montreal and GERAD , Technische Universit“at Darmstadt.
- 20) Adam N. Letchford, September 2011, “An Introduction to Branch and Price Part II: Algorithms and Applications” , Department of Management Science , Lancaster University.
- 21) John W. Chinneck, 2006, “Practical Optimization : a Gentle Introduction , Chapter 14 : Heuristics for Discrete Search : Genetic Algorithms and Simulated Annealing” .
- 22) Fred Glover and Rafael Marti, “Tabu Search”, Paper , Chapter 4 ,Universitat de Valencia.
- 23) Glover , 1989-1990, “Tabu Search” Part 1 and 2 ORSA J. Comp. 1 & 2.
- 24) Felix Streichert, “Introduction to Evolutionary Algorithms” , Publications ,University of Tuebingen.
- 25) Ματζαβάκης Κώστας , “Warehouse Logistics” , Wordpress
- 26) H. v. Stackelberg, “English translation: The theory of market economy”, Marktform und Gleichgewicht, Springer Verlag, Berlin, 1934, Oxford University Press, 1952.
- 27) Bracken and McGill, 1973, “Operations Research”, Vol. 21.
- 28) Tompkins JA, White JA, Bozer YA, Frazelle EH, Tanchoco JMA and Trevino J., “Facilities Planning”. New York: John Wiley & Sons, Inc. 2nd edn. 1996.
- 29) P.J. van Laarhoven,E.H. Aarts, “Simulated Annealing: Theory and Applications”.
- 30) Dipankar Dasgupta, Zbigniew Michalewicz, “Evolutionary Algorithms in Engineering Applications”.
- 31) Papadimitriou, C.H.; Steiglitz, K. (1998), “Combinatorial optimization: algorithms and complexity”, Mineola, NY: Dover , pp.308-309.
- 32) H. Donald Ratliff and Arnon S. RosenthalSource, “ Order-Picking in a Rectangular Warehouse: A Solvable Case of the TSP”, Operations Research, Vol. 31, No. 3, (May - Jun., 1983), pp. 507-521.
- 33) Γιώργος Κοζανίδης, Ευτυχία Κωσταρέλου, “Μεικτός Ακέραιος Διεπίπεδος Προγραμματισμός για βέλτιστη υποβολή προσφορών σε αγορές ημερήσιου προγραμματισμού ηλεκτρικής αδαιρετότητες”, Τμήμα Μηχ/γων Μηχανικών, Πανεπιστήμιο Θεσσαλίας.