

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ ΚΑΙ ΑΣΦΑΛΙΣΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΧΑΡΤΟΦΥΛΑΚΙΟΥ ΜΕΣΩ ΕΥΡΕΤΙΚΩΝ ΤΕΧΝΙΚΩΝ ΠΡΟΣΟΜΟΙΩΣΗΣ

Αντώνιος Σ. Πεσκώστας

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφα-
λιστικής Επιστήμης του Πανεπιστημίου Πειραιώς
ως μέρος των απαιτήσεων για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης στην Ε-
φαρμοσμένη Στατιστική

Πειραιάς
Νοέμβριος 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ ΚΑΙ ΑΣΦΑΛΙΣΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΧΑΡΤΟΦΥΛΑΚΙΟΥ ΜΕΣΩ ΕΥΡΕΤΙΚΩΝ ΤΕΧΝΙΚΩΝ ΠΡΟΣΟΜΟΙΩΣΗΣ

Αντώνιος Σ. Πεσκώστας

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην Εφαρμοσμένη Στατιστική

Πειραιάς
Νοέμβριος 2019

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή που ορίστηκε από τη ΓΣΕΣ του Τμήματος Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς στην υπ' αριθμ. συνεδρίασή του σύμφωνα με τον Εσωτερικό Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών στην Εφαρμοσμένη Στατιστική

Τα μέλη της Επιτροπής ήταν:

- Αναπλ. Καθηγητής Μπούτσικας Μιχαήλ (Επιβλέπων)
- Επικ. Καθηγητής Πελέκης Νικόλαος
- Επικ. Καθηγητής Ανθρωπέλος Μιχαήλ

Η έγκριση της Διπλωματικής Εργασίας από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνώμων του συγγραφέα.

UNIVERSITY OF PIRAEUS



**DEPARTMENT OF STATISTICS
AND INSURANCE SCIENCE**

**POSTGRADUATE PROGRAM IN
APPLIED STATISTICS**

**PORTFOLIO OPTIMIZATION USING
HEURISTIC SIMULATION
TECHNIQUES**

By

Antonios S. Peskostas

MSc Dissertation

submitted to the Department of Statistics and Insurance Science of the University of Piraeus in partial fulfilment of the requirements for the degree of Master of Science in Applied Statistics

Piraeus, Greece
November 2019

*Στην σύζυγό μου,
στις κόρες μου,
και στους γονείς μου*

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Επίκουρο Καθηγητή κύριο Πελέκη Νικόλαο και τον Επίκουρο Καθηγητή κύριο Ανθρωπέλο Μιχαήλ που ήταν μέλη της επιτροπής αξιολόγησης της Διπλωματικής μου Εργασίας. Ιδιαίτερα ευχαριστώ τον επιβλέποντα της Διπλωματικής μου Εργασίας, Αναπληρωτή Καθηγητή κύριο Μπούτσικα Μιχαήλ για την εμπιστοσύνη που μου έδειξε, για την πολύτιμη βοήθεια και καθοδήγηση που μου προσέφερε καθώς και για την άψογη συνεργασία μας. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την ηθική συμπαράσταση και βοήθεια που μου προσέφερε καθ' όλη την διάρκεια των σπουδών μου.

Περίληψη

Στην εργασία αυτή αναλύονται τα βασικά στοιχεία που καθορίζουν την σύσταση ενός χαρτοφυλακίου. Επισημαίνεται ο σκοπός της σύνθεσης ενός χαρτοφυλακίου καθώς και η αξία της διαφοροποίησης. Εξετάζονται αναλυτικές μέθοδοι και μοντέλα τα οποία οδηγούν σε κλειστούς τύπους και προσδιορίζουν την βέλτιστη σύνθεση ενός χαρτοφυλακίου επενδύσεων, είτε ελαχιστοποιώντας ένα συγκεκριμένο μέτρο κινδύνου του χαρτοφυλακίου είτε μεγιστοποιώντας της αναμενόμενη ωφελιμότητά του. Στη σύγχρονη προσέγγιση του προβλήματος μέτρα όπως το Sharpe ratio συνδυάζουν και τα δύο. Παρουσιάζουμε επίσης τις βασικότερες από τις κλασσικές μεθόδους βελτιστοποίησης όπως ο τετραγωνικός προγραμματισμός, και αναλύουμε εκτενώς μία νέα σχετικά κατηγορία αλγορίθμων που ονομάζονται Μεταευρετικοί και οι οποίοι μπορούν και δίνουν πολύ καλές λύσεις σε πολύ καλό υπολογιστικό χρόνο επιδεικνύοντας εξαιρετική απόδοση και επίδοση.

Στην πρώτη ενότητα αναφερόμαστε αναλυτικά στα πιο βασικά μοντέλα που διέπουν την σύγχρονη θεωρία Χαρτοφυλακίου όπως το Markowitz Model, το Black Model και το Tobin Model. Όλα αυτά τα μοντέλα τα εξετάζουμε από την σκοπιά της βελτιστοποίησης με περιορισμούς, όπου με την βοήθεια μεθόδων όπως οι πολλαπλασιαστές του Lagrange και οι αναγκαίες συνθήκες των Karush-Kuhn-Tucker μπορούν και δίνουν σε αρκετές περιπτώσεις κλειστού τύπου λύσεις στην αναζήτηση του βέλτιστου Χαρτοφυλακίου. Σε εκείνες όμως τις περιπτώσεις όπου είτε ο αριθμός είτε το είδος των περιορισμών καθιστά την εύρεση αναλυτικής λύσης αδύνατη, παρουσιάζονται σε επόμενη ενότητα άλλες αριθμητικές μέθοδοι προσέγγισης της βέλτιστης λύσης.

Στη δεύτερη ενότητα γίνεται αναφορά στον τρόπο λειτουργίας και στα χαρακτηριστικά κλασσικών αλγορίθμων βελτιστοποίησης, αλλά και των πιο σύγχρονων μεταευρετικών αλγορίθμων (Metaheuristics) και πιο συγκεκριμένα αυτών της νοημοσύνης σμηνών (Particle Swarm Optimization, Ant Colony Optimization) και της τυχαίας αναζήτησης (Simulated annealing, Tabu search).

Στην τρίτη ενότητα αναλύεται εκτενώς μία ακόμα σημαντική κατηγορία των μεταευρετικών αλγορίθμων, η οποία είναι οι γενετικοί αλγόριθμοι (Genetic Algorithms) οι οποίοι αποτελούν και το βασικό αντικείμενο της παρούσας εργασίας. Με την δημιουργία κώδικα μέσω του προγραμματιστικού περιβάλλοντος της γλώσσας R, την εφαρμογή τους σε διάφορα παραδείγματα και την παρουσίαση πινάκων και γραφημάτων αναλύονται οι μηχανισμοί και η λειτουργία τους.

Τέλος στην τέταρτη ενότητα γίνεται εφαρμογή των γενετικών αλγορίθμων σε ένα πρακτικό πρόβλημα (real case study) όπως αυτό της εύρεσης της βέλτιστης σύνθεσης ενός επενδυτικού χαρτοφυλακίου βασιζόμενοι σε πραγματικές αποδόσεις διαφόρων επενδυτικών τίτλων.

Abstract

In this MSc thesis we study the key elements that determine the composition of a portfolio and highlight the advantages of diversification. Employing analytical methods and models that offer closed-form solutions we determine the optimal composition of an investment portfolio, either by minimizing a specific portfolio risk or by maximizing its expected utility. In a modern approach of the problem, Sharpe indicators combine both of these approaches. We also present some classical optimization methods such as quadratic programming, and we extensively analyze a new class of metaheuristic algorithms, which can deliver very good solutions in relatively fast computational time, offering excellent performance.

In the first section we discuss in detail the most popular models of contemporary portfolio theory such as the Markowitz Model, the Black Model and the Tobin Model. Our review stems from the perspective of constrained optimization, and with the assistance of methods such as Lagrange multipliers and the Karush-Kuhn-Tucker conditions, we are able to derive closed-form solutions for the determination of the optimal composition of a financial portfolio. However, in cases where either the number or the type of constraints makes it impossible to find an analytical solution, we present in the next section appropriate numerical methods for approximating the optimal solution.

In the second section we review the mechanisms of well-known optimization algorithms, as well as some recently proposed metaheuristic algorithms and more specifically, Particle Swarm Optimization, Ant Colony Optimization and Random Search (Simulated annealing, Tabu search).

In the third section, which consists of the main subject of our thesis, we present in detail an important class of metaheuristic algorithms, the so-called Genetic Algorithms. By employing the R programming language environment, their mechanisms and performance with respect to their parameters are numerically studied through specific examples, by presenting illustrating tables and graphs.

Finally, in the fourth section, the genetic algorithms are applied to a specific case study related to the determination of the optimal composition of an investment portfolio based on historical values of returns on various investment securities.

Περιεχόμενα

Περίληψη	xi
Abstract	xiii
Κατάλογος Πινάκων	18
Κατάλογος Σχημάτων	21
1. Σύνθεση και βελτιστοποίηση Χαρτοφυλακίου	24
1.1 Εισαγωγή	24
1.2 Απόδοση και Κίνδυνος	25
1.3 Η Εκτίμηση της Απόδοσης και του Κινδύνου	26
1.3.1 Μέτρα Μεταβλητότητας	27
1.3.2 Κίνδυνος καθοδικής απόκλισης	29
1.3.3 Δείκτες ευαισθησίας (sensitivities)	30
1.4 Σύνθεση Χαρτοφυλακίου και Διαφοροποίηση	31
1.5 Βελτιστοποίηση χαρτοφυλακίου	33
1.5.1 The Black Model	34
1.5.2 The Markowitz Model	39
1.5.3 The Tobin Model	40
2. Βασικές έννοιες Βελτιστοποίησης και Αλγόριθμοι	45
2.1 Εισαγωγή στην Βελτιστοποίηση	45
2.2 Τοπικά και Ολικά Βέλτιστα	47
2.3 Κλασσικές τεχνικές Βελτιστοποίησης	48
2.4 Μεταερευρητικοί Αλγόριθμοι Αναζήτησης	49
2.5 Νοημοσύνη Σμηνών	50
2.5.1 Αλγόριθμος αποικίας Μυρμηγκιών	50
2.5.2 Βελτιστοποίηση με Σμήνος Σωματιδίων	56
2.5.3 Λειτουργία αλγόριθμου Σμήνους Σωματιδίων	57
2.5.4 Επίδραση των παραμέτρων του αλγόριθμου Σμήνους Σωματιδίων	58
2.6 Αλγόριθμος Περιορισμένης Αναζήτησης Tabu	58
2.6.1 Λειτουργία αλγορίθμου αναζήτησης Tabu	59
2.7 Αλγόριθμος Προσομοιωμένης Ανόπτησης (SANN)	60

2.7.1	Λειτουργία αλγορίθμου Προσομοιωμένης Ανόπτησης (SANN)	61
3.	Γενετικοί Αλγόριθμοι (GA)	63
3.1	Γενετικοί Αλγόριθμοι	63
3.2	Λειτουργία Γενετικού Αλγόριθμου (GA)	68
3.3	Λειτουργία Γενετικού Αλγόριθμου (GA) μέσω κώδικα	69
3.3.1	Γενετικός Αλγόριθμος με δυαδική (binary) κωδικοποίηση	70
3.3.2	Γενετικός Αλγόριθμος με αριθμητική (arithmetic) κωδικοποίηση	73
3.4	Παραδείγματα εφαρμογής Γενετικού Αλγόριθμου (GA)	77
3.4.1	Παράδειγμα 1 – Μονοδιάστατη μεταβλητή	77
3.4.2	Εκτέλεση κώδικα Γενετικού Αλγόριθμου (GA) μονοδιάστατης μεταβλητής	77
3.4.3	Παράδειγμα 2 – Δισδιάστατη μεταβλητή	88
3.4.4	Εκτέλεση κώδικα Γενετικού Αλγόριθμου (GA) δισδιάστατης μεταβλητής	89
4.	Βελτιστοποίηση Χαρτοφυλακίου με Γενετικούς Αλγόριθμους	99
4.1	Σύντομη Αναφορά στο μοντέλο	99
4.1.1	Αντικειμενική συνάρτηση και περιορισμοί	100
4.2	Μελέτη περίπτωσης με ιστορικές αποδόσεις πραγματικών χρηματιστηριακών τίτλων	102
4.3	Εφαρμογή Γενετικού αλγορίθμου	111
4.4	Σύνθεση Βέλτιστου Χαρτοφυλακίου	119
	Παραρτήματα	124
	Βιβλιογραφία	143

Κατάλογος Πινάκων

3.1	Διασταύρωση ομοιόμορφη, ενός και πολλαπλών (δύο) σημείων	66
3.2	Διαδικασία ομοιόμορφης μετάλλαξης	67
3.3	Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_c	78
3.4	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου $5 * 10^2$ φορές, για διάφορες τιμές της πιθανότητας διασταύρωσης p_c	80
3.5	Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_m	82
3.6	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου $5 * 10^2$ φορές, για διάφορες τιμές της πιθανότητας μετάλλαξης p_m	83
3.7	Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της E_r	85
3.8	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου $5 * 10^2$ φορές, για διάφορες τιμές του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r	86
3.9	Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_c	90
3.10	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου 10^2 φορές, για διάφορες τιμές της πιθανότητας διασταύρωσης p_c	91
3.11	Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_m	93
3.12	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου 10^2 φορές, για διάφορες τιμές της πιθανότητας μετάλλαξης p_m	94
3.13	Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της E_r	95
3.14	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου 10^2 φορές, για διάφορες τιμές του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r	97
4.1	Ιστορικές ημερήσιες τιμές μετοχών	103

4.2	Ημερήσιες αποδόσεις μετοχών	104
4.3	Περιγραφικά μέτρα αποδόσεων μετοχών	105
4.4	Πίνακας συσχετίσεων μετοχών	106
4.5	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_c	112
4.6	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_m	114
4.7	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της E_r	115
4.8	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διαφορετικές μεθόδους επιλογής	116
4.9	Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διαφορετικές μεθόδους διασταύρωσης	117
4.10	Συγκριτικός πίνακας απόδοσης και κινδύνου βέλτιστου χαρτοφυλακίου και μετοχών	119

Κατάλογος Σχημάτων

1.1	Value at Risk	30
1.2	Αποτελεσματικό σύνορο του Markowitz	34
1.3	Ακίνδυνο χρεόγραφο και αποτελεσματικό σύνορο του του Markowitz	41
2.1	Γράφημα συνάρτησης $f(x) = (x^2 + x) \cos(x), x \in [-10, +10]$	46
2.2	Απεικόνιση των σχηματισμών των τερμιτών	50
2.3	Σχεδιάγραμμα μήκους των μονοπατιών	51
2.4	Σχεδιάγραμμα με την υπολογισμένη συνολική φερομόνη χωρίς εξάτμιση	52
2.5	Σχεδιάγραμμα με υπολογισμένη την συνολική φερομόνη με εξάτμιση	53
2.6	Διάγραμμα κόστους, φερομόνης και πιθανών διαδρομών	54
3.1	Διάγραμμα ροής γενετικού αλγορίθμου	69
3.2	Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $p_{c1} = 0, p_{c2} = 0.5, p_{c3} = 1.0$	79
3.3	Διαγράμματα διασποράς για διάφορες τιμές της πιθανότητας διασταύρωσης p_c , με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης και διάγραμμα με καμπύλη παλινδρόμησης και με ευθεία παλινδρόμησης σε μετασχηματισμένα δεδομένα.	80
3.4	Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $p_{m1} = 0, p_{m2} = 0.1, p_{m3} = 1.0$	82
3.5	Διαγράμματα διασποράς για διάφορες τιμές της πιθανότητας μετάλλαξης p_m , με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης και διάγραμμα με καμπύλη παλινδρόμησης και με ευθεία παλινδρόμησης σε μετασχηματισμένα δεδομένα.	84
3.6	Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $E_{r1} = 0, E_{r2} = 0.5, E_{r3} = 1.0$.	85
3.7	Διαγράμματα διασποράς για διάφορες τιμές του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης, και διάγραμμα με καμπύλη παλινδρόμησης.	87
3.8	Ιστόγραμμα και διαγράμματα με τις εκτιμήσεις των λύσεων όπως προκύπτουν από $5 * 10^2$ επαναλήψεις του αλγορίθμου για $p_c = 0.86, p_m = 0.10, E_r = 0.05$.	87
3.9	Η τρισδιάστατη αναπαράσταση της $f(x)$, όπου διακρίνονται δύο κορυφές.	88
3.10	Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $p_{c1} = 0, p_{c2} = 0.5, p_{c3} = 1.0$	90

3.11	Διαγράμματα διασποράς για διάφορες τιμές της πιθανότητας διασταύρωσης p_c , με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης και καμπύλες παλινδρόμησης.	92
3.12	Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $p_{m1} = 0, p_{m2} = 0.1, p_{m3} = 1.0$	93
3.13	Διαγράμματα διασποράς για διάφορες τιμές της πιθανότητας μετάλλαξης p_m , με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης, και καμπύλες παλινδρόμησης.	94
3.14	Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $E_{r1} = 0, E_{r2} = 0.5, E_{r3} = 1.0$	96
3.15	Διαγράμματα διασποράς για διάφορες τιμές του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης, και διάγραμμα με καμπύλη παλινδρόμησης	97
3.16	Ιστογράμματα και διαγράμματα ισοϋψών καμπυλών με τις εκτιμήσεις των λύσεων όπως προκύπτουν από 10^2 επαναλήψεις του αλγορίθμου για $p_c = 0.86, p_m = 0.10, E_r = 0.05$	98
4.1	Χρονοσειρά ημερήσιας κίνησης μετοχών και δείκτη	103
4.2	Χρονοσειρά ημερησίων αποδόσεων μετοχών και δείκτη	105
4.3	MSFT Ιστόγραμμα και QQ-Plot	106
4.4	AMZN Ιστόγραμμα και QQ-Plot	107
4.5	AAPL Ιστόγραμμα και QQ-Plot	107
4.6	FB Ιστόγραμμα και QQ-Plot	108
4.7	NFLX Ιστόγραμμα και QQ-Plot	108
4.8	^GSPC Ιστόγραμμα και QQ-Plot	109
4.9	Παρουσίαση αποτελεσμάτων γενετικού αλγόριθμου	118
4.10	Σύγκλιση Γενετικού Αλγορίθμου	118
4.11	Βέλτιστο χαρτοφυλάκιο Χρονοσειρά αποδόσεων, Ιστόγραμμα και QQ-Plot	120
4.12	Ιστορικές αποδόσεις βέλτιστου χαρτοφυλακίου και μεμονωμένων μετοχών	122

Σύνθεση και βελτιστοποίηση Χαρτοφυλακίου

Στο κεφάλαιο αυτό αναλύονται οι μέθοδοι εκτίμησης της απόδοσης και του κινδύνου σε ένα επενδυτικό Χαρτοφυλάκιο. Παρουσιάζονται εκτενώς τα πιο σημαντικά και ευρέως διαδεδομένα μοντέλα σύνθεσης ενός Χαρτοφυλακίου με βασικό αντικείμενο την Βελτιστοποίηση, και τα οποία είναι βασισμένα στη σύγχρονη Θεωρία Χαρτοφυλακίου όπου δεσπάζουν ονόματα όπως αυτά των Harry Markowitz, James Tobin και William Sharpe.

1.1 Εισαγωγή

Με τον όρο σύνθεση Χαρτοφυλακίου νοείται ο συνδυασμός διαφόρων αξιογράφων (μετοχές, ομόλογα, έντοκα γραμμάτια, αμοιβαία κεφάλαια, γeros, παράγωγα χρηματοοικονομικά προϊόντα, κ.α.) σε ένα χαρτοφυλάκιο. Βασικός στόχος κατασκευής των χαρτοφυλακίων είναι η διαφοροποίηση των επενδύσεων, δηλαδή η μείωση του επενδυτικού κινδύνου χωρίς να θυσιάζεται απαραίτητα η απόδοση. Αντίστοιχα ο όρος βελτιστοποίηση χρησιμοποιείται για την διαδικασία εύρεσης της βέλτιστης σύνθεσης ενός χαρτοφυλακίου η οποία θα δημιουργήσει την απαραίτητη σταθερότητα, μεταξύ της απόδοσης και του κινδύνου που εμπεριέχει. Ουσιαστικά στόχος της βελτιστοποίησης είναι η σταθερότητα των αποδόσεων μέσα στο χρόνο.

Η βελτιστοποίηση ως κατηγορία των χρηματοοικονομικών εντάσσεται στην στρατηγική αντιστοίχισης η οποία αποτελεί κομμάτι της παθητικής διαχείρισης χαρτοφυλακίων και αποσκοπεί στο σχεδιασμό ενός χαρτοφυλακίου το οποίο θα αντιγράφει (replicate) τις αποδόσεις ενός δείκτη, ο οποίος θα είναι πρότυπο (benchmark index), όπως είναι για παράδειγμα ο γενικός δείκτης της αγοράς, ή ένας συγκεκριμένος κλαδικός δείκτης κ.α. Η κατασκευή του στηρίζεται στις ιστορικές αποδόσεις καθώς και στις συσχετίσεις των μετοχών του δείκτη, τα δεδομένα των οποίων εισάγονται σε ένα μαθηματικό μοντέλο προγραμματισμού, με σκοπό την εύρεση της βέλτιστης σύνθεσής του.

Στα μαθηματικά αντιστοίχως, ο όρος βελτιστοποίηση χρησιμοποιείται για την διαδικασία εύρεσης τοπικών και ολικών βέλτιστων σημείων μιας αντικειμενικής συνάρτησης, ανεξάρτητα αν αυτή υπόκειται ή μη σε περιορισμούς.

1.2 Απόδοση και Κίνδυνος

Με τον όρο απόδοση (return) ορίζεται η ποσοστιαία μεταβολή που αφορά την αξία της επένδυσης κατά την διάρκεια ενός δεδομένου χρονικού διαστήματος διακράτησης $[0, t]$ και ορίζεται ως εξής:

$$R_{i,t} = \frac{V_{i,t} - V_{i,0}}{V_{i,0}}$$

όπου:

$V_{i,0}$: Αξία (ή κόστος) της i επένδυσης στην αρχή της περιόδου διακράτησης

$V_{i,t}$: Αξία της i επένδυσης στο τέλος της περιόδου διακράτησης

Εναλλακτικά η απόδοση μπορεί να οριστεί και ως ο φυσικός λογάριθμος των ποσοσοστιαίων προσαυξήσεων της τιμής της επένδυσης και δίνεται από την σχέση:

$$R_{i,t} = \ln \frac{V_{i,t}}{V_{i,0}}$$

Στην περίπτωση όπου υπάρχει προσβαση σε ιστορικά στοιχεία μιας επένδυσης, όπως ημερήσιες τιμές κλεισίματος κάποιας μετοχής, είναι εφικτό να υπολογιστεί η μέση τιμή των αποδόσεων της μετοχής, με τον αριθμητικό και τον γεωμετρικό μέσο.

Αριθμητικός Μέσος (Arithmetic mean): Ισούται με το άθροισμα των αποδόσεων, διαιρεμένο δια του συνολικού αριθμού τους. Η σχέση αυτή αναπαρίσταται με τον εξής τρόπο:

Αριθμητικός Μέσος:

$$\bar{R}_i = \frac{1}{n} \sum_{i=1}^n R_{it}$$

Η μέση αριθμητική απόδοση μπορεί να χρησιμοποιηθεί και ως μία εκτίμηση για την απόδοση της χρονικής περιόδου που ακολουθεί άμεσα.

Γεωμετρικός Μέσος (Geometric mean): Πρόκειται για την ρίζα n τάξης του γινομένου των n αριθμητικών αποδόσεων $(1 + R_{it})$, για $t = 1, 2, \dots, n$.

Γεωμετρικός Μέσος:

$$\bar{R}_{ig} = \left[\prod_{i=1}^n (1 + R_{it}) \right]^{1/n} - 1$$

Εάν οι αποδόσεις μίας επένδυσης είναι ίδιες για όλες τις περιόδους τότε ο γεωμετρικός και ο αριθμητικός μέσος έχουν το ίδιο αποτέλεσμα, όμως είναι γνωστό και από την Στατιστική ότι ο αριθμητικός μέσος είναι πιο ευαίσθητος σε ακραίες παρατηρήσεις από τον γεωμετρικό, οπότε

όσο μεγαλύτερη είναι η μεταβλητότητα των αποδόσεων τόσο πιο μεγάλη και η διαφορά των δύο μέσων.

Ως κίνδυνος ορίζεται κάθε απόκλιση του πραγματικού αποτελέσματος από το αναμενόμενο ή υποσχόμενο αποτέλεσμα. Εκτός των ασφαλών επενδύσεων είναι σχεδόν βέβαιο ότι αυτό που θα ισχύει είναι $R_i \neq E(R_i)$. Συγκεκριμένα ως κίνδυνος μπορεί να οριστεί η μεταβλητότητα (variability) των δυνητικών αποτελεσμάτων γύρω από την αναμενόμενη τιμή τους, όπου μεγάλη διαφορά ανάμεσα στο πραγματικό και το αναμενόμενο αποτέλεσμα μίας επένδυσης συνεπάγεται και μεγάλο επενδυτικό κίνδυνο. Το σίγουρο είναι ότι το σύνολο των επενδύσεων εμπεριέχουν κίνδυνο με συνέπεια οι μελλοντικές τους αποδόσεις να είναι αβέβαιες.

Στο σημείο αυτό θα πρέπει να διαχωρίσουμε και τις έννοιες κινδύνου και αβεβαιότητας στην οικονομική θεωρία. όπου σύμφωνα με τον Knight (1921) ο κίνδυνος αναφέρεται σε περιπτώσεις όπου είναι αβέβαιο το αποτέλεσμα όμως με την χρήση πιθανοτήτων μπορεί να εκτιμηθεί, αντίθετα σε συνθήκες αβεβαιότητας δεν μπορεί να γίνει εκτίμηση του αποτελέσματος ούτε με εφαρμογή της θεωρίας πιθανοτήτων.

1.3 Η Εκτίμηση της Απόδοσης και του Κινδύνου

Όταν κάποιος επενδυτής προβαίνει σε μια επένδυση, στόχος του είναι να αποκομίσει κάποια απόδοση. Η απόδοση που επιθυμεί μπορεί να πραγματοποιηθεί μπορεί και όχι, είναι δηλαδή μια τυχαία μεταβλητή X , η πραγματοποίηση της οποίας είναι ουσιαστικά μια αλληλουχία τιμών σε ένα συγκεκριμένο χρονικό διάστημα. Η αναμενόμενη απόδοση της επένδυσης X για διακριτές κατανομές υπολογίζεται με το άθροισμα των n πιθανών αποδόσεων σταθμισμένων με τις αντίστοιχες πιθανότητες p_k που εκτιμάει ο επενδυτής και δίνεται από τον παρακάτω τύπο:

$$E(X) = \sum_{k=1}^n x_k p_k$$

Όπου p_k οι αντίστοιχες πιθανότητες των πιθανών αποδόσεων x_k , με $k = 1, 2, \dots, n$.

Προηγουμένως ως κίνδυνος ορίστηκε κάθε απόκλιση του πραγματικού από το αναμενόμενο αποτέλεσμα, για την ποσοτικοποίηση και εκτίμησή του οποίου ξεχωρίζουν τρεις βασικές κατηγορίες μέτρων κινδύνου οι οποίες δίνονται παρακάτω (π.χ. βλ. Ανθρωπέλος (2017), Διαχείριση Χαρτοφυλακίου Επενδύσεων).

• **Μέτρα μεταβλητότητας (volatility):** μετρούν τις αποκλίσεις της απόδοσης γύρω από την αναμενόμενη / προσδοκώμενη απόδοση.

• **Κίνδυνος καθοδικής απόκλισης (downside risk):** εστιάζει στην αρνητική απόκλιση της απόδοσης και εκφράζεται ως η «χειρότερη τιμή» που μπορεί να συμβεί με κάποια συγκεκριμένη πιθανότητα (π.χ. **αξία σε κίνδυνο- VaR**).

• **Δείκτες ευαισθησίας (sensitivities):** μετρούν την ευαισθησία της επένδυσης ως προς ένα συγκριμένο παράγοντα κινδύνου π.χ. **beta** για την μέτρηση του συστηματικού κινδύνου.

1.3.1 Μέτρα Μεταβλητότητας

Το πιο διαδεδομένο μέτρο ποσοτικοποίησης των αποκλίσεων των αποδόσεων γύρω από την προσδοκώμενη απόδοση, δεν είναι άλλο από την τυπική απόκλιση. Η τυπική απόκλιση αποτελεί ένα μέτρο του συνολικού κινδύνου, είτε ενός χαρτοφυλακίου είτε ενός περιουσιακού στοιχείου και ο τύπος που την υπολογίζει είναι ο εξής:

$$\sigma_x = \sqrt{E(X - E(X))^2}$$

όπου σ_x , η τυπική απόκλιση των αποδόσεων, X η τυχαία μεταβλητή που εκφράζει την απόδοση της επένδυσης και $E(X)$ η μέση τιμή της.

Από τον ορισμό και τον τύπο της τυπικής απόκλισης, γίνεται αντιληπτό ότι η σ_x είναι πάντα θετικός αριθμός, ενώ η μοναδική περίπτωση που είναι ίσος με το 0, είναι όταν η τυχαία μεταβλητή X είναι ίση με την μέση τιμή της με πιθανότητα 1. Επιπροσθέτως το γεγονός ότι οι διαφορές από την μέση τιμή είναι υψωμένες στο τετράγωνο, έχει ως αποτέλεσμα η τυπική απόκλιση να δίνει ίδια βαρύτητα και στις θετικές και στις αρνητικές αποκλίσεις. Θεωρητικά αν ήταν γνωστή η κατανομή της τ.μ. X όπως στο κλασικό υπόδειγμα περιγραφής της κίνησης της τιμής μιας μετοχής, όπου η ποσοστιαία απόδοση X στον χρόνο T , ακολουθεί τη λογαριθμοκανονική κατανομή $LN(T\mu, T\sigma^2)$, και άρα ο λογάριθμος αυτής $\ln X \sim N(T\mu, T\sigma^2)$, ακολουθεί κανονική κατανομή, (π.χ. βλ. Μπούτσικας (2018)), τότε στο διάστημα $(T\mu \pm \sqrt{T}\sigma)$, η πιθανότητα να εμπεριέχεται η τιμή της $\ln X$, κατόπιν τυποποίησης θα ήταν 68.3%, ενώ στα διαστήματα $(T\mu \pm 2\sqrt{T}\sigma)$ και $(T\mu \pm 3\sqrt{T}\sigma)$, 95.5% και 99.7 % αντίστοιχα. Στο σημείο αυτό πρέπει να επισημανθεί ότι υπάρχουν περιπτώσεις με κατανομές που έχουν τυπική απόκλιση η οποία τείνει στο άπειρο, οπότε η χρησιμοποίηση της τυπικής απόκλισης ως μέτρο διασποράς είναι ακατάλληλο.

Για την γενική περίπτωση που η κατανομή της μεταβλητής X είναι άγνωστη μπορεί να γίνει χρήση της ανισότητας του Chebyshev όπου δύναται να οριστούν κάποια όρια πιθανότητας για

αυτήν δεδομένου ότι η δεύτερη ροπή της είναι πεπερασμένη δηλαδή $E|X|^2 < \infty$. Ο μαθηματικός τύπος για αυτήν την περίπτωση είναι ο κάτωθι:

$$P(|X - E(X)| \leq k\sigma_x) \geq 1 - \frac{1}{k^2}$$

Για παράδειγμα για $k = 3$, η πιθανότητα $P(X \in E(X) \pm 3\sigma_x)$ είναι τουλάχιστον $1 - \frac{1}{3^2} = 88\%$.

Επίσης είναι σημαντικό να αναφερθεί ότι υπάρχουν περιπτώσεις όπου οι επενδυτές επιδιώκουν την σύγκριση του κινδύνου των διαφόρων επενδύσεων οι οποίες παρουσιάζουν σημαντικές διαφορές όσον αφορά τις αναμενόμενες αποδόσεις τους. Σε αυτή την περίπτωση η χρήση είτε της διακύμανσης ή της τυπικής απόκλισης μπορεί να οδηγήσει σε λάθος συμπεράσματα. Αυτό συμβαίνει γιατί τόσο η τυπική απόκλιση όσο και η διακύμανση είναι απόλυτες μετρήσεις της διασποράς μίας κατανομής. Αυτό το πρόβλημα μπορεί να εξαλειφθεί μέσω της χρησιμοποίησης ενός σχετικού μέτρου της διασποράς όπως ο συντελεστής μεταβλητότητας (**coefficient of variation – CV**), ο οποίος μετρά τον κίνδυνο ανά μονάδα αναμενόμενης απόδοσης, καθορίζεται από το λόγο της τυπικής απόκλισης προς την αναμενόμενη τιμή και είναι καθαρός αριθμός.

Η αναφερθείσα αυτή σχέση αναπαρίσταται ως εξής :

$$CV = \frac{\sigma_x}{E(X)}$$

Ενώ το σημαντικότερο πλεονέκτημα της τυπικής απόκλισης ως μέτρο κινδύνου είναι ότι είναι απλό και κατανοητό και για αυτό το λόγο είναι και ευρέως διαδεδομένο, έχει μία σειρά από μειονεκτήματα:

- Σε κατανομές όπου η τυπική απόκλιση δεν είναι πεπερασμένη, η χρησιμοποίηση της ως μέτρο κινδύνου δεν είναι δυνατή.
- Δίνει ίδια βαρύτητα και στις θετικές (π.χ. κέρδη) και στις αρνητικές αποκλίσεις (π.χ. ζημίες).
- Δεν μας παρέχει καμία πληροφορία για το πόσο μεγάλη ζημία μπορεί να παρουσιαστεί, π.χ. σε κατανομές με ίδια διασπορά, όπως η $N(0,2)$ και η t_4 , η πιθανότητα εμφάνισης πολύ μεγάλης ζημίας είναι μεγαλύτερη στην δεύτερη περίπτωση διότι έχει βαρύτερη ουρά (π.χ. βλ. Μπούτσικας, (2018)). Χρησιμοποιώντας λοιπόν ως μέτρο κινδύνου την τυπική απόκλιση γίνεται κατανοητό ότι δεν θα εντοπιστεί αυτή η σημαντική λεπτομέρεια.

Το λογικό συμπέρασμα που προκύπτει λοιπόν είναι ότι χρειάζόταν ένα μέτρο κινδύνου πάλι κατανοητό σαν έννοια και απλό στην χρήση, όπου θα μας δίνει την πιθανότητα για το χειρότερο δυνατό σενάριο (ζημία) που αφορά την εκάστοτε επένδυση σε έναν δεδομένο χρονικό ορίζοντα, ένα τέτοιο μέτρο το οποίο θα αναλυθεί στη συνέχεια και χρησιμοποιείται ευρέως στα Χρηματοοικονομικά είναι η Αξία σε Κίνδυνο (**VaR, Value at Risk**).

1.3.2 Κίνδυνος καθοδικής απόκλισης (downside risk)

Παραπάνω αναφέρθηκε η έννοια της επένδυσης, όπου ουσιαστικά είναι μια πράξη με χρόνο έναρξης $T = 0$ την στιγμή της επένδυσης του αρχικού κεφαλαίου V_0 , (π.χ. η αρχική τιμή κτίσης ενός αξιογράφου) και με μελλοντική αξία V_T στον χρόνο $T > 0$ η οποία είναι τυχαία μεταβλητή καθώς δεν είναι γνωστή εκ των προτέρων και μπορεί μόνο να εκτιμηθεί. Ορίζεται ως **κέρδος** λοιπόν στο χρονικό διάστημα $[0, T]$ η ποσότητα $P = V_T - V_0$ (με προεξόφληση σε παροντικό χρόνο), και ως **ζημία** η αντίθετη ποσότητα $L = -(V_T - V_0)$, (αρνητική ζημία σημαίνει κέρδος). Βάσει των παραπάνω εννοιών του κέρδους και της ζημίας, προκύπτει το εύλογο ερώτημα ποια μπορεί να είναι η τιμή την οποία υπερβαίνει η ζημία της επένδυσής, με πιθανότητα α σε συγκεκριμένο χρονικό ορίζοντα. Σε αυτό το ερώτημα στηρίζεται και ο παρακάτω ορισμός για την **Αξία σε Κίνδυνο VaR**:

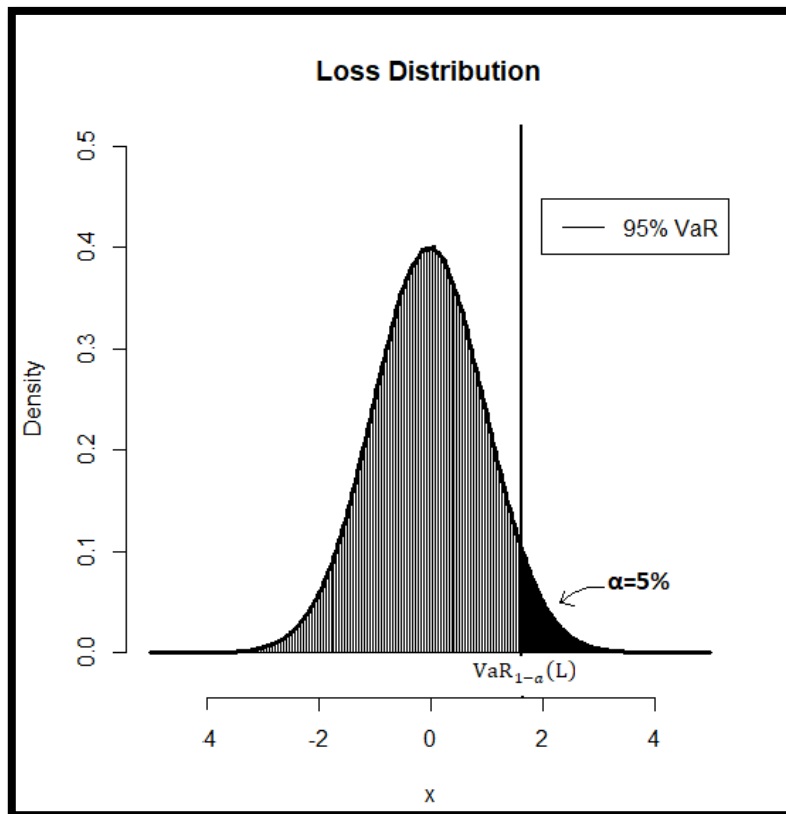
Ως Αξία σε Κίνδυνο λοιπόν ορίζεται η αναμενόμενη απώλεια σε ένα συγκεκριμένο χρονικό ορίζοντα σε κάποιο δεδομένο επίπεδο εμπιστοσύνης. Είναι δηλαδή η τιμή την οποία υπερβαίνει η **ζημία L** με πιθανότητα $\alpha \in (0,1)$.

$$P(L > \text{VaR}_{1-\alpha}(L)) = \alpha$$

Οι πιο γνωστές μέθοδοι εκτίμησης της Αξίας σε Κίνδυνο VaR είναι η παραμετρική μέθοδος, η μη παραμετρική μέθοδος και η Μέθοδος Monte Carlo (π.χ. βλ. Μπούτσικας, (2018)). Στην πιο απλή περίπτωση όπου είναι γνωστό ότι η κατανομή της ζημίας ακολουθεί την κανονική κατανομή (με γνωστές ή εκτιμημένες παραμέτρους, μ, σ^2) προκύπτει εύκολα η παρακάτω σχέση:

$$\text{VaR}_{1-\alpha}(L) = \mu + \sigma z_\alpha$$

Όπου z_α είναι το άνω α -σημείο της $N(0,1)$.



Διάγραμμα 1.1 Value at Risk

1.3.3 Δείκτες ευαισθησίας (sensitivities)

Ο πιο διαδεδομένος δείκτης ευαισθησίας είναι ο συντελεστής β_p , ο οποίος είναι το βασικό μέτρο του συστηματικού κινδύνου και ορίζεται ως ο λόγος της συνδιακύμανσης σ_{PM} των αποδόσεων του χαρτοφυλακίου P σε σχέση με τις αποδόσεις της αγοράς M , προς την διασπορά των αποδόσεων της αγοράς σ_M^2 :

$$\beta_p = \frac{\sigma_{PM}}{\sigma_M^2}$$

Η παράμετρος βήτα είναι η ευαισθησία που επιδεικνύει η απόδοση του περιουσιακού στοιχείου στην απόδοση της αγοράς. Τα χαρτοφυλάκια ή χρεόγραφα με συστηματικό κίνδυνο πιο υψηλό από την μονάδα $\beta_p > 1$ είναι πιθανό να παρουσιάσουν πιο υψηλές μεταβολές από την αγορά, να είναι δηλαδή πιο «επιθετικά» και αναμένεται να έχουν και υψηλότερη απόδοση. Αντιθέτως όσα χαρτοφυλάκια ή χρεόγραφα έχουν συστηματικό κίνδυνο πιο χαμηλό από τη μονάδα, $\beta_p < 1$ είναι πιο «αμυντικά» και αναμένεται να εμφανίσουν μικρότερες μεταβολές

σε σχέση με την αγορά ενώ αναμένεται να έχουν και χαμηλότερη αναμενόμενη απόδοση. Όσα χαρτοφυλάκια ή χρεόγραφα έχουν συστηματικό κίνδυνο μηδέν, δεν επηρεάζονται από τον τρόπο με τον οποίο μεταβάλλεται η αγορά.

1.4 Σύνθεση Χαρτοφυλακίου και Διαφοροποίηση

Τα κριτήρια τόσο της απόδοσης όσο και του κινδύνου μπορεί να οριστούν και στην περίπτωση των χαρτοφυλακίων χρεογράφων, όπου ο βασικός λόγος κατασκευής τους είναι η μείωση του επενδυτικού κινδύνου μέσω της διαφοροποίησης. Στην απλούστερη περίπτωση όπου εξετάζονται τα χαρτοφυλάκια δύο χρεογράφων, με αναμενόμενες αποδόσεις $E(r_1)$ και $E(r_2)$, διασπορές σ_1^2, σ_2^2 και συνδιακύμανση σ_{12} , ορίζοντας ως x_1, x_2 τα ποσοστά του διαθέσιμου κεφαλαίου που θα επενδυθούν στο κάθε χρεόγραφο, η αναμενόμενη απόδοση $E(r_p)$ και ο κίνδυνος σ_p^2 του χαρτοφυλακίου ορίζονται ως εξής:

- Απόδοση για σύνθεση χαρτοφυλακίου με 2 αξιόγραφα

$$E(r_p) = E(x_1 r_1 + x_2 r_2) = x_1 E(r_1) + x_2 E(r_2)$$

- Κίνδυνος για σύνθεση χαρτοφυλακίου με 2 αξιόγραφα

$$\begin{aligned} \sigma_p^2 &= V(x_1 r_1 + x_2 r_2) = x_1^2 V(r_1) + x_2^2 V(r_2) + 2x_1 x_2 \text{Cov}(r_1, r_2) \\ &= x_1^2 \sigma_1^2 + x_2^2 \sigma_2^2 + 2x_1 x_2 \sigma_{12} \end{aligned}$$

Η συνδιακύμανση σχετίζεται με τις τυπικές αποκλίσεις των αποδόσεων βάσει της σχέσης $\sigma_{ij} = \rho_{ij} \sigma_i \sigma_j$, όπου με $\rho_{ij} \in [-1, 1]$, συμβολίζεται ο συντελεστής συσχέτισης των αποδόσεων των χρεογράφων. Εάν ο συντελεστής συσχέτισης είναι $\rho = -1$ τότε οι αποδόσεις των δύο χρεογράφων είναι γραμμικά συσχετισμένες αλλά μεταβάλλονται προς τις αντίθετες κατευθύνσεις, εάν είναι $\rho = +1$ τότε οι αποδόσεις των χρεογράφων είναι γραμμικά συσχετισμένες και μεταβάλλονται προς την ίδια κατεύθυνση, ενώ για $\rho = 0$ δεν υπάρχει γραμμική συσχέτιση μεταξύ των αποδόσεων των δύο αξιόγραφων χωρίς αυτό να σημαίνει ότι είναι και ανεξάρτητες, παρά μόνο αν συγχρόνως ακολουθούν κανονική κατανομή. Στο σημείο αυτό θα οριστεί και **το χαρτοφυλάκιο ελάχιστης διακύμανσης (Minimum Variance Portfolio, MVP)** δηλαδή αυτό με τον μικρότερο κίνδυνο, η σύνθεση του οποίου για δύο αξιόγραφα υπολογίζεται μηδενίζοντας την μερική παράγωγο $\partial \sigma_p^2 / \partial x_1$, με $x_1 + x_2 = 1$.

$$\begin{aligned} \frac{\partial \sigma_p^2}{\partial x_1} = 0 &\Leftrightarrow \frac{\partial}{\partial x_1} (x_1^2 \sigma_1^2 + (1 - x_1)^2 \sigma_2^2 + 2x_1(1 - x_1)\sigma_{12}) = 0 \\ &\Leftrightarrow x_1 = \frac{\sigma_2^2 - \sigma_{12}}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}} \end{aligned}$$

Στην γενική περίπτωση ενός χαρτοφυλακίου P αποτελούμενο από n χρεόγραφα, η αναμενόμενη απόδοση και ο κίνδυνος του χαρτοφυλακίου υπολογίζονται ως εξής :

- Απόδοση για σύνθεση χαρτοφυλακίου με n αξιόγραφα

$$E(r_p) = \sum_{i=1}^n x_i E(r_i)$$

Όπου $E(r_p)$ η εκτιμώμενη απόδοση του χαρτοφυλακίου, x_i το ποσοστό του διαθέσιμου κεφαλαίου που επενδύεται στο i αξιόγραφο και $E(r_i)$ η αναμενόμενη απόδοση αυτού.

- Κίνδυνος χαρτοφυλακίου με n αξιόγραφα

$$\sigma_p^2 = \sum_{i=1}^n x_i^2 \sigma_i^2 + \sum_{i=1}^n \sum_{j=1, j \neq i}^n x_i x_j \sigma_{ij}$$

Όπου σ_{ij} = η συνδιακύμανση (covariance) μεταξύ των i και j , x_i, x_j τα ποσοστά του κεφαλαίου που έχουν επενδυθεί στα αξιόγραφα i και j αντίστοιχα, με $i \neq j$ και n ο συνολικός αριθμός των αξιογράφων. Οι προαναφερόμενες σχέσεις μπορούν να αποδοθούν απλούστερα σε μορφή πινάκων ως εξής :

$$E(r_p) = \mathbf{r}^T \mathbf{x}, \quad \sigma_p^2 = \mathbf{x}^T \mathbf{\Sigma} \mathbf{x}$$

Όπου:

- \mathbf{r} είναι ο πίνακας – στήλη των αναμενόμενων αποδόσεων, διάστασης $n \times 1$ και \mathbf{r}^T ο ανάστροφός του διάστασης $1 \times n$.
- \mathbf{x} είναι ο πίνακας – στήλη των ποσοστών συμμετοχής των χρεογράφων στο χαρτοφυλάκιο, διάστασης $n \times 1$ και \mathbf{x}^T ο ανάστροφός του διάστασης $1 \times n$.
- $\mathbf{\Sigma}$ ο πίνακας διακυμάνσεων – συνδιακυμάνσεων διαστάσεων $n \times n$, ο οποίος πρέπει να είναι συμμετρικός και θετικά ορισμένος, για να υπάρχει και ο αντίστροφός του $\mathbf{\Sigma}^{-1}$.

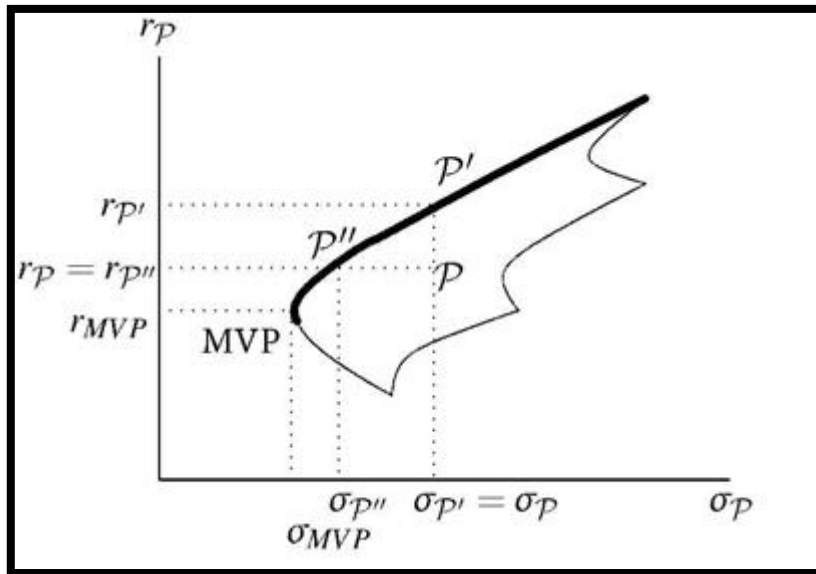
Όσον αφορά την αναμενόμενη απόδοση του χαρτοφυλακίου τα πράγματα δείχνουν απλά καθώς πρόκειται για γραμμική συνάρτηση των αποδόσεων των επιμέρους αξιόγραφων που το αποτελούν. Άρα θεωρώντας ότι $x \geq 0$ και θέτοντας ως $x_{j^*} = 1$ το ποσοστό του κεφαλαίου που θα επενδυθεί στο j^* αξιόγραφο με την μεγαλύτερη απόδοση και $x_i = 0$ για κάθε άλλο αξιόγραφο $i \neq j^*$, μπορεί να βρεθεί η μέγιστη απόδοση του χαρτοφυλακίου. Αυτό που χρήζει περαιτέρω ανάλυσης είναι η συμπεριφορά του κινδύνου του χαρτοφυλακίου ο οποίος είναι μη

γραμμική συνάρτηση των ποσοστών συμμετοχής των χρεογράφων. Άλλωστε όπως προαναφέρθηκε στόχος της βελτιστοποίησης είναι η σταθερότητα που είναι άμεσα συνυφασμένη με τον κίνδυνο, και όχι τόσο η μεγιστοποίηση της απόδοσης.

Ολοκληρώνοντας επισημαίνεται ότι ο κίνδυνος που εμπεριέχει ένα χαρτοφυλάκιο και μπορεί να εξαλειφθεί μέσω εφαρμογής της κατάλληλης στρατηγικής διαφοροποίησης ονομάζεται **μη συστηματικός κίνδυνος (non-systematic risk)** και αφορά αποκλειστικά το κάθε χρεόγραφο του χαρτοφυλακίου χωρίς να επηρεάζεται από τη συμπεριφορά των υπολοίπων χρεογράφων. Ενώ το μέρος του κινδύνου που δεν μπορεί να εξαλειφθεί μέσω της διαφοροποίησης αφορά **τον συστηματικό κίνδυνο (systematic risk)**, ο οποίος μπορεί να ποσοτικοποιηθεί μέσω του συντελεστή β_p ο οποίος αναφέρθηκε σε προηγούμενη ενότητα.

1.5 Βελτιστοποίηση χαρτοφυλακίου

Βάσει της παραπάνω ανάλυσης έγινε κατανοητό ότι δύναται να μειωθεί ο κίνδυνος μιας επένδυσης χωρίς απαραίτητα να θυσιαστεί η απόδοση, εφαρμόζοντας μια καλά σχεδιασμένη στρατηγική διαφοροποίησης. Η σχέση αυτή απόδοσης και κινδύνου θα είναι ο πυρήνας κατασκευής ενός εφικτού συνόλου χαρτοφυλακίων, όπου σαν **εφικτό σύνολο χαρτοφυλακίων (feasible set of portfolios)**, ορίζεται το σύνολο των πιθανών συνδυασμών αναμενόμενης απόδοσης και τυπικής απόκλισης των αποδόσεων των αξιογράφων που εμπεριέχονται σε πολλά χαρτοφυλάκια (βλ. Βασιλείου και Ηρειώτης (2009)). Πάνω σε αυτή τη σχέση αναμενόμενης απόδοσης και κινδύνου θα θεμελιωθεί και το μεθοδολογικό πλαίσιο που ανέπτυξε ο Νομπελίστας Harry Markowitz στη δεκαετία του 1950 (βλ. Markowitz, (1952)), όπου και εισήγαγε την έννοια των **αποτελεσματικών χαρτοφυλακίων (efficient portfolios)**, δηλαδή των χαρτοφυλακίων εκείνων του εφικτού συνόλου τα οποία υπερέχουν σε σχέση με τα υπόλοιπα, καθώς για ένα δεδομένο επίπεδο κινδύνου παρέχουν την μέγιστη δυνατή απόδοση, ενώ για δεδομένη απόδοση εμπεριέχουν τον μικρότερο κίνδυνο. Το καλύτερο χαρτοφυλάκιο το οποίο πρέπει να επιλέξει ο επενδυτής σε σχέση με το σύνολο των αποτελεσματικών χαρτοφυλακίων ονομάζεται άριστο ή **βέλτιστο χαρτοφυλάκιο (optimal portfolio)** όπου σημαντικό ρόλο στην επιλογή του διαδραματίζουν οι προτιμήσεις του εκάστοτε επενδυτή. Όμως κανένας επενδυτής ο οποίος είναι “λογικός” δεν θα προτιμούσε ένα χαρτοφυλάκιο κάτω από την καμπύλη αποτελεσματικών χαρτοφυλακίων καθώς θα ήταν ένα εφικτό μεν αλλά υποδεέστερο δε χαρτοφυλάκιο. Για παράδειγμα όπως φαίνεται και στο παρακάτω σχήμα κανείς δεν θα επέλεγε το χαρτοφυλάκιο P καθώς σε ίδιο επίπεδο κινδύνου με το P' , $\sigma_P = \sigma_{P'}$ υστερεί σε απόδοση $E(r_P) < E(r_{P'})$, ενώ σε ίδιο επίπεδο απόδοσης με το P'' , $E(r_P) = E(r_{P''})$ παρουσιάζει μεγαλύτερο κίνδυνο καθώς $\sigma_P > \sigma_{P''}$



Πηγή:(Dietmar Maringer,2005)

Διάγραμμα 1.2 Αποτελεσματικό σύνορο του Markowitz

Παρακάτω αναλύονται τα πιο βασικά μοντέλα βελτιστοποίησης, για τα οποία ως μέτρο κινδύνου χρησιμοποιείται η διακύμανση και πιο συγκεκριμένα η ημιδιακύμανση, όμως κάλλιστα μπορεί να χρησιμοποιηθούν και άλλα μέτρα κινδύνου όπως το VaR, η ζημία L ή ακόμα σε πιο σύγχρονη προσέγγιση, το Sharpe Ratio του χαρτοφυλακίου, το οποίο είναι η ανά μονάδα κινδύνου υπερβάλλουσα αναμενόμενη απόδοση (excess expected return) και χρησιμοποιείται συχνά και αποτελεσματικά ως αντικειμενική συνάρτηση προς βελτιστοποίηση και συγκεκριμένα μεγιστοποίηση.

1.5.1 The Black Model

Αρχικά θα εξεταστεί η πιο απλή περίπτωση όπου αναλύεται το μοντέλο που προτάθηκε από τον **Black (1972)** το οποίο είναι ουσιαστικά προέκταση του μοντέλου του Markowitz με την διαφορά ότι σε αυτό το μοντέλο επιτρέπονται οι ανοιχτές πωλήσεις. Με τον όρο ανοιχτές πωλήσεις (**short selling**) εννοούμε ότι ο επενδυτής μπορεί να αγοράσει ένα αξιόγραφο στην σημερινή του τιμή, το αντίτιμο του οποίου μπορεί να καταβάλει σε κάποια συμφωνημένη στιγμή στο μέλλον, ο επενδυτής προσδοκά ότι μελλοντικά η τιμή του αξιόγραφου θα ανέβει και η διαφορά αυτή είναι το εκτιμώμενο κέρδος του.

Πρακτικά για το μοντέλο μας σημαίνει ότι τα ποσοστά συμμετοχής των αξιόγραφων στην σύνθεση του χαρτοφυλακίου μπορούν να πάρουν και αρνητικές τιμές, και αυτή είναι και η

διαφορά με το μοντέλο του Markowitz το οποίο έχει ως περιορισμό την θετικότητα του διανύσματος των βαρών, $\mathbf{x} \geq \mathbf{0}$. Το πρόβλημα λοιπόν βελτιστοποίησης διαμορφώνεται ως εξής:

$$\begin{aligned} \min \quad & \sigma_p^2 = \frac{1}{2} \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} \\ \text{Περιορισμοί} \quad & \mathbf{e}^T \mathbf{x} = 1 \\ & \mathbf{r}^T \mathbf{x} = R^* \\ & \mathbf{x} \in \mathbb{R} \end{aligned}$$

Όπου:

- \mathbf{e} είναι ο πίνακας – στήλη διάστασης $n \times 1$, και \mathbf{e}^T ο ανάστροφός του διάστασης $1 \times n$.
- \mathbf{x} είναι ο πίνακας – στήλη των ποσοστών συμμετοχής των χρεογράφων στο χαρτοφυλάκιο, διάστασης $n \times 1$ και \mathbf{x}^T ο ανάστροφός του διάστασης $1 \times n$.
- \mathbf{r} είναι ο πίνακας – στήλη των αναμενόμενων αποδόσεων, διάστασης $n \times 1$ και \mathbf{r}^T ο ανάστροφός του διάστασης $1 \times n$.
- $\boldsymbol{\Sigma}$ είναι ο πίνακας διακυμάνσεων – συνδιακυμάνσεων διαστάσεων $n \times n$.

Το παραπάνω πρόβλημα μπορεί να δώσει μία κλειστού τύπου λύση για τα βάρη ελαχιστοποιώντας τον κίνδυνο, έχοντας συγκεκριμένο κατώφλι για την επιθυμητή απόδοση. Η μέθοδος που θα χρησιμοποιηθεί βασίζεται στους πολλαπλασιαστές του **Lagrange**. Συμβολίζοντας λοιπόν ως λ_1, λ_2 τις ψευδομεταβλητές που ονομάζονται πολλαπλασιαστές **Lagrange**, καθορίζεται η παρακάτω **Lagrangian** συνάρτηση,

$$L = \frac{1}{2} \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} + \lambda_1 (1 - \mathbf{e}^T \mathbf{x}) + \lambda_2 (R^* - \mathbf{r}^T \mathbf{x})$$

Στην συνέχεια οι παράγωγοι πρώτης τάξης της Lagrangian, ως προς \mathbf{x} και ως προς τους πολλαπλασιαστές **Lagrange** τίθενται ίσοι με 0 και λύνουμε το παρακάτω σύστημα εξισώσεων.

$$\frac{\partial L}{\partial \mathbf{x}} = 0 \Rightarrow \boldsymbol{\Sigma} \mathbf{x} - \lambda_1 \mathbf{e} - \lambda_2 \mathbf{r} = 0 \quad (1.1)$$

$$\frac{\partial L}{\partial \lambda_1} = 0 \Rightarrow \mathbf{e}^T \mathbf{x} = 1 \quad (1.2)$$

$$\frac{\partial L}{\partial \lambda_2} = 0 \Rightarrow \mathbf{r}^T \mathbf{x} = R^* \quad (1.3)$$

Από την σχέση (1.1) προκύπτει ότι:

$$\mathbf{x} = \lambda_1 \boldsymbol{\Sigma}^{-1} \mathbf{e} + \lambda_2 \boldsymbol{\Sigma}^{-1} \mathbf{r} \quad (1.4)$$

Αντικαθιστώντας το \mathbf{x} στις σχέσεις (1.2), (1.3) προκύπτει το εξής σύστημα εξισώσεων:

$$\begin{cases} \lambda_1 \mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{e} + \lambda_2 \mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{r} = 1 \\ \lambda_1 \mathbf{r}^T \boldsymbol{\Sigma}^{-1} \mathbf{e} + \lambda_2 \mathbf{r}^T \boldsymbol{\Sigma}^{-1} \mathbf{r} = R^* \end{cases}$$

Δεδομένου ότι ο πίνακας $\boldsymbol{\Sigma}^{-1}$ είναι συμμετρικός ισχύει ότι $\mathbf{r}^T \boldsymbol{\Sigma}^{-1} \mathbf{e} = \mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{r}$, θέτοντας λοιπόν $a = \mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{e}$, $b = \mathbf{r}^T \boldsymbol{\Sigma}^{-1} \mathbf{e}$, $c = \mathbf{r}^T \boldsymbol{\Sigma}^{-1} \mathbf{r}$ το παραπάνω σύστημα παίρνει την ακόλουθη μορφή.

$$\begin{cases} a\lambda_1 + b\lambda_2 = 1 \\ b\lambda_1 + c\lambda_2 = R^* \end{cases}$$

Στην πρώτη περίπτωση πολλαπλασιάζοντας την πάνω σχέση με $(-b)$ και την κάτω με (a) και προσθέτοντας κατά μέλη προκύπτουν τα εξής αποτελέσματα:

$$\begin{cases} a\lambda_1 + b\lambda_2 = 1 \\ b\lambda_1 + c\lambda_2 = R^* \end{cases} \xrightarrow{* \begin{pmatrix} -b \\ a \end{pmatrix}} \begin{cases} -ab\lambda_1 - b^2\lambda_2 = -b \\ ab\lambda_1 + ac\lambda_2 = aR^* \end{cases} \xrightarrow{(+)} \lambda_2 = \frac{aR^* - b}{ac - b^2}$$

Αντίστοιχα στην δεύτερη περίπτωση πολλαπλασιάζοντας την πάνω σχέση με $(-c)$ και την κάτω με (b) και προσθέτοντας κατά μέλη προκύπτουν τα εξής αποτελέσματα:

$$\begin{cases} a\lambda_1 + b\lambda_2 = 1 \\ b\lambda_1 + c\lambda_2 = R^* \end{cases} \xrightarrow{* \begin{pmatrix} -c \\ b \end{pmatrix}} \begin{cases} -ac\lambda_1 - bc\lambda_2 = -c \\ b^2\lambda_1 + bc\lambda_2 = bR^* \end{cases} \xrightarrow{(+)} \lambda_1 = \frac{c - bR^*}{ac - b^2}$$

Με βάση τα παραπάνω λοιπόν μπορεί να προσδιοριστεί η σύνθεση του **βέλτιστου χαρτοφυλακίου**, αντικαθιστώντας τα λ_1, λ_2 στην σχέση (1.4) όπου τελικά προκύπτει:

$$\mathbf{x} = \frac{c - bR^*}{ac - b^2} \boldsymbol{\Sigma}^{-1} \mathbf{e} + \frac{aR^* - b}{ac - b^2} \boldsymbol{\Sigma}^{-1} \mathbf{r} \quad (1.5)$$

Από τις σχέσεις (1.1), (1.2), (1.3), υπολογίζεται και ο **κίνδυνος του βέλτιστου χαρτοφυλακίου** ως εξής:

$$\begin{aligned} \boldsymbol{\Sigma} \mathbf{x} - \lambda_1 \mathbf{e} - \lambda_2 \mathbf{r} = 0 &\xrightarrow{* \begin{pmatrix} \mathbf{x}^T \end{pmatrix}} \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} - \lambda_1 \mathbf{x}^T \mathbf{e} - \lambda_2 \mathbf{x}^T \mathbf{r} = 0 \\ &\Rightarrow \sigma_P^2 - \lambda_1 - \lambda_2 R^* = 0 \end{aligned} \quad (1.6)$$

$$\sigma_P^2 = \frac{aR^2 - 2bR^* + c}{ac - b^2} \quad (1.7)$$

Όλη η διαδικασία που αναλύθηκε οδηγεί σε κλειστούς τύπους και σε εύκολους υπολογισμούς, των a, b, c για τον καθορισμό της σύνθεσης του βέλτιστου χαρτοφυλακίου πάντα βάση της επιθυμητής απόδοσης που τίθεται στους περιορισμούς. Εάν στην παραπάνω ανάλυση δεν οριστεί ως περιορισμός κάποιο επιθυμητό επίπεδο απόδοσης, θέτοντας δηλαδή $\lambda_2 = 0$ αυτό που υπολογίζεται είναι το χαρτοφυλάκιο ελάχιστου κινδύνου.

Από το σύστημα εξισώσεων

$$\begin{cases} a\lambda_1 + b\lambda_2 = 1 \\ b\lambda_1 + c\lambda_2 = R^* \end{cases} \xrightarrow{\lambda_2=0}$$

προκύπτει ότι $\lambda_1 = 1/a$ από την πρώτη σχέση, ενώ διαιρώντας κατά μέλη προκύπτει ότι $R^* = b/a$.

Σύμφωνα με τα παραπάνω λοιπόν η σύνθεση του χαρτοφυλακίου ελάχιστου κινδύνου όπως προκύπτει και από τις σχέσεις (1.4) θα είναι η εξής:

$$\begin{aligned} \mathbf{x} &= \lambda_1 \boldsymbol{\Sigma}^{-1} \mathbf{e} + \lambda_2 \boldsymbol{\Sigma}^{-1} \mathbf{r} \xrightarrow{\lambda_2=0} \mathbf{x} = \lambda_1 \boldsymbol{\Sigma}^{-1} \mathbf{e} \xrightarrow{\lambda_1=1/a} \mathbf{x} = \frac{1}{a} \boldsymbol{\Sigma}^{-1} \mathbf{e} \\ \mathbf{x} &= \frac{\boldsymbol{\Sigma}^{-1} \mathbf{e}}{\mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{e}} \end{aligned}$$

Και αντίστοιχα ο κίνδυνος του χαρτοφυλακίου ελάχιστου κινδύνου θα είναι :

$$\begin{aligned} \sigma_p^2 - \lambda_1 - \lambda_2 R^* &= 0 \xrightarrow{\lambda_2=0} \sigma_p^2 = \lambda_1 \\ \sigma_p^2 &= \frac{1}{a} \end{aligned}$$

Ακολουθεί μια απλή εφαρμογή των παραπάνω τύπων που οδηγούν σε κλειστού τύπου λύση, σε γλώσσα προγραμματισμού R. Το παράδειγμα αφορά 3 κοινές μετοχές (Rachev, 2008) με διάνυσμα αναμενόμενων αποδόσεων $\mathbf{r}^T = (1.8\%, 2.5\%, 1\%)$ και επιθυμητή απόδοση $R^* = 1.8\%$. Ο αντίστοιχος πίνακας Διακυμάνσεων – Συνδιακυμάνσεων δίνεται παρακάτω:

$$\Sigma = \begin{bmatrix} 1.68 & 0.34 & 0.38 \\ 0.34 & 3.09 & -1.59 \\ 0.38 & -1.59 & 1.54 \end{bmatrix}$$

Κώδικας:

```
### Function for optimal portfolio with Lagrangian multipliers ###
Lagr <- function(e, r, V, R) {

  a<-t(e)%%solve(V)%%e
  b<-t(r)%%solve(V)%%e
  c<-t(r)%%solve(V)%%r
  L1<-(c-(b*R))/((a*c)-b^2)
  L1<-as.numeric(L1)
  L2<-((a*R)-b)/((a*c)-b^2)
  L2<-as.numeric(L2)
  X<-L1*solve(V)%%e + L2*solve(V)%%r
  X<-c(round(X,3))
  Sp <- round(as.numeric((a*R^2)-(2*b*R)+c)/as.numeric(a*c-b^2),3)
  cat("Weights = (",X,")","\n")
  cat("Portfolio Variance = (",Sp,")","\n")
}
### Function inputs & execution ###
e <- matrix(c(1,1,1),3)
r <- matrix(c(0.018,0.025,0.01),3)
V <- matrix(c(1.68,0.34,0.38,0.34,3.09,-1.59,0.38,-1.59,1.54),3)
R <- 0.018
Lagr(e, r, V, R)
### Output ###
Weights = ( 0.046 0.509 0.445 )
Portfolio Variance = ( 0.42 )
```

Τελικά το πρόγραμμα μας δίνει σαν βέλτιστη λύση ότι ο ελάχιστος κίνδυνος είναι ίσος με: $\sigma_p^2 = 0.422$ και αντίστοιχα βάρη ίσα με 0.046, 0.509 και 0.445, δηλαδή $x_1 = 4.6\%$, $x_2 = 50.9\%$ και $x_3 = 44.5\%$. Αυτό που παρατηρείται είναι ότι για την ίδια απόδοση $r^T x = 0.018$, θα μπορούσε να επενδυθεί όλο το κεφάλαιο, δηλαδή να δοθεί βάρος ίσο με 1, ο στην πρώτη μετοχή η οποία έχει απόδοση ίση με 0.018, τότε το χαρτοφυλάκιο θα ήταν μεν εφικτό αλλά όχι αποτελεσματικό καθότι θα είχε κίνδυνο ίσο με $\sigma_p^2 = \sigma_1^2 = 1.68$. Εδώ φαίνεται ξεκάθαρα και το αποτέλεσμα της διαφοροποίησης όπου με έναν απλό συνδυασμό 3 μετοχών χωρίς να θυσιάσει η απόδοση, ο κίνδυνος μειώθηκε περίπου 4 φορές σε σχέση με αυτόν της μία μετοχής.

Είναι προφανές ότι το παραπάνω πρόβλημα βελτιστοποίησης το οποίο αφορούσε την ελαχιστοποίηση του κινδύνου με βάση κάποιο επίπεδο προσδοκώμενης απόδοσης μπορεί να μετατραπεί σε πρόβλημα μεγιστοποίησης της απόδοσης βάση κάποιου περιορισμού για το επιθυμητό επίπεδο κινδύνου, και το οποίο θα διαμορφωνόταν ως εξής:

$$\begin{aligned} \max & & r^T x \\ \text{Περιορισμοί:} & & e^T x = 1 \\ & & \frac{1}{2} x^T \Sigma x = W \\ & & x \in \mathbb{R} \end{aligned}$$

Το μοντέλο του Black που αναφέρθηκε είναι το πιο απλό καθώς επιτρέπει τις ανοιχτές πωλήσεις και δεν θέτει περιορισμό ως προς το πρόσημο των βαρών. Θέτοντας όμως ως περιορισμό στα βάρη να είναι θετικά δηλαδή $\mathbf{x} \in \mathbb{R}^+$, αυτόματα το πρόβλημα μας αντιστοιχεί στην κλασσική έκφραση του υποδείγματος μέσου-διακύμανσης (mean-variance), το οποίο προτάθηκε από τον Markowitz (1952).

1.5.2 The Markowitz Model

Στην περίπτωση αυτή θα εξεταστεί το κλασσικό και πιο βασικό μοντέλο στην βελτιστοποίηση χαρτοφυλακίου, αυτό που βασίζεται στο υπόδειγμα του Markowitz. Ουσιαστικά όπως ήδη έχει αναφερθεί, στο μοντέλο αυτό δεν επιτρέπονται οι ανοιχτές πωλήσεις και το αποτελεσματικό μέτωπο χαρτοφυλακίων προσδιορίζεται ελαχιστοποιώντας τον κίνδυνο για διάφορα επίπεδα απόδοσης. Και σε αυτήν την περίπτωση αν δεν οριστεί ως περιορισμός κάποιο επίπεδο προσδοκώμενης απόδοσης, αυτό που υπολογίζεται είναι το χαρτοφυλάκιο ελάχιστου κινδύνου. Το πρόβλημα λοιπόν βελτιστοποίησης διαμορφώνεται ως εξής:

$$\begin{aligned} \min \quad & \sigma_p^2 = \frac{1}{2} \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} \\ \text{Περιορισμοί:} \quad & \mathbf{e}^T \mathbf{x} = 1 \\ & \mathbf{r}^T \mathbf{x} = R^* \\ & \mathbf{x} \geq 0 \end{aligned}$$

Η μέθοδος επίλυσης αυτού του προβλήματος ακολουθεί σχεδόν την ίδια μεθοδολογία όπως αυτή αναλύθηκε προηγουμένως επιβάλλοντας κάποιες επιπλέον συνθήκες λόγω της ανισότητας του \mathbf{x} στους περιορισμούς. Για την επίλυση του προβλήματος αυτού γίνεται χρήση περιορισμών που είναι γνωστοί ως συνθήκες **Karush-Kuhn-Tucker (KKT)**. Η λογική τους είναι ότι εφόσον βρεθεί λύση που τις ικανοποιεί, τότε η λύση αυτή αντιστοιχεί στο βέλτιστο χαρτοφυλάκιο.

Η επίλυση των συνθηκών KKT είναι μια επίπονη εργασία που δυσκολεύει όσο αυξάνεται ο αριθμός των περιορισμών. Η μη αρνητικότητα των βαρών στους περιορισμούς καθιστά την εύρεση αναλυτικής λύσης πρακτικά αδύνατη, ουσιαστικά μετατρέπει το υπόδειγμα του Markowitz σε NP πρόβλημα (**non-deterministic polynomial time complete**) όπου δεν υπάρχει γνωστός αναλυτικός αλγόριθμος να μπορεί να βρει λύση σε πολυωνυμικό χρόνο. Σε αυτήν την περίπτωση αριθμητικές μέθοδοι όπως ο τετραγωνικός προγραμματισμός μπορούν να δώσουν λύση εντός λογικού χρονικού διαστήματος. Αυτό το μοντέλο άλλωστε θα είναι και το βασικό αντικείμενο βελτιστοποίησης σε επόμενη ενότητα με εφαρμογή νέων πιο ευέλικτων μεθόδων βασισμένων σε Μεταευρετικούς αλγόριθμους.

1.5.3 The Tobin Model

Ο **Tobin (1958,1965)** εισήγαγε την έννοια του ακίνδυνου χρεογράφου σε ένα χαρτοφυλάκιο, ότι δηλαδή ένα χαρτοφυλάκιο δεν χρειάζεται να αποτελείται αποκλειστικά και μόνο από χρεόγραφα που εμπεριέχουν αβεβαιότητα. Ο επενδυτής μπορεί λοιπόν να επενδύσει σε ένα ακίνδυνο χρεόγραφο, όπου ως τέτοιο μπορεί να θεωρηθεί ένα έντοκο γραμματίο δημοσίου, και να μπορεί είτε να δανείσει με μια σίγουρη απόδοση r_f είτε να δανειστεί με επιτόκιο r_f . Εφόσον λοιπόν θεωρητικά υπάρχει αυτό το χρεόγραφο και μπορεί να συνδυαστεί με ένα **επικίνδυνο χαρτοφυλάκιο P**, προκύπτουν τα εξής δεδομένα.

Η αναμενόμενη απόδοση του αξιογράφου χωρίς κίνδυνο (**risk free asset**), είναι μια σίγουρη απόδοση με πιθανότητα 1 και είναι ίση με την μέση τιμή της, δηλαδή $E(r_f) = r_f$. Ο κίνδυνος ενός τέτοιου αξιογράφου όπως είναι λογικό είναι ίσος με μηδέν $\sigma_f^2 = 0$ ενώ ο συντελεστής συσχέτισης του με το επικίνδυνο χαρτοφυλάκιο είναι επίσης μηδέν $\rho_{fp} = 0$. Επίσης συμβολίζουμε με $E(r_p)$, σ_p^2 , x_p την αναμενόμενη απόδοση, τον κίνδυνο του επικίνδυνου χαρτοφυλακίου και το ποσοστό επένδυσης σε αυτό. Και τέλος όπως προκύπτει έχουμε $x_f = 1 - x_p$, το ποσοστό επένδυσης στο ακίνδυνο χρεόγραφο. Με βάση λοιπόν τα παραπάνω ο κίνδυνος σ_{fp}^2 όπως αυτός προκύπτει από τον συνδυασμό fp , δηλαδή του επικίνδυνου χαρτοφυλακίου με το ακίνδυνο χρεόγραφο θα είναι:

$$\begin{aligned}\sigma_{fp}^2 &= (1 - x_p)^2 \sigma_f^2 + x_p^2 \sigma_p^2 + 2x_p(1 - x_p)\rho_{fp}\sigma_p\sigma_f \\ &\stackrel{\sigma_f=0}{\implies} \sigma_{fp}^2 = x_p^2 \sigma_p^2 \\ &\implies \sigma_{fp} = x_p \sigma_p\end{aligned}\tag{1.8}$$

Βάσει της παραπάνω σχέσης επαληθεύεται και αυτό που φαινόταν ήδη λογικό, δηλαδή ότι ο κίνδυνος του χαρτοφυλακίου fp εξαρτάται μόνο από τον κίνδυνο του επικίνδυνου χαρτοφυλακίου και του ποσοστού επένδυσης σε αυτό. Επίσης από την ίδια σχέση προκύπτει ότι:

$$x_p = \sigma_{fp}/\sigma_p\tag{1.9}$$

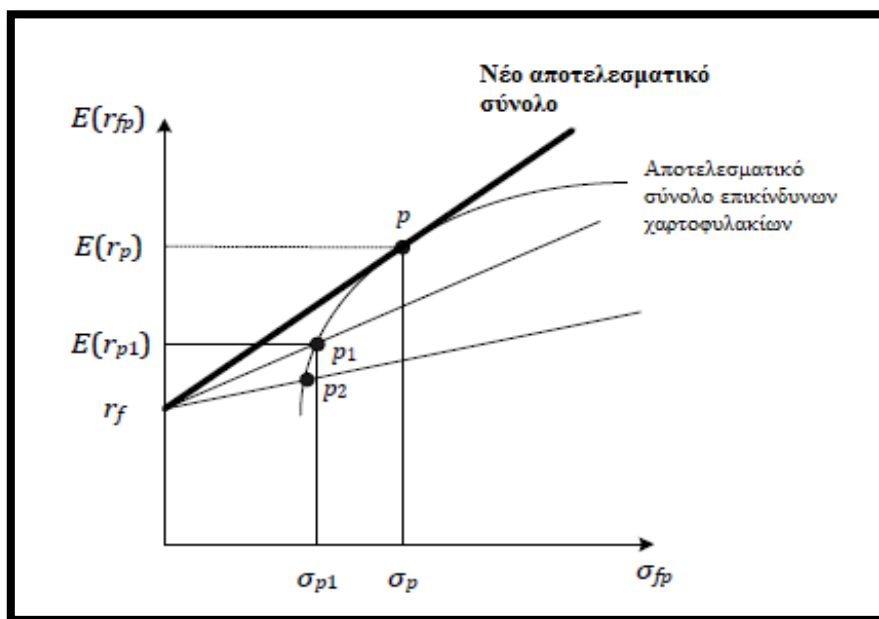
Αντίστοιχα η αναμενόμενη απόδοση $E(r_{fp})$ του χαρτοφυλακίου θα είναι:

$$E(r_{fp}) = (1 - x_p)r_f + x_p E(r_p)$$

Και χρησιμοποιώντας την σχέση (1.9) προκύπτει

$$E(r_{fp}) = r_f + \frac{E(r_p) - r_f}{\sigma_p} \sigma_{fp}$$

Θέτοντας $\theta_p = \frac{E(r_p) - r_f}{\sigma_p}$ και $\theta_0 = r_f$, η απόδοση θα πάρει την μορφή $E(r_{fp}) = \theta_0 + \theta_p \sigma_{fp}$, προκύπτει λοιπόν ότι η αποτελεσματική γραμμή του μοντέλου του Tobin είναι μια ευθεία γραμμή η οποία τέμνει τον άξονα των y στο σημείο $\theta_0 = r_f$ και έχει κλίση $\theta_p = \frac{E(r_p) - r_f}{\sigma_p}$, εν αντιθέσει με το μοντέλο του **Markowitz**, που το αποτελεσματικό σύνολο είναι καμπύλη γραμμή. Η σχέση λοιπόν της απόδοσης του χαρτοφυλακίου είναι γραμμική συνάρτηση του κινδύνου σ_{fp} όπως αυτός προκύπτει από τον συνδυασμό fp , δηλαδή του επικίνδυνου χαρτοφυλακίου με το ακίνδυνο χρεόγραφο. Το παρακάτω σχήμα συνδυάζει την αποτελεσματική γραμμή του **Tobin**, και το αποτελεσματικό σύνολο του **Markowitz**.



Διάγραμμα 1.3 Ακίνδυνο χρεόγραφο και αποτελεσματικό σύνολο του του Markowitz

Στο παραπάνω μοντέλο όλα τα χαρτοφυλάκια που μπορεί να συνθέσει ο εκάστοτε επενδυτής βρίσκονται πάνω στις ευθείες όπου τέμνουν τον άξονα των y στο σημείο $\theta_0 = r_f$. Ο συνδυασμός του ακίνδυνου χρεογράφου και κάποιου αποτελεσματικού επικίνδυνου χαρτοφυλακίου, για την σύνθεση του χαρτοφυλακίου fp , μπορεί να γίνει με οποιοδήποτε χαρτοφυλάκιο πάνω στο αποτελεσματικό σύνολο του **Markowitz** π.χ. P_1, P_2 , ουσιαστικά όμως η μόνη λογική επιλογή είναι το επικίνδυνο χαρτοφυλάκιο P στο οποίο εφάπτεται το αποτελεσματικό σύνολο του Tobin, αφού όλα τα χαρτοφυλάκια που βρίσκονται στο ευθύγραμμο τμήμα $r_f - P$, υπερτερούν έναντι όλων των άλλων π.χ. $r_f - P_1, r_f - P_2$, κλπ. καθότι σε δεδομένο επίπεδο κινδύνου αποδίδουν καλύτερα.

Για το πρόβλημα βελτιστοποίησης που θα κατασκευαστεί, τα δεδομένα μας είναι:

- το επικίνδυνο χαρτοφυλάκιο P που αποτελείται από n αξιόγραφα σταθμισμένα με τα αντίστοιχα βάρη x_i , και
- το ακίνδυνο χρεόγραφο r_f με ποσοστό συμμετοχής $x_f = 1 - \sum_{i=1}^n x_i$ όπως αυτό προκύπτει από τον περιορισμό $\sum_{i=1}^n x_i + x_f = 1$ με $x \in \mathbb{R}$ αφού το μοντέλο επιτρέπει τις ανοιχτές πωλήσεις και δεν έχει περιορισμό ως προς το θετικό πρόσημο των βαρών

Η αναμενόμενη λοιπόν απόδοση του χαρτοφυλακίου fp είναι:

$$E(r_{fp}) = x_f r_f + \sum_{i=1}^n x_i E(r_i)$$

και θέτοντας $x_f = 1 - \sum_{i=1}^n x_i$ προκύπτει ότι :

$$E(r_{fp}) = r_f + \sum_{i=1}^n x_i (E(r_i) - r_f)$$

και με μορφή πινάκων:

$$E(r_{fp}) = r_f + (\mathbf{r} - r_f \mathbf{e})^T \mathbf{x}$$

Το πρόβλημα λοιπόν βελτιστοποίησης διαμορφώνεται ως εξής:

$$\min \quad \sigma_{fp}^2 = \frac{1}{2} \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x}$$

$$\begin{aligned} \text{Περιορισμοί:} \quad & r_f + (\mathbf{r} - r_f \mathbf{e})^T \mathbf{x} = R^* \\ & \mathbf{x} \in \mathbb{R} \end{aligned}$$

Συμβολίζοντας ως λ τον **πολλαπλασιαστή Lagrange**, καθορίζεται η παρακάτω **Lagrangian** συνάρτηση,

$$L = \frac{1}{2} \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} + \lambda [R^* - r_f - (\mathbf{r} - r_f \mathbf{e})^T \mathbf{x}]$$

Στην συνέχεια οι μερικές παράγωγοι πρώτης τάξης της Lagrangian ως προς \mathbf{x} και ως προς τον **πολλαπλασιαστή Lagrange** τίθενται ίσοι με 0, και προκύπτει το παρακάτω σύστημα εξισώσεων.

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}} = 0 & \Rightarrow \boldsymbol{\Sigma} \mathbf{x} - \lambda (\mathbf{r} - r_f \mathbf{e}) = 0 \\ \stackrel{*}{\Rightarrow} \mathbf{x} & = \lambda \boldsymbol{\Sigma}^{-1} (\mathbf{r} - r_f \mathbf{e}) \quad (1.10) \\ \frac{\partial L}{\partial \lambda} = 0 & \Rightarrow R^* - r_f - (\mathbf{r} - r_f \mathbf{e})^T \mathbf{x} = 0 \end{aligned}$$

Το λ θα υπολογιστεί από την σχέση $R^* - r_f - (\mathbf{r} - r_f \mathbf{e})^T \mathbf{x} = 0$, όπου αντικαθιστώντας το \mathbf{x} από την σχέση (1.10), προκύπτουν τα εξής:

$$R^* - r_f = (\mathbf{r} - r_f \mathbf{e})^T \lambda \boldsymbol{\Sigma}^{-1}(\mathbf{r} - r_f \mathbf{e})$$

$$\lambda = \frac{R^* - r_f}{(\mathbf{r} - r_f \mathbf{e})^T \boldsymbol{\Sigma}^{-1}(\mathbf{r} - r_f \mathbf{e})}$$

Θέτοντας $c = (\mathbf{r} - r_f \mathbf{e})^T \boldsymbol{\Sigma}^{-1}(\mathbf{r} - r_f \mathbf{e})$ προκύπτει ότι:

$$\lambda = \frac{R^* - r_f}{c} \quad (1.11)$$

Από τις σχέσεις λοιπόν (1.10) και (1.11) προκύπτουν οι εξής κλειστοί τύποι υπολογισμού της βέλτιστης σύνθεσης \mathbf{x} του χαρτοφυλακίου fp .

$$\mathbf{x} = \lambda \boldsymbol{\Sigma}^{-1}(\mathbf{r} - r_f \mathbf{e}) \Rightarrow \mathbf{x} = \frac{R^* - r_f}{c} \boldsymbol{\Sigma}^{-1}(\mathbf{r} - r_f \mathbf{e})$$

Ο Κίνδυνος του χαρτοφυλακίου θα υπολογιστεί ως εξής:

$$\boldsymbol{\Sigma} \mathbf{x} - \lambda (\mathbf{r} - r_f \mathbf{e}) = 0 \Rightarrow \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} - \lambda \mathbf{x}^T (\mathbf{r} - r_f \mathbf{e}) = 0$$

Θέτοντας $\mathbf{x}^T (\mathbf{r} - r_f \mathbf{e}) = R^* - r_f$, και $\lambda = \frac{R^* - r_f}{c}$ προκύπτει ότι:

$$\mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x} = \lambda (R^* - r_f)$$

και τελικά

$$\sigma_{fp}^2 = \frac{(R^* - r_f)^2}{c}$$

Από την τελευταία σχέση για τον κίνδυνο προκύπτει και η γραμμική σχέση απόδοσης κινδύνου που αναφέρθηκε και αρχικά για το συγκεκριμένο μοντέλο δηλαδή $R^* = r_f + \sigma_{fp} \sqrt{c}$

Τέλος αναφέρθηκε ότι η κλίση της ευθείας του παραπάνω μοντέλου υπολογίζεται από τον τύπο $\theta_p = \frac{E(r_p) - r_f}{\sigma_p}$, και εκφράζει το λόγο της επιπρόσθετης απόδοσης προς τον κίνδυνο του επικίνδυνου χαρτοφυλακίου ή αλλιώς **Sharpe Ratio** του χαρτοφυλακίου. Προκύπτει λοιπόν λογικά ότι η βέλτιστη ευθεία που συνδέεται με το βέλτιστο χαρτοφυλάκιο του **Markowitz**, είναι αυτή που μεγιστοποιεί το Sharpe Ratio, δηλαδή αυτή με την μέγιστη κλίση η οποία ονομάζεται **Capital Allocation Line (CAL)** Το παραπάνω πρόβλημα λοιπόν βελτιστοποίησης μπορεί να λυθεί είτε με πολλαπλασιαστές Lagrange είτε με μερικές παραγώγους ενώ δίνει λογική λύση με δεδομένο ότι η απόδοση του ακίνδυνου χρεογράφου είναι χαμηλότερη από την απόδοση του χαρτοφυλακίου ελάχιστης διακύμανσης **MVP**, δηλαδή ισχύει η σχέση $r_f < r_{MVP}$ και διαμορφώνεται ως εξής:

$$\begin{array}{ll} \max & \theta_p = \frac{E(r_p) - r_f}{\sigma_p} \\ \text{Περιορισμοί:} & \mathbf{e}^T \mathbf{x} + x_f = 1 \\ & \mathbf{x} \in \mathbb{R} \end{array}$$

Ενώ στην περίπτωση απαγόρευσης των ανοιχτών πωλήσεων το πρόβλημα διαμορφώνεται ως εξής:

$$\begin{array}{ll} \max & \theta_p = \frac{E(r_p) - r_f}{\sigma_p} \\ \text{Περιορισμοί:} & \mathbf{e}^T \mathbf{x} + x_f = 1 \\ & \mathbf{x} \geq 0 \end{array}$$

Τα μοντέλα που αναλύθηκαν σε πολλές περιπτώσεις είδαμε ότι δίνουν λύσεις ακόμα και κλειστού τύπου. Σε περιπτώσεις όμως όπου το πρόβλημα έγκειται σε περιορισμούς (και αναλόγως και του πλήθους αυτών) δεν μπορεί να δοθεί αναλυτική λύση. Σε αυτό το πρόβλημα θα προσπαθήσουν να δώσουν λύση σε επόμενη ενότητα νέες μέθοδοι και τεχνικές βελτιστοποίησης βασισμένες στο Μεταερευρητικό προγραμματισμό.

Βασικές έννοιες Βελτιστοποίησης και Αλγόριθμοι

Στο κεφάλαιο αυτό γίνεται αναφορά στις πιο βασικές από τις κλασσικές μεθόδους βελτιστοποίησης, όπου επισημαίνονται ο τρόπος λειτουργίας τους και τα χαρακτηριστικά τους. Στη συνέχεια γίνεται μια συνοπτική εισαγωγή στους Μεταευρετικούς αλγόριθμους και πιο συγκεκριμένα στον **αλγόριθμο σμήνους σωματιδίων (PSO)**, στην **αποικία μυρμηγκιών (ACO)**, στην **περιορισμένη αναζήτηση (Tabu)**, στην **προσομοιωμένη ανόπτηση (SANN)** ενώ σε επόμενη ενότητα γίνεται εκτενέστερη ανάλυση στους **γενετικούς αλγόριθμους (GA)**.

2.1 Εισαγωγή στην Βελτιστοποίηση

Με τον όρο βελτιστοποίηση εννοούμε το πρόβλημα εκείνο κατά το οποίο αναζητούνται τα ακρότατα μια συνάρτησης. Τα πρόβλημα συνήθως έγκειται στην αναζήτηση κάποιου ολικού βέλτιστου μιας πολύπλοκης αντικειμενικής συνάρτησης, συνήθως ασυνεχής και μη παραγωγίσιμης με πληθώρα τοπικών ακροτάτων τα οποία δεν είναι επιθυμητά, εντός ενός καθορισμένου χώρου αναζήτησης. Η βελτιστοποίηση αυτή ονομάζεται ολική και η λύση της πολύ συχνά δύναται να ικανοποιεί κάποιες επιπλέον συνθήκες και περιορισμούς ώστε να είναι αποδεκτή.

Στις κλασσικές μεθόδους βελτιστοποίησης τα βασικά προβλήματα εστιάζονται στο χρόνο επίλυσης και στον εγκλωβισμό σε τοπικά βέλτιστα, προβλήματα στα οποία δίνει αξιόλογη λύση μια σχετικά νέα κατηγορία αλγορίθμων που ονομάζονται μεταευρετικοί (**Metaheuristics**) και οι οποίοι είναι ευέλικτοι, καταναλώνουν μικρή υπολογιστική ισχύ και διαθέτουν μηχανισμούς απεγκλωβισμού από τοπικά ακρότατα. Αυτό το καταφέρνουν καθώς όταν παγιδευτούν σε ένα τοπικό βέλτιστο, είτε συνεχίζουν την αναζήτηση δεχόμενοι μια υποδεέστερη λύση προκειμένου να απεγκλωβιστούν και να διευρύνουν τον χώρο αναζήτησης, είτε διαθέτουν μηχανισμούς με τους οποίους εναποθέτουν μια μικρή πιθανότητα ακόμα και στις πιο απομακρυσμένες περιοχές του χώρου αναζήτησης να εξερευνηθούν.

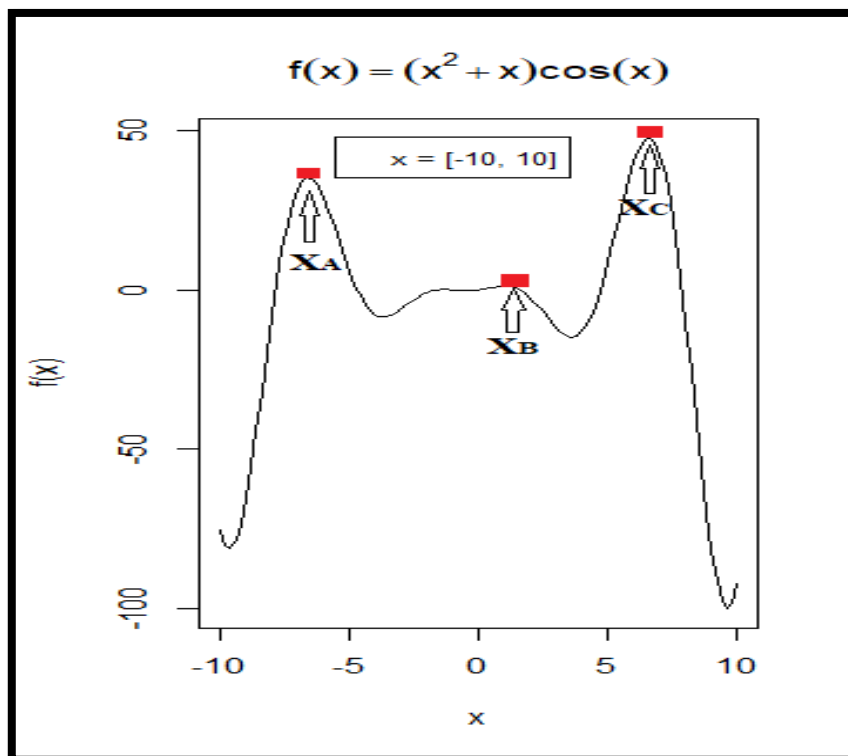
Στον όρο βελτιστοποίηση εμπεριέχονται δύο περιπτώσεις, αυτή της ελαχιστοποίησης και αυτή της μεγιστοποίησης, οι οποίες είναι ισοδύναμες καθότι το μέγιστο μια συνάρτησης f , αποτελεί ελάχιστο για την $-f$. Βάσει αυτής της ιδιότητας γίνεται αντιληπτό ότι ένα πρόβλημα μεγιστοποίησης εύκολα μπορεί να μετατραπεί σε πρόβλημα ελαχιστοποίησης και αντίστροφα. Γενικά ένα πρόβλημα βελτιστοποίησης (**εν προκειμένω ελαχιστοποίησης**) μπορεί να οριστεί ως εξής.

- Δοθείσης μιας αντικειμενικής συνάρτησης $f: S \rightarrow \mathbb{R}$
- Να βρεθεί $x^* \in S$, τέτοιο ώστε $f(x^*) \leq f(x), \forall x \in S$

Όπου για πρόβλημα χωρίς περιορισμούς (**unconstrained optimization problem**), ισχύει ότι $S = \mathbb{R}^n$, ενώ για πρόβλημα με περιορισμούς (**constrained optimization problem**) ισχύει $S \subseteq \mathbb{R}^n$.

Τυπικά για μια παραγωγίσιμη συνεχή συνάρτηση το βέλτιστο μπορεί να βρεθεί με την χρήση των πρώτων μερικών παραγώγων, οι οποίες εξισώνονται με το μηδέν $\partial y / \partial x = 0$. Το σημείο x^* ονομάζεται ολικός ελαχιστοποιητής (**global minimizer**) της f στο S , ενώ η τιμή της $f^* = f(x^*)$, καλείται ολικό ελάχιστο (**global minimum**) της f . Τέλος ένα σημείο $x' \in S$ ονομάζεται τοπικός ελαχιστοποιητής (**local minimizer**), όταν ισχύει $f(x') \leq f(x), \forall x \in S$, όπου $S = s \in S; d(x', x) \leq \varepsilon |_{\varepsilon \rightarrow 0}$, με d συμβολίζεται η απόσταση μεταξύ του x και του x' , η οποία είναι ίση με μια θετική τιμή ε . Η τιμή της $f' = f(x')$, καλείται τοπικό ελάχιστο (**local minimum**) της f .

Στο παρακάτω διάγραμμα που αφορά την απλή μονοδιάστατη συνάρτηση $f(x) = (x^2 + x) \cos(x), x \in [-10, +10]$ είναι εύκολο να αναγνωριστεί ότι τα x_A, x_B είναι τοπικοί βελτιστοποιητές με $f(x_A), f(x_B)$ αντίστοιχα **τοπικά βέλτιστα** (μέγιστα), ενώ το σημείο x_C είναι ολικός βελτιστοποιητής και $f(x_C)$ **ολικό βέλτιστο** (μέγιστο).



Διάγραμμα 2.1 Γράφημα συνάρτησης $f(x) = (x^2 + x) \cos(x), x \in [-10, +10]$

2.2 Τοπικά και Ολικά Βέλτιστα

Φυσικά δεν είναι πάντα μια συνάρτηση τόσο απλή και η διάκριση μεταξύ τοπικών και ολικών βέλτιστων τόσο ευδιάκριτη. Στην παραπάνω περίπτωση π.χ. ικανοποιείται η αναγκαία συνθήκη όπου η πρώτη παράγωγος μηδενίζει δηλαδή $f'(x_A) = 0$, ενώ ισχύει και ότι οι άμεσα γειτονικές κινήσεις είναι υποδεέστερες δηλαδή $f(x) \leq f(x_A \pm \varepsilon) \mid \varepsilon \rightarrow 0$, οπότε γίνεται αντιληπτό ότι ένας κλασσικός αλγόριθμος εύκολα θα μπορούσε να εγκλωβιστεί και να μην φτάσει στο σημείο x_C όπου βρίσκεται το ολικό βέλτιστο (μέγιστο), το φαινόμενο αυτό παρουσιάζεται ακόμα πιο έντονα σε περιπτώσεις με πιο πολύπλοκες συναρτήσεις που εμπεριέχουν πολλά τοπικά βέλτιστα.

Αυτό που συμβαίνει σε ένα πρόβλημα βελτιστοποίησης πολλές φορές είναι να απλοποιούνται οι συνθήκες του προβλήματος για να μπορεί να βρεθεί λύση. Το βασικότερο βήμα είναι η αναγνώριση του προβλήματος και η επιλογή της καλύτερης στρατηγικής επίλυσης. Μια ένδειξη της πολυπλοκότητας ενός προβλήματος είναι ο ασυμπτωτικός χρόνος που χρειάζεται ένας αλγόριθμος για να δώσει λύση. Σε ένα πολυωνυμικό πρόβλημα πρώτου βαθμού με n μεταβλητές ένας αλγόριθμος τάξης $O(n)$ θα καταναλώσει περίπου $c * n$ υπολογιστικό χρόνο (**CPU time**), όπου η σταθερά c είναι ενδεικτική της υπολογιστικής ισχύος και η επίδρασή της μειώνεται αισθητά όσο αυξάνεται η ισχύς του επεξεργαστή. Σε ένα πρόβλημα n μεταβλητών k τάξης για τον αλγόριθμο $O(n^k)$ ο χρόνος δεν αυξάνεται πλέον γραμμικά και ισούται με $c * n^k$, ενώ σε ένα συνδυαστικό πρόβλημα $\binom{n}{k}$, ο χρόνος που θα χρειαστεί ο αλγόριθμος $O\left(\binom{n}{k}\right)$ είναι τουλάχιστον $\frac{n!}{k!(n-k)!}$. Γίνεται λοιπόν αντιληπτό ότι όσο αυξάνεται το k ακόμα και στα σχετικά εύκολα P (**polynomial time**) προβλήματα, ο χρόνος μπορεί να αυξηθεί εκθετικά, ενώ στα NP (**non-deterministic polynomial time complete**) προβλήματα δεν υπάρχει γνωστός αναλυτικός αλγόριθμος να μπορεί να βρει λύση σε πολυωνυμικό χρόνο. Χαρακτηριστικά τέτοια παραδείγματα είναι τα γνωστά προβλήματα του περιοδεύοντος πωλητή (**Travelling Salesman Problem (TSP)**), όπου ένας πωλητής αναζητάει την συντομότερη διαδρομή ανάμεσα στις πόλεις που θέλει να επισκεφτεί από μία φορά, και να επιστρέψει στην αρχική του πόλη, όπου αν υπάρχει άμεση σύνδεση των πόλεων οι συνδυασμοί που προκύπτουν για n πόλεις είναι $n!$ Ενώ και στο πρόβλημα του σακιδίου, (**Knapsack Problem(KP)**), όπου σε ένα συγκεκριμένης χωρητικότητας σακίδιο πρέπει να χωρέσουν πολύτιμες πέτρες συγκεκριμένου μεγέθους μεγιστοποιώντας την τελική του αξία (Maringer, 2005). Γενικά το κλειδί της επιτυχίας ενός αλγορίθμου βελτιστοποίησης έγκειται στην ποιότητα αναζήτησης (**explorative quality**), για την εύρεση της βέλτιστης λύσης.

2.3 Κλασσικές τεχνικές Βελτιστοποίησης

Παρακάτω γίνεται μια συνοπτική αναφορά στις παραδοσιακές τεχνικές βελτιστοποίησης και οι οποίες είναι οι εξής.

- **Γραμμικός προγραμματισμός (Linear programming):** Με την ευρέως διαδεδομένη μέθοδο Simplex (**Simplex Algorithm**), η οποία μπορεί να εφαρμοστεί σε προβλήματα όπου και η αντικειμενική συνάρτηση, αλλά και οι περιορισμοί είναι γραμμικοί συνδυασμοί. Μπορεί να χρησιμοποιηθεί στην βελτιστοποίηση χαρτοφυλακίων είτε για την απόδοση είτε για την επιλογή επενδύσεων, αλλά όχι συνδυαστικά με τον κίνδυνο που είναι εξίσωση δευτέρου βαθμού.

- **Τετραγωνικός προγραμματισμός (Quadratic programming):** Εφαρμόζεται σε προβλήματα όπου η αντικειμενική συνάρτηση, και οι περιορισμοί μπορούν να είναι και δευτέρου βαθμού. Εφαρμόζεται ευρέως στην βελτιστοποίηση χαρτοφυλακίων με βάση το μοντέλο του Markowitz, όπου ουσιαστικά πάλι χρησιμοποιείται η μέθοδος Simplex, με την ενσωμάτωση των αναγκαίων συνθηκών των KKT.

- **Δυναμικός προγραμματισμός (Dynamic programming):** Εφαρμόζεται κυρίως σε σύνθετα πολυσταδιακά προβλήματα με μεγάλο αριθμό μεταβλητών απόφασης, όπου το συνολικό πρόβλημα διαιρείται σε n παρόμοια υποπροβλήματα με μικρότερο αριθμό μεταβλητών όπου μπορούν να επιλυθούν ευκολότερα, και οι λύσεις των οποίων συσχετίζονται λογικά. Είναι μέθοδος μερικής απαρίθμησης (**partial enumeration**), και η καλύτερη λύση σε κάθε στάδιο, συνδυαστικά αριστοποιεί την συνολική απόδοση του συστήματος.

- **Στοχαστικός προγραμματισμός (Stochastic programming):** Εφαρμόζεται σε προβλήματα, όπου στην αντικειμενική συνάρτηση εμπεριέχονται δεδομένα που χαρακτηρίζονται από αβεβαιότητα.

- **Άπληστοι Αλγόριθμοι (Greedy Algorithms):** Αλγόριθμοι που συναντώνται εδώ όπως του τυχαίου βηματισμού (**Random Walk**), και της αναρρίχησης λόφων (**Hill-Climbing**), ξεκινάνε από ένα τυχαία επιλεγμένο σημείο (random guess) και έχουν ως κοινό χαρακτηριστικό ότι μετακινούνται σε ένα νέο σημείο της περιοχής αναζήτησης μόνο αν αυτό βελτιώνει την τρέχουσα τιμή της αντικειμενικής συνάρτησης $f(X)$. Αυτό έχει ως αποτέλεσμα να παγιδεύονται εύκολα σε τοπικά βέλτιστα και να τερματίζεται η αναζήτηση προς άλλες καλύτερες λύσεις. Το μέγεθος της περιοχής αναζήτησης παίζει καθοριστικό ρόλο καθώς μεγάλη σε μέγεθος περιοχή βελτιώνει την πιθανότητα προσέγγισης της βέλτιστης λύσης, όμως στην περίπτωση κατά την οποία το αρχικό, τυχαίο και εφικτό σημείο από το οποίο ξεκινάει ο αλγόριθμος είναι κοντά στην βέλτιστη λύση, τότε ο αλγόριθμος μπορεί να απομακρυνθεί από την λύση αυτή.

- **Αναζήτηση κλίσης (Gradient Search):** Εφαρμόζεται σε αντικειμενικές συναρτήσεις παραγωγίσιμες και αυστηρώς κυρτές. Το βέλτιστο μπορεί να βρεθεί με την συνθήκη των πρώτων μερικών παραγώγων $\partial y / \partial x = 0$. Δοθέντος της τρέχουσας λύσης υπολογίζεται η κλίση $\nabla f(x') = (\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n})$ για $x' = x$. Η λύση υπολογίζεται ξανά για $x' = x' + \delta \nabla f(x)$. Οι υπολογισμοί επαναλαμβάνονται μέχρι να βρεθεί το βέλτιστο x^* με $\nabla f(x) = 0$.

Η όλη διαδικασία ουσιαστικά εξερευνάει το χώρο και με πολύ μικρές αλλαγές στο x , ψάχνει να βρει το σημείο που μηδενίζει η κλίση της εφαιπτομένης, και αν για μικρή αλλαγή στο x μετά από αυτό το σημείο η λύση για την αντικειμενική συνάρτηση είναι υποδεέστερη, τότε θεωρεί αυτό το σημείο ως το βέλτιστο.

2.4 Μεταεுρετικοί Αλγόριθμοι Αναζήτησης

Στην ενότητα αυτή θα παρουσιαστούν οι κυριότεροι μεταεுρετικοί αλγόριθμοι αναζήτησης οι οποίοι πλεονεκτούν σε σχέση με τους άπληστους εுρετικούς αλγόριθμους στο ότι έχουν σχεδιαστεί έτσι ώστε να μπορούν να αποφεύγουν να παγιδεύονται σε τοπικά βέλτιστα. Το ανταγωνιστικό τους πλεονέκτημα είναι η ευελιξία που επιδεικνύουν επιτρέποντας ακόμα και υποδεέστερες κινήσεις προκείμενου να οδηγηθούν μακροπρόθεσμα σε καλύτερη λύση. Σε γενικές γραμμές μια μεταεுρετική αναζήτηση τερματίζει βάση των παρακάτω κριτηρίων (Hamdy A. Taha,2012):

- Το πλήθος επαναλήψεων του αλγορίθμου ξεπερνάει ένα ορισμένο κατώφλι επαναλήψεων.
- Το πλήθος επαναλήψεων του αλγορίθμου από την τελευταία βέλτιστη λύση υπερβαίνει ένα συγκεκριμένο όριο επαναλήψεων χωρίς περαιτέρω βελτίωση της λύσης αυτής.
- Η περιοχή που σχετίζεται με το τρέχον σημείο αναζήτησης είναι είτε κενή είτε δεν μπορεί να οδηγήσει σε νέα βιώσιμη κίνηση αναζήτησης.
- Η τρέχουσα βέλτιστη λύση είναι αποδεκτή.

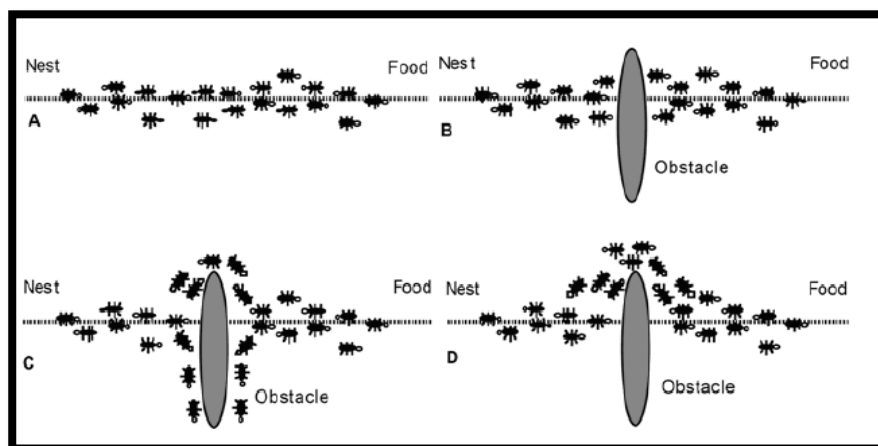
Σε αυτήν την ενότητα θα παρουσιαστούν συνοπτικά κάποιους από τους βασικότερους μεταεுρετικούς αλγόριθμους αναζήτησης όπως: το **σμήνος σωματιδίων (PSO)**, η **αποικία μυρμηγκιών (ACO)** η **περιορισμένη αναζήτηση (Tabu)** και η **προσομοιωμένη απόπτηση (SANN)**, ενώ αναλυτικότερα θα παρουσιαστούν στην επόμενη ενότητα οι **γενετικοί αλγόριθμοι(GA)**. Η βασική διαφοροποίηση μεταξύ των συγκεκριμένων αλγορίθμων είναι ο τρόπος με το οποίο καταφέρνουν να απεγκλωβίζονται από ένα τοπικό βέλτιστο, και ο τρόπος που εξερευνούν τον χώρο αναζήτησης (**exploration**), ενώ κοινό χαρακτηριστικό τους συνιστά η προοδευτική συρρίκνωση του χώρου αναζήτησης (**exploitation**) γύρω από την βέλτιστη λύση και την αποφυγή της απομάκρυνσης τους από αυτή.

2.5 Νοημοσύνη Σμηνών

Ο όρος Νοημοσύνη Σμηνών (ΝΣ) (**swarm intelligence**), χαρακτηρίζει τους αλγόριθμους που έχουν σαν πηγή έμπνευσης κάποια χαρακτηριστικά που εμφανίζονται σε κοινωνικές ομάδες στην φύση είτε πρόκειται για ομάδες εντόμων και ζώων είτε ακόμα και ομάδες ανθρώπων. Η συλλογική συμπεριφορά (**collective behavior**), και η εκδηλούμενη νοημοσύνη (**emergent intelligence**), που παρατηρούνται κατά την αλληλεπίδραση των οντοτήτων μέσα σε μια κοινωνική ομάδα σαν τις προαναφερθείσες ή σε ένα σμήνος, είναι αυτά τα οποία μιμείται ένας αλγόριθμος βελτιστοποίησης που βασίζεται στην νοημοσύνη σμηνών. Σε αυτή την ενότητα θα παρουσιαστούν δύο σημαντικοί αλγόριθμοι που βασίζονται στην νοημοσύνη των σμηνών, η αποικία μυρμηγκιών και το σμήνος σωματιδίων.

2.5.1 Αλγόριθμος αποικίας Μυρμηγκιών

Στην κατηγορία αυτή αλγορίθμων ανήκει ο αλγόριθμος Βελτιστοποίησης με αποικία Μυρμηγκιών (**BAM**, **Ant Colony Optimization**), ο οποίος αρχικά προτάθηκε από τον Ιταλό επιστήμονα Marco Dorigo το 1992, και ο οποίος μιμείται τον τρόπο με τον οποίο οι τερμίτες καταλήγουν να επιλέγουν την συντομότερη διαδρομή από την φωλιά τους (**Nest**), στην πηγή της τροφής (**Food**), αυτό το πετυχαίνουν αφήνοντας μια ουσία στις διαδρομές τους που ονομάζεται φερομόνη, και η οποία λειτουργεί σαν οδηγός προσανατολισμού για τους υπόλοιπους τερμίτες.



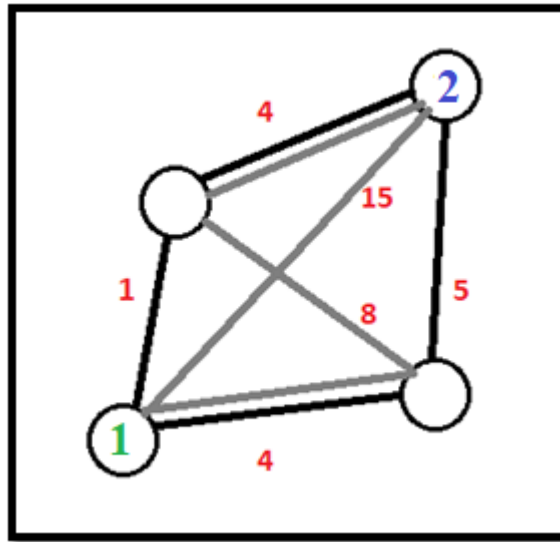
Πηγή: *International Journal of Trade, Economics and Finance*, Vol. 3, No. 2, April 2012

Εικόνα 2.2 Απεικόνιση των σχηματισμών των τερμιτών

Το πρώτο βήμα λοιπόν είναι ο υπολογισμός της φερομόνης, ο οποίος γίνεται με εφαρμογή της παρακάτω σχέσης.

$$\Delta\tau_{i,j}^k = \begin{cases} \frac{1}{L_k}, & \text{διαδρομή } i \text{ προς } j \\ 0, & \text{διαφορετικά} \end{cases}$$

Όπου $\Delta\tau_{i,j}^k$ η ποσότητα φερομόνης που εναποθέτει το k μυρμήγκι από τον κόμβο i στον κόμβο j , η οποία είναι 0 αν το μυρμήγκι δεν ολοκληρώσει τη διαδρομή από τον κόμβο i στον κόμβο j , και $\frac{1}{L_k}$ αν την ολοκληρώσει, όπου L_k το μήκος του μονοπατιού i, j . Χρησιμοποιώντας τον λόγο $\frac{1}{L_k}$ προκύπτει ότι η ποσότητα φερομόνης είναι αντιστρόφως ανάλογη με το μήκος του μονοπατιού, έτσι διασφαλίζεται ότι τα μικρότερα μονοπάτια θα έχουν μεγαλύτερη εναπόθεση φερομόνης και αντίστροφα.



Εικόνα 2.3 Σχεδιάγραμμα μήκους των μονοπατιών

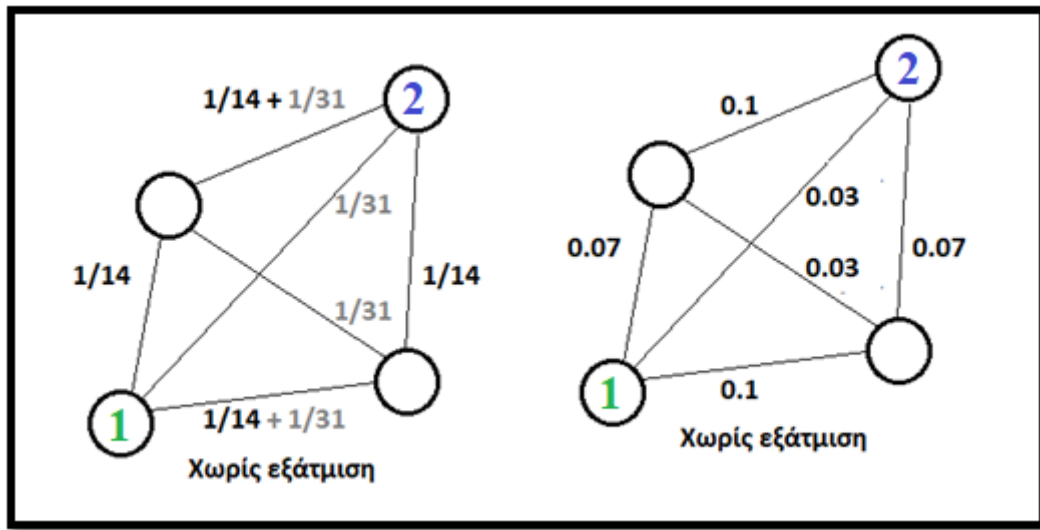
Έστω στο παραπάνω σχεδιάγραμμα L_1 η τυχαία μαύρη διαδρομή που ακολουθεί το μυρμήγκι N.1 και L_2 η τυχαία γκρι διαδρομή που ακολουθεί το μυρμήγκι N.2. Ξεκινώντας από την φωλιά (κόμβος 1) προς την τροφή (κόμβος 2), και επιστρέφοντας πίσω στην φωλιά. Εφαρμόζοντας λοιπόν τον παραπάνω τύπο:

Έχουμε $L_1 = 14 \rightarrow \Delta\tau_{i,j}^1 = \frac{1}{14}$, και $L_2 = 31 \rightarrow \Delta\tau_{i,j}^2 = \frac{1}{31}$. Ο αριθμός των μυρμηγκικών φυσικά που διασχίζει ένα μονοπάτι i, j είναι μεγάλος, και ο συνολικός αριθμός της φερομόνης που εναποτίθεται με το πέρασμα του χρόνου χωρίς να ληφθεί υπόψιν η διεργασία της εξάτμισης, δίνεται από τον τύπο:

$$\tau_{i,j}^m = \sum_{k=1}^m \Delta\tau_{i,j}^k$$

Όπου m ο συνολικός αριθμός των μυρμηγκιών.

Στη συνέχεια λοιπόν του παραδείγματος και βάσει του παραπάνω τύπου, υπολογίζεται η συνολική φερομένη που αφήνουν σε κάθε μονοπάτι και τα δύο μυρμήγκια.



Εικόνα 2.4 Σχεδιάγραμμα με την υπολογισμένη συνολική φερομένη χωρίς εξάτμιση.

Βάσει όσων έχουν ειπωθεί είναι λογική η υπόθεση ότι τα μονοπάτια με την μεγαλύτερη ποσότητα φερομένης η οποία είναι ίση με 0.1, έχουν και την μεγαλύτερη πιθανότητα να επιλεγτούν.

Η εξάτμιση είναι μια διεργασία του αλγορίθμου που ουσιαστικά προσομοιώνει την λειτουργία του φυσικού συστήματος όπου εξατμίζεται κάποια ποσότητα από τα ίχνη φερομένης με το πέρασμα του χρόνου π.χ. λόγω βροχής και στη συνέχεια εναποθέτεται ξανά από τα μυρμήγκια.

Για να προσομοιώσουμε την διαδικασία της εξάτμισης χρησιμοποιούμε τον παρακάτω μαθηματικό τύπο:

$$\tau_{i,j}^{t+1} = (1 - \rho)\tau_{i,j}^t + \sum_{k=1}^m \Delta\tau_{i,j}^k$$

Όπου $\tau_{i,j}^{t+1}$ η ποσότητα της φερομένης στην $t + 1$ επανάληψη του αλγορίθμου, $\tau_{i,j}^t$ η υφιστάμενη ποσότητα φερομένης, και $\sum_{k=1}^m \Delta\tau_{i,j}^k$ ο νέος ολικός αριθμός της φερομένης που εναποτίθεται μετά την εξάτμιση. Επεξηγώντας τον τύπο προκύπτει ότι η φερομένη που σχετίζεται με κάθε λύση εξατμίζεται (**evaporate**) με ρυθμό που καθορίζεται από τον συντελεστή $\rho \in [0,1]$. Ο ρυθμός εξάτμισης είναι αυτός που δημιουργεί και ισορροπία μεταξύ εξερευνησης (**exploration**) και εκμετάλλευσης (**exploitation**),

- Για $\rho = 0$, δεν υπάρχει εξάτμιση, ο τύπος παίρνει την μορφή

$$\tau_{i,j}^{t+1} = \tau_{i,j}^t + \sum_{k=1}^m \Delta\tau_{i,j}^k$$

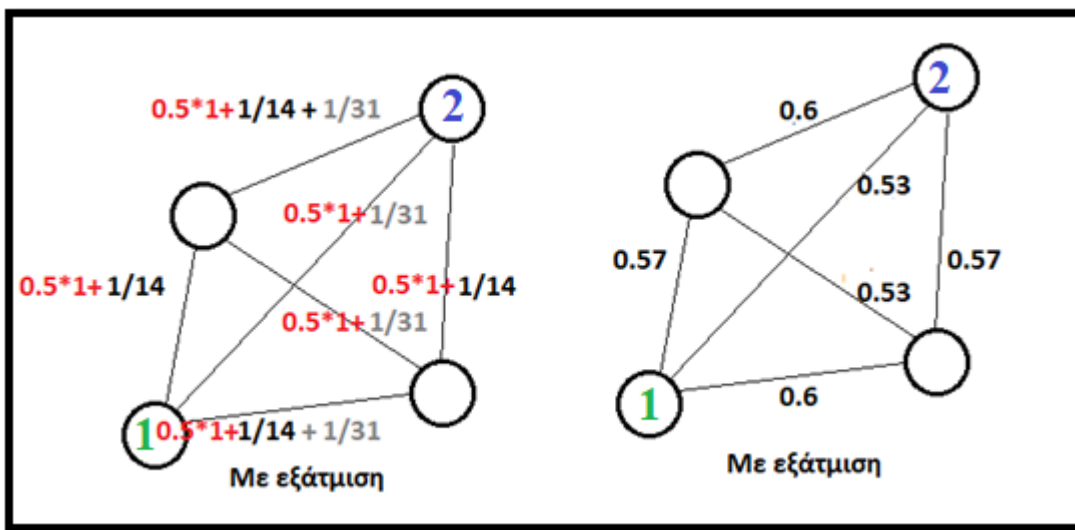
και ουσιαστικά προστίθεται στην υφιστάμενη ποσότητα φερομόνης η νέα ολική ποσότητα φερομόνης. Αυτό σημαίνει ότι λύσεις από προηγούμενες επαναλήψεις παίζουν καθοριστικό ρόλο. Δηλαδή στην επόμενη επανάληψη $t + 1$ του αλγόριθμου οι ποσότητες φερομόνης είναι υψηλά εξαρτώμενες από την ποσότητα φερομόνης της τρέχουσας λύσης κάτι που οδηγεί σε αναζήτηση γύρω από αυτήν την λύση και έτσι ενισχυεται η ικανότητα εκμετάλλευσης (**exploitation**) του αλγόριθμου.

- Αντίστοιχα για $\rho = 1$, ο ρυθμός εξάτμισης είναι στο μέγιστο βαθμό, ο τύπος παίρνει την μορφή

$$\tau_{i,j}^{t+1} = \sum_{k=1}^m \Delta\tau_{i,j}^k$$

και ουσιαστικά όλη η υφιστάμενη ποσότητα φερομόνης εξατμίζεται, έτσι διευρύνεται ο χώρος αναζήτησης και ενισχύεται η ικανότητα εξερεύνησης (**exploration**) του αλγόριθμου για νέες λύσεις.

Σε συνέχεια πάλι του παραδείγματος μας, έστω ότι έχουμε υφιστάμενη ποσότητα φερομόνης ίσης με 1.0 για διευκόλυνση των πράξεων, και ρυθμό εξάτμισης ρ ίσο με 0.5. βάσει του παραπάνω τύπου, υπολογίζεται η συνολική φερομόνη που αφήνουν σε κάθε μονοπάτι και τα δύο μυρμήγκια με ενσωματωμένη την διεργασία της εξάτμισης, όπως φαίνεται στα σχεδιαγράμματα που ακολουθούν.



Εικόνα 2.5 Σχεδιάγραμμα με υπολογισμένη την συνολική φερομόνη με εξάτμιση.

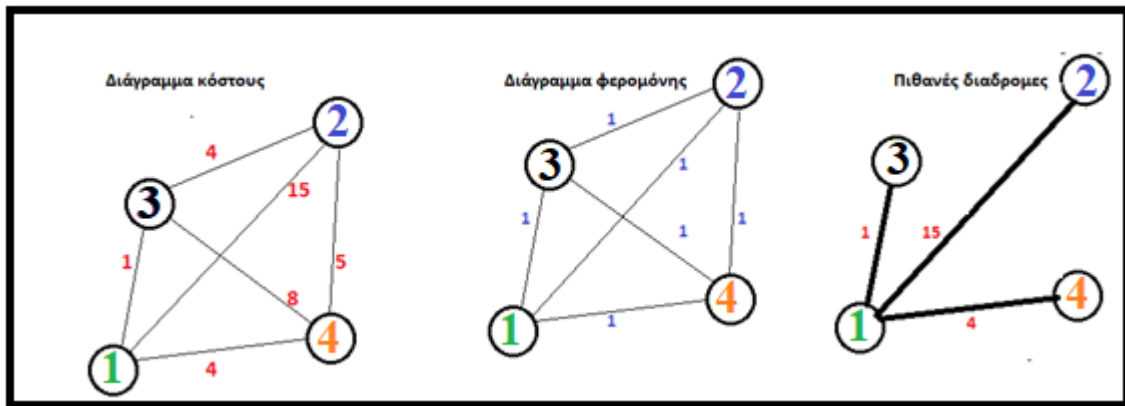
Παρατηρούμε ότι και μετά την διεργασία της εξάτμισης τα μονοπάτια που είχαν την υψηλότερη ποσότητα φερομόνης διατηρούν την ισχύ τους με αποτέλεσμα να μην χάνονται στην πορεία οι αρχικές καλές λύσεις. Είδαμε λοιπόν ότι οι διαδρομές με την υψηλότερη εναπόθεση φερομόνης ανά μονάδα χρόνου συμβαίνει να είναι και οι συντομότερες με αποτέλεσμα να έχουν την μεγαλύτερη πιθανότητα να επιλεγθούν καθώς οι τερμίτες προτιμούν την διαδρομές που είναι πιο πλούσιες σε αυτή. Ο αλγόριθμος ACO είναι ουσιαστικά ένα πιθανοθεωρητικό

μοντέλο που ονομάζεται μοντέλο φερομόνης (pheromone model) και αποτελείται από ένα διάνυσμα παραμέτρων T (pheromone trail parameters). Οι παράμετροι του διανύσματος παίρνουν τιμές $\tau_{i,j}$ (pheromone values) και σχετίζονται με την ποσότητα της φερομόνης της κάθε διαδρομής

Η πιθανότητα να επιλεγθεί το μονοπάτι i, j ανάμεσα σε άλλες δυνατές επιλογές δίνεται από τον τύπο:

$$prob_{i,j} = \frac{(\tau_{i,j})^\alpha (\eta_{i,j})^\beta}{\sum (\tau_{i,j})^\alpha (\eta_{i,j})^\beta}$$

Όπου η τιμή $\eta_{i,j} = \frac{1}{L_{i,j}}$ δείχνει την ποιότητα του μονοπατιού όπου όπως έχει αναφερθεί είναι συνυφασμένη με την απόσταση, και $\alpha \in [0,1]$, $\beta \in [0,1]$ οι παράμετροι για να καθοριστεί η επίδραση της φερομόνης $\tau_{i,j}$ και της ποιότητας του μονοπατιού $\eta_{i,j}$ αντίστοιχα. Αυτή η προσέγγιση διασφαλίζει ότι καλές διαδρομές (λύσεις) που είχαν επιλεγεί στο παρελθόν έχουν μεγάλη πιθανότητα να επιλεγθούν και πάλι, ενώ αφήνει και συγκεκριμένη πιθανότητα σε ανεξερεύνητα μονοπάτια να εξερευνηθούν και αν αποδειχθούν συντομότερα (καλύτερη λύση) μακροπρόθεσμα να επιλεγθούν εκείνα σαν πιο ελκυστική διαδρομή. Στο παράδειγμα που ακολουθεί φαίνεται ο τρόπος υπολογισμού της πιθανότητας επιλογής μονοπατιού. Θυμίζουμε ότι φωλιά είναι ο **κόμβος 1** και τροφή ο **κόμβος 2**.



Εικόνα 2.6 Διάγραμμα κόστους, φερομόνης και πιθανών διαδρομών.

Σαν δεδομένα έχουμε ίδιο επίπεδο φερομόνης και ίσο με την μονάδα για όλες τις διαδρομές (για διευκόλυνση των πράξεων), δηλαδή:

- $\tau_{1,2} = 1, \tau_{1,3} = 1, \tau_{1,4} = 1$

και επιπλέον:

- $\alpha = \beta = 1$ (για διευκόλυνση των πράξεων)

- $\eta_{1,2} = \frac{1}{15}, \eta_{1,3} = \frac{1}{1}, \eta_{1,4} = \frac{1}{4}$

Οι πιθανότητες λοιπόν υπολογίζονται ως εξής:

$$\bullet prob_{1,2} = \frac{1 \times \frac{1}{15}}{\left(1 \times \frac{1}{1}\right) + \left(1 \times \frac{1}{15}\right) + \left(1 \times \frac{1}{4}\right)} = 0.0506$$

$$\bullet prob_{1,3} = \frac{1 \times \frac{1}{1}}{\left(1 \times \frac{1}{1}\right) + \left(1 \times \frac{1}{15}\right) + \left(1 \times \frac{1}{4}\right)} = 0.7595$$

$$\bullet prob_{1,4} = \frac{1 \times \frac{1}{4}}{\left(1 \times \frac{1}{1}\right) + \left(1 \times \frac{1}{15}\right) + \left(1 \times \frac{1}{4}\right)} = 0.1899$$

Παρατηρούμε ότι η διαδρομή 1-3 είναι η πιθανότερη να επιλεγεί με πιθανότητα 76% και η 1-2 είναι αυτή με την μικρότερη πιθανότητα και ίση με 5%. Βάσει των παραπάνω με την μέθοδο **Roulette wheel** που θα επεξηγηθεί αναλυτικά και στην ενότητα των γενετικών αλγόριθμων, θεωρούμε ένα κυκλικό διάγραμμα (**pie chart**) όπου η πιθανότητα για την επιλογή του κάθε μονοπατιού καταλαμβάνει το ανάλογο κομμάτι (**pie**). Στη συνέχεια υπολογίζεται το προοδευτικό άθροισμα (**cumsum**) το οποίο είναι ίσο με:

$$(cumsum) = 1.00 \quad 0.24 \quad 0.05$$

- Και παράγοντας έναν αριθμό $R \sim U(0,1)$, αν προκύψει $0.24 < R \leq 1.00$, επιλέγεται η διαδρομή 1-3, για $0.05 < R \leq 0.24$, επιλέγεται η διαδρομή 1-4, και για $0.00 \leq R \leq 0.05$ επιλέγεται η διαδρομή 1-2.

Αναφέρθηκε προηγουμένως ότι σαν δεδομένα έχουμε ίδιο επίπεδο φερομόνης και ίσο με την μονάδα για όλες τις διαδρομές (για διευκόλυνση των πράξεων), τώρα λοιπόν για το ίδιο παράδειγμα θα αλλάξουμε στο μονοπάτι 1-4 την φερομόνη από 1.0 σε 5.0 για να δούμε την επίδρασή της στην πιθανότητα επιλογής διαδρομής. Ακολουθούν αναλυτικά τα αποτελέσματα.

$$\bullet prob_{1,2} = \frac{1 \times \frac{1}{15}}{\left(1 \times \frac{1}{1}\right) + \left(1 \times \frac{1}{15}\right) + \left(5 \times \frac{1}{4}\right)} = 0.0288$$

$$\bullet prob_{1,3} = \frac{1 \times \frac{1}{1}}{\left(1 \times \frac{1}{1}\right) + \left(1 \times \frac{1}{15}\right) + \left(5 \times \frac{1}{4}\right)} = 0.4317$$

$$\bullet prob_{1,4} = \frac{5 \times \frac{1}{4}}{\left(1 \times \frac{1}{1}\right) + \left(1 \times \frac{1}{15}\right) + \left(5 \times \frac{1}{4}\right)} = 0.5396$$

Παρατηρούμε τώρα ότι η διαδρομή 1-4 είναι πιθανότερο να επιλεγεί έναντι της διαδρομής 1-3 που ήταν η αρχικά η υψηλότερη, είναι λοιπόν εμφανής η επίδραση που έχει η ποσότητα φερομόνης στην λειτουργία του αλγόριθμου. Είναι αντιληπτό ότι με τους παραπάνω μηχανισμούς που αναλύθηκαν, ο αλγόριθμος είναι έτσι σχεδιασμένος ώστε οι καλές λύσεις να διατηρούν την μεγαλύτερη ποσότητα φερομόνης και τελικά να τερματίζεται είτε μετά από κάποιο συνολικά καθορισμένο αριθμό επαναλήψεων, είτε μετα από ενά αριθμό επαναληψεων όπου δεν παρατηρείται βελτίωση της τρέχουσας βέλτιστης λύσης. (π.χ. βλ. Haqiqi, Kazemi (2012)).

2.5.2 Βελτιστοποίηση με Σμήνος Σωματιδίων

Ο Αλγόριθμος της Βελτιστοποίησης με σμήνος σωματιδίων (ΒΣΣ),(**Particle Swarm Optimization, (PSO)**), είναι μια στοχαστική διαδικασία εμπνευσμένη από την συμπεριφορά κοινωνικών ομάδων όπως τα σμήνη πουλιών (π.χ.βλ. Panahian, Alizadeh, 2012). Ο αλγόριθμος αρχικοποιείται με έναν πληθυσμό (**swarm**) από τυχαίες λύσεις που ονομάζονται σωματίδια (**particles**). Η κίνηση του κάθε σωματιδίου γίνεται στον χώρο αναζήτησης με μια προσαρμοσσιμη ταχύτητα (**adaptable velocity**), και κάθε ένα διατηρεί στην μνήμη του την καλύτερη θέση που επισκέφτηκε, π.χ. σε ένα πρόβλημα μεγιστοποίησης θα είναι η θέση με την μεγαλύτερη συναρτησιακή τιμή. Επίσης τα μέλη του πληθυσμού ανταλλάζουν και πληροφορίες μεταξύ τους ως προς την καλύτερη θέση που έχουν επισκεφτεί, και προσαρμόζουν την ταχύτητα τους βάσει της δικιά τους εμπειρίας αλλά και της συνολικής εμπειρίας του σμήνους: Θεωρώντας λοιπόν ότι η περιοχή αναζήτησης είναι μια περιοχή διάστασης D , και ο συνολικός αριθμός των σωματιδίων του πληθυσμού είναι N ορίζονται τα κάτωθι:

- $f: S \rightarrow \mathbb{R}$, όπου $S \subseteq \mathbb{R}$, για πρόβλημα με περιορισμούς.
- $S = \{x_1, x_2, \dots, x_N\}$, το σμήνος (**swarm**) αποτελούμενο από N σωματίδια
- $x_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\}^T \in S$, $i = 1, 2, \dots, N$ η τρέχουσα θέση (**position**) του i σωματιδίου.
- $v_i = \{v_{i1}, v_{i2}, \dots, v_{iD}\}^T \in S$, $i = 1, 2, \dots, N$ η τρέχουσα ταχύτητα (**velocity**) του i σωματιδίου.
- $p_i = \{p_{i1}, p_{i2}, \dots, p_{iD}\}^T \in S$, $i = 1, 2, \dots, N$ η καλύτερη θέση του i σωματιδίου
- $p_g = \{p_{g1}, p_{g2}, \dots, p_{gD}\}^T \in S$, $g = 1, 2, \dots, N$ η καλύτερη θέση του σμήνους
- $t = 0$ μετρητής επαναλήψεων του αλγορίθμου

Βάσει των παραπάνω η ταχύτητα του κάθε σωματιδίου στην $(t + 1)$ επανάληψη του αλγορίθμου υπολογίζεται με την εφαρμογή της παρακάτω εξίσωσης.

$$v_{id}(t + 1) = v_{id}(t) + c_1 r_1 [p_{id} - x_{id}(t)] + c_2 r_2 [p_{gd} - x_{id}(t)] \quad (2.1)$$

Η διαφορά $[p_{id} - x_{id}(t)]$ πρακτικά εκφράζει την απόσταση του κάθε σωματιδίου από την καλύτερη υφιστάμενη ατομική λύση, και η διαφορά $[p_{gd} - x_{id}(t)]$ την απόσταση του κάθε σωματιδίου από την καλύτερη υφιστάμενη ολική λύση. Με r_1, r_2 , συμβολίζονται τυχαίοι αριθμοί από την ομοιόμορφη κατανομή $U \in [0, 1]$, οι οποίοι με στοχαστικό τρόπο αυξάνουν η μειώνουν την επίδραση αυτών των διαφορών. Τέλος οι c_1, c_2 εκφράζουν τον ατομικό και πληθυσμιακό ρυθμό εκμάθησης και καλούνται γνωστική (**cognitive**) και κοινωνική (**social**) αντίστοιχα, με αυτές μπορεί και ρυθμίζεται η ισορροπία μεταξύ ικανότητας εξερεύνησης (**exploration**) και ικανότητας εκμετάλλευσης (**exploitation**) του αλγορίθμου.

Από τη στιγμή που υπολογίζεται η ταχύτητα του i σωματιδίου στην $(t + 1)$ επανάληψη του αλγόριθμου υπολογίζεται και η νέα θέση του σωματιδίου με την εφαρμογή της παρακάτω εξίσωσης.

$$x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1) \quad (2.2)$$

Η οποία λέει ότι η θέση του i σωματιδίου στην $(t + 1)$ επανάληψη του αλγόριθμου, ισούται με την τρέχουσα θέση του συν την ταχύτητα του στην $(t + 1)$ επανάληψη.

Στην παραπάνω έκδοση του αλγορίθμου παρατηρήθηκε ότι οι ταχύτητες των σωματιδίων μπορούσαν να πάρουν γρήγορα μεγάλες τιμές οδηγώντας τον αλγόριθμο να συγκλίνει σε μία μη ολικά βέλτιστη λύση με αποτέλεσμα την μειωμένη του απόδοση. Έτσι κρίθηκε απαραίτητη η προσθήκη μίας επιπλέον παραμέτρου για τον έλεγχο της ταχύτητας των σωματιδίων, η οποία συμβολίζεται με w και ονομάζεται βάρος αδράνειας (**inertia weight**). Αυτή η παράμετρος ρυθμίζει την ισορροπία μεταξύ εξερεύνησης (**exploration**) και εκμετάλλευσης (**exploitation**) του αλγόριθμου. Κατόπιν λοιπόν των νέων προσθηκών η ταχύτητα του κάθε σωματιδίου και το βάρος αδράνειας το οποίο μειώνεται γραμμικά με τον αριθμό επαναλήψεων, συνήθως με εύρος τιμών από 0.9 σε 0.4 δίνονται από τις παρακάτω σχέσεις:

$$v_{id}(t + 1) = wv_{id}(t) + c_1r_1[p_{id} - x_{id}(t)] + c_2r_2[p_{gd} - x_{id}(t)] \quad (2.3)$$

$$w_i = w_{max} - \left(\frac{w_{max} - w_{min}}{t_{max}} \right) t \quad (2.4)$$

Όπου w_{min} η αρχική τιμή του βάρους αδράνειας και w_{max} η τελική, t ο μετρητής επαναλήψεων του αλγορίθμου, και t_{max} ο μέγιστος αριθμός επαναλήψεων που εκτέλεσε ο αλγόριθμος PSO.

2.5.3 Λειτουργία αλγόριθμου Σμήνους Σωματιδίων

- **Φάση 0.** Εισαγωγή αρχικών παραμέτρων $(N, c_1, c_2, w_{min}, w_{max}, t_{max})$.

- **Φάση 1.** Αρχικοποίηση του πληθυσμού N του σμήνους.

- **Φάση 2.** Υπολογισμός της objective function για κάθε σωματίδιο

Αναβάθμιση της υφιστάμενης καλύτερης ατομικής λύσης p_{id} , αν είναι απαραίτητο.

Αναβάθμιση της υφιστάμενης καλύτερης ολική λύσης p_{gd} , αν είναι απαραίτητο.

- **Φάση 3.** Ενημέρωση της τιμή του βάρους αδράνειας w . (Εύρος 0.9 – 0.4)

Αναβάθμιση της τρέχουσας θέση (**position**) x_i του i σωματιδίου

Αναβάθμιση της τρέχουσας ταχύτητας (**velocity**) v_i του i σωματιδίου.

- **Φαση 4.** Εφόσον ικανοποιηθεί κάποιο από τα κριτήρια τερματισμού, ο αλγόριθμος σταματάει και επιστρέφει σαν ολικό βέλτιστο την τελευταία ολική λύσης p_{gd} , αλλιώς θέτει $t = t + 1$ και επιστρέφει στην Φάση 2.

2.5.4 Επίδραση των παραμέτρων του αλγόριθμου Σμήνους Σωματιδίων

Όπως ειπώθηκε το βάρος αδράνειας w αποτελεί τον μηχανισμό με τον οποίο ο αλγόριθμος ρυθμίζει την ισορροπία μεταξύ **εξερεύνησης** και **εκμετάλλευσης**. Στην πρώτη περίπτωση λοιπόν κρατώντας σταθερές τις τιμές των παραμέτρων c_1, c_2 , ξεχωρίζουν οι εξής περιπτώσεις.

- Για $w = 0$, η αναζήτηση (**exploration**), είναι στον ελάχιστο βαθμό, ενώ είναι στον μέγιστο βαθμό για $w = 1$.
- Για $w = 0.9 - 0.4$ έχει εμπειρικά επικρατήσει ότι πετυχαίνετε η καλύτερη ισορροπία μεταξύ εξερεύνησης (**exploration**) και εκμετάλλευσης (**exploitation**).

Στην δεύτερη περίπτωση κρατώντας σταθερές τις τιμές της παραμέτρου w , ξεχωρίζουν οι εξής περιπτώσεις.

- Για $c_1 = 0$, που σημαίνει ότι κάθε σωματίδιο ακολουθεί μόνο την καλύτερη υφιστάμενη ολική λύση, δηλαδή δεν έχει ουσιαστικά ατομική πληροφόρηση, παρά μόνο την εμπειρία του σμήνους, η ικανότητας εξερεύνησης (**exploration**) είναι στον χαμηλότερο βαθμό, σε αντίθεση με την ικανότητα εκμετάλλευσης (**exploitation**) που είναι στον μέγιστο, για αυτό και ο αλγόριθμος συγκλίνει γρήγορα προς μία βέλτιστη λύση που βάσει της εμπειρίας του σμήνους θεωρείται το ολικό βέλτιστο.
- Αντίθετα για $c_2 = 0$, που σημαίνει ότι κάθε σωματίδιο έχει ουσιαστικά μόνο ατομική πληροφόρηση, και δεν υπάρχει η εμπειρία του σμήνους, η ικανότητας εξερεύνησης (**exploration**) είναι στο μέγιστο βαθμό, σε αντίθεση με την ικανότητα εκμετάλλευσης (**exploitation**) που είναι στον ελάχιστο, με αποτέλεσμα ο αλγόριθμος να μην μπορεί να συγκλίνει σε κάποια λύση.

2.6 Αλγόριθμος Περιορισμένης Αναζήτησης Tabu

Η αναζήτηση Tabu (TS) η οποία είναι τοπική αναζήτηση αναπτύχθηκε από τον Fred W. Glover το 1986. Προκειμένου να μην παγιδεύεται σε ένα τοπικό βέλτιστο και να μπορεί να συγκλίνει σε μια ολικά βέλτιστη λύση, επιλέγει την επόμενη κίνηση της η οποία πιθανό να είναι και υποδεέστερη μπλοκάροντας την επαναξιολόγηση προηγούμενων λύσεων. Αυτό το

καταφέρνει έχοντας καταγράψει σε μια λίστα (**tabu list**), τις προηγούμενες και τις υποδεέστερες κινήσεις της, τις οποίες έτσι μπορεί και τις ‘θυμάται’, και με αυτόν τον τρόπο μπορεί και τις μπλοκάρει εντός μιας καθορισμένης **περιόδου θητείας**. Με την ολοκλήρωση της περιόδου αυτής μια κίνηση tabu διαγράφεται από την λίστα και καθίσταται και πάλι ενεργή για το μέλλον. Τα επιμέρους στοιχεία ενός αλγόριθμου TS είναι τα κάτωθι (Hamdy A. Taha,2012):

- $F(x) = H$ αντικειμενική συνάρτηση προς βελτιστοποίηση
- $k =$ Αριθμός επανάληψης
- $x_k =$ Τρέχουσα δοκιμαστική λύση
- $N(x_k) =$ Περιοχή αναζήτησης του x_k
- $R_k =$ Τυχαίος αριθμός $(0,1)$ επιλογής της τυχαίας κίνησης x_{k+1} από την $N(x_k)$
- $L_{(k)} =$ Tabu λίστα με τις απαγορευμένες τιμές του x κατά την επανάληψη k
- $r =$ Πλήθος διαδοχικών επαναλήψεων Tabu, που αντιπροσωπεύουν την θητεία.
- $x^* =$ Η βέλτιστη λύση κατά την διάρκεια της αναζήτησης.

2.6.1 Λειτουργία αλγορίθμου αναζήτησης Tabu

Αρχικά ορίζουμε το σύνολο της περιοχής αναζήτησης του x_k , με τέτοιο τρόπο ώστε να είναι αυστηρώς αποκλεισμένα τα σημεία που ανήκουν στις μη εφικτές λύσεις. Ο ορισμός μπορεί να είναι της μορφής $N(x_k) = \{x_k - q, \dots, x_k - 1, x_k + 1, \dots, x_k + q\} - L_{(k)}$ όπου q μια ακέραια σταθερά και $L_{(k)}$ η λίστα Tabu με τις απαγορευμένες τιμές x_k . Επίσης ορίζεται ο αριθμός r , δηλαδή οι διαδοχικές επαναλήψεις, μετά την παρέλευση των οποίων θα απορρίπτεται ένα στοιχείο από την λίστα Tabu με βάση την πολιτική πρώτο μέσα-πρώτο έξω (FIFO).

• **Φάση 0.** Για $k = 0$, επιλέγεται τυχαία μια αρχική λύση $x_0 \in S$, αρχικοποιείται η λίστα Tabu $L_{(0)} = \emptyset$, και καθορίζεται το μέγεθός της.

• **Φάση 1.** Προσδιορίζεται η εφικτή περιοχή $N(x_k)$.

• **Φάση 2.** Επιλέγεται είτε τυχαία (επιλογή τυχαίου βηματισμού) η επόμενη κίνηση x_{k+1} , από το σύνολο $N(x_k)$, ενώ δύναται να επιλεγεί ακόμα και από την λίστα $L_{(k)}$ αν μπορεί να προσφέρει καλύτερη λύση, ταυτόχρονα ενημερώνεται και η λίστα tabu $L_{(k+1)}$.

• **Φάση 3.** Εφόσον ικανοποιηθεί κάποιο από τα κριτήρια τερματισμού, ο αλγόριθμος σταματάει αλλιώς θέτει $k = k + 1$ και επιστρέφει στην Φάση 1.

Ο αλγόριθμος TS είναι σχεδιασμένος όπως έχει ήδη ειπωθεί να μην επιτρέπει κινήσεις που βρίσκονται στην λίστα tabu, μόνη εξαίρεση στο να εξεταστούν οι μη επιτρεπόμενες κινήσεις της λίστας, είναι ότι αυτές μπορεί να οδηγήσουν σε μία βελτιωμένη λύση, εάν το κάνουν θα πρέπει να γίνουν αποδεκτές σαν κίνηση αναζήτησης. Αυτή η βελτίωση ονομάζεται **Κριτήριο Προσδοκίας**. Επίσης ελέγχοντας το μέγεθος της λίστας μπορούμε να υλοποιήσουμε τις στρατηγικές της **Εντατικοποίησης (exploitation)** και της **Διαφοροποίησης (exploration)**. Η εντατικοποίηση συνεπάγεται μικρή λίστα, το οποίο έχει σαν αποτέλεσμα την αύξηση του συνόλου της εφικτής περιοχής εντείνοντας την αναζήτηση γύρω από την άριστη λύση. Ενώ η διαφοροποίηση συνεπάγεται μεγάλη λίστα, και βοηθάει στον απεγκλωβισμό του αλγόριθμου από κάποιο τοπικό βέλτιστο αφού αφήνει μια μικρή πιθανότητα εξερεύνησης ακόμα και των πιο ‘απομακρυσμένων περιοχών’ (Hamdy A. Taha,2012). Οι παραπάνω ρυθμίσεις έχουν φυσικά σαν στόχο την ποιοτική βελτίωση της λύσης.

2.7 Αλγόριθμος Προσομοιωμένης Ανόπτωσης (SANN)

Η έμπνευση για τον αλγόριθμο της προσομοιωμένης ανόπτωσης (Simulated Annealing – SANN), προήλθε από την διαδικασία ανόπτωσης στην μεταλλουργία, όπου η θέρμανση και η ψύξη του μετάλλου είναι απαραίτητες διεργασίες για την σκλήρυνσή του. Η υιοθέτηση της διαλέκτου από την μεταλλουργία, με λέξεις όπως θερμοκρασία είναι για καθαρά παραδοσιακούς σκοπούς χωρίς να έχουν σχέση με την λειτουργία του αλγορίθμου.

Στο λειτουργικό κομμάτι του αλγορίθμου αυτό που πρέπει να επισημανθεί αρχικά, είναι ο τρόπος που καταφέρνει να απεγκλωβιστεί από τοπικά βέλτιστα. Για να το επιτύχει αυτό χρησιμοποιεί μια συνθήκη, όπου αποδέχεται με πιθανότητα 1 μια βελτιωμένη επόμενη κίνηση, ενώ αποδέχεται ή απορρίπτει μια υποδέεστερη επόμενη κίνηση βάσει μιας πιθανότητας, η οποία καθορίζεται με έναν συγκεκριμένο τρόπο. Χρησιμοποιώντας μια παράμετρο T , που ονομάζεται θερμοκρασία και η οποία παρουσιάζει μια προοδευτική πτώση βάσει μιας σταθεράς r_i δημιουργείται ένα **χρονοδιάγραμμα θερμοκρασίας** το οποίο αποτελείται από I στοιχεία και ορίζεται ως $\{T = T_i, i = 1, 2, \dots, I\}$. Κάθε T_i εφαρμόζεται για συγκεκριμένο αριθμό επαναλήψεων-αποδοχών. Δοθέντος σαν αρχική στρατηγική αναζήτησης το s_0 , το T_0 υπολογίζεται ως εξής:

$$\bullet T_0 = r_0 F(s_0), \quad 0 < r_0 < 1$$

$$\bullet T_i = r_i T_{i-1}, \quad 0 < r_i < 1, i = 1, 2, \dots, I$$

Ορίζουμε ως s_α την τελευταία αποδεκτή στρατηγική επίλυσης. Κατά την επανάληψη k , η πιθανοτική σχέση αποδοχής μιας στρατηγικής περιοχής της επόμενη κίνησης αναζήτησης s_{k+1} , δίνεται παρακάτω: (Hamdy A. Taha,2012)

$$\bullet P\{\text{αποδοχή } s_{k+1} | s_{k+1} \in N(s_k)\} = \begin{cases} 1, \text{ αν } F(s_{k+1}) \text{ δεν είναι χειρότερο από το } F(s_\alpha) \\ e^{-\frac{|F(s)-F(s_{k+1})|}{T}} = e^{-|Δ|/T}, \text{ διαφορετικά} \end{cases}$$

Από την ανωτέρω σχέση γίνεται αντιληπτό, ότι η επόμενη κίνηση αναζήτησης s_{k+1} γίνεται αποδεκτή με πιθανότητα 1 αν η $F(s_{k+1})$ δεν είναι χειρότερη από την $F(s_\alpha)$, διαφορετικά αν η $F(s_{k+1})$, είναι μια υποδεέστερη λύση, γίνεται αποδεκτή μόνο αν ισχύει η σχέση:

- $R_k \leq e^{-|d|/T}$ όπου: $R_k =$ Τυχαίος αριθμός $(0,1)$

Αν προκύψει ότι $R_k \geq e^{-|d|/T}$ και δεν γίνει αποδεκτή η νέα λύση απαιτείται επαναδειγματοληψία από την τελευταία αποδεκτή περιοχή αναζήτησης.

Τα επιμέρους στοιχεία ενός αλγόριθμου SANN είναι τα κάτωθι

- $F(s)$ = Η αντικειμενική συνάρτηση προς βελτιστοποίηση
- k = Αριθμός επανάληψης του αλγορίθμου
- s_k = Τρέχουσα δοκιμαστική λύση
- s_α = Τελευταία αποδεκτή στρατηγική επίλυσης
- $N(s_k)$ = Περιοχή αναζήτησης του s_k
- T = Θερμοκρασία
- t = Αποδοχές λύσεων στο πέρας των οποίων θα μειωθεί η θερμοκρασία
- p = Μετρητής αποδοχών – επαναλήψεων
- r_i = Σταθερά χρονοδιαγράμματος θερμοκρασίας.

Αυτό που προκύπτει από την τελευταία σχέση, είναι ότι ο αλγόριθμος δίνει την δυνατότητα αποδοχής μια υποδεέστερης λύσης διευρύνοντας τον χώρο και την ικανότητα αναζήτησης (**exploration**) όμως ταυτόχρονα η πιθανότητα να γίνει αποδεκτή μια υποδεέστερη λύση μειώνεται με την προοδευτική μείωση της θερμοκρασίας T . Η λειτουργία αυτή του αλγορίθμου βασίζεται στην λογική της προοδευτικής συρρίκνωσης του χώρου αναζήτησης, και της σύγκλισης γύρω από την βέλτιστη λύση, το οποίο ονομάζεται ικανότητα εκμετάλλευσης (**exploitation**).

2.7.1 Λειτουργία αλγορίθμου Προσομοιωμένης Ανόπτησης (SANN)

- **Φάση 0.** Για $k = 0$, $p = 0$ και $i = 0$ επιλέγεται τυχαία μια αρχική λύση $s_0 \in S$
- **Φάση 1.** Προσδιορίζεται η εφικτή περιοχή $N(s_k)$, ορίζεται θερμοκρασία $T = T_i$ και η σταθερά r_0 .

• **Φάση 2.** Επιλέγεται τυχαία (random) η επόμενη κίνηση s_{k+1} , από το σύνολο $N(s_k)$, εάν η $F(s_{k+1})$ δεν είναι χειρότερη από την τελευταία αποδεκτή λύση $F(s_\alpha)$, ή αν ισχύει $R < P \{αποδοχή s_{k+1}\}$, τότε γίνεται αποδεκτό το s_{k+1} , θέτουμε $p = p + 1$ και προχωράμε στην Φάση 3. Σε αντίθετη περίπτωση θέτουμε $N(s_{k+1}) = N(s_k)$, $k = k + 1$, και ο αλγόριθμος επιστρέφει στην Φάση 1.

• **Φάση 3.** Εφόσον ικανοποιηθεί κάποιο από τα κριτήρια τερματισμού, ο αλγόριθμος σταματάει αλλιώς θέτει $k = k + 1$. Εάν $p = t$ θέτουμε $i = i + 1$ και επιστρέφουμε στην Φάση 1 (Hamdy A. Taha, 2012).

Μετά από αυτήν την συνοπτική εισαγωγή για τους μεταερευτικούς αλγόριθμους βελτιστοποίησης, στην επόμενη ενότητα θα αναπτυχθεί αναλυτικότερα άλλη μία κατηγορία αλγορίθμων του μεταερευτικού προγραμματισμού, αυτή των γενετικών αλγορίθμων που είναι και το βασικό μας αντικείμενο, με εκτενή ανάλυση σε θεωρητικό και πρακτικό επίπεδο, όπου με απλές εφαρμογές θα αναδειχτούν οι λειτουργίες και οι δυνατότητές τους.

Γενετικοί Αλγόριθμοι (GA)

Στο κεφάλαιο αυτό αναλύονται εκτενώς οι **Γενετικοί αλγόριθμοι (GA)**, γίνεται επεξήγηση του θεωρητικού υπόβαθρου στο οποίο βασίζονται, ενώ με την υλοποίηση κωδικά και την εφαρμογή τους σε παραδείγματα απλών συναρτήσεων αναλύονται πλήρως οι μηχανισμοί και η λειτουργίες τους.

3.1 Γενετικοί Αλγόριθμοι

Οι Γενετικοί Αλγόριθμοι (GA) είναι στοχαστικοί αλγόριθμοι αναζήτησης οι οποίοι αναπτύχθηκαν από τον John Holland το 1970. Χρησιμοποιούνται ευρέως λόγω της απλότητας, της ταχύτητας και της αποδεδειγμένης ικανότητάς τους να πετυχαίνουν εκεί που οι άλλες εφαρμοζόμενες μέθοδοι αποτυγχάνουν. Εφαρμόζονται σε προβλήματα βελτιστοποίησης συνεχών (παραγωγίσιμων και μη) και διακριτών συναρτήσεων, με ή χωρίς την ενσωμάτωση περιορισμών. Είναι εμπνευσμένοι από τον βιολογικό μηχανισμό που χρησιμοποιεί η φύση για την εξέλιξη των γονιδίων και εντέλει των ειδών. Η φυσική αυτή εξελικτική διαδικασία βασίζεται στην ιδέα ότι συνδυάζοντας χρωμοσώματα μέσω της επιλογής και της διασταύρωσης, αλλά και την επίδραση άλλων στοχαστικών διαδικασιών στο γονιδίωμα όπως η μετάλλαξη μπορούν να προκύψουν απόγονοι βελτιωμένοι σε σχέση με την αρχική γενιά, με αποτέλεσμα την επικράτηση των ισχυρότερων εξ αυτών.

Σε ένα πρόβλημα βελτιστοποίησης κάθε εφικτή λύση που προκύπτει θεωρείται σαν **χρωμόσωμα**, ο όρος εφικτή αναφέρεται στην λύση εκείνη η οποία δεν παραβιάζει τυχόν περιορισμούς και αναγκαίες συνθήκες του προβλήματος. Το χρωμόσωμα ως γνωστό και από την Βιολογία κωδικοποιείται από ένα συνδυασμό **γονιδίων**. Ο κώδικας των γονιδίων μπορεί να αναπαρασταθεί με δύο τρόπους σαν μεταβλητή απόφασης, είτε με τις πραγματικές τιμές της μεταβλητής (**αριθμητικός**), είτε με βάση το δυαδικό σύστημα απαρίθμησης (**δυαδικός**). Για παράδειγμα μια μεταβλητή με διάλυση εφικτών τιμών $X = (0,1,2,3,4)$ μπορεί να αναπαρασταθεί, αριθμητικά σαν $\{0,1,2,3,4\}$ και δυαδικά σαν $\{0000,1000,0100,1100,0010\}$. Έχει επικρατήσει ότι σε αριθμητικά προβλήματα βελτιστοποίησης είναι καλύτερο να χρησιμοποιείται η αριθμητική αντί της δυαδικής αναπαράστασης.

Με τον όρο **πληθυσμός N** χρωμοσωμάτων εννοείται το σύνολο των N εφικτών λύσεων. Κάθε λύση δοκιμάζεται σε κάθε επανάληψη του αλγορίθμου για την ικανότητα της μέσω μιας διαδικασίας **αξιολόγησης φυσικής κατάστασης (fitness evaluation)**, βάσει μιας συνάρτησης που

ονομάζεται **συνάρτηση ικανότητας (fitness function)**, και που συνήθως είναι η αντικειμενική συνάρτηση προς βελτιστοποίηση.

Η λειτουργία του αλγορίθμου μετά το βήμα της αξιολόγησης συνεχίζεται με την **επιλογή (selection)** δύο γονέων από τον πληθυσμό βάση της αξιολόγησης που προηγήθηκε και αναλόγως του προβλήματος που εξετάζεται. Σε ένα πρόβλημα μεγιστοποίησης για παράδειγμα τα δύο χρωμοσώματα με την υψηλότερη τιμή αξιολόγησης είναι πιθανότερο να επιλεγθούν σαν γονείς για τη συνέχεια της διαδικασίας, ενώ το αντίθετο συμβαίνει σε ένα πρόβλημα ελαχιστοποίησης. Στη συνέχεια τα γονίδια των επιλεγμένων χρωμοσωμάτων **διασταυρώνονται (crossover)**, με αποτέλεσμα την αναπαραγωγή (**recombination**) δύο παιδιών όπου βάσει κάποιας πιθανότητας διασταύρωσης (**crossover probability**) επιλέγονται ή όχι για την συνέχεια της διαδικασίας. Στη συνέχεια στα χρωμοσώματα που συνεχίζουν, αλλάζουν με τυχαίο τρόπο οι τιμές των γονιδίων. Η τυχαία αυτή αλλαγή οποία μιμείται τον γενετικό μηχανισμό της **μετάλλαξης (mutation)**, λαμβάνει χώρα και πάλι βάση κάποιας πιθανότητας μετάλλαξης (**mutation probability**). Οι απόγονοι που προκύπτουν αν κριθούν κατόπιν αξιολόγησης ισχυρότεροι από ήδη υπάρχοντα και ασθενέστερα χρωμοσώματα τα αντικαθιστούν, και κατόπιν επαναλαμβάνεται ο κύκλος με την επιλογή νέων γονέων. Με την διαδικασία της επιλογής ο αλγόριθμος ουσιαστικά μιμείται την συμπεριφορά των φυσικών οργανισμών όπου σε ένα ανταγωνιστικό περιβάλλον επιβιώνουν μόνο οι ισχυρότεροι οργανισμοί και οι απόγονοί τους. Το πως δημιουργείται ένας πληθυσμός ή το πως επιλέγονται από τον αρχικό πληθυσμό οι δύο γονείς, υπάρχουν διάφοροι τρόποι με πιο σύνηθες τον τυχαίο τρόπο βάση κάποιας κατανομής π.χ. ομοιόμορφης. Ο κύκλος Αξιολόγηση→Επιλογή→Διασταύρωση→Μετάλλαξη συνιστά μια γενιά (**generation**) στην λειτουργία του αλγορίθμου.

Οι βασικές μέθοδοι επιλογής των δύο γονέων είναι η **Roulette wheel selection**, η **Rank selection**, η **Tournament selection** και η **Sigma scaling selection**. Ανεξαρτήτως ποια μέθοδος θα επιλεγεί το πρώτο βήμα είναι να τεθεί μια τιμή αξιολόγησης (f_i) για κάθε λύση και να υπολογιστεί η πιθανότητα $prob_i$ για το κάθε ένα από τα N χρωμοσώματα του πληθυσμού ως εξής:

$$prob_i = \frac{f_i}{\sum_{i=1}^N f_i}$$

Στην **Roulette wheel** μέθοδο επιλογής κατασκευάζουμε ένα κυκλικό διάγραμμα (**pie chart**) όπου κάθε χρωμόσωμα καταλαμβάνει κομμάτι (**pie**) ανάλογο της τιμής αξιολόγησής και της πιθανότητας να επιλεγθεί όπως αυτή υπολογίζεται από τον παραπάνω τύπο. Το κυκλικό διάγραμμα σαν νοητή ρουλέτα αλλά όχι με ισομερή τμήματα περιστρέφεται και επιλέγει τα δύο καταλληλότερα χρωμοσώματα, εκείνα φυσικά με την καλύτερη τιμή αξιολόγησης έχουν και μεγαλύτερη πιθανότητα να επιλεγθούν. Αυτό είναι και το μειονέκτημα της μεθόδου καθώς αν υπάρχουν δύο χρωμοσώματα με εμφανώς καλύτερη τιμή αξιολόγησης σε σχέση με τα άλλα και όχι αναγκαστικά βέλτιστη αυτά θα γίνουν οι γονείς για όλους τους απογόνους της νέας γενιάς με αποτέλεσμα την μη διεύρυνση του χώρου αναζήτησης, και την γρήγορη σύγκλιση του αλγορίθμου σε κάποιο τοπικό βέλτιστο.

Στην **Rank** μέθοδο επιλογής παίρνει βαθμό αξιολόγησης (rank) ο πληθυσμός και μετά κάθε χρωμόσωμα παίρνει τιμή αξιολόγησης βάση αυτού του βαθμού. Το χειρότερο χρωμόσωμα θα πάρει τιμή $1/rank$, το αμέσως επόμενο $2/rank$ κ.ο.κ έως το καλύτερο χρωμόσωμα που θα

πάρει τιμή $N/rank$, όπου N , το σύνολο των χρωμοσωμάτων του πληθυσμού. Μετά από αυτήν την διαδικασία εφαρμόζουμε την Roulette wheel με την διαφορά ότι τώρα όλα τα χρωμοσώματα έχουν πιθανότητα να επιλεγούν, αφού πλέον οι διαφορές τους δεν είναι τόσο μεγάλες, και αυτό έχει σαν αποτέλεσμα να αποτρέπεται η πρόωρη σύγκλιση του αλγορίθμου.

Στην **Tournament** μέθοδο επιλογής ουσιαστικά ανταγωνίζονται τα χρωμοσώματα ανά δύο επιλεγμένα συγκρίνοντας την τιμή αξιολόγησής τους με τρόπο στοχαστικό. Έχοντας καθοριστεί μια πιθανότητα, παράγεται τυχαία ένας αριθμός στο διάστημα $[0,1]$ και συγκρίνεται με την προκαθορισμένη πιθανότητα. Αν η τυχαία τιμή είναι μικρότερη ή ίση της προκαθορισμένης πιθανότητας επιλέγεται το χρωμοσώμα εκείνο με την καλύτερη τιμή αξιολόγησης, αλλιώς επιλέγεται το χειρότερο από τα δύο. Με αυτό το μηχανισμό αποτρέπεται η πρόωρη σύγκλιση του αλγορίθμου αφού δίνεται η πιθανότητα να επιλεχθούν και πιο αδύναμα χρωμοσώματα. Στην πράξη συμβαίνει η προκαθορισμένη πιθανότητα να παίρνει τιμή πάνω από 0.5 ώστε πάντα να ευνοούνται ελαφρώς τα πιο δυνατά χρωμοσώματα. Τελικά τα δύο καλύτερα από το σύνολο των χρωμοσωμάτων που επιλέχτηκαν τυχαία να διαγωνιστούν είναι αυτά που θα επιλεχθούν και ως γονείς για την επόμενη γενιά.



Στην **Sigma scaling** μέθοδο επιλογής, χρησιμοποιείται και εδώ ένας μηχανισμός αποτροπής της πρόωρης σύγκλισης του αλγορίθμου όταν εξ αρχής επιλέγονται χρωμοσώματα με πολύ καλύτερη φυσική κατάσταση σε σχέση με τα υπόλοιπα. Ο μηχανισμός χρησιμοποιεί ως αναφορά την τυπική απόκλιση του πληθυσμού ώστε να ισοσκελίσει τις μεγάλες διαφορές ανάμεσα στα χρωμοσώματα, ρυθμίζοντας τις τυπικές αποκλίσεις τους από την μέση τιμή. Έτσι ο αλγόριθμος καταφέρνει και παρουσιάζει μια σταθερή κατάσταση επιλογής στην διάρκεια της λειτουργίας του.

Στο επόμενο βήμα του αλγορίθμου το οποίο αφορά την διασταύρωση των γονιδίων των δύο γονέων, οι βασικότερες μέθοδοι που χρησιμοποιούνται αρχικά όσον αφορά την δυαδική κωδικοποίηση είναι οι εξής (Hamdy A. Taha,2012):

- **Ομοιόμορφη διασταύρωση (Uniform crossover):** Σε αυτήν την περίπτωση διασταύρωσης, τα κοινά γονίδια των γονέων περνάνε και στα δύο παιδιά. Τα υπόλοιπα γονίδια για το ένα παιδί επιλέγονται βάσει ενός τυχαίου αριθμού από την ομοιόμορφη κατανομή, π.χ. για ένα τυχαίο αριθμό $R \sim U(0,1)$, αν $0 \leq R < 0.5$ το γονίδιο θα παίρνει την τιμή 1, αλλιώς για $0.5 \leq R \leq 1$ θα παίρνει την τιμή 0. Στο δεύτερο παιδί τα γονίδια λαμβάνουν τιμές τα συμπληρώματα του πρώτου.
- **Διασταύρωση ενός σημείου (Single-point crossover):** Τα γονίδια των δύο γονέων διαχωρίζονται τυχαία σε ένα σημείο, και στην συνέχεια ανταλλάσσονται.
- **Διασταύρωση πολλαπλών σημείων (K-point crossover):** Επέκταση της μεθόδου ενός σημείου για περισσότερα τυχαία σημεία.

Οι παραπάνω τρόποι διασταύρωσης παρουσιάζονται σχηματικά στον παρακάτω πίνακα.

Πίνακας 3.1: Διασταύρωση ομοιόμορφη, ενός και πολλαπλών (δύο) σημείων.

	Ομοιόμορφη	Ενός σημείου	Δύο σημείων
Γονέας 1	<u>10</u> 0 1 <u>01</u>	100 101	10 01 01
Γονέας 2	<u>10</u> 1 0 <u>01</u>	 101 001	 10 10 01
Απόγονος 1	<u>10</u> 0 0 <u>01</u>	100 001	10 10 01
Απόγονος 2	<u>10</u> 1 1 <u>01</u>	101 101	10 01 01

Όσον αφορά την αριθμητική κωδικοποίηση των χρωμοσωμάτων οι επόμενοι τρόποι διασταύρωσης είναι αυτοί που χρησιμοποιούνται κατά κύριο λόγο και παρουσιάζονται παρακάτω:

• **Αριθμητική διασταύρωση (Arithmetic crossover)** : Σε αυτήν την περίπτωση διασταύρωσης τα γονίδια των γονέων διασταυρώνονται γραμμικά βάσει ενός προκαθορισμένου συστήματος εξισώσεων. Έστω Γονέας 1 = x και Γονέας 2 = y , τότε το σύστημα εξισώσεων θα είναι το εξής:

$$\begin{aligned} \text{Απόγονος 1} &= \alpha * x + (1 - \alpha) * y \\ \text{Απόγονος 2} &= (1 - \alpha) * x + \alpha * y \end{aligned}$$

Όπου $\alpha \in [0,1]$ ένας τυχαίος αριθμός που καθορίζει το βάρος της κάθε μεταβλητής (Γονέα).

• **Διασταύρωση ανάμειξης (Blending crossover)**: Και σε αυτήν την περίπτωση διασταύρωσης τα γονίδια των γονέων διασταυρώνονται γραμμικά βάσει ενός προκαθορισμένου συστήματος εξισώσεων. Έστω Γονέας 1 = x και Γονέας 2 = y , τότε το σύστημα εξισώσεων θα είναι το εξής:

$$\begin{aligned} \text{Απόγονος 1} &= x - [\beta * (x - y)] \\ \text{Απόγονος 2} &= y + [\beta * (x - y)] \end{aligned}$$

Όπου $\beta \in [0,1]$ ένας τυχαίος αριθμός που καθορίζει το βάρος της κάθε μεταβλητής (Γονέα).

Μετά την διασταύρωση η κάθε λύση θα υποστεί μια τυχαία μετάλλαξη σε κάποιο από τα γονίδια της, βάση κάποια πιθανότητας.

Η μετάλλαξη αυτό που ουσιαστικά προσφέρει είναι μια μικρή πιθανότητα τυχαίας αναζήτησης, έτσι ώστε κανένα σημείο του χώρου κατά την λειτουργία του αλγορίθμου, να μην έχει μηδενική πιθανότητα εξερεύνησης. Η βασικότερη μέθοδος μετάλλαξης είναι η ομοιόμορφη (uniform), η οποία επιλέγει με τυχαίο τρόπο κάποιο γονίδιο σε ένα χρωμόσωμα, και του αλλάζει τιμή, με κάποια άλλη τιμή που δεν παραβιάζει τις συνθήκες και περιορισμούς του προβλήματος.

Πίνακας 3.2: Διαδικασία ομοιόμορφης μετάλλαξης

Πριν την μετάλλαξη	Απόγονος	1100001 0 101001
Μετά την μετάλλαξη	Απόγονος	1100001 1 101001

Οι δύο πολύ βασικές διαδικασίες που λαμβάνουν χώρα κατά την λειτουργία των μεταερευτικών αλγορίθμων, αυτή της εξερευνησης (**exploration**) και αυτή της εκμετάλλευσης (**exploitation**) συναντώνται και εδώ. Με τον όρο αναζήτηση (**exploration**) εννοούμε την διαδικασία δημιουργίας πληθυσμών εξερευνώντας και διευρύνοντας τον χώρο αναζήτησης χρησιμοποιώντας γενετικούς μηχανισμούς (**genetic operators**) όπως η διασταύρωση (**crossover**) και η μετάλλαξη (**mutation**), ενώ με τον όρο εκμετάλλευση (**exploitation**), εννοούμε την διαδικασία μείωσης της ποικιλομορφίας των πληθυσμών διατηρώντας μόνο τους ισχυρότερους απογόνους και ουσιαστικά συρρικνώνοντας έτσι τον χώρο αναζήτησης γύρω από την βέλτιστη λύση. Συχνά για αυτό το σκοπό χρησιμοποιείται μια στρατηγική (**elitist strategy**), η οποία επιτρέπει σε ένα ποσοστό ισχυρών χρωμοσωμάτων να διατηρηθούν και στις επόμενες γενιές στην περίπτωση που αυτά για κάποιο λόγο μετά τις διαδικασίες της διασταύρωσης και της μετάλλαξης δεν επιβιώσουν.

Τέλος όσον αφορά τα κριτήρια τερματισμού – σύγκλισης (**convergence**) του αλγόριθμου αυτά μπορεί να είναι είτε ένα καθορισμένος αριθμός επαναλήψεων (**iterations**) δηλαδή δημιουργίας γενεών, είτε ένα μέγιστος αριθμός επαναλήψεων κατά τον οποίο δεν έχει παρατηρηθεί βελτίωση της υπάρχουσας λύσης, είτε απλά μία βέλτιστη λύση η οποία καλύπτει τις ανάγκες του προβλήματος και γίνεται αποδεκτή.

Ολοκληρώνοντας, αυτό που πρέπει να επισημανθεί είναι ότι το πλεονέκτημα του αλγορίθμου έγκειται στο γεγονός ότι μπορεί να δώσει αποδεκτά πολύ καλές λύσεις, όχι απαραίτητα βέλτιστες, σε πολύ καλό όμως χρόνο, ενώ μειονεκτεί ότι χρωμοσώματα με υψηλές τιμές αξιολόγησης μπορούν σε γρήγορο χρόνο να κατακλυσουν το πληθυσμό με απογόνους αφήνοντας μόνο τον μηχανισμό της μετάλλαξης για εξερεύνηση νέου χώρου λύσεων, με αποτέλεσμα το γρήγορο τερματισμό του αλγορίθμου. Παρακάτω παρουσιάζονται τα στάδια λειτουργίας του αλγόριθμου και το διάγραμμα ροής του.

3.2 Λειτουργία Γενετικού Αλγόριθμου (GA)

Φάση 0

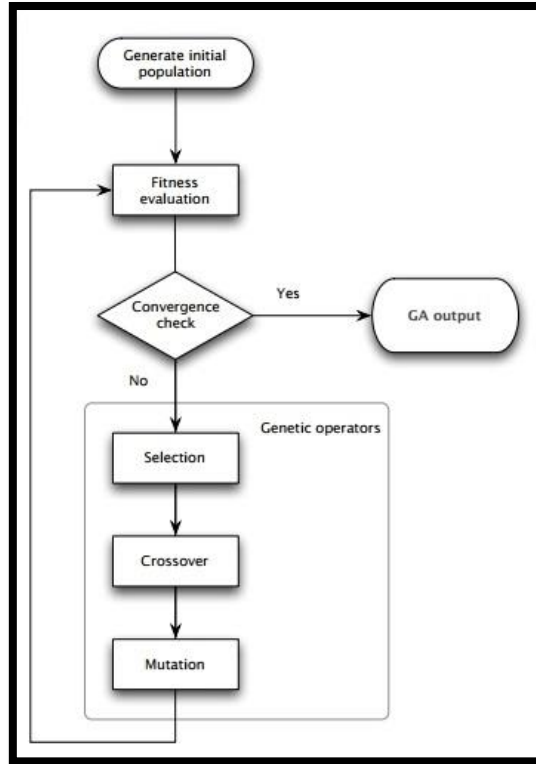
- Το κάθε χρωμόσωμα κωδικοποιείται είτε δυαδικά είτε αριθμητικά ανάλογα τη φύση του προβλήματος.
- Για $k = 0$, δημιουργείται ένας τυχαίος πληθυσμός X από $N = \{s_1^{(0)}, s_2^{(0)}, \dots, s_N^{(0)}\}$ εφικτά χρωμοσώματα.
- Εκτιμάται η συνάρτησης ικανότητας $f(S_i^{(k)})$, του κάθε χρωμοσώματος s , και καταγράφεται ως s^* , η καλύτερη διαθέσιμη λύση.

Φάση 1

- Εκτιμάται η πιθανότητα $p_i = f_i / \sum_{i=1}^N f_i$, όπου βάσει αυτής και της καθορισμένης μεθόδου επιλογής, επιλέγονται (**selection**) δύο χρωμοσώματα – γονείς από τον πληθυσμό X .
- Διασταυρώνονται (**crossover**) τα γονίδια τους, για τη δημιουργία δυο νέων απογόνων μέσω καθορισμένης μεθόδου διασταύρωσης. Αυτοί περνάνε στην επόμενη γενιά με τρόπο στοχαστικό βάσει κάποιας πιθανότητας, συνήθως υψηλής $p_i \approx 0.8 - 1.0$. Στην περίπτωση που η πιθανότητα διασταύρωσης πάρει την τιμή 0, οι απόγονοι προκύπτουν αντιγράφοντας (**duplicating**) πλήρως τους γονείς, και έτσι η διαδικασία δεν διακόπτεται.
- Γίνεται μετάλλαξη (**mutation**) σε κάποια γονίδια στοχαστικά βάσει κάποιας πιθανότητας, συνήθως χαμηλής $p_i \approx 0.01 - 0.1$
- Εάν οι λύσεις δεν είναι εφικτές γίνεται επανάληψη της φάσης 1 μέχρι να επιτευχθεί εφικτότητα. Σχηματίζεται λοιπόν νέος πληθυσμός $N = \{s_1^{(k+1)}, s_2^{(k+1)}, \dots, s_N^{(k+1)}\}$ ενημερώνεται η καλύτερη λύση s^* , το βήμα επανάληψης αυξάνεται κατά ένα $k = k + 1$ και ο αλγόριθμος περνάει στην φάση 2.
- Προαιρετικά για καλύτερη ικανότητα εκμετάλλευσης (**exploitation**) χρησιμοποιείται και μια στρατηγική διατήρησης ενός ποσοστού συνήθως 5% ισχυρών χρωμοσωμάτων (**Elitism**) σε επόμενες γενιές, σε περίπτωση που αυτά για κάποιο λόγο δεν επιβιώσουν.

Φάση 2

Εφόσον ικανοποιηθεί κάποιο από τα κριτήρια τερματισμού, ο αλγόριθμος σταματάει και το $s^* = \operatorname{argmax}_{S_i^{(k)}} f(S_i^{(k)})$, είναι η καλύτερη διαθέσιμη λύση, αλλιώς ο αλγόριθμος επιστρέφει στην Φάση 1 (Hamdy A. Taha,2012).



Πηγή:(Journal of Statistical Software, Volume 53)

Εικόνα 3.1 Διάγραμμα ροής γενετικού αλγόριθμου

3.3 Λειτουργία Γενετικού Αλγόριθμου (GA) μέσω κώδικά.

Στη συνέχεια θα παρουσιάσουν αναλυτικά με την εφαρμογή κώδικά μέσω του προγραμματιστικού περιβάλλοντος της R όλες οι βασικές λειτουργίες του γενετικού αλγόριθμου, και για την δυαδική και την αριθμητική κωδικοποίηση. Για την δυαδική κωδικοποίηση θα χρησιμοποιηθεί η διαστάυρωση ενός σημείου (single-point crossover), ενώ για την αριθμητική κωδικοποίηση θα χρησιμοποιηθεί η αριθμητική διαστάυρωση (arithmetic crossover). Σαν αντικειμενική συνάρτηση (objective function) η οποία θα λειτουργεί και ως συνάρτηση ικανότητας (Fitness function) θα χρησιμοποιηθεί συμβολικά η μεταβλητή f στην θέση της οποίας μπορεί να οριστεί οποιαδήποτε συνάρτηση. Τέλος όσον αφορά την μέθοδο επιλογής και μετάλλαξης θα χρησιμοποιηθούν από κοινού η Roulette wheel Selection method, και η Uniform Mutation.

3.3.1 Γενετικός Αλγόριθμος με δυαδική (binary) κωδικοποίηση

Για την περίπτωση λοιπόν όπου χρησιμοποιείται κωδικοποίηση των χρωμοσωμάτων με δυαδικές τιμές, παρουσιάζεται παρακάτω η λειτουργία του αλγόριθμου με την βοήθεια του κώδικα.

Φάση 0

- Το κάθε χρωμόσωμα κωδικοποιείται δυαδικά.
- Για $k = 0$, δημιουργείται ένας τυχαίος πληθυσμός X από $N = \{s_1^{(0)}, s_2^{(0)}, \dots, s_N^{(0)}\}$ εφικτά χρωμοσώματα μεγέθους l γονιδίων (genes).

```
##### BINARY Initialization Function #####
population <- function(N,l) {
  population <- data.frame(matrix(ncol = l, nrow = N))
  names(population)<-NULL
  for(i in 1:l){
    names(population)[i]<-paste("Gene",i,sep="")
  }
  for (i in 1:N) {
    for (j in 1:l) {
      population[i,j]<- round(runif(1,0,1))
    }
  }
  return(population)
}
##### Input Population parameters & Function Execution #####
N <- # Αριθμός Χρωμοσωμάτων
l <- Problem.nVar # Αριθμός γονιδίων(genes), ίσος με αριθμό μεταβλητών
population(N, l)
```

- Εκτιμάται η συνάρτησης ικανότητας $f(s_i^{(k)})$, του κάθε χρωμοσώματος s .

```
##### Objective function to be optimized #####
obj <- function(x) {
  y <- rep(0, N)
  for (i in 1:N){
    y[i] <- f
  }
  return (y)
}
##### Fitness Function #####
fit <- function(x) {
  x$fitness_values <- obj(x)
  return(x)
}
```

Φάση 1

• Εκτιμάται η πιθανότητα $p_i = f_i / \sum_{i=1}^N f_i$, όπου μέσω της Roulette wheel selection method, επιλέγονται (**selection**) δύο χρωμοσώματα – γονείς από τον πληθυσμό X .

```
##### Roulette Wheel SELECTION #####
RW_selection <- function(x) {
  x <- fit(population)
# Εδώ εξασφαλίζουμε ότι δεν θα έχουμε αρνητικές τιμές ικανότητας, ώστε στη συνέχεια
# να υπολογιστούν σωστά οι σχετικές και οι αθροιστικές σχετικές συχνότητες
  if(any(x$fitness_values < 0)) {
    a <- 1
    b <- min(x$fitness_values)
    b <- abs(b)
    scaled_fitness_values <- a*x$fitness_values + b + 1
    x$normalize_fitness_values <- scaled_fitness_values / sum(scaled_fitness_values)
  }else{
    x$normalize_fitness_values <- x$fitness_values / sum(x$fitness_values)
  }
# Υπολογισμός σχετικών συχνοτήτων (πιθανότητα), decreasing ordered
  x$normalize_fitness_values <- round(x$normalize_fitness_values,4)
  x <- x[order(-x$normalize_fitness_values),]
  row.names(x) <- 1:N
# Κατασκευή της Roulette Wheel μέσω υπολογισμού
# των αθροιστικών σχετικών συχνοτήτων (csum), decreasing από το 1 προς το 0
  csum <- rep(0, length(x$fitness_values))
  for (i in 1:length(x$fitness_values)){
    csum[i] <- sum(x$normalize_fitness_values[i:N])
  }
  csum <- round(csum,3)
  x$csum <- csum
# Επιλογή γονέα 1 συγκρίνοντας ένα τυχαίο αριθμό (0,1) από την ομοιόμορφη κατανομή
# και την τιμή της αθροιστικής συχνότητας του [i], χρωμοσώματος.
  i <- 1
  Parent1 <- x[N,] # δηλώνουμε σαν Parent1 το χειρότερο χρωμόσωμα να αποφύγουμε
  # το λάθος αν προκύψει runif < χειρότερου fit value

  repeat {
    i = i+1
    if (runif (1,0,1) > x$csum[i]){
      Parent1 <- x[i-1,]
      break
    }
  }
# Επιλογή γονέα 2
  j <- 1
  Parent2 <- x[N,]
  repeat {
    j = j+1
    if ((runif(1,0,1) > x$csum[j]) & ((j-1) != (i-1))){ # το & για !=γονείς
      Parent2 <- x[j-1,]
      break
    }
  }
  Parents <- rbind(Parent1, Parent2)
  return(Parents) }
```

- Διασταυρώνονται (**crossover**) τα γονίδια των χρωμοσωμάτων, για τη δημιουργία δυο νέων απογόνων με τρόπο στοχαστικό βάση κάποιας πιθανότητας, συνήθως υψηλής $p_i \approx 0.8 - 1.0$.

```
##### BINARY Single Point CROSSOVER #####

Crossover <- function(pc) {
  Gene_no <- 1
  Parents <- RW_selection(fit(population))
  Parent1 <- Parents[1,1:1]
  Parent2 <- Parents[2,1:1]

# Καθορισμός άνω και κάτω ορίου μήκους χρωμοσωμάτων.
  UB <- length(Parent1) - 1 # για αποφυγή της περίπτωσης να βγει το cross_p
  =maxLength

  LB <- 1

# Εύρεση του τυχαίου σημείου για τον διαχωρισμό των γονιδίων
  Cross_P <- round ((UB - LB) * runif(1,0,1) + LB)

# Δημιουργία Παιδιού 1
  Part1 <- Parent1[1:Cross_P]
  Part2 <- Parent2[(Cross_P+1):Gene_no]
  Child1 <- cbind(Part1, Part2)

# Πιθανότητα διασταύρωσης (pc), με την οποία συγκρίνεται ένας τυχαίος αριθμός (0,1)
# από την ομοιόμορφη και καθορίζεται αν στην επόμενη γενιά περάσει το παιδί 1 η ο
# γονιός 1.
  if (runif(1,0,1) <= pc) {
    Child1 <- Child1
  }else{
    Child1 <- Parent1
  }

# Δημιουργία Παιδιού 2
  Part1 <- Parent1[(Cross_P+1):Gene_no]
  Part2 <- Parent2[1:Cross_P]
  Child2 <- cbind(Part2, Part1)

# Πιθανότητα διασταύρωσης (pc) για το παιδί 2.
  if (runif(1,0,1) <= pc) {
    Child2 <- Child2
  }else{
    Child2 <- Parent2
  }
  Children <- rbind(Child1, Child2)
  Children
  return(Children)
}
```


- Γίνεται μετάλλαξη (**mutation**) σε κάποια γονίδια στοχαστικά βάση κάποιας πιθανότητας, συνήθως χαμηλής $p_i \approx 0.01 - 0.1$

```
##### BINARY MUTATION #####

Mutation <- function(pc,pm) {

  children <- Crossover(pc)
  child1 <- children [1,]
  child2 <- children [2,]
  # Παράγεται, και συγκρίνεται ένας τυχαίος αριθμός (0,1) για κάθε γονίδιο από την
  # ομοιόμορφη με την καθορισμένη πιθανότητα μετάλλαξης (pm), αν ο αριθμός είναι μι-
  # κρότερος του (pm), τότε αντικαθίσταται από τον συμπληρωματικό του.

  # Για το Χρωμόσωμα (Παιδί 1)
  for (k in 1:l){
    if(runif(1,0,1) < pm) {
      child1[,k] <- 1-child1[,k]
    }
  }
  # Για το Χρωμόσωμα (Παιδί 2)
  for (n in 1:l){
    if(runif(1,0,1) < pm) {
      child2[,n] <- 1-child2[,n]
    }
  }
  mutate_child1<- child1
  mutate_child2<- child2
  mutate_children <- rbind(mutate_child1, mutate_child2)
  return (mutate_children)
}
```

3.3.2 Γενετικός Αλγόριθμος με αριθμητική (arithmetic) κωδικοποίηση

Όταν στον αλγόριθμο χρησιμοποιείται κωδικοποίηση των χρωμοσωμάτων με πραγματικές τιμές, αυτό που αλλάζει σε σχέση με την δυαδική περίπτωση, ουσιαστικά είναι η αρχικοποίηση του πληθυσμού αφού πλέον τα γονίδια μπορούν να πάρουν οποιαδήποτε συνεχή τιμή εντός ενός οριοθετημένου χώρου αναζήτησης. Επίσης αλλάζει ο μηχανισμός της διασταύρωσης καθώς οι προηγούμενες περιπτώσεις, δεν μπορούν να εφαρμοστούν. Και τέλος διαφέρει και ο μηχανισμός μετάλλαξης, καθώς δεν υπάρχει πλέον η έννοια της συμπληρωματικής τιμής και η τιμή του γονιδίου που μεταλλάσσεται τυχαία, μπορεί να αντικατασταθεί από μία οποιαδήποτε τυχαία συνεχή τιμή εντός του προκαθορισμένου χώρου αναζήτησης. Βάσει όσον αναφέρθηκαν παρουσιάζονται με την βοήθεια του κώδικα οι αλλαγές στους μηχανισμούς στην λειτουργία του αλγόριθμου, με επιπλέον προσθήκη το βήμα δημιουργίας του νέου πληθυσμού κάθε γενέας και τον προαιρετικό μηχανισμό της στρατηγικής διατήρησης των ισχυρών χρωμασωμάτων (**Elitism**).

Φάση 0

- Το κάθε χρωμόσωμα κωδικοποιείται με πραγματικές τιμές (real – valued).
- Για $k = 0$, δημιουργείται ένας τυχαίος πληθυσμός X από $N = \{s_1^{(0)}, s_2^{(0)}, \dots, s_N^{(0)}\}$ από εφικτά χρωμοσώματα εντός ενός οριοθετημένου συνεχούς χώρου αναζήτησης, μεγέθους l γονιδίων (genes), όσες και οι μεταβλητές του προβλήματος.

```
##### Input Population parameters #####
Problem.nVar <- 1      # Αριθμός μεταβλητών του προβλήματος
N <-                  # Αριθμός Χρωμοσωμάτων
l <- Problem.nVar     # Αριθμός γονιδίων
ub <- c()             # Άνω όριο περιοχής αναζήτησης
lb <- c()             # Κάτω όριο περιοχής αναζήτησης

##### Real_Valued Initialization Function #####
population <- function(N,l,lb,ub) {
  a <- rep(0,N)
  b <- rep(0,N)

  for(i in 1:N){
    a[i] <- ((ub - lb) * runif(1,0,1) + lb)
  }
  population <- data.frame(Gene = a, fitness_values = b)
  return(population)
}

population <- population(N,l,lb,ub)
```

Φάση 1

- Διασταυρώνονται (crossover) τα γονίδια των χρωμοσωμάτων
- Αριθμητική διασταύρωση (Arithmetic crossover) :

```
##### Real_Valued Arithmetic CROSSOVER #####
Crossover <- function(pc) {
  Gene_no <- 1
  Parents <- RW_selection(fit(population))
  Parent1 <- Parents[1,1:l]
  Parent2 <- Parents[2,1:l]
  # Διανύσματα παιδιών
  child1 <- rep(0, Gene_no)
  child2 <- rep(0, Gene_no)
  # alpha ένας τυχαίος αριθμός (0,1) από την ομοιόμορφη κατανομή
  # που καθορίζει το βάρος της κάθε μεταβλητής (Γονέας).
  alpha <- runif(1,0,1)
  #Απόγονος 1 =  $\alpha * x + (1 - \alpha) * y$ 
  #Απόγονος 2 =  $(1 - \alpha) * x + \alpha * y$ 
  #Γονέας 1 =  $x$  και Γονέας 2 =  $y$ 
```

```

# Δημιουργία Παιδιού 1
  child1 <- (alpha * Parent1[1]) + ((1- alpha) * Parent2[1])

# Δημιουργία Παιδιού 2
  child2 <- ((1- alpha) * Parent1[1]) + (alpha * Parent2[1])

# Πιθανότητα διασταύρωσης (pc), με την οποία συγκρίνεται ένας τυχαίος αριθμός (0,1)
# από την ομοιόμορφη και καθορίζεται αν στην επόμενη γενιά περάσει το παιδί 1 η ο
# γονιός 1.
  if (runif(1,0,1) <= pc) {
    child1 <- child1
  }else{
    child1 <- Parent1
  }

# Πιθανότητα διασταύρωσης (pc) για το παιδί 2
  if (runif(1,0,1) <= pc) {
    child2 <- child2
  }else{
    child2 <- Parent2
  }
  children <- rbind(child1, child2)
  children
  return(children)
}

```

• Γίνεται μετάλλαξη (mutation) σε κάποια γονίδια στοχαστικά βάση κάποιας πιθανότητας.

```

##### Real_Valued MUTATION #####
Mutation <- function(l,pc,pm,lb,ub) {
  children <- Crossover(pc)
  child1 <- children [1,]
  child2 <- children [2,]

# Παράγεται, και συγκρίνεται ένας τυχαίος αριθμός (0,1) για κάθε γονίδιο από την
# ομοιόμορφη με την καθορισμένη πιθανότητα μετάλλαξης (pm), αν ο αριθμός είναι μι-
# κρότερος του (pm), τότε αντικαθίσταται από έναν τυχαίο συνεχή αριθμό εντός του χώ-
# ρου αναζήτησης.

  # Για το Χρωμόσωμα (Παιδί 1)
  if(runif(1,0,1) < pm) {
    child1 <- ((ub[1]- lb[1]) * runif(1,0,1) + lb[1])
  }

  # Για το Χρωμόσωμα (Παιδί 2)
  if(runif(1,0,1) < pm) {
    child2 <- ((ub[1]- lb[1]) * runif(1,0,1) + lb[1])
  }
  mutate_child1<- child1
  mutate_child2<- child2
  mutate_children <- rbind(mutate_child1, mutate_child2)
  return (mutate_children)
}

```

- Δημιουργείται ο νέο πληθυσμός (new_pop) μεγέθους ίσου με τον αρχικό, για αυτό και οι λειτουργίες Selection - Crossover – Mutation, λαμβάνουν χώρα $N/2$ φορές όπου N το μέγεθος του αρχικού πληθυσμού.

```
##### New Population after Selection, Crossover, Mutation #####
# Ο νέος Πληθυσμός πρέπει να είναι σε μέγεθος ίδιος με τον αρχικό
new_pop <- function(pc,pm) {
  df <- rep(0,N)

# Για αυτό επαναλαμβάνουμε N/2 φορές
  for(i in seq(from=1, to=N,by=2)) {
    df[i:(i+1)] <- Mutation(1,pc,pm,lb,ub)
  }
  mut_pop <- data.frame(Gene=df)
  return(mut_pop)
}
```

- Σε κάθε νέο πληθυσμό όπως αυτός δημιουργήθηκε παραπάνω αντικαθίστανται τα χρωμοσώματα με την χειρότερη αξιολόγηση, από εκείνα με την καλύτερη αξιολόγηση από τον προηγούμενο πληθυσμό, σε ποσοστό που καθορίζουμε μέσω του **Elite Ratio (Er)**. Έτσι ο τελικός πληθυσμός της γενιάς (new_population) κρατάει καλές λύσεις που πιθανόν να χάνονταν στην πορεία.

```
##### New Population & Elitism #####
new_population <- function(pc, pm, Er) {

# Το Er είναι το Elitism Ratio το οποίο πολλαπλασιάζεται με τον πληθυσμό
# για να δούμε πόσα Elite Chromosomes θα διατηρήσουμε στην νέα γενιά
  Elite_no <- round(Er*N)
  x <- fit(population)

# Ταξινόμηση του προηγούμενου πληθυσμού βάση των Fitness Values με φθίνουσα
σειρά(best~worst)
  elites <- data.frame(matrix(ncol = 2, nrow = Elite_no))
  x <- x[order(-x$fitness_values),]
# και αποθήκευση των Elite_No = (Er * N) πρώτων και ισχυροτέρων χρωμοσωμάτων
  elites[1:Elite_no,] <- x[1:Elite_no,]

# Ταξινόμηση του νέου πληθυσμού βάσει των Fitness Values με αύξουσα σειρά
(worst~best)
  new_pop <- new_pop(pc,pm)
  new_pop <- fit(new_pop)
  new_pop <- new_pop[order(new_pop$fitness_values),]
# και αντικατάσταση των Elite_No = (Er * N) πρώτων και ασθενέστερων χρωμοσωμάτων
με τα Elite που έχουμε αποθηκεύσει
  new_pop[1:Elite_no,] <- elites[1:Elite_no,]
  return(new_pop)
}
population <- new_population(pc, pm, Er)
```

3.4 Παραδείγματα εφαρμογής Γενετικού Αλγόριθμου (GA)

Παραπάνω είδαμε επιμέρους τα βασικά στάδια λειτουργίας του γενετικού αλγόριθμου, είτε πρόκειται για την περίπτωση που έχουμε δυαδική κωδικοποίηση των χρωμοσωμάτων, είτε όταν χρησιμοποιούνται οι πραγματικές τιμές. Πριν την βασική εφαρμογή που αφορά την βελτιστοποίηση χαρτοφυλακίου στην τέταρτη και τελευταία ενότητα η οποία έχει να κάνει με πολυδιάστατες μεταβλητές, δίνονται δύο απλά παραδείγματα εφαρμογής του γενετικού αλγόριθμου, με μονοδιάστατες και δισδιάστατες μεταβλητές ώστε να παρουσιαστούν πλήρως οι λειτουργίες του σε πιο απλές περιπτώσεις, όπου με την παρουσίαση γραφημάτων που δεν μπορούν να απεικονιστούν στο χώρο όταν αυξάνεται ο αριθμός των μεταβλητών, να μπορεί να κατανοηθεί πλήρως η επίδραση των παραμέτρων του.

3.4.1 Παράδειγμα 1 – Μονοδιάστατη μεταβλητή

Στο πρώτο αυτό παράδειγμα θα εφαρμοστεί ο αλγόριθμος, ώστε να βρει το ολικό μέγιστο της συνάρτησης $f(x) = (x^2 + x) \cos(x)$, $x \in [-10, +10]$, η οποία παρουσιάζει γνωστό ολικό μέγιστο ίσο με 47.70562 στο σημείο 6.560539. Ο αλγόριθμος θα τρέξει με τους παρακάτω μηχανισμούς, ενώ ο κώδικας του παρουσιάζεται αναλυτικά στο παράρτημα ΠΙ στο τέλος του συγγράμματος.

Επιλογή μεθόδων βασικών μηχανισμών λειτουργίας:

- **Μέθοδος επιλογής:** Roulette wheel selection
- **Μέθοδος Διασταύρωσης:** Arithmetic crossover
- **Μέθοδος Μετάλλαξης:** Uniform mutation

3.4.2 Εκτέλεση κώδικα Γενετικού Αλγόριθμου (GA) μονοδιάστατης μεταβλητής:

Στη συνέχεια θα εκτελεστεί ο κώδικας για μία επανάληψη του αλγορίθμου για διάφορες τιμές των σημαντικότερων παραμέτρων του, και θα μελετηθεί μέσω του τελικού πληθυσμού η επίδραση αυτών στην απόδοση του αλγορίθμου. Οι παράμετροι αυτοί είναι η πιθανότητα διασταύρωσης (**crossover probability**), η πιθανότητα μετάλλαξης (**mutation probability**), και το ποσοστό διατήρησης σε επόμενη γενιά των ισχυρότερων χρωμοσωμάτων (**Elitism ratio**).

Στην πρώτη περίπτωση θα παραμετροποιηθεί η πιθανότητα διασταύρωσης, επισημαίνοντας ότι παράγεται ένας τυχαίος αριθμός (0,1) από την ομοιόμορφη κατανομή και αν αυτός ο αριθμός είναι μικρότερος της πιθανότητας διασταύρωσης, τότε στην επόμενη γενιά συνεχίζει το νέο χρωμόσωμα-παιδί αλλιώς στην αντίθετη περίπτωση συνεχίζει το χρωμόσωμα με ακριβή αντιγραφή των χαρακτηριστικών του γονιού.

Σταθερές που εισάγονται:

- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) $M = 20$

Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 100$

Παράμετροι που μεταβάλλονται:

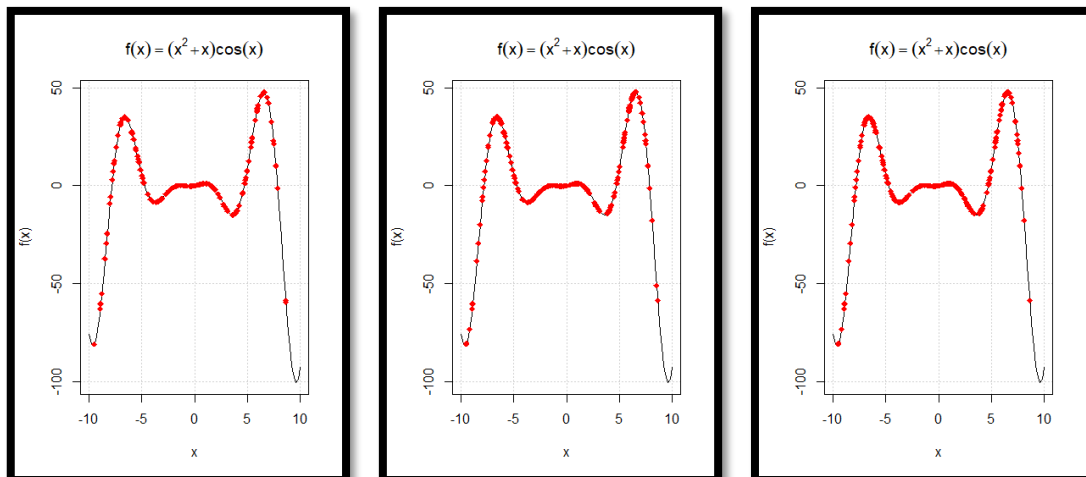
- Πιθανότητες διασταύρωσης (**Crossover probability**) $p_{c1} = 0, p_{c2} = 0.5, p_{c3} = 1.0$

Κώδικας:

```
#### CONTROLLING PARAMETERS OF GA ALGORITHM #####
Problem.nVar <- 1 # Number of problem Variables
M <- 20 # Number of chromosomes (population)
l <- Problem.nVar # Number of genes
MaxGen <- 100 # Generations (iterations)
pc <- 0, 0.5, 1.0 # Probability of Crossover
pm <- 0.1 # Probability of Mutation
Er <- 0.05 # Elitism Ratio
ub <- c(10) # UPPER BOUND FOR SEARCH SPACE
lb <- c(-10) # LOWER BOUND FOR SEARCH SPACE
CGenetic_Algorithm (M, l, MaxGen, pc, pm, Er, lb, ub)
```

Πίνακας 3.3 Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_c

p_c	Μέση τιμή \hat{x}	Τυπική α-πόκλιση \hat{x}	Βέλτιστη τιμή \hat{x}	Μέση τιμή $\hat{f}(x)$	Τυπική α-πόκλιση $\hat{f}(x)$	Βέλτιστη τιμή $\hat{f}(x)$
0.0	5.744323	2.262612	6.565472	40.35335	17.96719	47.70497
0.5	6.116999	1.811473	6.560373	45.2802	10.65962	47.70562
1.0	6.168343	1.823538	6.560545	45.31326	10.6674	47.70562



Εικόνα 3.2 Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $p_{c1} = 0$, $p_{c2} = 0.5$, $p_{c3} = 1.0$

Αναλύοντας τα αποτελέσματα, για $p_{c1} = 0$ το οποίο πρακτικά σημαίνει ότι δεν συνεχίζουν παιδιά στις επόμενες γενιές αλλά μόνο τα αντίγραφα των γονέων και μόνο μέσω του μηχανισμού της μετάλλαξης μπορεί να βελτιώνεται ο πληθυσμός, η βέλτιστη λύση είναι η χειρότερη εκ των τριών, ενώ και η τυπική απόκλιση είναι μεγαλύτερη που σημαίνει ότι ο τελικός πληθυσμός είναι πιο διασκορπισμένος καθώς δεν υπάρχει καλή ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης. Για $p_{c2} = 0.5$ εξάγονται τα καλύτερα αποτελέσματα καθώς υπάρχει καλύτερη ισορροπία και ο τελικός πληθυσμός συγκεντρώνεται γύρω από την βέλτιστη λύση. Τέλος για $p_{c3} = 1.0$, το οποίο πρακτικά σημαίνει στις επόμενες γενιές συνεχίζουν μόνο παιδιά τα αποτελέσματα είναι εξίσου καλά, όμως φαίνεται από τις αποκλίσεις ότι το αποτέλεσμα είναι ελαφρώς υποδεέστερο. Παρακάτω θα πραγματοποιηθούν πολλαπλές επαναλήψεις 5×10^2 στον αριθμό, του αλγορίθμου για κάθε επιλογή της πιθανότητας διασταύρωσης p_c , και σε πίνακα θα καταγράφονται οι εκτιμήσεις για τη μέση τιμή και τη διασπορά της βέλτιστης λύσης, καθώς και το μέσο τετραγωνικό σφάλμα MSE. Υπενθυμίζεται ότι το MSE των εκτιμήσεων είναι το άθροισμα, της διακύμανσης των εκτιμήσεων και του τετραγώνου της μεροληψίας (bias). Ο αλγόριθμος λοιπόν θα τρέξει με τα παρακάτω δεδομένα, ενώ ο κώδικας του παρουσιάζεται αναλυτικά στο παράρτημα Π1 στο τέλος του συγγράμματος.

Σταθερές που εισάγονται:

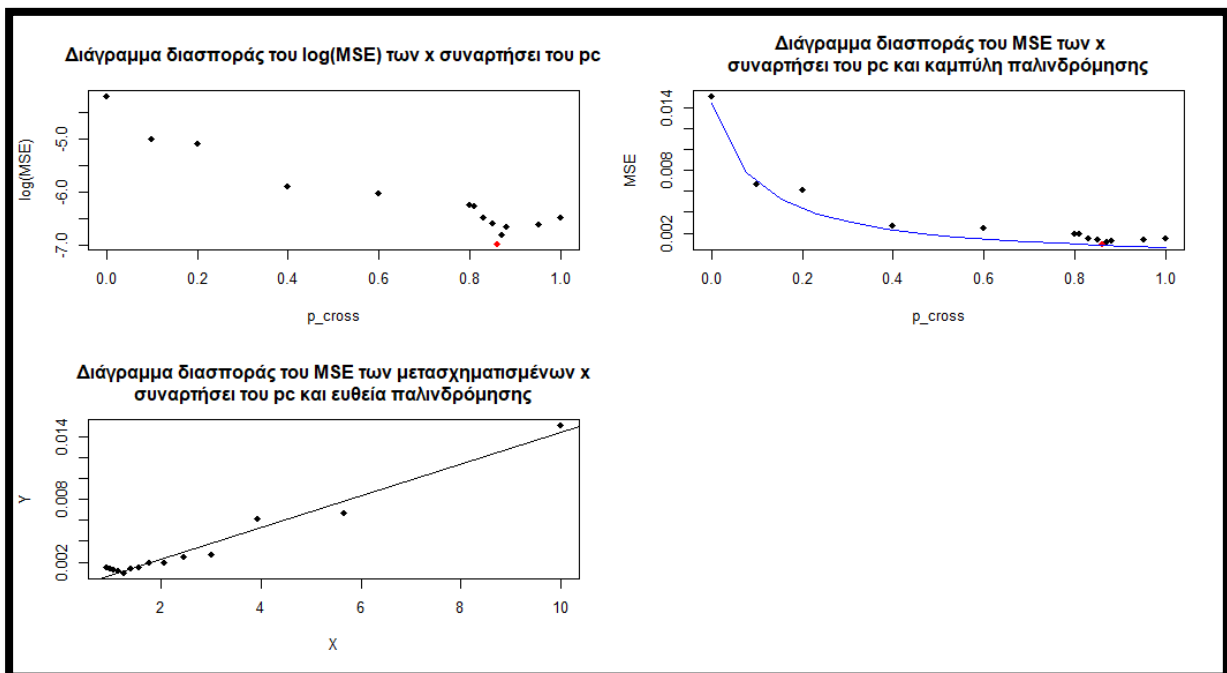
- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) $M = 30$

Κριτήρια τερματισμού (convergence):

- Μείστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 30$

Πίνακας 3.4 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου $5 * 10^2$ φορές, για διάφορες τιμές της πιθανότητας διασταύρωσης p_c .

p_c	Μέση τιμή \hat{x}	Τυπική απόκλιση \hat{x}	Μέση τιμή $\hat{f}(x)$	Τυπική απόκλιση $\hat{f}(x)$	MSE (\hat{x})
0.00	6.558078	0.1229062	47.30626	0.8388510	0.01511199
0.10	6.558017	0.0818489	47.53349	0.6976759	0.00670561
0.20	6.553499	0.0782842	47.54617	0.7384685	0.00617799
0.40	6.559390	0.0526273	47.63198	0.3862294	0.00277095
0.60	6.553075	0.0489330	47.64392	0.4008724	0.00245015
0.80	6.552030	0.0437163	47.65518	0.3064267	0.00198352
0.81	6.553416	0.0434984	47.65577	0.2677651	0.00194284
0.83	6.554850	0.0390467	47.66542	0.2567808	0.00155701
0.85	6.555014	0.0367648	47.66997	0.1962943	0.00138217
0.86	6.557374	0.0304820	47.68114	0.1648731	0.00093917
0.87	6.555410	0.0330378	47.67707	0.2185269	0.00111781
0.88	6.554780	0.0358220	47.67132	0.1741932	0.00131638
0.95	6.556125	0.0364572	47.67182	0.3246406	0.00134861
1.00	6.555662	0.0390009	47.66535	0.2451288	0.00154485



Εικόνα 3.3 Διαγράμματα διασποράς για διάφορες τιμές της πιθανότητας διασταύρωσης p_c , με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης και διάγραμμα με καμπύλη παλινδρόμησης και με ευθεία παλινδρόμησης σε μετασχηματισμένα δεδομένα.

Παρατηρούμε ότι το καλύτερο αποτέλεσμα καταγράφηκε για τιμή πιθανότητας ίση με 0.86 η οποία είναι εντός του εύρους πιθανοτήτων $p_c \approx 0.8 - 1.0$ που χρησιμοποιούνται συνήθως ως πιθανότητα διασταύρωσης. Ενώ από τα παραπάνω διαγράμματα του μέσου τετραγωνικού

σφάλματος MSE φαίνεται ότι το διάστημα $p_c \approx 0.8 - 1.0$ είναι το διάστημα με τις καλύτερες δυνατές επιλογές πιθανοτήτων. Επίσης εκτιμάται με μια καμπύλη παλινδρόμησης το πως μπορεί να σχετίζεται η ποιότητα της λύσης με την μεταβολή της πιθανότητας διασταύρωσης p_c . Ενώ επιπλέον απεικονίζεται και ο μετασχηματισμός που μπορεί να επιτευχθεί στα δεδομένα δημιουργώντας ένα γραμμικό μοντέλο πρόβλεψης.

Στην δεύτερη περίπτωση θα παραμετροποιηθεί η πιθανότητα μετάλλαξης, θυμίζοντας ότι παράγεται ένας τυχαίος αριθμός (0,1) για κάθε γονίδιο από την ομοιόμορφη κατανομή και αν αυτός ο αριθμός είναι μικρότερος της πιθανότητας μετάλλαξης, τότε η τιμή αυτού του γονιδίου αντικαθίσταται από μία τυχαία τιμή από τον οριοθετημένο συνεχή χώρο αναζήτησης.

Σταθερές που εισάγονται:

- Πιθανότητα διασταύρωσης (**Crossover probability**) $p_c = 0.8$
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) $M = 50$

Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 100$

Παράμετροι που μεταβάλλονται:

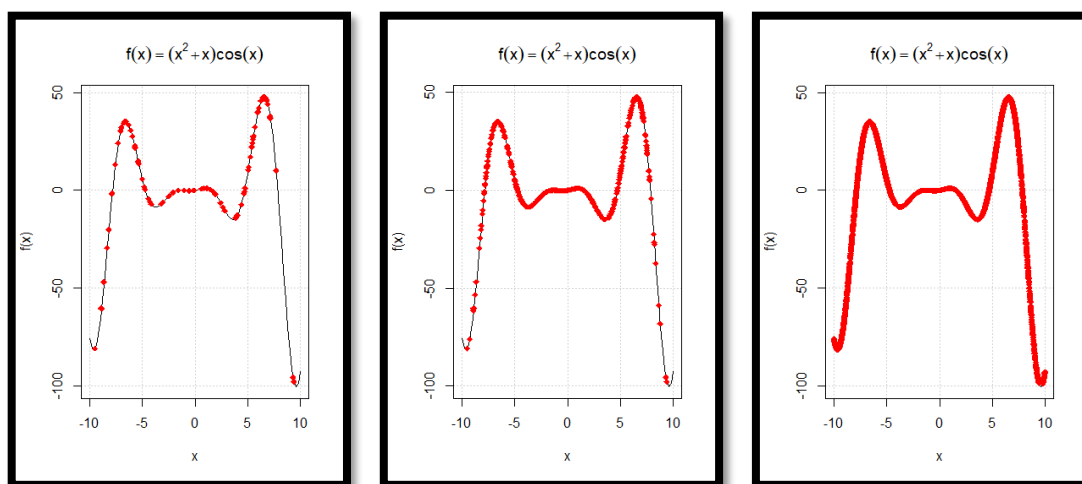
- Πιθανότητες μετάλλαξης (**Mutation probability**) $p_{m1} = 0, p_{m2} = 0.1, p_{m3} = 1.0$

Κώδικας:

```
##### CONTROLLING PARAMETERS OF GA ALGORITHM #####
Problem.nVar <- 1 # Number of problem Variables
M <- 50 # Number of chromosomes (population)
l <- Problem.nVar # Number of genes
MaxGen <- 100 # Generations (iterations)
pc <- 0.8 # Probability of Crossover
pm <- 0, 0.1, 1.0 # Probability of Mutation
Er <- 0.05 # Elitism Ratio
ub <- c(10) # UPPER BOUND FOR SEARCH SPACE
lb <- c(-10) # LOWER BOUND FOR SEARCH SPACE
CGenetic_Algorithm (M, l, MaxGen, pc, pm, Er, lb, ub)
```

Πίνακας 3.5 Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_m

p_m	Μέση τιμή \hat{x}	Τυπική απόκλιση \hat{x}	Βέλτιστη τιμή \hat{x}	Μέση τιμή $\hat{f}(x)$	Τυπική απόκλιση $\hat{f}(x)$	Βέλτιστη τιμή $\hat{f}(x)$
0.0	6.62619	0.0149177	6.561994	47.583	0.0317661	47.70556
0.1	6.31183	1.7514060	6.561112	46.09131	8.211501	47.70561
1.0	-1.620165	5.9545900	6.556982	-2.03429	36.17828	47.70528



Εικόνα 3.4 Διαγράμματα τελικού πληθυσμού του αλγόριθμου για $p_{m1} = 0$, $p_{m2} = 0.1$, $p_{m3} = 1.0$

Η μετάλλαξη αποτελεί τον μηχανισμό με τον οποίο ο αλγόριθμος διευρύνει τον χώρο αναζήτησης, έτσι ώστε κανένα σημείο να μην έχει μηδενική πιθανότητα εξερεύνησης. Αναλύοντας λοιπόν τα αποτελέσματα για, $p_{m1} = 0$, το οποίο πρακτικά σημαίνει ότι κανένα χρωμόσωμα δεν υφίσταται μετάλλαξη, η τοπική αναζήτηση γύρω από την λύση είναι στο μέγιστο βαθμό, ο χώρος αναζήτησης δεν διευρύνεται περαιτέρω και ο τελικός πληθυσμός είναι συγκεντρωμένος τοπικά κάτι που φαίνεται και από την σχεδόν μηδενική τυπική απόκλιση του τελικού πληθυσμού. Στην ακριβώς αντίθετη περίπτωση για $p_{m3} = 1$, η καθολική αναζήτηση είναι στον μέγιστο βαθμό και φαίνεται καθαρά ότι έχει καλυφθεί πλήρως ο χώρος αναζήτησης, ο αλγόριθμος λειτουργεί ουσιαστικά σαν αλγόριθμος τυχαίας αναζήτησης αφού κάθε γονίδιο μεταλλάσσεται τυχαία. Ο τελικός πληθυσμός παρουσιάζει την μεγαλύτερη διασπορά κάτι που φαίνεται από την τιμή της τυπικής απόκλισης που είναι στο μέγιστο βαθμό. Τέλος για $p_{m2} = 0.1$ εξάγεται η βέλτιστη λύση όλων, όπου βάσει των αποτελεσμάτων και των διαγραμμάτων είναι εμφανές ότι για αυτή την τιμή έχουμε την πιο ισορροπημένη ποιότητα αναζήτησης. Παρακάτω θα πραγματοποιηθούν πολλαπλές επαναλήψεις 5×10^2 στον αριθμό, του αλγόριθμου για κάθε επιλογή της πιθανότητας μετάλλαξης p_m και σε πίνακα θα καταγράφονται οι εκτιμήσεις για τη μέση τιμή και τη διασπορά της βέλτιστης λύσης, καθώς και το μέσο τετραγωνικό σφάλμα MSE. Ο αλγόριθμος λοιπόν θα τρέξει με τα παρακάτω δεδομένα, ενώ ο κώδικας του παρουσιάζεται αναλυτικά στο παράρτημα Π1 στο τέλος του συγγράμματος.

Σταθερές που εισάγονται:

- Πιθανότητα διασταύρωσης (**Crossover probability**) $p_c = 0.86$
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) $M = 30$

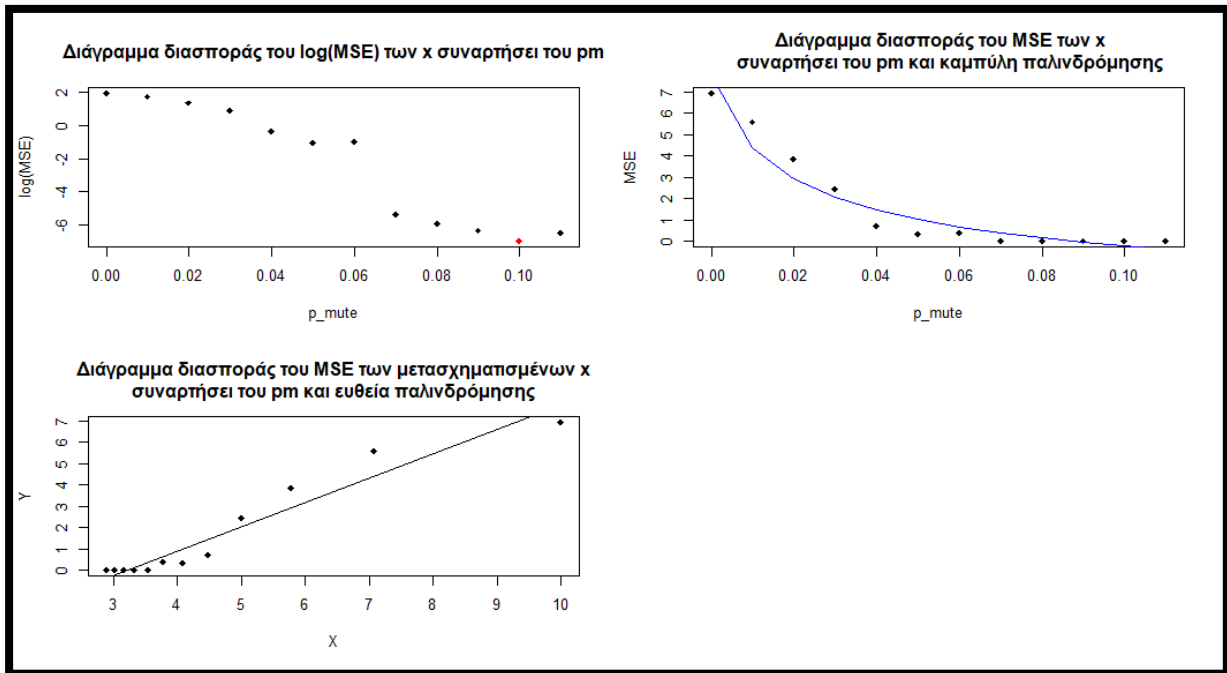
Κριτήρια τερματισμού (convergence):

- Μείστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 30$

Πίνακας 3.6 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου $5 * 10^2$ φορές, για διάφορες τιμές της πιθανότητας μετάλλαξης p_m

p_m	Μέση τιμή \hat{x}	Τυπική α-πόκλιση \hat{x}	Μέση τιμή $\hat{f}(x)$	Τυπική α-πόκλιση $\hat{f}(x)$	MSE (\hat{x})
0.00	5.983495	2.568438	46.56433	3.243110	6.929856
0.01	6.113335	2.316507	46.99357	2.495486	5.566194
0.02	6.245132	1.931420	47.10207	2.271298	3.829863
0.03	6.356684	1.547881	47.34880	1.675191	2.437494
0.04	6.492735	0.835526	47.44737	1.216102	0.702702
0.05	6.516505	0.593267	47.52165	0.922275	0.353905
0.06	6.521102	0.606645	47.55976	0.860905	0.369573
0.07	6.550167	0.067310	47.58947	0.707715	0.004638
0.08	6.551306	0.050967	47.63718	0.357106	0.002682
0.09	6.554608	0.040966	47.66124	0.220216	0.001713
0.10	6.557374	0.0304820	47.68114	0.1648731	0.00093917
0.11	6.555697	0.0382728	47.66987	0.4721207	0.00148825

Παρατηρούμε ότι το καλύτερο αποτέλεσμα καταγράφηκε για τιμή πιθανότητας ίση με 0.10 η οποία είναι εντός του εύρους πιθανοτήτων $p_m \approx 0.08 - 0.1$ που χρησιμοποιούνται συνήθως ως πιθανότητα μετάλλαξης. Ενώ από τα παρακάτω διαγράμματα του μέσου τετραγωνικού σφάλματος MSE φαίνεται ότι το διάστημα $p_m \approx 0.08 - 0.1$ είναι το διάστημα με τις καλύτερες δυνατές επιλογές πιθανοτήτων. Επίσης εκτιμάται με μια καμπύλη παλινδρόμησης το πως μπορεί να σχετίζεται η ποιότητα της λύσης με την μεταβολή της πιθανότητας μετάλλαξης p_m . Ενώ επιπλέον απεικονίζεται και ο μετασχηματισμός που μπορεί να επιτευχθεί στα δεδομένα δημιουργώντας ένα γραμμικό μοντέλο πρόβλεψης.



Εικόνα 3.5 Διαγράμματα διασποράς για διάφορες τιμές της πιθανότητας μετάλλαξης p_m με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης και διάγραμμα με καμπύλη παλινδρόμησης και με ευθεία παλινδρόμησης σε μετασχηματισμένα δεδομένα.

Στην τρίτη περίπτωση θα παραμετροποιηθεί το ποσοστό διατήρησης των ισχυρών χρωμοσωμάτων **Elitism ratio**, στην επόμενη γενιά και θα εξεταστεί η επίδρασή του στην λειτουργία του αλγορίθμου. Επισημαίνεται ότι αυτός ο μηχανισμός βοηθάει την ικανότητα εκμετάλλευσης του αλγορίθμου, δηλαδή την συρρίκνωση του χώρου αναζήτησης γύρω από την βέλτιστη λύση, και ο λόγος που υφίσταται ως μηχανισμός είναι ότι κατά την διάρκεια των διαδικασιών της επιλογής, της διασταύρωσης και της μετάλλαξης καλές υφιστάμενες λύσεις μπορεί να χαθούν.

Σταθερές που εισάγονται:

- Πιθανότητα διασταύρωσης (**Crossover probability**) $p_c = 0.8$
- Πιθανότητες μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Μέγεθος πληθυσμού (**Population size**) $M = 50$

Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 100$

Παράμετροι που μεταβάλλονται:

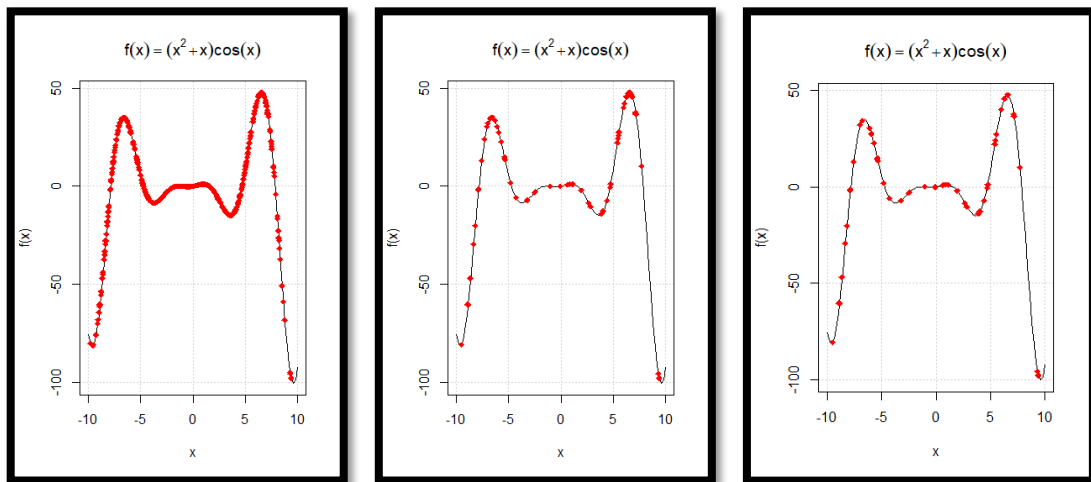
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_{r1} = 0, E_{r2} = 0.5, E_{r3} = 1.0$

Κώδικας:

```
#### CONTROLLING PARAMETERS OF GA ALGORITHM #####
Problem.nVar <- 1 # Number of problem Variables
M <- 50 # Number of chromosomes (population)
l <- Problem.nVar # Number of genes
MaxGen <- 100 # Generations (iterations)
pc <- 0.8 # Probability of Crossover
pm <- 0.1 # Probability of Mutation
Er <- 0,0.5,1.0 # Elitism Ratio
ub <- c(10) # UPPER BOUND FOR SEARCH SPACE
lb <- c(-10) # LOWER BOUND FOR SEARCH SPACE
CGenetic_Algorithm (M, l, MaxGen, pc, pm, Er, lb, ub)
```

Πίνακας 3.7. Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της E_r

E_r	Μέση τιμή \hat{x}	Τυπική απόκλιση \hat{x}	Βέλτιστη τιμή \hat{x}	Μέση τιμή $\hat{f}(x)$	Τυπική απόκλιση $\hat{f}(x)$	Βέλτιστη τιμή $\hat{f}(x)$
0.0	5.775028	3.252864	6.562159	38.37311	29.46339	47.70555
0.5	6.56054	5.5276e-09	6.5605	47.70562	2.6855e-15	47.70562
1.0	-0.5158613	5.868707	6.564874	-0.3648077	33.43383	47.70512



Εικόνα 3.6 Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $E_{r1} = 0$, $E_{r2} = 0.5$, $E_{r3} = 1.0$

Αναλύοντας τα αποτελέσματα, για $E_{r1} = 0$ το οποίο σημαίνει ότι καμία καλή λύση δεν σώζεται ώστε να μεταφερθεί στην επόμενη γενιά, ο αλγόριθμος δουλεύει κανονικά με τους άλλους μηχανισμούς και δίνει μια καλή λύση αλλά όχι την καλύτερη. Οι μηχανισμοί που λειτουργούν σε αυτήν την περίπτωση είναι κατά βάση οι μηχανισμοί εξερεύνησης, κάτι το οποίο φαίνεται από την διασπορά του πληθυσμού ο οποίος έχει καλύψει σχετικά πλήρως την περιοχή αναζήτησης. Αντίθετα για $E_{r3} = 1$ όλες οι αρχικές λύσεις μεταφέρονται από γενιά σε γενιά και ουσιαστικά η λύση που δίνεται είναι αυτή του τυχαίου αρχικού πληθυσμού και η χειρότερη. Τέλος για $E_{r2} = 0.5$, παρατηρείται μεγάλη συγκέντρωση του πληθυσμού γύρω από τη βέλτιστη λύση

κάτι που εξηγεί και τις μηδενικές τυπικές αποκλίσεις στον τελικό πληθυσμό. Ουσιαστικά αυτό που συμβαίνει είναι ότι μόλις βρεθεί ένα ισχυρό (**elite**) χρωμόσωμα πολύ κοντά στη βέλτιστη λύση όλες οι λύσεις κατευθύνονται προς τα εκεί, και με τέτοιο ποσοστό διατήρησης καλών λύσεων της τάξης του 50%, όλες οι υπόλοιπες διεργασίες της επιλογής, της διασταύρωσης και της μετάλλαξης γίνονται μεταξύ Elite χρωμοσωμάτων. Το θετικό είναι ότι ο αλγόριθμος μπορεί να συγκλίνει γρήγορα προς μία πολύ καλή λύση καθώς βελτιώνει πολύ την ικανότητα εκμετάλλευσής του, όμως η λύση αυτή μπορεί και να μην είναι η ολικά βέλτιστη καθώς χειροτερεύει η ικανότητα εξερεύνησης του, συρρικνώνεται δηλαδή αρκετά γρήγορα ο χώρος αναζήτησης και μπορεί περιοχές με καλύτερες λύσεις να μένουν ανεξερεύνητες. Παρακάτω θα πραγματοποιηθούν πολλαπλές επαναλήψεις 5×10^2 στον αριθμό, του αλγορίθμου για κάθε επιλογή του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r , και σε πίνακα θα καταγράφονται οι εκτιμήσεις για τη μέση τιμή και τη διασπορά της βέλτιστης λύσης, καθώς και το μέσο τετραγωνικό σφάλμα MSE. Ο αλγόριθμος λοιπόν θα τρέξει με τα παρακάτω δεδομένα, ενώ ο κώδικας του παρουσιάζεται αναλυτικά στο παράρτημα Π1 στο τέλος του συγγράμματος.

Σταθερές που εισάγονται:

- Πιθανότητα διασταύρωσης (**Crossover probability**) $p_c = 0.86$
- Πιθανότητες μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Μέγεθος πληθυσμού (**Population size**) $M = 30$

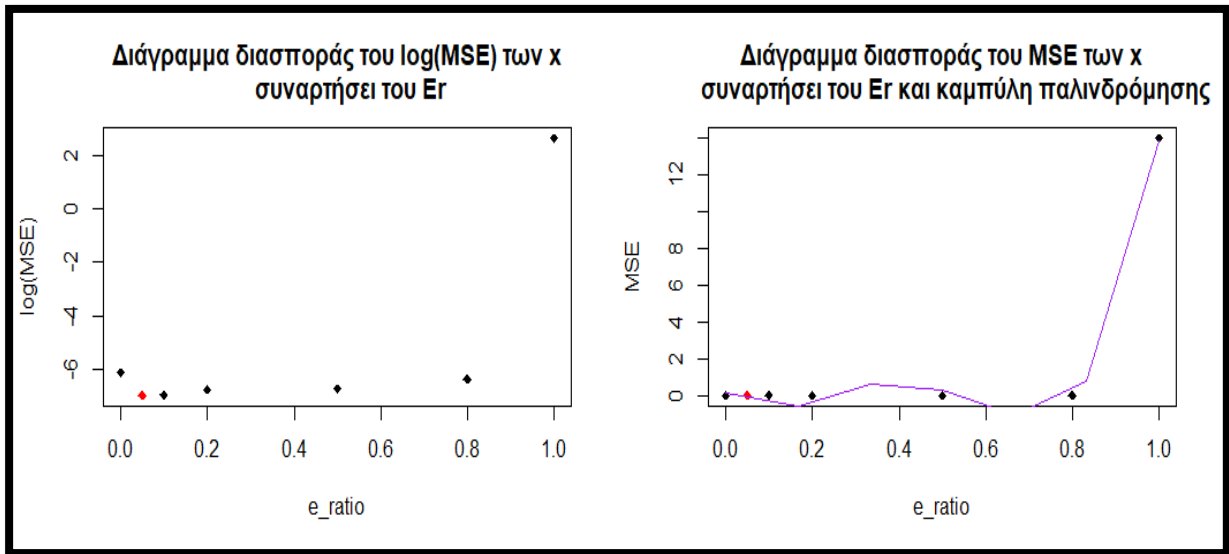
Κριτήρια τερματισμού (convergence):

- Μειστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 30$

Πίνακας 3.8 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγορίθμου $5 * 10^2$ φορές, για διάφορες τιμές του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r .

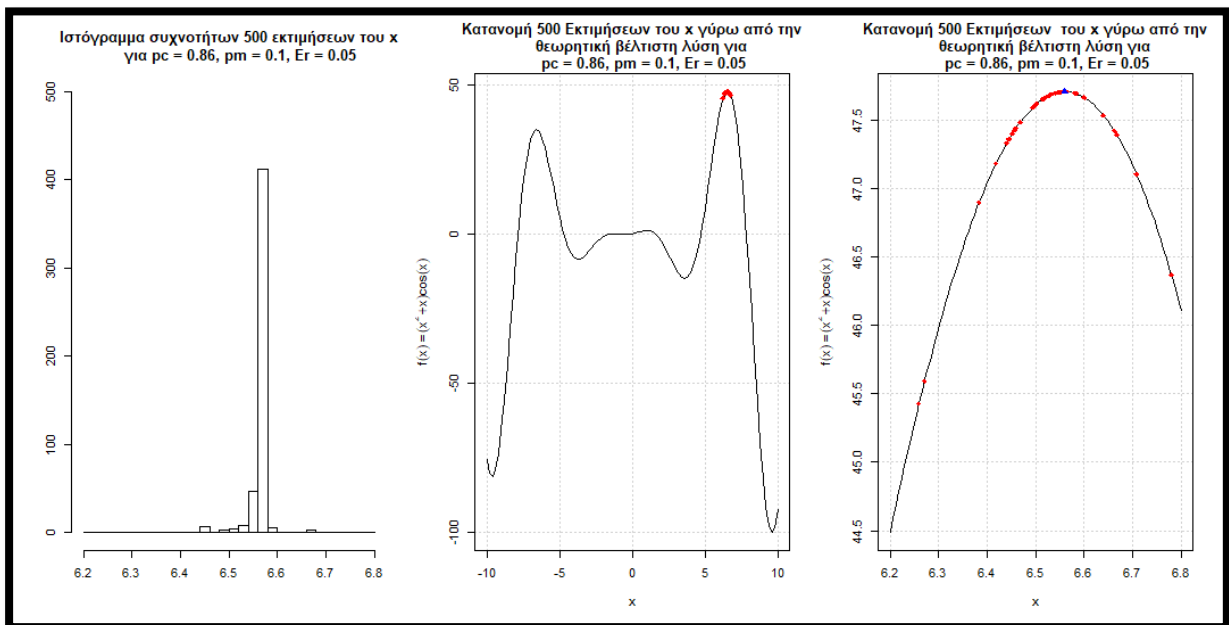
E_r	Μέση τιμή \hat{x}	Τυπική α-πόκλιση \hat{x}	Μέση τιμή $\hat{f}(x)$	Τυπική α-πόκλιση $\hat{f}(x)$	MSE (\hat{x})
0.00	6.550708	0.0455984	47.64989	0.2868165	0.00217587
0.05	6.557374	0.0304820	47.68114	0.1648731	0.00093917
0.10	6.555121	0.0303187	47.68094	0.1321167	0.00094858
0.20	6.557423	0.0340496	47.67552	0.2086110	0.00116908
0.50	6.556247	0.0342580	47.67431	0.1508950	0.00119203
0.80	6.555858	0.0412811	47.66060	0.2973181	0.00172603
1.00	5.479386	3.5765680	44.24901	4.6074750	13.9607300

Παρατηρούμε ότι το καλύτερο αποτέλεσμα εξάγεται για τιμή ποσοστού διατήρησης ισχυρών χρωμοσωμάτων $E_r = 5\%$. Επίσης εκτιμάται με μια καμπύλη παλινδρόμησης το πως μπορεί να σχετίζεται η ποιότητα της λύσης με την μεταβολή του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r .



Εικόνα 3.7 Διαγράμματα διασποράς για διάφορες τιμές του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r , με το $\log(MSE)$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης, και διάγραμμα με καμπύλη παλινδρόμησης

Παρακάτω δίνονται τα διαγράμματα και το ιστόγραμμα με τις 5×10^2 στον αριθμό, εκτιμήσεις των λύσεων για $p_c = 0.86$, $p_m = 0.10$, $E_r = 0.05$, από τις οποίες προέκυψε η βέλτιστη λύση και η οποία ήταν για $\hat{x} = 6.557374$ και ολικό μέγιστο $\hat{f}(x) = 47.68114$.



Εικόνα 3.8 Ιστόγραμμα και διαγράμματα με τις εκτιμήσεις των λύσεων όπως προκύπτουν από 5×10^2 επαναλήψεις του αλγορίθμου για $p_c = 0.86$, $p_m = 0.10$, $E_r = 0.05$

Τέλος να επισημάνουμε ότι οι εκτιμήσεις 5×10^2 στον αριθμό, της βέλτιστης λύσης για κάθε επιλογή της πιθανότητας διασταύρωσης p_c , της πιθανότητας μετάλλαξης p_m , και του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r πραγματοποιήθηκαν σε σύστημα Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz (4 CPUs), ~3.0GHz με 8192MB RAM, και ο μέσος χρόνος εκτέλεσης για την κάθε μία ήταν περίπου **7 λεπτά**.

3.4.3 Παράδειγμα 2 – Δισδιάστατη μεταβλητή

Στο δεύτερο παράδειγμα θα τρέξει ο αλγόριθμος ώστε να βρει το ολικό μέγιστο μιας συνάρτησης που κατασκευάσαμε. Η μαθηματική έκφραση της συνάρτησης f είναι:

$$f(x_1, x_2) = p f_1(x_1, x_2) + (1 - p)f_2(x_1, x_2)$$

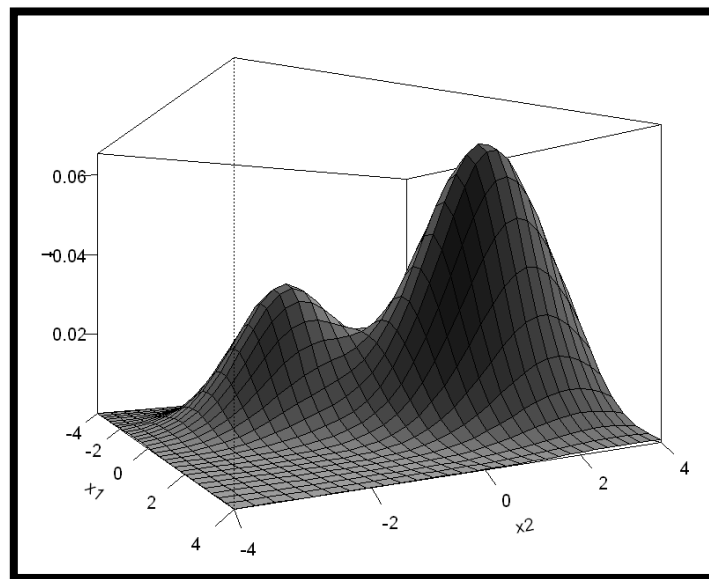
Όπου

$$f_1(x_1, x_2) = \frac{1}{2\pi} \exp \left(-\frac{1}{2} [(x_1 + 1)^2 + (x_2 + 1)^2] \right)$$

$$f_2(x_1, x_2) = \frac{1}{\pi\sqrt{15}} \exp \left(-\frac{8}{15} \left[(x_1 - 1)^2 + \left(\frac{x_2 - 2.5}{2} \right)^2 - \frac{1}{4} (x_1 - 1)(x_2 - 2.5) \right] \right)$$

και $p = 0.2$.

Η ανωτέρω συνάρτηση στο χωρίο $[-4, 4] \times [-4, 4]$, παρουσιάζει τοπικό μέγιστο στο σημείο $(-0.955467, -0.956622)$, και ολικό μέγιστο ίσο με 0.065759 στο σημείο $(0.999599, 2.498955)$. Ενώ η απεικόνιση της στον χώρο δίνεται παρακάτω:



Εικόνα 3.9 Η τρισδιάστατη αναπαράσταση της $f(x)$, όπου διακρίνονται δύο κορυφές.

Ο αλγόριθμός λοιπόν για αυτή την περίπτωση θα τρέξει με τους παρακάτω μηχανισμούς, ενώ ο κώδικας παρουσιάζεται αναλυτικά στο παράρτημα Π2 στο τέλος του συγγράμματος.

Επιλογή μεθόδων βασικών μηχανισμών λειτουργίας:

- **Μέθοδος επιλογής:** Roulette wheel selection
- **Μέθοδος Διασταύρωσης:** Arithmetic crossover
- **Μέθοδος Μετάλλαξης:** Uniform mutation

3.4.4 Εκτέλεση κώδικα Γενετικού Αλγόριθμου (GA) δισδιάστατης μεταβλητής:

Στη συνέχεια θα εκτελεστεί ο κώδικας για μία επανάληψη του αλγόριθμου για διάφορες τιμές των σημαντικότερων παραμέτρων του, και θα μελετηθεί μέσω του τελικού πληθυσμού η επίδραση αυτών στην απόδοση του αλγόριθμου, όπως ακριβώς και στην προηγούμενη περίπτωση της μονοδιάστατης μεταβλητής.

Στην πρώτη λοιπόν περίπτωση θα παραμετροποιηθεί η πιθανότητα διασταύρωσης.

Σταθερές που εισάγονται:

- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) $M = 50$
- Χωρίο αναζήτησης $[-4, 4] \times [-4, 4]$

Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 100$

Παράμετροι που μεταβάλλονται:

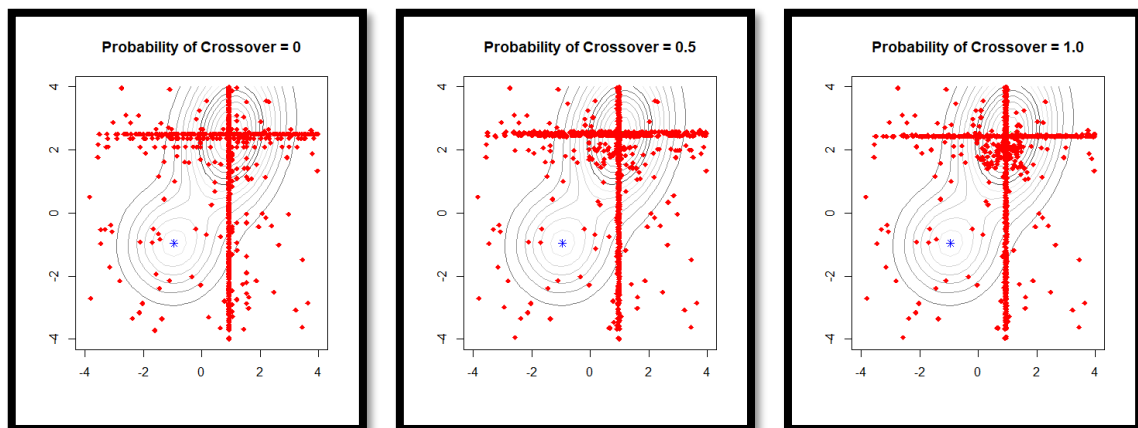
- Πιθανότητες διασταύρωσης (**Crossover probability**) $p_{c1} = 0, p_{c2} = 0.5, p_{c3} = 1.0$

Κώδικας:

```
##### CONTROLLING PARAMETERS OF GA ALGORITHM #####
Problem.nVar <- 2 # Number of problem Variables
M <- 50 # Number of chromosomes (population)
l <- Problem.nVar # Number of genes
MaxGen <- 100 # Generations (iterations)
pc <- 0,0.5,1.0 # Probability of Crossover
pm <- 0.1 # Probability of Mutation
Er <- 0.05 # Elitism Ratio
ub <- c(4,4) # UPPER BOUND FOR SEARCH SPACE
lb <- c(-4,-4) # LOWER BOUND FOR SEARCH SPACE
CGenetic_Algorithm (M, l, MaxGen, pc, pm, Er, lb, ub)
```

Πίνακας 3.9 Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_c

p_c	Μέση τιμή \hat{x}_1	Τυπική απόκλιση \hat{x}_1	Μέση τιμή \hat{x}_2	Τυπική απόκλιση \hat{x}_2	Βέλτιστη τιμή \hat{x}_1	Βέλτιστη τιμή \hat{x}_2	Μέση τιμή $\hat{f}(x)$	Τυπική απόκλιση $\hat{f}(x)$	Βέλτιστη τιμή $\hat{f}(x)$
0.0	0.85708	0.65188	2.49192	0.20091	1.04653	2.49723	0.06013	0.01667	0.06568
0.5	0.97440	0.61537	2.39605	0.53340	0.99767	2.49914	0.06061	0.01458	0.06575
1.0	1.03472	0.50590	2.47395	0.11741	0.97232	2.49370	0.06144	0.01377	0.06573



Εικόνα 3.10 Διαγράμματα τελικού πληθυσμού του αλγόριθμου για $p_{c1} = 0$, $p_{c2} = 0.5$, $p_{c3} = 1.0$

Αναλύοντας τα παραπάνω φαίνεται και σε αυτήν την περίπτωση ότι για $p_{c2} = 0.5$ εξάγονται τα καλύτερα αποτελέσματα καθώς υπάρχει καλύτερη ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης.

Παρακάτω θα πραγματοποιηθούν πολλαπλές επαναλήψεις 10^2 στον αριθμό, του αλγορίθμου, για κάθε επιλογή της πιθανότητας διασταύρωσης p_c και σε πίνακα θα καταγράφονται οι εκτιμήσεις για τη μέση τιμή και τη διασπορά της βέλτιστης λύσης, καθώς και το μέσο τετραγωνικό σφάλμα MSE. Σε αυτή την περίπτωση ως MSE θεωρείται η μέση τετραγωνική απόσταση της εκτίμησης από το βέλτιστο σημείο, στο επίπεδο, δηλαδή $MSE1 + MSE2 = E(d^2)$, όπου

$$d = \sqrt{(\hat{X}_1 - x_1)^2 + (\hat{X}_2 - x_2)^2}$$

Ο αλγόριθμος λοιπόν θα τρέξει με τα παρακάτω δεδομένα, ενώ ο κώδικας του παρουσιάζεται αναλυτικά στο παράρτημα Π2 στο τέλος του συγγράμματος.

Σταθερές που εισάγονται:

- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) $M = 40$

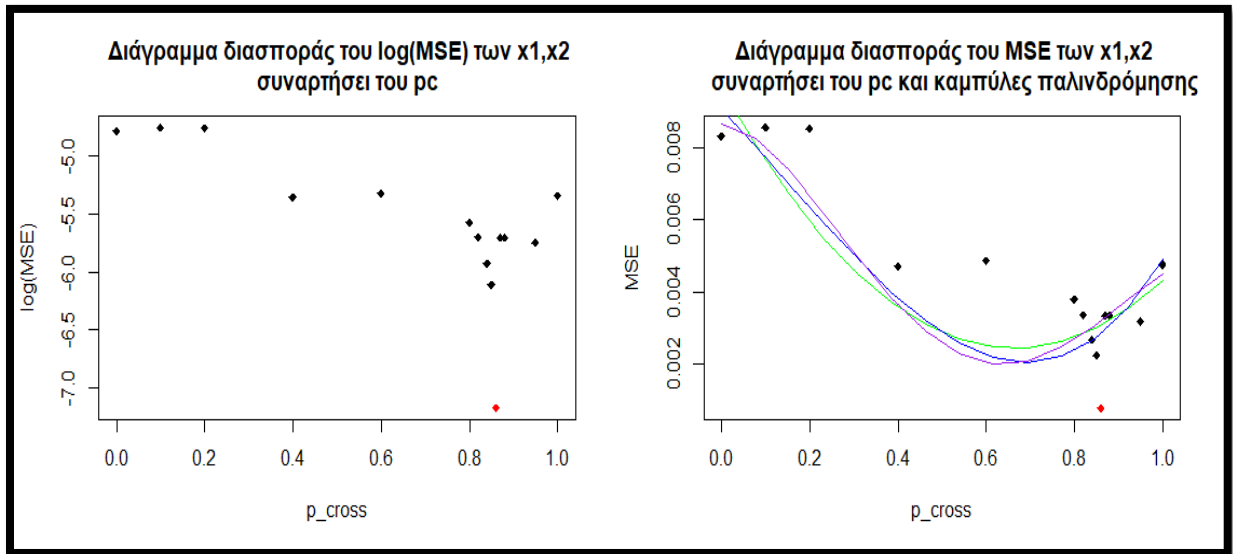
Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 40$

Πίνακας 3.10 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου 10^2 φορές, για διάφορες τιμές της πιθανότητας διασταύρωσης p_c

p_c	Μέση τιμή \hat{x}_1	Τυπική α-πόκλιση \hat{x}_1	Μέση τιμή \hat{x}_2	Τυπική α-πόκλιση \hat{x}_2	Μέση τιμή $\hat{f}(x_1, x_2)$	Τυπική α-πόκλιση $\hat{f}(x_1, x_2)$	MSE (\hat{x}_1, \hat{x}_2)
0.00	1.005426	0.0461981	2.504915	0.0489966	0.0656491	0.00015797	0.00833536
0.10	0.9987564	0.0267886	2.507490	0.0368739	0.0657116	6.86569e-05	0.00857615
0.20	1.0000650	0.0415932	2.507474	0.0370061	0.0656858	9.92774e-05	0.00853147
0.40	0.9964449	0.0244056	2.495452	0.0256939	0.0657290	9.60064e-05	0.00471381
0.60	0.9976997	0.0174991	2.494476	0.0206864	0.0657426	4.15124e-05	0.00486477
0.80	0.9974609	0.0118017	2.495820	0.0169805	0.0657500	1.96141e-05	0.00379453
0.82	0.9970519	0.0104859	2.496783	0.0131566	0.0657527	1.75099e-05	0.00334768
0.84	0.9972913	0.0111049	2.497625	0.0140080	0.0657520	1.89742e-05	0.00266367
0.85	0.9980269	0.0119009	2.497390	0.0115492	0.0657525	2.46507e-05	0.00221851
0.86	0.9988304	0.0111314	2.498923	0.0118555	0.0657532	1.73625e-05	0.00076927
0.87	0.9980021	0.0137689	2.496030	0.0175594	0.0657482	3.50002e-05	0.00333253
0.88	0.9983114	0.0172108	2.495881	0.0183871	0.0657442	5.54298e-05	0.00333277
0.95	0.9966690	0.0162724	2.497729	0.0148616	0.0657470	3.17126e-05	0.00317604
1.00	0.9987591	0.0098866	2.494270	0.0214672	0.0657479	3.84624e-05	0.00475999

Από τον πίνακα και τα διαγράμματα του μέσου τετραγωνικού σφάλματος MSE που ακολουθούν φαίνεται ότι το διάστημα $p_c \approx 0.8 - 1.0$ είναι το διάστημα με τις καλύτερες δυνατές επιλογές πιθανοτήτων. Επίσης εκτιμάται με διαφορετικές καμπύλες παλινδρόμησης το πως μπορεί να σχετίζεται η ποιότητα της λύσης με την μεταβολή της πιθανότητας διασταύρωσης p_c .



Εικόνα 3.11 Διαγράμματα διασποράς για διάφορες τιμές της πιθανότητας διασταύρωσης p_c , με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης και καμπύλες παλινδρόμησης.

Συνεχίζοντας στην δεύτερη περίπτωση θα παραμετροποιηθεί η πιθανότητα μετάλλαξης.

Σταθερές που εισάγονται:

- Πιθανότητα διασταύρωσης (**Crossover probability**) $p_c = 0.8$
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) $M = 50$
- Χωρίο αναζήτησης $[-4, 4] \times [-4, 4]$

Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 100$

Παράμετροι που μεταβάλλονται:

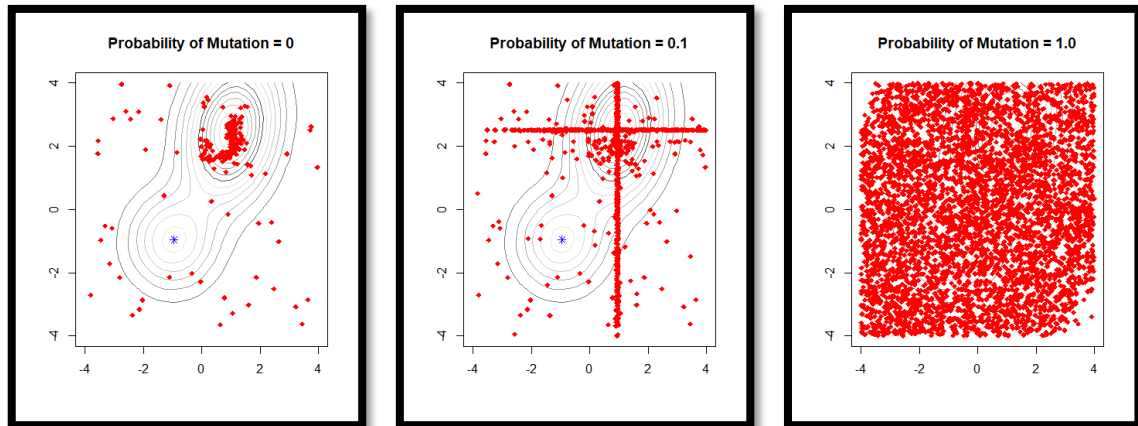
- Πιθανότητες μετάλλαξης (**Mutation probability**) $p_{m1} = 0, p_{m2} = 0.1, p_{m3} = 1.0$

Κώδικας:

```
#### CONTROLLING PARAMETERS OF GA ALGORITHM #####
Problem.nVar <- 2 # Number of problem Variables
M <- 50 # Number of chromosomes (population)
l <- Problem.nVar # Number of genes
MaxGen <- 100 # Generations (iterations)
pc <- 0.8 # Probability of Crossover
pm <- 0,0.1,1.0 # Probability of Mutation
Er <- 0.05 # Elitism Ratio
ub <- c(4,4) # UPPER BOUND FOR SEARCH SPACE
lb <- c(-4,-4) # LOWER BOUND FOR SEARCH SPACE
CGenetic_Algorithm (M, l, MaxGen, pc, pm, Er, lb, ub)
```

Πίνακας 3.11 Τα αποτελέσματα από μία εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_m

p_m	Μέση τιμή \hat{x}_1	Τυπική απόκλιση \hat{x}_1	Μέση τιμή \hat{x}_2	Τυπική απόκλιση \hat{x}_2	Βέλτιστη τιμή \hat{x}_1	Βέλτιστη τιμή \hat{x}_2	Μέση τιμή $\hat{f}(x)$	Τυπική απόκλιση $\hat{f}(x)$	Βέλτιστη τιμή $\hat{f}(x)$
0.0	1.02755	0.00552	2.40352	0.01917	1.00076	2.49652	0.06554	4.49e-05	0.06575
0.1	0.98367	0.40746	2.49122	0.22909	0.99705	2.49966	0.06245	0.01161	0.065759
1.0	0.39871	2.28008	0.02550	2.41129	1.01635	2.43032	0.01195	0.01745	0.06565



Εικόνα 3.12 Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $p_{m1} = 0$, $p_{m2} = 0.1$, $p_{m3} = 1.0$

Αναλύοντας λοιπόν τα αποτελέσματα επιβεβαιώνεται και πάλι ότι για $p_{m1} = 0$, η τοπική αναζήτηση γύρω από την λύση είναι στο μέγιστο βαθμό, και ο τελικός πληθυσμός παρουσιάζει μηδενική διασπορά. Στην ακριβώς αντίθετη περίπτωση για $p_{m3} = 1$, η καθολική αναζήτηση είναι στον μέγιστο βαθμό και φαίνεται καθαρά ότι έχει καλυφθεί πλήρως ο χώρος αναζήτησης, η διασπορά του τελικού πληθυσμού είναι στο μέγιστο βαθμό και ο αλγόριθμος λειτουργεί ουσιαστικά σαν αλγόριθμος τυχαία αναζήτησης αφού κάθε γονίδιο μεταλλάσσεται τυχαία. Τέλος για $p_{m2} = 0.1$, εξάγεται η βέλτιστη λύση καθώς για αυτή την τιμή έχουμε την πιο ισορροπημένη ποιότητα αναζήτησης. Παρακάτω θα πραγματοποιηθούν πολλαπλές επαναλήψεις 10^2 στον αριθμό, του αλγορίθμου για κάθε επιλογή της πιθανότητας μετάλλαξης p_m και σε πίνακα θα καταγράφονται οι εκτιμήσεις για τη μέση τιμή και τη διασπορά της βέλτιστης λύσης, καθώς και το μέσο τετραγωνικό σφάλμα MSE. Ο αλγόριθμος θα τρέξει με τα παρακάτω δεδομένα, ενώ ο κώδικας του παρουσιάζεται αναλυτικά στο παράρτημα Π2 στο τέλος του συγγράμματος.

Σταθερές που εισάγονται:

- Πιθανότητα διασταύρωσης (**Crossover probability**) $p_c = 0.86$
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) $M = 40$

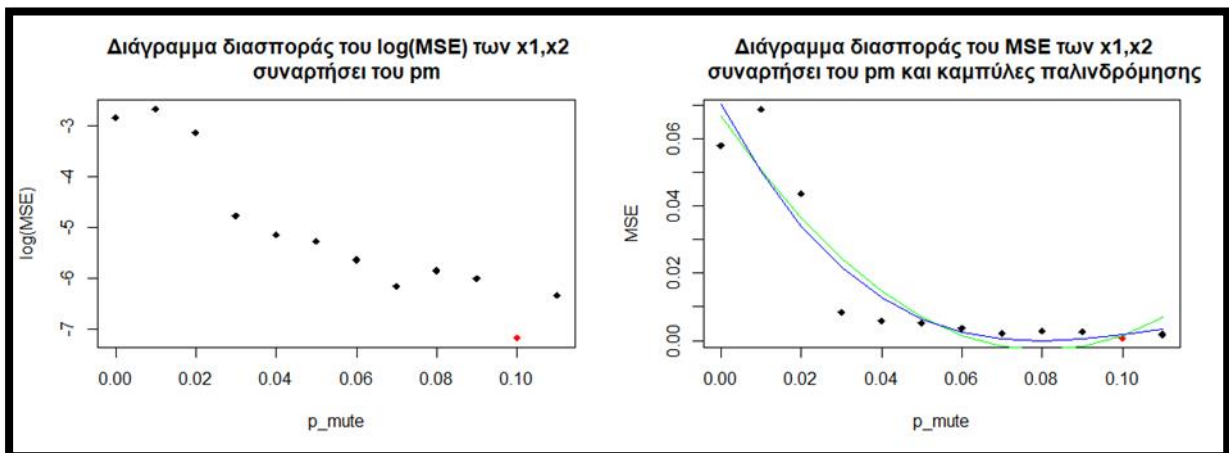
Κριτήρια τερματισμού (convergence):

- Μέγιστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 40$

Πίνακας 3.12 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου 10^2 φορές, για διάφορες τιμές της πιθανότητας μετάλλαξης p_m

p_m	Μέση τιμή \hat{x}_1	Τυπική α-πόκλιση \hat{x}_1	Μέση τιμή \hat{x}_2	Τυπική α-πόκλιση \hat{x}_2	Μέση τιμή $\hat{f}(x_1, x_2)$	Τυπική α-πόκλιση $\hat{f}(x_1, x_2)$	MSE (\hat{x}_1, \hat{x}_2)
0.00	0.9620979	0.1487852	2.454838	0.1615451	0.0647017	0.00374610	0.05790189
0.01	0.9646045	0.2007799	2.439838	0.3573296	0.0651127	0.00340632	0.06869783
0.02	0.9773614	0.1995501	2.461484	0.3474452	0.0653605	0.00331107	0.04357312
0.03	0.9935312	0.0282670	2.493110	0.0261224	0.0657215	9.80983e-05	0.00842478
0.04	0.9981846	0.0236373	2.493291	0.0244234	0.0657312	7.79429e-05	0.00583836
0.05	0.9973739	0.0211641	2.494304	0.0234923	0.0657367	7.14312e-05	0.00515544
0.06	0.9960743	0.0170582	2.498439	0.0241946	0.0657405	5.45302e-05	0.00356237
0.07	0.9979716	0.0156018	2.497600	0.0170187	0.0657468	3.64047e-05	0.00211789
0.08	0.9980787	0.0113828	2.496510	0.0154645	0.0657512	2.51655e-05	0.00287929
0.09	0.9977139	0.0136505	2.497363	0.0131890	0.0657502	3.15183e-05	0.00246721
0.10	0.9988304	0.0111314	2.498923	0.0118555	0.0657532	1.73625e-05	0.00076927
0.11	0.9983746	0.0122204	2.497673	0.0169609	0.0657498	2.4308e-05	0.00177242

Παρατηρούμε ότι το καλύτερο αποτέλεσμα εξάγεται για τιμή πιθανότητας ίση με 0.10, ενώ από τα παρακάτω διαγράμματα του μέσου τετραγωνικού σφάλματος MSE που ακολουθούν φαίνεται ότι το διάστημα $p_m \approx 0.08 - 0.1$ είναι το διάστημα με τις καλύτερες δυνατές επιλογές πιθανοτήτων. Επίσης εκτιμάται με διαφορετικές καμπύλες παλινδρόμησης το πως μπορεί να σχετίζεται η ποιότητα της λύσης με την μεταβολή της πιθανότητας μετάλλαξης p_m .



Εικόνα 3.13 Διαγράμματα διασποράς για διάφορες τιμές της της πιθανότητας μετάλλαξης p_m , με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης, και καμπύλες παλινδρόμησης.

Στην τρίτη περίπτωση θα παραμετροποιηθεί το ποσοστό διατήρησης των ισχυρών χρωμοσωμάτων σε επόμενη γενιά δηλαδή το **Elitism ratio** και θα εξεταστεί η επίδρασή του στην λειτουργία του αλγορίθμου.

Σταθερές που εισάγονται:

- Πιθανότητα διασταύρωσης (**Crossover probability**) $p_c = 0.8$
- Πιθανότητες μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Μέγεθος πληθυσμού (**Population size**) $M = 50$
- Χωρίο αναζήτησης $[-4, 4] \times [-4, 4]$

Κριτήρια τερματισμού (convergence):

- Μειστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 100$

Παράμετροι που μεταβάλλονται:

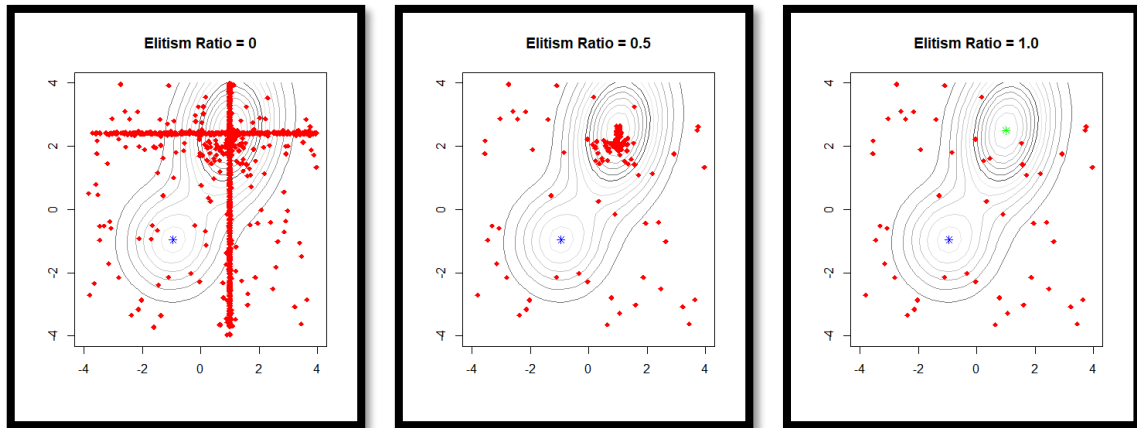
- Ποσοστό διατήρησης Ισχυρών Χρωμοσωμάτων (**Elitism Ratio**) $E_{r1} = 0, E_{r2} = 0.5, E_{r3} = 1.0$

Κώδικας:

```
#### CONTROLLING PARAMETERS OF GA ALGORITHM #####
Problem.nVar <- 2 # Number of problem Variables
M <- 50 # Number of chromosomes (population)
l <- Problem.nVar # Number of genes
MaxGen <- 100 # Generations (iterations)
pc <- 0.8 # Probability of Crossover
pm <- 0.1 # Probability of Mutation
Er <- 0,0.5,1.0 # Elitism Ratio
ub <- c(4,4) # UPPER BOUND FOR SEARCH SPACE
lb <- c(-4,-4) # LOWER BOUND FOR SEARCH SPACE
CGenetic_Algorithm (M, l, MaxGen, pc, pm, Er, lb, ub)
```

Πίνακας 3.13. Τα αποτελέσματα από την εκτέλεση του γενετικού αλγορίθμου, για διάφορες τιμές του E_r

E_r	Μέση τιμή \hat{x}_1	Τυπική απόκλιση \hat{x}_1	Μέση τιμή \hat{x}_2	Τυπική απόκλιση \hat{x}_2	Βέλτιστη τιμή \hat{x}_1	Βέλτιστη τιμή \hat{x}_2	Μέση τιμή $\hat{f}(x)$	Τυπική απόκλιση $\hat{f}(x)$	Βέλτιστη τιμή $\hat{f}(x)$
0.0	1.03662	0.53614	1.98359	1.35961	1.00151	2.49966	0.05682	0.01987	0.065759
0.5	0.99959	0	2.49895	0	0.99959	2.49895	0.06575	0	0.065759
1.0	-0.0994	2.39519	0.00908	2.40148	1.53713	2.09588	0.00868	0.01393	0.052441



Εικόνα 3.14 Διαγράμματα τελικού πληθυσμού του αλγορίθμου για $E_{r1} = 0$, $E_{r2} = 0.5$, $E_{r3} = 1.0$

Αναλύοντας τα αποτελέσματα για $E_{r1} = 0$, ο αλγόριθμος δουλεύει κανονικά με τους άλλους μηχανισμούς και δίνει μια πολύ καλή λύση. Αντίθετα για $E_{r3} = 1$ όλες οι αρχικές λύσεις μεταφέρονται από γενιά σε γενιά και ουσιαστικά η λύση που δίνεται είναι αυτή του τυχαίου αρχικού πληθυσμού και η χειρότερη όλων. Τέλος για $E_{r2} = 0.5$, αυτό που συμβαίνει είναι η μεγάλη συγκέντρωση του πληθυσμού γύρω από τη βέλτιστη λύση κάτι που εξηγεί και την μηδενική διασπορά. Παρακάτω θα πραγματοποιηθούν πολλαπλές επαναλήψεις 10^2 στον αριθμό, του αλγορίθμου, για κάθε επιλογή του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r , και σε πίνακα θα καταγράφονται οι εκτιμήσεις για τη μέση τιμή και τη διασπορά της βέλτιστης λύσης, καθώς και το μέσο τετραγωνικό σφάλμα MSE. Ο αλγόριθμος θα τρέξει με τα παρακάτω δεδομένα, ενώ ο κώδικας του παρουσιάζεται αναλυτικά στο παράρτημα Π2 στο τέλος του συγγράμματος.

Σταθερές που εισάγονται:

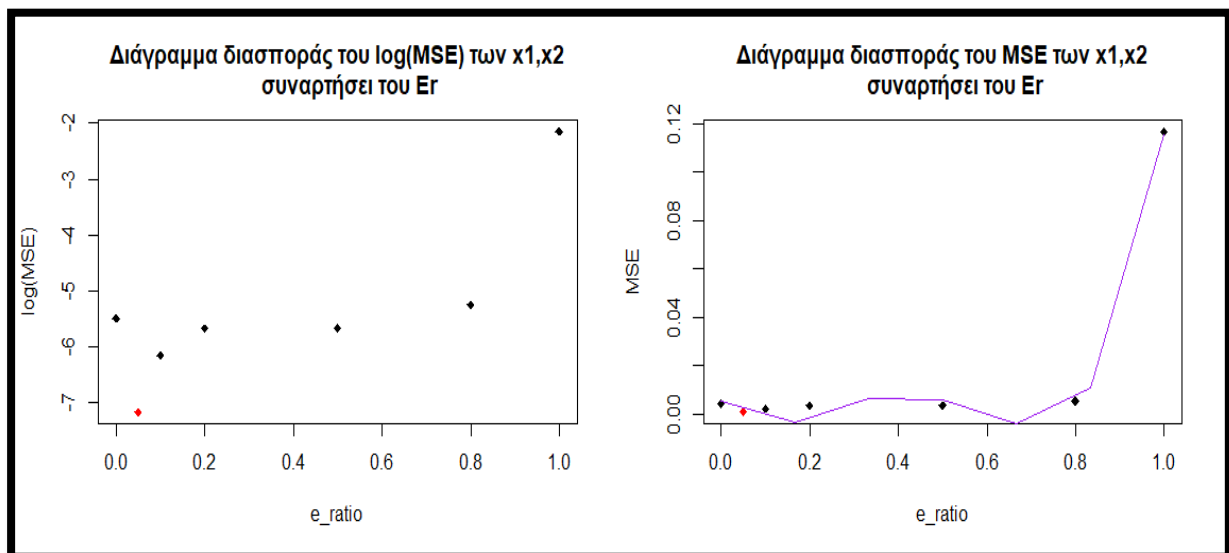
- Πιθανότητα διασταύρωσης (**Crossover probability**) $p_c = 0.86$
- Πιθανότητες μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Μέγεθος πληθυσμού (**Population size**) $M = 40$

Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) $MaxGen = 40$

Πίνακας 3.14 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου 10^2 φορές, για διάφορες τιμές του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r

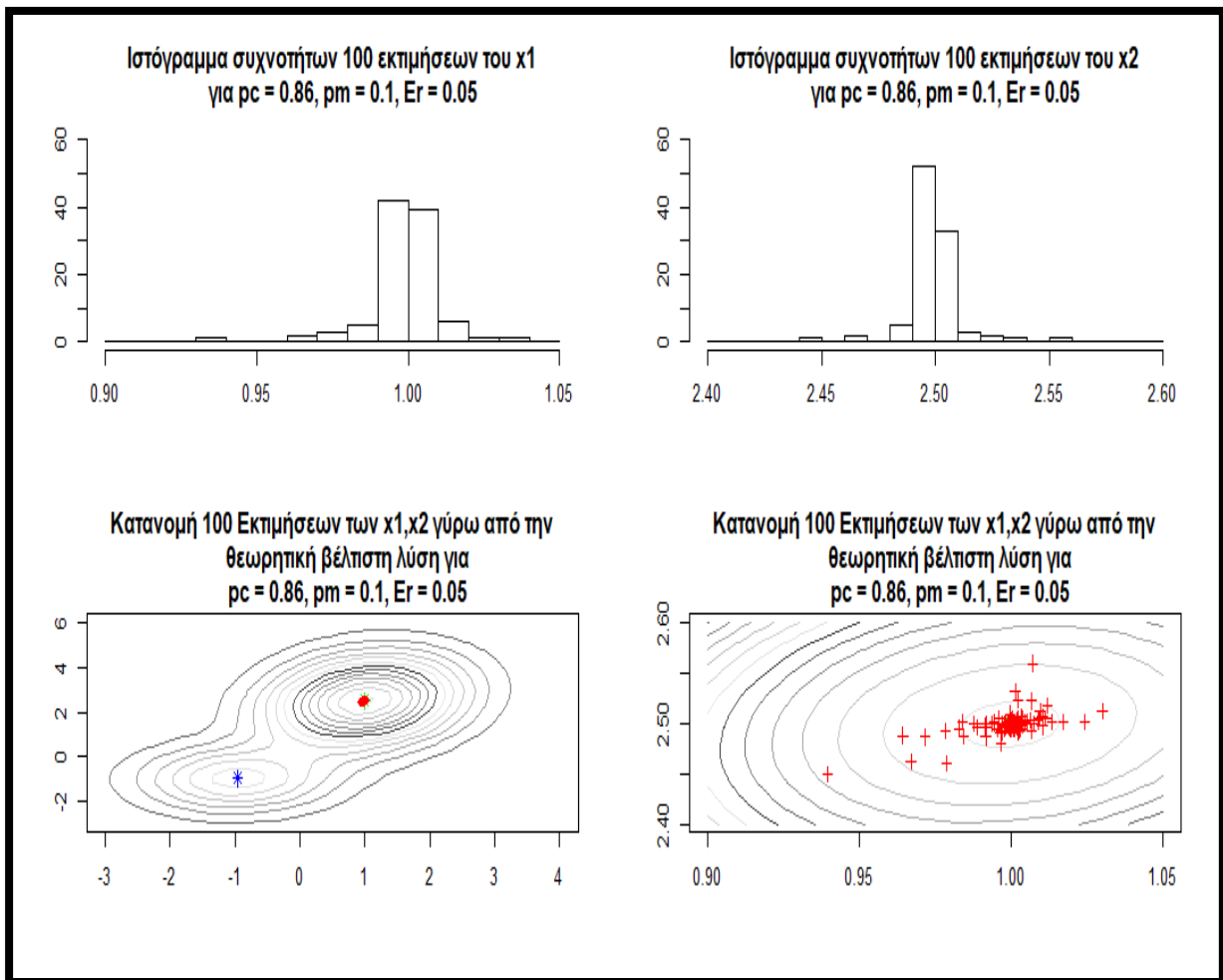
E_r	Μέση τιμή \hat{x}_1	Τυπική απόκλιση \hat{x}_1	Μέση τιμή \hat{x}_2	Τυπική απόκλιση \hat{x}_2	Μέση τιμή $\hat{f}(x_1, x_2)$	Τυπική απόκλιση $\hat{f}(x_1, x_2)$	MSE (\hat{x}_1, \hat{x}_2)
0.00	0.9955888	0.0262969	2.498047	0.0225083	0.0657286	0.00016836	0.00411169
0.05	0.9988304	0.0111314	2.498923	0.0118555	0.0657532	1.73625e-05	0.00076927
0.10	0.9976712	0.0094789	2.498099	0.0123293	0.0657534	1.90854e-05	0.00210910
0.20	0.9967962	0.0126276	2.496992	0.0105303	0.0657521	3.82122e-05	0.00342168
0.50	0.9972469	0.0157988	2.496428	0.0086293	0.0657494	5.43556e-05	0.00345228
0.80	0.9960453	0.0145480	2.495079	0.0138370	0.0657487	4.08899e-05	0.00525867
1.00	1.021405	0.453966	2.384286	0.6232184	0.0541104	0.00920155	0.11672370



Εικόνα 3.15 Διαγράμματα διασποράς για διάφορες τιμές του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r με το $\log(\text{MSE})$ και το MSE της αντίστοιχης εκτιμηθείσας λύσης, και διάγραμμα με καμπύλη παλινδρόμησης

Παρατηρούμε ότι το καλύτερο αποτέλεσμα εξάγεται για τιμή ποσοστού διατήρησης ισχυρών χρωμοσωμάτων $E_r = 5\%$. Επίσης εκτιμάται με μια καμπύλη παλινδρόμησης το πως μπορεί να σχετίζεται η ποιότητα της λύσης με την μεταβολή του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r .

Παρακάτω δίνονται τα διαγράμματα ισοϋψών καμπυλών και τα ιστογράμματα με τις 10^2 στον αριθμό εκτιμήσεις των λύσεων για $p_c = 0.86$, $p_m = 0.10$, $E_r = 0.05$, από τις οποίες προέκυψε η βέλτιστη λύση, η οποία ήταν για $\hat{x}_1 = 0.9988304$ και $\hat{x}_2 = 2.498923$ και ολικό μέγιστο $\hat{f}(x) = 0.0657532$.



Εικόνα 3.16 Ιστογράμματα και διαγράμματα ισοϋψών καμπυλών με τις εκτιμήσεις των λύσεων που προκύπτουν από 10^2 επαναλήψεις του αλγορίθμου για $p_c = 0.86$, $p_m = 0.10$, $E_r = 0.05$

Τέλος να επισημάνουμε ότι οι εκτιμήσεις 10^2 στον αριθμό, της βέλτιστης λύσης για κάθε επιλογή της πιθανότητας διασταύρωσης p_c , της πιθανότητας μετάλλαξης p_m , και του ποσοστού διατήρησης ισχυρών χρωμοσωμάτων E_r πραγματοποιήθηκαν σε σύστημα Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz (4 CPUs), ~3.0GHz με 8192MB RAM, και ο μέσος χρόνος εκτέλεσης για την κάθε μία ήταν περίπου **7 λεπτά**.

Στη συνέχεια στην τέταρτη και τελευταία ενότητα, θα εφαρμόσουμε τον γενετικό αλγόριθμο σε πραγματικό πρόβλημα όπως αυτό της βελτιστοποίησης χαρτοφυλακίου επενδύσεων, χρησιμοποιώντας τις ιστορικές αποδόσεις διάφορων χρηματιστηριακών τίτλων. Εκεί με την βοήθεια της βιβλιοθήκης GA που προσφέρει η R, θα έχουμε την δυνατότητα περαιτέρω πειραματισμού με τους μηχανισμούς του αλγορίθμου.

Βελτιστοποίηση Χαρτοφυλακίου με Γενετικούς Αλγόριθμους

Στο κεφάλαιο αυτό θα γίνει εφαρμογή τεχνικών βελτιστοποίησης που βασίζονται στους γενετικούς αλγόριθμους, η δομή και λειτουργία των οποίων αναλύθηκε στην προηγούμενη ενότητα. Σκοπός του κεφαλαίου η ανάδειξη της χρησιμότητας των γενετικών αλγορίθμων στην εύρεση ενός ολικού βέλτιστου σε αληθινά προβλήματα όπως η εύρεση της βέλτιστης σύνθεσης ενός χαρτοφυλακίου επενδύσεων με πραγματικές αποδόσεις μετοχών.

4.1 Σύντομη Αναφορά στο μοντέλο

Το μοντέλο στο οποίο θα εφαρμοστούν οι τεχνικές είναι το μοντέλο του Harry Markowitz το οποίο δεν δίνει κλειστού τύπου λύση λόγω του περιορισμού των θετικών βαρών, το οποίο πρακτικά μεταφράζεται σε απαγόρευση των ανοιχτών πωλήσεων. Έτσι στην μαθηματική διατύπωση του προβλήματος αντικείμενο προς βελτιστοποίηση, μπορεί να είναι η ελαχιστοποίηση του κινδύνου του χαρτοφυλακίου, ή σε μια πιο σύγχρονη προσέγγιση η μεγιστοποίηση του δείκτη του Sharpe. Παρακάτω δίνονται συνοπτικά σχέσεις που έχουν ήδη αναλυθεί και θα χρειαστούν σε αυτήν την ενότητα.

$$\max \quad \theta_p = \frac{E(r_p) - r_f}{\sigma_p}$$

$$\text{Περιορισμοί:} \quad \sum_{i=1}^n x_i = 1, 0 \leq x_i \leq 1$$

Το παραπάνω πρόβλημα είναι λοιπόν ένα πρόβλημα βελτιστοποίησης (μεγιστοποίησης εν προκειμένω), με σύνθετους περιορισμούς και άρα πρέπει να απλοποιηθεί για να μπορεί να λυθεί. Ουσιαστικά θα χρησιμοποιηθεί μια μέθοδος η οποία ενσωματώνει τους περιορισμούς στην αντικειμενική συνάρτηση μέσω μίας συνάρτησης ποινής (**penalty function**), κάτι ανάλογο δηλαδή με την συνάρτηση του **Lagrange** όπου ενσωματώνονταν οι περιορισμοί πολλαπλασιαζόμενοι από τους συντελεστές **Lagrange**. Στην προκειμένη περίπτωση η συνάρτηση ποινής η οποία ουσιαστικά αποτελεί μία παράμετρο ποινής (**penalty parameter**) πολλαπλασιάζεται με

ένα μέτρο παραβίασης των περιορισμών (**measure violation of the constraints**), το οποίο παίρνει μεγάλη κατά απόλυτο μη μηδενική τιμή όταν παραβιάζονται οι περιορισμοί, ενώ παίρνει την τιμή μηδέν στην περιοχή όπου δεν παραβιάζονται. Έτσι δημιουργείται μία νέα αντικειμενική συνάρτηση η οποία εξαφανίζει τους περιορισμούς στο πρόβλημά μας “τιμωρώντας” όλα τα σημεία της συνάρτησης που παραβιάζουν τους περιορισμούς. Πρακτικά αυτό που επιτυγχάνεται είναι η μετατροπή ενός προβλήματος με περιορισμούς σε ένα χωρίς. Για παράδειγμα σε ένα πρόβλημα ελαχιστοποίησης προσθέτοντας μία θετική ποινή σε εκείνα τα σημεία του χώρου στα οποία παραβιάζονται οι περιορισμοί γίνεται αντιληπτό ότι το ολικό ελάχιστο που θα βρεθεί αναγκαστικά θα τους ικανοποιεί.

4.1.1 Αντικειμενική συνάρτηση και περιορισμοί

Για την συνάρτηση ποινής υπάρχουν διάφορες τεχνικές κατασκευής της, κάποιες από τις οποίες θα χρησιμοποιηθούν για τις ανάγκες της εφαρμογής. Όταν ο περιορισμός είναι ανίσωση της μορφής $g(x) > 0$, μπορεί να κατασκευαστεί μία εξίσωση ποινής της μορφής $f(x) = \max(0, g(x))$. Με αυτό τον τρόπο σε ένα πρόβλημα π.χ. ελαχιστοποίησης εάν $g(x) < 0$, τότε $f(x) = 0$, αλλιώς $f(x) = g(x)$. Αυτό έχει σαν αποτέλεσμα να αυξηθεί η τιμή της συνάρτησης ποινής και να αποτρέπει την εξερεύνηση προς αυτό το χώρο για την εύρεση του ολικού ελάχιστου. Γενικά όσο μεγαλύτερη είναι η τιμή της συνάρτησης ποινής τόσο πιο μακριά από αυτό το χώρο βρίσκεται το ολικό ελάχιστο. Για διευκόλυνση των πράξεων και για να διατηρήσουμε την αντικειμενική μας συνάρτηση όσο το δυνατόν παραγωγίσιμη, διευκολύνει κάποιες φορές η συνάρτηση ποινής να υψώνεται στο τετράγωνο.

Έχει παρατηρηθεί ότι αυτός ο τρόπος διαχείρισης των περιορισμών, λειτουργεί καλύτερα σε προβλήματα ελαχιστοποίησης. Φυσικά αυτό δεν είναι αποτρεπτικό να χρησιμοποιηθεί και σε προβλήματα μεγιστοποίησης, αρκεί πριν την εφαρμογή, να τροποποιηθεί κατάλληλα η αντικειμενική συνάρτηση και πιο συγκεκριμένα, να αντικαταστήσουμε το Δf με $-\Delta f$ και τον έλεγχο αρνητικότητας στους περιορισμούς από ≤ 0 σε ≥ 0 .

Συγκεκριμένα οι μετασχηματισμοί που θα λάβουν χώρα στη δικιά μας περίπτωση βάσει όσων αναλύθηκαν παραπάνω μεταβαίνοντας από ένα πρόβλημα μεγιστοποίησης σε ένα ελαχιστοποίησης είναι οι εξής:

$$\bullet x_i \leq 1 \rightarrow x_i \geq 1 \rightarrow [\max(0, x_i - 1)]^2$$

$$\bullet x_i \geq 0 \rightarrow x_i \leq 0 \rightarrow [\max(0, -x_i)]^2$$

Όσον αφορά τον άλλο περιορισμό του προβλήματός, ο οποίος είναι της μορφής $f(x) - d = 0$, μετατρέπεται σε $[f(x) - d]^2$ και πιο συγκεκριμένα:

$$\bullet \sum_{i=1}^n x_i = 1 \rightarrow [\sum_{i=1}^n x_i - 1]^2$$

Μια εναλλακτική τεχνική για να διαχειριστούμε τους περιορισμούς σε ένα πρόβλημα βελτιστοποίησης είναι διαμορφώνοντας μια συνθήκη εντός της αντικειμενικής συνάρτησης. Για παράδειγμα σε ένα πρόβλημα ελαχιστοποίησης με περιορισμούς (**constrained minimization problem**) δοθείσης μια αντικειμενικής συνάρτησης $f: S \rightarrow \mathbb{R}$ όπου ισχύει ότι $S \subseteq \mathbb{R}$, θα έχουμε:

- Minimize $f(x) + \sigma P(x)$

Όπου

σ = το μέτρο παραβίασης του περιορισμού (**measure violation of the constraint**)

και $P(x)$ = η τιμή της συνάρτησης ποινής (**penalty function**), η οποία θα είναι:

- $P(x) = \begin{cases} 0, & \text{αν το } x \in S \\ +\infty, & \text{διαφορετικά} \end{cases}$

Κώδικας:

```
### Constrained Function #####
obj <- function(x){
  constr1 <-
  constr2 <-
  constr3 <-
  if(constr1==1 & constr2==1 & constr3 ==1 ){
    o = f(x)
  }else{
    o = f(x) + 200
  }
  return(o)
}
```

Οπότε για την κατασκευή της τελικής αντικειμενικής συνάρτησης, η οποία ουσιαστικά δημιουργείται με σκοπό την μετατροπή από ένα πρόβλημα με περιορισμούς σε ένα χωρίς, προκειμένου να βοηθηθεί περαιτέρω ο αλγόριθμος κατά την αναζήτηση του ολικού ελαχίστου, πολλαπλασιάζεται όπως έχει ήδη ειπωθεί με ένα μέτρο παραβίασης των περιορισμών. Στην τιμή του συντελεστή δίνονται τιμές συνήθως από 10 – 100 ανάλογα με το πόσο θέλει ο ερευνητής να ενισχύσει την διαδικασία της βελτιστοποίησης.

Τέλος τροποποιώντας κατάλληλα την συνάρτηση πριν εκτελεστεί ο αλγόριθμος και πιο συγκεκριμένα αντικαθιστώντας την $Sharpe(\{x_i\})$ με την $-Sharpe(\{x_i\})$. θα προχωρήσουμε στην εφαρμογή μας με τις πραγματικές ιστορικές αποδόσεις επιλεγμένων μετοχών.

4.2 Μελέτη περίπτωσης με ιστορικές αποδόσεις πραγματικών χρηματιστηριακών τίτλων

Αντικείμενο της εφαρμογής θα είναι η μεγιστοποίηση του Sharpe Ratio και πιο συγκεκριμένα η ελαχιστοποίηση του (-)Sharpe Ratio σε μια τροποποιημένη(**modified**) μορφή του χωρίς την προσθήκη του ακίνδυνου χρεογράφου το οποίο έτσι και αλλιώς είναι σταθερά και δεν επηρεάζει την λύση του προβλήματος. Στη συνέχεια λοιπόν δίνεται το μαθηματικό πρόβλημα ελαχιστοποίησης, όπου η αντικειμενική συνάρτηση όπως έχει προαναφερθεί είναι κατασκευασμένη έτσι ώστε να συμπεριλαμβάνει όλους τους περιορισμούς του προβλήματος

$$\max \text{ Sharpe Ratio: } \theta = \frac{E(r_p)}{\sigma_p}$$

Αντικειμενική συνάρτηση:

$$-\text{Sharpe}(\{x_i\}) + 100 \left[\left(\sum_{i=1}^n x_i - 1 \right)^2 + \sum_{i=1}^n (\max(0, x_i - 1))^2 + \sum_{i=1}^n (\max(0, -x_i))^2 \right]$$

Αρχικά θα συγκεντρώσουμε τις ημερήσιες τιμές κλεισίματος από τον ιστότοπο του [Yahoo Finance](#), στο διάστημα από 7 Οκτωβρίου 2013 έως 4 Οκτωβρίου 2019 για τις παρακάτω μετοχές μαζί με αυτές τους δείκτη S&P500:

- Microsoft
- Amazon
- Apple
- Facebook
- Netflix
- S&P 500 index

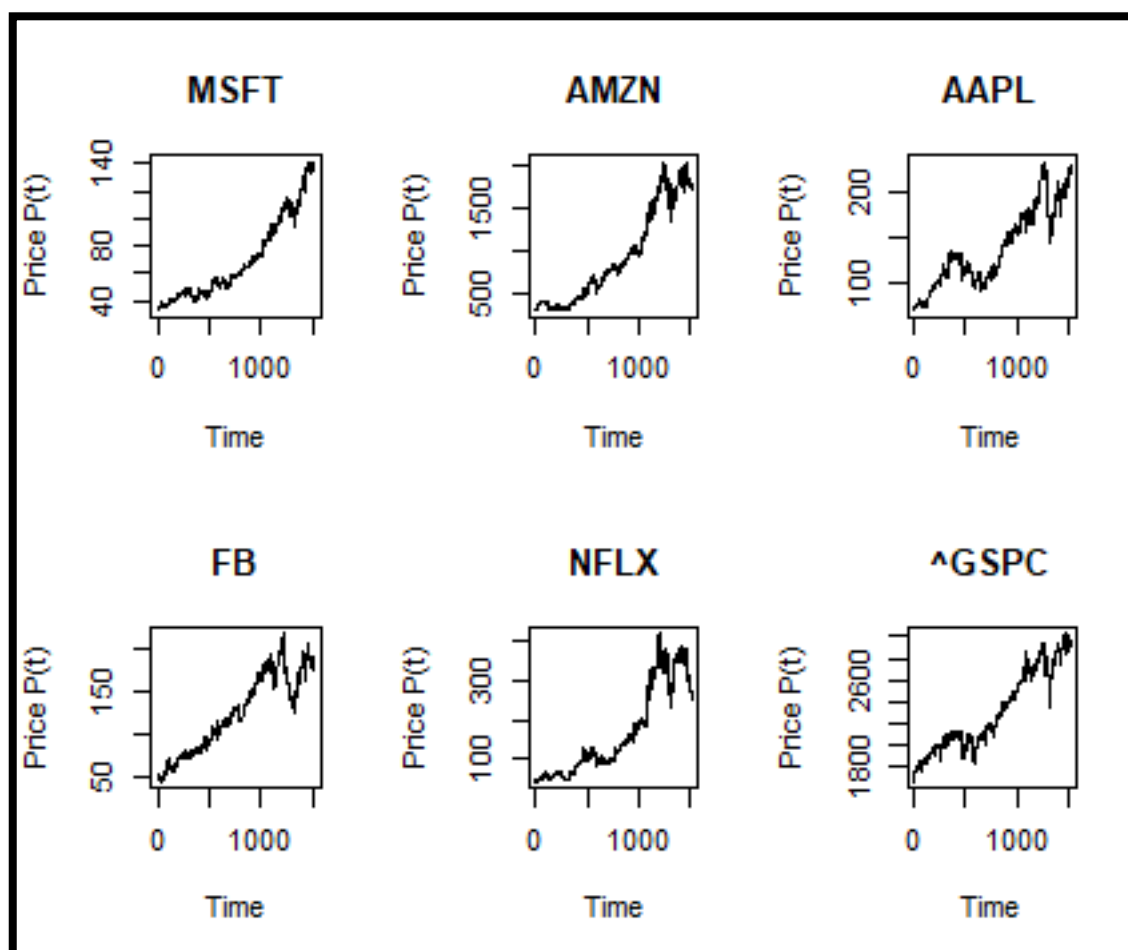
Παρακάτω δίνεται ο πίνακας με τις ιστορικές ημερήσιες τιμές των μετοχών όπως αυτές συλλέχθηκαν. (Ο Κώδικας δίνεται στο παράρτημα Π3 στο τέλος του συγγράμματος)

Πίνακας 4.1 Ιστορικές ημερήσιες τιμές μετοχών

	Date	MSFT.csv	AMZN.csv	AAPL.csv	FB.csv	NFLX.csv	^GSPC.csv
	2013-10-07	33.30	310.03	69.67857	50.52	45.45143	1676.12
	2013-10-08	33.01	303.23	68.70571	47.14	43.18857	1655.45
	2013-10-09	33.07	298.23	69.51286	46.77	41.20428	1656.40
	2013-10-10	33.76	305.17	69.94857	49.05	43.42714	1692.56
	2013-10-11	34.13	310.89	70.40143	49.11	42.97857	1703.20

	Date	MSFT.csv	AMZN.csv	AAPL.csv	FB.csv	NFLX.csv	^GSPC.csv
1506	2019-09-30	139.03	1735.91	223.97	178.08	267.62	2976.74
1507	2019-10-01	137.07	1735.65	224.59	175.81	269.58	2940.25
1508	2019-10-02	134.65	1713.23	218.96	174.60	268.03	2887.61
1509	2019-10-03	136.28	1724.42	220.82	179.38	268.15	2910.63
1510	2019-10-04	138.12	1739.65	227.01	180.45	272.79	2952.01

Στη συνέχεια δίνονται οι αντίστοιχες χρονοσειρές των ημερήσιων κινήσεων για όλο το χρονικό διάστημα. (Ο Κώδικας δίνεται στο παράρτημα Π3 στο τέλος του συγγράμματος)



Εικόνα 4.1 Χρονοσειρά ημερήσιας κίνησης μετοχών και δείκτη

Ακολουθεί ο πίνακας με τις αποδόσεις όπως αυτές υπολογίστηκαν βάση της γνωστής σχέσης:

$$R_{i,t} = \frac{V_{i,t} - V_{i,0}}{V_{i,0}}$$

όπου:

$V_{i,0}$: Αξία (ή κόστος) της i επένδυσης στην αρχή της περιόδου διακράτησης

$V_{i,t}$: Αξία της i επένδυσης στο τέλος της περιόδου διακράτησης

Κώδικας:

```
##### Υπολογισμός αποδόσεων #####
for (i in 2:ncol(f)) {
  # Χρονοσειρά τιμών μετοχών της i μετοχής
  prices <- f[,i]

  # Το lag των τιμών των αποδόσεων κατά 1
  prices_prev <- c(NA,prices[1:(length(prices)-1)])

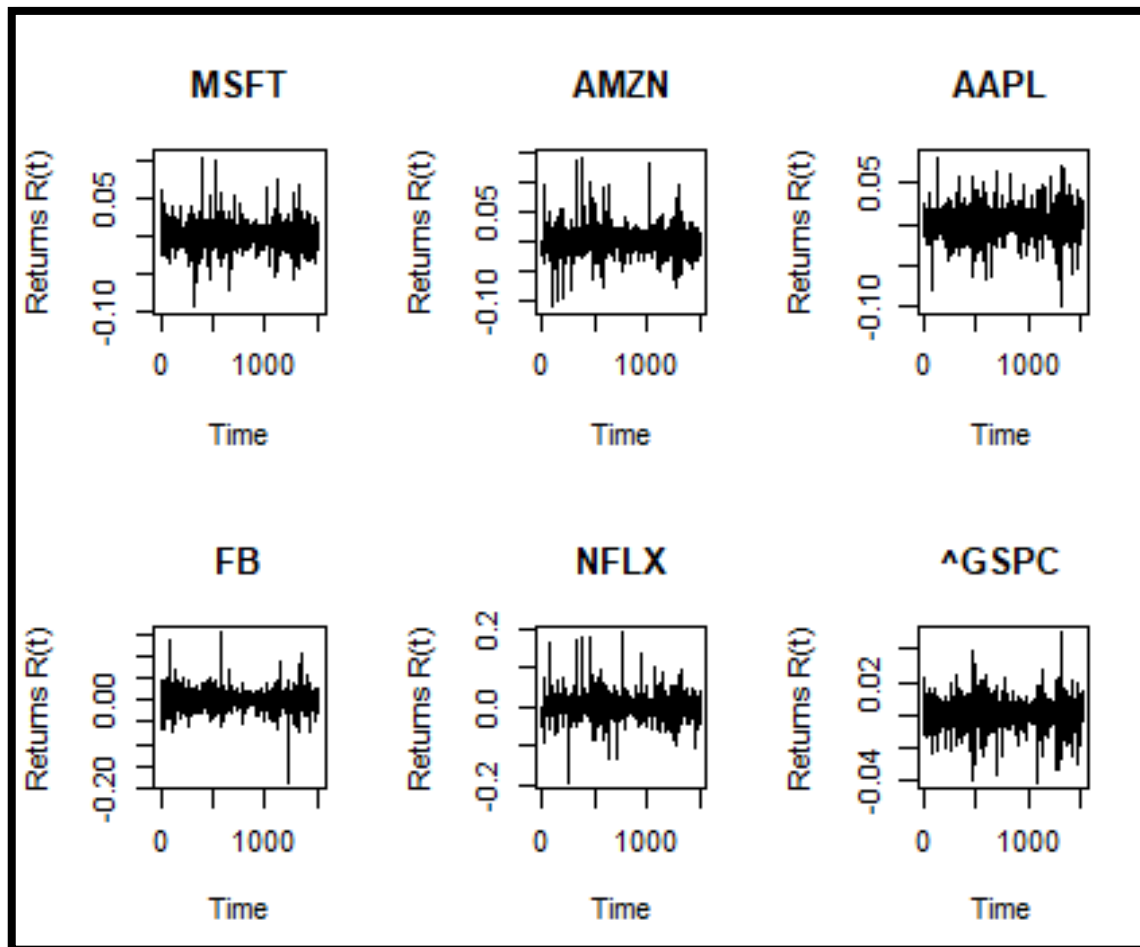
  # Χρονοσειρά αποδόσεων
  returns <- (prices-prices_prev)/prices_prev

  # Αντικατάσταση της i στήλης με τις αποδόσεις της i μετοχής
  f[,i] <- returns
}
# Αφαιρούμε την πρώτη γραμμή με το NAs και την στήλη με την ημερομηνία
asset_returns <- f[2:nrow(f),2:ncol(f)]
head(asset_returns,5)
tail(asset_returns,5)
```

Πίνακας 4.2 Ημερήσιες αποδόσεις μετοχών

	MSFT.csv	AMZN.csv	AAPL.csv	FB.csv	NFLX.csv	^GSPC.csv
2	-0.008708739	-0.0219333227	-0.013962154	-0.066904216	-0.04978623	-0.0123320789
3	0.001817692	-0.0164891331	0.011747859	-0.007848939	-0.04594472	0.0005739062
4	0.020864772	0.0232706359	0.006268107	0.048749177	0.05394725	0.0218304965
5	0.010959805	0.0187436568	0.006474157	0.001223282	-0.01032925	0.0062862715
6	0.009375915	-0.0006111583	0.006554214	0.008144919	0.07814522	0.0040747206
	MSFT.csv	AMZN.csv	AAPL.csv	FB.csv	NFLX.csv	^GSPC.csv
1506	0.009438779	0.0060622350	0.023535298	0.005533574	0.0172571394	0.005047607
1507	-0.014097619	-0.0001497831	0.002768206	-0.012747102	0.0073237876	-0.012258373
1508	-0.017655307	-0.0129173760	-0.025067853	-0.006882384	-0.0057496405	-0.017903203
1509	0.012105496	0.0065315598	0.008494702	0.027376855	0.0004476924	0.007971913
1510	0.013501585	0.0088319433	0.028031826	0.005964946	0.0173038042	0.014216898

Στη συνέχεια παρουσιάζονται οι αντίστοιχες χρονοσειρές των αποδόσεων για όλο το χρονικό διάστημα. (Ο Κώδικας δίνεται στο παράρτημα Π3 στο τέλος του συγγράμματος)



Εικόνα 4.2 Χρονοσειρά ημερησίων αποδόσεων μετοχών και δείκτη

Και εν συνεχεία ακολουθεί πίνακας με τα περιγραφικά μέτρα των αποδόσεων των μετοχών και του δείκτη (Ο Κώδικας δίνεται στο παράρτημα Π3 στο τέλος του συγγράμματος).

Πίνακας 4.3 Περιγραφικά μέτρα αποδόσεων μετοχών

	MSFT.csv	AMZN.csv	AAPL.csv	FB.csv	NFLX.csv	^GSPC.csv
Mean	0.00105	0.00132	0.00090	0.00103	0.00154	0.00041
Median	0.00073	0.00124	0.00076	0.00105	0.00045	0.00050
St.Dev.	0.01445	0.01906	0.01537	0.01914	0.02655	0.00830
Max	0.10452	0.14131	0.08198	0.15521	0.19028	0.04959
Min	-0.09253	-0.10997	-0.09961	-0.18961	-0.19370	-0.04098
Skewness	0.31732	0.50979	-0.20054	-0.07682	0.60711	-0.42878
Kurtosis	6.56598	9.17552	4.33530	12.69423	9.78661	3.56850

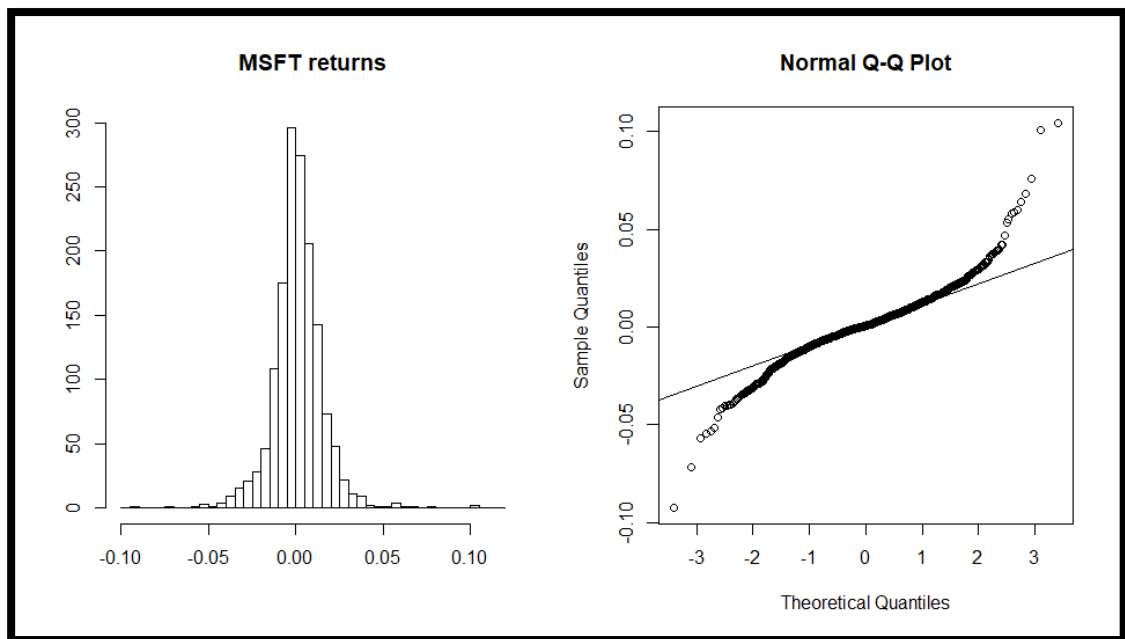
Τελευταίος δίνεται και ο αντίστοιχος πίνακας των μεταξύ τους συσχετίσεων.

Πίνακας 4.4 Πίνακας συσχετίσεων μετοχών

	MSFT.csv	AMZN.csv	AAPL.csv	FB.csv	NFLX.csv	^GSPC.csv
MSFT.csv	1.0000000	0.5518371	0.5150425	0.4539071	0.3936457	0.7273999
AMZN.csv	0.5518371	1.0000000	0.4374812	0.5365788	0.4455710	0.5880280
AAPL.csv	0.5150425	0.4374812	1.0000000	0.4110291	0.3339506	0.6297168
FB.csv	0.4539071	0.5365788	0.4110291	1.0000000	0.3941946	0.5459739
NFLX.csv	0.3936457	0.4455710	0.3339506	0.3941946	1.0000000	0.4629340
^GSPC.csv	0.7273999	0.5880280	0.6297168	0.5459739	0.4629340	1.0000000

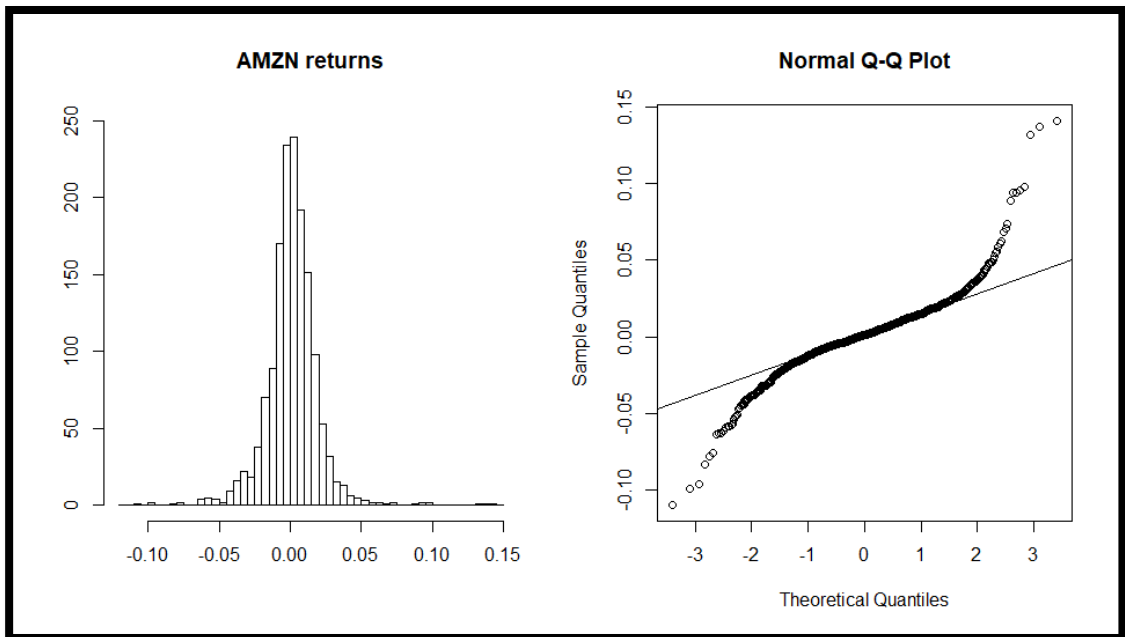
Τα τελευταία διαγράμματα αφορούν την απεικόνιση της κατανομής των αποδόσεων, με το αντίστοιχο Ιστόγραμμα συχνότητων και το QQ-plot διάγραμμα. (Ο Κώδικας δίνεται στο παράρτημα Π3 στο τέλος του συγγράμματος).

- Για την Μετοχή (MSFT)



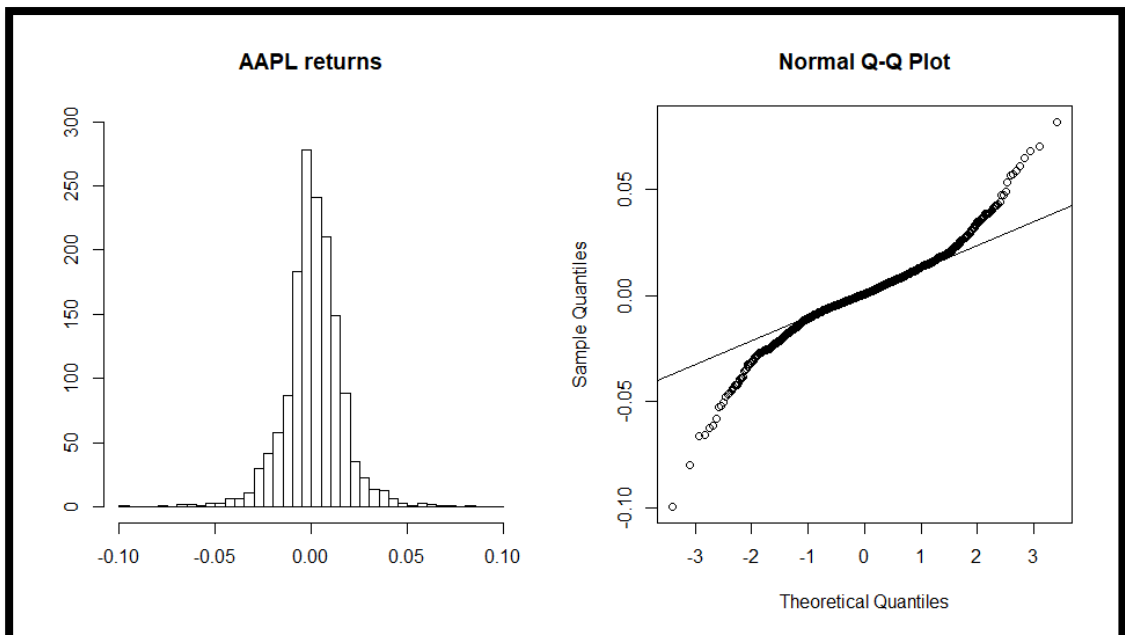
Εικόνα 4.3 MSFT Ιστόγραμμα και QQ-Plot

- Για την Μετοχή (AMZN)



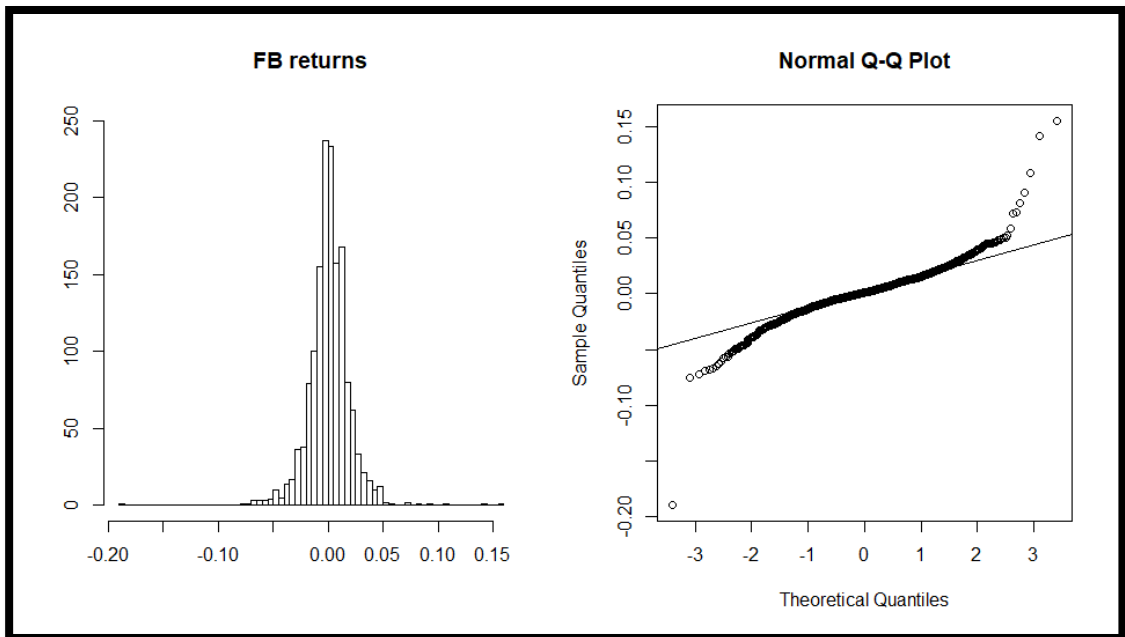
Εικόνα 4.4 AMZN Ιστόγραμμα και QQ-Plot

- Για την Μετοχή (AAPL)



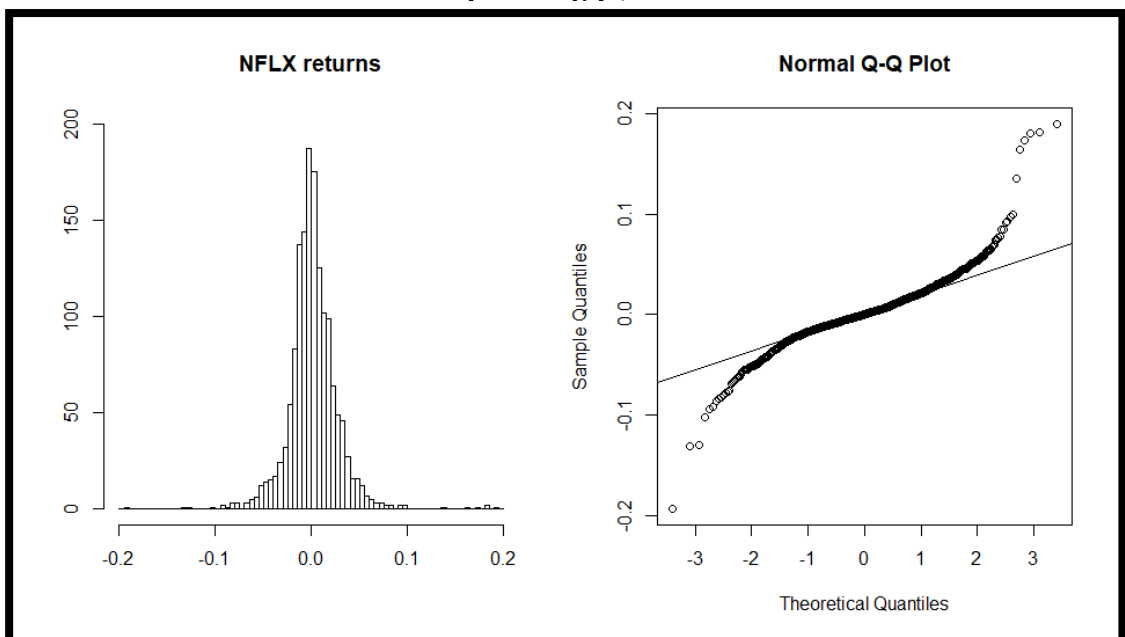
Εικόνα 4.5 AAPL Ιστόγραμμα και QQ-Plot

- Για την Μετοχή (FB)



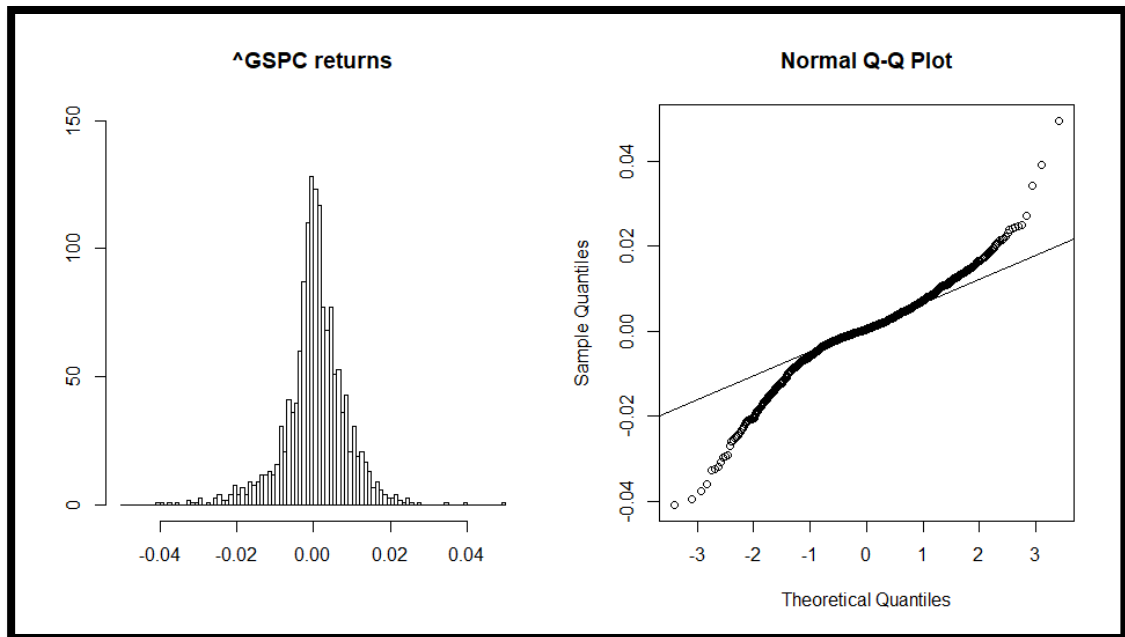
Εικόνα 4.6 FB Ιστόγραμμα και QQ-Plot

- Για την Μετοχή (NFLX)



Εικόνα 4.7 NFLX Ιστόγραμμα και QQ-Plot

- Για την Μετοχή (^GSPC)



Εικόνα 4.8 ^GSPC Ιστόγραμμα και QQ-Plot

Οι ενδείξεις από τα γραφήματα μας δίνουν μία πρώτη εκτίμηση ότι η κατανομή των αποδόσεων δεν φαίνεται να προέρχεται από την κανονική κατανομή, χωρίς αυτό να επηρεάζει την διαδικασία της βελτιστοποίησης. Θα μας δυσκόλευε μόνο στην περίπτωση που ως κριτήριο προς βελτιστοποίηση είχαμε χρησιμοποιούσαμε την αξία σε κίνδυνο VaR και θέλαμε να το εκτιμήσουμε.

Στη συνέχεια θα υπολογίσουμε την ημερήσια σταθμισμένη απόδοση του χαρτοφυλακίου, πολλαπλασιάζοντας με το αντίστοιχο βάρος x_i και κατόπιν προσθέτοντας την ημερήσια απόδοση της κάθε μετοχής. Εφαρμόζεται δηλαδή ο τύπος:

$$r_p = \sum_{i=1}^n x_i r_i$$

Όπου

r_p : Η ημερήσια σταθμισμένη απόδοση του χαρτοφυλακίου

x_i : Το βάρος της μετοχής i , στοιχείο i του διανύσματος των βαρών

r_i : Η ημερήσια απόδοση της μετοχής i

Κώδικας:

```
##### Σταθμισμένες ημερήσιες αποδόσεις χαρτοφυλακίου #####
portfolio_returns <- function(x) {
  port.returns <- 0
  #Πολλ/μός του i τίτλου με το αντίστοιχο i Βάρος(στοιχείο) του διανύσματος X.
  for (i in 1: length(x)) {
    port.returns <- port.returns + asset_returns[,i] * x[i]
  }
  return (port.returns)
}
```

Στο επόμενο βήμα θα υπολογιστεί και η μέση απόδοση του χαρτοφυλακίου $E(r_p)$ βάση του γνωστού τύπου.

$$E(r_p) = \sum_{i=1}^n x_i E(r_i)$$

Και θα κατασκευαστεί η συνάρτηση προς βελτιστοποίηση χωρίς περιορισμούς δηλαδή αυτή του Sharpe $\theta_p = \frac{E(r_p)}{\sigma_p}$, όπου $E(r_p)$ η αναμενόμενη απόδοση του χαρτοφυλακίου και σ_p η τυπική απόκλιση των αποδόσεών του.

Κώδικας:

```
##### Sharpe Function #####
sharpe <- function(x) {
  port.returns <- portfolio_returns(x)
  # Mean(port.returns) η αναμενόμενη απόδοση του χαρτοφυλακίου και
  # sd(port.returns) η τυπική απόκλιση των αποδόσεών του
  return (mean(port.returns)/sqrt(sd(port.returns)^2))
}
```

Τα δύο τελευταία βήματα πριν την εφαρμογή του γενετικού αλγορίθμου, αφορούν την κατασκευή της συνάρτησης ποινής (penalty function).

```
##### Συνάρτηση ποινής - Constraint function - penalty #####
constraint = function(x) {
  boundary_constr = (sum(x)-1)^2 # "sum x = 1" constraint
  for (i in 1:length(x)) {
    boundary_constr = boundary_constr +
      max(c(0,x[i]-1))^2 + # "x <= 1" constraint
      max(c(0,-x[i]))^2 # "x >= 0" constraint
  }
  return (boundary_constr)
}
```

Και της τελικής αντικειμενικής συνάρτησης προς βελτιστοποίηση με ενσωμάτωση των περιορισμών.

```
##### Τελική αντικειμενική συνάρτηση προς βελτιστοποίηση #####
obj <- function(x) {
  # Μεγιστοποίηση του Sharpe(x) μέσω ελαχιστοποίησης του -Sharpe(x)
  # πολλαπλασιάζοντας με -1
  return (-sharpe(x)+100*constraint(x))
}
```

Τέλος αφού έχουν ολοκληρωθεί όλα τα παραπάνω βήματα μπορούμε να προχωρήσουμε στην εφαρμογή του αλγόριθμου.

4.3 Εφαρμογή Γενετικού αλγόριθμου

Σε αυτήν την ενότητα θα εφαρμοστεί ο γενετικός αλγόριθμος με την χρήση βιβλιοθήκης της R με το όνομα “GA” (“GA” library), για το παραπάνω πρόβλημα βελτιστοποίησης που κατασκευάστηκε. Ο αλγόριθμος θα τρέχει για διαφορετικές τιμές των σημαντικότερων παραμέτρων του και μετά την εξαγωγή των αποτελεσμάτων, θα καταγράφονται όλα συγκεντρωτικά σε ένα πίνακα. Ουμίζουμε ότι το κάθε χρωμόσωμα αποτελεί μία εφικτή λύση του προβλήματος, και πιο συγκεκριμένα το διάλυμα των βαρών βάσει των οποίων κατανέμεται το κεφάλαιο.

Αρχικά δίνεται παρακάτω ο κώδικας ο οποίος θα παραμετροποιείται αναλόγως της περίπτωσης που εξετάζεται.

Κώδικας:

```
##### GENETIC ALGORITHMS #####
library("GA")
library("parallel")
library("doParallel")
library("mcga")

ga_res = ga(
  # Κωδικοποίηση χρωμοσωμάτων με πραγματικές τιμές
  type="real-valued",
  # Μέθοδος επιλογής (Selection method)
  ###selection = gareal_sigmaSelection, #(Sigma Selection method)
  ###selection = gareal_lrSelection, #(Rank Selection)
  ###selection = gareal_tourSelection, #(Tournament Selection)
  # Μέθοδος διασταύρωσης (Cross-over method)
  ###crossover = gareal_waCrossover, #(arithmetic crossover)
  ###crossover = gareal_blxCrossover, #(blending crossover)
  # Πιθανότητα διασταύρωσης (Likelihood of crossover) #
  pcrossover = 0.80:0.90,
  # Πιθανότητα μετάλλαξης (Likelihood of mutation) #
  pmutation = 0.01:0.10,
  # Ποσοστό διατήρησης ισχυρών χρωμοσωμάτων (Elitism ratio) #
  elitism = max(1, round(popSize*0.01:0.10)),
  # Μέγεθος πληθυσμού (population size) #
  popSize = 100,
  # Αντικειμενική συνάρτηση #
  function(x){-obj(x)},
  # x_i >= 0
  lower = rep(0, ncol(asset_returns)),
  # x_i <= 1
  upper = rep(1, ncol(asset_returns)),
  # Μέγιστος αριθμός επαναλήψεων αλγόριθμου #
  maxiter = 5000,
  # Μέγιστος αριθμός επαναλήψεων που δεν υπάρχει βελτίωση της υφιστάμενης λύσης
  run=100,
  # Ενισχύει την απόδοση του επεξεργαστή #
  parallel=TRUE,
  # Παρακολούθηση αποτελεσμάτων #
  monitor=TRUE,
)
```

Στην πρώτη περίπτωση θα τρέξει ο αλγόριθμος με τα εξής δεδομένα:

Επιλογή μεθόδων βασικών μηχανισμών λειτουργίας:

- **Μέθοδος επιλογής:** Tournament selection
- **Μέθοδος Διασταύρωσης:** Arithmetic crossover
- **Μέθοδος Μετάλλαξης:** Uniform mutation

Σταθερές που εισάγονται:

- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.1$
- Ποσοστό Ισχυρών χρωμοσωμάτων (**Elitism**) $Er = 0.05$
- Μέγεθος πληθυσμού (**Population size**) = 100

Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) = 5000
- Μεγιστος αριθμός επαναλήψεων χωρίς βελτίωση λύσης (**run**) = 100

Παράμετροι που μεταβάλλονται:

- Πιθανότητες διασταύρωσης (**Crossover probability**) $p_c = 0.8 - 0.9$

Βάσει λοιπόν των παραπάνω προκύπτουν οι εξής πίνακες, οι οποίοι παρουσιάζονται παρακάτω:

Πίνακας 4.5 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_c

p_c	Iterations	Βέλτιστη τιμή $\widehat{ob}_J(x)$
0.80	311	0.07173261
0.81	179	0.07762708
0.82	302	0.07531579
0.83	119	0.07808109
0.84	116	0.07804639
0.85	149	0.07533753
0.86	194	0.07655373
0.87	118	0.0777047
0.88	123	0.07554615
0.89	143	0.07821688
0.90	122	0.07690679

Η καλύτερη λοιπόν τιμή (**fitness value**) για την αντικειμενική συνάρτηση που προέκυψε είναι η 0.07821688, και αντιστοιχεί σε πιθανότητα διασταύρωσης $p_c = 0.89$. Η σύγκλιση του αλγορίθμου επιτεύχθηκε μετά από 143 επαναλήψεις (**iterations**), βάση του κριτηρίου τερματισμού που έχει τεθεί για τερματισμό του αλγορίθμου με την πάροδο 100 επαναλήψεων χωρίς βελτίωση της υφιστάμενης λύσης.

Στην δεύτερη περίπτωση κρατώντας την παραπάνω λύση ως βέλτιστη θα τρέξει ο αλγόριθμος με τα εξής δεδομένα:

Επιλογή μεθόδων βασικών μηχανισμών λειτουργίας:

- **Μέθοδος επιλογής:** Tournament selection
- **Μέθοδος Διασταύρωσης:** Arithmetic crossover
- **Μέθοδος Μετάλλαξης:** Uniform mutation

Σταθερές που εισάγονται:

- Πιθανότητες διασταύρωσης (**Crossover probability**) $p_c = 0.89$
- Ποσοστό Ισχυρών χρωμοσωμάτων (**Elitism**) $Er = 0.05$
- Μέγεθος πληθυσμού (**Population size**) = 100

Κριτήρια τερματισμού (**convergence**):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) = 5000
- Μεγιστος αριθμός επαναλήψεων χωρίς βελτίωση λύσης (**run**) = 100

Παράμετροι που μεταβάλλονται:

- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.01 - 0.10$

Βάσει λοιπόν των παραπάνω προκύπτουν οι εξής πίνακες, οι οποίοι παρουσιάζονται παρακάτω:

Πίνακας 4.6 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της p_m

p_m	Iterations	Βέλτιστη τιμή $\widehat{ob}J(x)$
0.01	118	0.07443824
0.02	173	0.07120654
0.03	116	0.08342222
0.04	122	0.07662676
0.05	137	0.08026962
0.06	124	0.07731983
0.07	117	0.07845945
0.08	126	0.07972629
0.09	125	0.07569517
0.10	143	0.07821688

Η καλύτερη λοιπόν τιμή (**fitness value**) για την αντικειμενική συνάρτηση που προέκυψε είναι η 0.08342222, και αντιστοιχεί σε πιθανότητα διασταύρωσης $p_c = 0.89$ και πιθανότητα μετάλλαξης $p_m = 0.03$. Η σύγκλιση του αλγορίθμου επιτεύχθηκε μετά από 116 επαναλήψεις (**iterations**), βάση του κριτηρίου τερματισμού που έχει τεθεί για τερματισμό του αλγορίθμου με την πάροδο 100 επαναλήψεων χωρίς βελτίωση της υφιστάμενης λύσης.

Στην τρίτη περίπτωση κρατώντας την παραπάνω λύση ως βέλτιστη θα τρέξει ο αλγόριθμος με τα εξής δεδομένα:

Επιλογή μεθόδων βασικών μηχανισμών λειτουργίας:

- **Μέθοδος επιλογής:** Tournament selection
- **Μέθοδος Διασταύρωσης:** Arithmetic crossover
- **Μέθοδος Μετάλλαξης:** Uniform mutation

Σταθερές που εισάγονται:

- Πιθανότητες διασταύρωσης (**Crossover probability**) $p_c = 0.89$
- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.03$
- Μέγεθος πληθυσμού (**Population size**) = 100

Κριτήρια τερματισμού (convergence):

- Μεγιστος αριθμός επαναλήψεων (**number of generations**) = 5000
- Μεγιστος αριθμός επαναλήψεων χωρίς βελτίωση λύσης (**run**) = 100

Παράμετροι που μεταβάλλονται:

- Ποσοστό Ισχυρών χρωμοσωμάτων (**Elitism**) $E_r = 0.01:0.10$

Βάσει λοιπόν των παραπάνω προκύπτουν οι εξής πίνακες, οι οποίοι παρουσιάζονται παρακάτω:

Πίνακας 4.7 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διάφορες τιμές της E_r

E_r	Iterations	Βέλτιστη τιμή $\widehat{ob}_J(x)$
0.01	167	0.08093023
0.02	119	0.08141523
0.03	123	0.08071044
0.04	121	0.08179579
0.05	116	0.08342222
0.06	126	0.08060473
0.07	126	0.08130888
0.08	124	0.08158778
0.09	119	0.08159665
0.10	144	0.07969571

Ως καλύτερη λύση λοιπόν και πάλι προέκυψε η 0.08342222, η οποία αντιστοιχεί σε πιθανότητα διασταύρωσης $p_c = 0.89$, πιθανότητα μετάλλαξης $p_m = 0.03$ και Elitism Ratio $E_r = 0.05$.

Στην τέταρτη περίπτωση κρατώντας την παραπάνω λύση ως βέλτιστη θα τροποποιηθεί η μέθοδος επιλογής (**selection method**), οπότε θα τρέξει ο αλγόριθμος με τα εξής δεδομένα:

Επιλογή μεθόδων βασικών μηχανισμών λειτουργίας:

- Μέθοδος διασταύρωσης: Arithmetic crossover
- Μέθοδος μετάλλαξης: Uniform mutation

Σταθερές που εισάγονται:

- Πιθανότητες διασταύρωσης (**Crossover probability**) $p_c = 0.89$
- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.03$
- Ποσοστό Ισχυρών χρωμοσωμάτων (**Elitism**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) = 100

Κριτήρια τερματισμού (convergence):

- Μειστος αριθμός επαναλήψεων (**number of generations**) = 10000
- Μειστος αριθμός επαναλήψεων χωρίς βελτίωση λύσης (**run**) = 100

Παράμετροι που μεταβάλλονται:

- Μέθοδος επιλογής (**Selection**): Tournament selection, Sigma Scaling, Rank.

Βάσει λοιπόν των παραπάνω προκύπτουν οι εξής πίνακες, οι οποίοι παρουσιάζονται παρακάτω:

Πίνακας 4.8 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διαφορετικές μεθόδους επιλογής

Selection Method	Iterations	Βέλτιστη τιμή $\widehat{ob}_J(x)$
Tournament	116	0.08342222
Sigma Scaling	7263	0.08379306
Rank	196	0.08189727

Ως καλύτερη λύση λοιπόν προέκυψε η 0.08379306, η οποία αντιστοιχεί σε πιθανότητα διασταύρωσης $p_c = 0.89$, πιθανότητα μετάλλαξης $p_m = 0.03$ και Elitism Ratio $E_r = 0.05$, ενώ σαν μέθοδος επιλογής προκρίνεται η Sigma Scaling.

Στην τελευταία περίπτωση κρατώντας την παραπάνω λύση ως βέλτιστη θα τροποποιηθεί η μέθοδος διασταύρωσης (**crossover method**), οπότε θα τρέξει ο αλγόριθμος με τα εξής δεδομένα:

Επιλογή μεθόδων βασικών μηχανισμών λειτουργίας:

- Μέθοδος επιλογής: **Sigma Scaling**
- Μέθοδος μετάλλαξης: Uniform mutation

Σταθερές που εισάγονται:

- Πιθανότητες διασταύρωσης (**Crossover probability**) $p_c = 0.89$
- Πιθανότητα μετάλλαξης (**Mutation probability**) $p_m = 0.03$
- Ποσοστό Ισχυρών χρωμοσωμάτων (**Elitism**) $E_r = 0.05$
- Μέγεθος πληθυσμού (**Population size**) = 100

Κριτήρια τερματισμού (convergence):

- Μειστος αριθμός επαναλήψεων (**number of generations**) = 10000
- Μειστος αριθμός επαναλήψεων χωρίς βελτίωση λύσης (**run**) = 100

Παράμετροι που μεταβάλλονται:

- Μέθοδος διασταύρωσης (**Crossover**): Arithmetic crossover, Blend Crossover

Βάσει λοιπόν των παραπάνω προκύπτουν οι εξής πίνακες, οι οποίοι παρουσιάζονται παρακάτω:

Πίνακας 4.9 Τα αποτελέσματα από την εκτέλεση του γενετικού αλγόριθμου, για διαφορετικές μεθόδους διασταύρωσης

Crossover-Method	Iterations	Βέλτιστη τιμή $\widehat{obj}(x)$
Arithmetic	7263	0.08379306
Blending	575	0.08515745

Συνολικά λοιπόν η καλύτερη λύση που προέκυψε είναι η 0.08515745, η οποία αντιστοιχεί σε πιθανότητα διασταύρωσης $p_c = 0.89$, πιθανότητα μετάλλαξης $p_m = 0.03$ και Elitism Ratio $E_r = 0.05$, σαν μέθοδος επιλογής προκρίνεται η Sigma Scaling και σαν μέθοδος διασταύρωσης η Blending crossover.

Το αντίστοιχο βέλτιστο διάνυσμα βαρών είναι το $\mathbf{x}^T = (0.369, 0.252, 0.172, 0.068, 0.139, 0)^T$ και το οποίο δίνει μια βέλτιστη λύση που τηρεί πλήρως όλους τους περιορισμούς του προβλήματος, δηλαδή $\sum_{i=1}^n x_i = 1$, και $0 \leq x_i \leq 1$. Ακολουθούν τα αποτελέσματα και η απεικόνιση του διαγράμματος της σύγκλισης του αλγόριθμου, στην βέλτιστη λύση ύστερα από 575 γενιές.

```

-- Genetic Algorithm -----
GA settings:
Type                = real-valued
Population size     = 100
Number of generations = 700
Elitism             = 5
Crossover probability = 0.89
Mutation probability = 0.03
Search domain =
  x1 x2 x3 x4 x5 x6
lower 0 0 0 0 0 0
upper 1 1 1 1 1 1

GA results:
Iterations          = 575
Fitness function value = 0.08515745

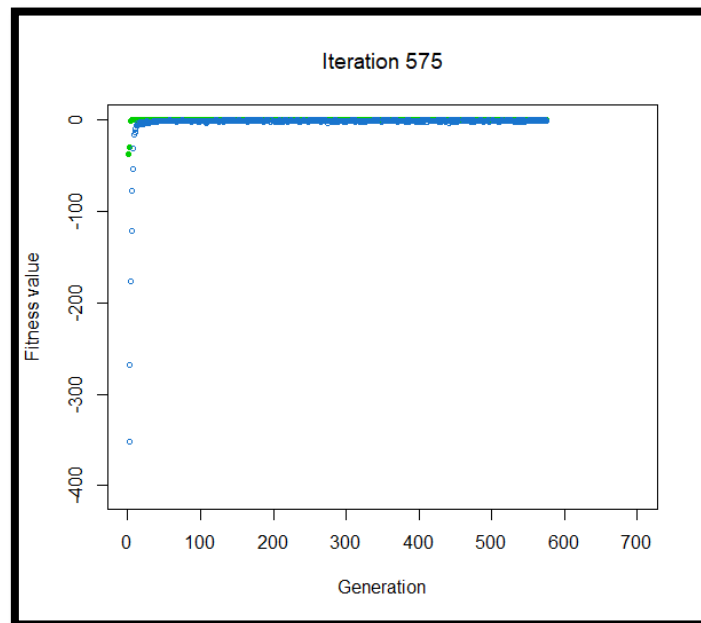
```

```

solution
[1,] "MSFT.csv" "0.369"
[2,] "AMZN.csv" "0.252"
[3,] "AAPL.csv" "0.172"
[4,] "FB.csv"   "0.068"
[5,] "NFLX.csv" "0.139"
[6,] "AGSPC.csv" "0"

```

Εικόνα 4.9 Παρουσίαση αποτελεσμάτων γενετικού αλγόριθμου



Εικόνα 4.10 Σύγκλιση Γενετικού Αλγορίθμου

4.4 Σύνθεση Βέλτιστου Χαρτοφυλακίου

Η βέλτιστη λοιπόν σύνθεση του χαρτοφυλακίου όπως αυτή προέκυψε με την εφαρμογή του γενετικού αλγόριθμου δίνεται από το διάνυσμα $\mathbf{x}^T = (0.369, 0.252, 0.172, 0.068, 0.139, 0)^T$. Βάσει αυτής της λύσης, η απόδοση και ο κίνδυνος του χαρτοφυλακίου θα είναι:

- Απόδοση Βέλτιστου Χαρτοφυλακίου: 0.001159
- Κίνδυνος Βέλτιστου Χαρτοφυλακίου: 0.013607

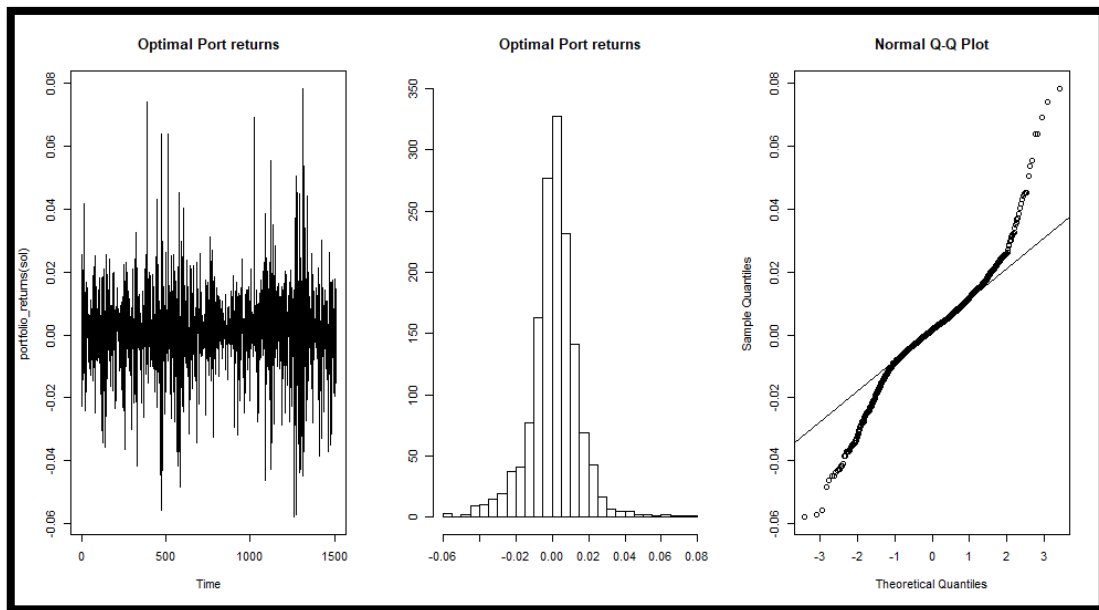
Ενώ παρακάτω δίνεται και ο συγκριτικός πίνακας απόδοσης και κινδύνου του βέλτιστου χαρτοφυλακίου και των υπόλοιπων μετοχών και του δείκτη (Ο Κώδικας δίνεται στο παράρτημα Π3 στο τέλος του συγγράμματος).

Πίνακας 4.10 Συγκριτικός πίνακας απόδοσης και κινδύνου βέλτιστου χαρτοφυλακίου και μετοχών

	Optimal	Port	MSFT.csv	AMZN.csv	AAPL.csv	FB.csv	NFLX.csv	^GSPC.csv
Mean	0.001159	0.00105	0.00132	0.00090	0.00103	0.00154	0.00041	
St.Dev.	0.013607	0.01445	0.01906	0.01537	0.01914	0.02655	0.00830	

Αυτό που φαίνεται είναι ότι η απόδοση του βέλτιστου χαρτοφυλακίου είναι λίγο χαμηλότερη από αυτή της Amazon και από αυτή του Netflix που είναι η και η υψηλότερη όλων, αλλά αυτό δεν είναι πρόβλημα γιατί ο βασικός στόχος της σύνθεσης του χαρτοφυλακίου δεν είναι η επίτευξη υψηλών αποδόσεων αλλά η σταθερότητα, και φαίνεται ότι αυτό επιτεύχθηκε αφού το χαρτοφυλάκιο φαίνεται να έχει τον πιο χαμηλό κίνδυνο από όλες τις μετοχές, χωρίς ουσιαστικά να υστερεί πολύ σε απόδοση. Μόνο ο δείκτης S&P 500, παρουσιάζει καλύτερη διαφοροποίηση ως προς τον κίνδυνο, κάτι που είναι λογικό και αναμενόμενο.

Στη συνέχεια παρουσιάζονται για το βέλτιστο χαρτοφυλάκιο, η χρονοσειρά, το Ιστόγραμμα και το QQ-plot των αποδόσεών του. (Ο Κώδικας δίνεται στο παράρτημα Π3 στο τέλος του συγγράμματος).



Εικόνα 4.11 Βέλτιστο χαρτοφυλάκιο Χρονοσειρά αποδόσεων, Ιστόγραμμα και QQ-Plot

Οι ενδείξεις από τα γραφήματα μας δείχνουν ότι η κατανομή των αποδόσεων δεν φαίνεται να προέρχεται από την κανονική κατανομή, όμως λόγω ΚΟΘ καθώς τα μεγέθη των δειγμάτων είναι μεγάλα, αλλά και ισομεγέθη, μπορούμε να προχωρήσουμε στη διενέργεια των παρακάτω στατιστικών ελέγχων που αφορούν την απόδοση του βέλτιστου χαρτοφυλακίου συγκριτικά με την μετοχή με την υψηλότερη απόδοση (Netflix), και τον κίνδυνο του χαρτοφυλακίου συγκριτικά με την μετοχή με τον χαμηλότερο κίνδυνο (Microsoft). Ακολουθούν οι στατιστικοί έλεγχοι.

Με test ισότητας μέσων τιμών (**t-test**) θα συγκρίνουμε την απόδοση του χαρτοφυλακίου και της αποδοτικότερη εκ των μετοχών, της μετοχής NFLX. Το test ελέγχει τις ακόλουθες υποθέσεις

H0: $\mu_P = \mu_5$

H1: $\mu_P < \mu_5$

Κώδικας:

```
##### t.test #####
```

```
t.test(portfolio_returns(solution), asset_returns[,5],mu=0,var.equal=FALSE,alternative="less")
```

Και τα αποτελέσματα δίνονται παρακάτω:

```
welch Two sample t-test
```

```
data: portfolio_returns(solution) and asset_returns[, 5]
t = -0.49343, df = 2249, p-value = 0.3109
```


Από τα αποτελέσματα του test δεν προκύπτουν στατιστικές ενδείξεις για την απόρριψη της μηδενικής υπόθεσης, δηλαδή σε επίπεδο στατιστικής σημαντικότητας π.χ. 5% δεν μπορεί να απορριφθεί στατιστικά η μηδενική υπόθεση ότι δηλαδή η απόδοση του βέλτιστου χαρτοφυλακίου είναι ίση με την απόδοση της μετοχής Netflix. Πρακτικά αυτό σημαίνει ότι δεν στηρίζεται στατιστικά ότι το βέλτιστο χαρτοφυλάκιο υστερεί σε απόδοση.

Όσον αφορά τον κίνδυνο, η μετοχή με τον χαμηλότερο κίνδυνο είναι η μετοχή της Microsoft, κάτι που θα εξετάσουμε και στατιστικά με ένα test ισότητας διακυμάνσεων (var-test), και με ένα Levene test. Τα test ελέγχουν τις ακόλουθες υποθέσεις

H0: $\sigma_P = \sigma_1$

H1: $\sigma_P < \sigma_1$

Κώδικας:

```
# var test #
var.test(portfolio_returns(sol),asset_returns[,6],alternative="less")

# Levene test #
library(lawstat)
V<-c(portfolio_returns(solution),asset_returns[,1])
G<-c(rep(1,times=length(portfolio_returns(solution))),rep(2,times=length(asset_re-
turns[,1])))
levene.test(V,G,location="median")
```

Τα αποτελέσματα του πρώτου ελέγχου δίνονται παρακάτω:

```
F test to compare two variances
data: portfolio_returns(solution) and asset_returns [, 1]
F = 0.88637, num df = 1508, denom df = 1508, p-value = 0.009618
```

Από τα αποτελέσματα του πρώτου ελέγχου έχουμε ότι σε ε.σ.σ π.χ.5% απορρίπτουμε την μηδενική υπόθεση, δηλαδή ο κίνδυνος του βέλτιστου χαρτοφυλακίου στατιστικά φαίνεται ότι είναι μικρότερος από αυτόν της μετοχής της Microsoft.

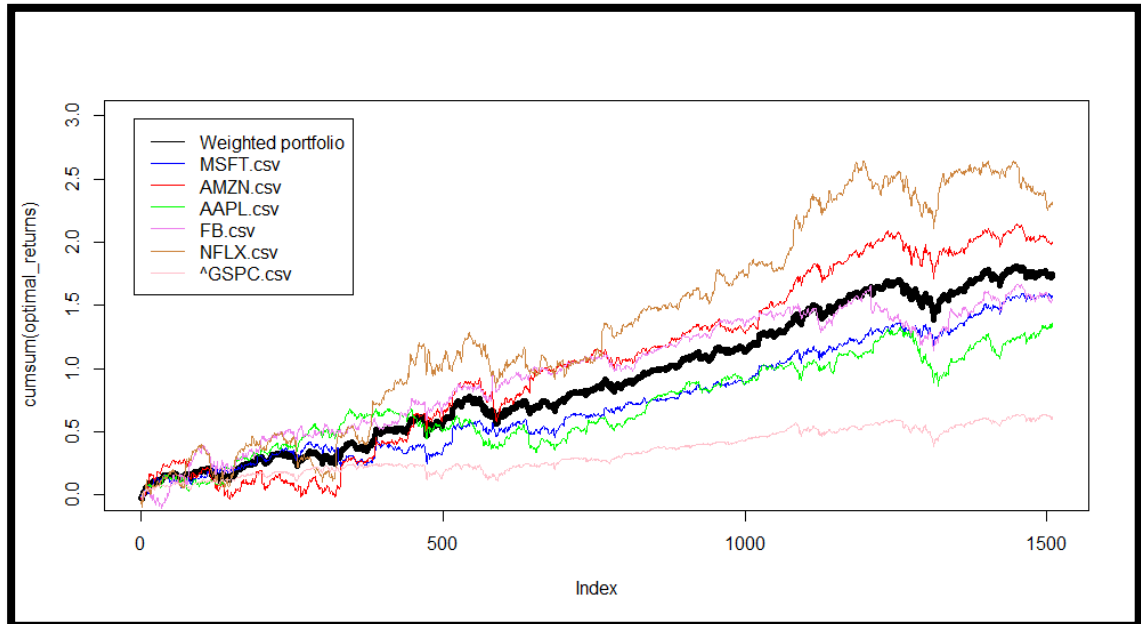
Και τα αποτελέσματα του δεύτερου ελέγχου:

```
modified robust Brown-Forsythe Levene-type test based on the absolute devi-
ations from the
      median
data: V
Test Statistic = 1.8769, p-value = 0.1708
```

Από τα αποτελέσματα του δεύτερου ελέγχου έχουμε ότι σε ε.σ.σ π.χ. 5% δεν απορρίπτουμε την μηδενική υπόθεση, δηλαδή ο κίνδυνος του βέλτιστου χαρτοφυλακίου στατιστικά φαίνεται ότι είναι ίσος με αυτόν της μετοχής της Microsoft

Πρακτικά και από τους δύο ελέγχους στηρίζεται στατιστικά το γεγονός ότι το βέλτιστο χαρτοφυλάκιο έχει χαμηλότερο ή ίσο κίνδυνο με την μετοχή με τον χαμηλότερο κίνδυνο. Έτσι από τα αποτελέσματα και της απόδοσης και του κινδύνου και των στατιστικών ελέγχων που κάναμε φαίνονται ξεκάθαρα τα οφέλη της διαφοροποίησης.

Η ισορροπία μεταξύ απόδοσης και κινδύνου για το βέλτιστο χαρτοφυλάκιο διακρίνονται και στην παρακάτω γραφική παράσταση όπου παρουσιάζονται μαζί οι ιστορικές αποδόσεις όλων των μετοχών, του δείκτη S&P 500 και αυτή του βέλτιστου χαρτοφυλακίου. (Ο Κώδικας δίνεται στο παράρτημα Π3 στο τέλος του συγγράμματος).



Εικόνα 4.12 Ιστορικές αποδόσεις βέλτιστου χαρτοφυλακίου και μεμονωμένων μετοχών

Σε αυτή λοιπόν την τελευταία ενότητα εφαρμόστηκε σταδιακά ο γενετικός αλγόριθμος αλλάζοντας κάθε φορά κάποια από τις παραμέτρους του και διατηρώντας σταθερές τις υπόλοιπες.

Έτσι καταλήξαμε στην εύρεση της βέλτιστης σύνθεσης του χαρτοφυλακίου που μεγιστοποιεί το Sharpe ratio. Τέλος αφού υπολογίστηκε η απόδοση και ο κίνδυνος του βέλτιστου χαρτοφυλακίου, επιβεβαιώθηκε και στατιστικά ότι δεν υστερεί σε απόδοση ενώ χάρη στη διαφοροποίηση παρουσιάζει τον μικρότερο κίνδυνο από όλες τις μετοχές άρα και την μεγαλύτερη σταθερότητα. Με λίγα λόγια επιτεύχθηκαν μόνο με πέντε μετοχές όλα τα θετικά αποτελέσματα της σύνθεσης χαρτοφυλακίων επενδύσεων, σε σχέση με τη διακράτηση μεμονωμένων αξιογράφων.

ΠΑΡΑΡΤΗΜΑΤΑ

Π1 Κώδικας:

- Κύριο σώμα κώδικα Γενετικού Αλγόριθμου (GA) μονοδιάστατης μεταβλητής.
- Παραμετροποίηση για διαγράμματα για 5×10^2 εκτελέσεις του αλγορίθμου.

Π2 Κώδικας :

- Κύριο σώμα κώδικα Γενετικού Αλγόριθμου (GA) δισδιάστατης μεταβλητής.
- Παραμετροποίηση και διαγράμματα για 10^2 εκτελέσεις του αλγορίθμου.

Π3 Κώδικας :

- Κώδικάς προπαρασκευής δεδομένων (data preparation) και Διαγράμματα (plots) τέταρτης ενότητας.

ΠΑΡΑΡΤΗΜΑ Π1

- Κύριο σώμα κώδικα Γενετικού Αλγόριθμου (GA) μονοδιάστατης μεταβλητής:

```
##### Genetic Algorithm Full Function #####
CGenetic_Algorithm <- function (M, l, MaxGen, pc, pm, Er, lb, ub){

##### Convergence Curve Vectors #####
  cgcurve <- rep(0,(MaxGen+1))
  mcurve <- rep(0,(MaxGen+1))

##### Initial Population #####
  population <- function(M,l,lb,ub) {
    a <- rep(0,M)
    b <- rep(0,M)
    for(i in 1:M){
      a[i] <- ((ub - lb) * runif(1,0,1) + lb)
    }
    population <-data.frame(Gene = a, fitness_values = b)
    return(population)
  }
  population <- population(M,l,lb,ub)

##### Objective function to be optimized #####
  obj <- function(x) {
    y <- rep(0,M)
    for(i in 1:M) {
      y[i] <- ((x[i,1])^2 + (x[i,1])) * cos(x[i,1])
    }
    return (y)
  }

##### Convergence Curves Initial Population Value #####
  cgcurve[1] <- max(obj(population))
  mcurve[1] <- mean(obj(population))

##### Initial gene population search position #####

  curve((x^2+x)*cos(x),lb,ub, ylab="f(x)",panel.first = grid())
  points(population$Gene,obj(population),col="red", pch = 18)
  title(expression(f(x)==(x^2+x)*cos(x)))

##### MAIN LOOP FOR ALL GENERATIONS #####

##### Fitness Function & Fitness Evaluation #####
  for (g in 1:MaxGen) {
    fit <- function(x) {
      x$fitness_values <- obj(x)
      return(x)
    }
  }
```

```

##### Roulette wheel Selection #####
RW_selection <- function(x){
  x <- fit(population)
  if(any(x$fitness_values < 0)) {
    a <- 1
    b <- min(x$fitness_values)
    b <- abs(b)
    scaled_fitness_values <- a*x$fitness_values + b + 1
    x$normalize_fitness_values <- scaled_fitness_values / sum(scaled_fit-
ness_values)
  }else{
    x$normalize_fitness_values <- x$fitness_values / sum(x$fitness_values)
  }
  x$normalize_fitness_values <- round(x$normalize_fitness_values,4)
  x <- x[order(-x$normalize_fitness_values),]
  row.names(x) <- 1:M
  csum <- rep(0,M)
  for (i in 1:M){
    csum[i] <- sum(x$normalize_fitness_values[i:M])
  }
  csum <- round(csum,3)
  x$csum <- csum

  i <- 1
  Parent1 <- x[M,]
  repeat {
    i = i+1
    if (runif(1,0,1) > x$csum[i]){
      Parent1 <- x[i-1,]
      break
    }
  }

  j <- 1
  Parent2 <- x[M,]
  repeat {
    j = j+1
    if ((runif(1,0,1) > x$csum[j]) & ((j-1) != (i-1))){
      Parent2 <- x[j-1,]
      break
    }
  }

  Parents <- rbind(Parent1, Parent2)
  return(Parents)
}

##### Arithmetic Crossover #####
Crossover <- function(pc) {
  Gene_no <- 1
  Parents <- RW_selection(fit(population))
  Parent1 <- Parents[1,1:1]
  Parent2 <- Parents[2,1:1]
  child1 <- rep(0,Gene_no)
  child2 <- rep(0,Gene_no)
  alpha <- runif(1,0,1)
  child1 <- (alpha* Parent1[1]) + ((1-alpha) * Parent2[1])
}

```

```

child2 <- ((1-alpha) * Parent1[1]) + (alpha * Parent2[1])

if (runif(1,0,1) <= pc) {
  child1 <- child1
}else{
  child1 <- Parent1
}

if (runif(1,0,1) <= pc) {
  child2 <- child2
}else{
  child2 <- Parent2
}
children <- rbind(child1, child2)
children
return(children)
}
##### Uniform Mutation #####
Mutation <- function(l,pc,pm,lb,ub) {
  children <- Crossover(pc)
  child1 <- children [1,]
  child2 <- children [2,]
  if(runif(1,0,1) < pm) {
    child1 <- ((ub[1]- lb[1]) * runif(1,0,1) + lb[1])
  }

  if(runif(1,0,1) < pm) {
    child2 <- ((ub[1]- lb[1]) * runif(1,0,1) + lb[1])
  }

  mutate_child1<- child1
  mutate_child2<- child2
  mutate_children <- rbind(mutate_child1, mutate_child2)
  return (mutate_children)
}

##### New Population after Selection, Crossover, Mutation #####
# Ο νέος Πληθυσμός πρέπει να είναι σε μέγεθος ίδιος με τον αρχικό
new_pop <- function(pc,pm) {
  df <- rep(0,M)

  # Για αυτό επαναλαμβάνουμε M/2 φορές
  for(i in seq(from=1, to=M,by=2)) {
    df[i:(i+1)] <- Mutation(l,pc,pm,lb,ub)
  }
  mut_pop <-data.frame(Gene=df)
  return(mut_pop)
}

##### New Population & Elitism #####
new_population <- function(pc, pm, Er) {

# Το Er είναι το Elitism Ratio το οποίο πολλαπλασιάζεται με τον πληθυσμό
# για να δούμε πόσα Elite Chromosomes θα διατηρήσουμε στην νέα γενιά
Elite_no <- round(Er*M)
x <- fit(population)

# Ταξινόμηση του προηγούμενου πληθυσμού βάση των Fitness Values με φθίνουσα σειρά
elites <- data.frame(matrix(ncol = 2, nrow = Elite_no))

```

```

    x <- x[order(-x$fitness_values),]
# και αποθήκευση των Elite_No = (Er * M) πρώτων και ισχυροτέρων χρωμοσωμάτων
  elites[1:Elite_no,] <- x[1:Elite_no,]

# Ταξινόμηση του νέου πληθυσμού βασή των Fitness Values με αύξουσα σειρά
  new_pop <- new_pop(pc,pm)
  new_pop <- fit(new_pop)
  new_pop <- new_pop[order(new_pop$fitness_values),]
# και αντικατάσταση των Elite_No = (Er * M) πρώτων και ασθενέστερων χρωμοσωμάτων
  με τα Elite που έχουμε αποθηκεύσει
  new_pop[1:Elite_no,] <- elites[1:Elite_no,]
  return(new_pop)
}
population <- new_population(pc, pm, Er)

# Καλύτερο Χρωμόσωμα βάσει Fitness Value
  chromo <- population[(population$fitness_values) == (max(population$fitness_values)),]
# Καλύτερη Fitness Value (ολικό Βέλτιστο)
  best <- chromo[1,2]
# Ολικός Βελτιστοποιητής
  sol <- chromo[1,1]

##### Statistics #####
  mgene <- mean(population$Gene)
  sdgene <- sd(population$Gene)
  m <- mean(population$fitness_values)
  cat("GA| ", "iter = (",g,")|", "Mean = (",m,")|", "Best = (",best,")|", "\n")

##### Convergence curves #####
  mcurve[g+1] <- mean(population$fitness_values)
  cgcurve[g+1] <- max(population$fitness_values)
  X <- 1:(MaxGen + 1)
  Y <- cgcurve
  Z <- mcurve
  plot(X,Y,type = "b", col = "green",xlim=c(0,100), ylim=c(-10,50),pch=18,
        ylab="Fitness Value",xlab="Generation",panel.first = grid())
  lines(X,Z,type = "b", col = "blue",xlim=c(0,100), ylim=c(-10,50),pch=18)

##### Gene Search Position Curves #####
  curve((x^2+x)*cos(x),lb,ub, ylab="f(x)",panel.first = grid())
  points(population$Gene,obj(population),col="red", pch = 18)
  title(expression(f(x)==(x^2+x)*cos(x)))
}

##### end of LOOP #####
  legend(locator(1), legend=c("Best","Mean"), col =
c("green","blue"),pch=18,cex=1 )
  cat("GA RESULTS", "\n", "Fitness Function Value = ",
      best, "", "\n", "Solution", "\n", "X1 = ", sol, "", "\n", "Mean(X1) = ", mgene, "
|", "Sd(X1) = ", sdgene)
}
##### end of full function #####

```


- Παραμετροποίηση για $5 \cdot 10^2$ εκτελέσεις του αλγορίθμου.

```
##### CONTROLLING PARAMETERS OF GA ALGORITHM #####
Problem.nVar <- 1          # Number of problem Variables
M <- 30                   # Number of chromosomes
l <- Problem.nVar         # Number of genes
MaxGen <- 30              # Generations (iterations)
pc <- 0.86                 # Probability of Crossover
pm <- 0.1                  # Probability of Mutation
Er <- 0.05                 # Elitism Ratio
ub <- c(10)                # UPPER BOUND FOR SEARCH SPACE
lb <- c(-10)               # LOWER BOUND FOR SEARCH SPACE

##### 5*10^2 ITERATIONS OF GA ALGORITHM #####
z <- replicate(500, CGenetic_Algorithm (M, l, MaxGen, pc, pm, Er, lb, ub))
class(z)
mean(z[1,])
sd(z[1,])
mean(z[2,])
sd(z[2,])

## MSE Calculation ###
mse <- (sd(z[1,]))^2 + (6.560539-mean(z[1,]))^2 ;mse

##### 1 #####
p_cross <- c(0,0.1,0.2,0.4,0.6,0.8,0.81,0.83,0.85,0.86,0.87,0.88,0.95,1);

MSE <- c(0.01511199,0.00670561,0.00617799,0.00277095,
         0.00245015,0.00198352,0.00194284,0.00155701,
         0.00138217,0.00093917,0.00111781,0.00131638,
         0.00134861,0.00154485)

par(mfrow=c(2,2))
plot(p_cross, log(MSE) ,pch = 18,main=" Διάγραμμα διασποράς του log(MSE) των x συν
αρτήσει του pc " )
points(0.86,log(0.00093917),col="red",pch = 18)

### Εύρεση καμπύλης ###
# Διάγραμμα διασποράς #
plot(p_cross, MSE ,pch = 18,main=" Διάγραμμα διασποράς του MSE των x
συναρτήσει του pc και καμπύλη παλινδρόμησης" )
points(0.86,0.00093917,col="red",pch = 18)
x <- seq(0,1,length=14)
# Μετασχηματισμός δεδομένων #
x2 <- 1/(x+0.1)
Y <- MSE
# Εκτίμηση Παραμέτρων ευθείας ελαχίστων τετραγώνων ##
fit1.model <- lm(Y~x2)
summary(fit1.model)
b0 <- fit1.model$coefficients[1];b0
b1 <- fit1.model$coefficients[2];b1
### Καμπύλη εκτίμησης του MSE ως προς την p ###

lines(x,b0+b1*x2,col="blue")
# Διάγραμμα διασποράς και ευθεία παλινδρόμηση για τα μετασχηματισμένα δεδομένα #
X <- seq(0,1,length=14)
```

```

X <- 1/(X+0.1)
plot(X,Y,pch = 18,main="Διάγραμμα διασποράς του MSE των μετασχηματισμένων x
συναρτήσεως του pc και ευθεία παλινδρόμησης")
fit2.model <- lm(Y~X)
abline(fit2.model)
summary(fit2.model)

##### 2 #####
p_mute <- c(seq(0,0.11,0.01))
MSE <- c(6.929856,5.566194,3.829863,2.437494,
0.702702,0.353905,0.369573,0.004638,
0.002682,0.001713,0.00093917,0.00148825)

par(mfrow=c(2,2))
plot(p_mute, log(MSE) ,pch = 18,main=" Διάγραμμα διασποράς του log(MSE) των x συνα
ρτήσεως του pm ")
points(0.10,log(0.00093917),col="red",pch = 18)

### Εύρεση καμπύλης ###
# Διάγραμμα διασποράς #
plot(p_mute, MSE ,pch = 18,main=" Διάγραμμα διασποράς του MSE των x
συναρτήσεως του pm και καμπύλη παλινδρόμησης" )
points(0.86,0.00093917,col="red",pch = 18)
x <- seq(0,0.11,length=12);x
# Μετασχηματισμός δεδομένων #
x2 <- 1/sqrt(x+0.01)
Y <- MSE
# Εκτίμηση Παραμέτρων ευθείας ελαχίστων τετραγώνων ##
fit1.model <- lm(Y~x2)
summary(fit1.model)
b0 <- fit1.model$coefficients[1];b0
b1 <- fit1.model$coefficients[2];b1

### Καμπύλη εκτίμησης του MSE ως προς την p ###
lines(x,b0+b1*x2,col="blue")
# Διάγραμμα διασποράς και ευθεία παλινδρόμηση για τα μετασχηματισμένα δεδομένα #
X <- seq(0,0.11,length=12);X
X <- 1/sqrt(X+0.01); X
plot(X,Y,pch = 18,main="Διάγραμμα διασποράς του MSE των μετασχηματισμένων x
συναρτήσεως του pm και ευθεία παλινδρόμησης")
fit2.model <- lm(Y~X)
abline(fit2.model)
summary(fit2.model)
##### 3 #####
e_ratio <- c(0,0.05,0.10,0.20,0.50,0.80,1.00)

MSE <- c(0.00217587,0.00093917,0.00094858,
0.00116908,0.00119203,0.00172603,13.9607300)

par(mfrow=c(1,2))
plot(e_ratio, log(MSE) ,pch = 18,main=" Διάγραμμα διασποράς του log(MSE) των x
συναρτήσεως του Er" )
points(0.05,log(0.00093917),col="red",pch = 18)

### Εύρεση καμπύλης ###
# Διάγραμμα διασποράς #
plot(e_ratio, MSE ,pch = 18, main="Διάγραμμα διασποράς του MSE των x
συναρτήσεως του Er και καμπύλη παλινδρόμησης")

```

```

points(0.05,0.00093917,col="red",pch = 18)
x <- seq(0,1,length=7);x
y <- MSE
#fourth degree
fit4 <- lm(y~poly(x,4,row=TRUE))
lines(x, predict(fit4, data.frame(x=x)), col="purple")

##### 4 #####
min(z[1,])
max(z[1,])
par(mfrow=c(1,3))
hist(z[1,],seq(6.2,6.8,0.02),ylim=c(0,500),
      xlim=c(6.2,6.8),xlab="",ylab="",main=" Ιστόγραμμα συχνοτήτων 500 εκτιμήσεων τ
ου x
      για pc = 0.86, pm = 0.1, Er = 0.05")
curve((x^2+x)*cos(x),lb,ub, ylab= expression(f(x)==(x^2+x)*cos(x)),panel.first = g
rid(),
      main=" Κατανομή 500 Εκτιμήσεων του x γύρω από την
      θεωρητική βέλτιστη λύση για
      pc = 0.86, pm = 0.1, Er = 0.05 ")
points(z[1,],z[2,],col="red", pch = 18)
curve((x^2+x)*cos(x),6.2,6.8, ylab= expression(f(x)==(x^2+x)*cos(x)),panel.first =
grid(),
      main=" Κατανομή 500 Εκτιμήσεων του x γύρω από την
      θεωρητική βέλτιστη λύση για
      pc = 0.86, pm = 0.1, Er = 0.05 ")
points(z[1,],z[2,],col="red", pch = 18)
points(6.560539,47.70562 ,col="blue", pch = 17)

```

ΠΑΡΑΡΤΗΜΑ Π2

- **Κύριο σώμα κώδικα Γενετικού Αλγόριθμου (GA) δισδιάστατης μεταβλητής:**

```
##### Genetic Algorithm Full Function #####
CGenetic_Algorithm <- function (M, l, MaxGen, pc, pm, Er, lb, ub){

##### Initial Population #####
population <- function(M,l,lb,ub) {
  population <- data.frame(matrix(ncol = l, nrow = M))
  for (i in 1:M) {
    for (j in 1:l) {
      population[i,j]<- ((ub[j]- lb[j]) * runif(1,0,1) + lb[j])
    }
  }
  return(population)
}
population <- (population(M,l,lb,ub))

##### Objective function to be optimized #####
obj <- function(x) {
  y <- rep(0,M)
  for(i in 1:M) {
    y[i] <- 0.2*dnorm((x[i,1]),-1,1)*dnorm((x[i,2]),-1,1)+0.8*
      (exp((-8/15)*(((x[i,1])-1)^2+((x[i,2])-2.5)^2)
        /2-(((x[i,1])-1)*((x[i,2])-2.5))/4)))/(pi*sqrt(15))
  }
  return (y)
}

##### Plot 3d Function #####
bouts <- function(x1,x2){
  return(0.2*dnorm(x1,-1,1)*dnorm(x2,-1,1)+0.8*
    (exp((-8/15)*((x1-1)^2+((x2-2.5)^2)
      /2-((x1-1)*(x2-2.5))/4)))/(pi*sqrt(15)))
}
x1 <- x2 <- seq(-4, 4, length = 30)
f <- outer(x1, x2, bouts)
persp(x1, x2, f, theta=60, phi=0, ticktype = "detailed", expand = 0.7, shade =
0.7)

##### Monitor Search History #####
monitor <- function(x) {
  contour(x1, x2, f,drawlabels = FALSE, col=gray.colors(7))
  Sys.sleep(0.2)
}
monitor(bouts)
points(-0.955467,-0.956622,col="blue", pch = 8)
points(0.999599,2.498955,col="green", pch = 8)
points(population$X1,population$X2,col="red", pch = 18)

##### MAIN LOOP #####
##### Fitness Function & Fitness Evaluation #####
```

```

for (g in 1:MaxGen) {
fit <- function(x) {
x$fitness_values <- obj(x)
return(x)
}

##### Roulette wheel Selection #####
RW_selection <- function(x){
x <- fit(population)
if(any(x$fitness_values < 0)) {
a <- 1
b <- min(x$fitness_values)
b <- abs(b)
scaled_fitness_values <- a*x$fitness_values + b + 1
x$normalize_fitness_values <- scaled_fitness_val-
ues)
}else{
x$normalize_fitness_values <- x$fitness_values / sum(x$fitness_values)
}
x$normalize_fitness_values<- round(x$normalize_fitness_values,4)
x <- x[order(-x$normalize_fitness_values),]
row.names(x) <- 1:M
csum <- rep(0,length(x$fitness_values))
for (i in 1:length(x$fitness_values)){
csum[i] <- sum(x$normalize_fitness_values[i:M])
}
csum <- round(csum,3)
x$csum <- csum

i <- 1
Parent1 <- x[M,]
repeat {
i = i+1
if (runif(1,0,1) > x$csum[i]){
Parent1 <- x[i-1,]
break
}
}

j <- 1
Parent2 <- x[M,]
repeat {
j = j+1
if ((runif(1,0,1) > x$csum[j]) & ((j-1) != (i-1))){
Parent2 <- x[j-1,]
break
}
}
Parents <- rbind(Parent1, Parent2)
return(Parents)
}

##### Arithmetic Crossover #####
Crossover <- function(pc) {
Gene_no <- 1
Parents <- RW_selection(fit(population))
Parent1 <- Parents[1,1:1]
Parent2 <- Parents[2,1:1]

```

```

child1 <- data.frame(matrix(ncol = Gene_no, nrow = 1))
child2 <- data.frame(matrix(ncol = Gene_no, nrow = 1))
for(j in 1:Gene_no){
alpha <- runif(1,0,1)
child1[j] <- (alpha * Parent1[1,j]) + ((1-alpha) * Parent2[1,j])
child2[j] <- ((1-alpha) * Parent1[1,j]) + (alpha * Parent2[1,j])
}
if (runif(1,0,1) <= pc) {
child1 <- child1
}else{
child1 <- Parent1
}
if (runif(1,0,1) <= pc) {
child2 <- child2
}else{
child2 <- Parent2
}
children <- rbind(child1, child2)
return(children)
}
##### Uniform Mutation #####
Mutation <- function(l,pc,pm,lb,ub) {
children <- Crossover(pc)
child1 <- children [1,]
child2 <- children [2,]
for (j in 1:l){
if(runif(1,0,1) < pm) {
child1[j] <- ((ub[j]- lb[j]) * runif(1,0,1) + lb[j])
}
}
for (j in 1:l){
if(runif(1,0,1) < pm) {
child2[j] <- ((ub[j]- lb[j]) * runif(1,0,1) + lb[j])
}
}
mutate_child1<- child1
mutate_child2<- child2
mutate_children <- rbind(mutate_child1, mutate_child2)
return (mutate_children)
}
##### New Population after Selection, Crossover, Mutation #####
new_pop <- function(pc,pm) {
df <- data.frame(matrix(ncol = 1, nrow = M))
for(i in seq(from=1, to=M,by=2)) {
df[i:(i+1),] <- Mutation(1,pc,pm,lb,ub)
}
return(df)
}
##### New Population & Elitism #####
new_population <- function(pc, pm, Er) {
Elite_no <- round(Er*M)
x <- fit(population)
elites <- data.frame(matrix(ncol = 3, nrow = Elite_no))
x <- x[order(-x$fitness_values),]
elites[1:Elite_no,] <- x[1:Elite_no,]
new_pop <- new_pop(pc,pm)
new_pop <- fit(new_pop)
}

```

```

new_pop <- new_pop[order(new_pop$fitness_values),]
new_pop[1:Elite_no,] <- elites[1:Elite_no,]
return(new_pop)
}
population <- new_population(pc, pm, Er)
chromo <- population[(population$fitness_values) == (max(population$fitness_values)),]
best <- chromo[1,3]
sol1 <- chromo[1,1]
sol2 <- chromo[1,2]
##### Statistics #####
mgene1 <- mean(population$X1)
mgene2 <- mean(population$X2)
sdgene1 <- sd(population$X1)
sdgene2 <- sd(population$X2)
m <- mean(population$fitness_values)
s <- sd(population$fitness_values)
cat("GA| ", "iter = (",g,")|", "Mean(F) = (",m,")|", "Sd(F) = (",s,")|", "Best = (",best,")|", "\n")

##### Gene Position Curves #####
points(population$X1,population$X2,col="red", pch = 18)
}
##### end of LOOP #####
cat("GA RESULTS", "\n", "Fitness Function Value = ",best, "", "\n", "Solution", "\n",
"X1 = ",sol1, " | ", "X2 = ",sol2, "\n", "Mean(X1) = ",mgene1, " |", "Sd(X1) = ",sdgene1, "\n",
"Mean(X2) = ",mgene2, " |", "Sd(X2) = ",sdgene2, "\n")
##### end of full function #####

```

- Παραμετροποίηση για 10^2 εκτελέσεις του αλγορίθμου.

```

#### CONTROLLING PARAMETERS OF GA ALGORITHM #####
Problem.nVar <- 2           # Number of problem Variables
M <- 40                    # Number of chromosomes
l <- Problem.nVar         # Number of genes
MaxGen <- 40              # Generations (iterations)
pc <- 0.86                 # Probability of Crossover
pm <- 0.1                  # Probability of Mutation
Er <- 0.05                 # Elitism Ratio
ub <- c(4,4)              # UPPER BOUND FOR SEARCH SPACE
lb <- c(-4,-4)           # LOWER BOUND FOR SEARCH SPACE

#### 10^2 ITERATIONS OF GA ALGORITHM #####
z <- replicate(100, CGenetic_Algorithm (M, l, MaxGen, pc, pm, Er, lb, ub))
class(z)
mean(z[1,])
sd(z[1,])
mean(z[2,])
sd(z[2,])
mean(z[3,])
sd(z[3,])

```

```

## MSE Calculation ###
mse <- sqrt((0.999599-mean(z[1,]))^2 +(2.498955-mean(z[2,]))^2);mse
##### 1 #####
p_cross <- c(0,0.1,0.2,0.4,0.6,0.8,0.82,0.84,0.85,0.86,0.87,0.88,0.95,1)
MSE <- c(0.00833536,0.00857615,0.00853147,0.00471381,
         0.00486477,0.00379453,0.00334768,0.00266367,
         0.00221851,0.00076927,0.00333253,0.00333277,
         0.00317604,0.00475999)
par(mfrow=c(1,2))
plot(p_cross, log(MSE) ,pch = 18,main=" Διάγραμμα διασποράς του log(MSE) των x1,x2
      συναρτήσε του pc " )
points(0.86,log(0.00076927),col="red",pch = 18)
### Εύρεση καμπύλης ###
# Διάγραμμα διασποράς #
plot(p_cross, MSE ,pch = 18,main=" Διάγραμμα διασποράς του MSE των x1,x2
      συναρτήσε του pc και καμπύλες παλινδρόμησης" )
points(0.86,0.00076927,col="red",pch = 18)
x <- seq(0,1,length=14)
y <- MSE
#second degree
fit2 <- lm(y~poly(x,2,row=TRUE))
#third degree
fit3 <- lm(y~poly(x,3,row=TRUE))
#fourth degree
fit4 <- lm(y~poly(x,4,row=TRUE))
lines(x, predict(fit2, data.frame(x=x)), col="green")
lines(x, predict(fit3, data.frame(x=x)), col="blue")
lines(x, predict(fit4, data.frame(x=x)), col="purple")
##### 2 #####
p_mute <- c(seq(0,0.11,0.01))
MSE <- c(0.05790189,0.06869783,0.04357312,0.00842478,
         0.00583836,0.00515544,0.00356237,0.00211789,
         0.00287929,0.00246721,0.00076927,0.00177242)
par(mfrow=c(1,2))
plot(p_mute, log(MSE) ,pch = 18,main=" Διάγραμμα διασποράς του log(MSE) των x1,x2
      συναρτήσε του pm " )
points(0.10,log(0.00076927),col="red",pch = 18)
### Εύρεση καμπύλης ###
# Διάγραμμα διασποράς #
plot(p_mute, MSE ,pch = 18,main=" Διάγραμμα διασποράς του MSE των x1,x2
      συναρτήσε του pm και καμπύλες παλινδρόμησης" )
points(0.10,0.00076927,col="red",pch = 18)
x <- seq(0,0.11,length=12);x
y <- MSE
#second degree
fit2 <- lm(y~poly(x,2,row=TRUE))
#third degree
fit3 <- lm(y~poly(x,3,row=TRUE))
lines(x, predict(fit2, data.frame(x=x)), col="green")
lines(x, predict(fit3, data.frame(x=x)), col="blue")
##### 3 #####
e_ratio <- c(0,0.05,0.10,0.20,0.50,0.80,1.00)
MSE <- c(0.00411169,0.00076927,0.00210910,0.00342168,
         0.00345228,0.00525867,0.11672370)
par(mfrow=c(1,2))
plot(e_ratio, log(MSE) ,pch = 18, main=" Διάγραμμα διασποράς του log(MSE) των x1,x
2

```



```

        συναρτήσει του Er" )
points(0.05,log(0.00076927),col="red",pch = 18)
### Εύρεση καμπύλης ###
# Διάγραμμα διασποράς #
plot(e_ratio, MSE,pch = 18,main=" Διάγραμμα διασποράς του MSE των x1,x2
        συναρτήσει του Er" )
points(0.05,0.00076927,col="red",pch = 18)
x <- seq(0,1,length=7);x
y <- MSE
#fourth degree
fit4 <- lm(y~poly(x,4,row=TRUE))
lines(x, predict(fit4, data.frame(x=x)), col="purple")
##### 4 #####
par(mfrow=c(2,2))
min(z[1,])
max(z[1,])
hist(z[1,],seq(0.90,1.05,0.01),ylim=c(0,60),
      xlim=c(0.9,1.05),xlab="",ylab="",main=" Ιστόγραμμα συχνότητων 100 εκτιμήσεων
του x1
      για pc = 0.86, pm = 0.1, Er = 0.05")
#
min(z[2,])
max(z[2,])
hist(z[2,],seq(2.40,2.60,0.01),ylim=c(0,60),
      xlim=c(2.40,2.60),xlab="",ylab="",main=" Ιστόγραμμα συχνότητων 100 εκτιμήσεων
του x2
      για pc = 0.86, pm = 0.1, Er = 0.05")
#
x1 <- seq(-3, 4, length = 30)
x2 <- seq(-3, 6, length = 30)
bouts <- function(x1,x2){
  return(0.2*dnorm(x1,-1,1)*dnorm(x2,-1,1)+0.8*
         (exp((-8/15)*((x1-1)^2+((x2-2.5)^2)
              /2-((x1-1)*(x2-2.5))/4)))/(pi*sqrt(15)))
}
f <- outer(x1, x2, bouts)
monitor <- function(x) {
  contour(x1, x2, f,drawlabels = FALSE, col=gray.colors(7))
  title(" Κατανομή 100 Εκτιμήσεων των x1,x2 γύρω από την
        Θεωρητική βέλτιστη λύση για
        pc = 0.86, pm = 0.1, Er = 0.05 ", font = 4)
  Sys.sleep(0.2)
}
monitor(bouts)
points(-0.955467,-0.956622,col="blue", pch = 8)
points(0.999599,2.498955,col="green", pch = 8)
points(z[1,],z[2,],col="red", pch = 18)
#
x1 <- seq(0.90, 1.05, length = 30)
x2 <- seq(2.40,2.60, length = 30)
bouts <- function(x1,x2){
  return(0.2*dnorm(x1,-1,1)*dnorm(x2,-1,1)+0.8*
         (exp((-8/15)*((x1-1)^2+((x2-2.5)^2)
              /2-((x1-1)*(x2-2.5))/4)))/(pi*sqrt(15)))
}
f <- outer(x1, x2, bouts)
monitor <- function(x) {

```

```

contour(x1, x2, f, drawlabels = FALSE, col=gray.colors(7))
title(" Κατανομή 100 Εκτιμήσεων των x1,x2 γύρω από την
      θεωρητική βέλτιστη λύση για
      pc = 0.86, pm = 0.1, Er = 0.05 ", font = 4)
Sys.sleep(0.2)
}
monitor(bouts)
points(0.999599,2.498955,col="green", pch = 8)
points(z[1,],z[2,],col="red", pch = 3)
#####

```

ΠΑΡΑΡΤΗΜΑ Π3

Κώδικας:

Εισαγωγή δεδομένων δημιουργία πίνακα

```
##### ME data.frame #####
f <- NULL
files <- c("MSFT.csv", "AMZN.csv", "AAPL.csv", "FB.csv", "NFLX.csv", "^GSPC.csv")
for (i in 1:length(files)) {
  csv <- read.csv(files[i])
  csv <- csv[,c("Date", "Close")]
  names(csv) <- c("Date", files[i])
  if (i == 1) f <- csv
  else f <- merge(f, csv)
}
head(f, 5)
tail(f, 5)
```

Κώδικας:

Χρονοσειρά ημερήσιας κίνησης μετοχών και δείκτη

```
par(mfrow=c(3,3))
ts.plot(f[,2], main="MSFT", ylab="Price P(t)")
ts.plot(f[,3], main="AMZN", ylab="Price P(t)")
ts.plot(f[,4], main="AAPL", ylab="Price P(t)")
ts.plot(f[,5], main="FB", ylab="Price P(t)")
ts.plot(f[,6], main="NFLX", ylab="Price P(t)")
ts.plot(f[,7], main="^GSPC", ylab="Price P(t)")
par(mfrow=c(1,1))
```

Κώδικας:

Χρονοσειρά αποδόσεων

```
par(mfrow=c(3,3))
ts.plot(asset_returns[,1], main="MSFT", ylab="Returns R(t)")
ts.plot(asset_returns[,2], main="AMZN", ylab="Returns R(t)")
ts.plot(asset_returns[,3], main="AAPL", ylab="Returns R(t)")
ts.plot(asset_returns[,4], main="FB", ylab="Returns R(t)")
ts.plot(asset_returns[,5], main="NFLX", ylab="Returns R(t)")
ts.plot(asset_returns[,6], main="^GSPC", ylab="Returns R(t)")
par(mfrow=c(1,1))
```

Κώδικας:

Περιγραφικά μέτρα αποδόσεων μετοχών

```
library(psych)
Mean<-round(sapply(asset_returns,mean),5)
Median<-round(sapply(asset_returns,median),5)
St.Dev.<-round(sapply(asset_returns,sd),5)
Max<-round(sapply(asset_returns,max),5)
Min<-round(sapply(asset_returns,min),5)
Skewness<-round(sapply(asset_returns,skew),5)
Kurtosis<-round(sapply(asset_returns,kurtosi),5)
descr<-rbind(Mean,Median,St.Dev.,Max,Min,Skewness,Kurtosis)
```

Κώδικας:

Διαγράμματα MSFT,AAMZN,AAPL,FB,NFLX,GSPC

```
min(asset_returns[,1:6])
max(asset_returns[,1:6])
par(mfrow=c(1,2))
hist(asset_returns[,1:6],seq( , ,0.005),ylim=c(0, ),
xlim=c( , ),xlab="",ylab="",main=" returns")
qqnorm(asset_returns[,1:6]);qqline(asset_returns[,1:6])
```

Κώδικας:

Περιγραφικά μέτρα βέλτιστου χαρτοφυλακίου

```
P<-round(mean(portfolio_returns(solution)),6)
V<-round(sd(portfolio_returns(solution)),6)
P_final<-matrix(c(P,V),2,1)
P_final<-as.data.frame(P_final)
names(P_final)<-c("Optimal Port")
descr2<-rbind(Mean,St.Dev.)
descr_final<-cbind(P_final,descr2)
```

Κώδικας:

Γραφήματα Βέλτιστου χαρτοφυλακίου

```
par(mfrow=c(1,3))
min(portfolio_returns(solution))
max(portfolio_returns(solution))
ts.plot(portfolio_returns(solution),main="Optimal Port returns")
hist(portfolio_returns(solution),seq(-0.06,0.08,0.005),ylim=c(0,350),
xlim=c(-0.06,0.08),xlab="",ylab="",main="Optimal Port returns")
qqnorm(portfolio_returns(solution));qqline(portfolio_returns(solution))
```

Κώδικας:

Συγκεντρωτικό Γράφημα όλων των αποδόσεων

```
optimal_returns = portfolio_returns(solution)
plot(cumsum(optimal_returns), type="l", lwd=5, ylim=c(0,3))
lines(cumsum(asset_returns[,1]), col="blue")
lines(cumsum(asset_returns[,2]), col="red")
lines(cumsum(asset_returns[,3]), col="green")
lines(cumsum(asset_returns[,4]), col="violet")
lines(cumsum(asset_returns[,5]), col="peru")
lines(cumsum(asset_returns[,6]), col="pink")
legend(locator(1), legend=c("Weighted portfolio", names(asset_returns)),
col = c("black", "blue", "red", "green", "violet", "peru", "pink"), lty=1)
```


ΒΙΒΛΙΟΓΡΑΦΙΑ

Ελληνική

1. Βασιλείου Δημήτριος, Ηρειώτης Νικόλαος, (2015). *Ανάλυση επενδύσεων και Διαχείριση Χαρτοφυλακίου*, Εκδόσεις Rosili.
2. Παναγιώτης Ξυδónας, Ιωάννης Ψαρράς, Κωνσταντίνος Ζοπουνίδης, (2010) *Σύγχρονη Θεωρία Χαρτοφυλακίου*, Κλειδάριθμος.
3. Παπαδήμου Στέφανος, (2009). *Διαχείριση Χαρτοφυλακίου μία σύγχρονη προσέγγιση*, Εκδόσεις Gutenberg.
4. Γεωργίου Σ. Οικονόμου Ανδρέας Κ. Γεωργίου (2016). *Επιχειρησιακή έρευνα για τη λήψη διοικητικών αποφάσεων* Μπένου.
5. Hamdy A. Taha (2011). *Εισαγωγή στην επιχειρησιακή έρευνα*, Εκδόσεις Τζιόλα.
6. Tan, Steinbach, Kumar (2017) *Εισαγωγή στην εξόρυξη δεδομένων*, Εκδόσεις Τζιόλα.
7. Γλεζάκος Μ., *Συνοπτικές Σημειώσεις στην Διαχείριση Χαρτοφυλακίου Επενδύσεων*, Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης, Πανεπιστήμιο Πειραιώς, 2016.
8. Μπούτσικας Μ., *Σημειώσεις μαθήματος (Χρηματοοικονομική) Διοίκηση Κινδύνου*, Τμήμα στατιστικής και Ασφαλιστικής Επιστήμης, Πανεπιστήμιο Πειραιώς, 2018.
9. Μπούτσικας Μ. (2004) *Μέθοδοι Προσομοίωσης και Στατιστικές Υπολογιστικές Τεχνικές*, Πανεπιστημιακές σημειώσεις, Πανεπιστήμιο Πειραιώς.
10. Ανθρωπέλος Μ., *Σημειώσεις στην Διαχείριση Χαρτοφυλακίου Επενδύσεων*, Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης, Πανεπιστήμιο Πειραιώς, 2017.
11. Δράκος Κ., *Σημειώσεις στην Διοίκηση Κινδύνου*, Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης, Πανεπιστήμιο Πειραιά, 2017.

Ξένη

1. Markowitz H. - *Portfolio Selection, Efficient Diversification of investment*, John Wiley and Sons, New York, 1959
2. William F. Sharpe - *A simplified model for Portfolio Analysis*, Management Science, 1963.
3. William F. Sharpe , Gordon J. Alexander , Jeffery V. Bailey - *Investments*, Prentice Hall,1999.
4. Bodie, Z., Kane, A. and Marcus, A. (2013). *Investments*. McGraw-Hill, 10th
5. A. Gupta - *Risk Management and Simulation*, 2014 by Taylor & Francis Group, LLC
6. Dietmar G. Maringer - *Portfolio Management with Heuristic Optimization* (2010, Springer)
7. Ali Hirta (2012) *Computational Methods in Finance*
8. RACHEV - *Advanced Stochastic Models, Risk Assessment, and Portfolio Optimization*, 2008 by John Wiley & Sons, Inc.
9. Colin R. Reeves - *Modern heuristic techniques for combinatorial problems*, John Wiley & Sons, Inc. 1993
10. Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. A Bradford Book.
11. John H. Holland - *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press Cambridge 1992
12. M. Gendreau, J-Y Potvin (2010) *Handbook of Metaheuristics*, Springer, New York
13. Markowitz, H 1952, *Portfolio selection, The Journal of Finance*, Vol.7,

14. Kambiz Forqandoost Haqiqi and Tohid Kazemi - *International Journal of Trade, Economics and Finance*, Vol. 3, No. 2, April 2012.
15. Zeinodin Alizadeh, Hossein Panahian - *International Journal of Computer Applications* (0975 – 8887) Volume 50 – No.2, July 2012
16. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi - *Science, New Series*, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680.

Πηγές από Ιστοχώρους

1. <https://finance.yahoo.com/>
2. <https://luca-scr.github.io/GA/>
3. <https://medium.com/data-science-journal>
4. <https://medium.com/@gianlucamalato>
5. <https://attic.gsfc.nasa.gov/ants/>
6. <https://docplayer.gr/38488441-Ypologistiki-noimosyni.html>
7. <https://www.udemy.com/>

