Εξελικτική Υπολογιστική - Αλγόριθμοι εμπνευσμένοι από τη Βιολογία



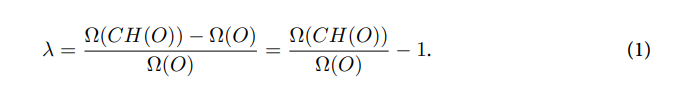
Κωνσταντίνος Χατζησταύρος 497

Καλομιτσίνης Γεώργιος 485

1. *Εισαγωγή*

Η παρούσα εργασία διαπραγματεύεται τον σχεδιασμό μιας βέλτιστης συνάρτησης καταλληλότητας, η οποία ως αποτέλεσμα θα βελτιστοποιεί το πρόβλημα της κοπής παραγγελιών ξύλου(order), από ένα αρχικά ορισμένο απόθεμα(stock). Συνεπώς οι αντικειμενικοί στόχοι που πρέπει να παρθούν υπόψη κατά την σχεδίαση είναι αφενός το απόθεμα να κόβεται με τρόπο που μπορεί να επαναξιοποιηθεί για την εκάστοτε επόμενη παραγγελία, και αφετέρου να ελαχιστοποιείται η απόσταση μεταξύ των διαφόρων κοπών ξύλου από το απόθεμα(δηλαδή να είναι συμπαγή τα κομμένα τμήματα μεταξύ τους πάνω στο αρχικό απόθεμα).

Στο σύνολο της εργασίας χρησιμοποιούνται οι εξελικτικοί αλγόριθμοι DynPSO και DEGL βάση της εκφώνησης. Προκειμένου να μοντελοποιηθούν τα συμπαγή πολύγωνα στο αρχικό απόθεμα, χρησιμοποιήθηκαν οι εξισώσεις (1) και (2) οι οποίες δίνονται στην εκφώνηση της άσκησης.





Επιπλέον, το πρόβλημα χρειάζεται να παραμετροποιηθεί σε σχέση με τις παραγγελίες και συγκεκριμένα με τον τρόπο οι οποίες θα τοποθετούνται στο απόθεμα. Συνεπώς η κάθε λύση του είναι συσχετισμένη και εξαρτώμενη από την θέση x,y και την γωνία περιστροφής θ του κάθε πολυγώνου. Η τελική λύση περιλαμβάνει τα [x,y,θ] για κάθε σχήμα που μετατοπίζεται - περιστρέφεται από την αρχή των αξόνων προκειμένου να χωρέσουν αποδοτικά όλες οι παραγγελίες. Τέλος το πλήθος των μεταβλητών για την επίλυση του προβλήματος, είναι 3\*(αριθμό παραγγελιών).

1. *Σχεδιασμός συνάρτησης καταλληλότητας - fit function*

Σε αυτό το κεφάλαιο θα παρουσιαστούν και θα αναλυθούν οι διάφοροι όροι της συνάρτησης καταλληλότητας σε μια προσπάθεια ώστε το αποτέλεσμα αυτής να μην αποτελεί αποκλειστικά προϊόν απλού αθροίσματος. Επιπλέον τα βάρη του κάθε όρου δίνονται στο τέλος προκειμένου να διαμορφωθεί η συνάρτηση καταλληλότητας.

Σχετικά με την κοπή των πολυγώνων τα οποία βρίσκονται εκτός ορίων του αποθέματος εισάγεται ένα όρος που θα αποτρέπει αυτή τη συμπεριφορά. Καθώς ο αλγόριθμος στοχεύει στην ελαχιστοποίηση της εξόδου της συνάρτησης καταλληλότητας, καλούμαστε να ορίσουμε το εκάστοτε άθροισμα του εμβαδού των σχημάτων που βρίσκεται έξω από το απόθεμα. Στην πράξη υπολογίζεται η ένωση των σχημάτων βάση των θέσεων και των στροφών που έχουν αποκτήσει ενώ στην πορεία υπολογίζεται η διαφορά της ένωσης με το απόθεμα. Στα σχετικά αποτελέσματα θα φανούν παραδείγματα όπου τοποθετούνται σχήματα εκτός αποθέματος, ενώστην python αυτό συμβαίνει μέσω των :

***unionNewOrder = shapely.ops.cascaded\_union(newOrder)***

***outOfStock = unionNewOrder.difference(Stock)***

Απαραίτητη προϋπόθεση για την συνέχεια είναι ο ορισμός τους όρου μέσω του οποίου θα αποτραπεί οποιαδήποτε επικάλυψη μεταξύ των πολυγώνων των παραγγελιων. Επομένως είναι λογικό να ακολουθεί ο υπολογισμός του εμβαδού των σχημάτων που επικαλύπτεται από τις διάφορες παραγγελίες. Συνεπώς απαιτείται ο υπολογισμός του αθροίσματος των εμβαδών της εκάστοτε παραγγελίας και το εμβαδόν της ένωσης των επιμέρους σχημάτων της παραγγελίας με τις τοποθετήσεις από την προηγούμενη λύση. Έπειτα μένει να δημιουργηθεί ο όρος από την αφαίρεση του εμβαδού του αθροίσματος μείον το εμβαδόν της ένωσης, που πρακτικά στην python γίνεται ως εξής :

***areaSum = sum([newOrder[w].area for w in range(0, len(newOrder))])***

***overlapArea = areaSum - unionNewOrder.area***

Στην πορεία υλοποιήθηκε ένας όρος - στόχος που αφορά την τοποθέτηση των σχημάτων προς την αρχή των αξόνων, χρησιμοποιώντας το εμβαδόν των σχημάτων. Σε αυτό το βήμα υπολογίζεται το άθροισμα των x και y των centroid των σχημάτων πολλαπλασιασμένο με το εμβαδόν του εκάστοτε σχήματος. Χρησιμοποιώντας τις συντεταγμένες x,y ουσιαστικά εκφράζεται με απλούστερο τρόπο η έλξη προς την αρχή των αξόνων, από την περίπτωση που θα υπολογιστεί η απόσταση.

***Dist\_from\_zero = sum([newOrder[i].area\*(newOrder[i].centroid.x+newOrder[i].centroid.y) for i in range(0, len(newOrder))])***

Έπειτα χρησιμοποιήθηκε ο τύπος του fsm που περιγράφει το πόσο συμπαγές είναι το απόθεμα μετά την αφαίρεση της παραγγελία για τις νέες θέσεις, δηλαδή :

***ch = (remaining.convexHull)***

***lamba = (ch.area)/(remaining.area) - 1***

***alpha = 1.11***

***fsm = 1/(1+alpha\*lambda)***

Στο πλαίσιο των δοκιμών που έγιναν για τον καθορισμό των βαρών των όρων, καταλήξαμε σε μια τελική μορφή της συνάρτησης καταλληλότητας. Η επιλογή των βαρών ουσιαστικά βασίζεται στο πόσο απαραίτητη προϋπόθεση αποτελούν οι εκάστοτε όροι της τελικής συνάρτησης καταλληλότητας. Επομένως, καθώς οι κύριες απαιτήσεις αφορούν να μην υπάρχει πολύγωνο εκτός αποθέματος καθώς και χωρίς επικάλυψη στην τοποθέτηση των πολυγώνων, αυτοί οι δύο αντίστοιχοι όροι λαμβάνουν μεγάλες τιμές βάρους. Παράλληλα λαμβάνεται υπόψην πως η κάθε προτεινόμενη τοποθέτηση, προκειμένου να γίνει όντως αποδεκτή, υπάρχει μια ανοχή επικάλυψης και παραγγελίας εκτός αποθέματος, της τάξης 10-4. Έπειτα το επόμενο βάρος που πρέπει να προσδιοριστεί αφορά τον όρο της συμπαγούς τοποθέτησης των πολυγώνων στον οποίο και δίνεται μικρότερο βάρος. Ουσιαστικά αυτό είναι λογικό καθώς αποτελεί σχεδιαστική προτίμησης στον υπολογισμό της εξόδου της συνάρτησης καταλληλότητας και όχι απαίτηση για τον σχεδιασμό της. Συμπληρωματικά, όπως προαναφέρθηκε θεωρείται πως τοποθέτηση οφείλει να τείνει στην αρχή των αξόνων. Τέλος, εισάγεται τυχαιότητα στον αλγόριθμο μέσω του γνωστού seed στην python.

Βάση των παραπάνω θα παρουσιαστούν οι τελικές μορφές των συναρτήσεων καταλληλότητας για τον DynPSO και τον DEGL αντίστοιχα. Ξεκινώντας με τον πρώτο αλγόριθμο η συνάρτηση καταλληλότητας, έπειτα από πειραματισμούς στα βάρη λαμβάνει την παρακάτω μορφή και συμπεριλαμβάνει όλους τους παραπάνω όρους οι οποίοι προσδιορίζουν τις απαιτήσεις και τις προτιμήσεις που αναλύθηκαν για τον σχεδιασμό της.

***Res = outOfStock.area \* 10000 + overlapArea \* 10000 + dist\_frm\_zero +100 \* fsm***

Για την περίπτωση του DEGL η μοναδική διαφοροποίηση ως προς την συνάρτηση καταλληλότητας αφορά τα βάρη των όρων αυτής. Συγκεκριμένα ρίξαμε κατά μια τάξη μεγέθους τα μεγέθη outOfStock.area και overlapArea, ενώ τριπλασιάστηκε ο όρος που αφορά τα συμπαγή υπόλοιπα της εκάστοτε γειτονιάς. Συνεπώς η συνάρτηση καταλληλότητας παίρνει την μορφή :

***Res = outOfStock.area \* 1000 + overlapArea \* 1000 + dist\_frm\_zero +300 \* fsm***

Τέλος σχετικά με τους υπόλοιπους αλγόριθμους βελτιστοποίησης (Simplex/Nelder-Mead, Interior Point Algorithm/L-BFGS-B και SLSQP) που θα εξεταστούν, χρησιμοποιείται η ίδια συνάρτηση καταλληλότητας με αυτή που χρησιμοποιήθηκε στην περίπτωση του DEGL.

1. *Διαδικασία τοποθέτησης των παραγγελιών στο απόθεμα.*

Στο παρόν κεφάλαιο θα αναλυθεί ο αλγόριθμος που ορίζει τη διαδικασία της τοποθέτησης των παραγγελιών στο απόθεμα. Για την αξιολόγηση αποθηκεύεται ο αριθμός των πολυγωνων που χώρεσαν στο απόθεμα ενώ ορίζονται τα κατάλληλα flags για την περίπτωση που έχει χωρέσει ολόκληρη η παραγγελία στο απόθεμα. Ουσιαστικά όσο δεν είναι άδεια η λίστα των παραγγελιών, υπολογίζεται το άθροισμα των εμβαδών των πολυγώνων της παραγγελίας(currentOrderArea) και αντίστοιχα το υπόλοιπο εμβαδόν του αποθέματος(remainingsArea). Έπειτα βάση των currentOrderArea και remainingsArea ταξινομούνται σε αύξουσα σειρά τα υπολειπόμενα εμβαδά του αποθέματος τα οποία μπορούν να ικανοποιήσουν την επόμενη παραγγελία. Η λογική πίσω από το παραπάνω είναι για κάθε επόμενο κομμάτι της παραγγελίας να ελέγχει πρώτα το μικρότερο διαθέσιμο εμβαδόν του αποθέματος προκειμένου να ελαχιστοποιείται το αναξιοποίητο απόθεμα μετά την τοποθέτηση της εκάστοτε παραγγελίας.

Ως επόμενο βήμα, τα τμήματα που είναι αρκετά μεγάλα ελέγχονται σειριακά ενώ παράλληλα ορίζονται τα ανώτατα/κατώτατα όρια των [x,y,theta] βάση του εκάστοτε αποθέματος. Στο σύνολο της εργασίας έτρεξαν οι αλγόριθμοι DynPSO και DEGL καθώς και οι Nelder-Mead, L-BFGS-B και SLSQP οι οποίοι υπολογίζουν τις πιθανές τελικές θέσεις των παραγγελιών βάση των αντίστοιχων μετασχηματισμών. Όπως είναι ήδη γνωστό, πο συνάρτηση καταλληλότητας τερματίζεται όταν παύει να βελτιώνεται για ένα ορισμένο αριθμό επαναλήψεων. Παρόλα αυτά είναι πιθανό να μην βρίσκει πάντοτε τη βέλτιστη λύση για την κάθε ταξινόμηση για τους λόγους που περιγράφουν οι όροι που περιλαμβάνει. Συνεπώς υπάρχει η ανάγκη ελέγχου της κάθε λύσης που παράγεται και αυτό υλοποιείται μέσω ενός ορίου ανοχής(0,0001) από το οποίο οφείλει να είναι μικρότερο το εμβαδόν της παραγγελίας εκτός του αποθέματος. Ομοίως πραγματοποιείται έλεγχος για το εμβαδόν επικάλυψης ως προς το ίδιο όριο ανοχής . Συνεπώς, βάση αυτών των δύο συνθηκών επιβεβαιώνεται ή όχι η εκάστοτε προτεινόμενη λύση. Στην περίπτωση που η λύση είναι αποδεκτή, αφαιρείται το εμβαδόν της από το απόθεμα. Συνεχίζοντας στην περίπτωση που η παραγγελία δεν χωράει στο απόθεμα, αυτή διαιρείται στη μέση σε 2 παραγγελίες του μισού εμβαδού. Αν οποιαδήποτε παραγγελία, είτε ολόκληρη είτε τεμαχισμένη, δεν είναι δυνατόν να ικανοποιηθεί, αφαιρείται από το σύνολο των παραγγελιών. Ως μέσο αξιολόγησης αποθηκεύεται ο χρόνος εκτέλεσης, ο αριθμός των πολυγώνων που ικανοποιήθηκαν, καθώς και ο min/max/mean αριθμός επαναλήψεων, ενώ οπτικοποιούνται και τα αποτελέσματα της τοποθέτησης των σχημάτων.

1. *Αποτελέσματα*

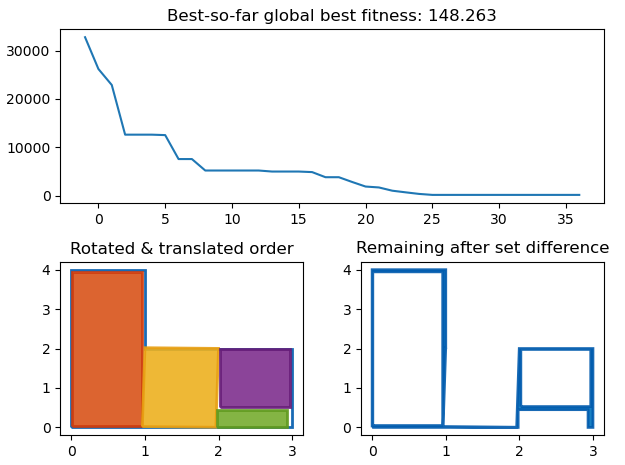
*4.1. Μέθοδος Αξιολόγησης*

Για κάθε μια από τις παραγγελίες εκτελούνται οι αλγόριθμοι DynPSO και DEGL με την σχεδιασμένη συνάρτηση καταλληλότητας, προκειμένου να εξαχθούν συμπεράσματα σχετικά με την ικανότητα ορθολογικής τοποθέτησης στο εκάστοτε απόθεμα. Αντίστοιχα για τους αλγορίθμους Nelder-Mead, L-BFGS-B και SLSQP θα παρουσιαστούν οι τελικές θέσεις των παραγγελιών, καθώς και ο χρόνος εκτέλεσης, ο αριθμός των πολυγώνων που ικανοποιήθηκαν, καθώς και ο min/max/mean αριθμός επαναλήψεων όπως προαναφέρθηκε.

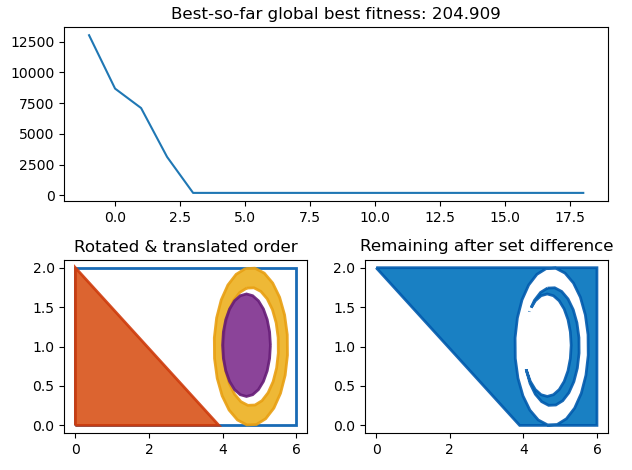
*4.2. Αποτελέσματα*

*4.2.1. DynPso*

Στο παρόν υποκεφάλαιο θα παρουσιαστεί συνοπτικά η ικανότητα τοποθέτησης των παραγγελιών στο απόθεμα για τον αλγόριθμο DynPSO. Συγκεκριμένα στα Σχήματα 1 και 2 παρουσιάζεται η τοποθέτηση της πρώτης και τρίτης παραγγελίας αντίστοιχα, τα οποία οπικά αποτελούν μια ικανοποιητική λύση του προβλήματος, καθώς περισσεύει ελάχιστο τμήμα του αρχικού αποθέματος ενώ στην δεύτερη περίπτωση βλέπουμε την τοποθέτηση του ελλειπτικού κύκλου εντός του αποθέματος που δημιουργείται εξαιτίας της κοίλης έλλειψης.



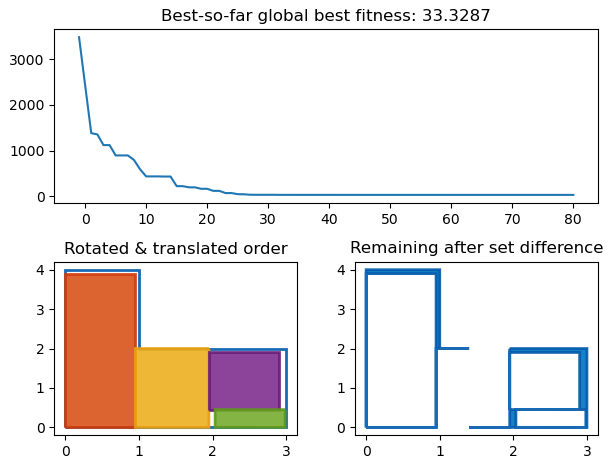
Σχήμα 1. Τοποθέτηση της πρώτης παραγγελίας.



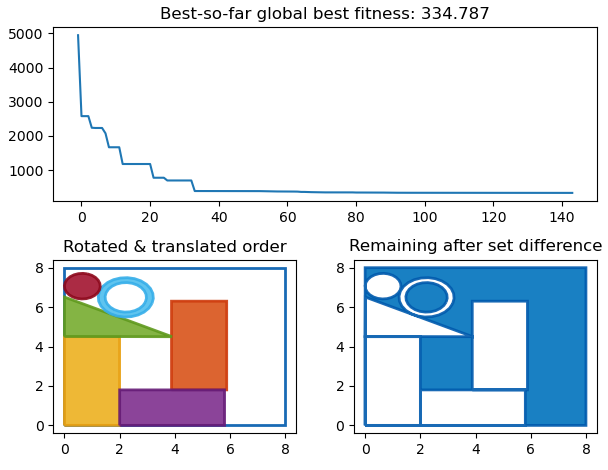
Σχήμα 2. Τοποθέτηση της τρίτης παραγγελίας.

*4.2.2. DEGL*

Αντίστοιχα για τον DEGL θα παρουσιαστούν μερικά παραδείγματα σχετικά με την ικανότητα τοποθέτησης των παραγγελιών στο απόθεμα. Στο Σχήμα 3 παρουσιάζεται η ίδια παραγγελία με το Σχήμα 1 για λόγους ποιοτικής σύγκρισης με τον DynPSO. Παρατηρείται πως ο αλγόριθμος DEGL τοποθετεί κατάλληλα στην παραγγελία στο απόθεμα και ενδεχομένως αγγίζει μια καλύτερη λύση βάση του υπολειπόμενου εμβαδού, όμως χρειάζεται τον διπλάσιο αριθμό εποχών προκειμένου να φτάσει σε αυτή τη λύση. Παρόμοια συμπεριφορά εμφανίζεται και στο Σχήμα 4 όπου ο αλγόριθμος τοποθετεί τις παραγγελίες όμως ο αριθμός των εποχών προκειμένου να φτάσει σε αυτή τη λύση ξεπερνά τις 140 παρόλα αυτά οι παραγγελίες δεν τοποθετούνται βέλτιστα.



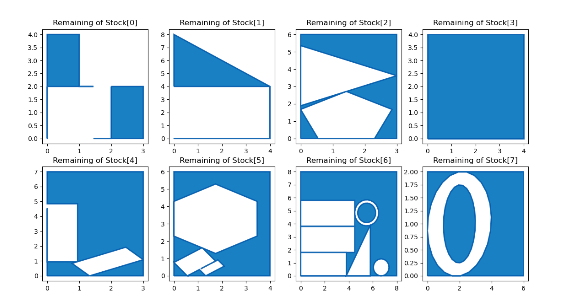
Σχήμα 3. Τοποθέτηση της πρώτης παραγγελίας.



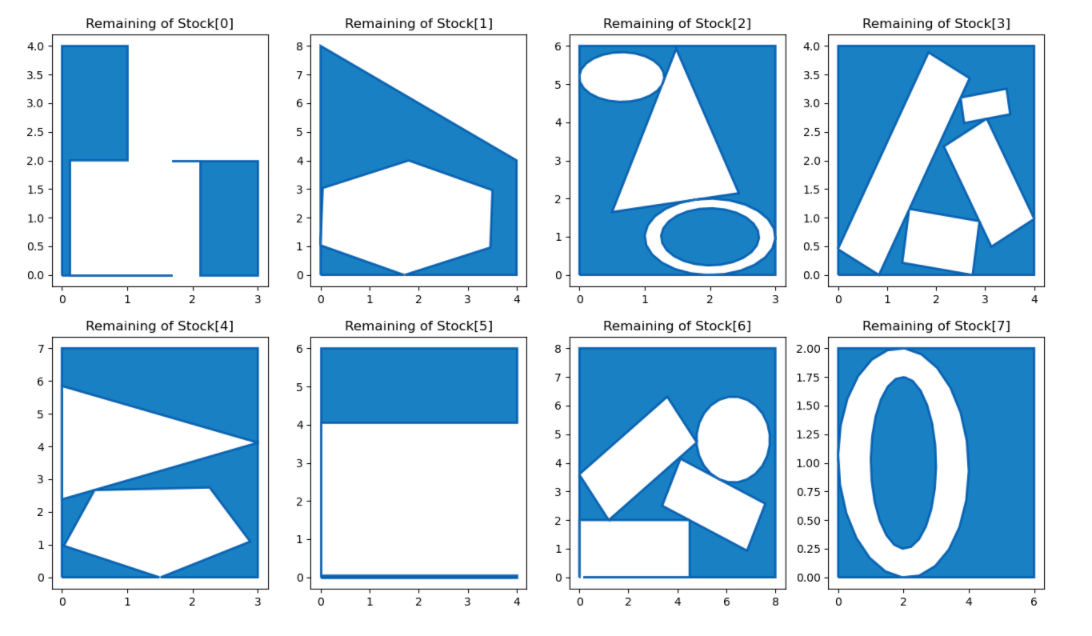
Σχήμα 4. Τοποθέτηση της δεύτερης παραγγελίας.

*4.2.3. Υπόλοιποι αλγόριθμοι (Nelder-Mead, L-BFGS-B και SLSQP)*

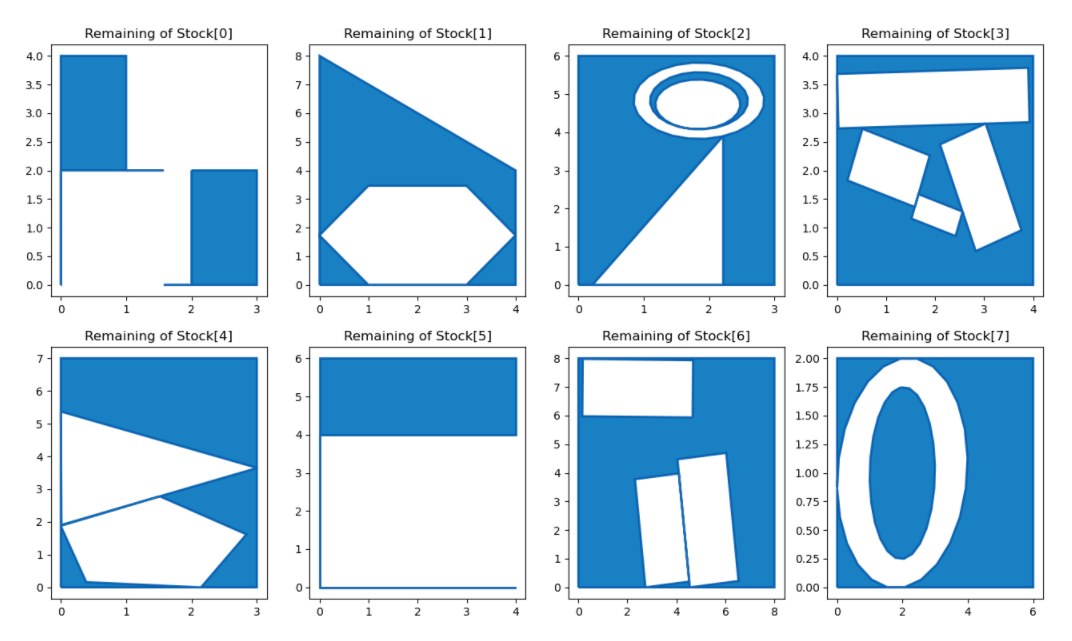
Σχετικά με την ικανότητα τοποθέτησης των υπόλοιπων αλγορίθμων, τα σχήματα 5, 6 και 7 παρουσιάζουν τις τελικές λύσεις για το πρόβλημα μας για τους αλγορίθμους SLSQP, L-BFGS-B και Nelder-Mead αντίστοιχα. Ο πρώτος κατάφερε να χωρέσει στο απόθεμα 16 από το σύνολο των 17 πολυγώνων, ενώ η απόδοση του μπορεί να χαρακτηριστεί μέτρια έως φτωχή για την επίλυση του προβλήματος. Έπειτα ο L-BFGS-B φαίνεται να αποδίδει σαφώς καλύτερα καθώς καταφέρνει να τοποθετήσει εντός αποθέματος 16 στα 17 πολύγωνα τα οποία είναι εμφανώς καλύτερα τοποθετημένα από την πρώτη περίπτωση του SLSQP. Τέλος ακολουθεί ο Nelder-Mead του οποίου τα αποτελέσματα είναι εμφανή στο Σχήμα 7 και ο αλγόριθμος καταφέρνει να τοποθετήσει όπως και στην περίπτωση του L-BFGS-B 16 στα 17 πολύγωνα. Το κοινό χαρακτηριστικό μεταξύ των διάφορων τοποθετήσεων είναι πως στις περισσότερες από αυτές περισσεύουν μεγάλα εμβαδά, γεγονός που θα διερευνηθεί περισσότερο στο τελευταίο κεφάλαιο αυτής της εργασίας προκειμένου να εξαχθούν βάσιμα συμπεράσματα της συμπεριφοράς αυτών.



Σχήμα 5. Ικανότητα τοποθέτησης παραγγελιών βάση του SLSQP .



Σχήμα 6. Ικανότητα τοποθέτησης παραγγελιών βάση του L-BFGS-B.



Σχήμα 7. Ικανότητα τοποθέτησης παραγγελιών βάση του Nelder-Mead.

1. *Συγκρίσεις και συμπεράσματα*

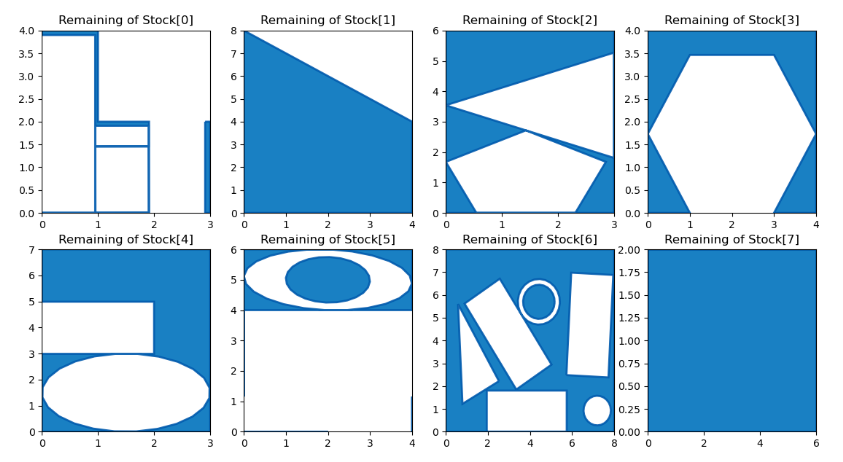
Στα προηγούμενα υποκεφάλαια 4.2.1, 4.2.2 και 4.2.3 παρουσιάστηκαν οι τοποθετήσεις των παραγγελιών στο σύνολο του αποθέματος προκειμένου να γίνει ένας αρχικός οπτικός ή ποιοτικός έλεγχος της ικανότητας τοποθέτησης του κάθε αλγορίθμου. Επίσης έχει ήδη αναφερθεί στο υποκεφάλαιο 4.1 πως η σύγκριση θα γίνει σύγκριση των αλγορίθμων ως προς τον χρόνος εκτέλεσης, τον αριθμό των πολυγώνων που ικανοποιήθηκαν, καθώς και το min/max/mean αριθμός επαναλήψεων. Τα αποτελέσματα που αναφέρονται στον Πίνακα 1, αφορούν τους μέσους όρους των αναφερόμενων μετρικών ή μεθόδων αξιολόγησης για 5 εκτελέσεις του εκάστοτε αλγορίθμου. Παρόλα αυτά για λόγους συντομίας δεν θα παρουσιαστούν όλες οι οπτικοποιήσεις των τοποθετήσεων, καθώς δεν θα μας βοηθήσουν στο παρόν στάδιο συγκρίσεων των αλγορίθμων.

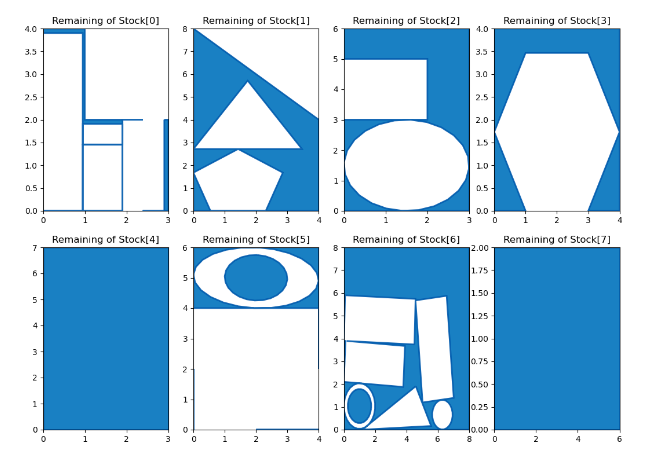
Από τον Πίνακα 1, η πρώτη παρατήρηση που μπορεί να εξαχθεί είναι πως οι αλγόριθμοι PSO και DEGL τοποθετούν επιτυχώς το σύνολο των παραγγελιών στο απόθεμα σε κάθε εκτέλεσή τους, σε αντίθεση με τους υπόλοιπους οι οποίοι με τη σειρά παρουσίασης στον Πίνακα 1, παρατηρούμε πως πέφτουν σε τελική απόδοση. Βέβαια αξιομνημόνευτος είναι ο Nelder - Mead ο οποίος έχασε μόλις 1 πολύγωνο σε 5 εκτελέσεις, εξού και ο μέσος όρος των 16.8 τοποθετημένων πολυγώνων. Επιπλέον όσων αφορά τον αριθμό των απαιτούμενων επαναλήψεων, ο PSO χρειάζεται μικρότερο αριθμό αυτών για τον τερματισμό της εκτέλεσής του, παρόλα αυτά και οι δύο αλγόριθμοι απαιτούν το λιγότερο 16 επαναλήψεις. Εκεί που διαφοροποιούνται στη σχετική μετρική είναι ο μέγιστος αριθμός επαναλήψεων για το συγκεκριμένο πρόβλημα, καθώς ο PSO χρειάζεται στην χειρότερη περίπτωση 113 επαναλήψεις για τον τερματισμό του ενώ ο DEGL 167, γεγονός που επηρεάζει και τον χρόνο εκτέλεσής του. Σχετικά με τους Nelder - Mead, L-BFGS-B και SLSQP ως προς τον χρόνο εκτέλεσης παρατηρούμε πως απαιτείται σημαντικά μικρότερο διάστημα, παρόλα αυτά σε αυτόν τον χρόνο δεν περιλαμβάνονται οπτικοποιήσεις των figures, οπότε θεωρουνται μη συγκρίσιμοι. Παρόλα αυτά ακόμη και μεταξύ τους παρατηρείται πως ο Nelder - Mead απαιτεί χονδρικά τριπλάσιο χρονικό διάστημα προκειμένου να ολοκληρωθεί η εκτέλεσή του.

|  | Τοποθετημένα Πολύγωνα | Επαναλήψεις (Mean,Min,Max) | Χρόνος εκτέλεσης (Mean,Min,Max sec) |
| --- | --- | --- | --- |
| PSO | 17 | 36.92, 16, 113 | 170.9, 134.46, 193.26 |
| DEGL | 17 | 58,26, 16, 167 | 321,78, 264,9, 372,06 |
| Nelder - Mead | 16,8 | N/A | 135,83 |
| L-BFGS-B | 15,6 | N/A | 47,108 |
| SLSQP | 14 | N/A | 42,66 |

Πίνακας 1. Σύγκριση αποτελεσμάτων για 5 εκτελέσεις των αλγορίθμων βάση των μετρικών που αναφέρθηκαν στο κεφάλαιο 4.1

Όσων αφορά τους αλγορίθμους DynPSO και DEGL, μπορούμε να εξάγουμε το συμπέρασμα πως σε γενικές γραμμές ο DEGL παράγει πιο συμπαγείς τοποθετήσεις πάνω στο απόθεμα(θεωρητικά απορρέει από τις αναζητήσεις βέλτιστων λύσεων κοντά σε υπάρχουσες καλές λύσεις), παρόλα αυτά αυτό δεν σημαίνει πως ο DynPSO υπολείπεται σημαντικά ως προς την ικανότητα τοποθέτησης των παραγγελιών. Συνεπώς η διαφορά μεταξύ των δύο ως προς την ικανότητα τοποθέτησης μπορεί να παραληφθεί ή να θεωρηθεί ακόμη και αμελητέα. Συνεχίζοντας την μεταξύ τουw σύγκριση, παρατηρείται πως ο DEGL χρειάζεται πολύ περισσότερο χρόνο εκτέλεσης, σχεδόν διπλάσιο σε σχέση με τον DynPSO. Αυτό δεν φαίνεται μόνο στην στήλη που αφορά τον χρόνο εκτέλεσης, αλλά εξαρτάται άμεσα από τον αριθμό των επαναλήψεων που χρειάζεται ο αλγόριθμος προκειμένου να συγκλίνει σε λύση. Κάτι επιπλέον που παρατηρείται στην συμπεριφορά του εκάστοτε αλγορίθμου, είναι πως για συγκεκριμένους συνδυασμούς αποθεμάτων και παραγγελιών, εμφανίζεται πολύ παρόμοια συμπεριφορά μεταξύ των διαδοχικών εκτελέσεων. Αυτό δείχνει πως έχουν πεπερασμένη ικανότητα τοποθέτησης στην εξεύρεση διαφορετικών λύσεων ενώ δεν είναι ιδιαίτερα δημιουργικοί ως προς αυτές τις τοποθετήσεις που πιθανόν θα αποτελούν και καλύτερες λύσεις για το πρόβλημα. Τέλος παρουσιάζονται στα Σχήματα 8 και 9 τα αποτελέσματα των αλγορίθμων DynPSO και DEGL για 1 εκτέλεση του αλγορίθμου. Ένα ακόμη συμπέρασμα που εξάγεται στη σύγκριση μεταξύ DynPSO και DEGL και δεν φαίνεται άμεσα από τα αποτελέσματα που παρουσιάζονται, είναι πως ο DEGL στο σύνολο των εκτελέσεων τείνει να αφήνει περισσότερα τμήματα του αποθέματος ακαίραια, δηλαδή χωρίς να χρειαστεί να τοποθετήσει καμία παραγγελία πάνω σε αυτά, γεγονός που ενισχύει την ικανότητα τοποθέτησης του δεύτερου.

Σχήμα 8. Αποτελέσματα τοποθέτησης παραγγελιών βάση του DynPSO.



Σχήμα 9. Αποτελέσματα τοποθέτησης παραγγελιών βάση του DEGL.