

Εθνικό Μετσόβιο Πολυτεχνείο

1η εργασία

Ονοματεπώνυμο: Μιχαήλ Κουσκούνης

Αριθμός μητρώου: mc21052

Κατεύθυνση: Ενεργειακή

Πίνακας Περιεχομένων

Εισαγωγή	0
1 Επίλυση προβλήματος βελτιστοποίησης ως βελτιστοποίηση μίας συνάρτησης (SOO).....	1
2 Επίλυση προβλήματος βελτιστοποίησης ως βελτιστοποίηση δύο συναρτήσεων (MOO)	5
3 Επίλυση προβλήματος βελτιστοποίησης ως βελτιστοποίηση Nash	8
4 Επίλυση προβλήματος βελτιστοποίησης ως βελτιστοποίηση SOO με χρήση μεταμοντέλου (MAEA)...	10

Εισαγωγή

Στα πλαίσια της 1^{ης} εργασίας του μαθήματος «Μέθοδοι Βελτιστοποίησης» ζητείται η βελτιστοποίηση δύο συναρτήσεων οι οποίες εξαρτώνται από τέσσερις μεταβλητές σχεδιασμού και υπόκεινται σε πέντε περιορισμούς. Αρχικά, γίνεται επίλυση του προβλήματος συνδυάζοντας τις δύο συναρτήσεις μέσω ενός συντελεστή βάρους και επιλύοντας ένα πρόβλημα βελτιστοποίησης μίας συνάρτησης (SOO). Αυτό γίνεται για ένα συνδυασμό βαρών. Έπειτα, επιλύεται το πρόβλημα ως βελτιστοποίηση πολλών συναρτήσεων (MOO) και συγκρίνονται δύο διαφορετικές μέθοδοι αξιολόγησης των ατόμων. Στη συνέχεια, το πρόβλημα επιλύεται ως ένα παιχνίδι Nash όπου το πρόβλημα χωρίζεται σε δύο υποπροβλήματα τα οποία βελτιστοποιούνται ανεξάρτητα το ένα από το άλλο ανταλλάσσοντας όμως πληροφορία μεταξύ τους. Τέλος, γίνεται επίλυση του SOO προβλήματος με χρήση μεταμοντέλου το οποίο οδηγεί σε εξοικονόμηση κλήσεων της συνάρτησης αξιολόγησης.

1 Επίλυση προβλήματος βελτιστοποίησης ως βελτιστοποίηση μίας συνάρτησης (SOO)

Σε αυτό το ερώτημα μας ζητείται η βελτιστοποίηση μίας συνάρτησης F η οποία αποτελείται από τις συναρτήσεις F_1 και F_2 (εξισώσεις (1) και (2) αντίστοιχα) που συνδέονται μεταξύ τους με την χρήση ενός βάρους k όπως φαίνεται στην εξίσωση (3).

$$F_1 = 4.9 * 10^{-5} * (x_2^2 - x_1^2) * (x_4 - 1) \quad (1)$$

$$F_2 = \frac{9.82 * 10^6 * (x_2^2 - x_1^2)}{(x_2^3 - x_1^3) * x_3 * x_4} \quad (2)$$

$$F = k * F_1 + F_2 \quad (3)$$

Για την εύρεση του εύρους τιμών που είναι εύλογο να πάρει το k καθώς και την συνάρτηση η οποία θα «κουβαλάει» αυτό το βάρος, χρησιμοποιήθηκε ο αλγόριθμος EASY προκειμένου να βρεθούν το ελάχιστο και το μέγιστο των δύο συναρτήσεων. Αυτά τα τρεξίματα έγιναν φυσικά χωρίς περιορισμούς και με όριο 5000 κλήσεις του λογισμικού αξιολόγησης. Έτσι, προέκυψαν οι σχέσεις (4) και (5).

$$-0.72 \leq F_1 \leq 8.44 \quad (4)$$

$$1.1 \leq F_2 \leq 48 \quad (5)$$

Από τις σχέσεις (4) και (5) ότι και οι δύο συναρτήσεις έχουν παρόμοια κάτω όρια όμως το άνω όριο της F_2 είναι περίπου έξι φορές μεγαλύτερο της F_1 , κάτι που μας οδηγεί στο να δώσουμε το βάρος k στην F_1 . Τότε, η αύξηση του k έχει σαν αποτέλεσμα την όλο και μεγαλύτερη ελαχιστοποίηση της F_1 με αντίτιμο την αύξηση της τιμής της F_2 . Αρχικά, επιλέχτηκε το $k_1 = 1.5$, το οποίο δίνει μεγαλύτερη βαρύτητα στην ελαχιστοποίηση της F_2 . Έπειτα, για $k_2 = 5$, η βαρύτητα των δύο συναρτήσεων στην ελαχιστοποίηση της F είναι περίπου ίδια. Για το τελευταίο βάρος επιλέγουμε $k_3 = 10$ ώστε η F_1 να παίζει κυρίαρχο ρόλο στην ελαχιστοποίηση της F .

Όσον αφορά τους περιορισμούς, οι πέντε αρχικοί περιορισμοί που αναγράφονται στην εκφώνηση της εργασίας (εξισώσεις (6), (7), (8), (9) και (10) αντίστοιχα) υλοποιούνται εύκολα στο λογισμικό αξιολόγησης. Το σημείο όμως που παρουσιάζεται πρόβλημα είναι η απαίτηση της μεταβλητής x_4 να παίρνει μόνο ακέραιες τιμές από 2 έως και 20. Ένας τρόπος αντιμετώπισης αυτού του προβλήματος είναι η κωδικοποίηση των μεταβλητών με binary coding και πιο συγκεκριμένα binary-Gray λόγω του πλεονεκτημάτος του στο mutation operator. Έχοντας επιλέξει αυτό τον τρόπο κωδικοποίησης, αρκεί να δώσουμε στο x_4 5 bits και να το αφήσουμε να πάρει τιμές από 0 έως και 31. Όσον αφορά τις υπόλοιπες μεταβλητές σχεδιασμού, λόγω του μικρού εύρους τιμών των x_1 και x_2 , δίνουμε σε εκάστοτε μεταβλητή

10 bits. Αντιθέτως για την x_3 , παρόλο που έχει μεγάλο εύρος τιμών, δεν απαιτείται τόσο μεγάλη ακρίβεια, κάτι που μας επιτρέπει να την εκφράσουμε με 12 bits. Τότε όμως, πρέπει να εισάγουμε δύο νέους περιορισμούς για την μεταβλητή x_4 (εξισώσεις (11) και (12) αντίστοιχα) που να μην της επιτρέπουν να πάει κάτω από 2 και πάνω από 20 όπως φαίνεται παρακάτω.

$$C_1 = x_1 - x_2 + 20 \leq 0 \leq 3 \quad (6)$$

$$C_2 = 2.5 * (x_4 + 1) - 30 \leq 0 \leq 3 \quad (7)$$

$$C_3 = \frac{x_3}{\pi * (x_2^2 - x_1^2)} - 0.4 \leq 0 \leq 3 \quad (8)$$

$$C_4 = \frac{2.22 * 10^{-3} * (x_2^3 - x_1^3) * x_3}{(x_2^2 - x_1^2)^2} - 1 \leq 0 \leq 3 \quad (9)$$

$$C_5 = 900 - \frac{2.66 * 10^{-2} * (x_2^3 - x_1^3) * x_3 * x_4}{(x_2^2 - x_1^2)} \leq 0 \leq 10 \quad (10)$$

$$C_6 = 2 - x_4 \leq 0 \leq 1 \quad (11)$$

$$C_7 = x_4 - 20 \leq 0 \leq 1 \quad (12)$$

Στις παραπάνω εξισώσεις, η πρώτη ανισότητα εκφράζει το nominal threshold ενώ η δεύτερη ανισότητα εκφράζει το relaxed threshold, πέραν του οποίου η υποψήφια λύση απορρίπτεται. Η εύρεση των relaxed threshold έγινε κάνοντας δοκιμές για $k_1 = 1.5$ και για prng seed = 1000. Βέβαια, λόγω της στοχαστικότητας του εξελικτικού αλγορίθμου, για κάποιο άλλο prng seed ή βάρος k θα μπορούσαν να προκύψουν διαφορετικά τα relaxed threshold. Εξαιτίας όμως των χρονικών περιορισμών, δεν είναι εφικτό να γίνουν τόσα πολλά τρεξίματα για κάθε μια από τις εννιά περιπτώσεις, οι οποίες φαίνονται στο πίνακα 1.1.

Περίπτωση	k	Prng seed
(1)	1.5	1000
(2)	1.5	2000
(3)	1.5	3000
(4)	5	1000
(5)	5	2000
(6)	5	3000
(7)	10	1000
(8)	10	2000
(9)	10	3000

Πίνακας 1.1: Οι διάφορες περιπτώσεις για τα τρεξίματα του EASY.

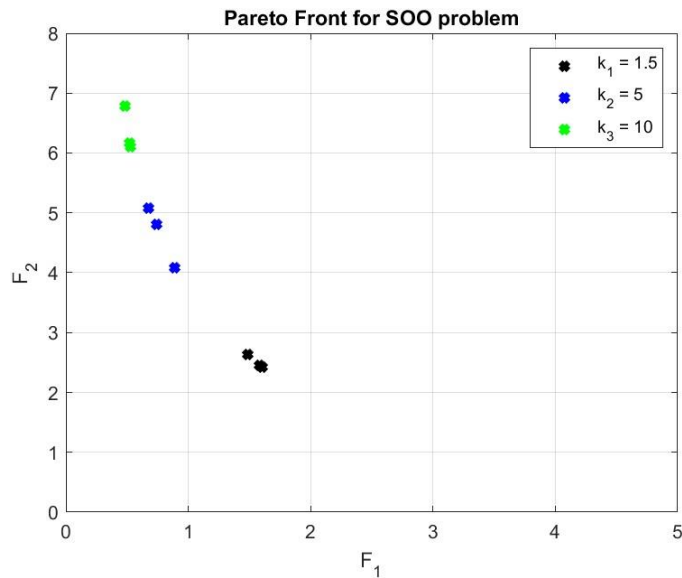
Πριν γίνει ο σχολιασμός των αποτελεσμάτων, πρέπει να γίνει αναφορά στα evolution operators καθώς και στις ρυθμίσεις που έχουν να κάνουν με το πληθυσμό. Οι περισσότερες δοκιμές έγιναν για $k_1 = 1.5$ και prng seed = 1000. Σε κάθε άλλη περίπτωση, έγιναν δύο με τρεις δοκιμές εποπτικά, οι οποίες οδήγησαν συνήθως σε χειρότερο αποτέλεσμα. Στις δοκιμές όπου το αποτέλεσμα βελτιώθηκε, η βελτίωση οφείλεται σε μικρή μείωση της πιθανότητας μετάλλαξης. Για τον πληθυσμό, μέσω διάφορων

δοκιμών, καταλήγουμε στο συμπέρασμα ότι η αύξηση του ελιτισμού και του πλήθους των γονέων (από είκοσι σε τριάντα) οδηγεί σε εύρεση μικρότερων τιμών της F. Στις περισσότερες περιπτώσεις, η τόσο μεγάλη αύξηση του ελιτισμού έχει σαν αποτέλεσμα τον «εγκλωβισμό» του εξελικτικού αλγορίθμου. Παρόλα αυτά, στην προκειμένη περίπτωση, ο αλγόριθμος καταλήγει στην σχεδόν στην ίδια τιμή της F για μεγάλο εύρος τιμών της πιθανότητας επιλογής ελίτ γονέα. Για αυτό, επιλέγουμε να κάνουμε τον αλγόριθμο πιο «εκμεταλλευτικό» με στόχο την εύρεση του ελαχίστου σε αυτή την περιοχή. Επίσης, στην φάση της επιλογής των γονέων, η αύξηση των απογόνων που συμμετέχουν στο τουρνουά από δύο σε τρεις και μικρή μείωση της πιθανότητας να επιλεχτεί ο καλύτερος απόγονος, οδηγούν σε βελτίωση της ελάχιστης τιμής. Στο crossover scheme, αποδείχτηκε ότι ανεξάρτητα από την πιθανότητα να γίνει crossover καθώς και οποιαδήποτε άλλη ρύθμιση στο mutation scheme, η καλύτερη μέθοδος είναι το two point crossover. Όσον αφορά το mutation scheme, η αλλαγή της πιθανότητας να γίνει μετάλλαξη καθώς και το πόσο θα άλλαζε η τιμή της μεταβλητής λόγω της μετάλλαξης είχε, τις περισσότερες φορές, αρνητική επίδραση στην λύση. Στο πίνακα 1.2 παρουσιάζονται συνοπτικά όλες οι αλλαγές στις παραμέτρους για την SOO βελτιστοποίηση.

Παράμετρος	Αρχική τιμή	Τελική τιμή
Πληθυσμός γονέων	15	30
Ελίτ γονείς ως απόγονοι	1	2
Πιθανότητα επιλογής ελίτ γονέα	0.15	0.5
Μέγεθος τουρνουά	2	3
Πιθανότητα τουρνουά	0.9	0.8
Μέθοδος crossover	Two point/var	Two point
Μετάλλαξη ανά στάσιμες γενιές	10	8

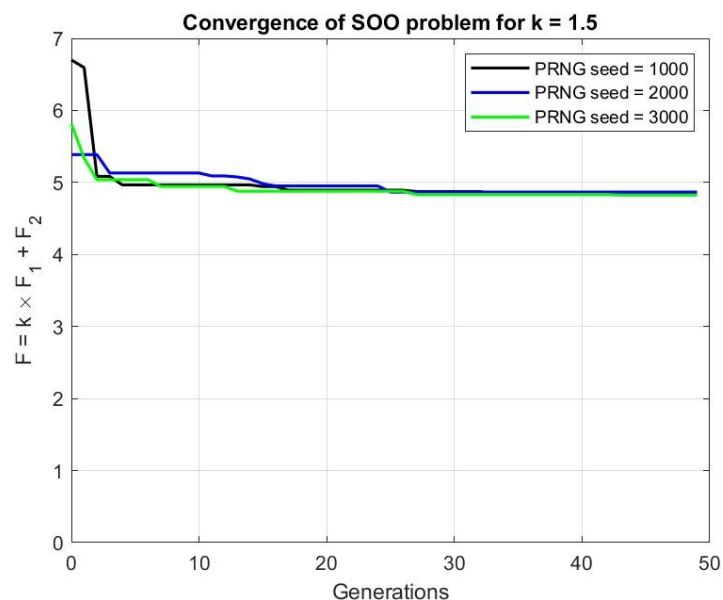
Πίνακας 1.2: Σύνοψη όλων των αλλαγών στις ρυθμίσεις του EASY.

Έχοντας κάνει πλέον την απαραίτητη ανάλυση των παραμέτρων του αλγορίθμου EASY, μπορούμε να προχωρήσουμε στην παράθεση των διαγραμμάτων και στον σχολιασμό των αποτελεσμάτων. Στο διάγραμμα 1.3 παρουσιάζεται το μέτωπο pareto που προέκυψε από τα εννέα τελικά τρεξίματα για τις τιμές του βάρους k και του rng seed που αναγράφονται στο πίνακα 1.1. Παρατηρούμε ότι όσο αυξάνεται ο συντελεστής βάρους k, τόσο μειώνονται οι τιμές που παίρνει η F₁ αλλά αυξάνονται οι τιμές της F₂. Αυτή η συμπεριφορά συμφωνεί πλήρως με αυτό που θεωρητικά περιμέναμε να συμβεί. Το γεγονός ότι για κάθε βάρος k προκύπτουν διαφορετικές λύσεις με διαφορετικές αποστάσεις μεταξύ τους (λχ. για k = 1.5 οι λύσεις είναι «μαζεμένες» ενώ για k = 5 οι λύσεις έχουν «ανοίξει») ανάλογα το rng seed, αναδεικνύει την στοχαστικότητα των εξελικτικών αλγορίθμων.



Διάγραμμα 1.3: Μέτωπο pareto που προκύπτει από την επίλυση του SOO προβλήματος.

Στο διάγραμμα 1.4 παρουσιάζεται η σύγκλιση του αλγορίθμου για τρία διαφορετικά rng seeds και για ένα βάρος ($k = 1.5$). Παρατηρούμε ότι, παρόλο που οι αρχικές τιμές για κάθε τρέξιμο είναι αρκετά διαφορετικές μεταξύ τους (λόγω της ψευδοτυχαίας αρχικοποίησης), πολύ γρήγορα σχεδόν στην ίδια τιμή. Επιπλέον, με την πάροδο των γενεών, βλέπουμε ότι υπάρχουν όλο και περισσότερες στάσιμες γενιές. Αυτό σημαίνει ότι είμαστε κοντά σε κάποιο ελάχιστο χωρίς όμως να γνωρίζουμε αν αυτό το ελάχιστο είναι ολικό ή τοπικό.



Διάγραμμα 1.4: Σύγκλιση αλγορίθμου για συντελεστή βάρους $k = 1.5$.

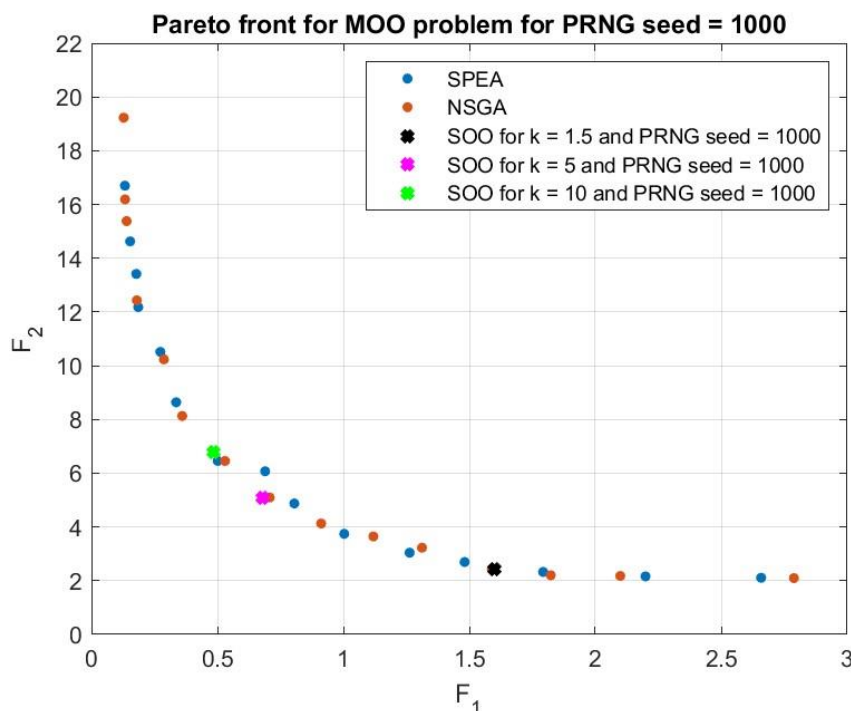
F	F_1	F_2	C_1	C_2	C_3	C_4	C_5	C_6	C_7
4.8308	1.6013	2.4289	-0.18235	-2.5	-0.13844	-0.75282	1.0664e+05	-8	-10

Πίνακας 1.5: Δεδομένα καλύτερου τρεξίματος SOO.

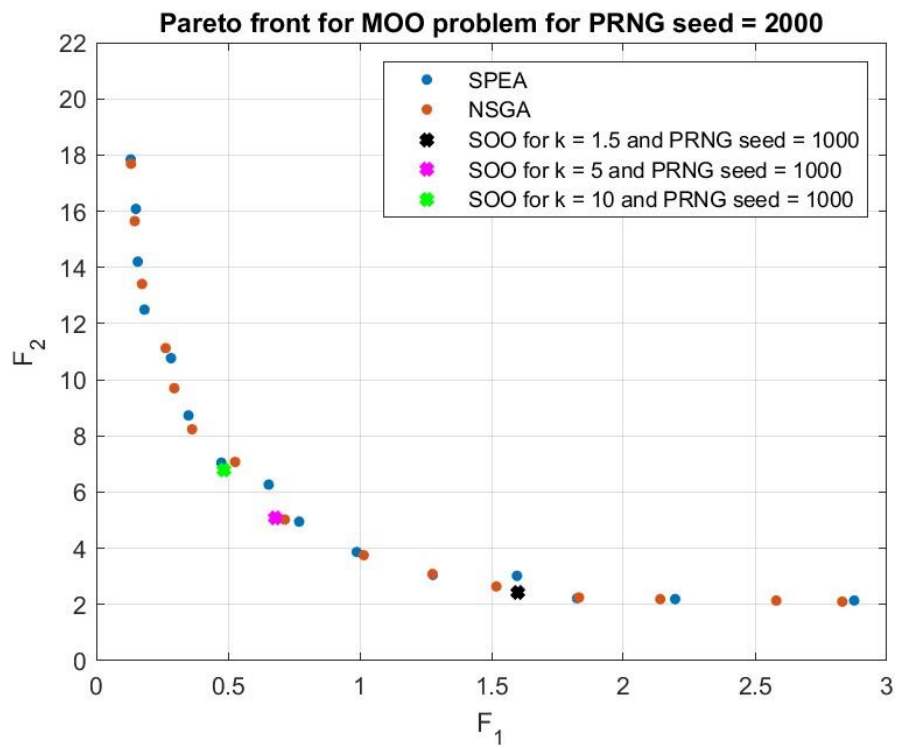
2 Επίλυση προβλήματος βελτιστοποίησης ως βελτιστοποίηση δύο συναρτήσεων (MOO)

Σε αυτό το ερώτημα μας ζητείται η βελτιστοποίηση των δύο συναρτήσεων χωρίς όμως να συνδέονται μέσω κάποιου συντελεστή βαρύτητας. Σε αυτή την περίπτωση, πρέπει να γίνει χρήση ειδικών αλγορίθμων όπως είναι ο NSGA, ο SPEA 1,1.5 και 2, οι οποίοι χρησιμοποιούν διαφορετικό τρόπο αξιολόγησης από αυτό που είχαμε για το πρόβλημα SOO. Πιο συγκεκριμένα, ο αλγόριθμος NSGA χωρίζει τις υποψήφιες λύσεις σε πολλά μέτωπα pareto και τους αναθέτει μια τιμή ανάλογα με τον αριθμό των υποψήφιων λύσεων που απέχουν λιγότερο από μια κρίσιμη απόσταση (σ_{share}). Αυξάνοντας την τιμή της κρίσιμης απόστασης οδηγεί σε ένα πιο «γεμάτο» μέτωπο pareto αλλά με χειρότερες τιμές για τις συναρτήσεις. Μετά από διάφορες δοκιμές, επιλέχθηκε $\sigma_{share} = 0.1$ καθώς για μεγαλύτερες τιμές όχι μόνο δίνει ένα πιο «μαζεμένο» μέτωπο αλλά οι συναρτήσεις έχουν χειρότερες τιμές. Σε αντίθεση με αυτό, όλες οι εκδοχές του αλγορίθμου SPEA αναθέτουν μια τιμή σε κάθε άτομο ανάλογα με τα πόσα άλλα άτομα «κυριαρχεί». Από τις τρεις εκδοχές του αλγορίθμου, επιλέχθηκε ο SPEA 2 καθώς προωθεί την δημιουργία ενός πιο «γεμάτου» μετώπου pareto χωρίς όμως να καθιστά χειρότερες τις τιμές των συναρτήσεων. Τα evolution schemes καθώς και το population scheme παραμένουν τα ίδια με αυτά που αναγράφονται στο πίνακα 1.2 εφόσον μετά από μερικές δοκιμές, αποδείχτηκε ότι αυτές οι ρυθμίσεις δίνουν το καλύτερο αποτέλεσμα.

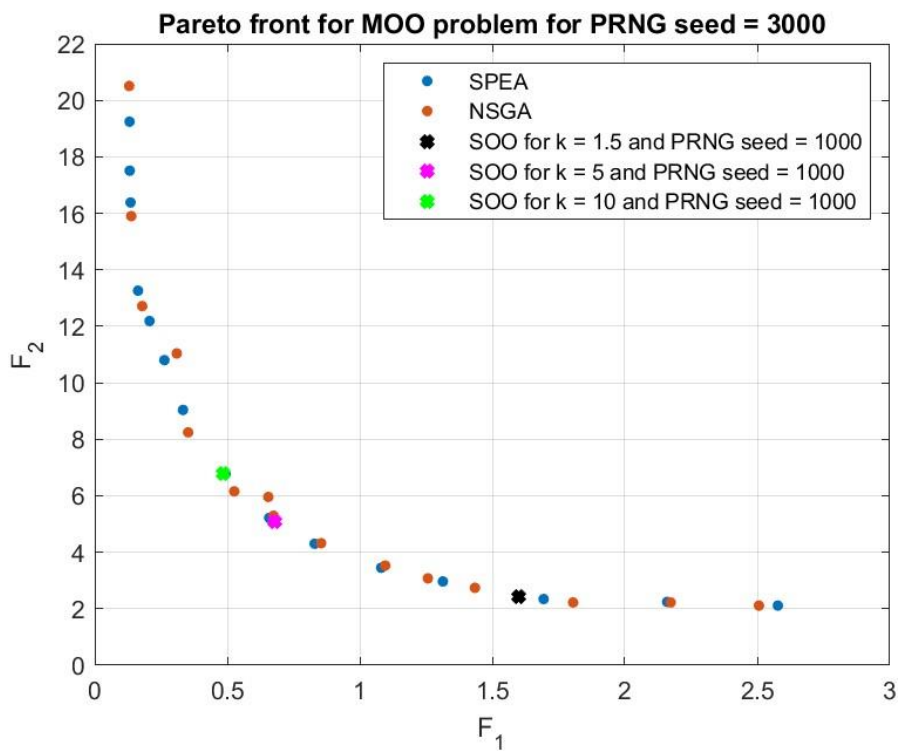
Έχοντας πλέον αναφερθεί στις ρυθμίσεις που έγιναν για την επίλυση του MOO προβλήματος, είμαστε σε θέση να παραθέσουμε τα τρία μέτωπα pareto (διάγραμμα 2.1, 2.2 και 2.3 αντίστοιχα) που προέκυψαν για τα γνωστά prng seeds.



Διάγραμμα 2.1: Μέτωπο pareto για prng seed = 1000.



Διάγραμμα 2.2: Μέτωπο pareto για prng seed = 2000.



Διάγραμμα 2.3: Μέτωπο pareto για prng seed = 3000.

Συγκρίνοντας τα παραπάνω διαγράμματα, παρατηρούμε ότι παρόλο που η μορφή του μετώπου παραμένει ίδια και στις τρεις περιπτώσεις, η μέγιστη τιμή που παίρνει η κάθε συνάρτηση αλλάζει δραστικά ανάλογα με το rng seed. Επίσης, στα διαγράμματα έχουν προστεθεί και τρία σημεία που προέκυψαν από την SOO βελτιστοποίηση, ένα για κάθε συντελεστή βάρους με rng seed = 1000. Και στα τρία τρεξίματα, είναι εμφανές ότι οι λύσεις που προέκυψαν από την SOO βελτιστοποίηση ανήκουν στα μέτωπα pareto που υπολογίστηκαν κατά την επίλυση του MOO προβλήματος και σε ορισμένες περιπτώσεις, μπορεί να είναι καλύτερα από αυτά όπως φαίνεται στο διάγραμμα 2.2 για το πράσινο σημείο. Παρόλα αυτά όμως, για την δημιουργία ενός ικανοποιητικού μετώπου pareto από βελτιστοποίηση SOO, απαιτούνται πολλά τρεξίματα του εξελικτικού αλγορίθμου για διάφορες τιμές του συντελεστή βάρους k και του rng seed, το οποίο έχει πολύ υψηλό υπολογιστικό κόστος. Τέλος, ανάλογα με το rng seed, παρατηρούμε διαφορετικά αποτελέσματα σχετικά με το πόσο «γεμάτο» είναι το μέτωπο pareto της κάθε μεθόδου. Συμπερασματικά, κανένας δεν μπορεί να ισχυριστεί ότι η μια μέθοδος αξιολόγησης είναι καλύτερη από την άλλη καθώς επηρεάζονται σημαντικά από τις ρυθμίσεις του εξελικτικού αλγορίθμου.

3 Επίλυση προβλήματος βελτιστοποίησης ως βελτιστοποίηση Nash

Στο τρίτο ερώτημα μας ζητείτε η επίλυση του προβλήματος βελτιστοποίησης χρησιμοποιώντας την μέθοδο Nash. Σύμφωνα με αυτή, μετατρέπουμε την ταυτόχρονη βελτιστοποίηση των δύο συναρτήσεων σε δυο ξεχωριστές βελτιστοποιήσεις, μία για κάθε συνάρτηση. Αυτό επιτυγχάνεται ορίζοντας δύο «παίκτες», Player 1 (P1) και Player 2 (P2), στους οποίους αναθέτουμε την βελτιστοποίηση μίας εκ των δύο συναρτήσεων. Ο κάθε παίκτης μπορεί να ρεγουλάρει ορισμένες μεταβλητές σχεδιασμού, θεωρώντας τις άλλες σταθερές. Με ακριβώς παρόμοιο τρόπο, ο κάθε παίκτης δεν χρειάζεται να ικανοποιεί όλους τους περιορισμούς αλλά μόνο αυτούς που του έχουν ανατεθεί. Μετά από κάποιες γενιές των δύο παικτών, ανταλλάσσουν μεταξύ τους τις τιμές των μεταβλητών σχεδιασμού για την καλύτερη λύση που έχουν βρει και επαναλαμβάνουν την διαδικασία.

Στο συγκεκριμένο πρόβλημα, αρχικά ορίσαμε ότι ο P1 θα βελτιστοποιήσει την F_1 και ο P2 την F_2 . Παρατηρώντας την συνάρτηση της F_1 (1), μια λογική σκέψη θα ήταν να δώσουμε στον P1 τις μεταβλητές x_1 και x_2 καθώς αυτές επηρεάζουν κατά κύριο λόγο την τιμή της F_1 . Επομένως, ο P2 θα καθορίσει τις x_3 και x_4 . Όσον αφορά τους περιορισμούς, ο C_1 (εξίσωση (6)) πάει στο P1 καθώς εξαρτάται μόνο από τα x_1 και x_2 . Ομοίως, οι περιορισμοί C_2 , C_6 και C_7 (εξισώσεις (7), (11) και (12) αντίστοιχα) λογίζονται στο P2. Οι τρεις εναπομείναντες περιορισμοί θα μπορούσαν να πάνε σε οποιοδήποτε από τους δύο παίκτες υπό την προϋπόθεση ότι τουλάχιστον ένας από αυτούς θα πάει στον P1 ειδάλλως ο P1 απλώς θα μεγιστοποιεί την διαφορά μεταξύ x_1 και x_2 . Επιπλέον, οι δύο παίκτες ανταλλάσσουν πληροφορίες μεταξύ τους ανά τριάντα γενιές. Οι ρυθμίσεις του EASY για κάθε παίκτη είναι οι ίδιες με αυτές που χρησιμοποιήθηκαν στο κεφάλαιο 1 για `rng seed = 1000`. Στο πίνακα 3.1 φαίνονται συνοπτικά τα δεδομένα για κάθε παίκτη.

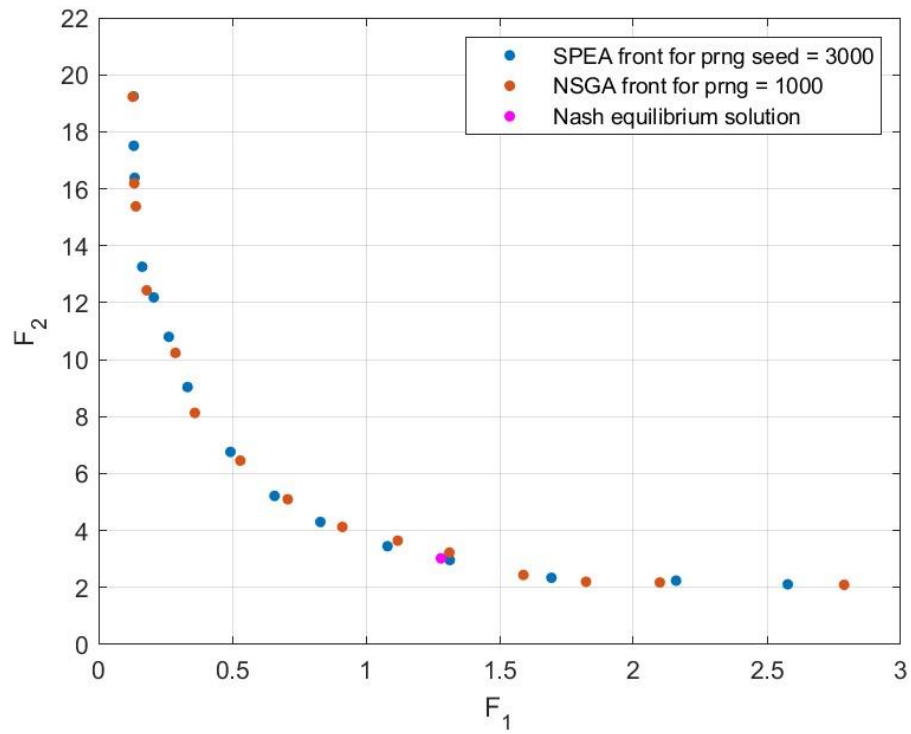
Player 1	Player 2
F_1	F_2
x_1 και x_2	x_3 και x_4
C_1 και C_4	C_2, C_3, C_5, C_6 και C_7

Πίνακας 3.1: Δεδομένα παικτών για βελτιστοποίηση Nash.

Υλοποιώντας τη διαδικασία που περιεγράφηκε παραπάνω, παρατηρούμε ότι, ανεξάρτητα από το πώς θα κατανομηθούν οι περιορισμοί C_3 , C_4 και C_5 , καταλήγουμε στην ίδια λύση μέσα σε δύο έως τρεις ανταλλαγές. Η λύση αυτή παρουσιάζεται στον πίνακα 3.2. Παρουσιάζοντας τη λύση Nash μαζί με δύο από τα καλύτερα μέτωπα pareto στο διάγραμμα 3.3, διαπιστώνουμε ότι η λύση Nash όχι μόνο ανήκει σε ένα από τα μέτωπα, αλλά κυριαρχεί και μιας άλλης λύσης, επιβεβαιώνοντας ότι η μέθοδος Nash βρήκε μια βέλτιστη λύση.

F_1	F_2	x_1	x_2	x_3	x_4
1.2805	3.0221	55.0977	75.1026	3000	11

Πίνακας 3.2: Τιμές συναρτήσεων και μεταβλητών σχεδιασμού για την λύση ισορροπίας Nash.



Διάγραμμα 3.3: Επίδειξη λύσης ισορροπίας Nash σε σχέση με δύο μέτωπα pareto από το MOO πρόβλημα.

4 Επίλυση προβλήματος βελτιστοποίησης ως βελτιστοποίηση SOO με χρήση μεταμοντέλου (ΜΑΕΑ)

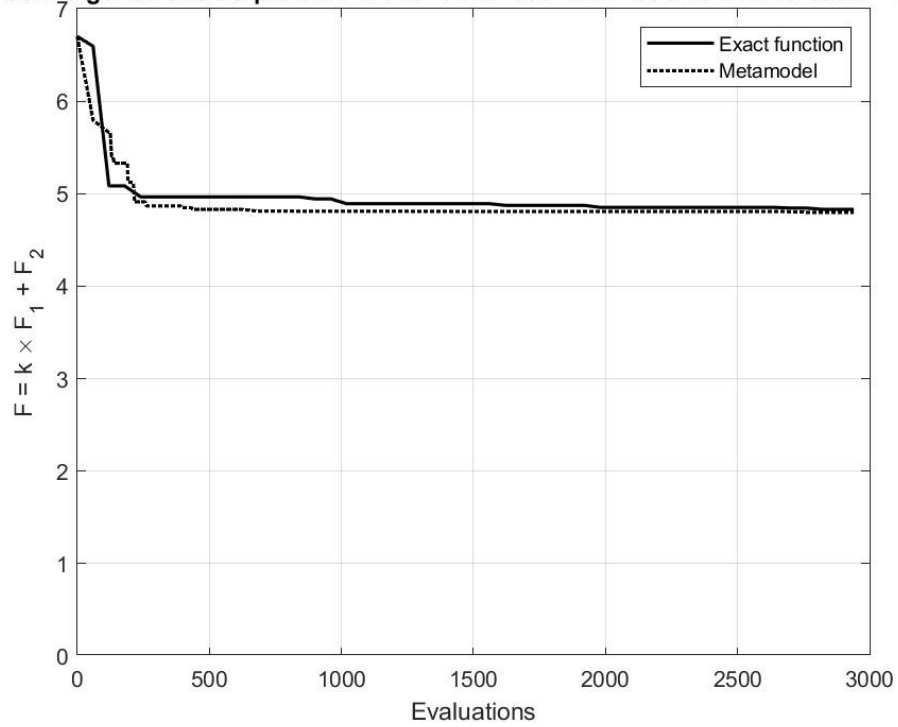
Στο τελευταίο ερώτημα μας ζητείται η βελτιστοποίηση SOO για ένα συντελεστή βάρους k , ο οποίος επιλέγεται ίσος με 1.5, κάνοντας χρήση μεταμοντέλου. Ως μεταμοντέλο ορίζεται μια συνάρτηση η οποία χρησιμοποιείται στη θέση της αντικειμενικής συνάρτησης με στόχο την μείωση του υπολογιστικού κόστους. Αρχικά, η τιμή του κάθε ατόμου υπολογίζεται αποκλειστικά από την αντικειμενική συνάρτηση καθώς δεν υπάρχουν ακόμα αρκετά δεδομένα ώστε να «εκπαιδευθεί» το μεταμοντέλο. Από την στιγμή που είναι έτοιμο το μεταμοντέλο, όλα τα άτομα αξιολογούνται αρχικά από αυτό και τα καλύτερα αξιολογούνται και από την ακριβή αλλά ακριβή αντικειμενική συνάρτηση. Έπειτα, όσα άτομα υπολογίζονται με ακρίβεια, συμβάλλουν στην βελτίωση του μεταμοντέλου. Με αυτό τον τρόπο, μειώνονται δραστικά οι κλήσεις της αντικειμενικής συνάρτησης.

Στο συγκεκριμένο πρόβλημα γίνεται χρήση μεταμοντέλου RBF του οποίου οι ρυθμίσεις παραμένουν ίδιες με αυτές που έχει αρχικά ο EASY καθώς οποιαδήποτε μεταβολή τους δεν οδηγεί σε βελτίωση του αποτελέσματος. Έχοντας πλέον επιλέξει τις ρυθμίσεις για το μεταμοντέλο, οφείλουμε να κάνουμε δοκιμές ώστε να ελέγξουμε αν κάποια αλλαγή στο population scheme ή στα evolution operators οδηγεί σε καλύτερη λύση. Αυτές οι δοκιμές γίνονται για rng seed = 1000. Οι ρυθμίσεις για τις οποίες έχουμε την καλύτερη λύση αναγράφονται στο πίνακα 4.1.

Παράμετρος	Αρχική τιμή	Τελική τιμή
Πληθυσμός γονέων	15	30
Ελίτ γονείς ως απόγονοι	1	2
Πιθανότητα επιλογής ελίτ γονέα	0.15	0.5
Μέγεθος τουρνουά	2	3
Πιθανότητα τουρνουά	0.9	0.8
Πιθανότητα μετάλλαξης	0.2	0.15
Μετάλλαξη ανά στάσιμες γενιές	10	8
Αριθμός γονέων ανά παιδί	2	3

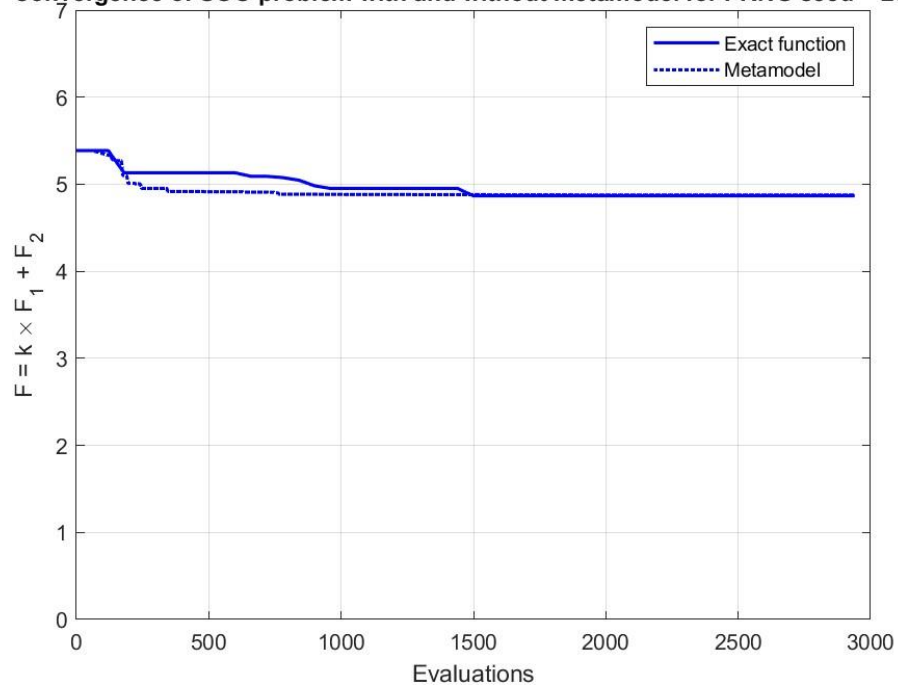
Πίνακας 4.1: Σύνοψη όλων των αλλαγών στις ρυθμίσεις του EASY για το μεταμοντέλο.

Convergence of SOO problem with and without metamodel for PRNG seed = 1000



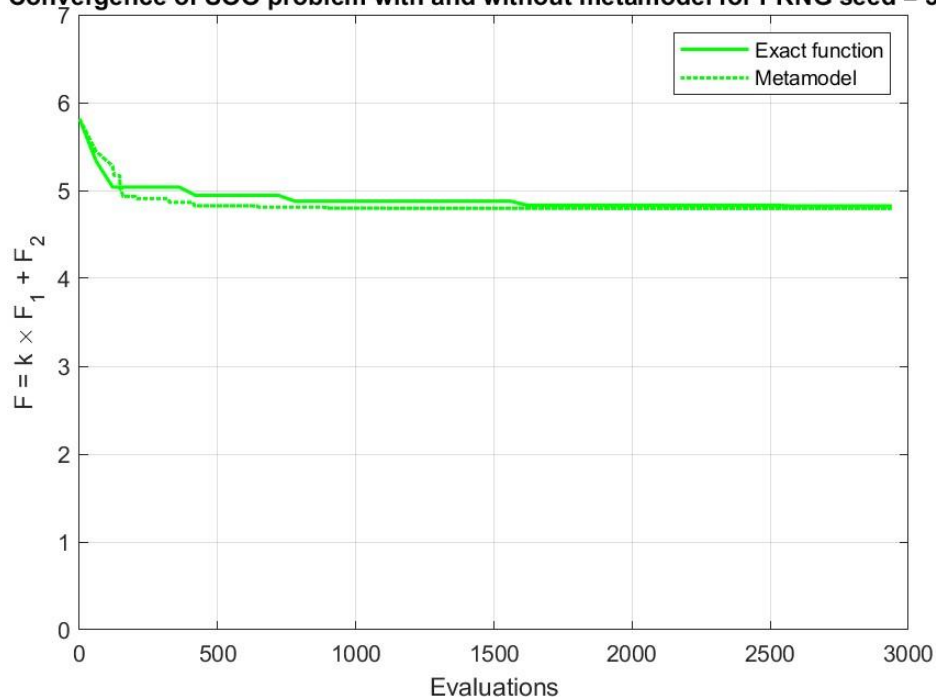
Διάγραμμα 4.2: Σύγκριση σύγκλιση προβλήματος SOO με μεταμοντέλο για prng seed = 1000.

Convergence of SOO problem with and without metamodel for PRNG seed = 2000



Διάγραμμα 4.3: Σύγκριση σύγκλιση προβλήματος SOO με μεταμοντέλο για prng seed = 2000.

Convergence of SOO problem with and without metamodel for PRNG seed = 3000



Διάγραμμα 4.4: Σύγκριση σύγκλιση προβλήματος SOO με μεταμοντέλο για prng seed = 3000.

Από τα διαγράμματα 4.2, 4.3 και 4.4 διαπιστώνεται το γεγονός ότι με την χρήση μεταμοντέλου μειώνεται δραματικά το υπολογιστικό κόστος καθώς καταλήγει στην ίδια λύση σε κάθε περίπτωση με λιγότερες από πεντακόσιες κλήσεις της ακριβούς συνάρτησης. Αυτή η οικονομία καθιστά πλέον αναγκαία την χρήση μεταμοντέλων στην πλειοψηφία των βελτιστοποιήσεων. Τέλος, παρατηρούμε ότι ακόμα και με την χρήση μεταμοντέλου, η βέλτιστη τιμή της συνάρτησης F δεν μειώνεται περαιτέρω. Αυτό είναι μια ένδειξη ότι είναι πιθανό να είμαστε στο ολικό βέλτιστο.