

Εξελικτική Υπολογιστική – Αλγόριθμοι εμπνευσμένοι από την βιολογία

Μεταπτυχιακό Πρόγραμμα «Προηγμένα Συστήματα Υπολογιστών και Επικοινωνιών»
Τμήμα Ηλεκτρολόγων Μηχανικών και Ηλεκτρονικών Υπολογιστών ΑΠΘ
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Ελλάδα

Θέμα Εργασίας

Χρήση εξελικτικών αλγορίθμων για βελτιστοποίηση διαχείρισης
αποθεμάτων ξύλου

Ονοματεπώνυμο: Παναγιώτα Καγιόγλου
pankagkon@ecc.auth.gr

Φεβρουάριος 2022

Πίνακας περιεχομένων

Μεθοδολογία επίλυσης του προβλήματος.	3
Διαμόρφωση Συνάρτησης Καταλληλότητας.	4
Επίλυση του προβλήματος με τον Particle Swarm Optimization Algorithm	6
Αποτελέσματα εκτέλεσης του PSO	7
Έλεγχος παραμέτρων βελτιστοποίησης του αλγορίθμου PSO.....	8
Αποτελέσματα διαδοχικών επαναλήψεων διαφοροποιώντας τις τιμές των παραμέτρων βελτιστοποίησης του PSO.	9
Επίλυση του προβλήματος με τον Differential Evolution Algorithm	14
Αποτελέσματα εκτέλεσης του DE για rand1bin	15
Αποτελέσματα εκτέλεσης του DE για την στρατηγική best1bin	16
Αποτελέσματα εκτέλεσης του DE για την στρατηγική best2bin	17
Αποτελέσματα εκτέλεσης του DE για την στρατηγική best1exp.....	18
Αποτελέσματα εκτέλεσης του DE για την στρατηγική currenttobest1exp	19
Αποτελέσματα διαδοχικών επαναλήψεων του αλγορίθμου DE για τις διάφορες στρατηγικές, διαφοροποιώντας τις τιμές των παραμέτρων βελτιστοποίησης του.....	20
Επίλυση του προβλήματος με τον Differential Evolutionary with Global Local Optimization Algorithm.....	32
Επίλυση του αλγορίθμου DEGL διαφοροποιώντας κάποιες από τις παραμέτρους του.....	33
Χρήση εναλλακτικών, μη εξελικτικών αλγορίθμων για την επίλυση του προβλήματος	35
Επίλυση του προβλήματος με τον SLSQP algorithm	35
Επίλυση του προβλήματος με τον Nelder-Mead algorithm.....	37
Επίλυση του προβλήματος με τον L-BFGS-B algorithm.....	38
Επίλυση του προβλήματος με τον Pattern search algorithm.....	39
Αύξηση του αποθέματος.....	41
Διπλασιασμός του αποθέματος.....	41
Τετραπλασιασμός του αποθέματος.....	43
REFERENCES.....	46

Μεθοδολογία επίλυσης του προβλήματος.

Αρχικά ο αλγόριθμος διαβάζει όλα τα αποθέματα (stocks) και τις παραγγελίες. Για κάθε μια παραγγελία διαδοχικά εκτελούνται τα παρακάτω βήματα:

- Υπολογίζεται το άθροισμα των εμβαδών όλων των πολυγώνων της παραγγελίας i
- Κατατάσσονται τα stocks με αύξουσα σειρά ανάλογα με το εμβαδόν τους σε 'έναν πίνακα (αργότερα ανανεώνεται ανάλογα με τις κοπές που γίνονται στα αποθέματα).
- Από τον παραπάνω πίνακα κρατιόνται μόνο τα stocks στα οποία χωράει ολόκληρη η παραγγελία.
- Για κάθε stock διαδοχικά εκτελείται ένας εξελικτικός ή μη αλγόριθμος ανάλογα με το ζητούμενο του προβλήματος και γίνεται προσπάθεια να χωρέσει ολόκληρη η παραγγελία μέσα στο απόθεμα.
- Έναν η παραγγελία χωρέσει ολόκληρη, το stock κόβεται και ανανεώνεται η λίστα των remaining stocks. Στην περίπτωση που η παραγγελία δεν χωρέσει, χωρίζεται σε δύο μικρότερες παραγγελίες. Ο διαχωρισμός των παραγγελιών δεν γίνεται τυχαία, αλλά κατάσσονται σε αύξουσα σειρά τα κομμάτια ανάλογα με το εμβαδόν τους. Με αυτόν τον τρόπο εξασφαλίζεται πως δεν θα αχρηστεύει ένα σχετικά μεγάλο stock για να κοπεί μια μικρή παραγγελία που ενδεχομένως να χωρούσε και σε άλλα stocks. Εάν αποτύχει και αυτό, η παραγγελία συνεχίζει να διαμελίζεται. Εάν η παραγγελία καταλήξει να αποτελείται από ένα μόνο πολύγωνο και ο αλγόριθμος δεν καταφέρει να τοποθετήσει αυτό το πολύγωνο εντός του αποθέματος, τότε αυτό το πολύγωνο μένει εκτός και κατατάσσεται σε μια λίστα που περιέχει τα σχήματα που δεν μπόρεσαν να «κοπούν».

Διαμόρφωση Συνάρτησης Καταλληλότητας

Η συνάρτηση καταλληλότητας θα διαμορφωθεί αθροίζονταν τα παρακάτω κριτήρια, αφού πρώτα πολλαπλασιαστούν με τα αντίστοιχα βάρη ανάλογα με την σπουδαιότητα τους. Βασική παράμετρος για την διαμόρφωσή της συνάρτησης καταλληλότητας είναι να εξασφαλίζει πως ο αλγόριθμος θα αποδώσει έγκυρες λύσεις ακόμα και αν αυτές δεν είναι οι βέλτιστες. Επίσης είναι απαραίτητο να μπορεί να γενικεύει και να μην επιλύει απλά το πρόβλημα για τα συγκεκριμένα αποθέματα και τις παραγγελίες. Έτσι βασικοί παράμετροι που υπολογίστηκαν είναι οι παρακάτω.

- Οι παραγγελίες να μην βγαίνουν εκτός αποθέματος

Για την ικανοποίηση αυτού του κριτηρίου υπολογίστηκε αρχικά το πολύγωνο (set) που σχηματίζουν όλες οι παραγγελίες αφού έχουν τοποθετηθεί από τον αλγόριθμο «μέσα» στο stock. Στην συνέχεια υπολογίστηκε η διαφορά αυτού του πολυγώνου με το stock που μελετάται εκείνη την στιγμή. Το εμβαδόν του set της διαφοράς των δύο σχημάτων, καθορίζει εάν υπάρχουν σχήματα που τοποθετήθηκαν εκτός του αποθέματος. Ζητούμενο αυτού του κριτηρίου είναι το εμβαδόν του set της διαφοράς να είναι μηδενικό, ώστε να μην έχει τοποθετηθεί καμία παραγγελία εκτός ορίων. Το εμβαδόν αυτό πολλαπλασιάζεται με ένα αρκετά μεγάλο βάρος και αθροίζεται στην συνάρτηση καταλληλότητας.

- Να μην υπάρχει επικάλυψη των κομματιών της παραγγελίας

Για την εξασφάλιση αυτού του κριτηρίου υπολογίζεται αρχικά το εμβαδόν όλων το σχημάτων καθώς και το εμβαδόν της ένωσης των παραγγελιών αφού έχουν τοποθετηθεί από τον αλγόριθμο. Στην συνέχεια γίνεται αφαίρεση των δύο αυτών τιμών. Όταν η διαφορά τους είναι μηδενική προφανώς σημαίνει ότι δεν υπάρχει επικάλυψη. Ο συγκεκριμένος έλεγχος έχει αρκετά μεγάλη τιμή στο βάρος του, καθώς αποτελεί βασικό κριτήριο για αν κριθεί μια παραγγελία έγκυρη.

- Τα σχήματα της παραγγελίας πρέπει να τοποθετούνται κοντά το ένα στο άλλο.

Αρχικά κατάσσονται όλα τα κομμάτια της παραγγελίας σε μια λίστα ανάλογα με το εμβαδόν τους. Στην συνέχεια ο αλγόριθμός υπολογίζει την έκκεντρη απόσταση του stock(i) με το stock(i+1). Όσο πιο μικρό είναι το άθροισμα όλων αυτών των τιμών τόσο πιο κοντά βρίσκονται τα σχήματα μεταξύ τους. Το βάρος που δίνεται για αυτό τον έλεγχο είναι κατά πολύ μικρότερο από τα δύο παραπάνω καθώς αποτελεί κριτήριο βελτιστοποίησης και όχι εγκυρότητας.

- Τα σχήματα να συγκεντρώνονται στην άκρη του αποθέματος

Για κάθε πολύγωνο της παραγγελίας ξεχωριστά υπολογίζεται το εμβαδόν του. Το εμβαδόν αυτό πολλαπλασιάζεται με τις αποστάσεις x και y διαδοχικά οι οποίες αντιστοιχούν στην οριζόντια και κατακόρυφη απόσταση του κέντρου βάρους του σχήματος από το (0,0). Η μείωση των αποστάσεων x και y των τμημάτων της παραγγελίας ως προς το (0,0) του αποθέματος, εξασφαλίζει ότι οι παραγγελίες συγκεντρώνονται σε

μια από τις άκρες του αποθέματος. Στο συγκεκριμένο πρόβλημα ορίστηκε ως (0,0) η κάτω αριστερή γωνία του κάθε stock. Το βάρος που δίνεται σε αυτό τον έλεγχο είναι σχετικά χαμηλό.

- Τα stock μετά την κοπή πρέπει να έχουν όσο πιο κανονικό σχήμα γίνεται.

Η ποσοτικοποίηση αυτού του κριτηρίου γίνεται λαμβάνοντας υπόψη το εμβαδόν του stock που απομένει μετά την κοπή και το εμβαδόν του εγγεγραμμένου πολύγωνου. Πιο συγκεκριμένα, υπολογίζεται ο λόγος του $\text{Area}_{\text{hull}}$ / $\text{Area}_{\text{remaining}}$ -1 . Ο λόγος αυτός κυμαίνεται μεταξύ 0-1. Η τιμή αυτού του λόγου (L) εισέρχεται στην εξίσωση $f(x) = 1 / (1 + 1.1 * L)$ και πολλαπλασιάζεται με ένα σχετικά μεγάλο βάρος πριν μπει στην συνάρτηση καταλληλότητας.

Για όλες τις εκτελέσεις που ακολουθούν η συνάρτηση καταλληλότητας παρέμεινε σταθερ. Ωστόσο τα βάρη τροποποιήθηκαν ελαφρός από αλγόριθμο σε αλγόριθμο για λόγους σύγκλισης και εγκυρότητας.

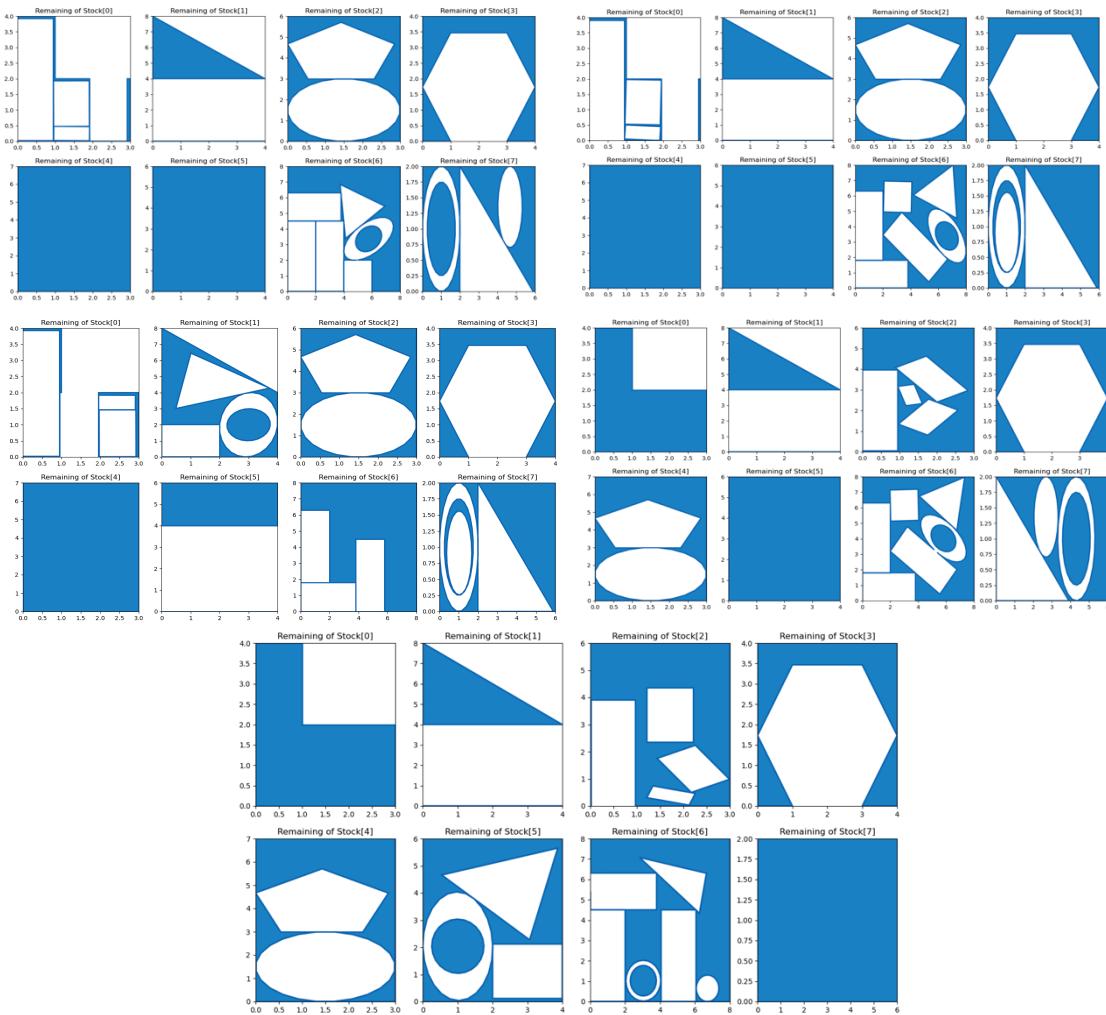
Επίλυση του προβλήματος με τον Particle Swarm Optimization Algorithm

Ο αλγόριθμος PSO είναι ένας αλγόριθμος που βασίζεται στην νοημοσύνη σμήνους. Η νοημοσύνη σμήνους βασίζεται σε τρεις βασικές αρχές συλλογικής συμπεριφοράς:

- ❖ **Αδράνεια (Inertia):** Τάση διατήρηση παλιών συμπεριφορών που αποδείχθηκαν σωστές
- ❖ **Κοινωνική επιρροή:** Επιρροή από την κυρίαρχη τάση της κοινωνίας
- ❖ **Επιρροή από γείτονες:** Η αλληλεπίδραση με γειτονικά πρόσωπα έχει μεγαλύτερη ισχύ από την επιρροή ολόκληρης της κοινωνίας

Γενικά ο αλγόριθμος αυτός επιλύει ένα πρόβλημα έχοντας έναν πληθυσμό υποψήφιων λύσεων (σωματίδια) και μετακινώντας αυτά τα σωματίδια στον χώρο αναζήτησης σύμφωνα με έναν απλό μαθηματικό τύπο που εξαρτάται από τη θέση και την ταχύτητα του σωματιδίου. Η κίνηση κάθε σωματιδίου επηρεάζεται από την τοπική του καλύτερη θέση, αλλά και από τις καλύτερες θέσεις των άλλων σωματιδίων. Αυτό αναμένεται να οδηγήσει το σμήνος προς τις καλύτερες λύσεις. [3][4]

Αποτελέσματα εκτέλεσης του PSO



Εικόνα 1 Αποτελέσματα κοπής των stock για τις πέντε διαδοχικές εκτελέσεις του PSO

Αρχικά πραγματοποιήθηκαν πέντε διαδοχικές εκτελέσεις του αλγορίθμου PSO, διατηρώντας όλες τις default παραμέτρους του αλγορίθμου. Όπως φαίνεται από τον πίνακα 1 ο αλγόριθμος κατάφερε να χωρέσει όλες τις παραγγελίες μέσα στο stock και ο μέσος χρόνος εκτέλεσης του είναι γύρω στα 213 second. Λαμβάνοντας υπόψη τα αποτελέσματα που της εικόνας 1, φαίνεται πως ο PSO καταφέρει να τοποθετήσει σχετικά καλά τις παραγγελίες μέσα στο απόθεμα. Ενδεικτικό παράδειγμα για αυτή την θεωρεία αποτελεί τη κοπή του πρώτου απόθεματος. Το πρώτο απόθεμα έχει περίπου ίσο εμβαδόν με αυτό της πρώτης παραγγελίας. Το γεγονός ότι ο αλγόριθμος καταφέρνει και τοποθετεί σχεδόν πάντα όλα τα κομμάτια της πρώτης παραγγελίας μέσα στο πρώτο stock αποδεικνύει πως ο κώδικας τείνει να βρίσκει τις βέλτιστες λύσεις και πως τα βάρη στην συνάρτηση καταλληλότητας έχουν επιλεγεί σχετικά καλά. Επιπλέον, ενθαρρυντικό είναι το γεγονός πως στα σχήματα με οπές καταφέρει να τοποθετήσει μέσα τους άλλα πολύγωνα που ενδεχομένως χωράνε. Ωστόσο, ενδέχεται ο αλγόριθμος να μην καταφέρει αν περιστρέψει σωστά τα κομμάτια της παραγγελίας, όπως γίνεται στις δύο τελευταίες περιπτώσεις του stock[2] και τα κομμάτια να μην τοποθετηθούν με τον βέλτιστο τρόπο.

Weights of fitness function criteria:			
w_{f_OUT} =5000.00, $w_{f_OVERLAP}$ =5000.00, w_{f_ATTR} =3.00, w_{f_SMO} =1000.00, w_{f_DIST} =5.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	221.78	59.53	17/17
Rep2	179.62	49.06	17/17
Rep3	224.89	60	17/17
Rep4	256.84	51.73	17/17
Rep5	186.62	43.76	17/17

Πίνακας 1 Αποτελέσματα 5 διαδοχικών εκτελέσεων του PSO με default παραμέτρους βελτιστοποίησης

Έλεγχος παραμέτρων βελτιστοποίησης του αλγορίθμου PSO

Για των PSO ζητείται να τροποποιηθούν οι τιμές των ακόλουθων δεικτών:

- P_s -> Social Adjustment Weight (y_2)
- P_c -> Self-Adjustment Weight (y_1)
- N_f -> Neighborhood size

Δείκτες P_s και P_c

Ο PSO είναι ένας αλγόριθμος που το διάνυσμα ταχύτητας του κάθε particle εκφράζεται από την παρακάτω σχέση.

$$v = W * v + y_1 * u_1 * (p - x) + y_2 * u_2 * (g - x).$$

όπου

- V είναι η προηγούμενη ταχύτητα του particle
- $p - x$, είναι η διαφορά της τωρινής θέσης και της καλύτερης θέσης του particle.
- $g - x$, είναι η διαφορά της τωρινής θέσης και της καλύτερης θέσης των γειτόνων του particle.
- W, y_1, y_2 είναι κάποια βάρη με τα οποία πολλαπλασιάζονται οι τιμές της συνάρτησης.

Οι δείκτες y_1 και y_2 αναφέρονται στην βιβλιογραφία και ως παράμετροι εμπιστοσύνης, όπου ο y_1 εκφράζει πόση εμπιστοσύνη έχει ένα σωματίδιο στον εαυτό του, ενώ το y_2 εκφράζει πόση εμπιστοσύνη έχει ένα σωματίδιο στους γείτονές του. Αν $y_1 > 0$ και $y_2 = 0$, όλα τα σωματίδια είναι ανεξάρτητα. Κάθε σωματίδιο βρίσκει την καλύτερη θέση στη γειτονιά του και αντικαθιστά την τρέχουσα καλύτερη θέση εάν η νέα θέση που βρήκε είναι καλύτερη. Τα σωματίδια εκτελούν τοπική αναζήτηση. Από την άλλη πλευρά, εάν $y_2 > 0$ και $y_1 = 0$, ολόκληρο το σμήνος έλκεται από ένα μόνο σημείο.

Οι χαμηλές τιμές για τα y_1 και y_2 έχουν ως αποτέλεσμα ομαλές τροχιές σωματιδίων, επιτρέποντας στα σωματίδια να περιφέρονται/μεταφέρονται μακριά από καλές περιοχές για εξερεύνηση πριν τραβηγτούν πίσω προς καλές περιοχές. Οι υψηλές τιμές προκαλούν μεγαλύτερη επιτάχυνση, με απότομη κίνηση προς ή μετά από καλές περιοχές.[1]

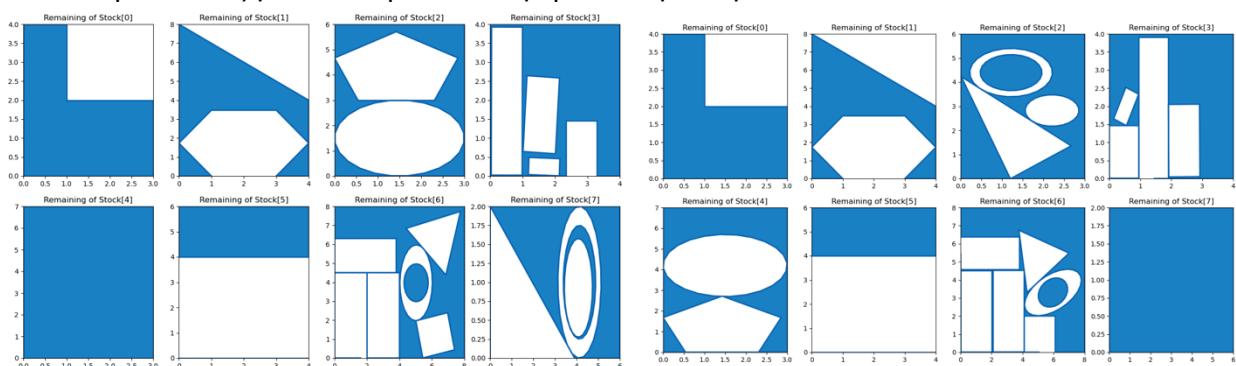
Συντελεστής N_f (Neighborhood size- Μέγεθος γειτονιάς)

Ο συντελεστής N_f καθορίζει την ένταση της κοινωνικής αλληλεπίδρασης μέσα στο σμήνος (swarm). Ουσιαστικά, ο N_f συντελεστής καθορίζει το μέγεθος της γειτονιάς. Όσο μικρότερες είναι οι γειτονιές, τόσο μικρότερη είναι η αλληλεπίδραση. Παρότι, οι μικρότερες γειτονιές οδηγούν σε πιο αργή σύγκλιση αλγορίθμου, οδηγούν και σε πιο αξιόπιστες λύσεις. Πιο συγκεκριμένα, τα μικρότερα μεγέθη γειτονιών είναι λιγότερο επιφρεπή στα τοπικά ελάχιστα. Από την βιβλιογραφία προτείνετε να ξεκινήσει η αναζήτηση με μικρές γειτονιές και σταδιακά να αυξάνεται το μέγεθος της γειτονιάς αναλογικά με την αύξηση του αριθμού των επαναλήψεων. Αυτή η προσέγγιση εξασφαλίζει μια αρχική υψηλή ποικιλομορφία με ταχύτερη σύγκλιση καθώς τα σωματίδια κινούνται προς μια πολλά υποσχόμενη περιοχή αναζήτησης.

Αποτελέσματα διαδοχικών επαναλήψεων διαφοροποιώντας τις τιμές των παραμέτρων βελτιστοποίησης του PSO.

Self-Adjustment Weight=0.5 & Social Adjustment Weight=1.49 & Neighborhood size=0.25

Μειώνοντας την παράμετρο του self-adjustment weight από 1.49, που είναι η default τιμή, σε 0.5, φαίνεται πως ο αλγόριθμος εξακολουθεί να χωράει όλες τις παραγγελίες μέσα στο διαθέσιμο stock. Επιπλέον, παρατηρείται πως ο χρόνος εκτέλεσης του αλγορίθμου δεν έχει αυξηθεί αισθητά σε σχέση με τα αποτελέσματα του πίνακα1. Ωστόσο, ο αλγόριθμος δεν καταφέρνει να χωρέσει ολόκληρη την πρώτη παραγγελία στο πρώτο stock, γεγονός που αποτελεί μια ένδειξη ότι δεν προτείνει την βέλτιστη λύση.

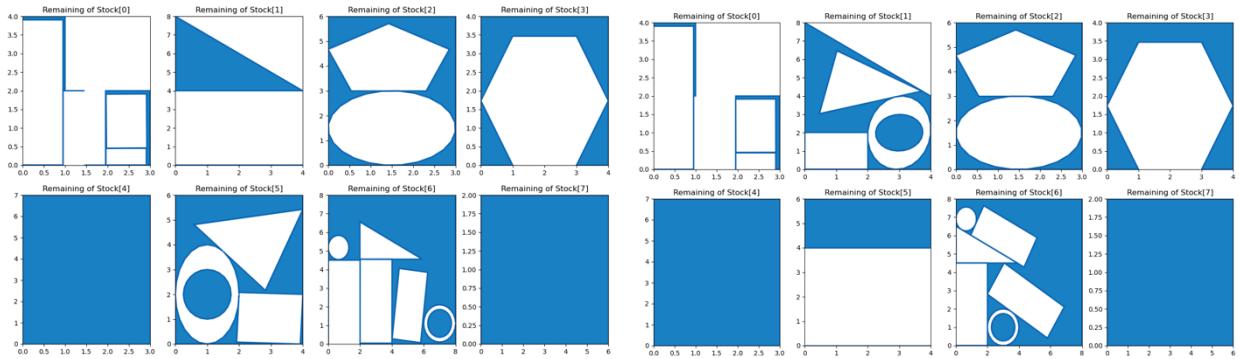


Εικόνα 2 Αποτελέσματα κοπής PSO για Self-Adjustment Weight=0.5 & Social Adjustment Weight=1.49 & Neighborhood size=0.25

Weights of fitness function criteria:			
$w_{f_OUT}=5000.00, w_{f_OVERLAP}=5000.00, w_{f_ATTR}=3.00, w_{f_SMO}=1000.00, w_{f_DIST}=5.00$			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	239.55	52.05	17/17
Rep2	214.48	43.50	17/17

Πίνακας 2 Πίνακας αποτελεσμάτων για Self-Adjustment Weight=0.5 & Social Adjustment Weight=1.49 & Neighborhood size=0.25

Self-Adjustment Weight=2.5 & Social Adjustment Weight=1.49 & Neighborhood size=0.25



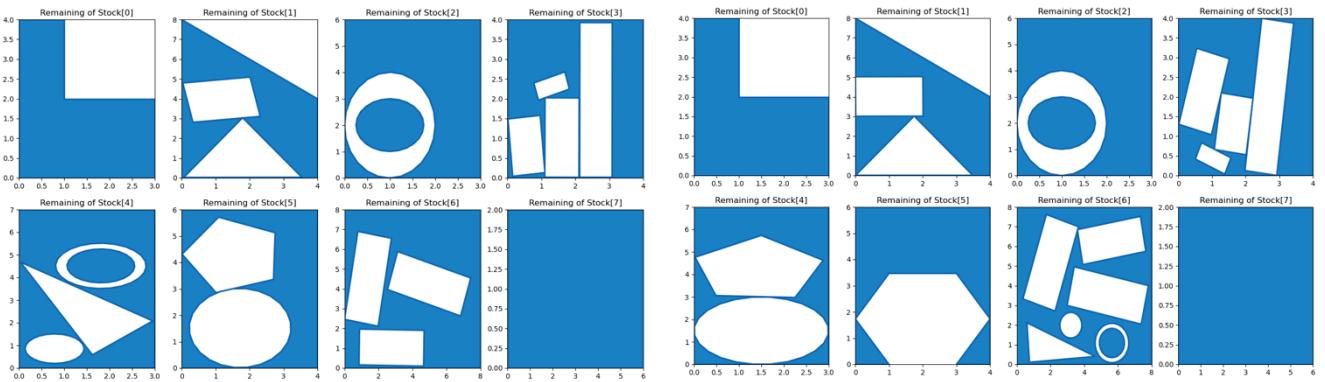
Εικόνα 3 Τοποθέτηση των παραγγελιών μέσα στα αποθέματα για SelfAdjustmentWeight=2.5 & SocialAdjustmentWeight=1.49 & Neighborhood size=0.25

Αυξάνοντας την παράμετρο του self-adjustment weight από 1.49 σε 2.5, υπάρχει μια μικρή αύξηση του χρόνου εκτέλεσης του αλγορίθμου συγκρίνοντας τα αποτελέσματα του πίνακα 1 και του πίνακα 3. Ωστόσο, ο αλγόριθμος καταφέρνει να τοποθετεί όλες τις παραγγελίες εντός του αποθέματος. Παράλληλα χωράει την πρώτη παραγγελία εντός του πρώτου αποθέματος που είναι αρκετά κοντά σε συνολικό εμβαδόν, όπως φαίνεται από την εικόνα 3. Ενθαρρυντικό, είναι το γεγονός ότι αφήνει δύο μεγάλα κομμάτια του stock αναλλοίωτα και στις δύο εκτελέσεις του.

Weights of fitness function criteria: w_f_OUT=5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=1000.00, w_f_DIST=5.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	308.09	67.46	17/17
Rep2	311.18	80.85	17/17

Πίνακας 3 Πίνακας αποτελεσμάτων για SelfAdjustmentWeight=2.5 & SocialAdjustmentWeight=1.49 & Neighborhood size=0.25

Self-Adjustment Weight=1.49 & Social Adjustment Weight=0.5 & Neighborhood size=0.25



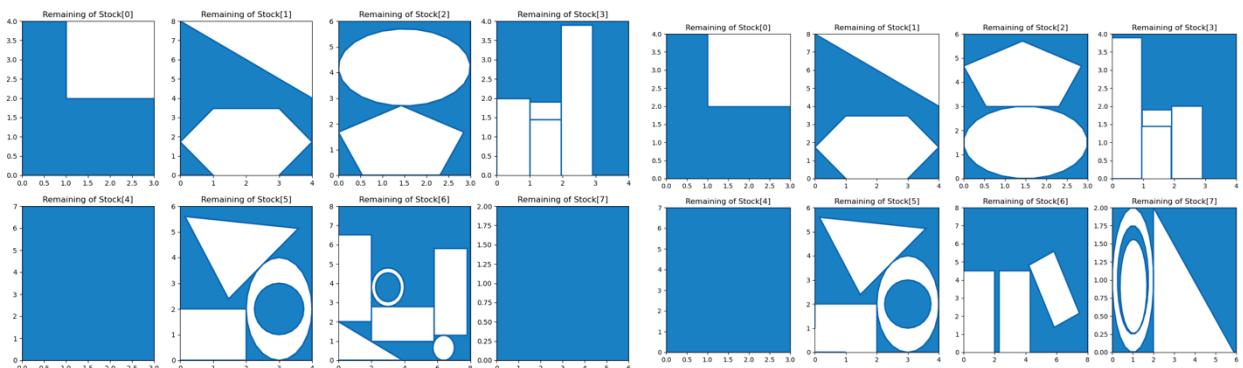
Εικόνα 4 Κοπή των αποθεμάτων μετά την τοποθέτηση των παραγγελιών για Self-Adjustment Weight=1.49 & Social Adjustment Weight=0.5 & Neighborhood size=0.25

Στην εικόνα 4 φαίνεται η συμπεριφορά του αλγορίθμου κατόπιν μείωσης της παραμέτρου social-adjustment weight. Τα particles σε αυτή την εκτέλεση δεν επηρεάζονται αρκετά από την συμπεριφορά των γειτόνων τους, με αποτέλεσμα ο αλγόριθμος να μην καταφέρνει να χωρέσει τις παραγγελίες εντός του αποθέματος, όπως φαίνεται στην πινακα 4. Παράλληλα, παρατηρείται τουλάχιστον διπλασιασμός στον χρόνο εκτέλεσης, συγκρίνοντας τα αποτελέσματα του πίνακα 1 και 4. Γενικά, φαίνεται πως αυτές οι υπερπαράμετροι δεν οδηγούν στις βέλτιστες λύσεις του αλγορίθμου.

Weights of fitness function criteria:			
w_{f_OUT} :5000.00, $w_{f_OVERLAP}$:5000.00, w_{f_ATTR} :3.00, w_{f_SMO} :1000.00, w_{f_DIST} :5.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	739.44	87.0	15/17
Rep2	599.83	93.26	16/17

Πίνακας 4 Πίνακας αποτελεσμάτων εκτέλεσης για SelfAdjustmentWeight=1.49 & SocialAdjustmentWeight=0.5 & Neighborhood size=0.25

SelfAdjustment Weight=1.49 & SocialAdjustment Weight=2.5 & Neighborhood size=0.25



Εικόνα 5 Τομή αποθέματος με βάση τις εκάστοτε παραγγελίες για Self-Adjustment Weight=1.49 & Social Adjustment Weight=2.5 & Neighborhood size=0.25

Στην εικόνα 5 φαίνονται η τοποθέτηση των παραγγελιών για social-adjustment weight 2.5. Τα σχήματα φαίνεται να τοποθετούνται αρκετά κοντά μεταξύ τους αλλά ο αλγόριθμος δεν

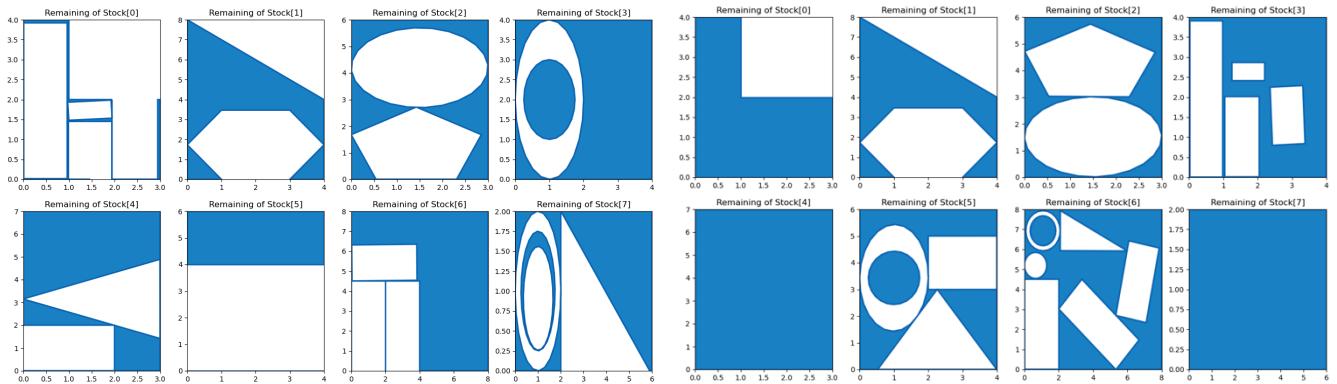
κατάφερε να χωρέσει όλες τις παραγγελίες. Επιπλέον υπάρχει αύξηση στον χρόνο εκτέλεσης συγκρίνοντας του πίνακες 1 και 5.

Weights of fitness function criteria:			
w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=1000.00, w_f_DIST=5.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	423.52	97.87	16/17
Rep2	532.53	104.9	16/17

Πίνακας 5 Πίνακας αποτελέσματων για Self-Adjustment Weight=1.49 & Social Adjustment Weight=2.5 & Neighborhood size=0.25

Γενικά, οι παράμετροι SelfAdjustment Weight& Social Adjustment Weight πρέπει να έχουν παρεμφερείς τιμές και να κυμαίνονται γύρω από τις default τιμές για αυτό το πρόβλημα.

Self-Adjustment Weight=1.49 & Social Adjustment Weight=1.49 & Neighborhood size=0.5



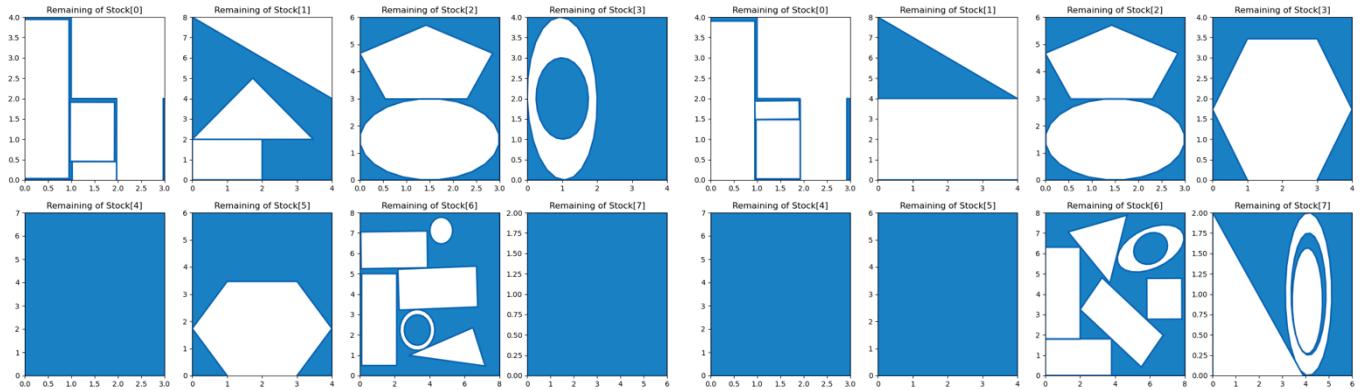
Εικόνα 6 Αποτελέσματα κοπής των αποθέματων με βάση τις παραγγελίες για SelfAdjustmentWeight=1.49 & SocialAdjustmentWeight=1.49 & Neighborhood size=0.5

Στην συνέχει μεταβάλλεται η παράμετρος N_f που διαμορφώνει το μέγεθος της γειτονίας. Αρχικά η παράμετρος N_f αυξάνεται από 0,25 σε 0,5. Στην πρώτη επίλυση, όπως φαίνεται στο πρώτο group σχημάτων της εικόνας 6. Ο αλγόριθμος τοποθετεί σωστά όλες τις παραγγελίες, και καταφέρει να εκμεταλλευτεί και τις τρύπες που ενδεχομένως έχουν κάποια σχήματα και να χωρέσει μέσα σε αυτές μικρότερα κομμάτια. Παράλληλα, ο χρόνος εκτέλεσης του είναι κοντά στους χρόνους εκτέλεσης του πίνακα 1. Ωστόσο, στην δεύτερη επανάληψη φαίνεται ένα εντελώς διαφορετικό, αποτέλεσμα και ο αλγόριθμος τοποθετεί εντός του stock μόνο τις 16 από τις 17 παραγγελίες. Γενικά, η τιμή του $N_f = 0,5$ μπορεί να οδηγήσει σε καλά αποτελέσματα έναν ο αλγόριθμος επιλέξει τα σωστά particles.

Weights of fitness function criteria:			
w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=1000.00, w_f_DIST=5.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	222.32	42.31	17/17
Rep2	151.16	37.31	16/17

Πίνακας 6 Αποτελέσματα εκτέλεσης αλγορίθμου για Self-Adjustment Weight=1.49 & Social Adjustment Weight=1.49 & Neighborhood size=0.5

Self-Adjustment Weight=1.49 & Social Adjustment Weight=1.49 & Neighborhood size=0.9



Εικόνα 7 Αποτελέσματα κοπής των αποθεμάτων για Self-Adjustment Weight=1.49 & Social Adjustment Weight=1.49 & Neighborhood size=0.9

Η παράμετρος N_f αυξάνεται από 0,25 σε 0.9. Σε συνέχεια των παρατηρήσεων που έγιναν για την εικόνα 6, φαίνεται πως και τα σχήματα της εικόνα 7 ακολουθούν την ίδια συμπεριφορά. Ο αλγόριθμος δεν καταφέρει πάντα να χωρέσει όλα τις παραγγελίες μέσα στο απόθεμα και ο χρόνος εκτέλεσης δεν διαφοροποιείται αισθητά από τους χρόνους εκτέλεσης των επαναλήψεων που χρησιμοποιήθηκαν οι default παράμετροι (πίνακας 1).

Weights of fitness function criteria:			
w_{f_OUT} =5000.00, $w_{f_OVERLAP}$ =5000.00, w_{f_ATTR} =3.00, w_{f_SMO} =1000.00, w_{f_DIST} =5.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	227.91	41.88	16/17
Rep2	171.22	47.13	17/17

Πίνακας 7 Πίνακας αποτελεσμάτων για Self-Adjustment Weight=1.49 & Social Adjustment Weight=1.49 & Neighborhood size=0.9

Γενικά, παρότι, οι μικρότερες γειτονιές ($N_f = 0.25$) οδηγούν σε πιο αργή σύγκλιση αλγορίθμου, οδηγούν σε πιο αξιόπιστες λύσεις. Ο αλγόριθμος τοποθετεί επιτυχώς όλες τις παραγγελίες και δεν υπάρχουν έντονες ανομοιομορφίες μεταξύ της τοποθέτησης των σχημάτων μεταξύ των επαναλήψεων.

Επίλυση του προβλήματος με τον Differential Evolution Algorithm

Η διαφορική εξέλιξη είναι στοχαστική μέθοδος (δεν χρησιμοποιεί gradient descent) για να βρει το ελάχιστο και μπορεί να πραγματοποιήσει αναζητήσεις σε μεγάλες περιοχές του υποψηφίου χώρου. Σε κάθε επανάληψη μέσα από τον πληθυσμό, ο αλγόριθμος μεταλλάσσει κάθε υποψήφια λύση αναμιγνύοντας με άλλες υποψήφιες λύσεις για να δημιουργήσει μια δοκιμαστική υποψήφια λύση. Υπάρχουν διάφορες στρατηγικές για τη δημιουργία υποψηφίων δοκιμαστικών λύσεων, οι οποίες ταιριάζουν σε ορισμένα προβλήματα περισσότερο από άλλα.

Ο αλγόριθμος DE θα επιλυθεί για τις ακόλουθες στρατηγικές. Οι διαφορετικές στρατηγικές μετάλλαξης που θα χρησιμοποιηθούν είναι οι εξής:

- rand/1:
 $v_i = x_{r0} + F_t \cdot (x_{r1} - x_{r2})$
- best/1:
 $v_i = x_{best} + F_t \cdot (x_{r1} - x_{r2})$
- ctb/1:
 $v_i = x_i + F_t \cdot (x_{best} - x_i) + F_t \cdot (x_{r1} - x_{r2})$
- best/2:
 $v_i = x_{best} + F_t \cdot (x_{r1} + x_{r2} - x_{r3} - x_{r4})$

όπου:

x_i είναι το αρχικό διάνυσμα που μεταλλάσσεται (donor)

v_i είναι το προκύπτουν μεταλλαγμένο διάνυσμα (mutant)

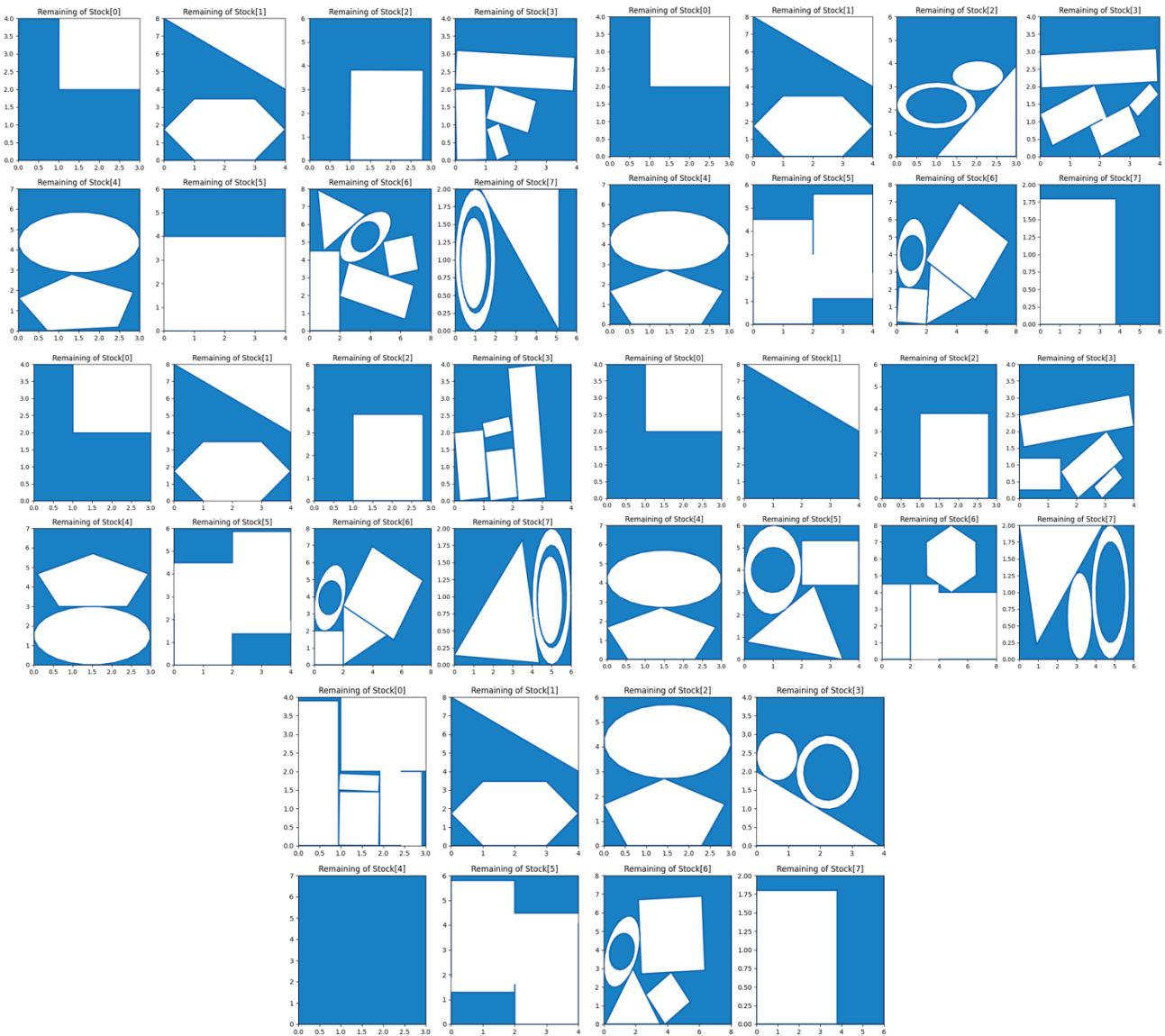
$r_0 = r_1 = r_2 = r_3 = r_4 = i$ είναι οι τυχαίοι δείκτες διανυσμάτων διαφορετικοί μεταξύ τους

x_{best} είναι το καλύτερο διάνυσμα που έχει βρεθεί ως τώρα

F_t είναι ο συντελεστής κλιμάκωσης της τρέχουσα επανάληψη $F_t \sim U(F_{min}, F_{max})$

Για την εκτέλεση των αλγορίθμου DE που φαίνονται παρακάτω χρησιμοποιήθηκαν οι default τιμές για κάθε περίπτωση που προτείνει η βιβλιοθήκη `scipy.optimize`.

Αποτελέσματα εκτέλεσης του DE για rand1bin



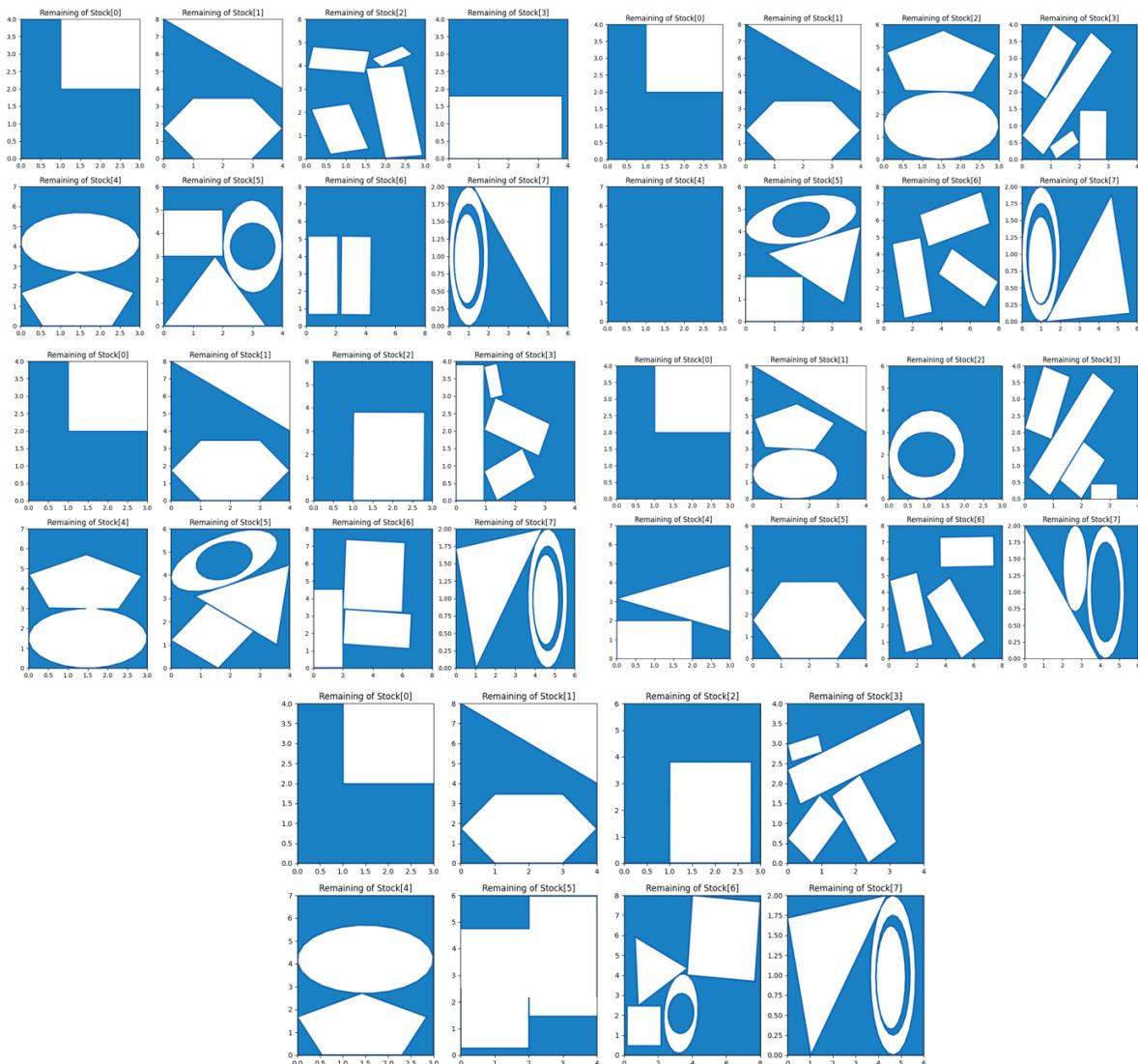
Εικόνα 8 Αποτελέσματα κοπής του αλγορίθμου DE με την στρατηγική rand1bin

Λαμβάνοντας υπόψη τα αποτελέσματα της εικόνας 8 φαίνεται πως ο αλγόριθμος DE με την στρατηγική rand1bin δεν προσεγγίζει επαρκώς την βέλτιστη λύση. Ενθαρρυντικό είναι το γεγονός πως καταφέρνει να τοποθετήσει όλες τις παραγγελίες, όμως στην πλειοψηφία των εκτελέσεών του αφήνει ανεκμετάλλευτο το 1^o stock στο οποίο χωράει η πρώτη παραγγελία οριακά εάν τοποθετηθούν σωστά όλα τα κομμάτια. Επιπλέον, παρατηρείται μια μικρή αύξηση στους χρόνους εκτέλεσης. Ο μέσος χρόνος εκτέλεσης του PSO για default τιμές ήταν 213 second και στον DE-rand1bin αυξήθηκε σε 336 second.

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	294.29	45.875	17/17
Rep2	402.13	49.23	17/17
Rep3	385.10	48.54	17/17
Rep4	293.80	49.80	17/17
Rep5	307.16	47.40	17/17

Πίνακας 8 Αποτελέσματα εκτέλεσης του αλγορίθμου DE με την στρατηγική rand1bin

Αποτελέσματα εκτέλεσης του DE για την στρατηγική best1bin



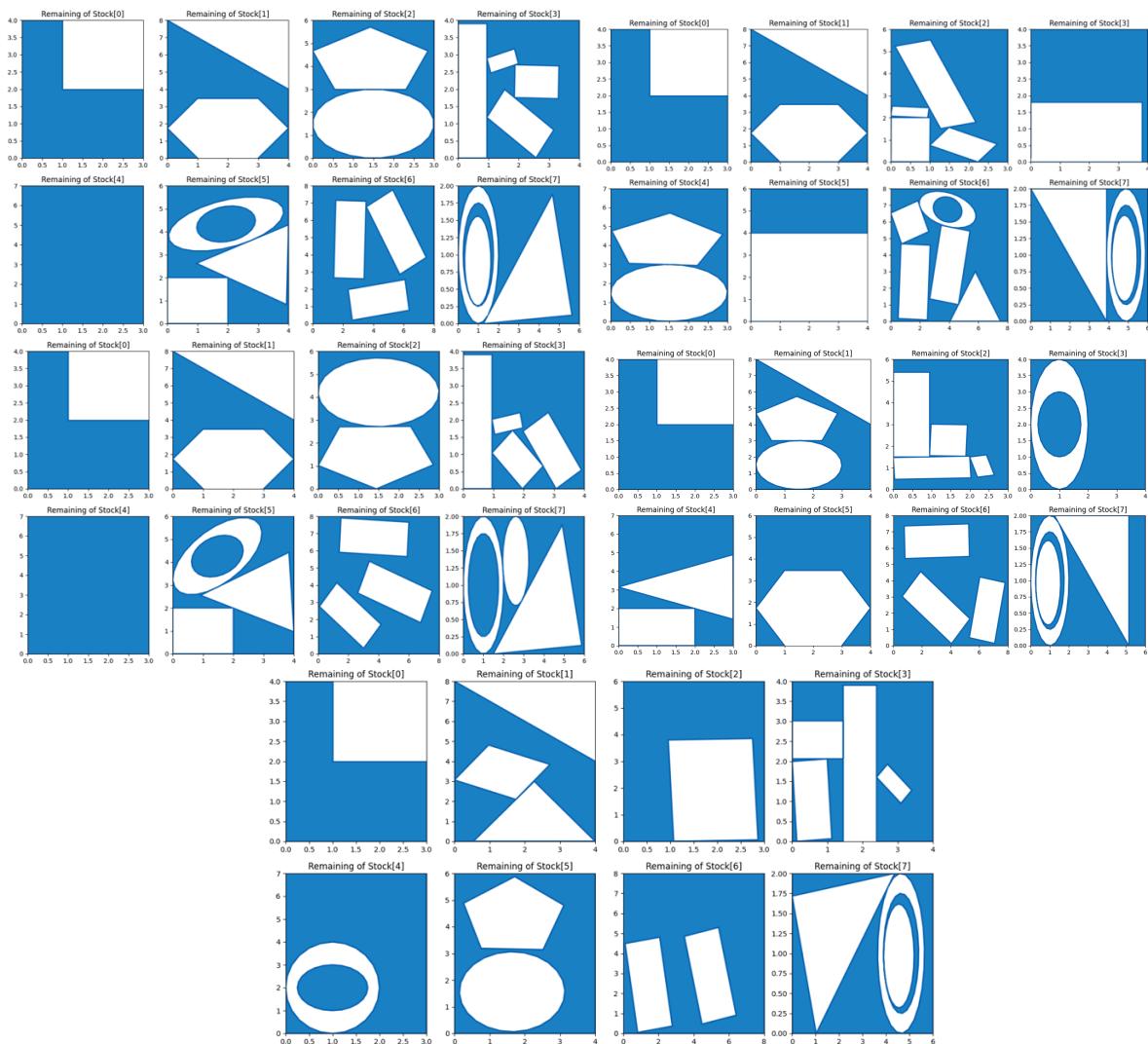
Εικόνα 9 Αποτελέσματα κοπής του αποθέματος με χρήση του αλγορίθμου DE με την στρατηγική best1bin

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	311.52	46.0	16/17
Rep2	270.76	44.45	16/17
Rep3	290.09	45.72	17/17
Rep4	335.82	44.21	16/17
Rep5	374.03	47.41	17/17

Πίνακας 9 Αποτελέσματα εκτέλεσης του αλγορίθμου DE με την στρατηγική best1bin

Από τα αποτελέσματα της εικόνας 9 και τα δεδομένα του πίνακα 9 φαίνεται πως η στρατηγική best1bin δεν ταιριάζει στην επίλυση αυτού του προβλήματος. Ο αλγόριθμος δεν τοποθετεί σωστά τις παραγγελίες καθώς τα σχήματα είναι σχετικά απομακρυσμένα μεταξύ τους και υπάρχουν και εκτελέσεις που αφήνουν μια παραγγελία εκτός του stock. Από άποψη ταχύτητας ο αλγόριθμος επιλύσει το πρόβλημα γύρω στα 316 second.

Αποτελέσματα εκτέλεσης του DE για την στρατηγική best2bin



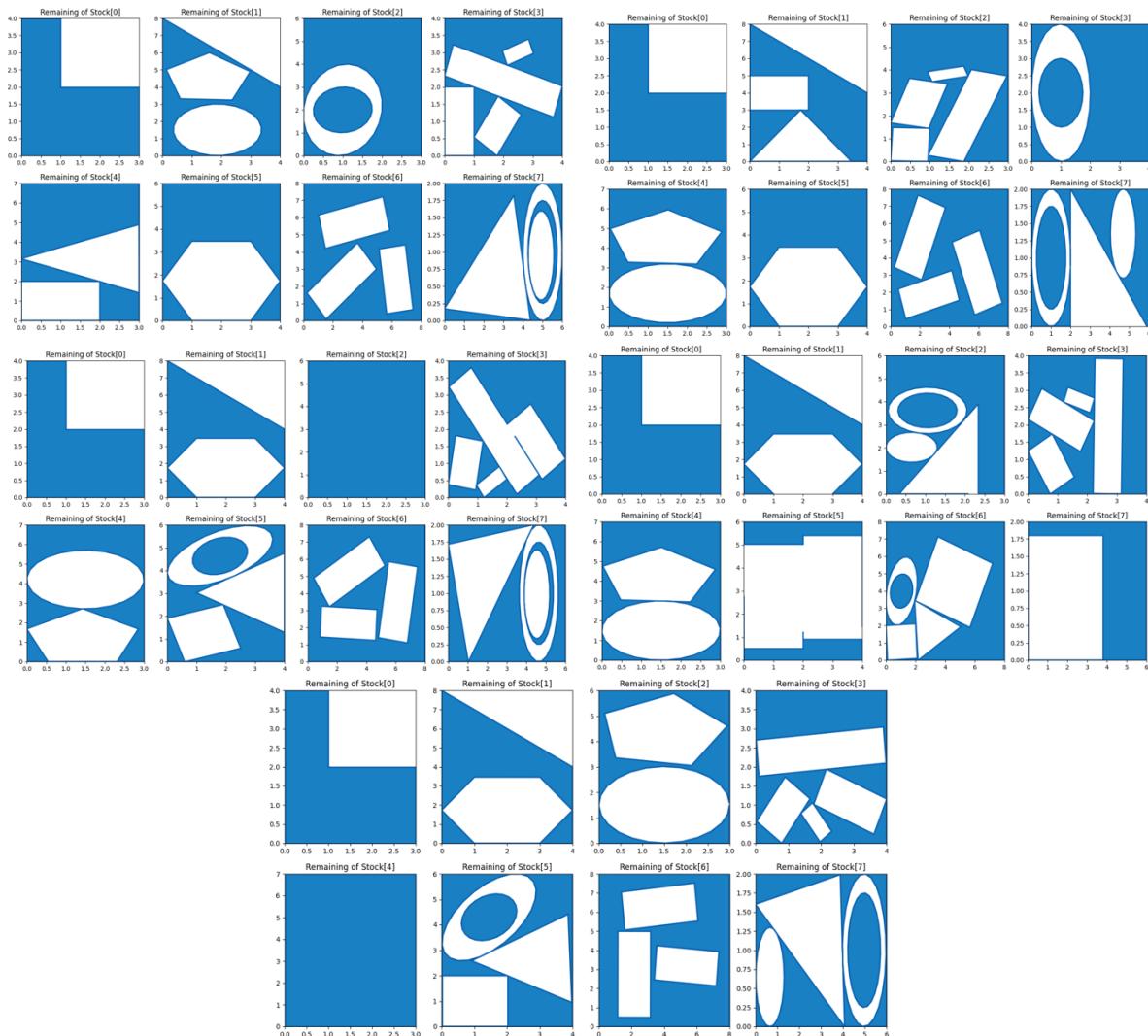
Εικόνα 10 Αποτελέσματα κοπής του αποθέματος με χρήση του αλγορίθμου DE με την στρατηγική best2bin

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	274.018	46.6	16/17
Rep2	302.63	49.4	17/17
Rep3	280.70	47.0	16/17
Rep4	352.82	46.9	16/17
Rep5	376.12	47.87	15/17

Πίνακας 10 Αποτελέσματα εκτέλεσης αλγορίθμου DE με την στρατηγική best1bin

Ο αλγόριθμος best2bin φαίνεται πως παρουσιάζει την χειρότερη απόδοση σε σύγκριση με όλους τους παραπάνω. Είναι ο πρώτος αλγόριθμος που αφήνει εκτός δύο κομμάτια του από τις παραγγελίες και τα σχήματα φαίνεται σαν να τοποθετούνται τυχαία μέσα στο απόθεμα. Παράλληλα, οι χρόνοι εκτέλεσης του είναι περίπου 316 second.

Αποτελέσματα εκτέλεσης του DE για την στρατηγική best1exp



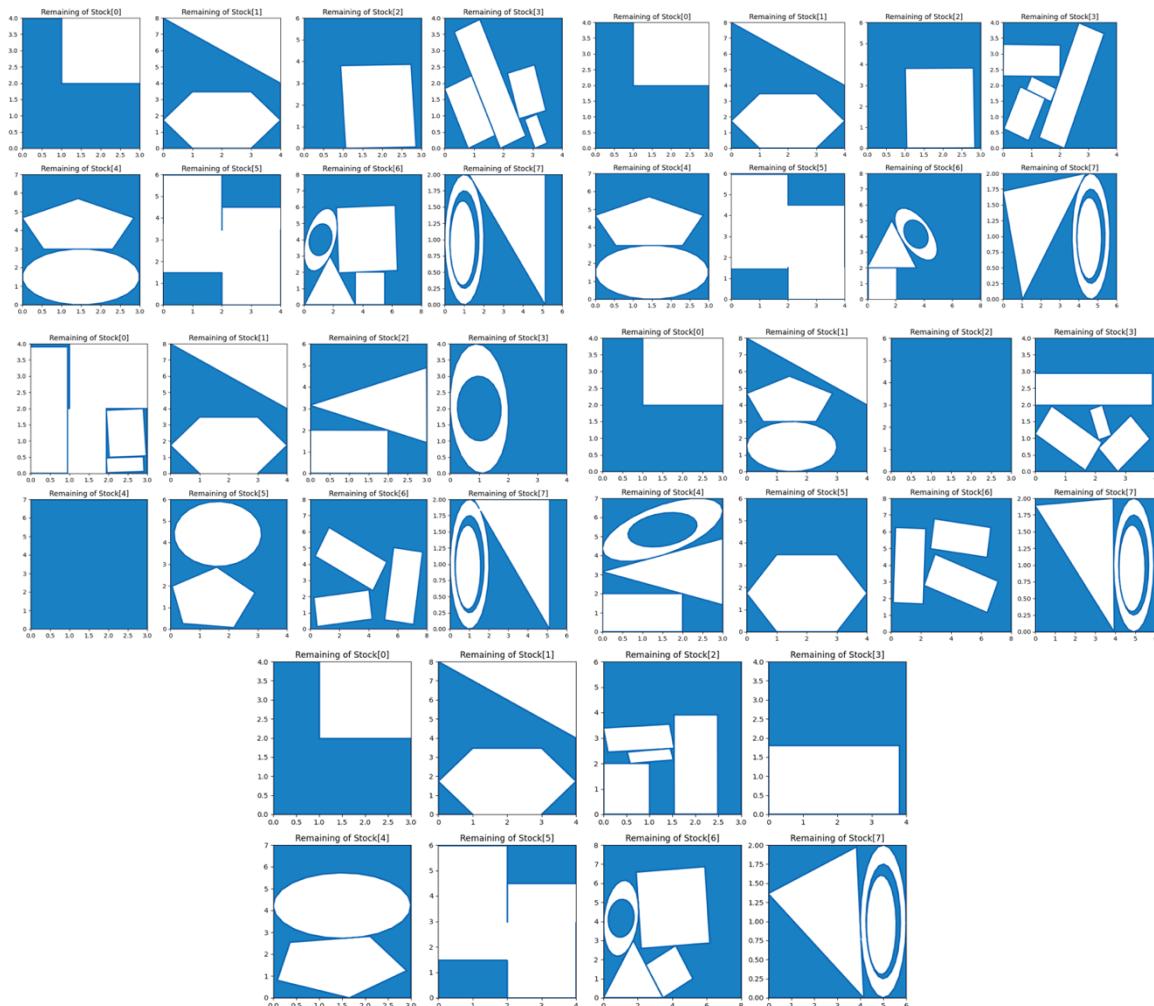
Εικόνα 11 Αποτελέσματα κοπής του αποθέματος με χρήση του αλγορίθμου DE με την στρατηγική best1exp

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
Rep1	363.52	47.5	16/17
Rep2	369.00	48.74	16/17
Rep3	308.36	47.09	16/17
Rep4	399.67	49.80	17/17
Rep5	279.97	47.7	16/17

Πίνακας 11 Αποτελέσματα εκτέλεσης αλγορίθμου με την χρήση του αλγορίθμου DE με την στρατηγική best1exp

Ο αλγόριθμος DE με την στρατηγική best1exp αποτυγχάνει να τοποθετήσει όλες τις παραγγελίες εντός του stock και ο χρόνος εκτέλεσης του είναι σχετικά αυξημένος σε σχέση με τους παραπάνω DE αλγορίθμους.

Αποτελέσματα εκτέλεσης του DE για την στρατηγική currenttobest1exp



Εικόνα 12 Αποτελέσματα κοπής του αποδέματος με χρήση του αλγορίθμου DE με την στρατηγική currenttobest1exp

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
Rep1	385.08	49.54	17/17
Rep2	362.37	49.375	16/17
Rep3	273.97	47.76	16/17
Rep4	242.42	47.94	16/17
Rep5	384.79	49.76	17/17

Πίνακας 12 Αποτελέσματα εκτέλεσης αλγορίθμου με την χρήση του αλγορίθμου DE με την στρατηγική currenttobest1exp

Τέλος ο αλγόριθμος currenttobest1exp έχει μια από τις καλύτερες υλοποιήσεις του DE σε αυτό το πρόβλημα, καθώς στην τρίτη επανάληψη κατάφερε να βρει μια αρκετά καλή λύση. Χώρεσε όλη την πρώτη παραγγελία μέσα στο πρώτο stock (μόνο ο rand1bin , από τις κατηγορίες του DE, κατάφερε να κάνει κάτι ανάλογο). Οι χρόνοι εκτέλεσής του και τα ποσοστά επιτυχίας είναι ανάλογα με όλους τους παραπάνω DE αλγορίθμους.

Γενικά, οι αλγόριθμοι DE φαίνεται πως δεν επιλύουν πάντα το πρόβλημα επιτυχώς. Στην πλειοψηφία των λύσεων τους αφήνουν κάποια παραγγελία εκτός του αποθέματος και οι χρόνοι εκτέλεσης τους είναι ελάχιστα αυξημένοι σε σχέση με αυτούς του PSO.

Αποτελέσματα διαδοχικών επαναλήψεων του αλγορίθμου DE για τις διάφορες στρατηγικές, διαφοροποιώντας τις τιμές των παραμέτρων βελτιστοποίησης του.

Οι παράμετροι που πρόκειται να διαφοροποιηθούν είναι οι ακόλουθοι.

Συντελεστή κλιμάκωσης F

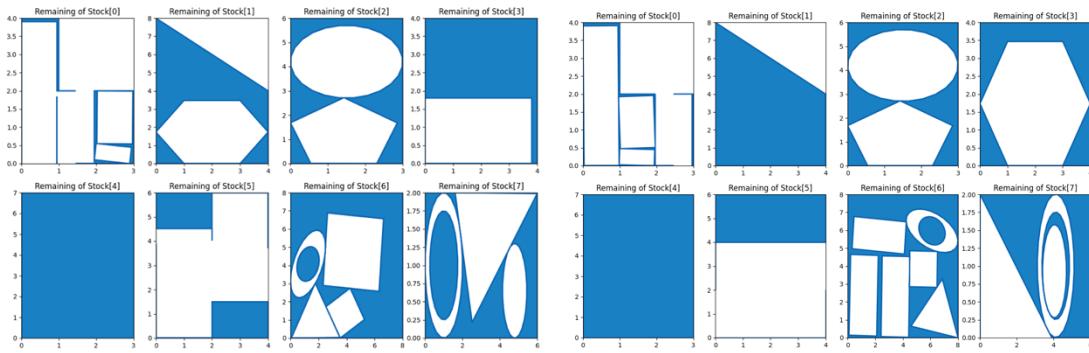
Στην βιβλιογραφία ο συντελεστής αυτός συναντάται και ως differential weight. Ο συντελεστής αυτός μπορεί να βοηθήσει σημαντικά στην επιτάχυνση της σύγκλισης. Γενικά, η αύξηση της σταθεράς μετάλλαξης μπορεί να αυξάνει την ακτίνα αναζήτησης, αλλά θα επιβραδύνει τη σύγκλιση του αλγορίθμου.[2]

Ο ρυθμός διασταύρωσης Cr (recombination constant)

Ο συντελεστής recombination constant πρέπει να είναι στην περιοχή [0, 1]. Στη βιβλιογραφία είναι επίσης γνωστός ως πιθανότητα διασταύρωσης (crossover probability), που συμβολίζεται με CR. Η αύξηση αυτής της τιμής επιτρέπει να προχωρήσουν στην επόμενη γενιά μεγαλύτερος αριθμός μεταλλαγμένων σωματιδίων, ώστόσο υπάρχει ο κίνδυνος σταθερότητας του πληθυσμού.

Αποτελέσματα διαδοχικών εκτελέσεων για:

Best1bin Cr=0.7 & F=[0.5-0.75]



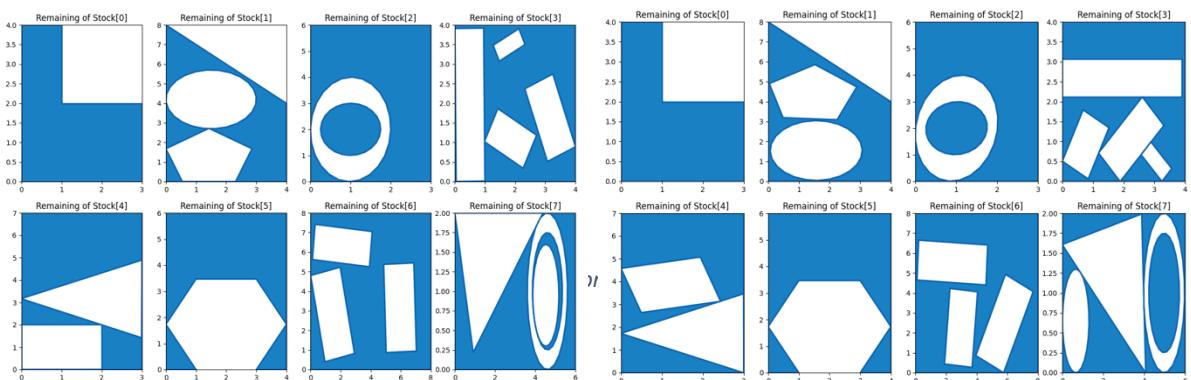
Εικόνα 13 Αποτελέσματα κοπής αποθεμάτων για τις διαδέσμευσης παραγγελίες με χρήση του αλγορίθμου DE για την στρατηγική best1bin, Cr=0.7 & F=[0.5-0.75]

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	301.75	42.75	17/17
Rep2	217.054	44.56	17/17

Πίνακας 13 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best1bin, Cr=0.7 & F=[0.5-0.75]

Η μείωση του συντελεστή κλιμάκωσης από [0.5-1,75] σε [0.5-0.75] έχει επηρεάσει θετικά την λειτουργία του αλγορίθμου. Στις δύο επαναλήψεις που φαίνονται στην πίνακα 13, ο αλγόριθμος κατάφερε να τοποθετήσει όλες τις παραγγελίες μέσα στο απόθεμα. Επιπλέον, ο χρόνος επίλυσης του είναι περίπου 259 second που θεωρείται καλή επίδοση για το συγκεκριμένο πρόβλημα, λαμβάνοντας υπόψη όλες τις παραπάνω εκτελέσεις. Επιπλέον, ο αλγόριθμος προτείνει αρκετά καλές λύσεις καθώς βλέπουμε ότι και στις δύο επαναλήψεις καταφέρνει να τοποθετήσει ολόκληρη την πρώτη παραγγελία στο πρώτο απόθεμα.

Best1bin Cr=0.7 & F=[0.5-1.75]

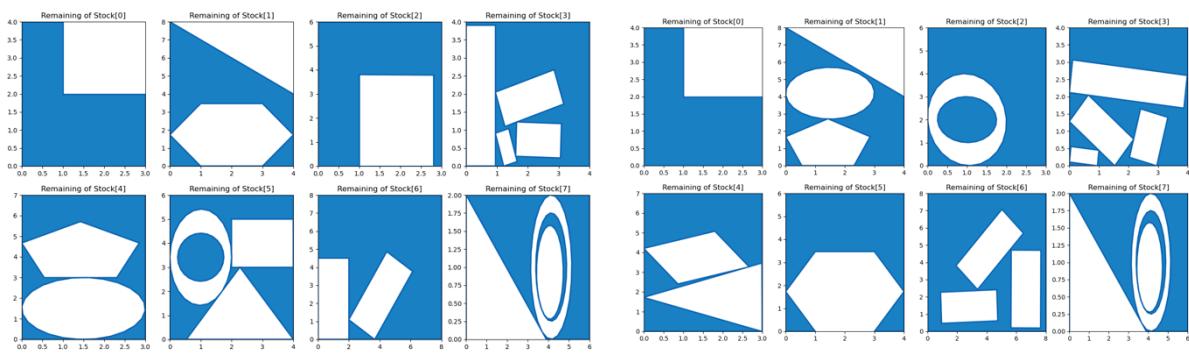


Εικόνα 14 Αποτελέσματα κοπής αποθεμάτων για τις διαδέσμευσης παραγγελίες με χρήση του αλγορίθμου DE για την στρατηγική best1bin, Cr=0.7 & F=[0.5-1.75]

Weights of fitness function criteria: w_f_OUT=5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	346.30	44.53	16/17
Rep2	349.66	44.64	16/17

Σε αντίθεση τα αποτελέσματα της παραπάνω εκτέλεσης η αύξηση του συντελεστή κλιμάκωση σε [0.5-1.75] δεν επιδρά ευεργετικά στον αλγόριθμο. Παρατηρείται πως υπάρχει αύξηση στους χρόνους εκτέλεσης που πλέον κυμαίνονται στα 350 second ενώ παράλληλα ο αλγόριθμος αφήνει ένα κομμάτι της παραγγελίας εκτός αποθέματος. Επιπλέον από την εικόνα 14 φαίνεται πως ο αλγόριθμος τοποθετεί σχεδόν τυχαία τις παραγγελίες μέσα στο stock.

Best1bin Cr=0.4 & F=[0.5-1.5]

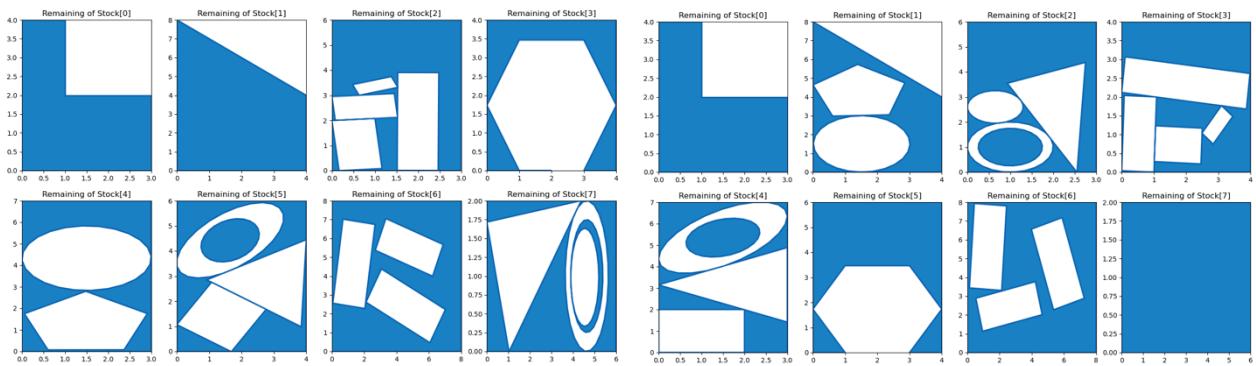


Εικόνα 15 Αποτελέσματα κοπής αποθεμάτων για τις διαθέσιμες παραγγελίες με χρήση του αλγορίθμου DE για την στρατηγική best1bin, Cr=0.4 & F=[0.5-1.5]

Weights of fitness function criteria: w_f_OUT=5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	280.22	47.56	16/17
Rep2	314.71	46.03	16/17

Πίνακας 15 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best1bin, Cr=0.4 & F=[0.5-1.5]

Best1bin Cr=0.9 & F=[0.5-1.5]



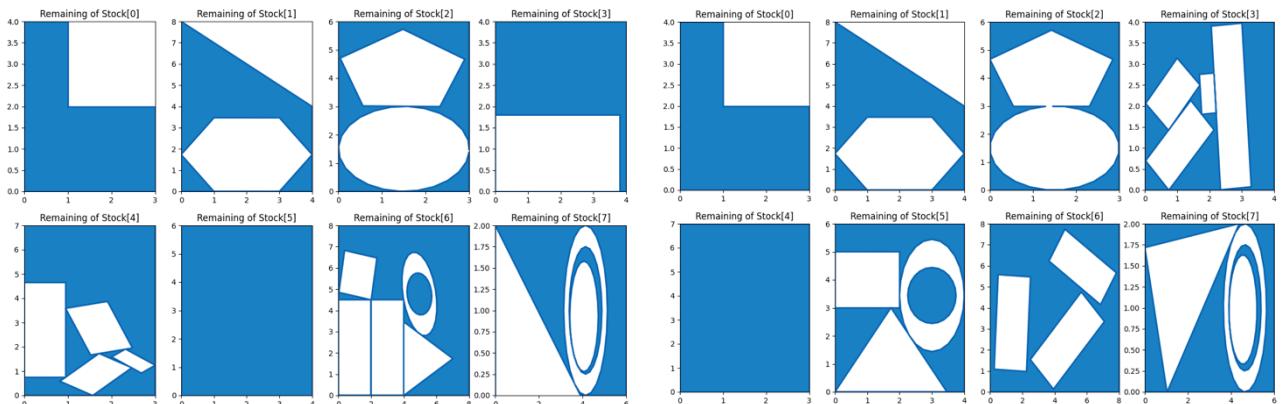
Εικόνα 16 Αποτελέσματα κοπής αποθεμάτων για τις διαθέσιμες παραγγελίες με χρήση του αλγορίθμου DE για την στρατηγική best1bin, Cr=0.9 & F=[0.5-1.5]

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
Rep1	299.45	45.72	16/17
Rep2	221.55	43.94	16/17

Πίνακας 16 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best1bin, Cr=0.9 & F=[0.5-1.5]

Στην συνέχεια πραγματοποιούνται δύο εκτελέσεις όπου στην πρώτη ο συντελεστής διασταύρωση στο μειώνεται 0.4, ενώ στην δεύτερη αυξάνεται στο 0.9. Ο αλγόριθμος δεν ανταποκρίνεται θετικά και στις δύο περιπτώσεις δεδομένου των αποτελεσμάτων της εικόνα 15 και 16. Τα σχήματα τοποθετούνται ακανόνιστα και ένα κομμάτι μιας παραγγελίας μένει εκτός αποθέματος.

Best1exp Cr=0.7 & F=[0.5-0.75]

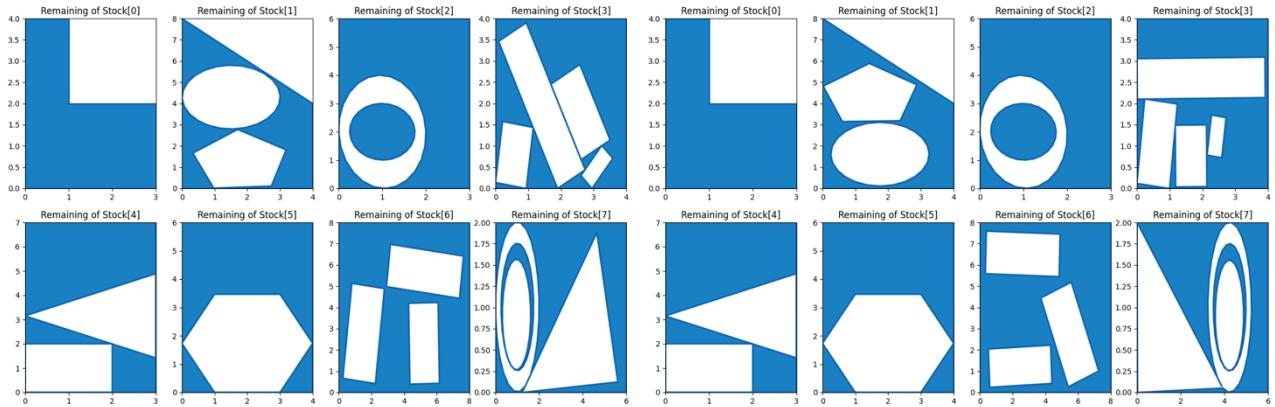


Εικόνα 17 Αποτελέσματα κοπής αποθεμάτων για τις διαθέσιμες παραγγελίες με χρήση του αλγορίθμου DE για την στρατηγική best1exp, Cr=0.7 & F=[0.5-0.75]

Weights of fitness function criteria: $w_{f_OUT}=5000.00, w_{f_OVERLAP}=5000.00, w_{f_ATTR}=3.00, w_{f_SMO}=100.00, w_{f_DIST}=3.00$			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	390.288	47.629	16/17
Rep2	298.327	45.35	16/17

Πίνακας 17 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best1exp, $Cr=0.7$ & $F=[0.5-0.75]$

Best1exp $Cr=0.7$ & $F=[0.5-1.75]$

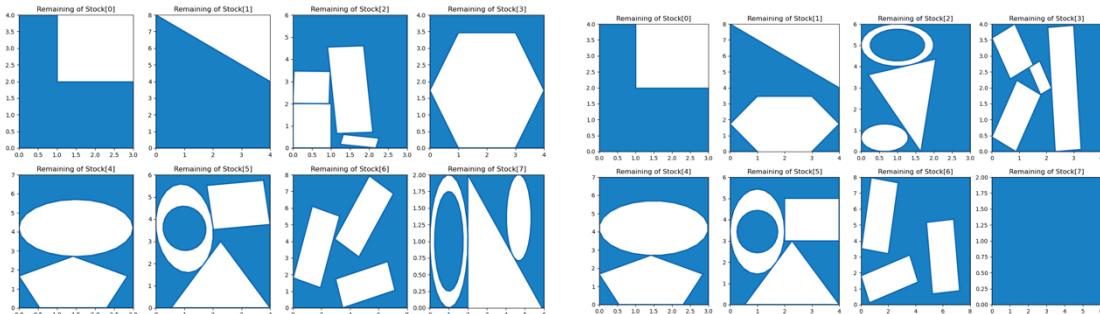


Εικόνα 18 Αποτελέσματα κοπής αποθεμάτων για τις διαθέσιμες παραγγελίες με χρήση του αλγορίθμου DE για την στρατηγική best1exp, $Cr=0.7$ & $F=[0.5-1.75]$

Weights of fitness function criteria: $w_{f_OUT}=5000.00, w_{f_OVERLAP}=5000.00, w_{f_ATTR}=3.00, w_{f_SMO}=100.00, w_{f_DIST}=3.00$			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	372.73	48.25	16/17
Rep2	355.03	48.14	16/17

Πίνακας 18 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best1exp, $Cr=0.7$ & $F=[0.5-0.75]$

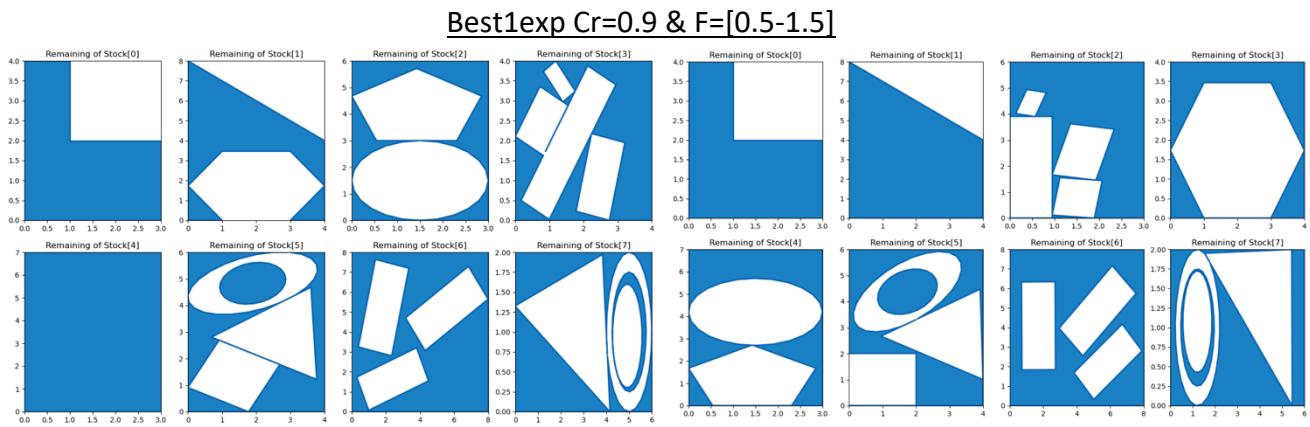
Best1exp $Cr=0.4$ & $F=[0.5-1.5]$



Εικόνα 19 Αποτελέσματα κοπής αποθεμάτων για τις διαθέσιμες παραγγελίες με χρήση του αλγορίθμου DE για την στρατηγική best1exp, $Cr=0.4$ & $F=[0.5-1.5]$

Weights of fitness function criteria:			
w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	309.27	50.0	16/17
Rep2	280.72	50.0	16/17

Πίνακας 19 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best1exp, Cr=0.4 & F=[0.5-1.5]



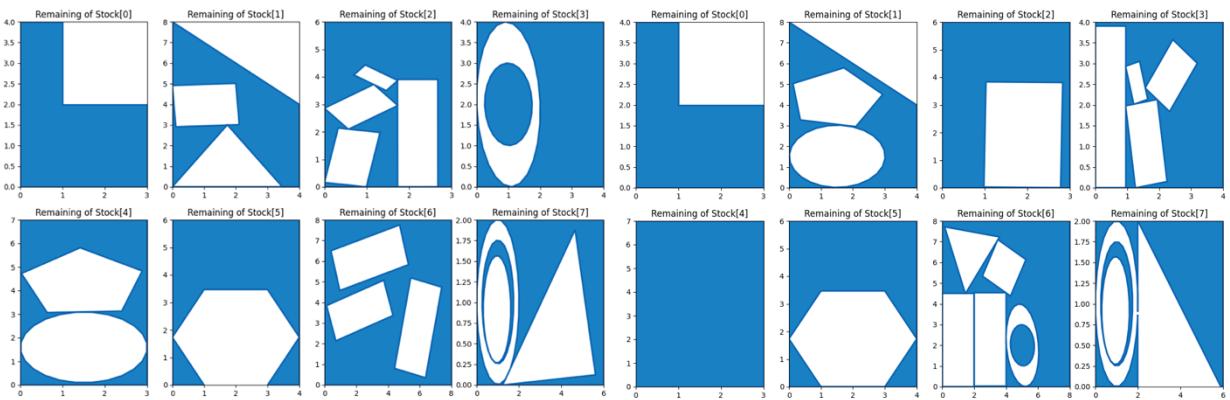
Εικόνα 20 Αποτελέσματα κοπής αποθεμάτων για τις διαθέσιμες παραγγελίες με χρήση του αλγορίθμου DE για την στρατηγική best1exp, Cr=0.9 & F=[0.5-1.5]

Weights of fitness function criteria:			
w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	262.41	45.55	16/17
Rep2	288.14	47.0	16/17

Πίνακας 20 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best1exp, Cr=0.9 & F=[0.5-1.5]

Οι διαφοροποιήσεις στον αλγόριθμο Best1exp τόσο του συντελεστή διασταύρωσης και του συντελεστή συσχέτισής δεν οδηγούν σε βέλτιστες λύσεις. Δεν τοποθετούνται όλες οι παραγγελίες εντός του αποθέματος και τα σχήματα μοιάζουν σαν να τοποθετήθηκαν τυχαία στο stock.

Best2bin Cr=0.7 & F=[0.5-0.75]



Εικόνα 21 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best2bin, Cr=0.7 & F=[0.5-0.75]

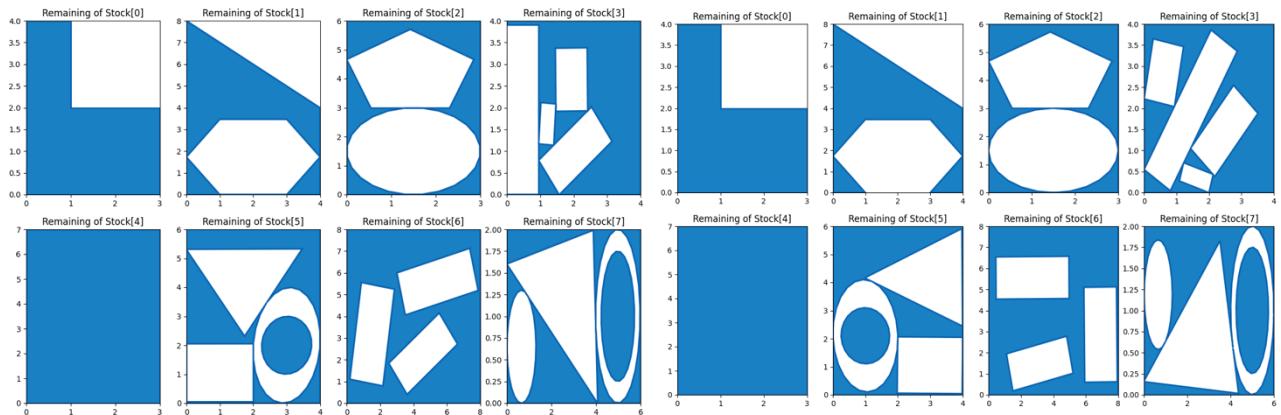
Weights of fitness function criteria:

w_f_OUT=5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00

	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	402.054	45.25	16/17
Rep2	345.653	44.39	16/17

Πίνακας 21 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best2bin, Cr=0.7 & F=[0.5-0.75]

Best2bin Cr=0.7 & F=[0.5-1.75]



Εικόνα 22 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best2bin, Cr=0.7 & F=[0.5-1.75]

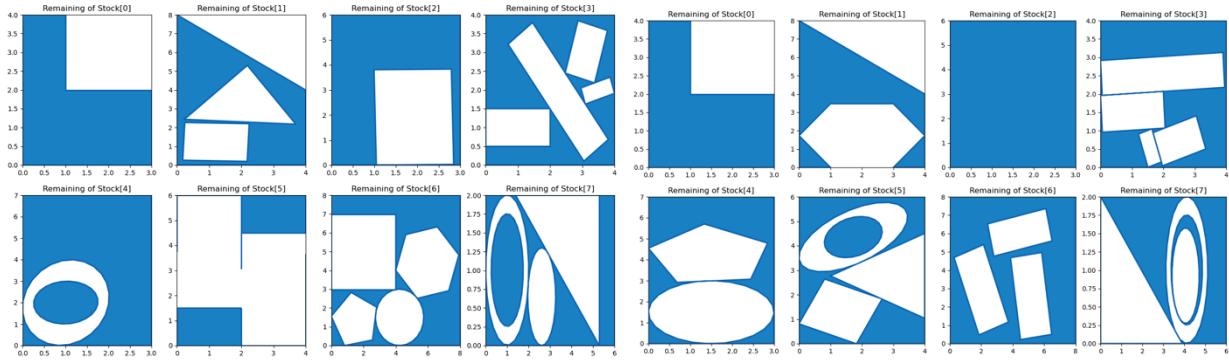
Weights of fitness function criteria:

w_f_OUT=5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00

	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	308.22	47.15	16/17
Rep2	325.39	48.65	16/17

Πίνακας 22 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best2bin, Cr=0.7 & F=[0.5-1.75]

Best2bin Cr=0.4 & F=[0.5-1.5]

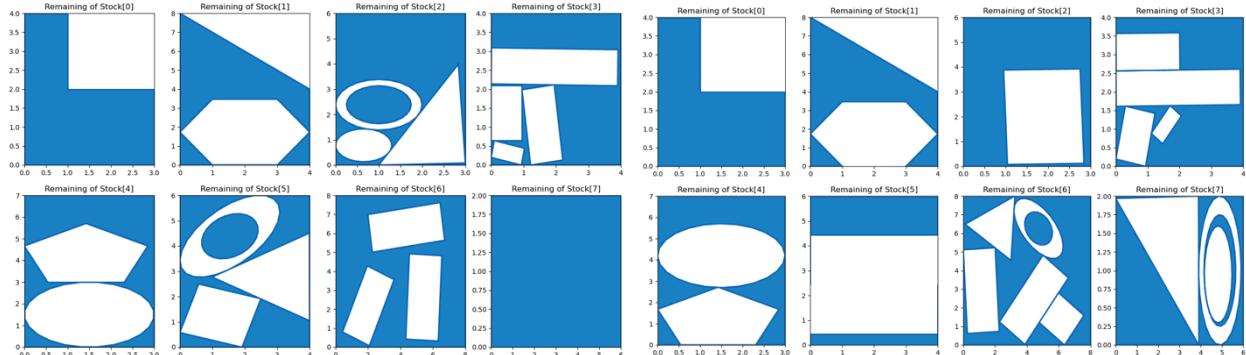


Εικόνα 23 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best2bin, Cr=0.4 & F=[0.5-1.5]

Weights of fitness function criteria: w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	360.30	49.96	17/17
Rep2	281.73	47.95	16/17

Πίνακας 23 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best2bin, Cr=0.4 & F=[0.5-1.5]

Best2bin Cr=0.9 & F=[0.5-1.5]



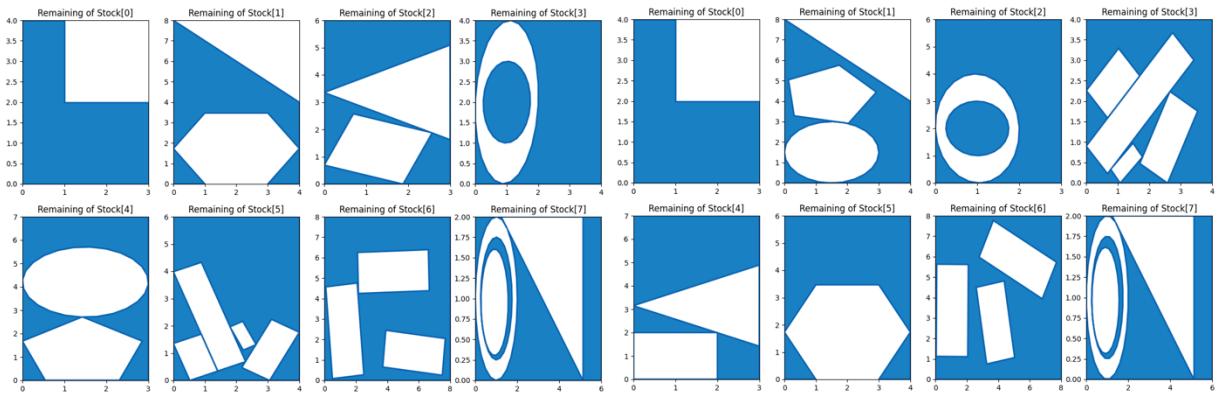
Εικόνα 24 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best2bin, Cr=0.9 & F=[0.5-1.5]

Weights of fitness function criteria: w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	271.74	47.55	16/17
Rep2	276.65	49.33	17/17

Πίνακας 24 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική best2bin, Cr=0.9 & F=[0.5-1.5]

Οι διαφοροποιήσεις στον αλγόριθμο Best2bin τόσο του συντελεστή διασταύρωσης και του συντελεστή συσχέτισής δεν οδηγούν σε βέλτιστες λύσεις. Δεν τοποθετούνται όλες οι παραγγελίες εντός του αποθέματος και τα σχήματα μοιάζουν σαν να τοποθετήθηκαν τυχαία στο stock.

Currenttobest1exp Cr=0.7 & F=[0.5-0.75]

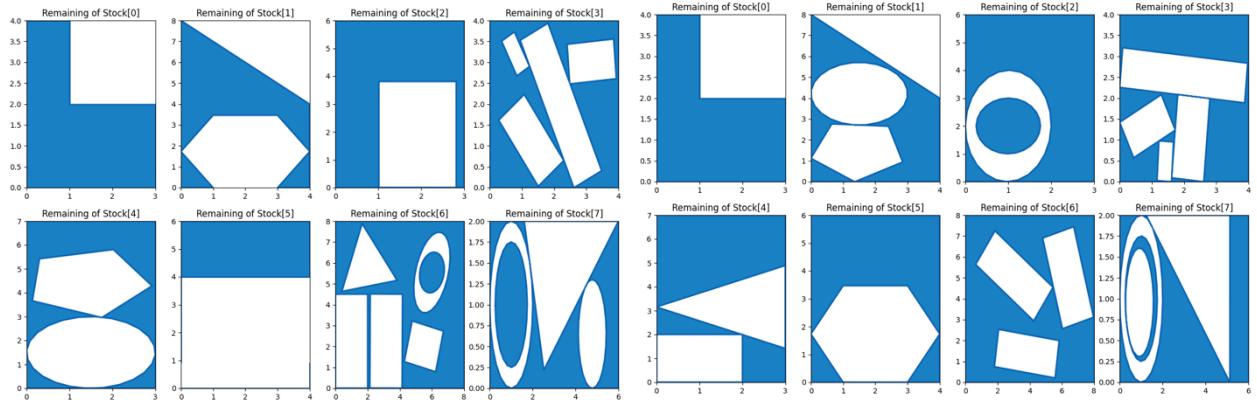


Εικόνα 25 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική currenttobest1exp, Cr=0.7 & F=[0.5-0.75]

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	483.087	49.25	16/17
Rep2	364.251	46.92	16/17

Πίνακας 25 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική currenttobest1exp, Cr=0.7 & F=[0.5-1.5]

Currenttobest1exp Cr=0.7 & F=[0.5-1.75]

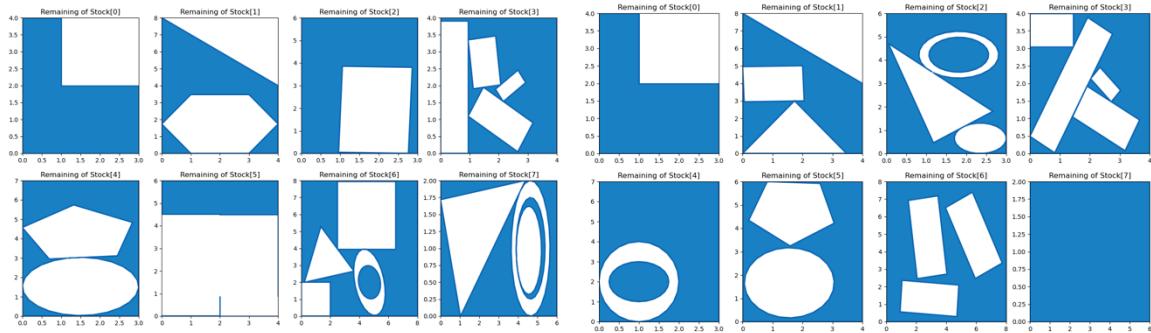


Εικόνα 26 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική currenttobest1exp, Cr=0.7 & F=[0.5-1.75]

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	299.59	49.25	17/17
Rep2	369.25	49.14	16/17

Πίνακας 26 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική currenttobest1exp, Cr=0.7 & F=[0.5-1.75]

Currenttobest1exp Cr=0.4 & F=[0.5-1.5]

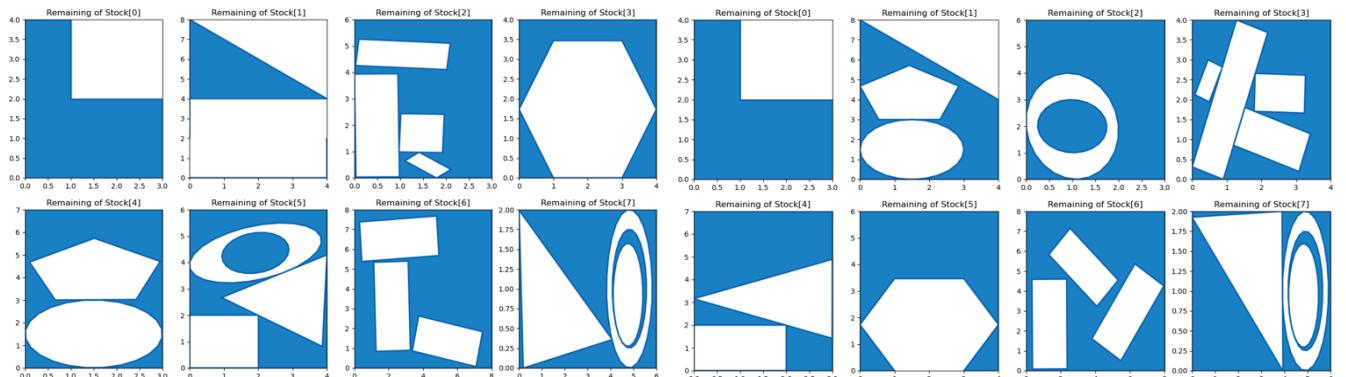


Εικόνα 27 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική currenttobest1exp, Cr=0.4 & F=[0.5-1.5]

Weights of fitness function criteria: w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	336.73	50.0	17/17
Rep2	361.61	50.0	15/17

Πίνακας 27 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική currenttobest1exp, Cr=0.4 & F=[0.5-1.5]

Currenttobest1exp Cr=0.9 & F=[0.5-1.5]



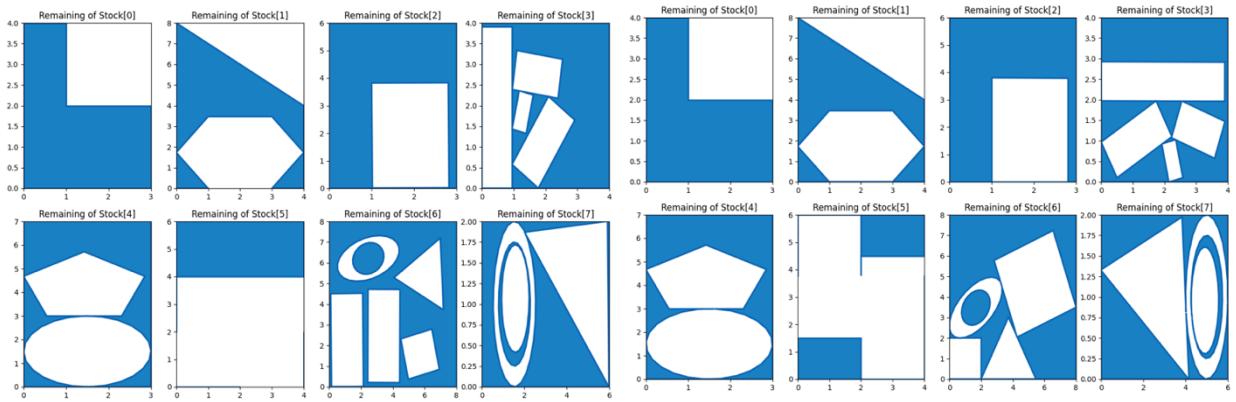
Εικόνα 28 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική currenttobest1exp, Cr=0.9 & F=[0.5-1.5]

Weights of fitness function criteria: w_f_OUT:5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	301.56	47.38	1717
Rep2	311.56	45.85	16/17

Πίνακας 28 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική currenttobest1exp, Cr=0.9 & F=[0.5-1.5]

Οι διαφοροποιήσεις στον αλγόριθμο currenttobest1exp τόσο του συντελεστή διασταύρωσης και του συντελεστή συσχέτισής δεν οδηγούν σε βέλτιστες λύσεις. Δεν τοποθετούνται όλες οι παραγγελίες εντός του αποθέματος και τα σχήματα μοιάζουν σαν να τοποθετήθηκαν τυχαία στο stock.

Rand1bin Cr=0.7 & F=[0.5-0.75]

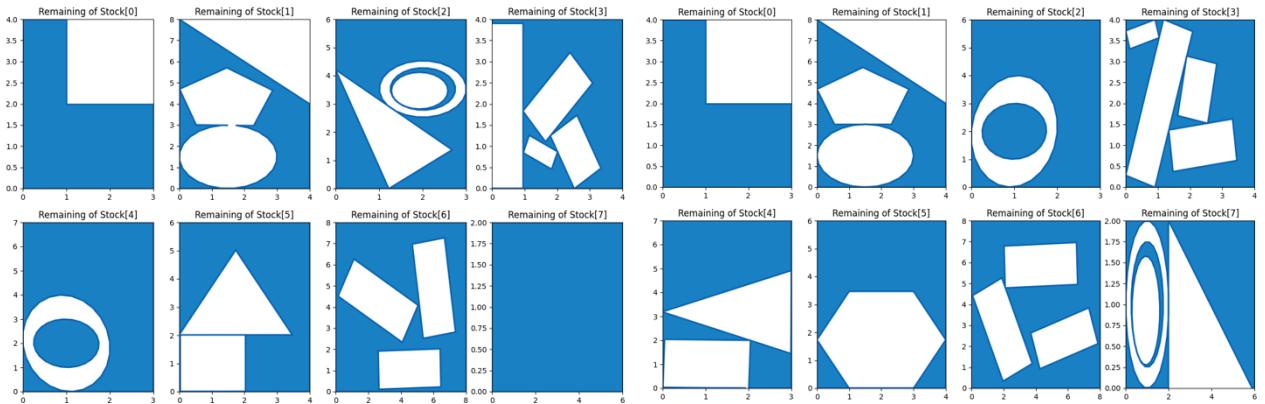


Εικόνα 29 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική rand1bin, Cr=0.7 & F=[0.5-0.75]

Weights of fitness function criteria: w_f_OUT=5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	306.8414	46.75	17/17
Rep2	397.1331	45.625	17/17

Πίνακας 29 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική rand1bin, Cr=0.7 & F=[0.5-0.75]

Rand1bin Cr=0.7 & F=[0.5-1.75]

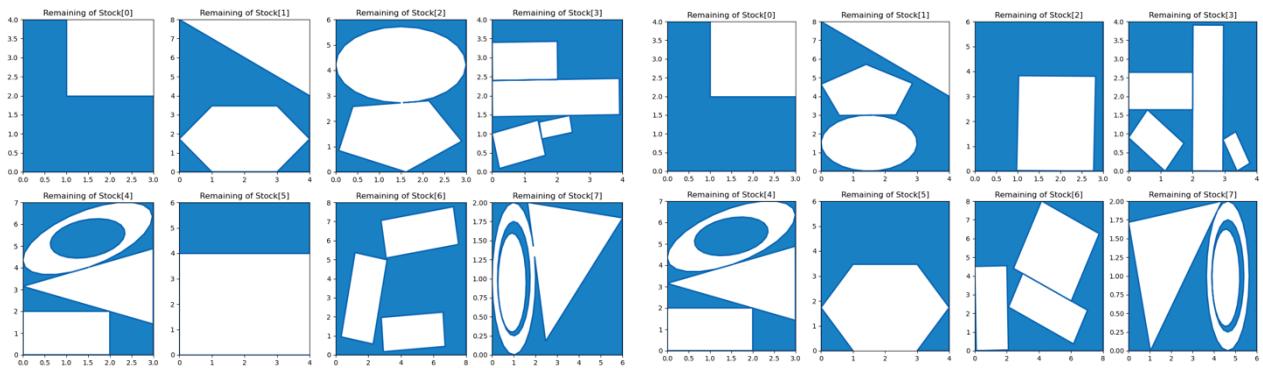


Εικόνα 30 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική rand1bin, Cr=0.7 & F=[0.5-1.75]

Weights of fitness function criteria: w_f_OUT=5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	392.34	45.16	15/17
Rep2	346.58	47.82	16/17

Πίνακας 30 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική rand1bin, Cr=0.7 & F=[0.5-1.75]

Rand1bin Cr=0.4 & F=[0.5-1.5]



Εικόνα 31 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική rand1bin, Cr=0.4 & F=[0.5-1.5]

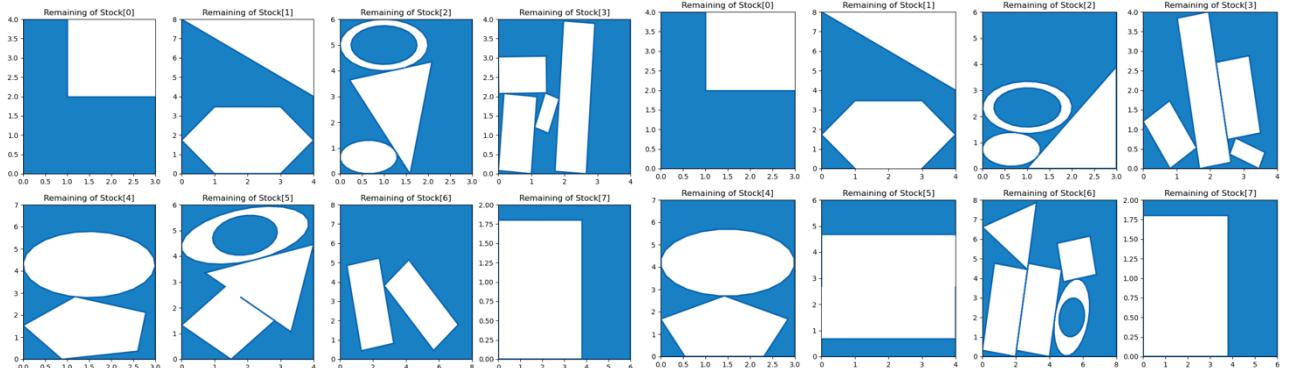
Weights of fitness function criteria:

$w_{f_OUT}=5000.00, w_{f_OVERLAP}=5000.00, w_{f_ATTR}=3.00, w_{f_SMO}=100.00, w_{f_DIST}=3.00$

	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	241.68	49.53	17/17
Rep2	255.21	47.43	17/17

Πίνακας 31 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική rand1bin, Cr=0.4 & F=[0.5-1.5]

Rand1bin Cr=0.9 & F=[0.5-1.5]



Εικόνα 32 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική rand1bin, Cr=0.9 & F=[0.5-1.5]

Weights of fitness function criteria:

$w_{f_OUT}=5000.00, w_{f_OVERLAP}=5000.00, w_{f_ATTR}=3.00, w_{f_SMO}=100.00, w_{f_DIST}=3.00$

	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	306.89	47.40	16/17
Rep2	303.59	48.73	17/17

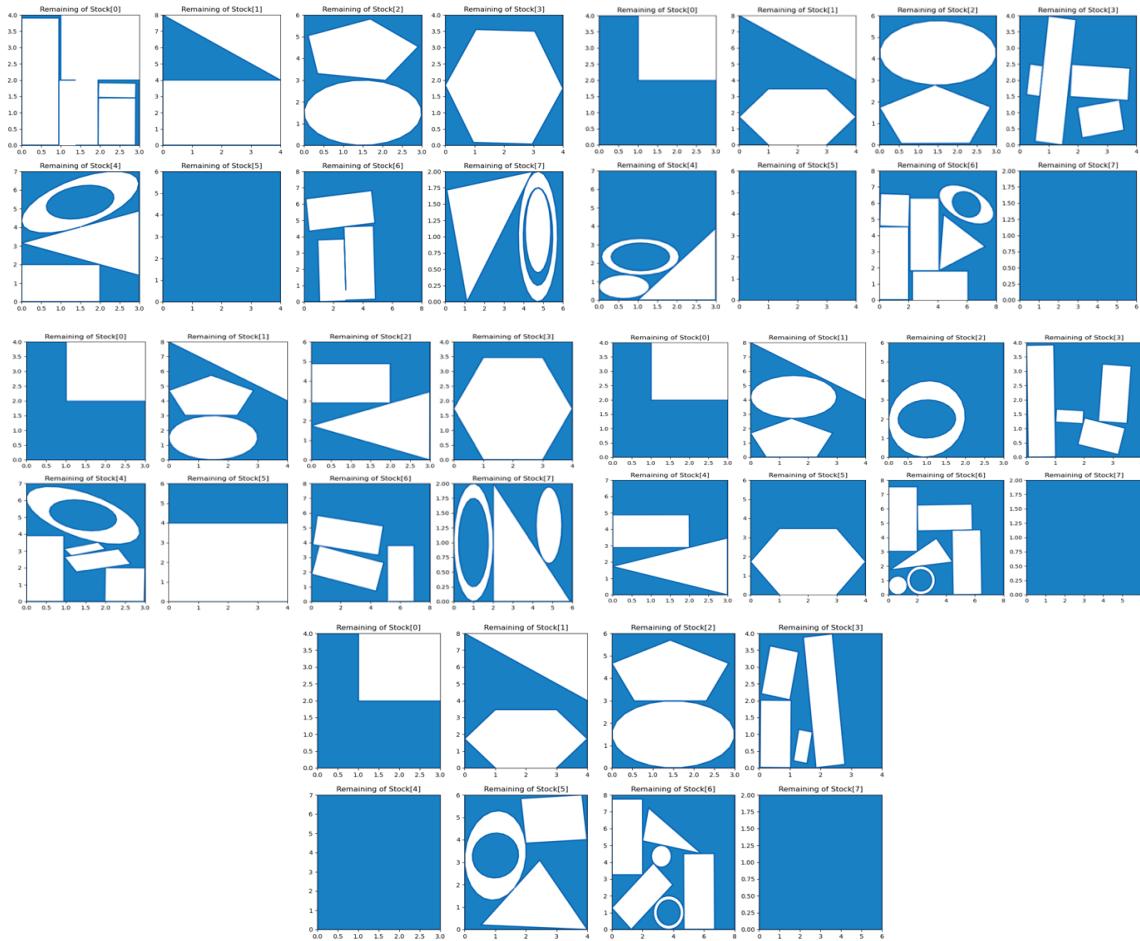
Πίνακας 32 Αποτελέσματα ανάλυσης αλγορίθμου με χρήση του DE για την στρατηγική rand1bin, Cr=0.9 & F=[0.5-1.5]

Οι διαφοροποιήσεις στον αλγόριθμο rand1bin τόσο του συντελεστή διασταύρωσης και του συντελεστή συσχέτισής δεν οδηγούν σε βέλτιστες λύσεις. Δεν τοποθετούνται όλες οι παραγγελίες εντός του αποθέματος και τα σχήματα μοιάζουν σαν να τοποθετήθηκαν τυχαία στο stock.

Επίλυση του προβλήματος με τον Differential Evolution with Global Local Optimization Algorithm

Ο αλγόριθμος DEGL (Διαφορική εξέλιξη με καθολικές και τοπικές γειτονιές) έχει κάνει κάποιες βελτιώσεις σε σχέση με τον DE στον χώρο βελτιστοποίησης. Η διαφορά με την τυπική DE είναι ότι θεωρείται ένας πιο εξελιγμένο τελεστής μετάλλαξης. Η μεθοδολογία του DEGL βασίζεται στην τοπολογία δακτυλίου.

Αποτελέσματα επίλυσης DEGL



Εικόνα 33 Αποτελέσματα κοπής του αποθέματος με τις default τιμές του αλγορίθμου DEGL

Weights of fitness function criteria:			
w_{f_OUT} : 5000.00, $w_{f_OVERLAP}$ =5000.00, w_{f_ATTR} =3.0, w_{f_SMO} =100.00, w_{f_DIST} =3.0			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	543.55	166.30	17/17
Rep2	770.84	132.04	16/17
Rep3	946.71	136.35	17/17
Rep4	1305.72	166.16	16/17
Rep5	888.48	160.62	16/17

Πίνακας 33 Αποτελέσματα εκτέλεσης αλγορίθμου DEGL για τις default τιμές του

Από τα αποτελέσματα της εικόνας 33 φαίνεται ότι ο αλγόριθμος δεν τοποθετεί τα κομμάτια των παραγγελιών με τον βέλτιστο τρόπο. Από εκτέλεση σε εκτέλεση τα αποτελέσματα διαφοροποιούνται αισθητά. Επιπλέον σε τρεις από τις πέντε εκτελέσεις δεν κατάφερε να εντάξει όλα τα κομμάτια εντός του αποθέματος καθώς τοποθέτησε μόνο 16/17. Εντυπωσιακοί είναι οι χρόνοι εκτέλεσης του που είναι περίπου γύρω στα 890 second, χρόνος που είναι αρκετά αυξημένος σε σχέση με όλους τους παραπάνω αλγορίθμους.

Επίλυση του αλγορίθμου DEGL διαφοροποιώντας κάποιες από τις παραμέτρους του.

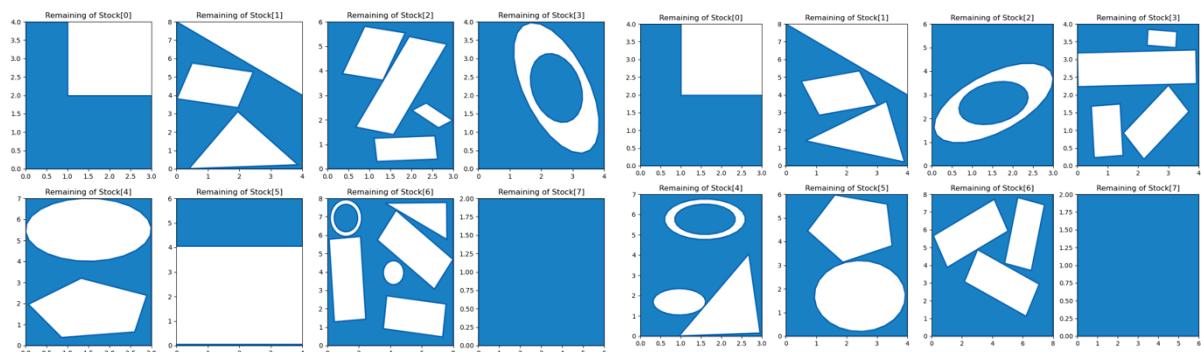
Οι συντελεστές κλιμάκωσης α και β .

Η ειδική μετάλλαξη της DEGL πραγματοποιείται δημιουργώντας αρχικά για κάθε χρωμόσωμα z_i ένα νέο διάνυσμα, λαμβάνοντας υπόψη μόνο την γειτονιά του, στη φιλοσοφία της DE/ctb/1/bin:

$$L_i = z_i + \alpha \cdot (z_{best\ i} - z_i) + \beta \cdot (z_p - z_q),$$

όπου α και β συντελεστές κλιμάκωσης,
 p και q τυχαίοι δείκτες εντός του διαστήματος $[i-k, i+k]$,
 για τους οποίους ισχύει $p \neq q \neq i$ και z_{besti} το χρωμόσωμα της γειτονιάς N_i που είχε την υψηλότερη τιμή καταλληλότητας την προηγούμενη γενιά.

Έλεγχο για $a=b=0.4$



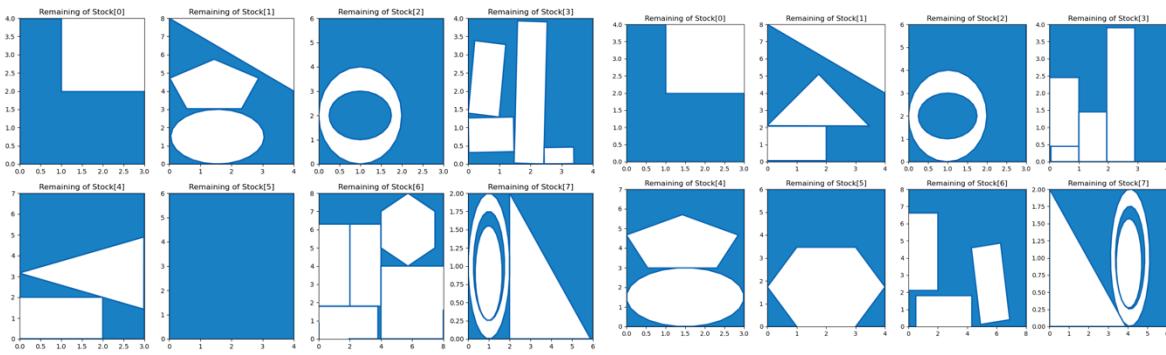
Εικόνα 34 Κοπή του αποθέματος του αλγορίθμου DEGL για $a=b=0.4$

Weights of fitness function criteria:			
w_f_OUT:5000.00, w_f_OVERLAP:5000.00, w_f_ATTR=3.00, w_f_SMO=100.00, w_f_DIST=3.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	824.84	113.82	16/17
Rep2	767.11	91.43	15/17

Πίνακας 34 Επίλυση του αλγορίθμου DEGL για $a=b=0.4$

Μειώνοντας τους συντελεστές α και β από 0.8 που είναι οι default τιμές σε 0.4 η επίδοση του αλγορίθμου δεν βελτιώνεται. Τα κομμάτια των παραγγελιών δεν τοποθετούνται σωστά και στην δεύτερη επανάληψη μένουν δύο κομμάτια εκτός του αποθέματος. Φαίνεται πως ο αλγόριθμος τοποθετεί τις παραγγελίες σχεδόν τυχαία.

Έλεγχο για $a=b=0.9$



Εικόνα 35 Κοπή του αποθέματος του αλγορίθμου DEGL για $a=b=0.9$

Weights of fitness function criteria:			
$w_{f_OUT}=5000.00, w_{f_OVERLAP}=5000.00, w_{f_ATTR}=3.00, w_{f_SMO}=100.00, w_{f_DIST}=3.00$			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	778.99	118.65	17/17
Rep2	871.05	108.62	16/17

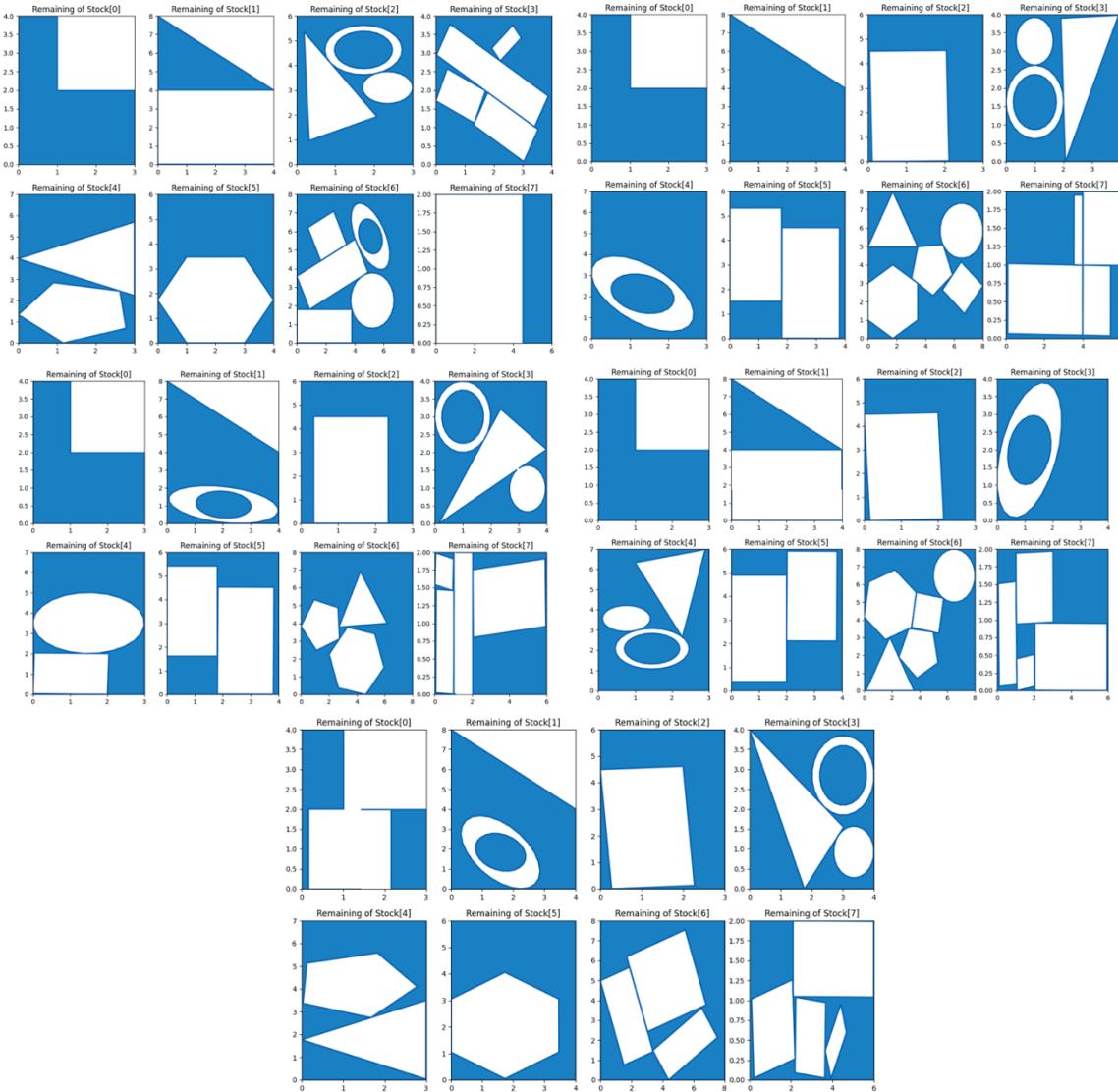
Πίνακας 35 Επίλυση του αλγορίθμου DEGL για $a=b=0.9$

Δεδομένου, ότι η μείωση των συντελεστών δεν έχει ευεργετική επίδραση στον αλγόριθμο, αυτή την φορά επιλέχθηκε να αυξηθούν οι συντελεστές a και b και να πάρουν την τιμή 0.9. Τα αποτελέσματα είναι καλύτερα σε σχέση με αυτά που φαίνονται στις εικόνες 34 και 33. Τα σχήματα δεν απέχουν πολύ μεταξύ τους παρόλο που δεν είναι τοποθετημένα κατά τον βέλτιστο τρόπο. Γενικά, για την επίλυση του συγκεκριμένου προβλήματος προτείνεται χρήση μεγάλων τιμών στους συντελεστές a και b .

Χρήση εναλλακτικών, μη εξελικτικών αλγορίθμων για την επίλυση του προβλήματος

Εκτός από τους εξελικτικούς αλγορίθμους επιλέχθηκαν και απλοί εναλλακτικοί που μπορούν να προσεγγίσουν κάποιες λύσεις του προβλήματος. Τα αρχικό γονίδιο χο για αυτούς επιλέγεται κάθε φορά τυχαία και κυμαίνεται μεταξύ του Lower και Upper bound του stock που μελετάται.

Επίλυση του προβλήματος με τον SLSQP algorithm



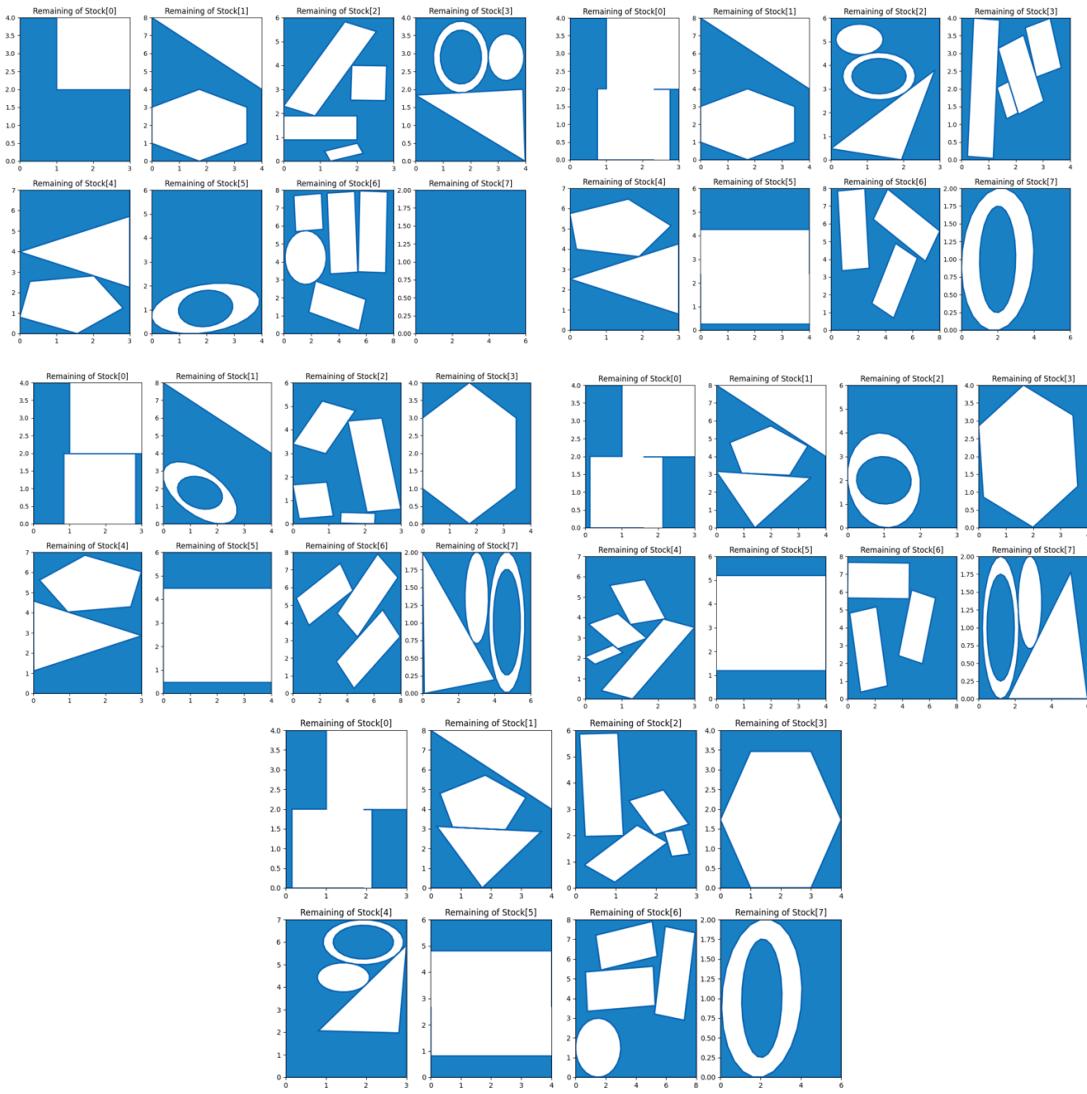
Εικόνα 36 Κοπή του αποθέματος με την χρήση του αλγορίθμου SLSQC

Weights of fitness function criteria: w_f_OUT=500.00, w_f_OVERLAP=500.00, w_f_ATTR= 0.01, w_f_SMO=500, w_f_DIST=1.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	51.24	84.67	17/17
Rep2	57.82	74.32	16/17
Rep3	66.78	89.08	16/17
Rep4	42.13	65.13	17/17
Rep5	38.57	51.04	16/17

Πίνακας 36 Αποτελέσματα εκτέλεσης με την χρήση του αλγορίθμου SLSQC

Λαμβάνοντας υπόψη τα αποτελέσματα της εικόνας 36 καθώς και του χρόνου εκτέλεσης του αλγορίθμου που φαίνεται στον πίνακα 36, παρατηρείται πως ο αλγόριθμος SLSQC επιλύει ικανοποιητικά το πρόβλημα. Ο μέσος χρόνος εκτέλεσης του είναι τα 51 second και καταφέρνει τα τοποθετήσει όλα τα κομμάτια μέσα στο απόθεμα δύο στις πέντε φορές. Γενικά συγκρίνοντας τον με εξελικτικούς αλγορίθμους όπως ο DE και ο DEGL συμπεραίνεται πως δεν τοποθετεί τόσο άσχημα τις παραγγελίες μέσα στο stock.

Επίλυση του προβλήματος με τον Nelder-Mead algorithm



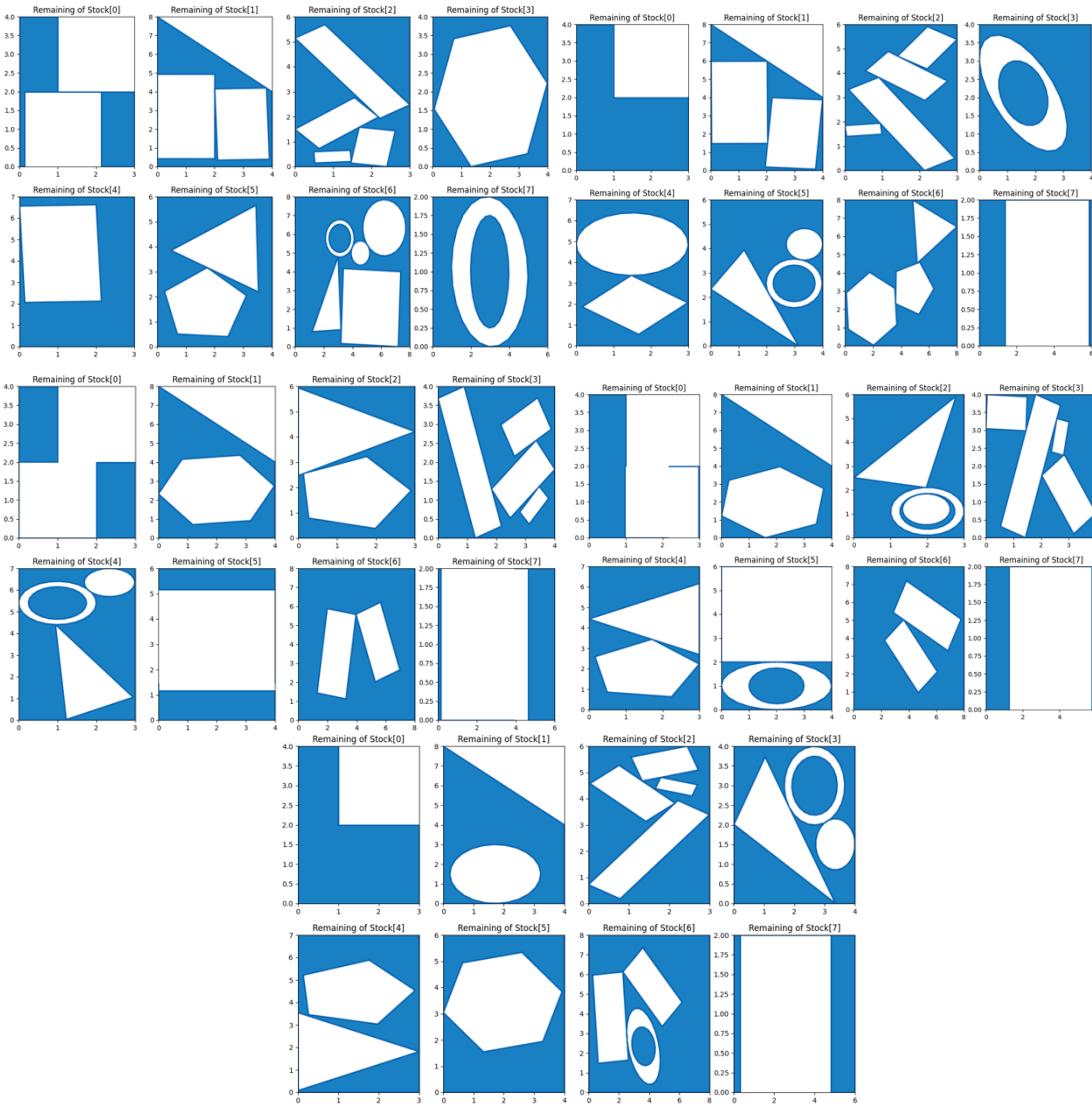
Εικόνα 37 Κοπή του αποθέματος με την χρήση του αλγορίθμου Nelder-Mead

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	143.05	1567.54	16/17
Rep2	138.908	1266.30	16/17
Rep3	154.63	1469.23	16/17
Rep4	141.50	1490.62	16/17
Rep5	166.58	1551.41	17/17

Πίνακας 37 Αποτελέσματα εκτέλεσης με την χρήση του αλγορίθμου Nelder-Mead

Η λύση που προτείνει ο αλγόριθμος Nelder-Mead φαίνεται πώς δεν είναι τόσο καλή σε σχέση με του SLSQC. Οι χρόνοι εκτέλεσής του είναι μεγαλύτεροι και το ποσοστό επιτυχίας του μικρότερο. Ωστόσο, τα αποτελέσματα που προτείνει ανταγωνίζονται αυτά των δύο τελευταίων εξελικτικών αλγορίθμων.

Επίλυση του προβλήματος με τον L-BFGS-B algorithm



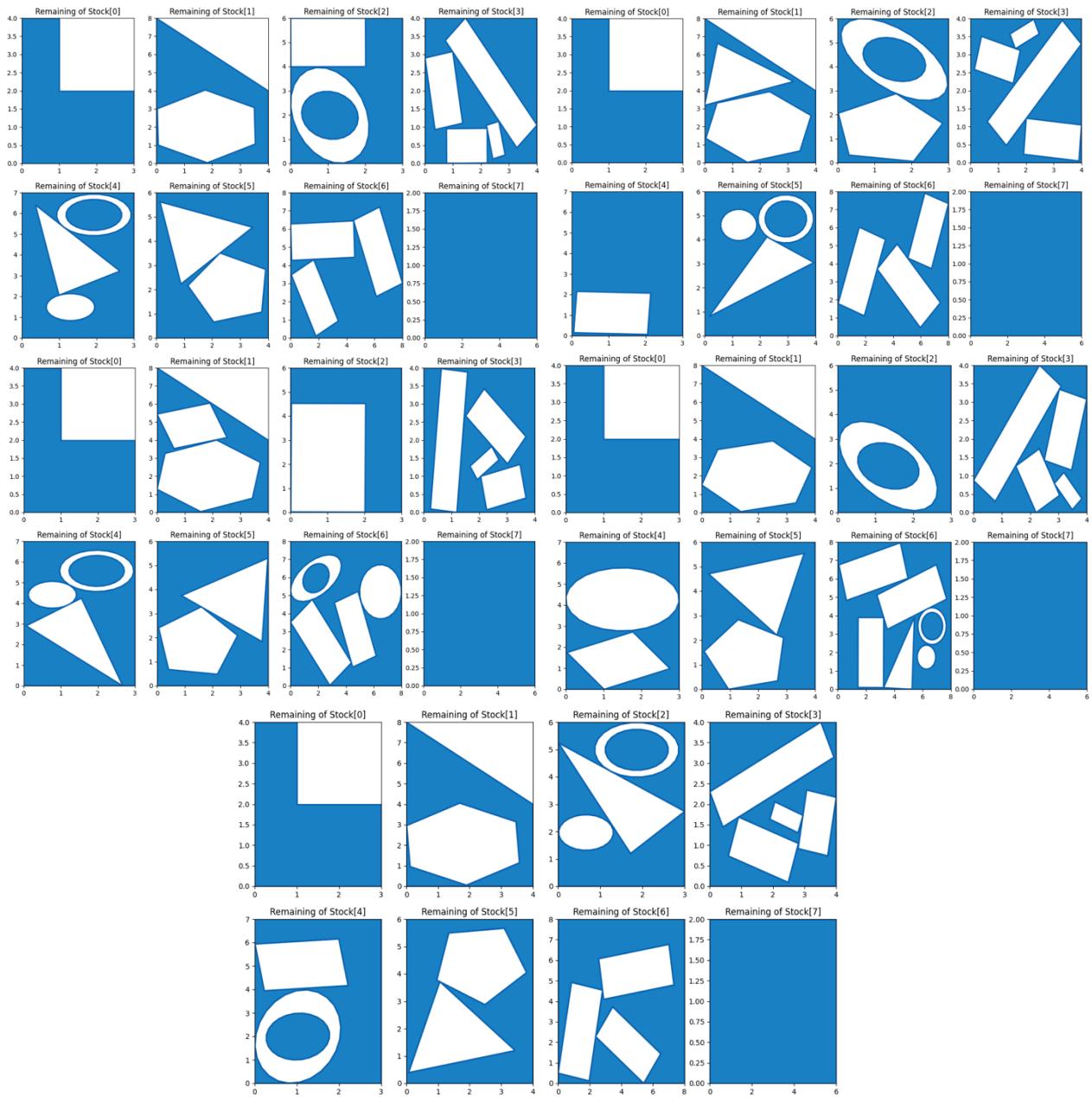
Εικόνα 38 Κοπή του αποθέματος με την χρήση του αλγορίθμου L-BFGS-B

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	55.88	26.21	17/17
Rep2	62.82	41.45	16/17
Rep3	36.82	28.04	15/17
Rep4	39.68	29.67	16/17
Rep5	41.07	24.35	15/17

Πίνακας 38 Αποτελέσματα εκτέλεσης με την χρήση του αλγορίθμου L-BFGS-B

Η λύση που προτείνει ο αλγόριθμος L-BFGS δεν είναι κοστοβόρα. Ωστόσο, δύο από τις πέντε εκτελέσεις του αφήσαν εκτός του αποθέματος δύο κομμάτια των παραγγελιών ενώ μόνο σε μία εκτέλεση κατάφερε να τοποθετήσει επιτυχώς 17/17 κομμάτια.

Επίλυση του προβλήματος με τον Pattern search algorithm



Εικόνα 39 Κοπή του αποθέματος με την χρήση του αλγορίθμου Pattern search

Weights of fitness function criteria:			
	Time (sec)	Number of Iterations (avg)	Fitted
W_f_OUT:500.00, w_f_OVERLAP=500.00, w_f_ATTR= 0.01, w_f_SMO=500, w_f_DIST=1.00			
Rep1	154.36	96.84	15/17
Rep2	176.24	100.74	15/17
Rep3	148.73	88.39	16/17
Rep4	149.30	113.57	16/17
Rep5	184.03	98.78	15/17

Πίνακας 39 Αποτελέσματα εκτέλεσης με την χρήση του αλγορίθμου Pattern search

Ο αλγόριθμος Pattern Search εμφανίζει την χειρότερη συμπεριφορά για την επίλυση αυτού του προβλήματος σε σχέση με όλους τους μη εξελικτικούς αλγορίθμους. Οι χρόνοι εκτέλεσης του είναι σχετικά μεγάλοι και δεν έχει καταφέρει σε καμία επανάληψη να χωρέσει όλες τις παραγγελίες μέσα στο απόθεμα.

Γενικά, οι μη ευέλικτοι αλγόριθμοι είναι ταχύτεροι από τους εξελικτικούς. Ωστόσο, σε όλες τις παραπάνω αναλύσεις δεν κατάφεραν να βρουν την βέλτιστοι λύση και τοποθετούν τα σχήματα λίγο ακανόνιστα. Ωστόσο, συμπεριφέρονται καλύτερα από τον DEGL (ο οποίος έχει μεγάλους χρόνους εκτέλεσης και μη επιθυμητές λύσεις) και από αρκετές μεταλλάξεις του DE.

Σημειώνεται πως εξελικτικοί και μη χρησιμοποιούν την ίδια συνάρτηση καταλληλότητας με τα ίδια βάρη.

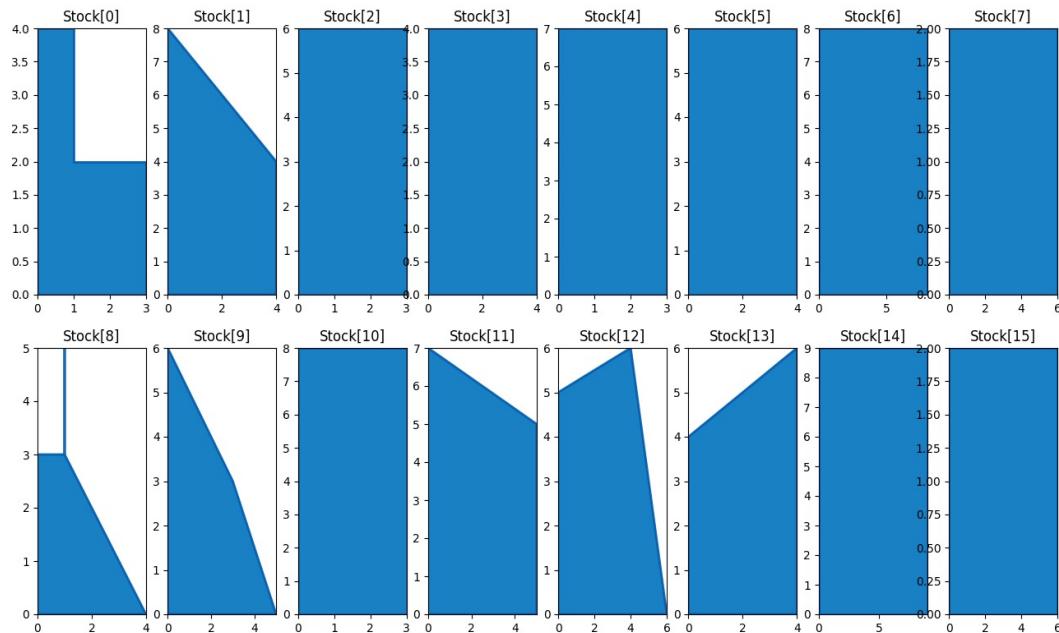
Αύξηση του αποθέματος

Για τις ανάγκες αυτού του ερωτήματος διπλασιάστηκε/τετραπλασιάστηκε τόσο ο αριθμός των κομματιών του αποθέματος όσο και το συνολικό εμβαδό του αποθέματος. Ως αλγόριθμος επίλυσης του προβλήματος επιλέχθηκε ο PSO διότι κατάφερε να χωρέσει όλες τις απαγγελίες μέσα στο απόθεμα και είχε την καλύτερη απόδοση και από άποψη χρόνου.

“Οπως φαίνεται και από τις εικόνες 41 και 43 που ακολουθούν ο διπλασιασμός ή ο τετραπλασιασμός του αποθέματος δεν επιφέρει τρομερές αλλαγές στην επίδοση του αλγορίθμου. Αυτό συμβαίνει διότι πριν ξεκινήσει να εκτελείται ο αλγόριθμος γίνεται μια προεπεξεργασία των δεδομένων όπως περιγράφεται στην ενότητα 1. Τα αποθέματα κατατάσσονται από το μικρότερο προς το μεγαλύτερο και πρέπει να εξασφαλιστεί ο η παραγγελία χωράει εντός του αποθέματος πριν διθεί στον αλγόριθμο.

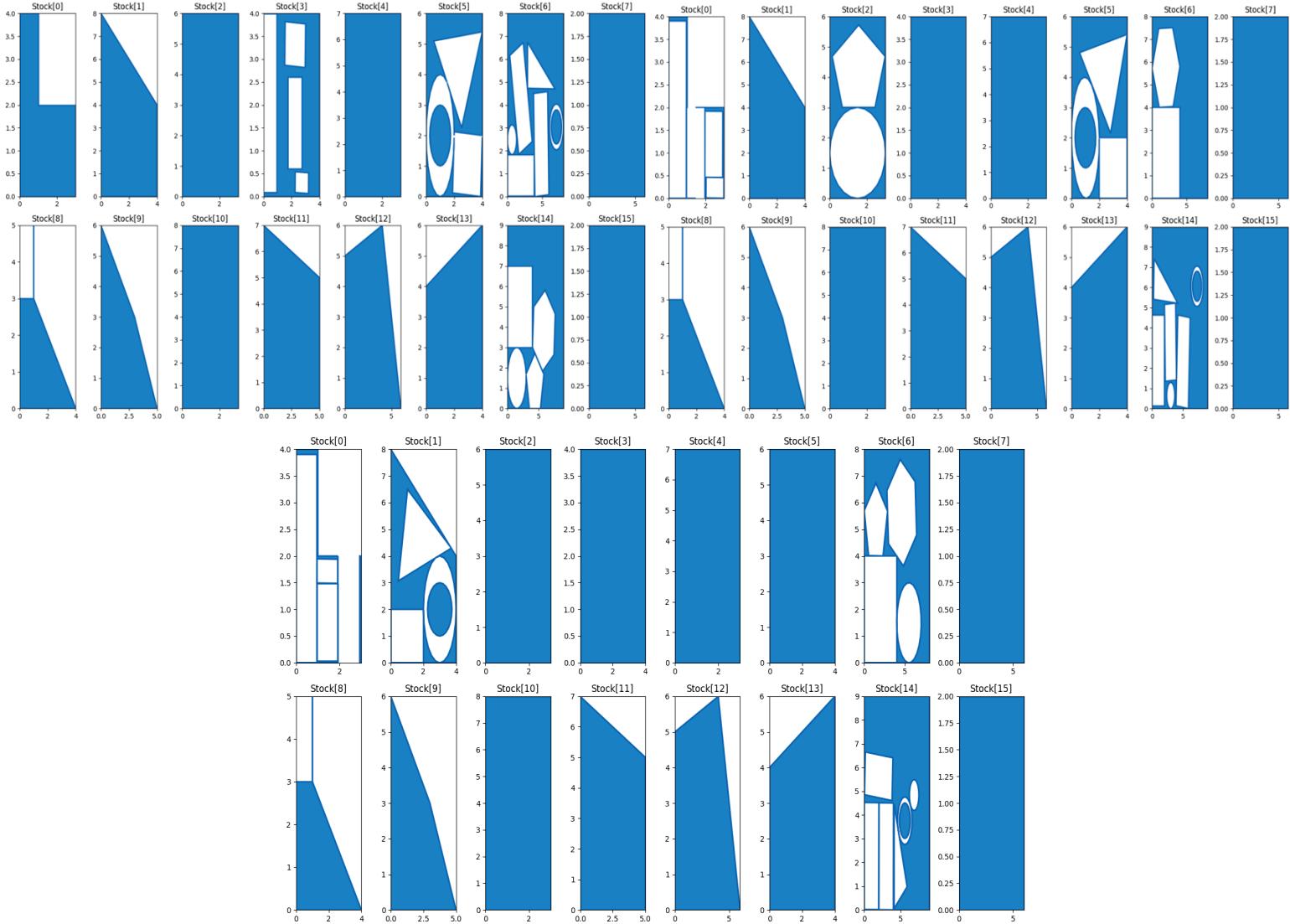
Στις εικόνες 40 και 42 φαίνονται τα αρχικά κομμάτια του αποθέματος μετά τον διπλασιασμό και τετραπλασιασμό του stock.

Διπλασιασμός του αποθέματος



Εικόνα 40 Διπλασιασμός των αποθεμάτων ως προς τον συνολικό αριθμό τους και το εμβαδόν τους

Αποτελέσματα αλγορίθμου PSO μετά από διπλασιασμό του αποθέματος.



Εικόνα 41 Αποτελέσματα PSO κατόπιν διπλασιασμού του αποθέματος

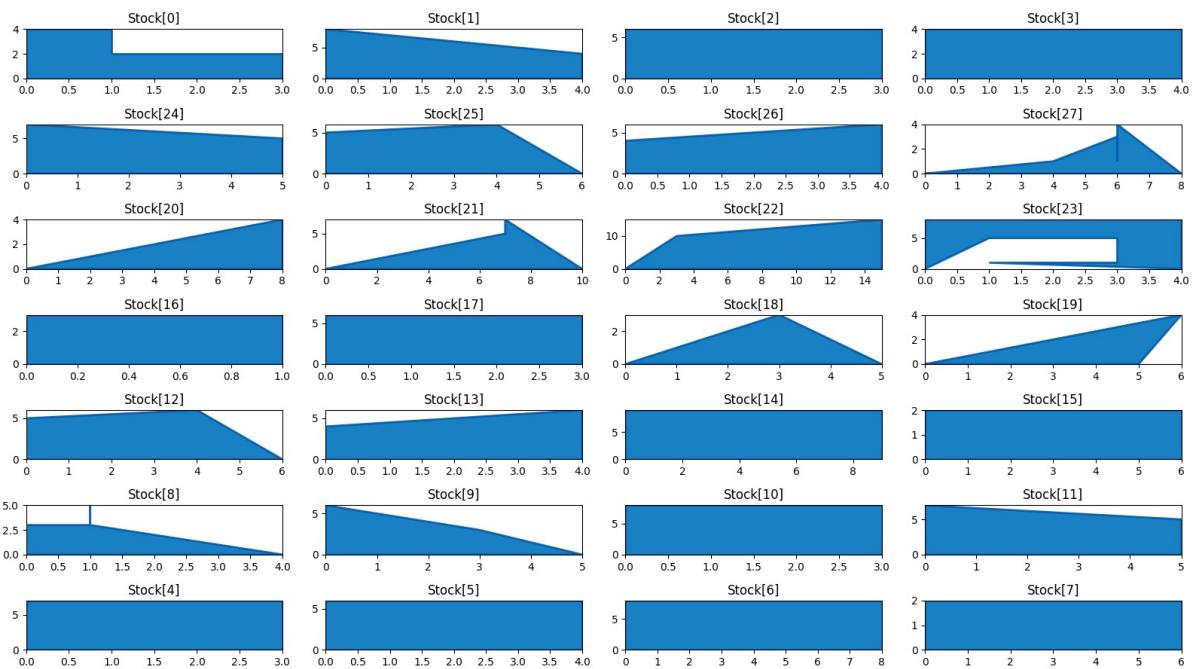
Weights of fitness function criteria:

$w_f_{OUT}=5000.00$, $w_f_{OVERLAP}=5000.00$, $w_f_{ATTR}=3.00$, $w_f_{SMO}=100$, $w_f_{DIST}=5.00$

	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	239.33	58.69	17/17
Rep2	269.26	45.8	17/17
Rep3	291.10	73.83	17/17

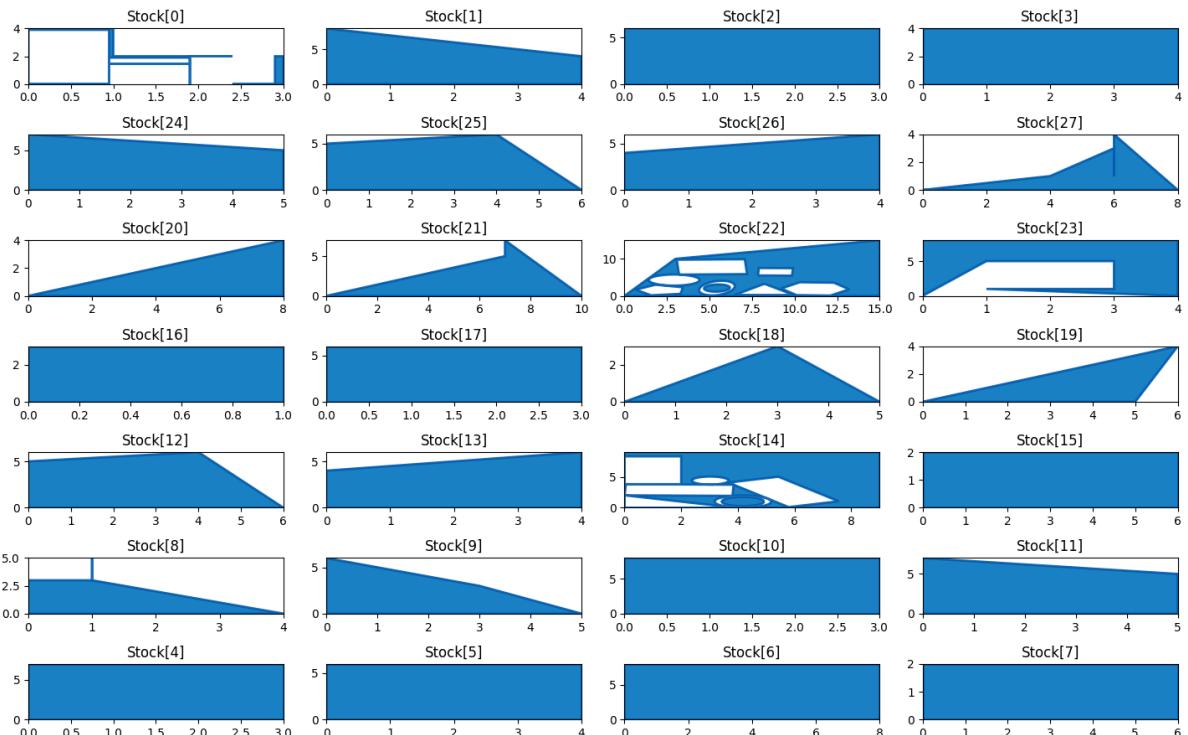
Πίνακας 40 Πίνακας 40 Πίνακας αποτελεσμάτων του PSO για διπλασιασμό του αποθέματος

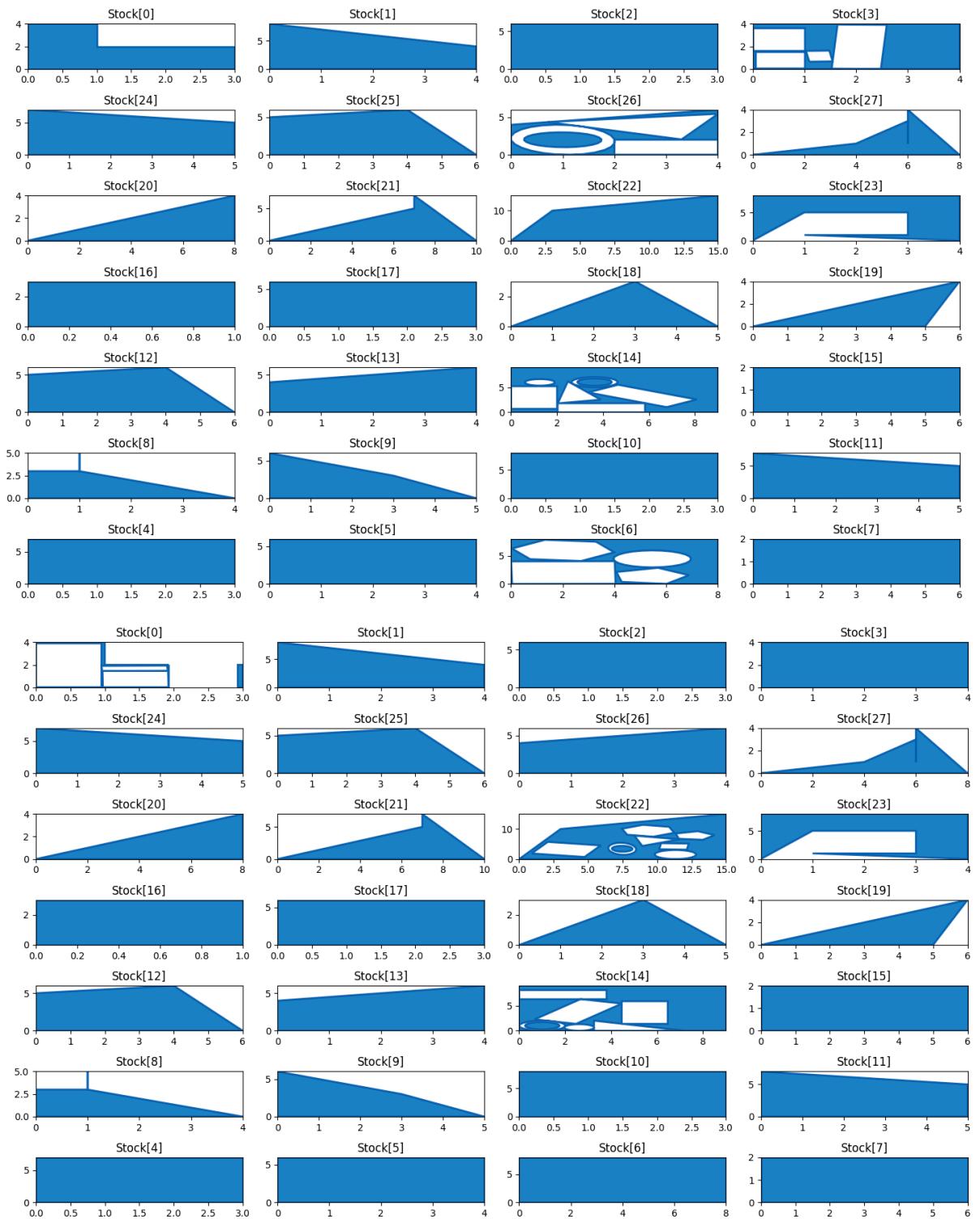
Τετραπλασιασμός του αποθέματος



Εικόνα 42 Τετραπλασιασμός του αποθέματος ως προς τον συνολικό αριθμό τους και το εμβαδόν τους

Αποτελέσματα αλγορίθμου PSO μετά από τετραπλασιασμό του αποθέματος.





Εικόνα 43 Αποτελέσματα PSO κατόπιν τετραπλασιασμού του stock

Weights of fitness function criteria: w_f_OUT=5000.00, w_f_OVERLAP=5000.00, w_f_ATTR=3.00 , w_f_SMO=100, w_f_DIST=5.00			
	Time (sec)	Number of Iterations (avg)	Fitted
Rep1	196.98	105.6	17/17
Rep2	318.81	53.78	17/17
Rep3	217.20	111.8	17/17

REFERENCES

- [1] Lecture Notes, Gianni Di Caro, Collective Intelligence, Lesson 15-382, Carnegie Mellon Universiyt Qatar, 2018
- [2] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261-272.
- [3] Wikipedia. "Particle swarm optimization", Last modified 19 October 2021, https://en.wikipedia.org/wiki/Particle_swarm_optimization
- [4] Gao, Yuelin, Kaiguang Wang, Chenyang Gao, Yulong Shen, and Teng Li. 2019. "Application of Differential Evolution Algorithm Based on Mixed Penalty Function Screening Criterion in Imbalanced Data Integration Classification" *Mathematics* 7, no. 12: 1237. <https://doi.org/10.3390/math7121237>