



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΣΧΕΔΙΑΣΗΣ ΠΡΟΪΟΝΤΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ

ΚΑΡΑΓΙΑΝΝΙΔΗ ΧΡΙΣΤΙΝΑ

ΑΜ: 5112001/017

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ:

**ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΒΙΟΜΗΧΑΝΙΚΗΣ ΓΡΑΜΜΗΣ ΠΑΡΑΓΩΓΗΣ ΜΕ ΧΡΗΣΗ
ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ**

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΜΟΥΛΙΑΝΙΤΗΣ ΒΑΣΙΛΕΙΟΣ

ΣΥΡΟΣ, ΔΕΚΕΜΒΡΙΟΣ 2007

Περίληψη:

Αυτή η εργασία αναφέρεται στο πρόβλημα βελτιστοποίησης της βιομηχανικής γραμμής παραγωγής, όπου λαμβάνεται υπόψη και η αλληλεπίδραση μεταξύ των κριτηρίων βελτιστοποίησης. Τρία γνωστά κριτήρια τίθενται ταυτόχρονα προς ελαχιστοποίηση: το κόστος κατασκευής (makespan), ο μέγιστος χρόνος καθυστερήσεων (total tardiness) και ο συνολικός χρόνος ροής (total flow time). Σε αυτό το πρόβλημα προτείνεται ο πολυ- αντικειμενικός γενετικός αλγόριθμος (Multi-Objective Genetic Algorithm, MOGA) για την επίλυσή του, με τα ακόλουθα κύρια χαρακτηριστικά: α) χρησιμοποιεί τη μέθοδο της ασυνεχής ολοκλήρωσης Choquet, ώστε να διατυπώσει τη συνάρτηση κόστους της κάθε ατομικής λύσης του προβλήματος, συνυπολογίζοντας έτσι και την αλληλεπίδραση μεταξύ των τριών αντικειμενικών, β) διατηρεί ένα ξεχωριστό πληθυσμό με ποικιλία στις βέλτιστες λύσεις Pareto, γ) «ψεκάζει» το τρέχων εξελισσόμενο πληθυσμό με κάποιες βέλτιστες λύσεις Pareto, και δ) δημιουργεί νέες ατομικές λύσεις χρησιμοποιώντας μια προσαρμοσμένη συνεργία ενός ζεύγους τελεστών διασταύρωσης. Τέλος, χρησιμοποιούνται συνδυασμοί αντικειμενικών συναρτήσεων και εξάγονται συμπεράσματα, για το αν και ποιος συνδυασμός καταφέρνει να επιλύσει το πρόβλημα, προσφέροντας λύση ανώτερη των υπολοίπων.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

| | |
|--|----|
| ΠΕΡΙΛΗΨΗ:..... | 2 |
| ΕΙΣΑΓΩΓΗ: | 7 |
| 1. ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ: | 11 |
| 2. ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ: | 11 |
| 2.1 ΤΙ ΕΙΝΑΙ ΟΙ ΑΛΓΟΡΙΘΜΟΙ: | 11 |
| 2.2 ΤΙ ΕΙΝΑΙ ΟΙ ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ: | 13 |
| <i>ΓΕΝΕΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ:</i> | 15 |
| <i>Η ΟΡΘΗ ΣΥΝΑΡΤΗΣΗ:</i> | 15 |
| <i>ΕΠΙΛΟΓΗ:</i> | 15 |
| <i>ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ:</i> | 16 |
| <i>ΦΑΙΝΟΤΥΠΟΣ:</i> | 16 |
| <i>ΦΑΙΝΟΤΥΠΙΚΗ ΠΟΙΚΙΛΙΑ:</i> | 16 |
| <i>ΓΟΝΙΔΙΑ:</i> | 17 |
| <i>ΓΟΝΟΤΥΠΟΣ:</i> | 17 |
| <i>ΑΝΑΠΑΡΑΣΤΑΣΗ ΥΠΟΨΗΦΙΩΝ ΛΥΣΕΩΝ:</i> | 17 |
| 2.3 ΓΕΝΕΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ: | 18 |
| 2.4 ΑΞΙΟΠΟΙΗΣΗ ΚΑΙ ΔΙΕΡΕΥΝΗΣΗ: | 20 |
| 2.5 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ: | 20 |
| <i>ΣΤΟΧΟΣ ΤΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ:</i> | 21 |
| 2.6 Η ΤΥΧΑΙΑ ΑΝΑΖΗΤΗΣΗ ΣΤΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ: | 21 |
| 2.7 ΠΟΥ ΔΙΑΦΕΡΟΥΝ ΟΙ ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΑΠΟ ΤΙΣ ΠΑΡΑΔΟΣΙΑΚΕΣ ΜΕΘΟΔΟΥΣ:..... | 22 |
| 3. ΟΙ ΘΕΩΡΗΣΕΙΣ ΤΩΝ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ (ΓΑΣ): | 24 |
| 3.1 ΒΑΣΙΚΟΣ ΜΗΧΑΝΙΣΜΟΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΓΑ: | 25 |
| 3.2 ΕΦΑΡΜΟΓΗ ΣΤΟΝ ΑΛΓΟΡΙΘΜΟ: | 27 |
| 4. ΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ (SCHEDULING) ΠΡΟΒΛΗΜΑ:..... | 32 |

| | | |
|-----|---|----|
| 4.1 | ΤΟ ΠΡΟΒΛΗΜΑ ΑΣΥΣΧΕΤΙΣΤΩΝ ΜΗΧΑΝΩΝ(UNRELATED MACHINES):..... | 33 |
| 4.2 | Ο ΠΛΗΘΥΣΜΟΣ ΤΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΚΑΤΑΛΛΗΛΟΤΗΤΑΣ:..... | 34 |
| 4.3 | ΡΑΡΕΤΟ- ΒΕΛΤΙΣΤΕΣ ΛΥΣΕΙΣ: | 35 |
| 4.4 | ΠΟΛΥ- ΚΡΙΤΗΡΙΑΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ: | 36 |
| | <i>I. ΜΗ ΑΛΛΗΛΕΠΙΔΡΩΝΤΑ ΚΡΙΤΗΡΙΑ:</i> | 36 |
| | <i>II. ΑΛΛΗΛΕΠΙΔΡΩΝΤΑ ΚΡΙΤΗΡΙΑ:</i> | 38 |
| 4.5 | ΓΙΑΤΙ ΧΡΗΣΙΜΟΠΟΙΟΥΝΤΑΙ ΟΙ ΓΑ ΣΤΑ ΠΟΛΥ- ΥΠΟΚΕΙΜΕΝΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΠΡΟΒΛΗΜΑΤΑ: | 40 |
| 4.6 | ΈΝΑΣ ΓΕΝΕΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ (MOGA) ΣΤΑ ΠΟΛΥ- ΥΠΟΚΕΙΜΕΝΙΚΑ PMSP: 41 | |
| 5. | ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ:..... | 42 |
| | 5.1 ΒΑΣΙΚΕΣ ΥΠΟΘΕΣΕΙΣ:..... | 42 |
| | 5.2 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΥΠΟΚΕΙΜΕΝΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ:..... | 42 |
| | 5.3 ΚΑΘΟΡΙΣΜΟΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΚΑΤΑΛΛΗΛΟΤΗΤΑΣ: | 44 |
| | 5.4 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ: | 48 |
| 6. | ΠΑΡΑΔΕΙΓΜΑ: | 49 |
| 7. | ΑΠΟΤΕΛΕΣΜΑΤΑ:..... | 57 |
| 8. | ΥΠΟΛΟΓΙΣΤΙΚΟ ΚΟΣΤΟΣ: | 66 |
| 9. | ΣΥΜΠΕΡΑΣΜΑΤΑ: | 68 |
| 10. | ΒΙΒΛΙΟΓΡΑΦΙΑ: | 70 |
| 11. | ΠΑΡΑΡΤΗΜΑ:..... | 73 |

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

| | |
|--|----|
| ΕΙΚΟΝΑ 1. ΤΑ ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ ΧΡΗΣΙΜΟΠΟΙΟΥΝΤΑΙ ΣΥΧΝΑ ΩΣ ΓΡΑΦΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΩΝ ΑΛΓΟΡΙΘΜΩΝ. | 12 |
| ΕΙΚΟΝΑ 2. ΜΕΤΑΒΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ ΜΕΤΑΞΥ ΔΙΑΔΟΧΙΚΩΝ ΓΕΝΕΩΝ. | 18 |
| ΕΙΚΟΝΑ 3. Ο ΚΥΚΛΟΣ ΤΟΥ ΓΕΝΕΤΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ. | 19 |
| ΕΙΚΟΝΑ 4. Η ΓΕΝΙΚΗ ΔΟΜΗ ΤΩΝ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ. | 25 |
| ΕΙΚΟΝΑ 5. ΓΕΝΙΚΟ ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ. | 26 |
| ΕΙΚΟΝΑ 6. ΔΙΑΣΤΑΥΡΩΣΗ ΕΝΟΣ- ΣΗΜΕΙΟΥ (ONE- POINT CROSSOVER). | 28 |
| ΕΙΚΟΝΑ 7. ΔΙΑΣΤΑΥΡΩΣΗ ΤΡΙΩΝ- ΣΗΜΕΙΩΝ (THREE- POINT CROSSOVER). | 28 |
| ΕΙΚΟΝΑ 8. ΜΕΤΑΛΛΑΞΗ (MUTATION). | 29 |
| ΕΙΚΟΝΑ 9. ΠΑΡΑΔΕΙΓΜΑ "ΤΟΠΙΟΥ" ΤΗΣ FITNESS. | 36 |
| ΕΙΚΟΝΑ 10. ΑΛΛΗΛΕΠΙΔΡΩΝΤΑ ΚΑΙ ΜΗ- ΑΛΛΗΛΕΠΙΔΡΩΝΤΑ ΚΡΙΤΗΡΙΑ. | 38 |
| ΕΙΚΟΝΑ 11. ΠΙΝΑΚΑΣ ΧΡΟΝΟΥ ΕΠΕΞΕΡΓΑΣΙΑΣ P_1 (PROCESSING TIME) ΜΕ ΜΟΝΑΔΑ ΜΕΤΡΗΣΗΣ ΤΟ ΔΕΥΤΕΡΟΛΕΠΤΟ (SECOND). | 49 |
| ΕΙΚΟΝΑ 12. ΠΙΝΑΚΑΣ ΗΜΕΡΟΜΗΝΙΑΣ ΠΑΡΑΔΟΣΗΣ D_1 (DUE DATE) ΜΕ ΜΟΝΑΔΑ ΜΕΤΡΗΣΗΣ ΤΟ ΔΕΥΤΕΡΟΛΕΠΤΟ (SECOND). (ΘΕΩΡΕΙΤΑΙ $D_0=0$ ΚΑΙ $D_1= D_R- D_0$, ΟΠΟΥ D_R ΕΙΝΑΙ Ο ΧΡΟΝΟΣ ΣΤΟΝ ΟΠΟΙΟ ΕΙΝΑΙ ΕΤΟΙΜΗ ΠΡΟΣ ΠΑΡΑΔΟΣΗ Η ΕΡΓΑΣΙΑ). | 49 |
| ΕΙΚΟΝΑ 13. ΠΙΝΑΚΑΣ ΧΡΟΝΟΥ ΟΛΟΚΛΗΡΩΣΗΣ C_1 (COMPLETION TIME) ΜΕ ΜΟΝΑΔΑ ΜΕΤΡΗΣΗΣ ΤΟ ΔΕΥΤΕΡΟΛΕΠΤΟ (SECOND). | 49 |
| ΕΙΚΟΝΑ 14. ΠΙΝΑΚΑΣ ΧΡΟΝΟΥ ΕΤΟΙΜΑΣΙΑΣ R_1 (READY TIME) ΜΕ ΜΟΝΑΔΑ ΜΕΤΡΗΣΗΣ ΤΟ ΔΕΥΤΕΡΟΛΕΠΤΟ (SECOND). | 49 |
| ΕΙΚΟΝΑ 15. ΠΑΡΑΔΕΙΓΜΑΤΑ ΠΙΘΑΝΩΝ ΣΥΝΔΥΑΣΜΩΝ ΚΑΙ ΥΠΟΛΟΓΙΣΜΟΣ ΟΛΩΝ ΤΩΝ ΧΡΟΝΩΝ ΜΕ ΤΟΥΣ ΔΕΔΟΜΕΝΟΥΣ ΣΥΝΔΥΑΣΜΟΥΣ. | 50 |
| ΕΙΚΟΝΑ 16. ΤΟ ΓΟΝΙΔΙΟ ΚΑΙ ΤΑ ΤΜΗΜΑΤΑ ΤΟΥ. | 51 |
| ΕΙΚΟΝΑ 17. ΑΝΑΠΑΡΑΣΤΑΣΗ ΕΡΓΑΣΙΩΝ ΣΕ ΔΥΑΔΙΚΗ ΜΟΡΦΗ. | 52 |
| ΕΙΚΟΝΑ 18. ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΟΝΙΔΙΩΝ ΣΕ ΔΥΑΔΙΚΗ ΜΟΡΦΗ. | 52 |
| ΕΙΚΟΝΑ 19. ΠΙΝΑΚΑΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΩΝ (3) ΚΡΙΤΗΡΙΩΝ ΤΟΥ ΠΑΡΑΔΕΙΓΜΑΤΟΣ. | 55 |
| ΕΙΚΟΝΑ 20. ΠΙΝΑΚΑΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ. | 65 |

ΠΙΝΑΚΑΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

| | |
|---|----|
| ΔΙΑΓΡΑΜΜΑ 1. ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΩΝ (3) ΜΕΘΟΔΩΝ, ΠΟΥ ΠΑΡΟΥΣΙΑΖΕΙ ΤΗΝ ΑΠΟΔΟΣΗ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ ΣΕ ΣΥΣΧΕΤΙΣΜΟ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΠΡΟΒΛΗΜΑΤΩΝ. | 59 |
| ΔΙΑΓΡΑΜΜΑ 2. ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΩΝ (3) ΜΕΘΟΔΩΝ, ΠΟΥ ΠΑΡΟΥΣΙΑΖΕΙ ΤΙΣ ΤΙΜΕΣ ΤΟΥ ΚΡΙΤΗΡΙΟΥ TFT ΣΕ ΣΥΣΧΕΤΙΣΜΟ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΠΡΟΒΛΗΜΑΤΩΝ. | 60 |
| ΔΙΑΓΡΑΜΜΑ 3. ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΩΝ (3) ΜΕΘΟΔΩΝ, ΠΟΥ ΠΑΡΟΥΣΙΑΖΕΙ ΤΙΣ ΤΙΜΕΣ ΤΟΥ ΚΡΙΤΗΡΙΟΥ TARDINESS ΣΕ ΣΥΣΧΕΤΙΣΜΟ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΠΡΟΒΛΗΜΑΤΩΝ..... | 60 |
| ΔΙΑΓΡΑΜΜΑ 4. ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΩΝ (3) ΜΕΘΟΔΩΝ, ΠΟΥ ΠΑΡΟΥΣΙΑΖΕΙ ΤΙΣ ΤΙΜΕΣ ΤΟΥ ΚΡΙΤΗΡΙΟΥ MAKESPAN ΣΕ ΣΥΣΧΕΤΙΣΜΟ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΠΡΟΒΛΗΜΑΤΩΝ. | 61 |
| ΔΙΑΓΡΑΜΜΑ 5. ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΠΟΥ ΠΑΡΟΥΣΙΑΖΕΙ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΛΥΣΕΩΝ ΣΕ ΣΥΣΧΕΤΙΣΜΟ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΠΡΟΒΛΗΜΑΤΩΝ. | 61 |
| ΔΙΑΓΡΑΜΜΑ 6. ΣΥΓΚΡΙΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΤΩΝ (3) ΜΕΘΟΔΩΝ, ΠΟΥ ΠΑΡΟΥΣΙΑΖΕΙ ΤΟΝ ΥΠΟΛΟΓΙΣΤΙΚΟ ΧΡΟΝΟ ΣΕ ΣΥΣΧΕΤΙΣΜΟ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΠΡΟΒΛΗΜΑΤΩΝ..... | 62 |

Εισαγωγή:

Στη παρούσα διπλωματική γίνεται συγκριτική μελέτη (3) μεθόδων βελτιστοποίησης γραμμής παραγωγής με χρήση γενετικών αλγορίθμων, και αποδεικνύεται η υπερίσχυση, της προτεινόμενης από τη διπλωματική μεθόδου, δηλαδή της ασυνεχής ολοκλήρωσης Choquet. Η σύγκριση γίνεται σε επίπεδο αποτελεσματικότητας των αντικειμενικών συναρτήσεων, όσο αναφορά την ελαχιστοποίηση των επιλεγμένων κριτηρίων: του κόστους κατασκευής (makespan), του συνολικού χρόνου καθυστερήσεων (total tardiness) και του συνολικού χρόνου ροής (total flow time). Το πρόβλημα που παρουσιάζεται ανήκει στη κατηγορία των *Ασυσχέτιστων Μηχανών (Unrelated Machines- Rm)* και περιγράφεται από m - μηχανές / n - εργασίες, όπου κάθε i - εργασία μπορεί να επεξεργασθεί από μια και μόνο μηχανή, σε συγκεκριμένα χρονικά περιθώρια. Η αντικειμενική συνάρτηση θα καθορίσει τη τοποθέτηση- ακολουθία των εργασιών, που θα βελτιστοποιεί ορισμένα μέτρα εκτέλεσης, και κατά συνέπεια τα κριτήρια που έχουν τεθεί.

Η βελτιστοποίηση της γραμμής παραγωγής της βιομηχανίας ανήκει στην ομάδα προβλημάτων προγραμματισμού (scheduling). Η μελέτη του προγραμματισμού έχει τραβήξει το ενδιαφέρον των ερευνητών μέσα στα χρόνια. Η βάση λύσης αυτών, όπως και άλλων προβλημάτων, είναι οι αλγόριθμοι. Οι καλοί αλγόριθμοι προγραμματισμού μπορούν να μειώσουν το συνολικό κόστος παραγωγής σε μια διαδικασία παραγωγής, καθιστώντας την εταιρία ανταγωνιστική. Μετά τον Johnson, ο οποίος ήταν πρωτοπόρος στο τομέα (1954), πολλοί ανέπτυξαν αλγόριθμους και θεωρίες (βασισμένες σε επαναληπτικές βελτιωτικές διαδικασίες) με βάση αυτό το πρόβλημα, όπως ο branch- and- bound αλγόριθμος από τους Ignall, Schrage και Lominicki (1965), ευρετικές προσεγγίσεις από τους Campbell (1970) και Nawaz (1983) με υποκειμενική συνάρτηση ελαχιστοποίησης του κόστους παραγωγής, ευρετικές προσεγγίσεις από τους Bansal και Szwarcz με υποκειμενική συνάρτηση ελαχιστοποίησης του συνολικού χρόνου ροής, ευρετικοί αλγόριθμοι από τους Miyazaki και Ho & Chang (1991) με υποκειμενική συνάρτηση ελαχιστοποίησης του συνολικού χρόνου ροής (ή του αθροίσματος των χρόνων εκπλήρωσης) των εργασιών, simulated annealing heuristics από τους Osman & Potts (1989) και taboo search heuristics από τους Widmer & Hertz (1990) και Taillard (1990), ο τελευταίος

εκ των οποίων, ανέπτυξε ιδιαίτερα και τα benchmarks. Λόγω του ότι και οι υπολογιστικές δυνατότητες που διατίθενται ολοένα και αυξάνονται με τη πάροδο του χρόνου, πολλοί επιστήμονες, όπως ο Holland (1975), ο Goldberg (1989) και ο Davis (1991), εφήρμοσαν τους γενετικούς αλγορίθμους σε συνδυαστικά προβλήματα βελτιστοποίησης όπως αυτά του «περιπλανώμενου πωλητή», και κάποιοι άλλοι, όπως οι Fox & McMahon (1991), ο Ishibuchi (1994) και ο Manderick (1994) τους εφήρμοσαν σε προβλήματα scheduling.

Η διπλωματική αυτή χρησιμοποιεί σαν μέσο αναζήτησης της λύσης του προβλήματος, τους Γενετικούς Αλγόριθμους. Η εισαγωγή των γενετικών αλγορίθμων έγινε το 1958 από τον Friedberg, ο οποίος επιχείρησε να συνδυάσει μικρά προγράμματα Fortran, ωστόσο τα προγράμματα που προέκυψαν τις περισσότερες φορές δεν ήταν εκτελέσιμα. Ο Holland το 1975, έδωσε μια νέα ώθηση στο χώρο, χρησιμοποιώντας σειρές bits για να αναπαραστήσει λειτουργίες, με τρόπο τέτοιο ώστε κάθε συνδυασμός bits να είναι μια έγκυρη λειτουργία.

Οι γενετικοί αλγόριθμοι λειτουργούν ανιχνεύοντας ένα χώρο πιθανών λύσεων, με άλλα λόγια οι γενετικοί αλγόριθμοι εξετάζουν πολλαπλές λύσεις με μια «ματιά», για να βρεθεί η βέλτιστη λύση. Αυτό είναι που επιτρέπει σε ένα γενετικό αλγόριθμο να βρίσκει μια λύση γρηγορότερα (μικρότερος χρόνος εκτέλεσης) από κάποιον άλλο αλγόριθμο. Οι γενετικοί αλγόριθμοι βασίζονται στις εξελικτικές αρχές του Δαρβίνου, δηλαδή στη φυσική μετάλλαξη και στην επιβίωση του «καλύτερου» ατόμου. Κατά συνέπεια μέσα στους γενετικούς αλγορίθμους λειτουργούν διαδικασίες επιλογής, ανταλλαγής πληροφοριών και μετάλλαξης.

Το πρόβλημα που αντιμετωπίζεται είναι ένα πολύπλοκο πρόβλημα, με πολυπλοκότητα (complexity): $m * \sum_{i=1}^2 \binom{n}{i}$, όπου n : αριθμός εργασιών, m : αριθμός μηχανών. Για τη λύση του προβλήματος επιλέγεται ο πολυ- αντικειμενικός γενετικός αλγόριθμος (MOGA, multi- objective genetic algorithm) με ποικίλες διευθύνσεις αναζήτησης. Υπάρχουν δυο χαρακτηριστικά στοιχεία στον αλγόριθμο MOGA. Το 1^ο είναι η διαδικασία επιλογής: στη διαδικασία επιλογής, ο πολυ- υποκειμενικός γενετικός αλγόριθμος, χρησιμοποιεί ένα άθροισμα από πολυ- υποκειμενικές συναρτήσεις με βάρος, για να τις συνδυάσει σε μια μονόμετρη συνάρτηση

καταλληλότητας. Τα βάρη που συνδέονται με τις πολλαπλές υποκειμενικές συναρτήσεις δεν είναι σταθερά αλλά καθορίζονται τυχαία για κάθε επιλογή. Γι' αυτό η κατεύθυνση της αναζήτησης του αλγορίθμου δεν είναι σταθερή. Το 2^ο χαρακτηριστικό του αλγορίθμου, είναι η στρατηγική διαφύλαξης της ελίτ: ένα προσωρινό σετ από Pareto βέλτιστες λύσεις, διατηρείται κατά την εκτέλεση του πολυ- υποκειμενικού γενετικού αλγορίθμου (ο γενετικός αλγόριθμος μιας- υποκειμενικής με σταθερά βάρη δε μπορεί να βρει ένα concave Pareto front ακόμα κι αν εφαρμοστούν ποικίλες τιμές βαρών). Ένας συγκεκριμένος αριθμός από άτομα σ' αυτό το σετ κληρονομούνται στην επόμενη γενιά, σαν επίλεκτα άτομα. Ο αλγόριθμος εκτελείται στο πρόβλημα, έχοντας τρεις υποκειμενικές συναρτήσεις: την ελαχιστοποίηση του κόστους κατασκευής (makespan), την ελαχιστοποίηση των συνολικών καθυστερήσεων (total tardiness) και την ελαχιστοποίηση του συνολικού χρόνου ροής (total flowtime). Επίσης χρησιμοποιούνται γενετικοί τελεστές για να σχεδιαστεί ο γενετικός αλγόριθμος, και σε αυτή τη περίπτωση, αυτοί που εξυπηρετούν περισσότερο είναι η διασταύρωση τριών- σημείων (three- point crossover) και η μετάλλαξη (shift mutation).

Το κόστος κατασκευής και οι συνολικές καθυστερήσεις είναι γνωστό πως είναι κριτήρια που καθορίζουν σε πάρα πολύ μεγάλο ποσοστό τη γραμμή παραγωγής. Το κριτήριο της ελαχιστοποίησης του συνολικού χρόνου ροής (flowtime) βρέθηκε πως μειώνει σημαντικά το κόστος του προγραμματισμού (scheduling). Ακόμη, παρατηρείται σε εφαρμογές στη βιομηχανία, πως το κριτήριο της ελαχιστοποίησης του συνολικού χρόνου ροής είναι μια σημαντική υποκειμενική για τις βιομηχανίες, αφού καταλήγει στη καθ' αυτή χρησιμοποίησιμότητα των πόρων, στο κύκλο των τελειωμένων εργασιών και στη μείωση της σε εξέλιξη απογραφής.

Τα πολυ- υποκειμενικά προβλήματα βελτιστοποίησης εστιάζουν στο πως θα βρεθούν όλες οι δυνατές ανταλλαγές μεταξύ των πολλαπλών υποκειμενικών συναρτήσεων, που συνήθως συγκρούονται. Επειδή είναι δύσκολο να επιλέξουμε μια μοναδική λύση για ένα πολυ- υποκειμενικό πρόβλημα βελτιστοποίησης χωρίς την επαναλαμβανόμενη διάδραση με το λήπτη- αποφάσεων, μια γενική προσέγγιση είναι να δείξουμε ένα σετ από βέλτιστες λύσεις Pareto στο λήπτη- αποφάσεων, και τότε εκείνος θα είναι σε θέση να επιλέξει οποιαδήποτε από τις βέλτιστες λύσεις Pareto.

Για να βρεθούν όλες οι Pareto βέλτιστες λύσεις από τους γενετικούς αλγόριθμους, η ποικιλία των ατόμων πρέπει να διατηρηθεί σε κάθε γενιά.

Έχει ερευνηθεί ότι ο πολυ- υποκειμενικός γενετικός αλγόριθμος (MOGA) είναι ανώτερος (έχει υψηλότερη απόδοση) από άλλους αλγόριθμους, π.χ. γενετικός αλγόριθμος μιας υποκειμενικής συνάρτησης, Schaffer's VEGA (Vector Evaluated Genetic Algorithm), κ.τ.λ., όσο αναφορά το συγκεκριμένο πρόβλημα. Και αυτό γιατί οι λύσεις των άλλων αλγορίθμων είναι κυριαρχημένες από τις λύσεις που βρίσκει ο πολυ- υποκειμενικός γενετικός αλγόριθμος. Ακόμη και όταν βρίσκονται από τους άλλους αλγόριθμους κάποιες καλύτερες λύσεις, παρ' όλα αυτά αποτυγχάνουν να βρουν μεγάλα Pareto fronts. Επίσης, πολύ μεγάλο ρόλο στην επιλογή αλγορίθμου παίζει το ποσοστό του χώρου αναζήτησης (searching space) το οποίο ελέγχει. Χρησιμοποιώντας το πολυ- υποκειμενικό γενετικό αλγόριθμο σε συνδυασμό με την ασυνεχή ολοκλήρωση Choquet, επιτυγχάνεται ο έλεγχος ενός πάρα πολύ μικρού ποσοστού του χώρου αναζήτησης (searched space), της τάξεως του 1.49% στη χειρότερη περίπτωση, παράγοντας επιπλέον και τη βέλτιστη λύση (δηλαδή οι υπόλοιπες μέθοδοι είναι συνολικά υποδεέστερες ως προς το χώρο που έχει αναζητηθεί και ως προς τη βέλτιστη λύση). Ο χρόνος υπολογισμού (computational time- ο χρόνος που ξοδεύει ένας αλγόριθμος στις υπολογιστικές προσομοιώσεις) είναι μεγαλύτερος συγκριτικά με τις άλλες μεθόδους, λόγω τις ιδιαίτερα αυξημένης πολυπλοκότητας με την οποία επιλύεται το πρόβλημα, αλλά σε συνδυασμό με τα τελικά αποτελέσματα και την ποιότητα αυτών, ο λίγο παραπάνω υπολογιστικός χρόνος δεν αποτελεί αποτρεπτικό παράγοντα για την επιλογή του συγκεκριμένου αλγορίθμου.

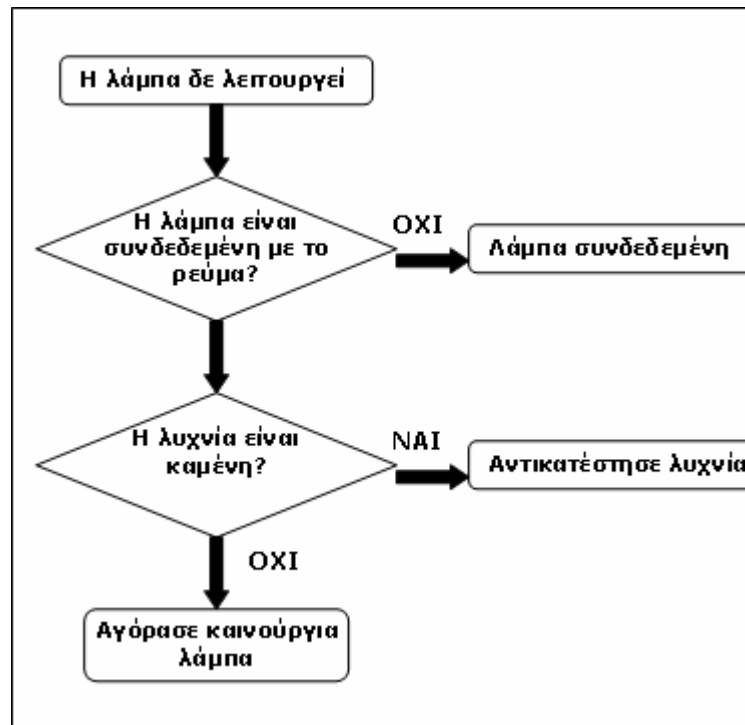
1. Προσδιορισμός Προβλήματος:

Είναι γνωστό ότι ένα από τα σημαντικότερα προβλήματα που αντιμετωπίζει μια γραμμή βιομηχανικής παραγωγής είναι το κόστος. Το κόστος είτε αυτό είναι σε χρόνο, είτε σε κόπωση, είτε σε αχρηστία, μεταφράζεται στη πραγματικότητα σε χρήμα. Και φυσικά στόχος μιας οποιαδήποτε βιομηχανίας, είναι η παραγωγή των προϊόντων της με το χαμηλότερο δυνατό χρηματικό κόστος. Στη προσπάθεια βελτιστοποίησης της γραμμής παραγωγής, θεωρούνται σα σημαντικότερα κριτήρια ελαχιστοποίησης: το κόστος κατασκευής, το κόστος των συνολικών καθυστερήσεων και το κόστος του συνολικού χρόνου ροής. Σκοπός της εργασίας είναι να αποδειχθεί πως μπορεί με δεδομένο τρόπο (MOGA σε συνδυασμό με Choquet integral) να επιτευχθεί η ελάχιστη τιμή στο κόστος.

2. Αλγόριθμοι και Γενετικοί Αλγόριθμοι:

2.1 Τι είναι οι Αλγόριθμοι:

Στα μαθηματικά και στην αυτοματοποίηση, αλγόριθμος είναι μια διαδικασία (ένα πεπερασμένο σετ με πολύ- καλά καθορισμένες εντολές) για την εκτέλεση κάποιας εργασίας, η οποία, δεδομένης της αρχικής κατάστασης, θα τερματίσει σε μια καθορισμένη κατάσταση- τερματισμού. Άτυπα, η γενική ιδέα του αλγορίθμου επεξηγείται με το παράδειγμα της συνταγής, αν και οι περισσότεροι είναι αρκετά πιο πολύπλοκοι από αυτό. Οι αλγόριθμοι συχνά έχουν βήματα, τα οποία επαναλαμβάνονται, ή χρειάζονται αποφάσεις, όπως λογική ή σύγκριση. Οι αλγόριθμοι μπορούν να συντεθούν ώστε να δημιουργήσουν πιο σύνθετους αλγορίθμους.



Εικόνα 1. Τα διαγράμματα ροής χρησιμοποιούνται συχνά ως γραφική αναπαράσταση των αλγορίθμων.

Ο ψευδοκώδικας και τα διαγράμματα ροής είναι δομημένοι τρόποι έκφρασης των αλγορίθμων, που αποφεύγουν τις ασάφειες που συμβαίνουν συχνά στις δηλώσεις της φυσικής γλώσσας, ενώ παραμένουν ανεξάρτητοι από συγκεκριμένες εφαρμογές της γλώσσας. Οι γλώσσες προγραμματισμού έχουν στόχο την έκφραση των αλγορίθμων σε τέτοια μορφή, που να μπορούν να εκτελούνται από τον υπολογιστή, αλλά συχνά χρησιμοποιούνται και ως τρόποι καθορισμού ή τεκμηρίωσης των αλγορίθμων.

Οι αλγόριθμοι εφαρμόζονται ή προσομοιώνονται από προγράμματα υπολογιστή (λένε στον υπολογιστή πια ακριβώς βήματα να ακολουθήσει στην εκτέλεση), ώστε να φέρουν εις πέρας μια συγκεκριμένη εργασία- ακολουθία διαδικασιών. Επειδή ο αλγόριθμος είναι μια ακριβής λίστα από ακριβή βήματα, η σειρά των υπολογισμών είναι πάντα πολύ κρίσιμη για τη λειτουργικότητα του αλγορίθμου. Ρητές οδηγίες θεωρείται πάντα ότι δίδονται σε λίστα, και περιγράφονται, αρχίζοντας «από τη κορυφή» και πηγαίνοντας «προς τα κάτω», σαν μια ιδέα που τυπικά ονομάζεται *έλεγχος ροής (flow of control)*.

Πολλά υπολογιστικά προβλήματα απαιτούν αναζήτηση ανάμεσα σε ένα τεράστιο αριθμό πιθανών λύσεων. Τέτοια προβλήματα μπορούν συχνά να επωφεληθούν από μια αποτελεσματική χρήση του παραλληλισμού, κατά τον οποίο πολλές διαφορετικές πιθανότητες εξερευνώνται ταυτόχρονα με αποτελεσματικό τρόπο. Αυτό που χρειάζεται είναι υπολογιστικός παραλληλισμός και έξυπνη στρατηγική για την επιλογή του επόμενου σετ αλληλουχίας προς αξιολόγηση.

Αλγόριθμος = Διαδικασία Απόφασης = Μέθοδος Απόφασης

2.2 Τι είναι οι Γενετικοί Αλγόριθμοι:

Οι Γενετικοί Αλγόριθμοι (GAs) είναι αλγόριθμοι αναζήτησης (search algorithms) που βασίζονται στους μηχανισμούς της φυσικής επιλογής και της γενετικής (θεωρία εξέλιξης). Συνδυάζουν την επιβίωση του δυνατότερου/ καλύτερου ανάμεσα στις δομές των συμβολοσειρών, με μια δομημένη αλλά και τυχαία ανταλλαγή πληροφοριών, για τη συγκρότηση ενός αλγορίθμου αναζήτησης με μια καινοτόμα κλίση στην ανθρώπινη αναζήτηση. Σε κάθε γενιά, ένα νέο σετ από τεχνητά δημιουργήματα (συμβολοσειρές- strings) κατασκευάζεται, χρησιμοποιώντας bits και κομμάτια από τη παλιά καλύτερη συμβολοσειρά.

Οι γενετικοί αλγόριθμοι, έχει θεωρητικά και εμπειρικά αποδειχθεί, ότι παρέχουν εύρωστη/ αυτοδύναμη αναζήτηση σε περίπλοκα διαστήματα. Είναι υπολογιστικά απλοί, παρ' όλα αυτά ισχυροί στην αναζήτησή τους για βελτίωση. Ακόμα, δεν είναι πλήρως περιορισμένοι από αυστηρές θεωρήσεις για το διάστημα αναζήτησης (θεωρήσεις που αφορούν τη συνέχεια, την ύπαρξη των παραγώγων, τη μονοκορυφότητα και άλλα ζητήματα).

Είναι υποσχόμενες μέθοδοι, για την επίλυση δύσκολων τεχνικών προβλημάτων (όπως προβλήματα βελτιστοποίησης) και για την εκμάθηση μηχανών. Γενικότερα, οι γενετικοί αλγόριθμοι είναι μέρος μιας καινούργιας «κίνησης» στην επιστήμη των υπολογιστών, που εξερευνά βιολογικά εμπνευσμένες υπολογιστικές προσεγγίσεις. Υποστηρικτές αυτής της κίνησης πιστεύουν ότι για να δημιουργηθούν τα είδη των υπολογιστικών συστημάτων που χρειαζόμαστε- συστήματα τα οποία

είναι προσαρμόσιμα, πολύ δυνατά, ικανά να αντιμετωπίσουν κάθε πολυπλοκότητα, ικανά να εκμάθουν, ακόμα και να δημιουργήσουν- πρέπει να αντιγραφούν τα φυσικά συστήματα με αυτά τα χαρακτηριστικά. Η φυσική εξέλιξη είναι μια ιδιαιτέρως ελκυστική πηγή έμπνευσης.

Οι γενετικοί αλγόριθμοι είναι επίσης υποσχόμενες προσεγγίσεις για τη μοντελοποίηση των φυσικών συστημάτων, που έχουν εμπνεύσει το σχέδιο τους. Τα περισσότερα μοντέλα που χρησιμοποιούν γενετικούς αλγόριθμους είναι ιδεατά μοντέλα, παρά ακριβής απομιμήσεις που προσπαθούν να είναι ισάξιες με τα δεδομένα του πραγματικού κόσμου. Σκοπός αυτών των ιδεατών μοντέλων είναι να κάνουν τις ιδέες ακριβής και να επιβεβαιώσουν την αληθοφάνειά τους, εφαρμόζοντας αυτές ως προγράμματα υπολογιστή, ώστε να καταλάβουν και να προβλέψουν τις γενικές τάσεις των φυσικών συστημάτων, και να δουν πως αυτές οι τάσεις επηρεάζονται από τις λεπτομέρειες στις αλλαγές του συστήματος. Αυτά τα μοντέλα επιτρέπουν στους επιστήμονες να εκτελέσουν προγράμματα που δεν θα μπορούσαν να είναι εφικτά στο πραγματικό κόσμο, και να προσομοιώσουν φαινόμενα που είναι δύσκολο ή αδύνατο να συλληφθούν και να αναλυθούν στο σετ των εξισώσεων. Αυτές οι μέθοδοι έχουν επίσης μια ευρέως ανεξερεύνητη και πιθανώς πολύ ενδιαφέρουσα πλευρά, που μέχρι τώρα δεν έχει αναφερθεί: την ακριβή μοντελοποίηση της εξέλιξης σαν πρόγραμμα υπολογιστή και υπολογιστική διαδικασία. Για παράδειγμα, είναι δυνατό να υπολογισθούν οι «πληροφορίες» που εμπεριέχονται σε ένα πληθυσμό και να γίνει προσπάθεια κατανόησης, του πως η εξέλιξη διαχειρίζεται αυτές τις πληροφορίες, ώστε να δημιουργηθούν δομές που οδηγούν σε μεγαλύτερη καταλληλότητα. Τέτοια υπολογιστική άποψη, που έγινε πραγματικότητα από τα υπολογιστικά μοντέλα τύπου γενετικών αλγορίθμων, θα γίνει τελικά ουσιώδες κομμάτι της κατανόησης των σχέσεων ανάμεσα στην εξέλιξη, τη θεωρία πληροφοριών και τη δημιουργία και προσαρμοστικότητα των οργανισμών στα βιολογικά συστήματα.

Πολλά προβλήματα βελτιστοποίησης από το κόσμο της βιομηχανίας, και κυρίως των κατασκευαστικών συστημάτων, είναι πολύ πολύπλοκα και είναι πολύ δύσκολο να λυθούν με τις συμβατικές τεχνικές βελτιστοποίησης. Από το 1960 υπάρχει αυτή η αύξηση ενδιαφέροντος στη μίμηση των ζωντανών όντων, στο να λύσουν αυτού του είδους δύσκολα προβλήματα βελτιστοποίησης. Προσομοιώνοντας τη φυσική διαδικασία εξέλιξης των ανθρώπινων όντων, έχει ως αποτέλεσμα τις

τεχνικές στοχαστικής βελτιστοποίησης που ονομάζονται *εξελικτικοί αλγόριθμοι*, που μπορεί συχνά να παρουσιαστούν ως συμβατικές μέθοδοι βελτιστοποίησης, όταν εφαρμόζονται σε δύσκολα, πραγματικού κόσμου προβλήματα. Υπάρχουν (3) κύριοι «οδοί» αναζήτησης: *οι γενετικοί αλγόριθμοι (GAs)*, *ο εξελικτικός προγραμματισμός (EP)*, και *οι εξελικτικές στρατηγικές (ESs)*. Ανάμεσα σε αυτά, οι γενετικοί αλγόριθμοι είναι ο πιο ευρέως γνωστός τύπος αλγορίθμου σήμερα.

Οι γενετικοί αλγόριθμοι βασίζονται στις αρχές του Δαρβίνου όσο αναφορά τις διαδικασίες εξέλιξης των όντων. Παρακάτω παρουσιάζονται κάποιοι ορισμοί πάνω στους οποίους «χτίζονται» οι γενετικοί αλγόριθμοι.

Γενετική Αναπαράσταση:

Η Γενετική Αναπαράσταση είναι ένας τρόπος αναπαράστασης λύσεων/ατόμων στις εξελικτικές υπολογιστικές μεθόδους. Οι γενετικοί αλγόριθμοι χρησιμοποιούν γραμμικές δυαδικές αναπαραστάσεις. Η πιο συνηθισμένη αναπαράσταση είναι αυτή με το πίνακα από bits. Η κυριότερη ιδιότητα που κάνει αυτές τις γενετικές αναπαραστάσεις εύχρηστες είναι ότι τα τμήματα της ευθυγραμμίζονται εύκολα, λόγω του σταθερού μεγέθους τους.

Η Ορθή Συνάρτηση:

Είναι ένας συγκεκριμένος τύπος αντικειμενικής συνάρτησης που προσδιορίζει ποσοτικά την ορθότητα της λύσης (δηλαδή του χρωμοσώματος) σε ένα γενετικό αλγόριθμο, έτσι ώστε το συγκεκριμένο χρωμόσωμα να ταξινομηθεί καλύτερα σε σχέση με όλα τα άλλα χρωμοσώματα. Τα βέλτιστα χρωμοσώματα, ή τουλάχιστον τα χρωμοσώματα που είναι ορθότερα από άλλα, επιτρέπεται να αναπαραχθούν και να αναμειχθούν τα στοιχεία τους με διάφορες τεχνικές, παράγοντας μια νέα γενιά που θα είναι καλύτερη από τη προηγούμενη.

Επιλογή:

Επιλογή είναι μια κατάσταση των γενετικών αλγορίθμων, κατά την οποία μεμονωμένοι γόννοι επιλέγονται από ένα πληθυσμό για μετ' έπειτα αναπαραγωγή (διασταύρωση- ανταλλαγή γονιδίων).

Κανονικοποίηση:

Κανονικοποίηση είναι ο πολλαπλασιασμός της ορθότερης/ καταλληλότερης τιμής του κάθε «ατόμου» με ένα σταθερό αριθμό, ώστε το άθροισμα όλων των ορθότερων τιμών να είναι ίσο με τη μονάδα.

Φαινότυπος:

Ο φαινότυπος ενός μοναδικού οργανισμού περιγράφει ένα από τα γνωρίσματα ή χαρακτηριστικά, το οποίο είναι μετρήσιμο και εκφράζεται μόνο σε υποκατηγορία των ατόμων μέσα στο πληθυσμό. Κάποιοι φαινότυποι ελέγχονται εξ' ολοκλήρου από τα γονίδια των ατόμων. Άλλοι ελέγχονται από τα γονίδια αλλά είναι και σημαντικά επηρεασμένοι από μη- γενετικούς ή περιβαλλοντικούς παράγοντες. Παρ' όλα αυτά υπάρχουν ακόμα φαινότυποι οι οποίοι είναι εντελώς μη- γενετικοί.

Επειδή ακριβώς οι φαινότυποι καθορίζονται από πολλαπλά γονίδια και μπορούν επίσης να επηρεαστούν από περιβαλλοντικούς παράγοντες, η ταυτότητα ενός ή περισσότερων τμημάτων γονιδιακής αλληλουχίας ενός οργανισμού, δε καθιστά ικανούς τους επιστήμονες να ξέρουν τι να αναμένουν σαν φαινότυπο.

Φαινοτυπική Ποικιλία:

Η φαινοτυπική ποικιλία είναι θεμελιακή προϋπόθεση για την εξέλιξη μέσω της φυσικής επιλογής. Η ικανότητα ενός οργανισμού είναι ένας υψηλού επιπέδου φαινότυπος που καθορίζεται από τις συνεισφορές χιλιάδων ακόμα πιο εξειδικευμένων φαινοτύπων. Χωρίς τη φαινοτυπική ποικιλία, οι μεμονωμένοι οργανισμοί θα είχαν όλοι την ίδια ικανότητα και οι αλλαγές στη φαινοτυπική συχνότητα θα γινόντουσαν χωρίς την επιλογή (τυχαία).

Η αλληλεπίδραση μεταξύ γονότυπου και φαινότυπου σχηματίζεται με τη παρακάτω σχέση:

$$\text{Γονότυπος} + \text{Περιβάλλον} + \text{Τυχαία- Επιλογή} \rightarrow \text{Φαινότυπος}$$

Ένας φαινότυπος είναι ένα οποιοδήποτε εντοπίσιμο χαρακτηριστικό (δομικό, φυσιολογικό, βιοχημικό, κ.τ.λ.) ενός οργανισμού, καθορισμένο από την αλληλεπίδραση μεταξύ του γονοτύπου του και του περιβάλλοντος.

Γονίδια:

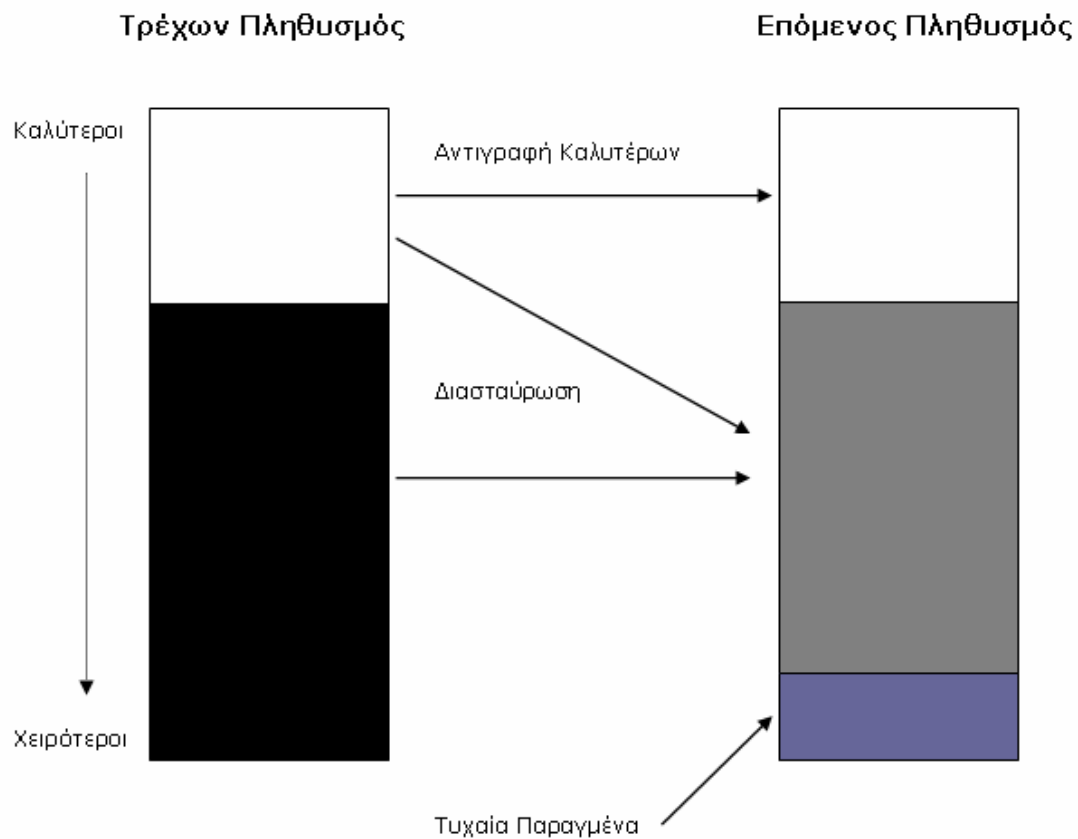
Τα γονίδια αλληλεπιδρούν μεταξύ τους και με το περιβάλλον για να καθορίσουν τη φυσική ανάπτυξη και συμπεριφορά των οργανισμών, είναι μονάδες κληρονομιάς. Τα γονίδια μεταφέρουν το γενετικό κώδικα, ο οποίος αποτελεί τις απαραίτητες πληροφορίες για τη παραγωγή και εξέλιξη του «οργανισμού».

Γονότυπος:

Ο γονότυπος είναι μια σύνθεση από κομμάτια από μοναδικά χαρακτηριστικά, τα οποία κληρονομούνται και τα οποία συνεισφέρουν στο καθορισμό συγκεκριμένων γνωρισμάτων. Με προσεκτικό πειραματικό σχεδιασμό, μπορεί κάποιος να χρησιμοποιήσει στατιστικές μεθόδους ώστε να συσχετίσει διαφορές στους γονότυπους του πληθυσμού με διαφορές στους παρατηρούμενους φαινοτύπους. Εμπνευσμένη από τη γενική εικόνα της βιολογίας και τη χρησιμότητα των γονότυπων, η επιστήμη των υπολογιστών θέτει σε εφαρμογή προσομοιωμένους γονότυπους στο γενετικό προγραμματισμό και στους εξελικτικούς αλγόριθμους. Τέτοιες τεχνικές μπορούν να βοηθήσουν στην εξέλιξη μαθηματικών λύσεων σε συγκεκριμένα NP- hard προβλήματα.

Αναπαράσταση Υποψηφίων Λύσεων:

Θεωρείται ότι μια πιθανή λύση σε ένα πρόβλημα μπορεί να αναπαρασταθεί ως ένα σετ από παραμέτρους. Αυτές οι παράμετροι (γνωστοί ως γονίδια) ενώνονται μεταξύ τους ώστε να σχηματίσουν μια συμβολοσειρά/ αλληλουχία (string) από τιμές (χρωμόσωμα). Στη γενετική ορολογία, το σετ των παραμέτρων που αναπαρίσταται από συγκεκριμένα χρωμοσώματα, αναφέρεται σαν *άτομο* (individual). Η καταλληλότητα ενός ατόμου εξαρτάται από το χρωμόσωμά του και αξιολογείται από τη συνάρτηση καταλληλότητάς του.



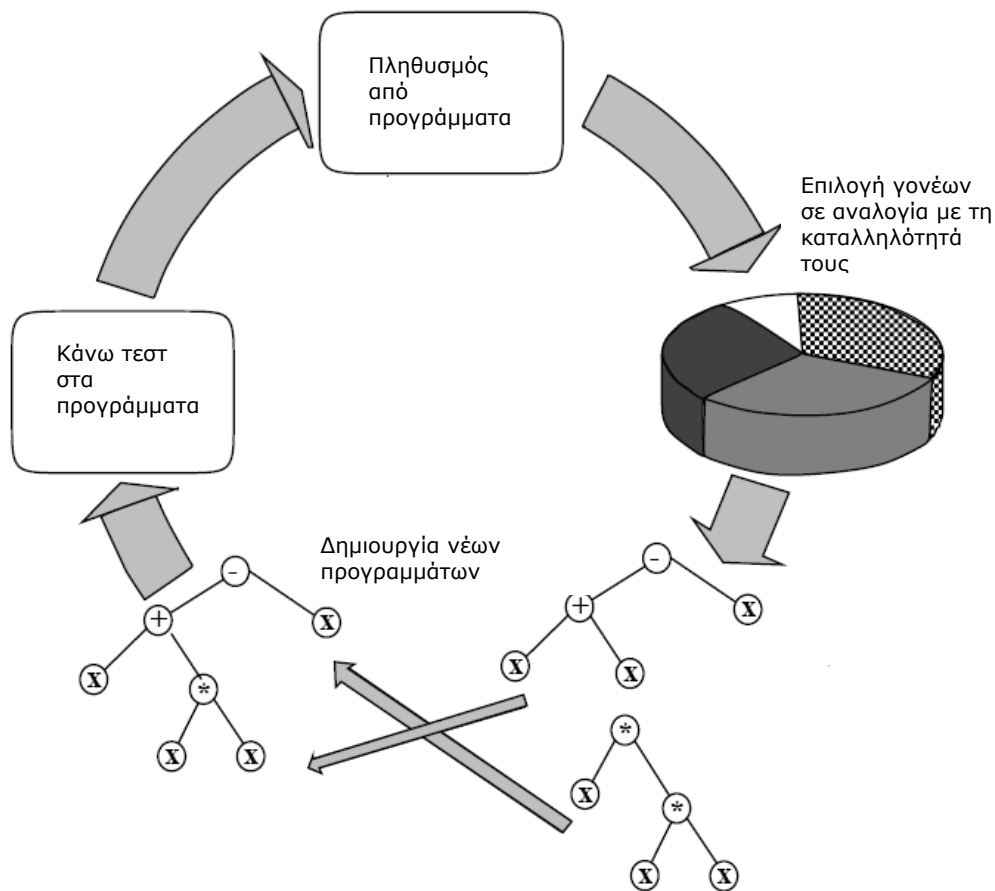
Εικόνα 2. Μεταβατική διαδικασία μεταξύ διαδοχικών γενεών.

2.3 Γενετικός Προγραμματισμός:

Ο γενετικός προγραμματισμός είναι μια πατενταρισμένη αυτοματοποιημένη μεθοδολογία, εμπνευσμένη από τη βιολογική εξέλιξη, ώστε να βρίσκει υπολογιστικά προγράμματα που εκτελούν καλύτερα μια καθορισμένη από το χρήστη εργασία (αυτόματη κατασκευή προγραμμάτων υπολογιστών). Γι' αυτό το λόγο είναι μια συγκεκριμένη τεχνική εκμάθησης μηχανής, που χρησιμοποιεί έναν εξελικτικό αλγόριθμο ώστε να βελτιστοποιήσει ένα πληθυσμό από υπολογιστικά προγράμματα, σύμφωνα με ένα «τοπίο καταλληλότητας» καθορισμένο από την ικανότητα του προγράμματος να εκτελεί μια υπολογιστική εργασία.

Στη τεχνητή εξέλιξη (artificial evolution) οι περισσότεροι υπολογιστικοί πόροι, είναι αφοσιωμένοι στο να αποφασίσουν ποια άτομα θα κάνουν παιδιά. Αυτό

συνήθως απαιτεί την ανάθεση μιας τιμής καταλληλότητας σε κάθε καινούργιο άτομο. Στο γενετικό προγραμματισμό τα άτομα του πληθυσμού είναι υπολογιστικά προγράμματα. Η καταλληλότητά τους συνήθως υπολογίζεται, τρέχοντάς τα μια ή περισσότερες φορές με μια ποικιλία από εισόδους (γνωστοί ως test set), και βλέποντας πόσο κοντά είναι οι έξοδοι του προγράμματος, σε κάποια επιθυμητή έξοδο, την οποία έχει καθορίσει ο χρήστης.



Εικόνα 3. Ο κύκλος του γενετικού προγραμματισμού.

Ο κύκλος του γενετικού προγραμματισμού είναι όπως κάθε άλλη εξελικτική διαδικασία. Καινούργια άτομα (στη περίπτωση του ΓΠ καινούργια προγράμματα) δημιουργούνται. Υπόκεινται σε τεστ. Τα ισχυρότερα στο πληθυσμό επιτυγχάνουν στο να δημιουργήσουν παιδιά. Τα ανίσχυρα πεθαίνουν και απομακρύνονται από το πληθυσμό.

2.4 Αξιοποίηση και διερεύνηση:

Η αναζήτηση είναι μια από τις καθολικές μεθόδους λύσης- προβλήματος για τέτοια προβλήματα, όπου ένας δε μπορεί να καθορίσει εκ των προτέρων την ακολουθία των βημάτων που ακολουθούν στη λύση. Η αναζήτηση μπορεί να εκτελεστεί είτε με *τυφλές στρατηγικές (blind strategies)* είτε με *ευρετικές στρατηγικές (heuristic strategies)*. Οι τυφλές στρατηγικές αναζήτησης δε χρησιμοποιούν πληροφορίες για τη περιοχή του προβλήματος. Οι ευρετικές στρατηγικές αναζήτησης χρησιμοποιούν επιπρόσθετες πληροφορίες ώστε να οδηγήσουν την αναζήτηση σύμφωνα με τις καλύτερες κατευθύνσεις αναζήτησης. Υπάρχουν (2) σημαντικά θέματα στις στρατηγικές αναζήτησης: αξιοποιούμε τη καλύτερη λύση και διερευνούμε το χώρο αναζήτησης. Ο Michalewicz (1996) σύγκρινε τη hill- climbing αναζήτηση, τη τυχαία αναζήτηση και τη γενετική αναζήτηση. Το hill- climbing είναι ένα παράδειγμα στρατηγικής, κατά την οποία αξιοποιείται η καλύτερη λύση για πιθανή βελτίωση ενώ ταυτόχρονα αγνοείται η διερεύνηση του χώρου αναζήτησης. Η τυχαία αναζήτηση είναι παράδειγμα στρατηγικής, το οποίο διερευνά το χώρο αναζήτησης ενώ ταυτόχρονα αγνοεί την αξιοποίηση των πολλά- υποσχόμενων περιοχών του χώρου αναζήτησης. Ενώ οι γενετικοί αλγόριθμοι είναι μια κατηγορία μεθόδων αναζήτησης γενικού- αντικειμένου, συνδυάζοντας στοιχεία από κατευθυνόμενη και στοχαστική αναζήτηση, τα οποία μπορούν να δημιουργήσουν μια αξιοθαύμαστη ισορροπία μεταξύ της διερεύνησης και της αξιοποίησης του χώρου αναζήτησης.

2.5 Βελτιστοποίηση:

Η βελτιστοποίηση παίζει ένα κεντρικό ρόλο στις λειτουργίες επιστημονικής έρευνας/ διαχείρισης και στη σχεδιαστική μηχανική των προβλημάτων. Η βελτιστοποίηση σχετίζεται με προβλήματα ελαχιστοποίησης ή μεγιστοποίησης μιας συνάρτησης με πολλές διαφορετικές μεταβλητές, που υπόκειται συνήθως σε ισότητα ή/ και ανισότητα των περιορισμών. Οι τεχνικές βελτιστοποίησης έχουν μια αυξανόμενα μεγάλη επιρροή στη κοινωνία μας. Ο αριθμός και η ποικιλία των εφαρμογών τους συνεχίζουν να αυξάνονται ραγδαία, και μία καθυστέρηση δε φαίνεται να υπάρχει.

Παρ' όλα αυτά, πολλά από τα προβλήματα της σχεδιαστικής μηχανικής είναι πολύ δύσκολα στη φύση τους και δύσκολα να επιλυθούν με συμβατικές τεχνικές βελτιστοποίησης. Τα τελευταία χρόνια οι γενετικοί αλγόριθμοι έχουν γίνει δέκτες ιδιαίτερης προσοχής, εξαιτίας των δυνατοτήτων τους ως καινοφανής τεχνικές βελτιστοποίησης.

Στόχος της βελτιστοποίησης:

Με τη βελτιστοποίηση αναζητούμε τη βελτίωση της εκτέλεσης αποσκοπώντας σε κάποιο βέλτιστο σημείο ή σημεία. Η σημασία αυτού χωρίζεται σε δυο τμήματα: (1) αναζητούμε τη βελτίωση για να επιτύχουμε κάποιο/α (2) βέλτιστο/α σημείο/α. Υπάρχει σαφής διαχωρισμός μεταξύ της *διαδικασίας*(*process*) της βελτιστοποίησης και του *προορισμού*(*destination*) ή βέλτιστου σημείου καθ' αυτού.

2.6 Η τυχαία αναζήτηση στους αλγορίθμους:

Οι γενετικοί αλγόριθμοι είναι ένα παράδειγμα μιας διαδικασίας αναζήτησης που χρησιμοποιεί τη τυχαία επιλογή σαν ένα εργαλείο για να καθοδηγήσει μια πολύ αξιοποιήσιμη αναζήτηση, μέσα σ' ένα κωδικοποιημένο διάστημα παραμέτρων. Η χρήση της τυχαίας αναζήτησης φαίνεται κατά τη διάρκεια εφαρμογής του *μηχανισμού αναζήτησης*, ο οποίος αναλύεται παρακάτω, και κατά τον οποίο εφαρμόζεται η *μέθοδος της ρουλέτας*, της οποίας χαρακτηριστικό στοιχείο είναι η τυχειότητα. Μια άλλη τρέχουσα διάσημη τεχνική αναζήτησης, η προσομοιωμένη απόπτηση (*simulated annealing*), χρησιμοποιεί τις τεχνικές της τυχαίας επιλογής ως βοήθεια για να καθοδηγήσει το είδος της αναζήτησης στη κατάσταση της χαμηλότερης ενέργειας. Είναι όμως βασικό να τονίσουμε ότι η τυχειότητα στην αναζήτηση δε σημαίνει απαραίτητα αναζήτηση χωρίς καθοδήγηση.

2.7 Που διαφέρουν οι Γενετικοί Αλγόριθμοι από τις παραδοσιακές μεθόδους:

Οι γενετικοί αλγόριθμοι διαφέρουν από τις περισσότερες συμβατικές διαδικασίες βελτιστοποίησης και αναζήτησης σε (4) πολύ βασικούς τομείς. Ο Goldberg (1989) τους έχει συνοψίσει ως εξής:

1. Λειτουργούν με τη κωδικοποίηση του σετ των παραμέτρων και των λύσεων και όχι τις ίδιες τις παραμέτρους ή τις λύσεις καθ' αυτού.
2. Ψάχνουν σε ένα πληθυσμό από σημεία, και όχι σε ένα μόνο σημείο.
3. Χρησιμοποιούν «ξεκάθαρες» πληροφορίες - payoff information (από τη συνάρτηση καταλληλότητας), όχι παράγωγες/ δευτερεύοντες ή άλλες βοηθητικές γνώσεις.
4. Χρησιμοποιούν προαμπιλιστικούς/ πιθανολογικούς κανόνες μετάβασης, και όχι ντετερμινιστικούς/ αιτιοκρατικούς κανόνες.

Οι γενετικοί αλγόριθμοι απαιτούν το σετ των φυσικών παραμέτρων του προβλήματος βελτιστοποίησης να είναι κωδικοποιημένο σαν μια πεπερασμένου μήκους συμβολοσειρά, πάνω σε μια καθορισμένη αλφάβητο.

Σε πολλές μεθόδους βελτιστοποίησης, η κίνηση γίνεται προσεκτικά από το ένα σημείο του χώρου αναζήτησης στο επόμενο, χρησιμοποιώντας κάποιους κανόνες μετάβασης, για να αποφασίσουμε το επόμενο σημείο. Αυτή η μέθοδος σημείο-προς-σημείο είναι επικίνδυνη, γιατί είναι μια τέλεια συνταγή για την εύρεση των λάθος κορυφών- μέγιστων τιμών (peaks) σε πολυτροπικά (με πολλές κορυφές) διαστήματα αναζήτησης. Σε αντίθεση με αυτό, οι γενετικοί αλγόριθμοι δουλεύουν με μια πλούσια βάση δεδομένων από σημεία ταυτόχρονα (ένας πληθυσμός από συμβολοσειρές), σκαρφαλώνοντας πολλές κορυφές ταυτόχρονα, έτσι η πιθανότητα του να βρεθεί μια λάθος κορυφή είναι μειωμένη σε σχέση με τις μεθόδους που πάνε σημείο-προς-σημείο. Το χαρακτηριστικό του παραλληλισμού συνεισφέρει στην ευρωστία του γενετικού αλγορίθμου.

Οι γενετικοί αλγόριθμοι δεν έχουν ανάγκη από βοηθητικές πληροφορίες, είναι τυφλοί σε αυτές, χρειάζονται μόνο payoff τιμές (τιμές υποκειμενικής συνάρτησης) σχετικές με μεμονωμένες συμβολοσειρές.

Επίσης, χρησιμοποιούν πιθανολογικούς κανόνες μετάβασης για να καθοδηγήσουν την αναζήτησή τους. Η χρήση της πιθανότητας δε σημαίνει ότι η μέθοδος είναι κάποια απλή τυχαία αναζήτηση. Οι γενετικοί αλγόριθμοι χρησιμοποιούν τη τυχαία επιλογή σαν εργαλείο, ώστε να καθοδηγήσουν την αναζήτηση προς περιοχές του διαστήματος αναζήτησης με πιθανότητα βελτίωσης.

Οι (4) αυτές διαφορές – απ'ευθείας χρήση της κωδικοποίησης, αναζήτηση μέσα από πληθυσμό, «τύφλωση» (αγνόηση) στις βοηθητικές πληροφορίες, τελεστές τυχειότητας- συνεισφέρουν στην ευρωστία των γενετικών αλγορίθμων και αποτελούν προνόμιο έναντι σε άλλες κοινές τεχνικές.

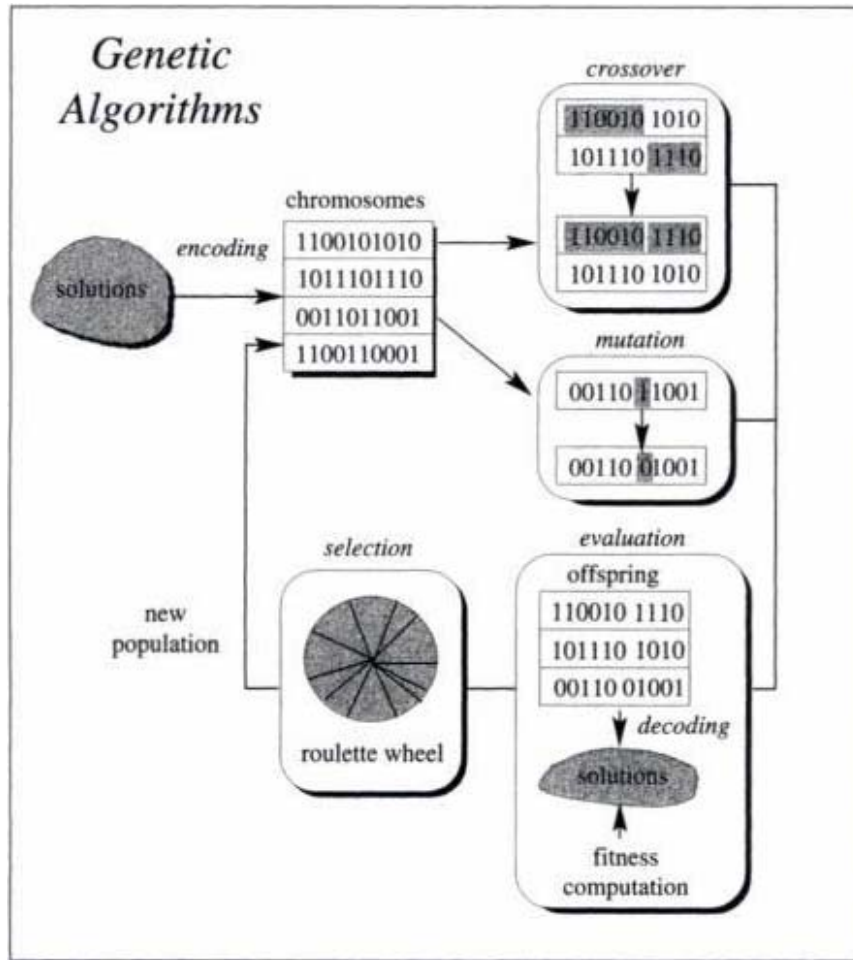
3. Οι θεωρήσεις των Γενετικών Αλγορίθμων (ΓΑς):

Η συνήθης μορφή του γενετικού αλγορίθμου έχει περιγραφεί από τον Goldberg. Οι γενετικοί αλγόριθμοι είναι στοχαστικές τεχνικές αναζήτησης βασισμένες σε μηχανισμούς της φυσικής επιλογής και της εξέλιξης. Οι γενετικοί αλγόριθμοι, διαφέροντας από τις συμβατικές τεχνικές αναζήτησης, ξεκινούν με ένα αρχικό σετ από τυχαίες/ υποψήφιες λύσεις που ονομάζεται *πληθυσμός*. Κάθε άτομο στο πληθυσμό ονομάζεται *χρωμόσωμα*, αναπαριστώντας μια λύση στο πρόβλημα. Ένα χρωμόσωμα είναι ένα *string* από σύμβολα και είναι συνήθως, αλλά όχι απαραίτητα, μια συμβολοσειρά από δυαδικά bit. Τα χρωμοσώματα εξελίσσονται μέσω επιτυχημένων επαναλήψεων, που ονομάζονται *γενεές*. Κατά τη διάρκεια κάθε γενιάς, τα χρωμοσώματα *αξιολογούνται*, χρησιμοποιώντας το μέτρο της καταλληλότητας (*fitness*). Για να δημιουργηθεί η επόμενη γενιά, καινούργια χρωμοσώματα, που ονομάζονται *παιδιά*, δημιουργούνται α) από τη συγχώνευση (2) χρωμοσωμάτων από τη τρέχουσα γενιά χρησιμοποιώντας τη λειτουργία της διασταύρωσης και β) τροποποιώντας το χρωμόσωμα χρησιμοποιώντας το χειριστή της μετάλλαξης. Μια καινούργια γενιά σχηματίζεται α) επιλέγοντας, σύμφωνα με τις τιμές της καταλληλότητας, κάποιους από τους γονείς και τα παιδιά και β) απορρίπτοντας άλλους, έτσι ώστε να διατηρηθεί το μέγεθος του πληθυσμού σταθερό. Μετά από μερικές γενιές, ο αλγόριθμος συγκλίνει στο καλύτερο χρωμόσωμα, το οποίο με μεγάλες πιθανότητες επιτυχίας αντιπροσωπεύει τη βέλτιστη ή την υποβέλτιστη λύση του προβλήματος.

Συνήθως, η αρχικοποίηση θεωρείται ότι γίνεται τυχαία. Υπάρχουν μόνο (2) είδη τελεστών στους γενετικούς αλγορίθμους:

- i. γενετικοί τελεστές: διασταύρωση (crossover) και μετάλλαξη (mutation),
- ii. εξελικτικοί τελεστές: επιλογή (selection).

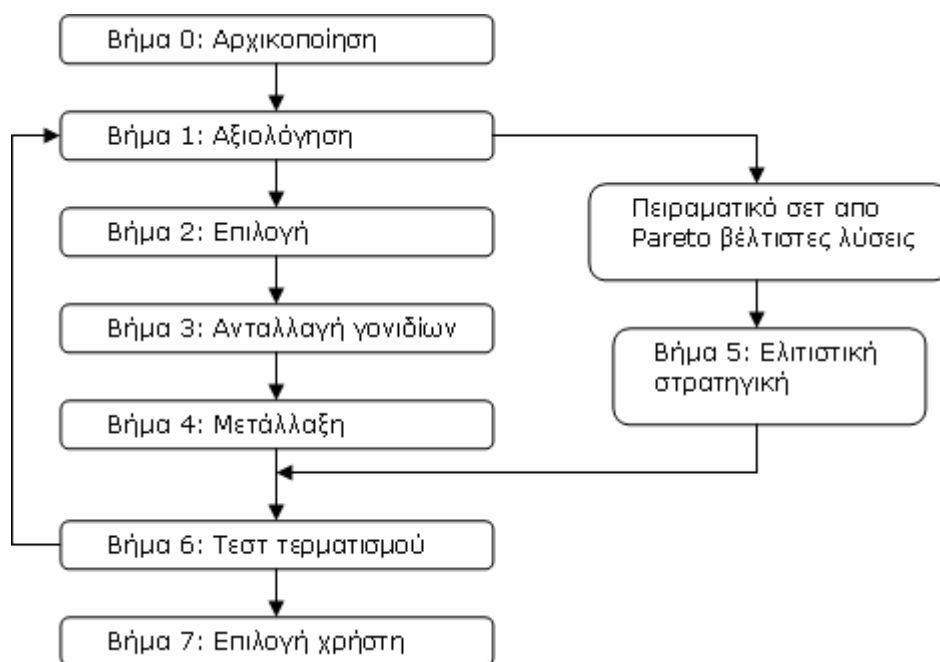
Οι γενετικοί τελεστές μιμούνται τις διαδικασίες κληρονομικότητας των γονιδίων για να δημιουργήσουν παιδιά σε κάθε γενιά. Οι εξελικτικές διαδικασίες μιμούνται τις διαδικασίες εξέλιξης του *Δαρβίνου* για να δημιουργήσουν πληθυσμούς από γενιά σε γενιά.



Εικόνα 4. Η γενική δομή των γενετικών αλγορίθμων.

3.1 Βασικός μηχανισμός λειτουργίας του ΓΑ:

- 1) Δημιουργία (συνήθως τυχαία) ενός πληθυσμού ατόμων που αντιπροσωπεύουν πιθανές λύσεις στο φυσικό πρόβλημα.
- 2) Υπολογισμός της ποιότητας του κάθε ατόμου του πληθυσμού, υπολογίζοντας την συνάρτηση καταλληλότητας (fitness function) του κάθε ατόμου.
- 3) Προβιβασμός ατόμων υψηλής ποιότητας, εισάγοντας πίεση επιλογής σε ολόκληρο το πληθυσμό.
- 4) Δημιουργία νέων ατόμων εφαρμόζοντας ποικίλους τελεστές, όπως ανταλλαγή γονιδίων (crossover) και μετάλλαξη (mutation) στο πληθυσμό.
- 5) Επανάληψη των βημάτων 2-4 αρκετές φορές, μέχρι την ικανοποίηση ανάλογων κριτηρίων τερματισμού.



Εικόνα 5. Γενικό διάγραμμα ροής γενετικών αλγορίθμων.

Οι μεγαλύτεροι παράγοντες που μπορούν να επηρεάσουν τα παραπάνω (5) βήματα είναι:

Αναπαραστατικός Μηχανισμός: ένας τρόπος κωδικοποίησης των λύσεων του προβλήματος σε «τεχνητά» χρωμοσώματα.

Μηχανισμός Αξιολόγησης: ο υπολογισμός της συνάρτησης καταλληλότητας.

Γενετικοί Τελεστές: ποιοι τύποι τελεστών για ανταλλαγή γονιδίων και μετάλλαξη να χρησιμοποιηθούν, πως να επιλεχθούν οι γονείς των χρωμοσωμάτων.

Οι απαραίτητες ρυθμίσεις για τις παραμέτρους ελέγχου: μέγεθος πληθυσμού, ρυθμοί/ποσοστά διασταύρωσης και μετάλλαξης, αριθμός γενιών.

3.2 Εφαρμογή στον Αλγόριθμο:

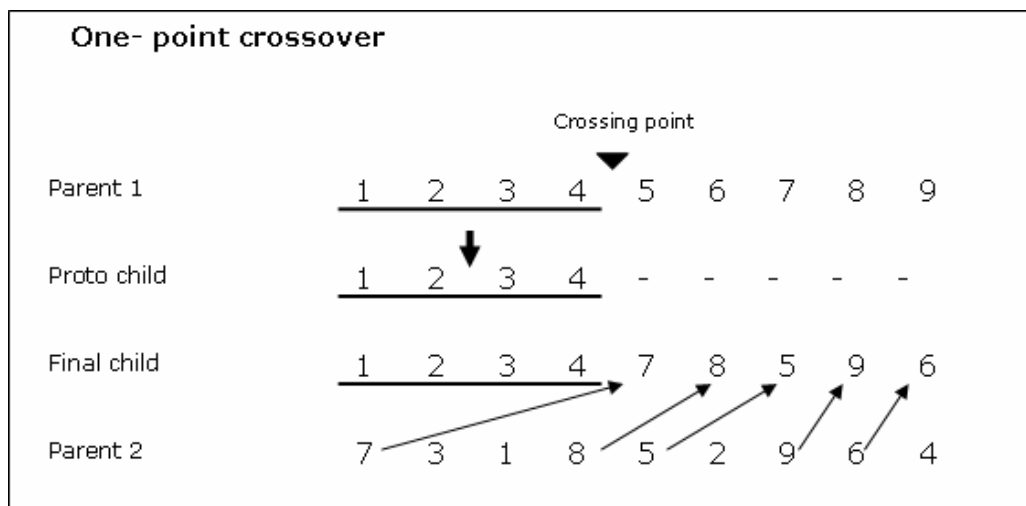
Αναπαραστατικός Μηχανισμός: Χρησιμοποιούνται μεταθετικές συμβολοσειρές, όπως strings από ακέραιους, με κάθε συμβολοσειρά να αντιστοιχεί σε μια μοναδική προγραμματιστική λύση.

Αρχικός Πληθυσμός: Θεωρούμε ένα πληθυσμό από N_p χρωμοσώματα, τα N_p-1 χρωμοσώματα δημιουργήθηκαν τυχαία, ενώ ένα μοναδικό χρωμόσωμα δημιουργήθηκε χρησιμοποιώντας τη NEH δομημένη ευρετική μέθοδο.

Μηχανισμός Αξιολόγησης: Η συνάρτηση καταλληλότητας αντανakλά την ικανότητα ενός ατόμου να επιβιώσει και να αναπαράγει τη δομή του στο πληθυσμό της επόμενης γενιάς. Δέχεται ως είσοδο ένα χρωμόσωμα και επιστρέφει έναν αριθμό, που υποδηλώνει το πόσο κατάλληλο είναι. Η συνάρτηση καταλληλότητας ισοδυναμεί στην υποκειμενική συνάρτηση, όταν αυτή βελτιστοποιηθεί. Σκοπός είναι η μεγιστοποίηση της υποκειμενικής συνάρτησης f . Ως εκ τούτου, η συνάρτηση καταλληλότητας για κάθε χρωμόσωμα i ($i=1, 2, \dots, N_p$) καθορίζεται ως: $fitness_i = f_i$. Όσο υψηλότερη η καταλληλότερη τιμή του χρωμοσώματος, τόσο καλύτερα ποιοτικά αναπαρίσταται η λύση από το χρωμόσωμα.

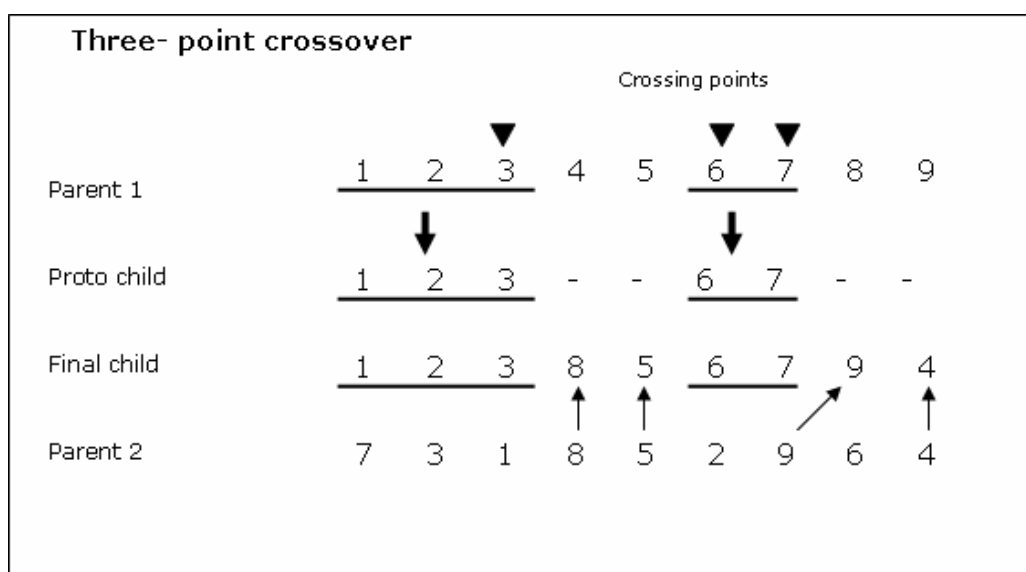
Γενετικοί Τελεστές: (3) γενετικοί τελεστές χρησιμοποιούνται: ένα ζεύγος από τελεστές διασταύρωσης (διασταύρωση ενός σημείου/ one- point, και τριών σημείων/ three- point) καθ' ένας από τους οποίους ακολουθείται από ένα μηχανισμό εξυγίανσης, έτσι ώστε να διασφαλιστεί η δημιουργία έγκυρων λύσεων, και ένας τελεστής μετάλλαξης (shift mutation). Η διασταύρωση είναι ένας κύριος γενετικός τελεστής. Λειτουργεί πάνω σε (2) χρωμοσώματα κάθε φορά και δημιουργεί παιδιά συνδυάζοντας χαρακτηριστικά και από τα (2) χρωμοσώματα.

- *Διασταύρωση ενός σημείου (one- point crossover):* ένα σημείο διασταύρωσης επιλέγεται τυχαία από όλο το μήκος του πρώτου χρωμοσώματος γονέα. Ο υποτομέας των εργασιών πριν από αυτό το σημείο διασταύρωσης αντιγράφεται στο παιδί. Τα υπόλοιπα τμήματα του παιδιού συμπληρώνονται παίρνοντας με τη σειρά κάθε γνήσιο γονίδιο από το 2^ο γονέα.



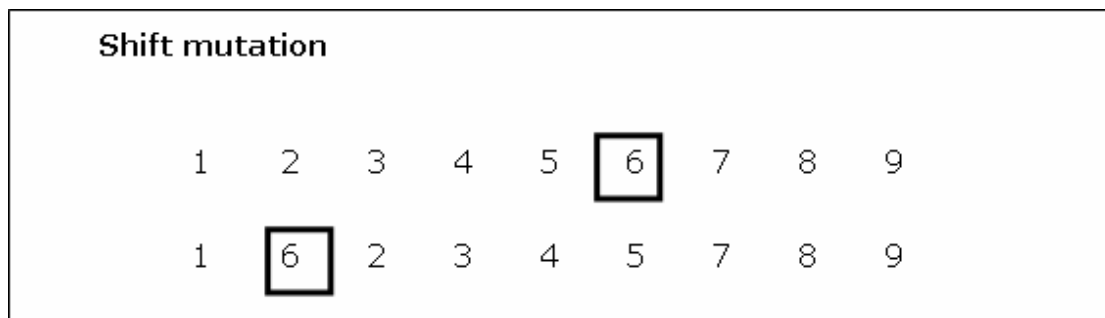
Εικόνα 6. Διασταύρωση ενός- σημείου (one- point crossover).

- *Διασταύρωση τριών σημείων (three- point crossover):* (3) σημεία διασταύρωσης επιλέγονται τυχαία από όλο το μήκος του 1^{ου} χρωμοσώματος γονέα. Το χρωμόσωμα χωρίζεται τότε σε (4) ξεχωριστά τμήματα. Αντιγράφεται ο 1^{ος} και ο 3^{ος} υποτομέας εργασιών στα ίδια σημεία του παιδιού. Συμπληρώνονται οι εναπομείναντες ελεύθερες θέσεις του παιδιού με εργασίες από το 2^ο γονέα σύμφωνα με τη σειρά εμφάνισης τους. Επίσης δεν επιτρέπεται η διπλογραφία των εργασιών.



Εικόνα 7. Διασταύρωση τριών- σημείων (three- point crossover).

- *Μετάλλαξη (Shift Mutation)*: μια μοναδική εργασία επιλέγεται τυχαία και τοποθετείται σε μια τυχαία θέση. Η μετάλλαξη είναι ένας «παρασκηνιακός» τελεστής, ο οποίος παράγει αυτογενώς τυχαίες αλλαγές σε ποικιλία χρωμοσωμάτων. Ένας τυχαίος τρόπος να επιτευχθεί αυτό, είναι η τροποποίηση ενός ή περισσότερων γονιδίων. Στους γενετικούς αλγορίθμους, η μετάλλαξη εξυπηρετεί το κρίσιμο ρόλο του να α) αντικαταστήσει τα γονίδια από κάθε πληθυσμό που χάθηκαν κατά τη διάρκεια της διαδικασίας της επιλογής, έτσι ώστε να «δοκιμαστούν» σε ένα καινούργιο γενικό πλαίσιο, ή να β) προμηθεύσει τα γονίδια που δεν ήταν παρόντα στον αρχικό πληθυσμό.



Εικόνα 8. Μετάλλαξη (mutation).

Μηχανισμός επιλογής: η επιλογή των γονεϊκών χρωμοσωμάτων γίνεται χρησιμοποιώντας τη τεχνική της ρουλέτας: όλες οι πιθανές λύσεις ή τα χρωμοσώματα προσδιορίζονται από μια καταλληλότητα που τους ανατίθεται από τη συνάρτηση καταλληλότητας. Στη τεχνική της ρουλέτας, αυτό το επίπεδο καταλληλότητας χρησιμοποιείται για να συσχετίσει μια πιθανότητα επιλογής με το κάθε ατομικό χρωμόσωμα. Ενώ οι υποψήφιας λύσεις με υψηλότερη καταλληλότητα θα είναι λιγότερο πιθανό να εξαλειφθούν, υπάρχει ακόμα η πιθανότητα αυτό όντως να γίνει. Σε αντίθεση με άλλους αλγόριθμους επιλογής, στη χρησιμοποιηθείσα μέθοδο, υπάρχει ακόμα η δυνατότητα κάποιες ασθενέστερες λύσεις να επιβιώσουν της διαδικασίας επιλογής. Αυτό είναι ένα πλεονέκτημα, γιατί παρ' όλο που η λύση είναι ασθενής, μπορεί να εμπεριέχει κάποιες συνισταμένες, οι οποίες μπορεί να φανούν χρήσιμες ακολουθώντας τη διαδικασία του επανασυνδυασμού. Η αναλογία με τη ρόδα

μιας ρουλέτας μπορεί να οραματισθεί, θεωρώντας μια ρόδα ρουλέτας, στην οποία κάθε υποψήφια λύση αναπαριστά ένα κομμάτι στη ρουλέτα. Το μέγεθος του κομματιού είναι ανάλογο της πιθανότητας επιλογής της λύσης.

Παράμετροι ελέγχου: ο καθορισμός των καταλλήλων παραμέτρων είναι μια πολλή δύσκολη εργασία λόγω της ποικιλίας τους και των πολλών πιθανών επιλογών που υπάρχουν. Κάποιες από τις ρυθμίσεις που έχουν εξετασθεί είναι οι ακόλουθες:

- Ένα σταθερό μέγεθος πληθυσμού N_p .
- Ένα σταθερό ποσοστό διασταύρωσης p_c ίσο με 1. Εννοώντας ότι (2) τυχαία επιλεγμένα χρωμοσώματα- γονείς, ανταλλάσσουν πάντα γονίδια και παράγουν ένα ζεύγος παιδιών.
- Ένα υιοθετημένο ποσοστό μετάλλαξης p_m , το οποίο είναι υψηλό αρχικά και μειώνεται σταδιακά από τη ποικιλομορφία του πληθυσμού. Εάν η ποικιλομορφία του πληθυσμού γίνει πολύ «μικρή», τότε το ποσοστό της μετάλλαξης παίρνει ξανά την αρχική του υψηλή τιμή. Ποιο συγκεκριμένα, το p_m είναι αρχικά καθορισμένο ίσο με 0.8, και μειώνεται σε κάθε νέα γενιά με συντελεστή $\theta = 0.99$ χρησιμοποιώντας τη σχέση $p_m = \theta * p_m$. Η ποικιλομορφία του πληθυσμού χαρακτηρίζεται «μικρή», όταν η ελάχιστη πληθυσμιακή καταλληλότητα (minimum population fitness) ($fitness_{min}$) και η μέση πληθυσμιακή καταλληλότητα (average population fitness) ($fitness_{avg}$) είναι σχεδόν τα ίδια, π.χ. όταν ο λόγος $fitness_{min}/fitness_{avg}$ ξεπερνάει το καθορισμένο από το χρήστη όριο $D (=0.95)$. Όταν αυτή η συνθήκη επικρατεί, τα χρωμοσώματα σε όλο το πληθυσμό είναι σχεδόν τα ίδια και τότε έχουμε τη περίπτωση της πληθυσμιακής σύγκλισης.
- Ο γενετικός αλγόριθμος σταματά (κριτήριο τερματισμού), όταν ο μέγιστος αριθμός των γενιών (που έχει θέσει ο χρήστης) έχει ξεπεραστεί.

Το ποσοστό διασταύρωσης (*crossover rate*), που δηλώνεται με το p_c , ορίζεται ως ο λόγος του αριθμού των παιδιών που παράγονται από κάθε γενεά ως προς το μέγεθος του πληθυσμού (το οποίο δηλώνεται ως pop_size). Αυτός ο λόγος ελέγχει τον αναμενόμενο αριθμό $p_c \times pop_size$ των χρωμοσωμάτων που υφίστανται τη λειτουργία της διασταύρωσης. Ένα υψηλό ποσοστό διασταύρωσης επιτρέπει τη διερεύνηση περισσότερων γονιδίων από το χώρο των λύσεων και μειώνει τις περιπτώσεις ορισμού ενός εσφαλμένου βέλτιστου, αλλά αν το ποσοστό είναι πολύ υψηλό, έχει ως αποτέλεσμα την απώλεια ενός μεγάλου μέρους υπολογιστικού χρόνου εξερευνώντας μη-υποσχόμενες περιοχές του χώρου των λύσεων.

Το ποσοστό μετάλλαξης (*mutation rate*) που δηλώνεται ως p_m , ορίζεται ως το ποσοστό επί τοις εκατό του συνολικού αριθμού των γονιδίων στο πληθυσμό. Το ποσοστό μετάλλαξης ελέγχει το ποσοστό κατά το οποίο νέα γονίδια παρουσιάζονται στο πληθυσμό για δοκιμή. Αν είναι πολύ χαμηλό, πολλά γονίδια που θα ήταν χρήσιμα δε δοκιμάζονται ποτέ, αλλά αν είναι πολύ υψηλό, θα υπάρξει μεγάλη τυχαία διατάραξη, τα παιδιά θα αρχίσουν να χάνουν την ομοιότητα τους στους γονείς, και ο αλγόριθμος θα χάσει την ικανότητα του να μαθαίνει από την ιστορία της αναζήτησης.

4. Το προγραμματιστικό (scheduling) πρόβλημα:

Ο προγραμματισμός ασχολείται με το καταμερισμό των δραστηριοτήτων σε λιγιστούς πόρους, με την υποκειμενική της βελτιστοποίησης ενός ή παραπάνω μέτρων ικανοποίησης. Ανάλογα με τη κατάσταση, οι πόροι και οι δραστηριότητες, μπορούν να έχουν ποικίλες δομές. Πόροι μπορεί να είναι οι μηχανές ενός εργοστασίου, οι επεξεργαστές, η μνήμη και οι συσκευές εισόδου/ εξόδου σε ένα υπολογιστικό σύστημα, οι έξοδοι κινδύνου σε ένα αεροδρόμιο, κτλ. Ενώ δραστηριότητες μπορεί να είναι οι πολλαπλές λειτουργίες μιας κατασκευαστικής διαδικασίας, οι εκτελέσεις υπολογιστικών προγραμμάτων, οι προσγειώσεις και οι αναχωρήσεις σε ένα αεροδρόμιο, κτλ. Όσο αναφορά τα διάφορα μέτρα ικανοποίησης που μπορούν να βελτιστοποιηθούν, αυτά θα μπορούσαν να είναι το κόστος παραγωγής, το συνολικό κόστος ροής, κτλ.

Τα προγραμματιστικά προβλήματα κατηγοριοποιούνται, όσο αναφορά το περιβάλλον μηχανής στο οποίο ανήκουν. Πιθανό περιβάλλον μηχανής που μπορούμε να έχουμε στο προγραμματιστικό (scheduling) πρόβλημα είναι:

- *Single Machine (l)*: υπάρχει μια μόνο μηχανή στο σύστημα. Ειδική περίπτωση των άλλων πιο πολύπλοκων περιβάλλοντων μηχανών.
- *Parallel & Identical Machines (Pm)*: υπάρχουν m - ταυτόσημες μηχανές εν παραλλήλω. Ο αριθμός των μηχανών καθορίζεται συνήθως σαν παράμετρος στην είσοδο. Κάθε εργασία j χρειάζεται μια μόνον επεξεργασία και μπορεί να επεξεργασθεί σε οποιαδήποτε από τις m - μηχανές.
- *Uniform Machines (Qm)*: υπάρχουν m - μηχανές εν παραλλήλω, αλλά οι μηχανές έχουν διαφορετικές ταχύτητες. Η μηχανή i , $1 \leq i \leq m$, έχει ταχύτητα s_i . Ο χρόνος p_{ij} που η εργασία j ξοδεύει στη μηχανή i , είναι ίσο με p_j / s_i , θεωρώντας ότι η εργασία j έχει επεξεργασθεί εντελώς στη μηχανή i .
- *Unrelated Machines (Rm)*: υπάρχουν m - μηχανές εν παραλλήλω, αλλά κάθε μηχανή μπορεί να επεξεργασθεί τις εργασίες με διαφορετική ταχύτητα. Η μηχανή i μπορεί επεξεργασθεί την εργασία j με ταχύτητα s_{ij} . Ο χρόνος p_{ij} που μια εργασία j ξοδεύει σε μια μηχανή i ισούται με p_j / s_{ij} , θεωρώντας ότι η εργασία j έχει επεξεργασθεί εντελώς στη μηχανή i .

- *Job Shop (Jm)*: σε ένα job shop με m - μηχανές, κάθε εργασία έχει τη δική της προκαθορισμένη πορεία να ακολουθήσει. Μπορεί να επισκεφθεί κάποιες μηχανές περισσότερες από μια φορές, και μπορεί να μην επισκεφθεί κάποιες άλλες ούτε μια φορά.
- *Flow Shop (Fm)*: σε ένα flow shop με m μηχανές, οι μηχανές είναι γραμμικά τοποθετημένες και οι εργασίες ακολουθούν όλες την ίδια διαδρομή (από τη πρώτη έως τη τελευταία).
- *Open Shop (Om)*: σε ένα open shop με m - μηχανές, κάθε εργασία πρέπει να επεξεργασθεί ακριβώς μια φορά από κάθε μηχανή. Αλλά η σειρά επεξεργασίας είναι αδιάφορη.

4.1 Το πρόβλημα Ασυσχέτιστων Μηχανών(Unrelated Machines):

Στο πρόβλημα Ασυσχέτιστων Μηχανών (R_m) με αριθμό μηχανών m , οι μηχανές είναι παράλληλα τοποθετημένες και οι εργασίες δεν ακολουθούν ίδια διαδρομή. Το προγραμματιστικό πρόβλημα ασυσχέτιστων μηχανών (R_mSP) περιγράφεται γενικώς με το παρακάτω τρόπο: υπάρχουν m - μηχανές και n - εργασίες, κάθε i - εργασία μπορεί να επεξεργασθεί εντελώς από μια και μόνο j - μηχανή, σε συγκεκριμένα χρονικά περιθώρια. Ο χρόνος επεξεργασίας της εργασίας i στη μηχανή j δίνεται από το t_{ij} ($i= 1, \dots, n; j= 1, \dots, m$). Κάθε μηχανή μπορεί να επεξεργασθεί τις εργασίες με διαφορετική ταχύτητα. Η υποκειμενική συνάρτηση συνίσταται στο να βρεθεί η τοποθέτηση των εργασιών, που ελαττώνει τα (3) κριτήρια. Οι κύριες υποθέσεις για αυτού του είδους τα προβλήματα είναι:

- Οι μηχανές είναι διαθέσιμες συνεχώς.
- Κάθε μηχανή επεξεργάζεται μόνο μια εργασία κάθε φορά.
- Κάθε εργασία επεξεργάζεται μόνο από μια μηχανή κάθε φορά.
- Οι λειτουργίες δεν είναι προ- εγκατεστημένες.
- Οι χρόνοι ρύθμισης για τις λειτουργίες είναι ανεξάρτητοι από την ακολουθία και μπορεί να συμπεριλαμβάνονται στους χρόνους επεξεργασίας (p_i) ή στους χρόνους εκπλήρωσης (c_i).

Το προγραμματιστικό πρόβλημα ασυσχέτιστων μηχανών μπορεί να ταξινομηθεί μέσα στις επόμενες (3) κατηγορίες:

- Ντετερμινιστικό πρόβλημα
- Στοχαστικό πρόβλημα
- Ασαφές πρόβλημα

Τα ντετερμινιστικά προβλήματα (το παρόν πρόβλημα ανήκει σε αυτή τη κατηγορία) θεωρούν ότι οι σταθεροί χρόνοι επεξεργασίας είναι γνωστοί. Στις στοχαστικές περιπτώσεις, οι χρόνοι επεξεργασίας ποικίλουν σύμφωνα με τις επιλεγμένες πιθανότητες διανομών. Στο γενικό πλαίσιο της ασαφούς απόφασης, η ασαφής ημερομηνία παράδοσης (due date) καθορίζεται για κάθε εργασία να αναπαριστά το βαθμό της ικανοποίησης αυτών που λαμβάνουν αποφάσεις (decision makers) για το χρόνο εκπλήρωσης (completion time) μιας εργασίας. Αν μια εργασία έχει χρόνο επεξεργασίας ίσο με μηδέν σε μια ή περισσότερες μηχανές, τότε ονομάζεται *γενικό πρόβλημα Unrelated Machines (general unrelated machines problem)* ή αλλιώς *καθαρό πρόβλημα Unrelated Machines (pure unrelated machines problem)*. Η πλειονότητα της προσπάθειας των ερευνητών τα τελευταία 30 χρόνια είναι αφοσιωμένη στο καθαρό ντετερμινιστικό προγραμματιστικό πρόβλημα. Ένα τέτοιο πρόβλημα συνήθως χαρακτηρίζεται σαν $n/m/Rm/c_{max}$, το οποίο σημαίνει n -εργασίες/ m -μηχανές/*Unrelated Machines* / *maximum flow time*.

4.2 Ο πληθυσμός των συναρτήσεων καταλληλότητας:

Για ένα δεδομένο πρόβλημα, η συνάρτηση καταλληλότητας ικανοποιεί το: $f(u) \geq 0$ με $f(u) = 0$ εάν το u είναι μια λύση. Λέμε ότι η $f(u)$ είναι μια *αυστηρή (strict)* συνάρτηση καταλληλότητας, εάν επιπροσθέτως: $f(u) = 0$ εάν και μόνο εάν το u είναι μία λύση.

Το άθροισμα οποιουδήποτε αριθμού από συναρτήσεις καταλληλότητας με θετικά βάρη είναι επίσης μια συνάρτηση καταλληλότητας, και είναι αυστηρή, εάν οποιοδήποτε «συστατικό» της είναι αυστηρό.

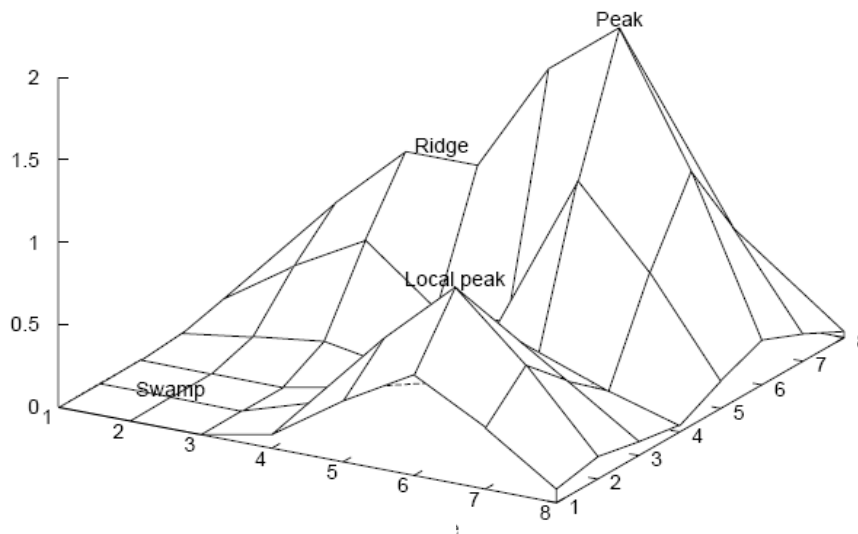
4.3 Pareto- βέλτιστες λύσεις:

Αυτό που ερευνείται σε μια γενική πολυ- αντικειμενική βελτιστοποίηση, είναι το ολικό μέγιστο (παγκόσμιο βέλτιστο). Μια ακριβής λύση, κατά την οποία όλες οι μεταβλητές απόφασης ικανοποιούν τους σχετικούς περιορισμούς, και όλες οι μοναδικές υποκειμενικές συναρτήσεις έχουν φθάσει τις σχετικές τους βέλτιστες τιμές. Μια τέτοια λύση μπορεί ακόμα και να μην υπάρχει. Συνήθως, δεν υπάρχει μοναδική βέλτιστη λύση σε προβλήματα αυτού του είδους, αλλά ένα σετ από βέλτιστες λύσεις. Αυτό το σετ είναι γνωστό ως *Pareto- βέλτιστες λύσεις*. Οι λύσεις σε αυτό το σετ είναι τέτοιες που είναι μη- κυριαρχημένες από άλλες λύσεις του χώρου αναζήτησης, όταν όλες οι υποκειμενικές λαμβάνονται υπόψη, καθώς επίσης δεν κυριαρχούν η μια την άλλη μέσα στο σετ.

Ακόμη ένα θέμα των πολυ- υποκειμενικών προβλημάτων βελτιστοποίησης, είναι πως θα βρεθούν όλες οι δυνατές ανταλλαγές μεταξύ των πολλαπλών υποκειμενικών συναρτήσεων που συνήθως συγκρούονται. Επειδή είναι δύσκολο να επιλεγεί μια μοναδική λύση για ένα πολυ- υποκειμενικό πρόβλημα βελτιστοποίησης χωρίς την επαναλαμβανόμενη διάδραση με το λήπτη- αποφάσεων, επιλέγεται η γενική προσέγγιση του σετ από βέλτιστες λύσεις Pareto, οι οποίες εμφανίζονται στο λήπτη- αποφάσεων, και τότε εκείνος μπορεί να επιλέξει οποιαδήποτε από τις βέλτιστες λύσεις Pareto. Για να βρεθούν όλες οι Pareto βέλτιστες λύσεις από τους γενετικούς αλγορίθμους, η ποικιλία των ατόμων πρέπει να διατηρηθεί σε κάθε γενιά.

Σύμφωνα με το 2^ο χαρακτηριστικό του αλγορίθμου (διαφύλαξη της ελίτ), ένα προσωρινό σετ από Pareto βέλτιστες λύσεις διατηρείται κατά την εκτέλεση του πολυ- υποκειμενικού γενετικού αλγορίθμου (ο γενετικός αλγόριθμος *μιας*- υποκειμενικής με σταθερά βάρη δε μπορεί να βρει ένα concave Pareto front ακόμα κι αν εφαρμοστούν ποικίλες τιμές βαρών). Ένας συγκεκριμένος αριθμός από άτομα σ' αυτό το σετ κληρονομούνται στην επόμενη γενιά, σαν επίλεκτα άτομα.

Ένας τρόπος για να αποφευχθεί το φαινόμενο των επικαλυπτόμενων λύσεων, είναι η επιλογή μεγάλου αριθμού υποκειμενικών συναρτήσεων. Σε ένα πρόβλημα (2) υποκειμενικών συναρτήσεων, ο αριθμός των διαφορετικών λύσεων είναι μικρότερος από ότι σε ένα πρόβλημα τριών ή τεσσάρων πολυ- υποκειμενικών συναρτήσεων. Όταν ο αριθμός των υποκειμενικών συναρτήσεων είναι μεγάλος, δεν υπάρχουν πολλές επικαλυπτόμενες λύσεις σε κάθε γενιά.



Εικόνα 9. Παράδειγμα "τοπίου" της fitness.

4.4 Πολυ- κριτηριακή βελτιστοποίηση:

i. Μη αλληλεπιδρώντα κριτήρια:

Ελάχιστο- Μέγιστο: η πιθανότητα της συμπεριφορικής έκφρασης των κριτηρίων. (3) κριτήρια με τα ίδια βάρη, μπορούν να έχουν διαφορετικό ποσοστό «ικανοποίησης», αφού η συμπεριφορά τους για τη λήψη μιας απόφασης διαφέρει. Τυπικά η συμπεριφορά μπορεί να έχει ένα μεγάλο εύρος που κυμαίνεται, από τη τέλεια ανοχή έως τη πλήρη έλλειψη ανεκτικότητας. Οι ανεκτικοί λήπτες αποφάσεων (γιατί όπως έχουμε ήδη πει, ο λήπτης αποφάσεων είναι αυτός που θα αποφασίσει για τη βέλτιστη λύση μέσα από το σετ βέλτιστων λύσεων Pareto) μπορούν να δεχθούν ότι μόνο μερικά κριτήρια- τουλάχιστον ένα- εκπληρώνονται. Αυτό ανταποκρίνεται σε

μια διαζευκτική συμπεριφορά, της οποίας ακραίο παράδειγμα είναι το maximum. Από την άλλη πλευρά, οι μη ανεκτικοί λήπτες αποφάσεων απαιτούν ότι όλα τα κριτήρια πρέπει να εκπληρώνονται εξίσου. Μια συνδετική συμπεριφορά, της οποίας ακραίο παράδειγμα είναι το minimum.

Για παράδειγμα επιθυμείται να :

$$\min f(x),$$

ελαχιστοποιώντας την $f(x)$, ουσιαστικά ελαχιστοποιείται η μέγιστη εκ των (3) συναρτήσεων:

$$f(x) = \max (\text{Total Flowtime}, \text{Makespan}, \text{Total Tardiness}),$$

άρα:

$$\min (\max (\text{Total Flowtime}, \text{Makespan}, \text{Total Tardiness}))$$

Μέσος Όρος: η πιθανότητα έκφρασης μιας αντισταθμιστικής συνέπειας. Αντιστάθμιση υπάρχει, όταν ένα κακό αποτέλεσμα ενός κριτηρίου μπορεί να αντισταθμιστεί από ένα καλό αποτέλεσμα ενός άλλου κριτηρίου.

Για παράδειγμα επιθυμείται να :

$$\min f(x),$$

ελαχιστοποιώντας την $f(x)$, ουσιαστικά ελαχιστοποιείται ο μέσος όρος των (3) συναρτήσεων:

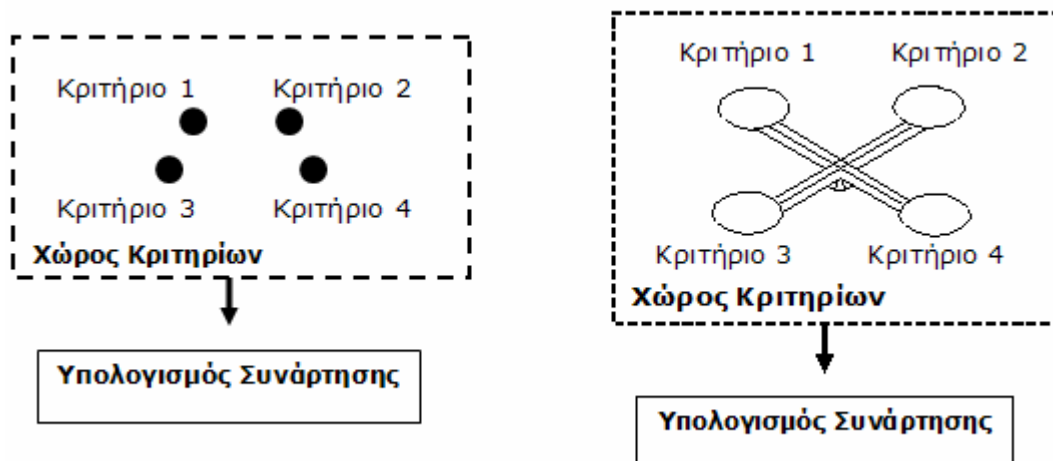
$$f(x) = \frac{\text{TotalFlowtime} + \text{Makespan} + \text{TotalTardiness}}{3}$$

άρα:

$$\min \left(\frac{\text{TotalFlowtime} + \text{Makespan} + \text{TotalTardiness}}{3} \right)$$

ii. Αλληλεπιδρώντα κριτήρια:

Συνήθως, οι σχεδιαστές ενός πολυ- υποκειμενικής βελτιστοποίησης αλγορίθμου χρησιμοποιούν ανεξάρτητα κριτήρια για να απλοποιήσουν το πρόβλημα της λήψης αποφάσεων. Η αλληλεπίδραση μεταξύ των κριτηρίων αυξάνει τη πολυπλοκότητα του προβλήματος, αλλά αν δεν είναι μοντελοποιημένο τότε κάποιες πληροφορίες για το πρόβλημα δε χρησιμοποιούνται. Όπως φαίνεται και στην Εικόνα (10), η αλληλεπίδραση ανάμεσα στα κριτήρια παρουσιάζεται σαν μια ζυγαριά με (4) πλάστιγγες, αναπαριστώντας τα (4) κριτήρια. Αν η τιμή ενός κριτηρίου αλλαχθεί, τότε επηρεάζονται και οι τιμές των άλλων κριτηρίων. Η αλληλεπίδραση μεταξύ των κριτηρίων μπορεί να αναπαρασταθεί χρησιμοποιώντας την ασαφή ολοκλήρωση Choquet. Η δυνατότητα όμως που δίνεται στην αναπαράσταση έχει ως αντίτιμο τη πολυπλοκότητα του μοντέλου, η οποία είναι αυξημένη σε σχέση με τα μη-αλληλεπιδρώντων κριτηρίων μοντέλα.



Εικόνα 10. Αλληλεπιδρώντα και μη-αλληλεπιδρώντα κριτήρια.

Χρησιμοποιείται το discrete Choquet integral (ασυνεχές ολοκλήρωμα) για το καθορισμό της υποκειμενικής συνάρτησης, μέσω του οποίου δίδεται η δυνατότητα να συνυπολογιστεί η αλληλεπίδραση μεταξύ των κριτηρίων.

Μαθηματικό Υπόβαθρο

Θεωρείται ότι $S = \{S_1, S_2, \dots, S_m\}$ είναι ένα σετ από εναλλακτικές λύσεις, από τις οποίες ο λήπτης αποφάσεων πρέπει να αποφασίσει. Επιπροσθέτως, θεωρείται ένα σετ από n - κριτήρια $C = \{C_1, C_2, \dots, C_n\}$ και η συσχέτιση μεταξύ κάθε μέλους των S και C (Marichal, 2000):

$$x^i = (x_1^i, x_2^i, \dots, x_n^i) \in \mathbb{R}^n \quad i = 1, \dots, m$$

Η ασαφής μέτρηση (ή χωρητικότητα Choquet) στο C είναι ένα μονοτονικό σετ συναρτήσεων:

$$u : 2^C \rightarrow [0,1] \text{ με } u(\emptyset) = 0 \text{ και } u(C) = 1$$

Τα ασαφή μέτρα αναπαριστούν τη σημαντικότητα των κριτηρίων C . Δεν είναι μόνο οι παράγοντες βάρους που καθορίζονται για κάθε κριτήριο, αλλά επιπροσθέτως οι παράγοντες βάρους που καθορίζονται για όλους τους συνδυασμούς μεταξύ των κριτηρίων του σετ C .

Ένας κατάλληλος συγκεντρωτικός τελεστής, που γενικοποιεί το σταθμισμένο αριθμητικό μέσο, είναι το ασυνεχές ολοκλήρωμα Choquet (Garbisch, 1996):

$$C_u(x^i) := \sum_{j=1}^n x_j^i [u(A(C_j)) - u(A(C_{j+1}))]$$

όπου, $u \in F_C$, F_C δηλώνει το σετ από όλα τα ασαφή μέτρα στο C , (\cdot) υποδεικνύει μια μετάθεση του C έτσι ώστε: $x_j \leq \dots \leq x_n$, $A(C_j) = \{C_j, \dots, C_n\}$, $A(C_{n+1}) = \emptyset$.

4.5 Γιατί χρησιμοποιούνται οι ΓΑ στα πολυ- υποκειμενικής βελτιστοποίησης προβλήματα:

Όπως ειπώθηκε και παραπάνω χρησιμοποιείται η έννοια των γονιδίων και των χρωμοσωμάτων στο πρόβλημα. Όταν χρησιμοποιείται η πολυπλοκότητα των χρωμοσωμάτων σαν μέρος μιας συνάρτησης καταλληλότητας με μια υποκειμενική, πρέπει να οριστούν κάποιες παράμετροι και να καθορισθεί πόσο η πολυπλοκότητα ενός χρωμοσώματος ενός ατόμου θα επηρεάσει τη καταλληλότητά του. Είναι όμως το λιγότερο δύσκολο και χρονοβόρο να βρεθούν οι βέλτιστες ρυθμίσεις για αυτές τις παραμέτρους. Για να υπάρχουν και να ρυθμίζονται με ευκολία πολλές παράμετροι, ανεξάρτητες μεταξύ τους, χρησιμοποιούνται πολλές υποκειμενικές αντί για μια.

Στα πολυ- υποκειμενικής βελτιστοποίησης προβλήματα, χρησιμοποιούνται συνήθως οι ΓΑ, κυρίως λόγω των κληρονομήσιμων χαρακτηριστικών που αφορούν τον πληθυσμό του χώρου αναζήτησης, που διερευνείται με βάση κάποια σετ (set-based). Σκοπός του ανεπτυγμένου πολυ- υποκειμενικού γενετικού αλγορίθμου (MOGA), δεν είναι να βρει μια μοναδική σχεδόν- βέλτιστη λύση, αλλά να αποφασίσει ένα σετ από Pareto- βέλτιστες (μη- κυριαρχημένες) λύσεις για το πρόβλημα. Αυτός που λαμβάνει τις αποφάσεις μπορεί μετά να επιλέξει αυτή τη λύση που επιτυγχάνει περισσότερο τις προτιμήσεις/ προτεραιότητες του.

Οι Γενετικοί Αλγόριθμοι είναι διαδικασίες αναζήτησης που βασίζονται στο πληθυσμό και λειτουργούν πολύ καλύτερα από τις υπόλοιπες μεθόδους, ειδικά όταν αντιμετωπίζεται το πρόβλημα της βελτιστοποίησης με μη-παραγωγίσιμες ή ασυνεχείς συναρτήσεις.

4.6 Ένας γενετικός αλγόριθμος (MOGA) στα πολυ- υποκειμενικά PmSP:

Σε ένα πολυ- υποκειμενικό πρόβλημα, γίνεται προσπάθεια να βρεθεί ένα Pareto σετ από βέλτιστες λύσεις. Το σετ Pareto περιέχει όλες αυτές τις μη-κυριαρχημένες λύσεις του προβλήματος, ευρεθείσες έτσι ώστε καμία άλλη λύση να υπερσχύει αυτών ως προς όλες τις υποκειμενικές ξεχωριστά. Με το σκοπό αυτό, ο προτεινόμενος MOGA προσπαθεί να βρει ένα σετ από μη-κυριαρχημένες λύσεις παρά από μια μοναδική λύση. Ο MOGA έχει τα (2) παρακάτω χαρακτηριστικά:

- 1) Διατηρεί ένα ξεχωριστό πληθυσμό από ποικίλες Pareto- βέλτιστες λύσεις επαναληπτικά ανανεώσιμες από γενιά σε γενιά.
- 2) Χρησιμοποιεί μια στρατηγική ελιτιστικής διατήρησης, με την οποία μια μερίδα από το σχετιζόμενο πληθυσμό αντικαθίσταται τυχαία από έναν αριθμό ελιτιστικές λύσεων Pareto (μια λύση με τις καλύτερες τιμές για κάθε υποκειμενική μπορεί να θεωρηθεί σαν ελιτιστική λύση).

Για το προγραμματιστικό πρόβλημα (3) κριτηρίων υπάρχουν (3) ελίτ (ακραίες) λύσεις στον αναπτυσσόμενο πληθυσμό, από τις οποίες κάθε μια βελτιστοποιεί μια υποκειμενική. Αυτές οι λύσεις αντιγράφονται πάντα μέσα στο πληθυσμό Pareto. Το σετ Pareto ολοκληρώνεται στη συνέχεια και με επιπρόσθετες ελιτιστικές λύσεις. Ο πληθυσμός Pareto της τελικής γενιάς περιέχει τις σχετικά-βέλτιστες λύσεις του πολυ- υποκειμενικού προγραμματιστικού προβλήματος. Ο λήπτης αποφάσεων μπορεί τότε να επιλέξει τη λύση που ολοκληρώνει περισσότερο τις προτιμήσεις του.

5. Μοντελοποίηση προβλήματος:

5.1 Βασικές υποθέσεις:

Οι βασικές υποθέσεις που διέπουν το πρόβλημα μιας μηχανής διέπουν και τα ανώτερου αριθμού μηχανών προβλήματα, και περιγράφονται ως εξής:

- Ένα σετ από n - ανεξάρτητες εργασίες είναι διαθέσιμες για επεξεργασία σε χρόνο μηδέν (0).
- Οι χρόνοι ρύθμισης των εργασιών είναι ανεξάρτητες από την ακολουθία των εργασιών και μπορούν να περικλείονται στους χρόνους επεξεργασίας ή στους χρόνους εκπλήρωσης.
- Οι εργασίες είναι γνωστές εκ των προτέρων.
- Μια μηχανή είναι συνεχώς διαθέσιμη και ποτέ δεν μένει αδρανής όταν μια δουλειά περιμένει.
- Οι λειτουργίες δεν είναι προ- εγκατεστημένες.

Το πρόβλημα είναι να καθοριστεί μια επεξεργαστική ακολουθία εργασιών, έτσι ώστε να βελτιστοποιούνται κάποια μέτρα επιτυχίας.

5.2 Καθορισμός των υποκειμενικών συναρτήσεων:

Δηλώνεται από την αρχή η πρόθεση να ελαχιστοποιηθούν τα (3) κριτήρια: αυτό του κόστους (makespan), των καθυστερήσεων (tardiness) και του συνολικού χρόνου ροής (total flowtime). Τα (3) αυτά κριτήρια θα αποτελέσουν και τις (3) υποκειμενικές συναρτήσεις.

Θεωρούνται οι εργασίες (j_1, j_2, \dots, j_n) και p_i ο χρόνος επεξεργασίας (processing time), d_i η ημερομηνία παράδοσης (due date), w_i το βάρος (weight), c_i ο χρόνος εκπλήρωσης (completion time) και r_i ο χρόνος στον οποίο είναι έτοιμη (ready/ release time) για κάθε εργασία j_i αντιστοίχως.

Έτσι ορίζεται:

- *Makespan*: ορίζεται ως $c_{max} = c(j_n, m)$, όπου το j_n παίρνει τιμές $\{j_1, \dots, j_n\}$. Δηλαδή το κόστος κατασκευής για μια μηχανή υπολογίζεται με βάση το μεγαλύτερο χρόνο εκπλήρωσης, από όλες τις εργασίες που μπορεί να παίρνει αυτή η μηχανή.
- *Flowtime*: F_i είναι ορισμένη ως $F_i = c_i - r_i$, η ποσότητα του χρόνου την οποία η εργασία j_i ξοδεύει στο σύστημα. Ο χρόνος ροής μετρά την αντίδραση του συστήματος στις ατομικές απαιτήσεις για εξυπηρέτηση και αναπαριστά το διάστημα μια εργασία περιμένει μεταξύ της άφιξής της και της αποχώρησής της. Στο πρόβλημα ισχύει πως: ο χρόνος επεξεργασίας p_i δεν εμπεριέχει καθυστερήσεις, ο χρόνος ολοκλήρωσης c_i δεν εμπεριέχει καθυστερήσεις αλλά εμπεριέχει τους χρόνους ρύθμισης της μηχανής, και ο χρόνος στον οποίο η εργασία είναι έτοιμη r_i εμπεριέχει τους χρόνους ρύθμισης και τις καθυστερήσεις του συστήματος.
(Total flowtime: το άθροισμα όλων των χρόνων εκπλήρωσης όλων των εργασιών)
- *Lateness*: L_i είναι ορισμένη ως $L_i = c_i - d_i$, η ποσότητα του χρόνου κατά την οποία ο χρόνος εκπλήρωσης (completion time) για μια εργασία j_i ξεπερνά την ημερομηνία παράδοσής της (due date). Η *καθυστέρηση* μετρά τη συμφωνία του χρονοδιαγράμματος με τις δεδομένες ημερομηνίες παράδοσης. Μπορεί να έχει είτε θετική είτε αρνητική τιμή. Η αρνητική καθυστέρηση σημαίνει ότι μια εργασία έχει ετοιμαστεί πριν την ημερομηνία παράδοσής της. Σε πολλές περιπτώσεις, κυρώσεις και κόστος σχετίζονται με τη θετική καθυστέρηση, αλλά όφελος δε σχετίζεται με την αρνητική καθυστέρηση. Γι' αυτό μόνο η θετική καθυστέρηση λαμβάνεται υπ' όψη, την οποία και ονομάζουμε *tardiness*.
- *Tardiness*: T_i ορίζεται ως $T_i = \max \{0, L_i\}$, όπου το L_i παίρνει τιμές από $\{L_1, \dots, L_i\}$, και αναπαριστά τη καθυστέρηση μιας εργασίας j_i , εάν αυτή

αποτύχει να εκπληρώσει την ημερομηνία παράδοσής της, ή αλλιώς μηδέν (Tardiness: η max Lateness που σημειώθηκε από μια εργασία σε μια μηχανή στο γονίδιο. Total Tardiness: το άθροισμα των επιμέρους καθυστερήσεων για κάθε μηχανή).

- Η *due date* (d_i) για κάθε εργασία j_i ($i= 1, \dots, n$) προσδιορίζεται χρησιμοποιώντας τη σχέση $d_i = C(j_i, m) + \text{rnd}(-100, 100)$, με το $C(j_i, m)$ να έχει την έννοια του χρόνου ολοκλήρωσης (completion time) της εργασίας j_i στη τελευταία μηχανή (m) και $\text{rnd}(\text{random})(-100, 100)$ είναι μια συνάρτηση που επιστρέφει ένα τυχαίο ακέραιο αριθμό, ο οποίος βρίσκεται στο διάστημα $(-100, 100)$.

5.3 Καθορισμός της συνάρτησης καταλληλότητας:

Οι εξισώσεις των τριών κριτηρίων:

$$\text{Makespan: } c_{max} = c(j_n, m)$$

$$\text{Flowtime: } \sum_{i=1}^n F_i = c_i - r_i$$

$$\text{Tardiness: } T_i = \max\{0, L_{ij}\}$$

Επίσης τα κριτήρια αναπαρίστανται και ως: $C = \{f_1(x), f_2(x), f_3(x)\}$

όπου:

$f_1(x)$: Makespan

$f_2(x)$: Total Tardiness

$f_3(x)$: Total Flowtime

Επιδιώκεται η ελαχιστοποίηση της συνάρτησης καταλληλότητας:

$$\text{Minimize } f(x) = (f_1(x), f_2(x), f_3(x))$$

Η συνάρτηση καταλληλότητας (fitness function ή score) είναι η «οδός», μέσω της οποίας βρίσκεται το σετ βέλτιστων λύσεων. Όπως ειπώθηκε και προηγουμένως, υπάρχουν πολύ τρόποι εύρεσης αυτών των λύσεων: με την ελαχιστοποίηση του μεγίστου, το μέσο όρο, τα βάρη και την ασαφή λογική Choquet. Σε κάθε μια από αυτές τις περιπτώσεις, η συνάρτηση καταλληλότητας είναι διαφορετική.

Ελαχιστοποίηση Μεγίστου:

Απαιτείται όλα τα κριτήρια να εκπληρώνονται εξίσου. Ελαχιστοποιείται η μέγιστη εκ των (3) συναρτήσεων:

$$\text{Score} = \min (\max (\text{Total Flowtime}, \text{Makespan}, \text{Total Tardiness}))$$

Μέσος Όρος:

Αντισταθμίζεται ένα κακό αποτέλεσμα με ένα καλό. Ελαχιστοποιείται ο μέσος όρος των (3) συναρτήσεων:

$$\text{Score} = \min \left(\frac{\text{TotalFlowtime} + \text{Makespan} + \text{TotalTardiness}}{3} \right)$$

Βάρη:

Η κάθε υποκειμενική συνάρτηση επηρεάζει τη συνάρτηση καταλληλότητας σε ένα βαθμό, δηλαδή δεν έχει το κάθε κριτήριο την ίδια βαρύτητα στη συνάρτηση καταλληλότητας. Για να υποδηλωθεί το ποσοστό που επηρεάζει το κάθε κριτήριο τη συνάρτηση καταλληλότητας, χρησιμοποιείται η έννοια του βάρους. Συμβολίζεται με w_i και όσο πιο μεγάλο είναι τόσο περισσότερη βαρύτητα έχει στη συνάρτηση καταλληλότητας. Άρα η συνάρτηση γίνεται ως εξής:

$$\text{Score} = \frac{w_1 \cdot f_1(x) + w_2 \cdot f_2(x) + w_3 \cdot f_3(x)}{w_1 + w_2 + w_3}$$

Ισχύει επίσης:

$$\sum_{i=1}^3 w_i = 1$$

Για παράδειγμα, σε κάποιο πρόβλημα μπορεί να θεωρείται ως μέγιστης σημαντικότητας κριτήριο, ο συνολικός χρόνος ροής (Total Flowtime, $f_3(x)$), χαμηλότερης βαρύτητας κριτήριο οι συνολικές καθυστερήσεις (Total Tardiness, $f_2(x)$) και τελευταίας βαρύτητας το κόστος κατασκευής (Makespan, $f_1(x)$). Τα βάρη λοιπόν θα ήταν ως εξής:

$$w_1 = 0.2$$

$$w_2 = 0.3$$

$$w_3 = 0.5$$

Ασαφής λογική Choquet:

Ασαφή μέτρα για κάθε κριτήριο και για κάθε ζεύγος κριτηρίων πρέπει να δημιουργηθούν. Στη συγκεκριμένη περίπτωση (6) ασαφή μέτρα πρέπει να δημιουργηθούν. Ο σχεδιαστής πρέπει να αποδώσει (3) ασαφή μέτρα για τη σημαντικότητα του κάθε κριτηρίου και (3) για την αλληλεπίδραση μεταξύ τους. Τα ασαφή μέτρα για την αλληλεπίδραση των κριτηρίων μεταξύ τους πρέπει να αποδοθούν έτσι ώστε να ικανοποιήσουν συγκεκριμένα κριτήρια.

Εάν:

$f_1(x)$: Makespan

$f_2(x)$: Total Tardiness

$f_3(x)$: Total Flowtime

| | |
|-------------------|--|
| $u(\{f_1\})$ | → Ποσοστό επηρεασμού του κριτηρίου f_1 στην υπ. συνάρτηση u |
| $u(\{f_2\})$ | → Ποσοστό επηρεασμού του κριτηρίου f_2 στην υπ. συνάρτηση u |
| $u(\{f_3\})$ | → Ποσοστό επηρεασμού του κριτηρίου f_3 στην υπ. συνάρτηση u |
| $u(\{f_1, f_2\})$ | → Ποσοστό επηρεασμού αλληλεπίδρασης των κριτηρίων f_1 & f_2 στην υπ. συνάρτηση u |
| $u(\{f_2, f_3\})$ | → Ποσοστό επηρεασμού αλληλεπίδρασης των κριτηρίων f_2 & f_3 στην υπ. συνάρτηση u |
| $u(\{f_1, f_3\})$ | → Ποσοστό επηρεασμού αλληλεπίδρασης των κριτηρίων f_3 & f_1 στην υπ. συνάρτηση u |

Βρέθηκαν ως βέλτιστες παράμετροι της ασαφής ολοκλήρωσης Choquet:

$$\begin{aligned} u(\{f_1\}) &= 0.45 \\ u(\{f_2\}) &= 0.45 \\ u(\{f_3\}) &= 0.3 \\ u(\{f_1, f_2\}) &= 0.9 \\ u(\{f_1, f_3\}) &= 0.46 \\ u(\{f_2, f_3\}) &= 0.46 \end{aligned}$$

$$\begin{aligned} u(\{f_1\}) + u(\{f_2\}) &= u(\{f_1, f_2\}) \\ u(\{f_2\}) + u(\{f_3\}) &> u(\{f_2, f_3\}) \\ u(\{f_1\}) + u(\{f_3\}) &> u(\{f_1, f_3\}) \end{aligned}$$

Σύμφωνα με το πίνακα, υπάρχει αρνητική αλληλεπίδραση μεταξύ των συνολικών καθυστερήσεων και του συνολικού χρόνου ροής, και μεταξύ του κόστους κατασκευής και του συνολικού χρόνου ροής. Δεν υπάρχει αλληλεπίδραση μεταξύ του κόστους κατασκευής και των συνολικών καθυστερήσεων.

Ο υπολογισμός της συνάρτησης καταλληλότητας προσδιορίζεται από το discrete Choquet integral (ασυνεχές ολοκλήρωμα) ως εξής:

$$\text{Score}(x^i) := \sum_{j=1}^3 x_j^i [u(A(C_j)) - u(A(C_{j+1}))]$$

Για παράδειγμα, εάν υποθεθεί ότι οι τιμές για την εναλλακτική λύση S_1 σχετίζονται με το προφίλ x^1 , έτσι ώστε: $x^1 = (x_1^1, x_2^1, x_3^1) = (x_{f_1(x)}^1, x_{f_2(x)}^1, x_{f_3(x)}^1)$, όπου $x_{f_1(x)}^1 < x_{f_3(x)}^1 < x_{f_2(x)}^1$. Τα σετ για τον υπολογισμό της συνάρτησης καταλληλότητας σύμφωνα με τη παραπάνω σχέση, φαίνονται στο πίνακα παρακάτω.

| j | x_j^1 | $A(C_j)$ | $A(C_{j+1})$ |
|---|------------------------|--|-------------------------------------|
| 1 | $x_1^1 = x_{f_1(x)}^1$ | $\{C_1, C_2, C_3\} = \{f_1(x), f_2(x), f_3(x)\} = C$ | $\{C_2, C_3\} = \{f_2(x), f_3(x)\}$ |
| 2 | $x_2^1 = x_{f_2(x)}^1$ | $\{C_2, C_3\} = \{f_2(x), f_3(x)\}$ | $\{C_3\} = \{f_3(x)\}$ |
| 3 | $x_3^1 = x_{f_3(x)}^1$ | $\{C_3\} = \{f_3(x)\}$ | \emptyset |

Σε αυτή την εργασία, αντιπαρατίθεται η ασαφής λογική Choquet με τις υπόλοιπες εφαρμόσιμες συναρτήσεις καταλληλότητας, και διαπιστώνεται, όπως φαίνεται και παρακάτω μέσω των αποτελεσμάτων, πως η μέθοδος Choquet υπερέχει των άλλων.

5.4 Καθορισμός των παραμέτρων:

Πιθανότητα διασταύρωσης (pcross): 0.8

Πιθανότητα μετάλλαξης (pm): 0.1

Μέγεθος πληθυσμού (popsize): 60

Κριτήριο τερματισμού- μέγιστος πληθυσμός γενιών (maxgen): 100

6. Παράδειγμα:

Θεωρείται ένα πρόβλημα $m \times n$, όπου m - αριθμός των μηχανών, ο οποίος είναι ίσος με 3, και n - αριθμός των εργασιών, ο οποίος είναι ίσος με 4 (3×4). Οι παρακάτω πίνακες είναι δεδομένα του προβλήματος:

| $n \setminus m$ | $m = 1$ | $m = 2$ | $m = 3$ |
|-----------------|---------|---------|---------|
| $n = 1$ | 10 | 8 | 30 |
| $n = 2$ | 15 | 35 | 10 |
| $n = 3$ | 6 | 50 | 5 |
| $n = 4$ | 28 | 40 | 70 |

Εικόνα 11. Πίνακας χρόνου επεξεργασίας p_i (processing time) με μονάδα μέτρησης το δευτερόλεπτο (second).

| $n \setminus m$ | $m = 1$ | $m = 2$ | $m = 3$ |
|-----------------|---------|---------|---------|
| $n = 1$ | 13 | 12 | 34 |
| $n = 2$ | 17 | 42 | 17 |
| $n = 3$ | 5 | 53 | 14 |
| $n = 4$ | 34 | 49 | 74 |

Εικόνα 12. Πίνακας ημερομηνίας παράδοσης d_i (due date) με μονάδα μέτρησης το δευτερόλεπτο (second). (Θεωρείται $d_0=0$ και $d_i = d_r - d_0$, όπου d_r είναι ο χρόνος στον οποίο είναι έτοιμη προς παράδοση η εργασία).

| $n \setminus m$ | $m = 1$ | $m = 2$ | $m = 3$ |
|-----------------|---------|---------|---------|
| $n = 1$ | 13 | 13 | 34 |
| $n = 2$ | 19 | 40 | 17 |
| $n = 3$ | 8 | 54 | 12 |
| $n = 4$ | 34 | 49 | 74 |

Εικόνα 13. Πίνακας χρόνου ολοκλήρωσης c_i (completion time) με μονάδα μέτρησης το δευτερόλεπτο (second).

| $n \setminus m$ | $m = 1$ | $m = 2$ | $m = 3$ |
|-----------------|---------|---------|---------|
| $n = 1$ | 12 | 11 | 31 |
| $n = 2$ | 17 | 39 | 13 |
| $n = 3$ | 7 | 53 | 9 |
| $n = 4$ | 31 | 45 | 73 |

Εικόνα 14. Πίνακας χρόνου ετοιμασίας r_i (ready time) με μονάδα μέτρησης το δευτερόλεπτο (second).

Υπάρχει πληθώρα δυνατών συνδυασμών με τους οποίους μπορεί μια εργασία να τοποθετηθεί σε μια μηχανή. Η λειτουργία των (3) μηχανών είναι ταυτόχρονη. Μια εργασία δε μπορεί να επεξεργάζεται από (2) μηχανές, ούτε από την ίδια μηχανή (2) φορές. Παρακάτω παρατίθενται (3) παραδείγματα πιθανών συνδυασμών:

| Μηχανή (m _i) | Αλληλουχία Εργασιών (n _i) | Χρόνος επεξεργασίας (p _i) | Χρόνος Ολοκλήρωσης (c _i) | Χρόνος Ετοιμασίας (r _i) |
|--------------------------|---------------------------------------|---------------------------------------|--------------------------------------|-------------------------------------|
| 1 | 1 | p ₁ =10 | c ₁ =13 | r ₁ =12 |
| 2 | 2,3 | p _{2,3} =35+50=85 | c _{2,3} =40+54=94 | r _{2,3} =39+53=92 |
| 3 | 4 | p ₄ =70 | c ₄ =74 | r ₄ =73 |

| Μηχανή (m _i) | Αλληλουχία Εργασιών (n _i) | Χρόνος επεξεργασίας (p _i) | Χρόνος Ολοκλήρωσης (c _i) | Χρόνος Ετοιμασίας (r _i) |
|--------------------------|---------------------------------------|---------------------------------------|--------------------------------------|-------------------------------------|
| 1 | 2,3 | p _{2,3} =15+6=21 | c _{2,3} =19+8=27 | r _{2,3} =12+7=19 |
| 2 | 4 | p ₄ =40 | c ₄ =49 | r ₄ =45 |
| 3 | 1 | p ₁ =30 | c ₁ =34 | r ₁ =31 |

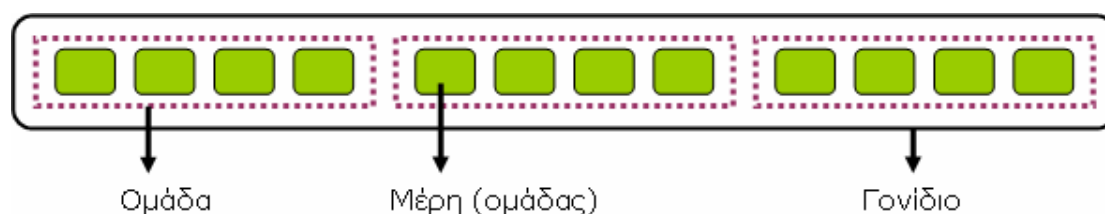
| Μηχανή (m _i) | Αλληλουχία Εργασιών (n _i) | Χρόνος επεξεργασίας (p _i) | Χρόνος Ολοκλήρωσης (c _i) | Χρόνος Ετοιμασίας (r _i) |
|--------------------------|---------------------------------------|---------------------------------------|--------------------------------------|-------------------------------------|
| 1 | 1 | p ₁ =10 | c ₁ =13 | r ₁ =12 |
| 2 | 2 | p ₂ =35 | c _{2,3} =40 | r _{2,3} =39 |
| 3 | 4,3 | p _{4,3} =70+5=75 | c _{4,3} =74+12=86 | r _{4,3} =73+9=82 |

Εικόνα 15. Παραδείγματα πιθανών συνδυασμών και υπολογισμός όλων των χρόνων με τους δεδομένους συνδυασμούς.

Στο καθένα από τα παραπάνω (3) παραδείγματα, υπολογίζεται ο χρόνος επεξεργασίας, ο χρόνος ολοκλήρωσης και ο χρόνος ετοιμασίας. Οι μαρκαρισμένοι χρόνοι είναι οι μέγιστοι. Κάθε μια από τις παραπάνω (3) περιπτώσεις αποτελεί ένα γονίδιο. Οι εργασίες αναπαρίστανται με bits και σε δυαδική μορφή. Άρα σύμφωνα με τη δυαδική μορφή:

$$\begin{array}{ccccccc}
 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^0 & & & & & & \\
 \downarrow & & \downarrow & & \downarrow & & \\
 2 & & 1 & & 0 & & = 4
 \end{array}$$

Άρα με (2) bits αναπαρίστανται και οι (4) εργασίες (με την ίδια λογική οι (8) εργασίες χρειάζονται (3) bits, ενώ οι (32) εργασίες (5) bits). Ο αριθμός των μηχανών είναι (3), άρα το γονίδιο αποτελείται από (3) ομάδες. Κάθε ομάδα αποτελείται από (4) μέρη, αφού υπάρχουν (4) διαφορετικές εργασίες που μπορούν να μπουν σε κάθε μηχανή.



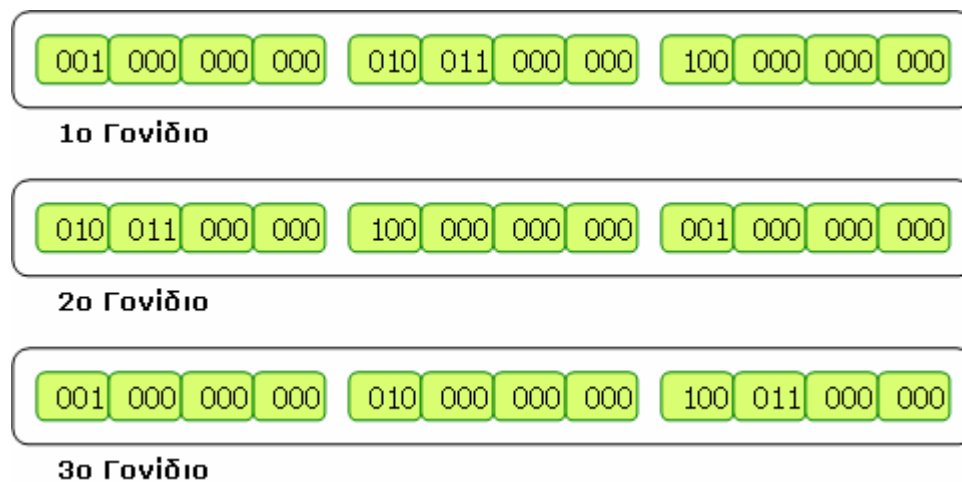
Εικόνα 16. Το γονίδιο και τα τμήματά του.

Με τη χρήση των 2 bits μπορούν να αναπαρασταθούν ακριβώς (4) εργασίες. Άρα υπάρχουν (12) πιθανές «θέσεις», στις οποίες μπορούν να μπουν οι (4) εργασίες. Όπως όμως έχει ήδη ειπωθεί, δεν επιτρέπεται μια εργασία να υπάρχει (2) φορές μέσα σε ένα γονίδιο. Άρα οι (4) θέσεις καλύπτονται με τις εργασίες, και υπόλοιπες $12 - 4 = (8)$ θέσεις θα καλυφθούν με μια εργασία που θα είναι μηδενικού χρόνου (η μηχανή δε κάνει καμία επεξεργασία), και θα ονομάζεται *μηδενική εργασία*. Η κωδικοποίηση με (2) μόνο bits δε επιτρέπει να αναπαρασταθούν παραπάνω από (4) εργασίες. Κατ' επέκταση, χρησιμοποιείται η αναπαράσταση με (3) bits, με την οποία αναπαρίσταται επίσης και η μηδενική εργασία, της οποίας η τιμή δεν είναι προσμετρήσιμη. Καλύπτει μόνο τις κενές θέσεις ώστε να εξασφαλίζεται η μοναδικότητα κάθε εργασίας μέσα στο γονίδιο (Εικόνα 17).

| Εργασία (n_i) | Αναπαράσταση σε δυαδική μορφή με 2 bits | Αναπαράσταση σε δυαδική μορφή με 3 bits |
|----------------------|---|---|
| 0 (μηδενική εργασία) | | 000 |
| 1 | 00 | 001 |
| 2 | 01 | 010 |
| 3 | 10 | 011 |
| 4 | 11 | 100 |

Εικόνα 17. Αναπαράσταση εργασιών σε δυαδική μορφή.

Έχοντας την αναπαράσταση των εργασιών σε bits είναι πλέον δυνατόν να παρουσιαστεί ένα ολόκληρο γονίδιο σε δυαδική μορφή. Παρακάτω παρατίθεται η δυαδική μορφή των (3) γονιδίων που αντιστοιχούν στο παράδειγμα (Εικόνα 18).



Εικόνα 18. Αναπαράσταση γονιδίων σε δυαδική μορφή.

Το γονίδιο είναι μια συμβολοσειρά (string), και το μήκος της συμβολοσειράς βρίσκεται από τον εξής τύπο:

$$\text{Μήκος string} = \text{Αριθμός Εργασιών} \times \text{Αριθμός bit/ Εργασία} \times \text{Αριθμός Μηχανών}$$

Στο παράδειγμα:

$$\text{Μήκος string} = 4 \times 3 \times 3 = 27$$

Για να ερευνηθεί αν κάθε ένα από τα γονίδια είναι καλό ή κακό, πρέπει να υπολογισθούν τα (3) κριτήρια: *flowtime*, *makespan* και *tardiness*. Υπολογίζονται τα (3) κριτήρια για το καθ' ένα από τα γονίδια. Με λίγα λόγια υπολογίζεται η συνάρτηση καταλληλότητας κάθε γονιδίου, και από αυτή κρίνεται το γονίδιο. Όπως έχει ήδη προαναφερθεί:

$$\text{Flowtime: } F_i = c_i - r_i$$

$$\text{Lateness: } L_i = c_i - d_i$$

$$\text{Tardiness: } T_i = \max(0, L_i)$$

$$\text{Makespan: } M_i = \max(c_i, m)$$

Άρα για το 1^ο γονίδιο (η μηδενική εργασία δεν είναι υπολογίσιμη):

$$1^{\text{η}} \text{ Εργασία: } F_1 = c_1 - r_1 = 13 - 12 = 1, \quad L_1 = c_1 - d_1 = 13 - 13 = 0$$

$$2^{\text{η}} \text{ Εργασία: } F_2 = c_2 - r_2 = 40 - 39 = 1, \quad L_2 = c_2 - d_2 = 19 - 17 = 2$$

$$3^{\text{η}} \text{ Εργασία: } F_3 = c_3 - r_3 = 54 - 53 = 1, \quad L_3 = c_3 - d_3 = 54 - 53 = 1$$

$$4^{\text{η}} \text{ Εργασία: } F_4 = c_4 - r_4 = 74 - 73 = 1, \quad L_4 = c_4 - d_4 = 74 - 74 = 0$$

$$\text{Total Flowtime: } \text{TFT} = 1 + 1 + 1 + 1 = 4$$

$$\text{Makespan: } M_i = \max(c_i) = 94 \quad (\text{για μια μηχανή})$$

$$\text{Tardiness: } T_i = \max(0, L_i) = 2$$

Για το 2^ο γονίδιο:

$$1^{\text{η}} \text{ Εργασία: } F_1 = c_1 - r_1 = 34 - 31 = 3, \quad L_1 = c_1 - d_1 = 34 - 34 = 0$$

$$2^{\text{η}} \text{ Εργασία: } F_2 = c_2 - r_2 = 19 - 12 = 7, \quad L_2 = c_2 - d_2 = 19 - 17 = 2$$

$$3^{\text{η}} \text{ Εργασία: } F_3 = c_3 - r_3 = 8 - 7 = 1, \quad L_3 = c_3 - d_3 = 8 - 5 = 3$$

$$4^{\text{η}} \text{ Εργασία: } F_4 = c_4 - r_4 = 49 - 45 = 4, \quad L_4 = c_4 - d_4 = 49 - 49 = 0$$

$$\text{Total Flowtime: } \text{TFT} = 3 + 7 + 1 + 4 = 15$$

$$\text{Makespan: } M_i = \max(c_i) = 49 \quad (\text{για μια μηχανή})$$

$$\text{Tardiness: } T_i = \max(0, L_i) = 3$$

Για το 3^ο γονίδιο:

$$1^{\text{η}} \text{ Εργασία: } F_1 = c_1 - r_1 = 13 - 12 = 1, \quad L_1 = c_1 - d_1 = 13 - 13 = 0$$

$$2^{\text{η}} \text{ Εργασία: } F_2 = c_2 - r_2 = 40 - 39 = 1, \quad L_2 = c_2 - d_2 = 40 - 42 = -2$$

$$3^{\text{η}} \text{ Εργασία: } F_3 = c_3 - r_3 = 12 - 9 = 3, \quad L_3 = c_3 - d_3 = 12 - 14 = -2$$

$$4^{\text{η}} \text{ Εργασία: } F_4 = c_4 - r_4 = 74 - 73 = 1, \quad L_4 = c_4 - d_4 = 74 - 74 = 0$$

$$\text{Total Flowtime: } \quad \text{TFT} = 1 + 1 + 3 + 1 = 6$$

$$\text{Makespan: } \quad M_i = \max(c_i) = 86 \quad (\text{για μια μηχανή})$$

$$\text{Tardiness: } \quad T_i = \max(0, L_i) = 0$$

(Τα «-» στους υπολογισμούς του L_i , σημαίνουν ότι η συγκεκριμένη εργασία ολοκληρώθηκε πριν τη προσδιορισμένη ημερομηνία παράδοσης της -due date)

Αυτό που επιδιώκεται είναι η ελαχιστοποίηση της συνάρτησης καταλληλότητας. Αυτό επιτυγχάνεται με (3) τρόπους:

1^{ος} Τρόπος:

$$\min(\max(\text{TFT}, M_i, T_i))$$

2^{ος} Τρόπος:

$$\min\left(\frac{\text{TFT} + M_i + T_i}{3}\right)$$

3^{ος} Τρόπος:

$$\min \frac{w_1 \cdot \text{TFT} + w_2 \cdot M_i + w_3 \cdot T_i}{w_1 + w_2 + w_3}$$

Για να μπορέσουν να συγκριθούν τα (3) αυτά μεγέθη μεταξύ τους και να πραγματοποιηθούν οι υπολογισμούς, πρέπει να κανονικοποιηθούν οι τιμές των μεγεθών. Με τη κανονικοποίηση θα μεταφερθούν όλες οι τιμές στο διάστημα από 0 έως 1. Εφόσον επιδιώκεται ελαχιστοποίηση των μεγεθών, το 0 είναι η βέλτιστη λύση ενώ το 1 είναι η χειρίστη.

$$\text{Κατώτερο όριο TFT} = c_{\max} - r_{\min} = 74 - 7 = 67$$

$$\text{Ανώτερο όριο TFT} = 4 \times \max(\text{TFT}) = 4 \times 67 = 268$$

Με τις προσαρμογές των τιμών στο διάστημα [0, 1], το κατώτερο όριο (και η βέλτιστη τιμή) είναι το 67, το οποίο είναι το 0, και το ανώτερο όριο είναι το 268, το οποίο είναι το 1. Έτσι στο:

1^ο γονίδιο:

$$\text{TFT} = 4 : 268 = 0.014925$$

2^ο γονίδιο:

$$\text{TFT} = 15 : 268 \approx 0.055970$$

3^ο γονίδιο:

$$\text{TFT} = 6 : 268 = 0.022388$$

Αναλόγως υπολογίζονται οι τιμές και για τα άλλα (2) κριτήρια:

| | TFT | Makespan | Tardiness |
|------------------------------|------------|-----------------|------------------|
| 1^ο γονίδιο | 0.014925 | 0.516483 | 0.666666 |
| 2^ο γονίδιο | 0.055970 | 0.269230 | 1 |
| 3^ο γονίδιο | 0.022388 | 0.472527 | 0 |

Εικόνα 19. Πίνακας αποτελεσμάτων των (3) κριτηρίων του παραδείγματος.

Όπως παρατηρείται και από την Εικόνα 19, ο γενετικός αλγόριθμος δεν εξασφαλίζει ότι οι βέλτιστες τιμές για τα TFT, Makespan και Tardiness, θα δοθούν όλες από το ίδιο γονίδιο. Στο παράδειγμα φαίνεται ότι τη βέλτιστη τιμή για το TFT την έχει το 1^ο γονίδιο, για το Makespan την έχει το 2^ο γονίδιο και για το Tardiness την έχει το 3^ο γονίδιο. Σε αυτή τη φάση αναλαμβάνει «δράση» η έκαστος αντικειμενική συνάρτηση, που με μια μέθοδο βρίσκει και συνδυάζει τις διάφορες τιμές των κριτηρίων, και εξάγει σαν αποτέλεσμα το βέλτιστο συνδυασμό τιμών-κριτηρίων και κατά συνέπεια το γονίδιο που φέρει τη βέλτιστη λύση.

7. Αποτελέσματα:

Η υλοποίηση του προβλήματος έγινε με το πρόγραμμα: MATLAB Version 7.1.0.246 (R14) Service Pack 3. Τα χαρακτηριστικά του υπολογιστή, στον οποίο έτρεξε το MATLAB είναι: Επεξεργαστής Intel[®] Core™ Duo T2250 (1,73GHz, 533MHz FSB, 2MB L2 cache), με μνήμη 2GB DDR2.

Έγινε επίλυση του προβλήματος, με χρήση πολύ-υποκειμενικού γενετικού αλγορίθμου (MOGA), εφαρμόζοντας τη μέθοδο του μέσου όρου (Mean), τη μέθοδο του ελαχίστου (Min) και τη μέθοδο της ασαφής ολοκλήρωσης Choquet (CI) στην αντικειμενική συνάρτηση. Ως παράμετροι του προβλήματος τέθηκαν οι εξής:

- Πιθανότητα διασταύρωσης (pcross): 0.8
- Πιθανότητα μετάλλαξης (pm): 0.1
- Μέγεθος πληθυσμού (popsize): 60
- Κριτήριο τερματισμού- μέγιστος πληθυσμός γενεών (maxgen): 100

Ο αριθμός των μηχανών του προβλήματος ξεκίνησε από 25 και αυξήθηκε έως 60, με βήμα 5 (25, 30, 35...), ενώ ο αριθμός εργασιών σε κάθε βήμα μηχανών, ξεκινούσε από τον αριθμό των μηχανών συν 15 και αυξανόταν έως το διπλάσιο από τον αριθμό των μηχανών, με βήμα επίσης 5.

Ο πίνακας τιμών των μεταβλητών c_i , p_i , r_i , d_i , παρέμενε σταθερός έως ότου εφαρμοστούν και οι (3) αντικειμενικές συναρτήσεις σε ένα συγκεκριμένο συνδυασμό αριθμού μηχανών και αριθμού εργασιών, ενώ όταν αυτός ο συνδυασμός αριθμών μεταβαλλόταν, μεταβάλλονταν τα μεγέθη και οι τιμές των πινάκων των μεταβλητών.

Όπως φαίνεται και στο Διάγραμμα 1, χρησιμοποιώντας την αντικειμενική συνάρτηση της μεθόδου Choquet Integral (CI), βρίσκονται λύσεις με χαμηλότερες τιμές, από ότι χρησιμοποιώντας τις αντικειμενικές συναρτήσεις των (2) άλλων μεθόδων (Mean, Min), σε κάθε περίπτωση αριθμού προβλημάτων. Δεδομένου ότι το πρόβλημα που αντιμετωπίζεται είναι ένα πρόβλημα ελαχιστοποίησης, οι όσο το δυνατό χαμηλότερες τιμές είναι καλύτερες. Πιθανή αποτυχία στην εύρεση βέλτιστης

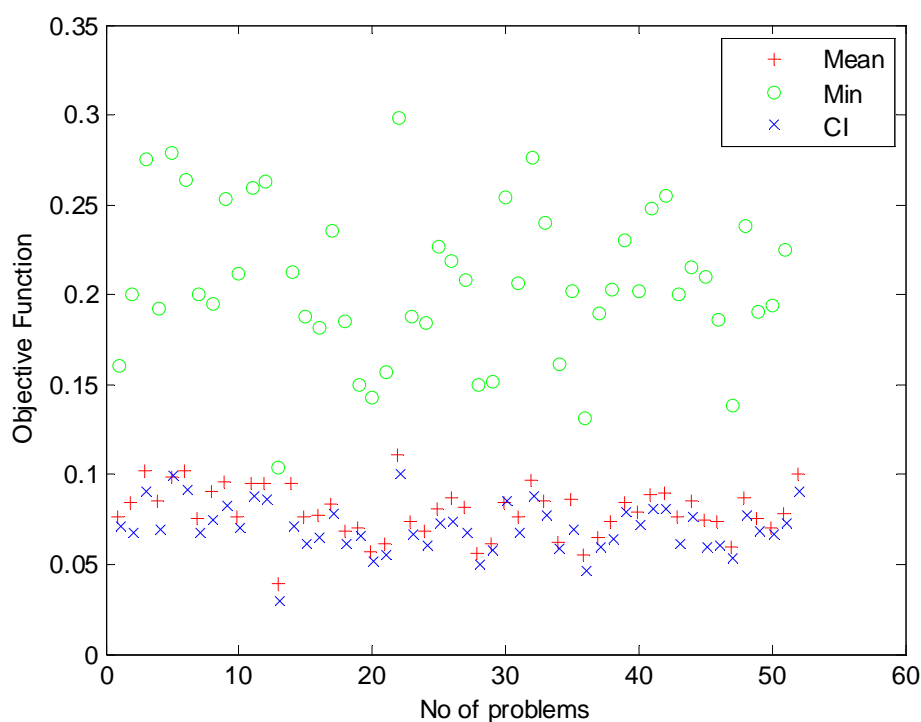
λύσης, δικαιολογείται από τη στοχαστικότητα του γενετικού αλγορίθμου, ο οποίος παγιδεύεται σε τοπικά ελάχιστα, ενώ οι τελεστές διασταύρωσης και μετάλλαξης δεν μπόρεσαν να μεταβάλλουν την ποικιλότητα του πληθυσμού, μέχρι να συμπληρωθεί ο απαραίτητος αριθμός γενιών που σημαίνει τον τερματισμό της διαδικασίας. Αυτό αποτελεί ένα γνωστό μειονέκτημα των γενετικών αλγορίθμων και για να μειωθεί η επίδραση του αλγορίθμου απαιτούνται συνεχείς εκτελέσεις του αλγορίθμου με τα ίδια δεδομένα.

Στα Διαγράμματα 2, 3 και 4 παρουσιάζονται οι τιμές των (3) κριτηρίων TFT, Tardiness, Makespan (αντιστοίχως). Είναι φανερό πως στη πλειονότητα των περιπτώσεων, η μέθοδος Choquet Integral (CI) καταφέρνει να βρει τις βέλτιστες λύσεις (δηλαδή τις χαμηλότερες λύσεις), στους διάφορους αριθμούς προβλημάτων.

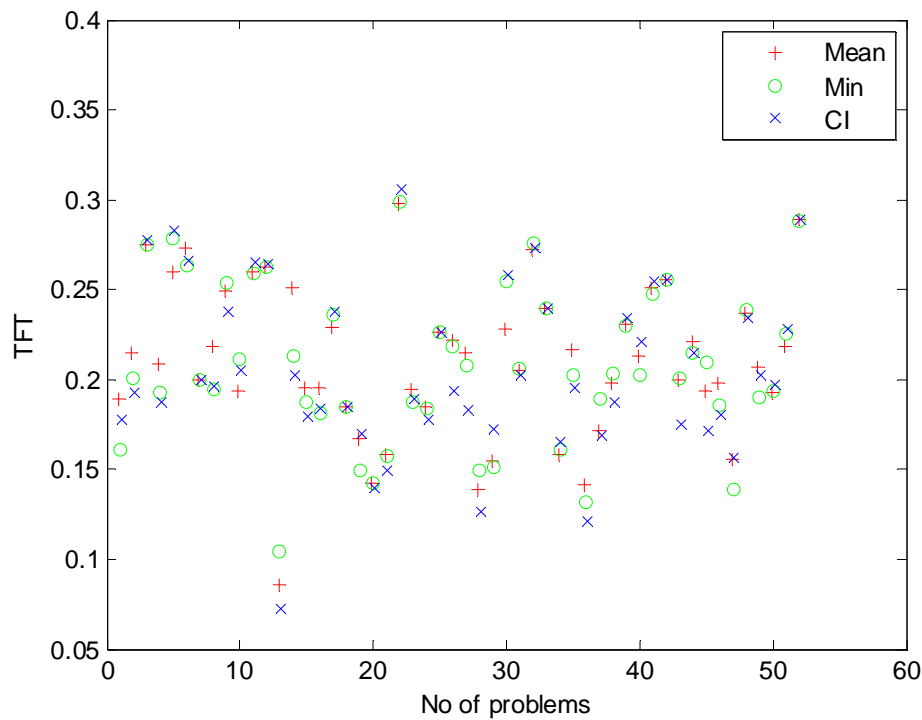
Σημαντικό είναι να τονιστεί πως υπάρχουν φορές, που οι άλλες μέθοδοι (Mean, Min), καταφέρνουν να βρουν χαμηλότερες λύσεις σε ένα από τα κριτήρια. Αυτό όμως δεν έχει σαν συνέπεια την καλύτερη απόδοση της αντικειμενικής συνάρτησης. Αυτό συμβαίνει γιατί, όπως έχει ήδη διατυπωθεί, όταν μια ιδιαίτερα χαμηλή τιμή σε ένα κριτήριο συνδυάζεται με υψηλές τιμές στα άλλα κριτήρια, δεν είναι δεδομένο ότι η αντικειμενική συνάρτηση βρίσκει τη βέλτιστη τιμή. Η μέθοδος Choquet Integral όμως, καταφέρνει βρίσκοντας τις χαμηλότερες (ή σχετικά χαμηλές) τιμές, να βρίσκει τη βέλτιστη λύση. Αυτό εξασφαλίζεται από την αλληλεπίδραση των κριτηρίων μεταξύ τους αλλά και από τη μέθοδο του αλγορίθμου, να μην αποκλείει τα σχετικά καλά γονίδια, αλλά να τα μεταφέρει σε επόμενες γενιές, να τα μεταλλάσσει και να έχει τα βέλτιστα αποτελέσματα (σε αντίθεση με άλλους που συγκρατούν γονίδια μόνο εάν διαθέτουν ένα ή περισσότερα ιδιαίτερα χαμηλά κριτήρια).

Στο Διάγραμμα 5 παρουσιάζεται για κάθε αριθμό προβλημάτων, ο ιδιαίτερα αυξημένος χώρος αναζήτησης των λύσεων (searching space). Στον άξονα των x, όπου βρίσκεται ο αριθμός των προβλημάτων, απεικονίζεται ο χώρος των λύσεων που έχει ελεγχθεί και μέσα στον οποίο έχει βρεθεί η βέλτιστη λύση (searched space- searched space \subseteq searching space). Ο χώρος που έχει ελεγχθεί, φαίνεται σχεδόν πάνω στον x-άξονα, και αυτό γιατί είναι πάρα πολύ μικρός σε σχέση με όλο το χώρο αναζήτησης (της τάξεως του 1.49%).

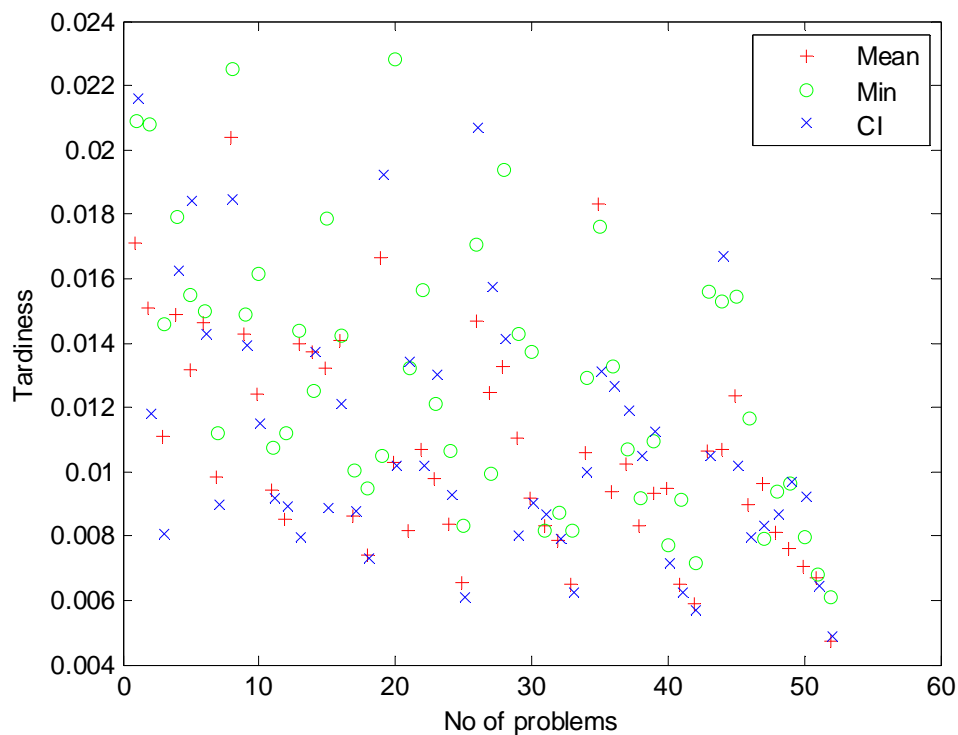
Τέλος, στο Διάγραμμα 6 απεικονίζεται ο υπολογιστικός χρόνος που απαιτείται από κάθε μέθοδο, για την επίλυση του προβλήματος. Η προτεινόμενη μέθοδος Choquet Integral, έχει μεγαλύτερες απαιτήσεις σε υπολογιστικό χρόνο, σε σχέση με τις άλλες (2) μεθόδους, γεγονός που αποτελεί μειονέκτημα, το οποίο όμως μπορεί να κριθεί αμελητέο, αν το αντισταθμίσουμε με τη γενική απόδοση της μεθόδου.



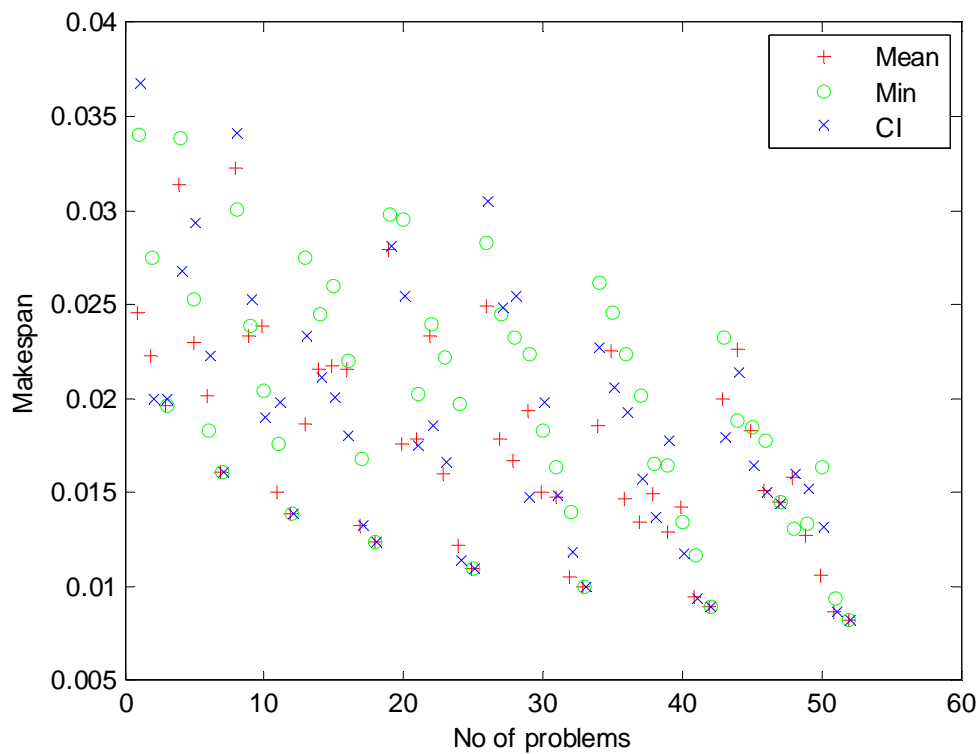
Διάγραμμα 1. Συγκριτικό Διάγραμμα των (3) μεθόδων, που παρουσιάζει την απόδοση των αντικειμενικών συναρτήσεων σε συσχέτισμό με τον αριθμό των προβλημάτων.



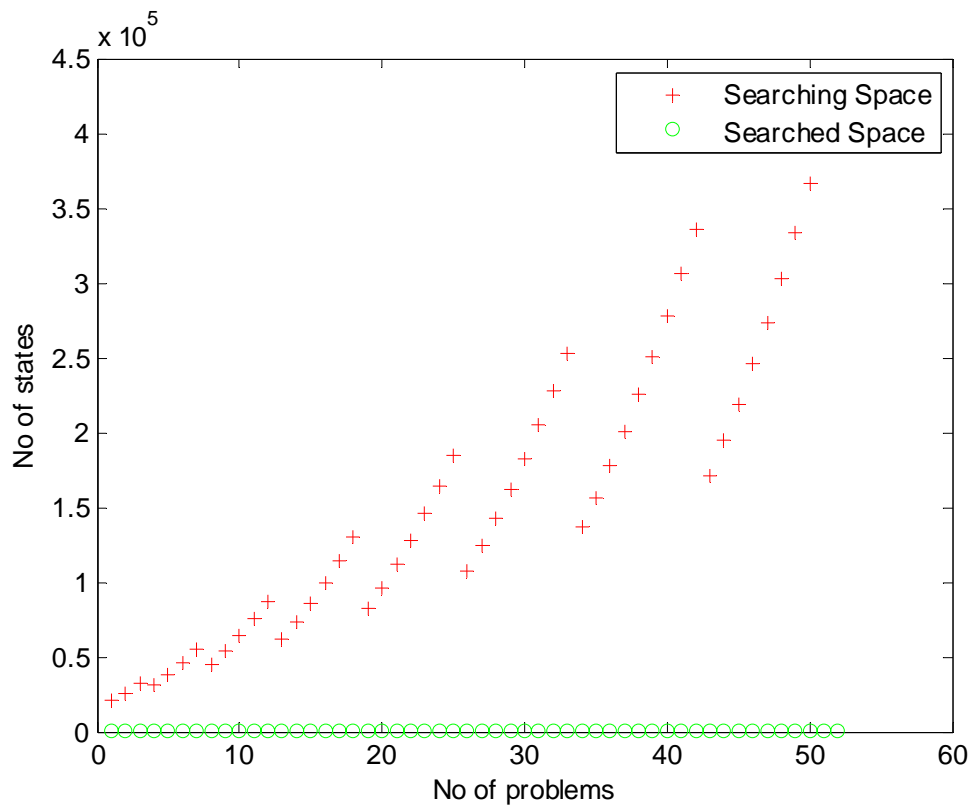
Διάγραμμα 2. Συγκριτικό διάγραμμα των (3) μεθόδων, που παρουσιάζει τις τιμές του κριτηρίου TFT σε συσχέτισμό με τον αριθμό των προβλημάτων.



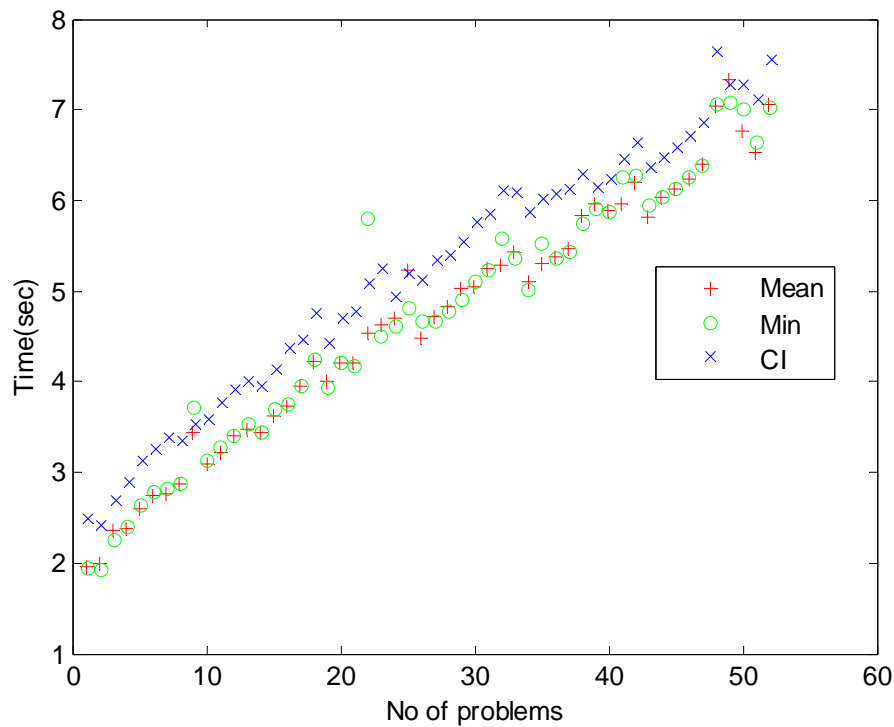
Διάγραμμα 3. Συγκριτικό διάγραμμα των (3) μεθόδων, που παρουσιάζει τις τιμές του κριτηρίου Tardiness σε συσχέτισμό με τον αριθμό των προβλημάτων.



Διάγραμμα 4. Συγκριτικό διάγραμμα των (3) μεθόδων, που παρουσιάζει τις τιμές του κριτηρίου Makespan σε συσχέτισμό με τον αριθμό των προβλημάτων.



Διάγραμμα 5. Συγκριτικό διάγραμμα που παρουσιάζει τον αριθμό των λύσεων σε συσχέτισμό με τον αριθμό των προβλημάτων.



Διάγραμμα 6. Συγκριτικό διάγραμμα των (3) μεθόδων, που παρουσιάζει τον υπολογιστικό χρόνο σε συσχέτιση με τον αριθμό των προβλημάτων.

Παρακάτω παρατίθεται ο πίνακας των αποτελεσμάτων για κάθε περίπτωση αριθμού μηχανών, αριθμού εργασιών και αντικειμενικής συνάρτησης. Για να διευκρινιστεί η αντιστοιχία των αντικειμενικών συναρτήσεων:

Objective Function 1: Μέθοδος Μέσου Όρου

Objective Function 2: Μέθοδος Ελαχιστοποίησης Μεγίστου

Objective Function 3: Μέθοδος Choquet Integral

| Machines | Tasks | Obj.Function | TFT | Tard | Makespan | Opt.Solution |
|----------|-------|--------------|-------------|------------|------------|--------------|
| 25 | 40 | 1 | 0,10341152 | 0,01452185 | 0,03574046 | 0,05122461 |
| 25 | 40 | 2 | 0,09685031 | 0,023795 | 0,02803978 | 0,09685031 |
| 25 | 40 | 3 | 0,121170952 | 0,01144363 | 0,02710762 | 0,04686806 |
| 25 | 45 | 1 | 0,188589214 | 0,01319809 | 0,02263011 | 0,07480581 |
| 25 | 45 | 2 | 0,192139705 | 0,01704969 | 0,03910207 | 0,1921397 |
| 25 | 45 | 3 | 0,162974673 | 0,01073523 | 0,02354212 | 0,05845617 |
| 25 | 50 | 1 | 0,19829131 | 0,01036339 | 0,01949196 | 0,07604889 |
| 25 | 50 | 2 | 0,197637194 | 0,01505135 | 0,01949196 | 0,19763719 |
| 25 | 50 | 3 | 0,19829131 | 0,00941788 | 0,01949196 | 0,06769176 |
| 30 | 45 | 1 | 0,161483771 | 0,01789748 | 0,03190372 | 0,07042832 |
| 30 | 45 | 2 | 0,159038955 | 0,03277057 | 0,0391056 | 0,15903895 |
| 30 | 45 | 3 | 0,155761132 | 0,02324083 | 0,03637178 | 0,06509787 |
| 30 | 50 | 1 | 0,214184405 | 0,01858118 | 0,02644937 | 0,08640499 |
| 30 | 50 | 2 | 0,197018278 | 0,02105041 | 0,02676332 | 0,19701828 |
| 30 | 50 | 3 | 0,212388162 | 0,01575358 | 0,02848464 | 0,07678092 |
| 30 | 55 | 1 | 0,158209703 | 0,01185046 | 0,018005 | 0,06268839 |
| 30 | 55 | 2 | 0,157435902 | 0,01134345 | 0,018005 | 0,1574359 |
| 30 | 55 | 3 | 0,156388089 | 0,01280503 | 0,02102084 | 0,05719448 |
| 30 | 60 | 1 | 0,243740858 | 0,01002824 | 0,01627752 | 0,09001554 |
| 30 | 60 | 2 | 0,243740858 | 0,01213495 | 0,01627752 | 0,24374086 |
| 30 | 60 | 3 | 0,243740858 | 0,01058398 | 0,01627752 | 0,08144201 |
| 35 | 50 | 1 | 0,184325069 | 0,01700746 | 0,02968512 | 0,07700588 |
| 35 | 50 | 2 | 0,186138901 | 0,0193029 | 0,03030954 | 0,1861389 |
| 35 | 50 | 3 | 0,170260704 | 0,0176698 | 0,03704545 | 0,06654718 |
| 35 | 55 | 1 | 0,153054065 | 0,01116948 | 0,0214637 | 0,06189575 |
| 35 | 55 | 2 | 0,137050524 | 0,01085857 | 0,02041734 | 0,13705052 |
| 35 | 55 | 3 | 0,158726971 | 0,0132755 | 0,02207169 | 0,05831833 |
| 35 | 60 | 1 | 0,251518124 | 0,013614 | 0,0225922 | 0,09590811 |
| 35 | 60 | 2 | 0,261187989 | 0,01434804 | 0,02452486 | 0,26118799 |
| 35 | 60 | 3 | 0,270386231 | 0,01298484 | 0,01662098 | 0,09078704 |
| 35 | 65 | 1 | 0,141818429 | 0,01000049 | 0,01566751 | 0,05582881 |
| 35 | 65 | 2 | 0,143887781 | 0,01335791 | 0,02098878 | 0,14388778 |
| 35 | 65 | 3 | 0,139635867 | 0,00932524 | 0,01591112 | 0,04947217 |
| 35 | 70 | 1 | 0,235618539 | 0,00819994 | 0,01376846 | 0,08586231 |
| 35 | 70 | 2 | 0,235218744 | 0,01124523 | 0,01401108 | 0,23521874 |
| 35 | 70 | 3 | 0,235618539 | 0,00736382 | 0,01376846 | 0,07686498 |
| 40 | 55 | 1 | 0,214039069 | 0,02069814 | 0,03193521 | 0,08889081 |
| 40 | 55 | 2 | 0,204533452 | 0,01750873 | 0,02615295 | 0,20453345 |
| 40 | 55 | 3 | 0,199919627 | 0,01177591 | 0,02687362 | 0,07063466 |
| 40 | 60 | 1 | 0,148558442 | 0,01093635 | 0,01720267 | 0,05889916 |
| 40 | 60 | 2 | 0,179615431 | 0,01367068 | 0,02710406 | 0,17961543 |
| 40 | 60 | 3 | 0,189529416 | 0,01310812 | 0,01874876 | 0,06693701 |
| 40 | 65 | 1 | 0,289443365 | 0,0113521 | 0,02168707 | 0,10749418 |
| 40 | 65 | 2 | 0,25327856 | 0,01899138 | 0,02733724 | 0,25327856 |
| 40 | 65 | 3 | 0,247158981 | 0,01406323 | 0,02651975 | 0,085985 |
| 40 | 70 | 1 | 0,258087496 | 0,01182774 | 0,01685992 | 0,09559172 |
| 40 | 70 | 2 | 0,263377376 | 0,01504851 | 0,02143184 | 0,26337738 |
| 40 | 70 | 3 | 0,250089167 | 0,00872672 | 0,02264895 | 0,08336301 |
| 40 | 75 | 1 | 0,205416812 | 0,00794783 | 0,01327674 | 0,07554713 |
| 40 | 75 | 2 | 0,2083244 | 0,01029683 | 0,01566248 | 0,2083244 |
| 40 | 75 | 3 | 0,209538563 | 0,01114859 | 0,01327674 | 0,07100608 |
| 40 | 80 | 1 | 0,225563906 | 0,00758094 | 0,01228383 | 0,08180956 |
| 40 | 80 | 2 | 0,225563906 | 0,00915429 | 0,01228383 | 0,22556391 |

| | | | | | | |
|----|-----|---|-------------|------------|------------|------------|
| 40 | 80 | 3 | 0,226534184 | 0,00705635 | 0,01228383 | 0,0737361 |
| 45 | 60 | 1 | 0,191778104 | 0,01379286 | 0,02606727 | 0,07721274 |
| 45 | 60 | 2 | 0,173069548 | 0,01469847 | 0,02551127 | 0,17306955 |
| 45 | 60 | 3 | 0,153644534 | 0,01759438 | 0,02668416 | 0,05986379 |
| 45 | 65 | 1 | 0,170312063 | 0,01583885 | 0,02710593 | 0,07108561 |
| 45 | 65 | 2 | 0,185802786 | 0,01847006 | 0,02562018 | 0,18580279 |
| 45 | 65 | 3 | 0,171015217 | 0,01511424 | 0,02472925 | 0,06342293 |
| 45 | 70 | 1 | 0,173947226 | 0,01042732 | 0,01423212 | 0,06620222 |
| 45 | 70 | 2 | 0,186193483 | 0,01843976 | 0,02646307 | 0,18619348 |
| 45 | 70 | 3 | 0,156188029 | 0,00990204 | 0,01981847 | 0,05537447 |
| 45 | 75 | 1 | 0,166465072 | 0,01077681 | 0,0183139 | 0,06518526 |
| 45 | 75 | 2 | 0,183945173 | 0,01156128 | 0,01699595 | 0,18394517 |
| 45 | 75 | 3 | 0,193104541 | 0,0100654 | 0,01522099 | 0,06580204 |
| 45 | 80 | 1 | 0,212661835 | 0,00948417 | 0,01499763 | 0,07904788 |
| 45 | 80 | 2 | 0,216333992 | 0,00839255 | 0,01665253 | 0,21633399 |
| 45 | 80 | 3 | 0,235276444 | 0,00906255 | 0,01548581 | 0,07795444 |
| 45 | 85 | 1 | 0,24220048 | 0,00795 | 0,01450672 | 0,08821907 |
| 45 | 85 | 2 | 0,24002129 | 0,01409875 | 0,01903234 | 0,24002129 |
| 45 | 85 | 3 | 0,24211741 | 0,0085694 | 0,01955943 | 0,08039221 |
| 45 | 90 | 1 | 0,226967584 | 0,00725945 | 0,01089018 | 0,08170574 |
| 45 | 90 | 2 | 0,226967584 | 0,00723261 | 0,01089018 | 0,22696758 |
| 45 | 90 | 3 | 0,227298609 | 0,00679649 | 0,01085302 | 0,07359617 |
| 50 | 65 | 1 | 0,187990915 | 0,01652619 | 0,02339437 | 0,07597049 |
| 50 | 65 | 2 | 0,182091514 | 0,01830287 | 0,02789462 | 0,18209151 |
| 50 | 65 | 3 | 0,194969117 | 0,01558475 | 0,0236997 | 0,07069845 |
| 50 | 70 | 1 | 0,149117675 | 0,01397338 | 0,02400873 | 0,0623666 |
| 50 | 70 | 2 | 0,153738406 | 0,01605241 | 0,02554162 | 0,15373841 |
| 50 | 70 | 3 | 0,160343148 | 0,01280258 | 0,02126202 | 0,05841826 |
| 50 | 75 | 1 | 0,173804368 | 0,00727342 | 0,02091459 | 0,06733079 |
| 50 | 75 | 2 | 0,148524502 | 0,01292831 | 0,01943686 | 0,1485245 |
| 50 | 75 | 3 | 0,159659064 | 0,01192957 | 0,01785519 | 0,05719652 |
| 50 | 80 | 1 | 0,123616284 | 0,00765215 | 0,0166991 | 0,04932251 |
| 50 | 80 | 2 | 0,149898849 | 0,01505085 | 0,0220204 | 0,14989885 |
| 50 | 80 | 3 | 0,109171336 | 0,0127468 | 0,01647537 | 0,04227073 |
| 50 | 85 | 1 | 0,164158105 | 0,00995155 | 0,01294057 | 0,06235007 |
| 50 | 85 | 2 | 0,160152623 | 0,01228651 | 0,01571163 | 0,16015262 |
| 50 | 85 | 3 | 0,144572882 | 0,01148316 | 0,01663977 | 0,05223514 |
| 50 | 90 | 1 | 0,228712413 | 0,00804798 | 0,01162817 | 0,08279619 |
| 50 | 90 | 2 | 0,228811894 | 0,01274314 | 0,01580674 | 0,22881189 |
| 50 | 90 | 3 | 0,217204876 | 0,00811953 | 0,01272542 | 0,07158208 |
| 50 | 95 | 1 | 0,202345226 | 0,00567828 | 0,01041868 | 0,07281406 |
| 50 | 95 | 2 | 0,20930448 | 0,00763481 | 0,01048156 | 0,20930448 |
| 50 | 95 | 3 | 0,212867098 | 0,00544167 | 0,01048156 | 0,06847568 |
| 50 | 100 | 1 | 0,226225934 | 0,00619655 | 0,00997659 | 0,08079969 |
| 50 | 100 | 2 | 0,225523402 | 0,00692466 | 0,00997659 | 0,2255234 |
| 50 | 100 | 3 | 0,225541568 | 0,00629836 | 0,00997659 | 0,07265984 |
| 55 | 70 | 1 | 0,204716116 | 0,01296375 | 0,02180568 | 0,07982851 |
| 55 | 70 | 2 | 0,223248265 | 0,01935639 | 0,02465359 | 0,22324827 |
| 55 | 70 | 3 | 0,216496372 | 0,01019895 | 0,02337184 | 0,07419584 |
| 55 | 75 | 1 | 0,187122387 | 0,01359707 | 0,01998327 | 0,07356758 |
| 55 | 75 | 2 | 0,214165387 | 0,0178247 | 0,02514163 | 0,21416539 |
| 55 | 75 | 3 | 0,196156368 | 0,01800699 | 0,02329893 | 0,07229852 |
| 55 | 80 | 1 | 0,13091169 | 0,00976523 | 0,01896034 | 0,05321242 |
| 55 | 80 | 2 | 0,148332717 | 0,01165886 | 0,01683101 | 0,14833272 |

| | | | | | | |
|----|-----|---|-------------|------------|------------|------------|
| 55 | 80 | 3 | 0,166698515 | 0,00801708 | 0,0220688 | 0,05786978 |
| 55 | 85 | 1 | 0,25186741 | 0,01117746 | 0,01949094 | 0,0941786 |
| 55 | 85 | 2 | 0,241667936 | 0,01265747 | 0,0167789 | 0,24166794 |
| 55 | 85 | 3 | 0,253877155 | 0,01018166 | 0,0188711 | 0,08468062 |
| 55 | 90 | 1 | 0,21292046 | 0,00913362 | 0,01351372 | 0,0785226 |
| 55 | 90 | 2 | 0,204985191 | 0,01138532 | 0,01716845 | 0,20498519 |
| 55 | 90 | 3 | 0,217744186 | 0,00956578 | 0,01513656 | 0,07291063 |
| 55 | 95 | 1 | 0,208182437 | 0,00827996 | 0,01354795 | 0,07667012 |
| 55 | 95 | 2 | 0,192236448 | 0,00731178 | 0,01614724 | 0,19223645 |
| 55 | 95 | 3 | 0,202497799 | 0,00759759 | 0,01482166 | 0,06722351 |
| 55 | 100 | 1 | 0,204094432 | 0,00878718 | 0,01293223 | 0,07527128 |
| 55 | 100 | 2 | 0,197275721 | 0,0078239 | 0,01315041 | 0,19727572 |
| 55 | 100 | 3 | 0,196763222 | 0,00834089 | 0,0116725 | 0,06540064 |
| 55 | 105 | 1 | 0,229198527 | 0,0067326 | 0,00951463 | 0,08181525 |
| 55 | 105 | 2 | 0,236439612 | 0,00717089 | 0,0108547 | 0,23643961 |
| 55 | 105 | 3 | 0,237414248 | 0,00665794 | 0,01275891 | 0,07686099 |
| 55 | 110 | 1 | 0,221819613 | 0,00622891 | 0,00903509 | 0,07902787 |
| 55 | 110 | 2 | 0,221674284 | 0,0063585 | 0,00903509 | 0,22167428 |
| 55 | 110 | 3 | 0,221674284 | 0,0060758 | 0,00903509 | 0,07122883 |
| 60 | 75 | 1 | 0,217042149 | 0,01320995 | 0,02208875 | 0,08411362 |
| 60 | 75 | 2 | 0,207730774 | 0,01580262 | 0,02566823 | 0,20773077 |
| 60 | 75 | 3 | 0,205433471 | 0,00968646 | 0,01851558 | 0,06982322 |
| 60 | 80 | 1 | 0,231447056 | 0,01145072 | 0,01965551 | 0,08751776 |
| 60 | 80 | 2 | 0,22736349 | 0,01308011 | 0,02089343 | 0,22736349 |
| 60 | 80 | 3 | 0,234093702 | 0,01081313 | 0,01977726 | 0,07923156 |
| 60 | 85 | 1 | 0,194823675 | 0,01372987 | 0,01937921 | 0,07597759 |
| 60 | 85 | 2 | 0,210438748 | 0,01327668 | 0,02035028 | 0,21043875 |
| 60 | 85 | 3 | 0,19755603 | 0,00880811 | 0,01939745 | 0,06712678 |
| 60 | 90 | 1 | 0,238705473 | 0,00799561 | 0,01365091 | 0,086784 |
| 60 | 90 | 2 | 0,239045695 | 0,01484016 | 0,02092126 | 0,2390457 |
| 60 | 90 | 3 | 0,249633211 | 0,01012861 | 0,01735525 | 0,08313625 |
| 60 | 95 | 1 | 0,229324907 | 0,01362635 | 0,01649826 | 0,08648317 |
| 60 | 95 | 2 | 0,212524582 | 0,01484415 | 0,01783201 | 0,21252458 |
| 60 | 95 | 3 | 0,213155753 | 0,01176254 | 0,01810186 | 0,07319479 |
| 60 | 100 | 1 | 0,184523181 | 0,00885951 | 0,01224386 | 0,06854218 |
| 60 | 100 | 2 | 0,194626426 | 0,00928401 | 0,0159737 | 0,19462643 |
| 60 | 100 | 3 | 0,190912084 | 0,0071113 | 0,01260137 | 0,06312994 |
| 60 | 105 | 1 | 0,208054058 | 0,00642876 | 0,00920887 | 0,0745639 |
| 60 | 105 | 2 | 0,199056224 | 0,0096052 | 0,01598004 | 0,19905622 |
| 60 | 105 | 3 | 0,194054386 | 0,00774166 | 0,01256711 | 0,06440755 |
| 60 | 110 | 1 | 0,216399149 | 0,00617748 | 0,00994316 | 0,07750659 |
| 60 | 110 | 2 | 0,228990369 | 0,00933319 | 0,01336142 | 0,22899037 |
| 60 | 110 | 3 | 0,206444061 | 0,00619504 | 0,01283123 | 0,06733154 |
| 60 | 115 | 1 | 0,268032338 | 0,00604725 | 0,00924209 | 0,09444056 |
| 60 | 115 | 2 | 0,269843738 | 0,00682201 | 0,01126791 | 0,26984374 |
| 60 | 115 | 3 | 0,268635572 | 0,00590732 | 0,00929605 | 0,08526799 |
| 60 | 120 | 1 | 0,264282081 | 0,0061364 | 0,0082862 | 0,09290156 |
| 60 | 120 | 2 | 0,265167037 | 0,00556566 | 0,0082862 | 0,26516704 |
| 60 | 120 | 3 | 0,266157084 | 0,00492636 | 0,0082862 | 0,08383315 |

Εικόνα 20. Πίνακας αποτελεσμάτων.

8. Υπολογιστικό Κόστος:

Το πρόβλημα έχει πολυπλοκότητα:

$$m * \sum_{i=1}^2 \binom{n}{i}$$

όπου m: αριθμός μηχανών,

n: αριθμός εργασιών.

Το άθροισμα παίρνει τιμές $i = (1,2)$. Αυτό βοηθάει ώστε να αυξάνεται η διασπορά των εργασιών στις μηχανές. Πιο συγκεκριμένα, δεδομένου ότι υπάρχει αυξημένος αριθμός μηδενικών εργασιών, οι μηχανές έχουν ανώτατο όριο επεξεργασίας εργασιών το (2). Έτσι μειώνεται ο συνωστισμός πολλών εργασιών σε μερικές μηχανές, δηλαδή μείωση πιθανών καθυστερήσεων, και ο συνωστισμός μηδενικών εργασιών σε άλλες μηχανές, δηλαδή αδράνεια των μηχανών.

Η πολυπλοκότητα ενός προβλήματος είναι η πολυπλοκότητα του αλγορίθμου που επιλύει αυτό το πρόβλημα. Πολλά δύσκολα υπολογιστικά προβλήματα όπως αυτά του σχεδιασμού, του χρονοδιαγραμματισμού και τις συνδυαστικής βελτιστοποίησης, θεωρούνται σα παραδείγματα των όπως αποκαλούνται «προβλήματα ικανοποίησης περιορισμών». Τα προβλήματα ικανοποίησης περιορισμών συνίστανται από ένα σετ μεταβλητών, κάθε ένα έχοντας ένα πεπερασμένο σετ από μια περιοχή τιμών, μαζί με ένα σετ από λογικές σχέσεις (ή «περιορισμούς») ανάμεσα στις μεταβλητές που απαιτείται να υφίστανται ταυτόχρονα. Μια λύση καθορίζεται από ένα ολοκληρωμένο σετ από κατανομές μεταβλητών/τιμών, τέτοιο ώστε κάθε μεταβλητή να έχει μια τιμή, καμία μεταβλητή να μη προσδιορίζεται από αντικρουόμενες τιμές και όλοι οι περιορισμοί να ικανοποιούνται. Σε ένα τέτοιο πρόβλημα ικανοποίησης περιορισμών, υπάρχει συχνά ένας μεγάλος αριθμός κοινών μη- λύσεων. Κάποιος βρίσκει, για παράδειγμα, ότι συγκεκριμένοι συνδυασμοί από κατανομές τιμών σε υποτιμήματα μεταβλητών είναι αντιφατικοί (παραβιάζουν ένα ή περισσότερους περιορισμούς) και δε μπορούν γι' αυτό να συμμετάσχουν σε καμία λύση. Αυτά οι κοινές μεταβλητές (πολλές μη- λύσεις να μοιράζονται τον ίδιο πρόγονο ο οποίος είναι αντιφατικός) μπορούν να επωφεληθούν

και να εστιάσουν την αναζήτηση (πάνω τους) για μια λύση. Έτσι, ένας κλασσικά δομημένος αλγόριθμος αναζήτησης, μπορεί να βρει μια λύση για ένα πρόβλημα ικανοποίησης περιορισμών σε λιγότερα βήματα, από αυτά που απαιτούνται από μια μη-δομημένο αναζήτηση, αποφεύγοντας περιοχές του χώρου αναζήτησης που εγγυημένα δεν υπάρχουν λύσεις. Η ιδέα- κλειδί είναι ότι κάποιος μπορεί να πετύχει ολοκληρωμένες λύσεις σε ένα πρόβλημα ικανοποίησης περιορισμών, εκτείνοντας συστηματικά ατελής λύσεις, για παράδειγμα ανάθεση μεταβλητών/ τιμών που εφαρμόζονται μόνο σε ένα υποτμήμα των μεταβλητών του προβλήματος. Όχι όλες οι ατελής λύσεις είναι όμως το ίδιο επιθυμητές. Μια ατελής λύση είναι «καλή» αν είναι σύμφωνη με όλους τους περιορισμούς με τους οποίους μπορεί να δοκιμαστεί (μπορεί όμως και να μην είναι η καλύτερη- αυτό δεν είναι αρνητικό). Μια ατελής λύση είναι «όχι καλή» αν παραβιάζει έναν ή περισσότερους περιορισμούς. Οι πολύπλοκοι αλγόριθμοι αναζήτησης δουλεύουν με επαυξηντικά εκτεινόμενες καλές ατελής λύσεις και συστηματικά τερματίζουν τις όχι- καλές ατελής λύσεις. Με το να διατηρεί καλές ατελής λύσεις εκτός από τις φαινομενικά βέλτιστες, καταφέρνει να αυξάνει τη ποικιλία των κάθε γενιών, αλλά επίσης και να αποφεύγει τοπικά μέγιστα που αποτελούν συνήθως οι φαινομενικά βέλτιστες λύσεις (οι φαινομενικά βέλτιστες λύσεις έχουν πολύ υψηλές τιμές σε κάποια κριτήρια, αλλά πολύ χαμηλές σε κάποια άλλα- τοπικά μέγιστα).

Ο πολύ- υποκειμενικός γενετικός αλγόριθμος (MOGA) με χρήση της μεθόδου Choquet καταφέρνει, σε προγραμματιστικά προβλήματα ασυσχέτιστων μηχανών (R_mSP) με πολύ μεγάλη πολυπλοκότητα, και κατά συνέπεια σε ένα τεράστιο χώρο αναζήτησης (searching space), να αναζητά και να βρίσκει τη βέλτιστη λύση (καλύτερη από των άλλων συναρτήσεων) σε ένα πάρα πολύ μικρό κομμάτι του χώρου (searched space) της τάξεως του 1.49% στην χειρότερη περίπτωση, το οποίο καταφέρνει να βρει εφαρμόζοντας τη προαναφερθείσα μέθοδο. Είναι σημαντικό να διατυπωθεί η διαπίστωση, πως όσο πιο πολύπλοκο το πρόβλημα τόσο πιο μεγάλη η αποτελεσματικότητα και η υπεροχή του της μεθόδου Choquet, σε σχέση με τη δράση των υπολοίπων συναρτήσεων.

9. Συμπεράσματα:

Το προγραμματιστικό πρόβλημα των ασυσχέτιστων μηχανών λύθηκε με τη χρήση του γενετικού αλγορίθμου, και την ενσωμάτωση σε αυτόν της ασαφής ολοκλήρωσης Choquet. Αφού βρέθηκαν οι κατάλληλες παράμετροι όσο αναφορά το γενετικό, όπως πιθανότητα ανταλλαγής γονιδίων, πιθανότητα μετάλλαξης, κ.τ.λ., μοντελοποιήθηκε το πρόβλημα πάνω στη μέθοδο Choquet, συναρτήσεως των συναρτήσεων αλληλεπίδρασης των κριτηρίων, τα οποία έχουν ορισθεί ως: κόστος κατασκευής (makespan), μέγιστος χρόνος καθυστέρησης (total tardiness) και συνολικός χρόνος ροής (total flow time).

Έγινε συγκριτική μελέτη του προβλήματος κρατώντας σταθερές τις παραμέτρους του γενετικού, και εφαρμόζοντας διαφορετικές αντικειμενικές συναρτήσεις κάθε φορά. Η σύγκριση έγινε μεταξύ της μεθόδου Choquet, της μεθόδου ελαχίστου και της μεθόδου μέσου όρου. Η πολυπλοκότητα του προβλήματος είναι πολύ μεγάλη και βρίσκεται για κάθε περίπτωση, συναρτήσεως του αριθμού των μηχανών- m και του αριθμού των εργασιών- n , μέσω του τύπου: $m * \sum_{i=1}^2 \binom{n}{i}$.

Όπως φαίνεται και από τα αποτελέσματα, οι βέλτιστες λύσεις προσδιορίζονται από τη μέθοδο Choquet, όπου εφαρμόζεται η αλληλεπίδραση των κριτηρίων. Η πλειοψηφία των καλύτερων αποτελεσμάτων που διαφαίνονται από τα διαγράμματα, ανήκουν στη προαναφερθείσα μέθοδο, και ένα μικρό ποσοστό αποτυχίας (βρίσκει μια πάρα πολύ καλή τιμή αλλά όχι τη βέλτιστη δυνατή) που μπορεί να παρατηρηθεί, δικαιολογείται από τη στοχαστικότητα του γενετικού αλγορίθμου. Πρέπει να τονισθεί, ότι εκτός από τη βέλτιστη λύση σε σημείο, εξίσου σημαντικό, αν όχι σημαντικότερο, ρόλο παίζουν οι τιμές που λαμβάνουν τα κριτήρια, καθ' ότι δε πρέπει να αφηφάται ότι σκοπός της βελτιστοποίησης είναι η μεγαλύτερη δυνατή ελαχιστοποίηση των τιμών των (3) κριτηρίων.

Παρά τη μεγάλη πολυπλοκότητα του προβλήματος, ο αλγόριθμος καταφέρνει να βρει τη καλύτερη λύση, σε μικρό χρονικό διάστημα. Αυτό συμβαίνει εξαιτίας του ότι ο αλγόριθμος δεν εξετάζει όλο το χώρο αναζήτησης, αλλά ένα μικρό τμήμα αυτού. Η πολυπλοκότητα του προβλήματος έχει συνέπειες στον υπολογιστικό χρόνο

του προβλήματος (ελαφρώς αυξημένος υπολογιστικός χρόνος), όμως η ανωτερότητα του σε όλους τους άλλους τομείς, κάνει τη συγκεκριμένη επιλογή αλγορίθμου-αντικειμενικής, τη καλύτερη δυνατή.

Η μέθοδος αυτή έχει τα μειονεκτήματα της αυξημένης πολυπλοκότητας και του αυξημένου υπολογιστικού χρόνου. Όταν όμως κρίνεται σε προβλήματα του πραγματικού κόσμου, όπου η πολυπλοκότητά τους είναι ήδη ιδιαίτερα αυξημένη και η εύρεση της σίγουρα βέλτιστης λύσης κρίνεται πολύ σημαντική, τότε η χρήση του MOGA σε συνδυασμό με τη μέθοδο του Choquet Integral, είναι η καλύτερη.

Ο αλγόριθμος αυτός μπορεί να χρησιμοποιηθεί σε πραγματικά προβλήματα και να δώσει έγκυρα αποτελέσματα για ένα προγραμματιστικό πρόβλημα μεγάλης πολυπλοκότητας. Με δεδομένα και παραμέτρους μιας βιομηχανίας, μπορεί να βελτιστοποιήσει τη γραμμή παραγωγής της.

Μελλοντική εργασία:

Μια μελλοντική εργασία που αποφαίνεται χρήσιμη στη βελτιστοποίηση της βιομηχανικής γραμμής παραγωγής, είναι η υλοποίηση των αντικειμενικών συναρτήσεων, των μεθόδων Choquet, ελαχίστου και μέσου όρου, και για τα υπόλοιπα προγραμματιστικά προβλήματα, όπως οι Parallel & Identical Machines, το Flow-Shop, το Open-Shop. Μια ανάλογη διεργασία ακολουθείται και για αυτά τα προβλήματα, όπως:

- μοντελοποίηση του προβλήματος,
- εύρεση των αντικειμενικών συναρτήσεων των μεθόδων Choquet, ελαχίστου και μέσου όρου,
- συγκριτική μελέτη τους,
- ανίχνευση κατάλληλων παραμέτρων, ώστε η εύρεση βέλτιστης λύσης να γίνεται σε εύλογο χρονικό διάστημα.

Μερικά από τα υπόλοιπα προβλήματα, όπως το Flow-Shop και το Open-Shop είναι προβλήματα ιδιαίτερα αυξημένης πολυπλοκότητας και θα ήταν ενδιαφέρον να εξετασθεί εάν, η χρήση πολυ-υποκειμενικού γενετικού αλγορίθμου σε συνδυασμό με τη μέθοδο Choquet, έχει καλύτερα αποτελέσματα από τις υπόλοιπες μεθόδους.

10. Βιβλιογραφία:

- Goldberg, D. E., “*Genetic Algorithms in Search, Optimization & Machine Learning*”, Boston, MA: Addison Wesley Longman, Inc., 1989.
- Andreas C. Nearchou and Vasilis C. Moulianitis “*Multi- objective optimization with fuzzy measures and its application to Flow-Shop Scheduling*” Department of Business Administration, University of Patras, and Department of Mechanical Engineering & Aeronautics, University of Patras, Greece.
- Moulianitis V. C. and Aspragathos N.A. “*Design evaluation with mechatronics index using the discrete Choquet integral*” University of Patras Mechanical Engineering & Aeronautics Department, Greece, 2006.
- Murata T., Ishibuchi H. & Tanaka H. “*Multi- objective Genetic Algorithm and its application to flowshop scheduling*”, Department of industrial engineering, Osaka Prefecture University, Gakuen-cho I-1, Sakai, Osaka 593, Japan, 1996.
- Michel Grabisch “*The application of fuzzy integrals in multicriteria decision making*”, Thomson- CSF, Central research laboratory, Domaine de Corbeville, 91404 Orsay cedex, France, May 1996.
- Marichal, J.-L., “*An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria*”, IEEE Transactions on Fuzzy Systems, 8, 800-807, 2000.
- M. Garey and D. Johnson, “*Computers and intractability: A guide to the theory of NP- completeness*”, Freeman, New York, 1979.
- William L. Chapman, Jerzy Rosenblit & A. Terry Bahill, “System Design is an NP- Complete Problem”, 2001.
- Βλαχαβάς Ι., Κεφάλας Π., Βασιλειάδης Ν., Ρεφανίδης Ι., Κόκκορας Φ., Σακελλαρίου Η., “Τεχνητή Νοημοσύνη”, Θεσσαλονίκη, Ιανουάριος 2002.
- Yusuke Nojima, Kaname Narukawa, Shiori Kaige, and Hisao Ishibuchi “*Effects of removing overlapping solutions on the Performance of the NSGA- II Algorithm*”, Department of Industrial Engineering, Osaka Prefecture University, 1-1 Gakuen-cho, Sakai, Osaka, 599-8531, Japan.
- Dr. James F. Smith III, “*Evolving fuzzy decision tree- structure that adapts in real- time*”, Naval Research Laboratory, Code 5741, Washington, USA, 2005.
- Chapman & Hall/ CRC, “*Handbook of scheduling: algorithms, models & performance analysis*”, CRC Computer & Information Science Series, Edited by Joseph Y- T. Leung.

- Ishibuchi H. & Tanaka H. “*Multi- objective programming in optimization of the interval objective function*”, Department of Industrial Engineering, College of Engineering, University of Osaka Prefecture, 4- 804, Mozu-umemachi, Sakai, Osaka, 591, 1989,Japan.
- Chandrasekharan Rajendran, “*Heuristic algorithm for scheduling in a flowshop to minimize total flowtime*”, Division of Industrial Engineering and Management, Department of Humanities and Social Sciences, Indian Institute of Technology, Madras-600 036, 1992, India, International journal of production economics 29(1995), 65-73.
- Hisao Ishibuchi and Takashi Yamamoto, “*Effects of Three-Objective Genetic Rule Selection on the Generalization Ability of Fuzzy Rule-Based Systems*”, Department of Industrial Engineering, Osaka Prefecture University, 1-1 Gakuencho, Sakai, Osaka 599-8531, (Publication of 2003), Japan.
- E. Taillard, “*Benchmarks for basic scheduling problems*”, Ecole Polytechnique Federale de Lausanne, Departement de Mathématiques, CH-1015 Lausanne, 1989, Switzerland, European journal of operational research 64(1993), 278-285.
- Hisao Ishibuchi, “*Balance Between Genetic Search and Local Search in Memetic Algorithms for Multi- objective Permutation Flowshop Scheduling*”, Member, IEEE, Tadashi Yoshida, and Tadahiko Murata, Member, IEEE Trans. On evolutionary computation 7 (2003) 204-223.
- Johnny C. Ho and Yih-Long Chang, “*A new heuristic for the n-job, M-machine flow-shop problem*”, College of Management, Georgia Institute of Technology, Atlanta, GA 30332-0520, USA.
- José Fernando Gonçalves, Jorge José de Magalhães Mendes, Maurício G. C. Resende, “*A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem*”, Faculdade de Economia da Universidade do Porto, Instituto Superior de Engenharia do Porto Depto. de Engenharia Informática, Internet and Network Systems Research AT&T Labs Research Florham, AT&T Labs Research Technical Report TD-5EAL6J, September 2002.
- T'Kindt, V., Billaut, J.-C. “*Multicriteria scheduling problems: A survey*” RAIRO operation research 35(2001), 143-163.
- Johnson, S.M. “*Optimal two & three-stage production schedules with set up times included*”, Naval Res. Logistics Q.1(1) (1954), 61-68.
- Horn, J., Nafpliotis,N., Goldberg, D.E. “*A niched Pareto genetic algorithm for multi- objective optimization*”, Proc. 1st CIEC (1994), 82-87.
- Horavitz E. & Sahnis S., “*Fundamentals of Computer Algorithms*”, Rockville, MD: Computer Science Press Inc 1978

- Michalewicz CZ Z., “*Genetic Algorithms + Data Structures = Evolution Programs*”, Third, Revised & Extended Edition, New York, New York: Springer-Verlag, 1999.
- Mitchell M., “*An introduction to Genetic Algorithms*”, Cambridge: The MIT Press, 1999.

11. Παράρτημα:

Main Program

```
clear
num_of_mach=3
num_of_tasks=4;
popsize=10;
maxgen=50;
pcross=0.8;
pm=0.01;

v1b=zeros(1,num_of_mach*num_of_tasks);
vub=ones(1,num_of_mach*num_of_tasks)*num_of_tasks;
gen=init(v1b,vub,popsize,num_of_mach*num_of_tasks);
[fitness,object]=score(gen,popsize);
[best_obj(1), index] = min(object);
best_gen=gen(index,:);
[worst_obj(1), index1] = max(object);
worst_cur_gen=gen(index1);
avg_obj(1)=0;

for k=1:popsize
    avg_obj(1)=avg_obj(1)+object(k);
end;

avg_obj(1)=avg_obj(1)/popsize;
for i1=1:2
    fprintf(1, '%f ',best_gen(1));
end;

fprintf('\n');
fprintf(1, 'BEST : %f WORST : %f AVG : %f \n',
        best_obj(1),worst_obj(1),avg_obj(1));

for i=1:maxgen
    gen=reproduc(gen,fitness);
    gen=mate(gen);
    gen=xover8(gen, pcross);
    gen=mutatepuma8(gen, pm);
    [fitness, object]=score(gen,popsize);
    [best_cur_obj, index] = min(object);
    best_cur_gen=gen(index, :);
    [worst_obj(i+1), index1] = max(object);
    worst_cur_gen=gen(index1);
    avg_obj(i+1)=0;

    for k=1:popsize
        avg_obj(i+1)=avg_obj(i+1)+object(k);
    end;
```

```
avg_obj(i+1)=avg_obj(i+1)/popsize;

    if(best_cur_obj > best_obj(i))
gen(index1,:) = best_gen;
    object(index1) = best_obj(i);
    best_obj(i+1) = best_obj(i);
elseif(best_cur_obj <= best_obj(i))
    best_gen = best_cur_gen;
    best_obj(i+1) = best_cur_obj;
end;
best_x(i+1)=best_gen(1);
best_y(i+1)=best_gen(2);

for il=1:num_of_mach*num_of_tasks
    fprintf(1,'%i ',best_gen(il));
end;
fprintf(1,'---> %f\n',best_obj(i+1));
fprintf('\n');
fprintf(1,'BEST : %f WORST : %f AVG : %f \n',
best_obj(i+1),worst_obj(i+1),avg_obj(i+1));
end;
xx=1:maxgen+1;
clear;
```

Choquet Integral

```

function fuzzy_measure_score=Choquet_I(criteria_values)

m{1}=0.45;
m{2}=0.45;
m{3}=0.3;
m2{1}=0.5;
m2{2}=0.9;
m2{3}=0.9;

[sorted_cv,index]=sort(criteria_values);
for i=1:3
    sorted_fm(i)=(m{index(i)});
end
if index==[1 2 3]
    sorted_fm2=[(m2{1}) (m2{2}) (m2{3})];
elseif index==[1 3 2]
    sorted_fm2=[(m2{2}) (m2{1}) (m2{3})];
elseif index==[2 1 3]
    sorted_fm2=[(m2{1}) (m2{3}) (m2{2})];
elseif index==[2 3 1]
    sorted_fm2=[(m2{3}) (m2{1}) (m2{2})];
elseif index==[3 1 2]
    sorted_fm2=[(m2{2}) (m2{3}) (m2{1})];
elseif index==[3 2 1]
    sorted_fm2=[(m2{3}) (m2{2}) (m2{1})];
end

fuzzy_measure_score=0;
for i=1:3
    if i==1
        a=1;
        b=sorted_fm2(3);
    elseif i==2
        a=sorted_fm2(3);
        b=sorted_fm(3);
    elseif i==3
        a=sorted_fm(3);
        b=0;
    end

    fuzzy_measure_score=fuzzy_measure_score+sorted_cv(i)*(a-
b);
end

```