



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΤΩΝ ΚΑΤΕΡΓΑΣΙΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΣΤΟ ΘΕΜΑ

**“ΣΥΝΘΕΣΗ ΣΥΣΤΗΜΑΤΩΝ ΚΑΤΕΡΓΑΣΙΩΝ
ΜΕ ΒΑΣΗ ΕΝΑΛΛΑΚΤΙΚΑ ΦΑΣΕΟΛΟΓΙΑ”**

ΣΑΚΕΛΛΑΡΙΟΥ ΦΩΤΙΟΣ

Επιβλέπων καθηγητής: **Γ. ΒΟΣΝΙΑΚΟΣ**

ΑΘΗΝΑ, ΝΟΕΜΒΡΙΟΣ 2007

Πρόκειται για τη διπλωματική εργασία που εκπόνησα στα πλαίσια των σπουδών μου στη Σχολή Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου. Το θέμα της συγκαταλέγεται στη θεματολογία του ευρύτερου τομέα Τεχνολογίας των Κατεργασιών και συγκεκριμένα των Συστημάτων Κατεργασιών. Θα ήθελα να ευχαριστήσω τον καθηγητή Γ. Βοσνιάκο και τον υποψήφιο διδάκτορα Ξ. Γωγουβίτη για την πολύτιμη βοήθεια που μου προσέφεραν καθ' όλη τη διάρκεια εκπόνησης της εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	3
Κεφάλαιο 1^ο: Εισαγωγή	4
1.1 Εικονικά συστήματα κατεργασιών.....	5
1.2 e-manufacturing.....	9
1.3 Ανασκόπηση.....	11
Κεφάλαιο 2^ο: Γενετικοί αλγόριθμοι	16
2.1 Γενικά.....	17
2.2 Σύντομη ιστορική αναδρομή.....	18
2.3 Πλεονεκτήματα γενετικών αλγορίθμων.....	20
2.4 Διαφοροποίηση από της κλασσικές μεθόδους βελτιστοποίησης.....	22
2.5 Η δομή ενός γενετικού αλγόριθμου.....	24
2.6 Παράμετροι βελτιστοποίησης.....	34
2.7 Τεχνικές κωδικοποίησης.....	37
2.8 Ενσωμάτωση περιορισμών.....	39
2.9 Γνωστές εφαρμογές γενετικών αλγορίθμων.....	40
2.10 Συμπεράσματα.....	41
Κεφάλαιο 3^ο: Αντικειμενική συνάρτηση	43
3.1 Κωδικοποίηση προβλήματος.....	44
3.2 Κριτήρια βελτιστοποίησης.....	45
3.3 Χρόνος ολοκλήρωσης κατεργασιών.....	46
3.4 Περιγραφή λειτουργίας αντικειμενικής συνάρτησης.....	49
3.5 Διάγραμμα ροής.....	53
3.6 Κώδικας – Matlab.....	55
Κεφάλαιο 4^ο: Βάση δεδομένων	63
4.1 Κεντρικό σχεσιακό διάγραμμα – Βασικές οντότητες.....	64
4.2 Συμπληρωματικό σχεσιακό διάγραμμα – Δευτερεύουσες οντότητες.....	69
4.3 Σύνδεση βάσης δεδομένων με αντικειμενική συνάρτηση.....	79
Κεφάλαιο 5^ο: Βελτιστοποίηση ενός προβλήματος	84
5.1 Βάση δεδομένων.....	85
5.2 Πίνακας Phases/Machines.....	95
5.3 fdatabase.m.....	100
5.4 Αποτελέσματα βελτιστοποίησης του προβλήματος.....	106
5.5 Πειραματισμοί με παραμέτρους των γενετικών αλγορίθμων.....	115
Συμπεράσματα	121
Βιβλιογραφία	122
Παράρτημα	124

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία αφορά στο σχεδιασμό συστημάτων κατεργασιών με έμφαση στη διαχείριση της σύνθεσής τους από εναλλακτικά στοιχεία. Πρόκειται για τη δημιουργία συστημάτων κατεργασιών που απαρτίζονται από εξοπλισμό ο οποίος δε βρίσκεται απαραίτητα στον ίδιο χώρο και υπό αυτή την έννοια δεν πρόκειται για ενιαία – πραγματικά συστήματα αλλά για εικονικά.

Στο 1^ο κεφάλαιο περιγράφεται η έννοια των εικονικών συστημάτων και παρουσιάζεται η χρησιμότητά τους στην υλοποίηση προγραμμάτων e-manufacturing. Η βελτιστοποίηση που απαιτείται στα πλαίσια τέτοιων συστημάτων γίνεται με τη μέθοδο των γενετικών αλγορίθμων.

Το 2^ο κεφάλαιο αφορά στη δομή και λειτουργία των γενετικών αλγορίθμων, τα πλεονεκτήματά και τις διαφορές τους από τις κλασσικές μεθόδους βελτιστοποίησης. Αναφέρονται οι λόγοι για την επιλογή της μεθόδου αυτής στη σύνθεση συστημάτων κατεργασιών με βάση εναλλακτικά φασεολόγια. Τα τρία σημαντικότερα στοιχεία των γενετικών αλγορίθμων είναι η κωδικοποίηση του προβλήματος σε ένα χρωμόσωμα, η αντικειμενική συνάρτηση και μία βάση δεδομένων από την οποία αντλούνται πληροφορίες για το πραγματικό πρόβλημα.

Το 3^ο κεφάλαιο αναφέρεται στην αντικειμενική συνάρτηση και στην κωδικοποίηση του προβλήματος. Ορίζονται τα κριτήρια βελτιστοποίησης. Παρουσιάζονται ακόμα η δομή και λειτουργία της συνάρτησης που προγραμματίστηκε στα πλαίσια της υλοποίησης των εικονικών συστημάτων.

Η βάση δεδομένων περιγράφεται στο 4^ο κεφάλαιο. Συγκεκριμένα παρουσιάζονται η δομή και τα χαρακτηριστικά της, οι οντότητες και οι μεταξύ τους σχέσεις καθώς και η σύνδεσή της με την αντικειμενική συνάρτηση.

Το 5^ο κεφάλαιο αφορά ένα υποθετικό πρόβλημα (παραγγελία προϊόντων με εναλλακτικά φασεολόγια) το οποίο βελτιστοποιείται προς τη δημιουργία του εικονικού συστήματος. Περιγράφονται όλα τα βήματα της μεθόδου από την βάση δεδομένων έως και τη βελτιστοποίηση, αναλύονται και σχολιάζονται τα αποτελέσματα. Στο τέλος του κεφαλαίου πειραματιζόμαστε με διάφορες παραμέτρους των γενετικών αλγορίθμων.

ΚΕΦΑΛΑΙΟ 1^ο

ΕΙΣΑΓΩΓΗ

Στο εισαγωγικό αυτό κεφάλαιο περιγράφεται η έννοια των εικονικών συστημάτων κατεργασιών σε προβλήματα παραγωγής προϊόντων με εναλλακτικά φασεολόγια (παράγραφος 1.1).

Η επόμενη παράγραφος (1.2) αφορά στα συστήματα παραγγελίας μηχανολογικών προϊόντων μέσω του διαδικτύου (e-manufacturing). Λειτουργικός παράγοντας για την υλοποίησή τους αποτελούν τα εικονικά συστήματα κατεργασιών.

Στην τελευταία παράγραφο (1.3) παρουσιάζονται περιλήψεις άρθρων και εργασιών με θέματα σχετικά του θέματος της παρούσας διπλωματικής εργασίας, τα οποία χρησιμοποιήθηκαν ως βιβλιογραφικές πηγές.

1.1 ΕΙΚΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΤΕΡΓΑΣΙΩΝ

Έστω το πρόβλημα της παραγωγής προϊόντων με γνωστές κατεργασίες παραγωγής οι οποίες είναι οργανωμένες σε ένα ή περισσότερα εναλλακτικά φασεολόγια. Θεωρούμε ακόμα πως υπάρχει ένα δίκτυο πραγματικών συστημάτων παραγωγής, τα οποία περιέχουν μεταξύ των άλλων και μέρος του εξοπλισμού που μπορεί να χρησιμοποιηθεί για την υλοποίηση κάποιων από τις κατεργασίες που περιλαμβάνονται στα φασεολόγια των προϊόντων.

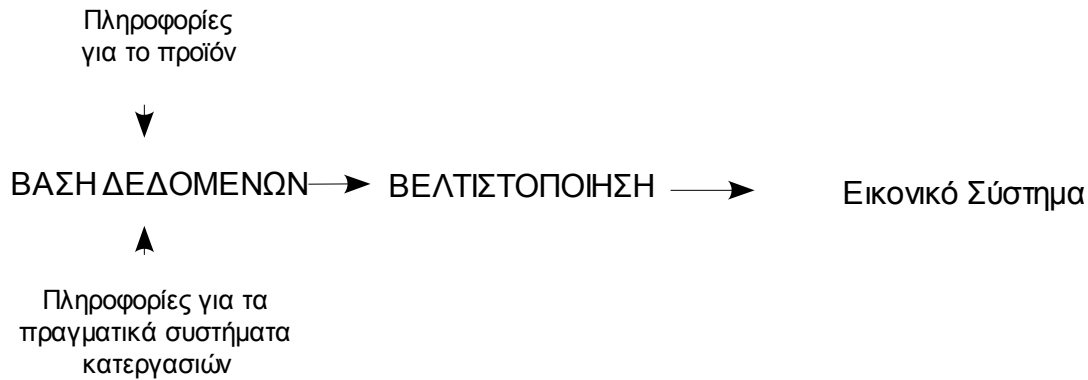
Τα συστήματα παραγωγής δε βρίσκονται απαραίτητα στον ίδιο τόπο, που σημαίνει ότι η μεταφορά τεμαχίων από κατασκευαστή σε κατασκευαστή απαιτεί χρόνο και κοστίζει. Παράλληλα ο εξοπλισμός αυτός έχει κάποια δυναμικότητα και διαθεσιμότητα σε συγκεκριμένες χρονικές περιόδους, δεδομένου ότι χρησιμοποιείται ως μέρος ενός πραγματικού συστήματος κατεργασιών, που παράγει κι άλλα προϊόντα πέραν αυτών που μελετώνται.

Το πρόβλημα σε αυτή την περίπτωση αφορά στην κατανομή των απαιτούμενων κατεργασιών σε συγκεκριμένες μηχανές, με βέλτιστο τρόπο. Η βελτιστοποίηση του προβλήματος οδηγεί στη δημιουργία ενός εικονικού συστήματος κατεργασιών, με την έννοια ότι οι πόροι που χρησιμοποιούνται μπορεί να ανήκουν σε διαφορετικούς κατασκευαστές οι οποίοι είναι διασκορπισμένοι γεωγραφικά.

Ανακύπτουν τα ακόλουθα ερωτήματα:

1. Ποιες πληροφορίες απαιτούνται όσον αφορά τα προϊόντα και τα συστήματα κατεργασιών; Πως θα πρέπει να είναι οργανωμένες αυτές οι πληροφορίες;
2. Πως θα δημιουργηθεί ένα πλάνο παραγωγής όπου η χρήση των μηχανών θα είναι βελτιστοποιημένη;

Δύο είναι τα βασικά εργαλεία που χρησιμοποιούνται για τη σύνθεση ενός εικονικού συστήματος κατεργασιών. Πρόκειται για τη βάση δεδομένων και το λογισμικό βελτιστοποίησης.



ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Το πρώτο πρόβλημα που θα μας απασχολήσει είναι η σύνθεση ενός συνόλου αναγκαίων πληροφοριών σχετικών με το προϊόν.

Οι πληροφορίες ταξινομούνται σε τρεις κατηγορίες:

1. Παραγγελία: ονομασία, περιγραφή και κωδικός του προϊόντος, μέγεθος της παραγγελίας, χρόνοι παραγωγής και παράδοσης.
2. Γεωμετρία: το σχέδιο CAD του τελικού προϊόντος και οι διαστάσεις του ακατέργαστου τεμαχίου.
3. Τεχνολογικά χαρακτηριστικά: πρόκειται για ένα ή περισσότερα εναλλακτικά φασεολόγια.

Στο φασεολόγιο σημαντικό είναι τα γεωμετρικά χαρακτηριστικά να παρουσιάζονται ενοποιημένα με τα τεχνολογικά. Κάθε φάση αντιστοιχεί σε ένα γεωμετρικό χαρακτηριστικό (geometric feature), μία συγκεκριμένη κατεργασία (process), κάποιο εργαλείο, υπολογισμένο χρόνο και κόστος παραγωγής.

Με τον όρο γεωμετρικό χαρακτηριστικό εννοούμε μία συγκεκριμένη μορφή πάνω στο τεμάχιο, που ισοδυναμεί με τον όγκο που πρέπει να αφαιρεθεί από το ακατέργαστο block ώστε να προκύψει η τελική μορφολογία. Ένα γεωμετρικό χαρακτηριστικό χαρακτηρίζεται

από το είδος του (πχ ποκέτα, σκαλοπάτι, οπή, σπείρωμα), τις διαστάσεις του και την ακρίβεια που απαιτεί.

Η κατεργασία χαρακτηρίζεται από διάφορες τεχνολογικές απαιτήσεις ανάλογα με το είδος της.

Το εργαλείο χαρακτηρίζεται από το είδος του, το υλικό του και τα βασικά γεωμετρικά μεγέθη του.

Είναι θεμιτό να χρησιμοποιούνται πολλαπλά φασεολόγια (κατεργασίες διαφορετικού είδους, σε διαφορετική σειρά, με ποικιλία εργαλείων). Η ύπαρξη εναλλακτικών φασεολογιών δίνει τη δυνατότητα αξιολόγησης πολλών εναλλακτικών σχεδίων παραγωγής και εύρεσης του καλύτερου με βάση κριτήρια κόστους και χρόνου.

Αναφορικά με τις μηχανές και τα εργαλεία οι παρακάτω πληροφορίες είναι απαραίτητες:

1. Ικανότητα μηχανής για συγκεκριμένη κατεργασία: Η ικανότητα μιας μηχανής να εκτελέσει μια συγκεκριμένη κατεργασία εξαρτάται από διάφορους παράγοντες όπως τα διαθέσιμα εργαλεία, τις διαθέσιμες ιδιοσυσκευές, τον προσανατολισμό του τεμαχίου σε σχέση με τη μηχανή, τους βαθμούς ελευθερίας της μηχανής και την κατεύθυνση της κατεργασίας. Για λόγους απλοποίησης του προβλήματος, θα θεωρήσουμε την ικανότητα για συγκεκριμένη κατεργασία ως μια έμφυτη ικανότητα της μηχανής, χωρίς να προχωρήσουμε σε πλήρη μοντελοποίηση του συστήματος.
2. Γεωμετρία (χώρος υποδοχής ακατέργαστου τεμαχίου)
3. Γεωμετρία (χώρος εργασίας): Σε ορισμένες περιπτώσεις ο χώρος εργασίας ταυτίζεται με το χώρο υποδοχής του ακατέργαστου τεμαχίου, συνήθως όμως είναι μικρότερος.
4. Το μέγιστο βάρος τεμαχίου που η μηχανή μπορεί να αντέξει.
5. Τεχνολογικά χαρακτηριστικά: Είναι ανάλογα με το είδος της κατεργασίας που η κάθε μηχανή εκτελεί. Σημαντικό είναι τα τεχνολογικά χαρακτηριστικά να εμφανίζονται συμβατά με τις τεχνολογικές απαιτήσεις του τεμαχίου ώστε να μπορεί να ελεγχθεί η

ικανότητα της μηχανής για τη συγκεκριμένη κατεργασία.

6. Διαθέσιμα εργαλεία

7. Διαθεσιμότητα μηχανής: Είναι αναγκαία για τον χρονικό προγραμματισμό της παραγωγής.

Όλες οι πληροφορίες θα οργανωθούν σε μία βάση δεδομένων και θα είναι στη διάθεση του αλγόριθμου βελτιστοποίησης. Η βάση δεδομένων που σχεδιάστηκε στα πλαίσια της υλοποίησης εικονικών συστημάτων κατεργασιών παρουσιάζεται αναλυτικότερα στο 4^ο κεφάλαιο της εργασίας. Η μεταφορά των πληροφοριών από τη βάση δεδομένων στο λογισμικό βελτιστοποίησης μπορεί να γίνει με SQL προγραμματισμό.

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Στόχος είναι οι κατανομή κατά βέλτιστο τρόπο των κατεργασιών στους διαθέσιμους πόρους (μηχανές).

Το εργαλείο που θα χρησιμοποιηθεί στη βελτιστοποίηση των προβλημάτων είναι οι γενετικοί αλγόριθμοι. Οι λόγοι της επιλογής αυτής παρουσιάζονται στο 2^ο κεφάλαιο της εργασίας, όπου και περιγράφεται αναλυτικά η δομή και η λειτουργία της μεθοδολογίας.

Δύο είναι τα σημαντικότερα συστατικά των γενετικών αλγορίθμων. Πρόκειται για την κωδικοποίηση του προβλήματος και την αντικειμενική συνάρτηση, στοιχεία που συνδέουν τον κώδικα βελτιστοποίησης με το εκάστοτε πρόβλημα. Παρουσιάζονται στο 3^ο κεφάλαιο της εργασίας.

1.2 E-MANUFACTURING

Η ανάπτυξη των αγορών σε παγκόσμια κλίμακα μαζί με την επιτακτική ανάγκη παραγωγής ποιοτικών προϊόντων σε μειωμένο κόστος και χρόνο παραγωγής έχουν αλλάξει ριζικά τη δομή και φιλοσοφία των σύγχρονων επιχειρήσεων. Μεγάλες κατασκευαστικές εταιρείες με ιεραρχικές και συγκεντρωτικές δομές, αποτελούν βραδυκίνητους μηχανισμούς, που αδυνατούν να αντεπεξέλθουν στις σύγχρονες απαιτήσεις.

Η ευελιξία και η προσαρμοστικότητα που απαιτούν οι ταχύτατες μεταβολές στην οικονομία και στις απαιτήσεις των πελατών είναι τα σημαντικότερα ίσως στοιχεία που πρέπει να χαρακτηρίζουν τις σύγχρονες επιχειρήσεις. Ο σχεδιασμός των προϊόντων οφείλει να έχει επίκεντρο τον πελάτη και τις ιδιαίτερες ανάγκες του.

Μέσα σε αυτό το πλαίσιο δημιουργούνται μοντέλα επιχειρήσεων που αποτελούνται από φορείς ανεξάρτητους, ίσως και διασκορπισμένους γεωγραφικά, καθένας από τους οποίους επιτελεί συγκεκριμένο έργο. Η συνεργασία των φορέων αυτών δημιουργεί ένα ενιαίο σύστημα προσφοράς υπηρεσιών. Ο σημαντικότερος παράγοντας που ευνόησε τη δημιουργία και ανάπτυξη τέτοιων συστημάτων είναι η χρήση του διαδικτύου που επιτρέπει τη εύκολη και φτηνή ανταλλαγή των απαιτούμενων πληροφοριών μεταξύ των διαφορετικών φορέων.

Στον τομέα της παραγωγής μηχανολογικών προϊόντων είναι δυνατόν να αναπτυχθεί ένα μοντέλο εικονικού εργοστασίου. Ο πελάτης καταθέτει την παραγγελία του σε ηλεκτρονική μορφή, το σύστημα αναλύει τις πληροφορίες και διαμοιράζει τις απαιτούμενες κατεργασίες στις διαθέσιμες μηχανές. Οι μηχανές είναι δυνατό να ανήκουν σε διαφορετικές κατασκευαστικές εταιρείες, που συνεργάζονται όμως στην υλοποίηση του συστήματος αυτού. Σε κάθε περίπτωση ο πελάτης αντιλαμβάνεται όλες τις μηχανές σαν ένα ενιαίο εικονικό εργοστάσιο.

Αρχικά, ο πελάτης δημιουργεί και καταθέτει την παραγγελία του σε έναν διαμεσολαβητή (interface) που έχει δημιουργηθεί στο διαδίκτυο ειδικά για το σκοπό αυτό. Συμπληρώνει

μια φόρμα με πληροφορίες για τις μερίδες και τις ημερομηνίες παράδοσης. Παράλληλα, μέσω ενός συστήματος CAD online δημιουργεί τα μηχανολογικά σχέδια των προϊόντων. Ορίζει τη γεωμετρία, τα υπόλοιπα γεωμετρικά χαρακτηριστικά, το υλικό και την ποιότητα των επιφανειών.

Το εικονικό σύστημα κατεργασιών είναι ο φορέας που αξιοποιεί όλες τις παραπάνω πληροφορίες για να δημιουργήσει το φασεολόγιο που θα χρησιμοποιηθεί στον προγραμματισμό και την εκτέλεση της παραγγελίας. Έχει προταθεί το μοντέλο των τριών planner agents. Συγκεκριμένα:

Process Planner Agent

Από το σχέδιο CAD της παραγγελίας και τις λοιπές πληροφορίες υλικών και ποιότητας επιφανειών ο μηχανισμός αυτός παράγει φασεολόγια. Αφού αναγνωρίσει τα γεωμετρικά χαρακτηριστικά (features recognition), αντιστοιχεί σε κάθε ένα από αυτά τις μηχανολογικές κατεργασίες που απαιτούν (process attribution). Δημιουργούνται έτσι εναλλακτικά φασεολόγια 1^{ης} μορφής με τις απαιτούμενες κατεργασίες στη σειρά που πρέπει να γίνουν καθώς και το γεωμετρικό χαρακτηριστικό στο οποίο αναφέρεται η κάθε μία.

Manufacturing planner agent

Τα εναλλακτικά φασεολόγια 1^{ης} μορφής περνούν στο δεύτερο μηχανισμό που έχει σκοπό να εντοπίσει από τους πραγματικούς πόρους του συστήματος, εκείνους που δύνανται να εκτελέσουν τις κατεργασίες. Απαιτούνται πληροφορίες για τις προδιαγραφές των μηχανών, τα εργαλεία και τις ιδιοσυσκευές που διαθέτουν, τον προσανατολισμό του τεμαχίου σε σχέση τη μηχανή καθώς και την κατεύθυνση της κατεργασίας. Επίσης, μέσω αλγορίθμων παράγονται οι αναμενόμενοι χρόνοι και τα κόστη των κατεργασιών. Αποτέλεσμα της δράση του μηχανισμού αυτού είναι η δημιουργία εναλλακτικών φασεολογίων 2^{ης} μορφής στα οποία κάθε κατεργασία αντιστοιχείται στις μηχανές που μπορούν να την εκτελέσουν με τους αναμενόμενους χρόνους και τα κόστη της παραγωγής.

Production Planner Agent

Τα αποτελέσματα των παραπάνω εργασιών προωθούνται στο τελευταίο σύστημα. Πρόκειται για το σχεδιαστή του εικονικού συστήματος. Δουλειά του είναι να αξιολογήσει τα εναλλακτικά φασεολόγια 2^{ης} μορφής και να επιλέξει το βέλτιστο βάσει χρονικών και κοστολογικών κριτηρίων. Καταλήγουμε έτσι στο τελικό φασεολόγιο (3^{ης} μορφής) με τις συγκεκριμένες μηχανές στις οποίες θα προωθηθεί η παραγγελία.

Το θέμα αυτής της διπλωματικής εργασίας αποτελεί μια απλοποιημένη εκδοχή των παραπάνω λειτουργιών (δεν πραγματεύεται τα θέματα features recognition, process attribution και αυτοματοποιημένου υπολογισμού κοστών και χρόνων κατεργασιών). Παράλληλα, η βελτιστοποίηση που προτείνεται με χρήση των γενετικών αλγορίθμων μπορεί εύκολα να ενσωματωθεί στη λειτουργία του production planner agent.

1.3 ΑΝΑΣΚΟΠΗΣΗ

Οι εργασίες [1] έως [8] πραγματεύονται τη σύνθεση συστημάτων κατεργασιών στην περίπτωση που τα προϊόντα προς παραγωγή διαθέτουν εναλλακτικά φασεολόγια. Συγκεκριμένα:

Στο άρθρο [1] γίνεται θεώρηση ενός γενικευμένου τεχνολογικού προβλήματος ομαδοποίησης και παραγωγής συνόλου τεμαχίων καθένα από τα οποία μπορεί να έχει εναλλακτικά φασεολόγια και κάθε κατεργασία των φασεολογίων μπορεί να πραγματοποιείται σε εναλλακτικές μηχανές. Στόχος είναι να αναλυθεί και να μοντελοποιηθεί ο τρόπος με τον οποίο τα εναλλακτικά φασεολόγια μπορούν να επηρεάσουν τη χρήση των πόρων όταν οι οικογένειες προϊόντων και μηχανών ορίζονται ταυτόχρονα.

Η εργασία [2] αναφέρεται στο πρόβλημα σχεδιασμού κυτάρων μηχανών για την παραγωγή τεμαχίων με εναλλακτικά φασεολόγια. Ένας αλγόριθμος τριών βημάτων αναλύει το αρχικό πρόβλημα σε τρία υποπρόβλήματα: επιλογής φασεολογίου, ανάθεσης

κατεργασιών σε τεμάχια και σε μηχανές. Έγιναν έλεγχοι σε προβλήματα της βιβλιογραφίας και σε άλλα που προέκυψαν κατά τη μελέτη ώστε να υπολογιστεί η απόδοση του προτεινόμενου αλγορίθμου.

Όταν πρόκειται για εναλλακτικά φασεολόγια [3] κατά το σχεδιασμό ενός κυτάρου, προκύπτει η δυνατότητα να πραγματοποιηθεί μια κατεργασία ενός τεμαχίου σε εναλλακτικές μηχανές. Ο σχεδιασμός του κυτάρου χωρίζεται σε δύο μέρη: στο πρόβλημα των πολλών μηχανών κάθε είδους για ένα και μοναδικό φασεολόγιο και στο πρόβλημα της κατανομής τεμαχίων και μηχανών σε κύτταρα. Παρουσιάζεται η δημιουργία ενός ρεαλιστικού μοντέλου στην περίπτωση που οι αναμειξείς προϊόντων και τεμαχίων είναι σταθερές στον χρονικό ορίζοντα. Καθώς το πρόβλημα είναι np-hard δύο διαφορετικές ανωτέρου επιπέδου ευρετικές μέθοδοι, βασισμένες στην αναζήτηση tabu, παρουσιάζονται.

Το πρόβλημα της δημιουργίας κυτάρων από εναλλακτικά φασεολόγια παραγωγής αναλύεται στην εργασία [4]. Η επιλογής της βέλτιστης διαδρομής προκύπτει αρχικά με αντικειμενικό στόχο την μεγιστοποίηση της συμβατότητας μεταξύ επιλεγμένων φασεολογίων. Δημιουργούνται οικογένειες προϊόντων βασισμένες στα αποτελέσματα της πρώτης φάσης χρησιμοποιώντας ένα p-median πρόβλημα. Οι λύσεις συγκρίνονται με αυτές του γενικευμένου p-median μοντέλου με κριτήρια αποτελεσματικότητας στην ομαδοποίηση.

Η σημασία της συμβατότητας μεταξύ σχεδιασμού φασεολογίων και χρονοπρογραμματισμού στα ευέλικτα συστήματα κατεργασιών επισημαίνεται στο άρθρο [5]. Μια αποτελεσματική συνεργασία αυξάνει τη δυνατότητα για αυξημένη απόδοση του συστήματος και καλύτερη λήψη αποφάσεων. Παρουσιάζεται ένα πλαίσιο συνεργασίας ευέλικτων φασεολογίων σε συνδυασμό με τον απαραίτητο χρονοπρογραμματισμό. Συζητείται ακόμα η ευελιξία στο σχεδιασμό φασεολογίων, που περιλαμβάνει την ευελιξία των κατεργασιών, στην αλληλουχία τους και στις εναλλακτικές μηχανές. Το πλαίσιο αποτελείται από τέσσερις συνεργαζόμενες βαθμίδες και στοχεύει τη μείωση του χρόνου.

Το άρθρο [6] πραγματεύεται τη χρήση εναλλακτικών φασεολογίων στο επίπεδο των

μηχανών. Προτείνεται ένα μοντέλο αναπαράστασης των πλάνων αυτών και εξετάζονται οι περιορισμοί που σχετίζονται με την εκτέλεσή τους. Εναλλακτικά φασεολόγια, για συγκεκριμένες μηχανές, μελετώνται χωρίς όμως να λαμβάνεται υπ όψιν η αστοχία των μηχανών ή των εργαλείων. Στήνεται πείραμα για να αναλυθεί η απόδοση ενός τέτοιου σχήματος, όπου μεταβλητές είναι τα τεμάχια προς κατεργασία, ένα πρόγραμμα παραγωγής που μεταβάλλεται χρονικά και ένα δυναμικό σύνολο εργαλείων κοπής που απαιτούνται για την κατεργασία των τεμαχίων. Γίνεται σύγκριση για να διαπιστωθεί η βέλτιστη στρατηγική: χρήση ενός στατικού φασεολογίου, να χρησιμοποιούνται τα υπάρχοντα εργαλεία της μηχανής για όλα τα τεμάχια ή να σετάρεται μερικώς η μηχανή και να χρησιμοποιούνται εναλλακτικά εργαλεία που δεν έχουν ακόμα τοποθετηθεί;

Η εργασία [7] αφορά στην ανάπτυξη ενός προγράμματος σχεδιασμού συστημάτων κατεργασιών σε μορφή κυττάρων. Προϋποθέτει την ύπαρξη εναλλακτικών φασεολογίων για κάθε τεμάχιο, ενώ κάθε κατεργασία είναι δυνατόν να πραγματοποιηθεί σε σύνολο εναλλακτικών μηχανών. Στόχος του μοντέλου είναι η ελαχιστοποίηση του κόστους της επένδυσης, των κατεργασιών, των υλικών και του κόστους χειρισμού. Για κάθε κατεργασία θεωρούνται συγκεκριμένοι χρόνοι, οι ικανότητες των μηχανών είναι προσδιορισμένες, ενώ υπάρχουν περιορισμοί στο μέγεθος του κάθε κυττάρου. Οικογένειες τεμαχίων, μηχανών και φασεολογίων δίνονται ταυτόχρονα. Από το πρόγραμμα δημιουργείται μία στήλη αποτελεσματικότητας της κάθε λύσης που προτείνεται. Εξετάζονται τρεις διαφορετικές στρατηγικές παραγωγής της στήλης αυτής, η καλύτερη από τις οποίες αναλύεται και δοκιμάζεται περαιτέρω.

Το πρόβλημα της ομαδοποίησης μηχανών σε κύτταρα και της επιλογής ενός μοναδικού φασεολογίου για κάθε τεμάχιο που κατασκευάζεται εξετάζεται και στην εργασία [8]. Πρόκειται για συστήματα κατεργασιών με συγκεκριμένες σχεδιαστικές απαιτήσεις και περιορισμούς στην αλληλουχία των κατεργασιών. Το πρόβλημα της ομαδοποίησης τεμαχίων σε ομάδες, είναι μέτριας δυσκολίας όταν υπάρχει μοναδική μηχανή από κάθε είδος, ενώ γίνεται αρκετά δύσκολο, ίσως όμως πιο ρεαλιστικό στην περίπτωση ύπαρξης πολλών μηχανών του ίδιου είδους.

Στις εργασίες [9] έως [10] προτείνεται η ιδέα χρήσης της μεθόδου των γενετικών

αλγορίθμων για βελτιστοποίηση συστημάτων με εναλλακτικά φασεολόγια. Συγκεκριμένα:

Η χρήση γενετικών αλγορίθμων για την περίπτωση των προβλημάτων που εξετάζουμε είναι το θέμα της εργασίας [9]. Παρουσιάζει μία αυτοματοποιημένη διαδικασία επίλυσης προβλημάτων χρονοπρογραμματισμού εργασιών εν μέσω περιορισμών, όπως εργασίες με υπορουτίνες, εναλλακτικά φασεολόγια για τεμάχια, εναλλακτικοί πόροι για κάθε κατεργασία και απαιτήσεις περισσότερων πόρων για μία εργασία. Προσεγγίζεται και η περίπτωση χρήσης πολλαπλών αντικειμενικών συναρτήσεων. Το σύστημα δημιουργεί μέσω ειδικών αλγορίθμων αρχικές λύσεις του προβλήματος και ένας υβριδικός γενετικός αλγόριθμος εφαρμόζεται για την εύρεση του βέλτιστου προγράμματος.

Στο άρθρο [10] παρουσιάζεται ένας γενετικός αλγόριθμος ικανός να παράγει βέλτιστα φασεολόγια στα πλαίσια υλοποίησης ευέλικτων συστημάτων κατεργασιών. Τονίζεται ιδιαίτερα η ικανότητα του συστήματος να παράγει εναλλακτικά φασεολόγια κάτω από απρόβλεπτες καταστάσεις που προκύπτουν στο σύστημα και αναθεωρήσεις στη ροή των κατεργασιών των τεμαχίων. Λαμβάνεται υπ όψιν και ο αντικειμενικός στόχος ελαχιστοποίησης των νεκρών χρόνων μηχανής.

Η μοντελοποίηση των συστημάτων ηλεκτρονικής παραγγελίας μηχανολογικών προϊόντων μέσω των τριών agents εισάγεται για πρώτη φορά στα κεφάλαια του βιβλίου *Designing and evaluating value added services in manufacturing e-marketplaces*. Πρόκειται για τις αναφορές [11] έως [13].

Η τάση των σύγχρονων επιχειρήσεων προς πιο κατανομημένα μοντέλα μελετάται στο κεφάλαιο [11] του βιβλίου περί εικονικών συστημάτων και επιχειρήσεων. Πρόκειται για μια εισαγωγή στο ερευνητικό πρόγραμμα “Κατανομημένος σχεδιασμός κατεργασιών και φασεολογίων σε σύγχρονα κατασκευαστικά δίκτυα” που χρηματοδοτήθηκε από το Ιταλικό Υπουργείο Παιδείας. Το ερευνητικό πρόγραμμα είχε στόχο την ανάπτυξη καινοτόμων ιδεών για ουδέτερες, γραμμικές ηλεκτρονικές αγορές ικανές να ενισχύσουν κατασκευαστικά εταιρικά δίκτυα στον τομέα των συναλλαγών, των πληροφοριών και της συνεργασίας των στοιχείων τους.

Στο επόμενο κεφάλαιο [12] προτείνεται μία αρχιτεκτονική βασισμένη σε 3 φορείς, τον Process Planner Agent, τον Manufacturing Planner Agent και τον Production Planner Agent.

Στην αναφορά [13] αναλύονται λεπτομερειακά οι λειτουργίες του σημαντικότερου ίσως φορέα, του Process Planner Agent. Πρόκειται για τον φορέα αναγνώρισης των κατεργασιών που απαιτούνται και δημιουργίας των εναλλακτικών φασεολογίων.

Για τη συγγραφή του κεφαλαίου περί γενετικών αλγορίθμων σημαντική ήταν η συνδρομή των σημειώσεων του προπτυχιακού μαθήματος Υπολογιστική Νοημοσύνη II της σχολής των Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Πατρών [14]. Απαραίτητο βοήθημα αποτέλεσε και το Genetic Algorithm Toolbox reference manual που συνοδεύει το πακέτο της Matlab [15].

Το βιβλίο [16] αποτελεί μία εισαγωγή στο σχεδιασμό βάσεων δεδομένων και στον SQL προγραμματισμό.

ΚΕΦΑΛΑΙΟ 2^ο

ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Η συγκρότηση ενός εικονικού συστήματος από μηχανές πραγματικών συστημάτων κατεργασιών, βάσει κάποιων κριτηρίων βελτιστοποίησης, ανήκει στη κατηγορία των προβλημάτων χρονοπρογραμματισμού.

Ο χρονοπρογραμματισμός (scheduling) μπορεί να οριστεί ως το πρόβλημα εύρεσης μιας βέλτιστης σειράς για την εκτέλεση ενός πεπερασμένου συνόλου λειτουργιών, χωρίς να παραβιάζεται ένα συγκεκριμένο σύνολο κανόνων. Είναι από τα πιο συνηθισμένα προβλήματα που συναντώνται στο χώρο της βελτιστοποίησης, αλλά και από τα πιο δύσκολα, αφού ανήκει στην κατηγορία των NP-complete προβλημάτων. Στην πιο συχνή τους μορφή, αυτού του είδους τα προβλήματα έχουν ως στόχο τη μεγιστοποίηση της χρήσης ανθρώπων ή πόρων και την ταυτόχρονη ελαχιστοποίηση του χρόνου που απαιτείται για την ολοκλήρωση μιας διεργασίας. Προκύπτουν συγκρούσεις από το γεγονός ότι ένα άτομο ή κάποιος πόρος δεν μπορεί να χρησιμοποιηθεί σε περισσότερες από μια εργασίες ταυτόχρονα, ενώ υπάρχουν και περιορισμοί όπως η τήρηση προτεραιοτήτων, η μη διαθεσιμότητα κάποιων πόρων για κάποιο χρονικό διάστημα, κτλ.

Η πιο συνηθισμένη πρακτική για την επίλυση των προβλημάτων χρονοπρογραμματισμού είναι η χρήση των γενετικών αλγορίθμων. Στο κεφάλαιο αυτό παρουσιάζονται οι γενετικοί αλγόριθμοι, ο τρόπος λειτουργίας τους, τα πλεονεκτήματα και μειονεκτήματά τους.

2.1 ΓΕΝΙΚΑ

Τα τελευταία τριάντα χρόνια, έχει παρατηρηθεί ένα συνεχώς αυξανόμενο ενδιαφέρον για την ανάπτυξη συστημάτων επίλυσης προβλημάτων βασισμένων στις αρχές της Γενετικής Εξέλιξης και της Κληρονομικότητας. Τα μειονεκτήματα των κλασσικών μεθόδων αναζήτησης και βελτιστοποίησης, καθώς και η συνεχώς αυξανόμενη ανάγκη για παραγωγή λογισμικού που να μπορεί να εκμεταλλεύεται πιο αποδοτικά τις τεράστιες δυνατότητες του υλικού, ήταν η βασική αιτία που ώθησε τους επιστήμονες σ' αυτήν την αναζήτηση. Αυτού του είδους τα συστήματα λειτουργούν διατηρώντας ένα πληθυσμό κωδικοποιημένων πιθανών λύσεων και εφαρμόζοντας πάνω σε αυτό διάφορες διαδικασίες επιλογής του καλύτερου, καθώς και διάφορους γενετικούς τελεστές. Οι τελεστές αυτοί αντιγράφουν τον τρόπο με τον οποίο αναπαράγονται και μεταλλάσσονται τα χρωμοσώματα των κυττάρων των ζωντανών οργανισμών. Έτσι, περνώντας από γενιά σε γενιά, τα συστήματα αυτά δημιουργούν συνεχώς νέους πληθυσμούς πιθανών λύσεων χρησιμοποιώντας, τόσο κομμάτια και στοιχεία από την προηγούμενη γενιά, όσο και εντελώς καινούρια κομμάτια που δοκιμάζονται για τυχόν καλή απόδοσή τους.

Επανεπιλημμένες δοκιμές και πειράματα έχουν δείξει ότι μια "φυσική" αναπαράσταση των πιθανών λύσεων για ένα δεδομένο πρόβλημα, σε συνδυασμό με την εφαρμογή σε αυτή μιας οικογένειας γενετικών τελεστών, αποτελεί πολύ χρήσιμο εργαλείο στην προσπάθεια προσέγγισης των πραγματικών λύσεων σε μια πολύ μεγάλη ποικιλία προβλημάτων και εφαρμογών. Αυτό το γεγονός, μετατρέπει αυτή τη "φυσικού μοντέλου" προσέγγιση σε μια πολλά υποσχόμενη κατεύθυνση, όσον αφορά την επίλυση προβλημάτων γενικότερα.

Ένας Γ.Α. για ένα συγκεκριμένο πρόβλημα πρέπει να αποτελείται από τα παρακάτω πέντε συστατικά:

1. Μια γενετική αναπαράσταση των πιθανών λύσεων του προβλήματος.
2. Ένα τρόπο δημιουργίας ενός αρχικού πληθυσμού των πιθανών λύσεων.
3. Μια αντικειμενική συνάρτηση αποτίμησης, που παίζει το ρόλο του περιβάλλοντος κατατάσσοντας τις λύσεις με βάση την καταλληλότητά τους.
4. Γενετικούς τελεστές που μετατρέπουν τη σύνθεση των παιδιών.

5. Τιμές για διάφορες παραμέτρους που χρησιμοποιεί ο Γ.Α. (μέγεθος πληθυσμού, πιθανότητες εφαρμογής των γενετικών τελεστών, κτλ.).

Η πρώτη εμφάνιση των Γενετικών Αλγόριθμων (Γ.Α.) χρονολογείται στις αρχές του 1950, όταν διάφοροι βιολόγοι επιστήμονες αποφάσισαν να χρησιμοποιήσουν υπολογιστές στην προσπάθειά τους να προσομοιάσουν πολύπλοκα βιολογικά συστήματα. Η συστηματική τους ανάπτυξη όμως, που οδήγησε στην μορφή με την οποία είναι γνωστοί και σήμερα, πραγματοποιήθηκε στις αρχές του 1970 από τον John Holland και τους συνεργάτες του στο Πανεπιστήμιο του Michigan.

2.2 ΣΥΝΤΟΜΗ ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Ιδιαίτερο ενδιαφέρον παρουσιάζει η πορεία των ερευνών που οδήγησαν στην ανάπτυξη της θεωρίας των Γ.Α. Στην παράγραφο αυτή παρατίθεται μια λιτή ιστορική αναδρομή που ξεκινά από τις πρώτες προσπάθειες θεμελίωσης της θεωρίας της Εξέλιξης και φτάνει μέχρι τα τελευταία χρόνια, όπου πλέον η εφαρμογή των Γ.Α. είναι καθημερινή πρακτική.

- Charles Darwin, 1809-1882. Κυρίαρχη μορφή στην επιστήμη της Βιολογίας. Ανακάλυψε και διατύπωσε τη θεωρία για την εξέλιξη μέσω της Φυσικής Επιλογής. Βάση αυτής της ανακάλυψης ήταν οι παρατηρήσεις που έκανε κατά την διάρκεια του πενταετούς ταξιδιού του στη Νότιο Αμερική, στη Νότια Αφρική και στην Αυστραλία. Βασικά σημεία αυτής της θεωρίας παρουσιάστηκαν σε προηγούμενη παράγραφο του παρόντος κεφαλαίου.
- Νεοδαρβινισμός, 1930-σήμερα. Είναι η σύνθεση της Δαρβινικής Εξέλιξης και των σύγχρονων αντιλήψεων για την γενετική δομή. Στη βάση αυτής της θεωρίας βρίσκεται η πεποίθηση ότι η μετάλλαξη συμβαίνει τυχαία και επιφέρει ποικιλία στο γενετικό υλικό.
- Ο John Holland αρχίζει την έρευνά του στην προσαρμογή των προγραμμάτων υπολογιστών, 1960. Θεωρείται ο "πατέρας" των Γ.Α. παρότι δεν τους βάφτισε ο ίδιος. Σειρά διαφόρων εργασιών που εκδόθηκαν το 1962 πάνω στη θεωρία των

προσαρμοστικών συστημάτων έβαλαν τα θεμέλια για την ανάπτυξη του χώρου.

- Ο J.D. Bagley βαφτίζει τους Γ.Α., 1967. Η διδακτορική διατριβή του Bagley περιέχει την πρώτη δημοσιευμένη εφαρμογή των Γ.Α. που πρώτη φορά παρουσιάζονται με το όνομα αυτό.
- R.S. Rosenberg, 1967. Δημοσιεύει εργασία, στην οποία γίνεται λόγος για προσομοίωση πληθυσμών μονοκύτταρων οργανισμών σε υπολογιστικό περιβάλλον.
- "Προσαρμογή στα Φυσικά και Τεχνητά συστήματα", 1975. Τίτλος του βιβλίου που εκδίδει ο Holland το 1975, στο οποίο αναπτύσσει τις ιδέες και την θεωρία των Γ.Α. Το βιβλίο θεωρείται πλέον κλασικό για τον χώρο. Θίγονται θέματα όπως η θεωρία των σχημάτων, η βέλτιστη κατανομή των ευκαιριών, σχέδια αναπαραγωγής, γενετικές λειτουργίες, η ευρωστία των Γ.Α. και πλήθος άλλα.
- K.A. De Jong, 1975. Με την εργασία που εκδίδει βοηθά την πειραματική αξιολόγηση των Γ.Α. Σύμφωνα με αυτήν, προτείνονται λειτουργίες που ελέγχουν έναν Γ.Α. και την ικανότητά του να αντιμετωπίζει δύσκολα προβλήματα.
- Ο J.J. Grefenstette δημιουργεί το GENESIS, 1980. Το GENESIS είναι ένα σύστημα ανάπτυξης Γ.Α. υλοποιημένο στη γλώσσα προγραμματισμού C, που έχει βοηθήσει σημαντικά στη διάδοση του γενετικού προγραμματισμού καθώς έγινε διαθέσιμο στο ευρύ κοινό.
- 1ο Διεθνές Συνέδριο των Γ.Α. και των εφαρμογών τους, 1985. Ο χώρος αποκτά ένα μεγάλο συνέδριο που πλέον λαμβάνει χώρα κάθε δύο χρόνια και αντικατοπτρίζει το μεγάλο οργασμό που παρατηρείται σε επίπεδο τόσο θεωρίας, όσο και εφαρμογών.
- Πολυάριθμες εκδόσεις βιβλίων για Γ.Α., 1989-1999. Άλλη μια ένδειξη της τεράστιας ανάπτυξης του χώρου και της αποδοχής της νέας τεχνολογίας.
- Ανάπτυξη πακέτων λογισμικού για Γ.Α., 1990-1999. Πολλές εταιρίες δημιουργούν εμπορικά πακέτα που επιτρέπουν σε χρήστες να ενσωματώσουν στις εφαρμογές

τους στοιχεία Γενετικού Προγραμματισμού (Genetic Programming). Ένα τέτοιο πακέτο είναι το EOS (Evolutionary Object System). Βασίζεται στη δημοφιλή γλώσσα αντικειμενοστραφούς προγραμματισμού C++ και παρέχει μεγάλες δυνατότητες προσαρμογών και επεκτάσεων.

2.3 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Η χρήση των Γ.Α. σε διάφορες εφαρμογές είναι ελκυστική για αρκετούς λόγους. Οι κυριότεροι είναι οι εξής:

1. Μπορούν να λύσουν δύσκολα προβλήματα γρήγορα και αξιόπιστα. Ένας από τους σημαντικούς λόγους χρήσης των Γ.Α. είναι η μεγάλη τους αποδοτικότητα. Τόσο η θεωρία, όσο και η πράξη έχουν δείξει ότι προβλήματα που έχουν πολλές, δύσκολα προσδιορισμένες λύσεις μπορούν να αντιμετωπιστούν καλύτερα από Γ.Α. Είναι δε αξιοσημείωτο ότι συναρτήσεις που παρουσιάζουν μεγάλες διακυμάνσεις και καθιστούν ανεπαρκείς άλλες μεθόδους στην εύρεση των ακρότατων τους, για τους Γ.Α. δεν αποτελούν σημεία δυσχέρειας.
2. Μπορούν εύκολα να συνεργαστούν με τα υπάρχοντα μοντέλα και συστήματα. Οι Γ.Α. προσφέρουν το σημαντικό πλεονέκτημα της χρήσης τους με προσθετικό τρόπο στα μοντέλα που χρησιμοποιούνται σήμερα, μη απαιτώντας την επανασχεδίασή τους. Μπορούν εύκολα να συνεργαστούν με τον υπάρχοντα κώδικα, χωρίς μεγάλο κόπο. Αυτό συμβαίνει, διότι χρησιμοποιούν μόνο πληροφορίες της διαδικασίας ή συνάρτησης που πρόκειται να βελτιστοποιήσουν, δίχως να ενδιαφέρει άμεσα ο ρόλος της μέσα στο σύστημα ή η όλη δομή του συστήματος.
3. Είναι εύκολα επεκτάσιμοι και εξελίξιμοι. Όπως θα γίνει σαφές στα επόμενα κεφάλαια, οι Γ.Α. δεν αντιστέκονται σε αλλαγές, επεκτάσεις και μετεξελίξεις, ανάλογα με την κρίση του σχεδιαστή. Σε πολλές εφαρμογές, έχουν αναφερθεί λειτουργίες των Γ.Α. που δεν είναι δανεισμένες από τη φύση ή που έχουν υποστεί σημαντικές αλλαγές, πάντα προς όφελος της απόδοσης. Παραλλαγές στο βασικό σχήμα δεν είναι απλά αναγκαίες, αλλά σε ορισμένες περιπτώσεις επιβάλλονται.

4. Μπορούν να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους. Αν και η ισχύς των Γ.Α. είναι μεγάλη, σε μερικές ειδικές περιπτώσεις προβλημάτων, όπου άλλες μέθοδοι συμβαίνει να έχουν πολύ υψηλή αποδοτικότητα, λόγω εξειδίκευσης, υπάρχει η δυνατότητα χρησιμοποίησης ενός υβριδικού σχήματος Γ.Α. με άλλη μέθοδο. Αυτό είναι αποτέλεσμα της μεγάλης ευελιξίας των Γ.Α.
5. Εφαρμόζονται σε πολύ περισσότερα πεδία από κάθε άλλη μέθοδο. Το χαρακτηριστικό που τους εξασφαλίζει αυτό το πλεονέκτημα είναι η ελευθερία επιλογής των κριτηρίων που καθορίζουν την επιλογή μέσα στο τεχνικό περιβάλλον. Έτσι, Γ.Α. μπορούν να χρησιμοποιηθούν στην οικονομία, στο σχεδιασμό μηχανών, στην επίλυση μαθηματικών εξισώσεων, στην εκπαίδευση Νευρωνικών Δικτύων και σε πολλούς άλλους τομείς.
6. Δεν απαιτούν περιορισμούς στις συναρτήσεις που επεξεργάζονται. Ο κύριος λόγος που καθιστά τις παραδοσιακές μεθόδους δύσκαμπτες και ακατάλληλες για πολλά προβλήματα είναι η απαίτησή τους για ύπαρξη περιορισμών, όπως ύπαρξη παραγώγων, συνέχεια, όχι "θορυβώδεις" συναρτήσεις κτλ. Τέτοιου είδους ιδιότητες είναι αδιάφορες για τους Γ.Α. πράγμα που τους κάνει κατάλληλους για μεγάλο φάσμα προβλημάτων.
7. Δεν ενδιαφέρει η σημασία της υπό εξέταση πληροφορίας. Η μόνη "επικοινωνία" του Γ.Α. με το περιβάλλον του είναι η αντικειμενική συνάρτηση. Αυτό εγγυάται την επιτυχία του ανεξάρτητα από την σημασία του προβλήματος. Βέβαια, δεν σημαίνει ότι δεν υπάρχουν άλυτα προβλήματα για τους Γ.Α. Όπου όμως δεν τα καταφέρνουν, η αιτία είναι η φύση του χώρου που ερευνούν και όχι το πληροφοριακό περιεχόμενο του προβλήματος.
8. Έχουν από τη φύση τους το στοιχείο του παραλληλισμού. Οι Γ.Α. σε κάθε τους βήμα επεξεργάζονται μεγάλες ποσότητες πληροφορίας, αφού κάθε άτομο θεωρείται αντιπρόσωπος πολλών άλλων. Έχει υπολογιστεί ότι η αναλογία αυτή είναι της τάξεως n^3 , δηλαδή 10 άτομα αντιπροσωπεύουν περίπου 1000. Είναι λοιπόν προφανές ότι μπορούν να καλύψουν με αποδοτικό ψάξιμο μεγάλους χώρους σε μικρούς χρόνους.

9. Είναι η μόνη μέθοδος που κάνει ταυτόχρονα εξερεύνηση του χώρου αναζήτησης και εκμετάλλευση της ήδη επεξεργασμένης πληροφορίας. Ο συνδυασμός αυτός σπάνια συναντάται σε οποιαδήποτε άλλη μέθοδο. Με το τυχαίο ψάξιμο γίνεται καλή εξερεύνηση του χώρου, αλλά δεν γίνεται εκμετάλλευση της πληροφορίας. Αντίθετα, με το hill-climbing γίνεται καλή εκμετάλλευση της πληροφορίας, αλλά όχι καλή εξερεύνηση. Συνήθως τα δύο αυτά χαρακτηριστικά είναι ανταγωνιστικά και το επιθυμητό είναι να συνυπάρχουν και τα δύο προς όφελος της όλης διαδικασίας. Οι Γ.Α. επιτυγχάνουν το βέλτιστο συνδυασμό εξερεύνησης και εκμετάλλευσης [32], πράγμα που τους κάνει ιδιαίτερα αποδοτικούς και ελκυστικούς.
10. Επιδέχονται παράλληλη υλοποίηση. Οι Γ.Α. μπορούν να εκμεταλλευτούν τα πλεονεκτήματα των παράλληλων μηχανών, αφού λόγω της φύσης τους, εύκολα μπορούν να δεχτούν παράλληλη υλοποίηση. Το χαρακτηριστικό αυτό αυξάνει ακόμη περισσότερο την απόδοσή τους, ενώ σπάνια συναντάται σε ανταγωνιστικές μεθόδους.

2.4 ΔΙΑΦΟΡΟΠΟΙΗΣΗ ΑΠΟ ΤΙΣ ΚΛΑΣΣΙΚΕΣ ΜΕΘΟΔΟΥΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Όπως αναφέρθηκε, οι Γ.Α. πλεονεκτούν αισθητά στη λύση προβλημάτων αναζήτησης και βελτιστοποίησης από τις παραδοσιακές μεθόδους. Αυτό συμβαίνει, διότι διαφέρουν θεμελιωδώς από αυτές. Τα κυριότερα νέα χαρακτηριστικά που τους διαφοροποιούν, αλλά και τους δίνουν υπεροχή είναι:

1. Οι Γ.Α. δουλεύουν με μια κωδικοποίηση ενός συνόλου τιμών που μπορούν να λάβουν οι μεταβλητές και όχι με τις ίδιες τις μεταβλητές του προβλήματος: Για παράδειγμα, αναφέρεται το εξής πρόβλημα βελτιστοποίησης: Έστω ένα μαύρο κουτί με πέντε δυαδικούς διακόπτες (on-off). Για κάθε συνδυασμό των διακοπών s παράγεται μία έξοδος $f(s)$. Ζητείται ο συνδυασμός των διακοπών που μεγιστοποιεί την έξοδο. Με τις παραδοσιακές μεθόδους, το μέγιστο θα εντοπιζόταν με "παίξιμο" των διακοπών πηγαίνοντας από συνδυασμό σε συνδυασμό με ψάξιμο στα τυφλά, καθ' ότι δεν είναι γνωστός ο τύπος της συνάρτησης. Σε ένα Γ.Α. όμως, η πρώτη ενέργεια είναι η κωδικοποίηση των διακοπών ως συμβολοσειρές πεπερασμένου μήκους. Μια απλή κωδικοποίηση θα μπορούσε να γίνει θεωρώντας μια δυαδική

συμβολοσειρά μήκους πέντε, όπου η κάθε θέση αναπαριστά ένα διακόπτη. Το 0 αντιστοιχεί στη θέση off και το 1 στη θέση on. Δηλαδή, η συμβολοσειρά 11110 κωδικοποιεί το συνδυασμό κατά τον οποίο οι πρώτοι τέσσερις διακόπτες είναι on και ο τελευταίος off. Η κωδικοποίηση δεν είναι απαραίτητο να είναι πάντα δυαδική. Όπως θα φανεί και αργότερα, μπορεί να γίνει με πολλούς τρόπους, αρκετοί από τους οποίους ίσως και να μην είναι προφανείς. Το στοιχείο της κωδικοποίησης, όπως εξηγείται παρακάτω, είναι εκείνο που επιτρέπει στους Γ.Α. να κάνουν παράλληλη επεξεργασία δεδομένων.

2. Οι Γ.Α. κάνουν αναζήτηση σε πολλά σημεία ταυτόχρονα και όχι μόνο σε ένα: Σε πολλές μεθόδους βελτιστοποίησης, η επεξεργασία γίνεται βήμα προς βήμα, πηγαίνοντας προσεκτικά από σημείο σε σημείο του πεδίου ορισμού του προβλήματος. Αυτό το βήμα προς βήμα ενέχει αρκετούς κινδύνους, ο κυριότερος από τους οποίους είναι να περιοριστεί η αναζήτηση σε μια περιοχή τοπικού ακρότατου, που δεν είναι ολικό. Οι Γ.Α. εξαλείφουν αυτόν τον κίνδυνο ενεργώντας ταυτόχρονα πάνω σε ένα ευρύ σύνολο σημείων (σύνολο από συμβολοσειρές). Έτσι μπορούν να "ανεβαίνουν" πολλούς λόφους (hill-climbing) την ίδια στιγμή, ελαχιστοποιώντας την πιθανότητα να βρουν μια λάθος κορυφή. Γυρίζοντας στο παράδειγμα με το μαύρο κουτί, οι κλασσικές μέθοδοι θα ξεκινούσαν το ψάξιμο από ένα συνδυασμό των διακοπών και στη συνέχεια, εφαρμόζοντας κάποιο κανόνα μετάβασης, θα δοκίμαζαν τον επόμενο (ψάξιμο δηλαδή σημείο προς σημείο). Αντιθέτως, ένας Γ.Α. αρχίζει το ψάξιμο του από ένα πληθυσμό συνδυασμών συμβολοσειρών και κατόπιν παράγει διαδοχικά καινούριους. Ένας αρχικός πληθυσμός θα μπορούσε να είναι, πχ 01101, 11000, 01000 και 10011. Έπειτα, "τρέχοντας" ο αλγόριθμος δημιουργεί νέους πληθυσμούς που σιγά σιγά συγκλίνουν προς την επιθυμητή λύση. Διαλέγοντας ένα πληθυσμό που να καλύπτει αντιπροσωπευτικά ένα μεγάλο εύρος τιμών μπορούν να προκύψουν ικανοποιητικά αποτελέσματα.
3. Οι Γ.Α. χρησιμοποιούν μόνο την αντικειμενική συνάρτηση και καμία επιπρόσθετη πληροφορία: Πολλές μέθοδοι αναζήτησης απαιτούν αρκετές βοηθητικές πληροφορίες για τη συνάρτηση που επεξεργάζονται. Τέτοιου είδους πληροφορίες δεν προαπαιτούνται από τους Γ.Α. Το ψάξιμο τους είναι κατά κάποιο τρόπο "τυφλό".

Αξιοποιούν μόνο όση πληροφορία περιέχεται στην αντικειμενική συνάρτηση πράγμα που τους προσδίδει μεγάλη ευελιξία. Έτσι όμως, προκύπτει το ερώτημα αν συμφέρει να αγνοούνται οι βοηθητικές πληροφορίες. Γι' αυτό ακριβώς το λόγο έχουν αναπτυχθεί μορφές Γ.Α. που αξιοποιούν και τέτοιου είδους πληροφορίες (Knowledge - Based Genetic Algorithms).

4. Οι Γ.Α. χρησιμοποιούν πιθανοθεωρητικούς κανόνες μετάβασης και όχι ντετερμινιστικούς: Η χρήση πιθανοθεωρητικών κανόνων μετάβασης είναι κυρίαρχο γνώρισμα των Γ.Α., χωρίς αυτό να σημαίνει ότι η όλη διαδικασία βαδίζει στην τύχη. Δε λαμβάνονται αποφάσεις με το "στρίψιμο ενός νομίσματος". Το στοιχείο της τύχης χρησιμοποιείται ως οδηγός για αναζήτηση σε περιοχές που αναμένεται να δώσουν καλά αποτελέσματα.

2.5 Η ΔΟΜΗ ΕΝΟΣ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ

Στην ουσία, ένας τυπικός Γ.Α. περιλαμβάνει απλές λειτουργίες, που όμως κρύβουν μέσα τους μεγάλη ισχύ. Αυτός ο συνδυασμός απλοϊκότητας και ισχύος είναι το μεγαλύτερο θέλημα της τεχνικής τους. Στην παράγραφο αυτή παρουσιάζονται τα βασικά στοιχεία, που πρέπει να έχει ένας αλγόριθμος, ώστε να θεωρείται γενετικός και δίνεται ένα παράδειγμα που κάνει αντιληπτή την εφαρμογή της θεωρίας.

Αρχικά σε ένα Γ.Α. πρέπει να υπάρχουν στοιχεία που θα τον συνδέουν με το πρόβλημα που επιλύει. Η κωδικοποίηση και η αντικειμενική συνάρτηση επιτελούν αυτό το σκοπό και είναι "εκ των ων ουκ άνευ" συστατικά για ένα Γ.Α.

Η κωδικοποίηση αφορά ένα σύνολο πιθανών λύσεων του προβλήματος. Η αναπαράσταση των λύσεων πρέπει να γίνει με ένα μαθηματικό, φορμαλιστικό τρόπο, ώστε να είναι δυνατή η επεξεργασία από τον υπολογιστή. Εξ' άλλου, κωδικοποίηση υπάρχει και στο φυσικό μοντέλο (χρωμοσώματα) και μάλιστα, όλες οι αλλαγές που παρατηρούνται στους οργανισμούς γίνονται πάνω στα κωδικοποιημένα χαρακτηριστικά των χρωμοσωμάτων. Κύριος στόχος της κωδικοποίησης είναι να αναπαριστά με

ικανοποιητικό τρόπο τα επιμέρους χαρακτηριστικά των λύσεων, ώστε να διευκολύνει τις επόμενες λειτουργίες του αλγορίθμου (κυρίως την επιλογή). Αποτέλεσμα της κωδικοποίησης πρέπει να είναι η ύπαρξη ομοιοτήτων ανάμεσα στα άτομα με σκοπό την κατάλληλη εκμετάλλευσή τους, διότι οι ομοιότητες βοηθούν την κατεύθυνση του ψαξίματος.

Διάφορα είναι τα είδη της κωδικοποίησης που μπορούν να γίνουν από πρόβλημα σε πρόβλημα. Η πιο απλή είναι η κωδικοποίηση με δυαδικά ψηφία (bits): κάθε λύση αναπαρίσταται από μια δυαδική συμβολοσειρά (binary string) καθορισμένου μήκους. Πάντως, έχουν αναφερθεί ποικίλες μορφές κωδικοποιήσεων, που κάθε μια εξαρτάται από το υπό εξέταση πρόβλημα. Καμιά δεν είναι αποτελεσματική για όλα τα προβλήματα, ενώ είναι πιθανό ένα πρόβλημα να επιδέχεται περισσότερες από μια κωδικοποιήσεις. Το σίγουρο είναι ότι η κωδικοποίηση είναι ένα κρίσιμο βήμα στην εφαρμογή του Γ.Α. και αν δεν είναι προσεκτική, πιθανότατα θα αποβεί μοιραία για την επιτυχία του. Η καταλληλότητα της κωδικοποίησης εξαρτάται σε μεγάλο βαθμό από τη διαίσθηση και την πείρα του σχεδιαστή. Συμβαίνει μερικές φορές μάλιστα, προφανείς τρόποι κωδικοποίησης να είναι λίγο (ή και καθόλου) αποτελεσματικοί.

Κατά συνέπεια προκύπτει το κρίσιμο ερώτημα: ποιοι είναι οι παράγοντες που καθορίζουν το είδος της κωδικοποίησης που πρέπει να επιλεγεί για ένα συγκεκριμένο πρόβλημα; Δεν υπάρχει ξεκάθαρη απάντηση που να καλύπτει όλες τις περιπτώσεις. Μερικές γενικού τύπου συμβουλές θα φανούν στην παραπέρα ανάπτυξη του θέματος.

Για παράδειγμα, έστω η συνάρτηση $f(x)=x^2$, με $x=[0,3]$ και x ακέραιος. Ζητείται το μέγιστο της συνάρτησης στο πεδίο ορισμού της. Για να λυθεί το πρόβλημα από ένα Γ.Α. πρέπει να επινοηθεί ένας τρόπος κωδικοποίησης των πιθανών λύσεων. Ο πιο προφανής και τελικά, όπως θα αποδειχθεί, πιο αποτελεσματικός τρόπος κωδικοποίησης είναι να αναπαρασταθεί η κάθε λύση με μια δυαδική συμβολοσειρά μήκους 5, που αριθμητικά θα ισοδυναμεί με την αντίστοιχη δεκαδική τιμή της λύσης. Έτσι καλύπτεται όλο το πεδίο ορισμού από τις 32 δυνατές συμβολοσειρές αυτού του είδους. Πχ η συμβολοσειρά 10010 αντιστοιχεί, κατά τα γνωστά, στην τιμή 18 του δεκαδικού συστήματος. Συνήθως, σε προβλήματα βελτιστοποίησης μαθηματικών συναρτήσεων, η δυαδική είναι η πιο βολική

και πιο αποδοτική κωδικοποίηση.

Το δεύτερο βασικό στοιχείο της σύνδεσης ενός Γ.Α. με το πρόβλημα που λύνει, είναι η αντικειμενική συνάρτηση. Αυτή παίρνει ως είσοδο μια αποκωδικοποιημένη συμβολοσειρά και επιστρέφει μια τιμή (συνήθως πραγματική), που είναι ανάλογη του πόσο καλά λύνει το πρόβλημα η συγκεκριμένη συμβολοσειρά. Η τιμή αυτή αποτελεί και τον καθοριστικό παράγοντα επιβίωσης και πολλαπλασιασμού ή όχι του ατόμου.

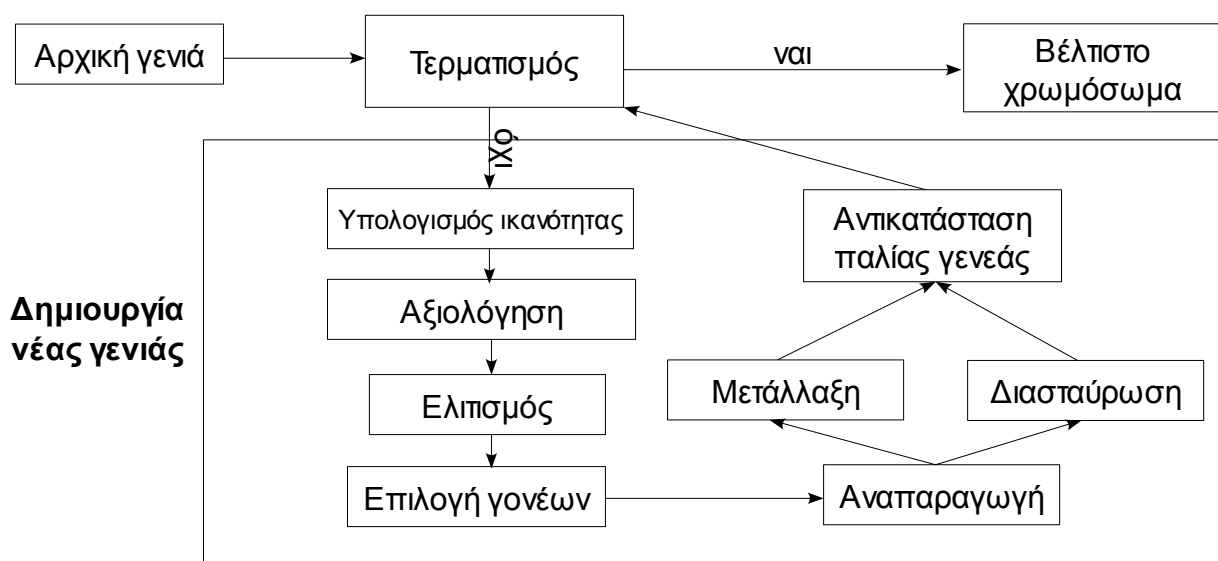
Η αντικειμενική συνάρτηση παίζει το ρόλο του περιβάλλοντος στο τεχνικό μοντέλο. Ουσιαστικά, είναι η μόνη πληροφορία που δέχεται ο αλγόριθμος για το πρόβλημα που λύνει. Είναι σημαντικό αυτή η συνάρτηση να είναι εύκολα υπολογίσιμη, ώστε να μην επιβραδύνει τους ρυθμούς της διαδικασίας.

Επιστρέφοντας στο παράδειγμα, η συνάρτηση ικανότητας του προβλήματος μεγιστοποίησης είναι φανερό ότι πρέπει να είναι η ίδια η f , γιατί ουσιαστικά το ζητούμενο είναι η μεγιστοποίηση αυτής της συνάρτησης. Έτσι σε κάθε λύση, δηλαδή σε κάθε πιθανή τιμή της μεταβλητής x , αντιστοιχεί μια τιμή ικανότητας ή απόδοσης (fitness ή score), μια τιμή που αξιολογεί το πόσο καλή είναι η λύση για τη μεγιστοποίηση της συνάρτησης και που, για αυτή την περίπτωση είναι η ίδια η εικόνα της στην f .

Με τον καθορισμό της κωδικοποίησης και της αντικειμενικής συνάρτησης ορίζεται πλέον το πρόβλημα και ολοκληρώνεται το πρώτο στάδιο εφαρμογής ενός Γ.Α. Αξίζει να σημειωθεί η αυτονομία και ανεξαρτησία αυτού του σταδίου από τα επόμενα μέρη. Οι λειτουργίες που ακολουθούν από εδώ και πέρα δεν εξαρτώνται από το πώς γίνεται η αναπαράσταση των ατόμων στο τεχνητό περιβάλλον και με ποιο τρόπο αξιολογούνται οι ικανότητές τους. Αυτό είναι σπουδαίο χαρακτηριστικό, διότι επιτρέπει την διαπραγμάτευση πολλών προβλημάτων με μια απλή αλλαγή στην συνάρτηση ικανότητας, ίσως και στην κωδικοποίηση. Η φάση ορισμού της κωδικοποίησης και της αντικειμενικής συνάρτησης υπάρχουν πάντα σε κάθε Γ.Α. ανεξαρτήτως του προβλήματος.

Στο επόμενο στάδιο περιλαμβάνονται λειτουργίες που ανήκουν στη φάση τρεξίματος του Γ.Α. Εδώ γίνεται ο κύριος όγκος της εργασίας και παράγεται το αποτέλεσμα της βελτιστοποίησης. Η δομή της λειτουργίας ενός Γ.Α. αποτελείται από τα παρακάτω βήματα:

1. Δημιουργία αρχικού πληθυσμού (Initialization)
2. Επαναληπτικό μέρος του αλγόριθμου (δημιουργεί νέες γενιές βάσει των προηγούμενων)
 - a) Υπολογισμός της ικανότητας των ατόμων (Fitness calculation)
 - b) Διαβάθμιση των ικανοτήτων (scaling)
 - c) Επιλογή των ικανότερων ατόμων και αντιγραφή τους στην επόμενη γενιά (ελπισμός).
 - d) Επιλογή γονέων
 - e) Αναπαραγωγή
 - Διασταύρωση ζευγαριών ατόμων (Crossover)
 - Μετάλλαξη μεμονωμένων ατόμων (Mutation)
 - f) Αντικατάσταση της παλιάς γενεάς με την καινούρια.
 - g) Ο αλγόριθμος ολοκληρώνεται όταν ικανοποιηθούν τα κριτήρια τερματισμού.



Η αρχικοποίηση είναι το βήμα στο οποίο ορίζεται ο αρχικός πληθυσμός, πάνω στον οποίο θα λάβουν χώρα οι λειτουργίες του Γ.Α. Ο πληθυσμός αυτός διαλέγεται με τυχαίο τρόπο ανάμεσα σε όλες τις δυνατές τιμές των μεταβλητών του προβλήματος, ενώ το μέγεθός του ορίζεται από το χρήστη (συνήθως, όμως, εξαρτάται από τους πόρους τους που έχει στη διάθεσή του). Σε μερικές μερικές υλοποιήσεις, η επιλογή των αρχικών σημείων γίνεται με ευρετικές μεθόδους, δίνοντας ένα εξ αρχής πλεονέκτημα στην αναζήτηση. Έστω στο παράδειγμά μας, ότι το μέγεθος του πληθυσμού είναι 4. Μένει να επιλεγθούν τυχαία τέσσερις συμβολοσειρές από τις 32 πιθανές. Αυτό μπορεί να γίνει με 20 διαδοχικές ρίψεις ενός τίμιου νομίσματος, ώστε να προκύψουν 4 συμβολοσειρές μήκους 5 η κάθε μία. Ένα πιθανό σενάριο θα μπορούσε να βγάλει τις συμβολοσειρές 01101, 11000, 01000 και 10011.

Αφού προκύψει η πρώτη γενιά, ο Γ.Α. εισέρχεται στο επαναληπτικό μέρος του. Ο πληθυσμός πρέπει να αξιολογηθεί, δηλαδή να μετρηθεί η ικανότητα επιβίωσης του κάθε ατόμου χωριστά. Για να συμβεί αυτό πρέπει να γίνει αποκωδικοποίηση χαρακτηριστικών και έπειτα υπολογισμός της απόδοσης των ατόμων. Ο παραλληλισμός με το φυσικό μοντέλο ίσως βοηθά στην κατανόηση αυτής της διαδικασίας: Στη φύση τα χρωμοσώματα ενός οργανισμού έχουν στα γονίδιά τους κωδικοποιημένα τα χαρακτηριστικά τους. Το σύνολο αυτής της κωδικοποιημένης γενετικής πληροφορίας ονομάζεται, όπως είπαμε, γονότυπος. Ο γονότυπος δεν είναι αντιληπτός με τις φυσικές αισθήσεις των έμβιων όντων. Αντίθετα, αντιληπτή γίνεται η αλληλεπίδραση του με το περιβάλλον, που έχει ως αποτέλεσμα την ορατή εμφάνιση των χαρακτηριστικών αυτών.

Ανάλογος είναι ο ρόλος της αποκωδικοποίησης στο τεχνητό μοντέλο. Εδώ το ρόλο του γονότυπου παίζει η δομή της συμβολοσειράς με τα δυαδικά ψηφία ως αντίστοιχο των γονιδίων. Ο φαινότυπος αναφέρεται στην παρατηρήσιμη εμφάνιση μιας συμβολοσειράς, δηλαδή στο πώς φαίνεται στο περιβάλλον της. Περιβάλλον όμως, θεωρείται η αντικειμενική συνάρτηση, άρα ο φαινότυπος μιας συμβολοσειράς αντιστοιχεί στην αποκωδικοποιημένη τιμή του, που ανήκει στο σύνολο ορισμού της αντικειμενικής συνάρτησης.

Σκοπός της λειτουργίας αξιολόγησης είναι να υπολογιστεί για κάθε άτομο του πληθυσμού

η ικανότητα του για επιβίωση. Στη φύση οι ικανότητες των ατόμων δεν μπορούν να προσδιοριστούν με αυστηρό τρόπο. Είναι όμως καθορισμένες από το γενετικό υλικό των χρωμοσωμάτων τους. Εύκολα πάντως θα μπορούσε κανείς να ισχυριστεί, πχ για τα ζώα, ότι μεγαλύτερη τύχη για επιβίωση έχουν όσα μπορούν να ξεφεύγουν από άρπαγες, να αντέχουν σε αρρώστιες και γενικά να αντιπαρέρχονται τις όποιες αντιξοότητες παρουσιάζονται κατά τη διάρκεια της ζωής τους. Συνεπώς, ο υπολογισμός της ικανότητας είναι θεμελιώδης λειτουργία για το Γ.Α. Η εφαρμογή της είναι πολύ απλή (τουλάχιστον για απλά προβλήματα): για κάθε συμβολοσειρά του τρέχοντος πληθυσμού υπολογίζεται η απόδοσή της από την ήδη γνωστή αντικειμενική συνάρτηση. Σε πιο σύνθετα προβλήματα, ο υπολογισμός ικανότητας μπορεί να ισοδυναμεί με την εκτέλεση μιας εργαστηριακής προσομοίωσης.

Μετά από την παραγωγή μερικών γενεών, ο Γ.Α. αρχίζει σιγά σιγά να συγκλίνει προς το βέλτιστο σημείο του χώρου αναζήτησης. Ο όρος “σύγκλιση”, όσον αφορά τα άτομα του πληθυσμού, σημαίνει ότι τα περισσότερα από αυτά μοιάζουν αρκετά μεταξύ τους. Άμεσο αποτέλεσμα αυτού είναι οι ικανότητές τους να διαφέρουν από ελάχιστα έως καθόλου και κατά συνέπεια στην επιλογή υπάρχει πολύ μεγάλος ανταγωνισμός για το ποιος θα δώσει απογόνους στην επόμενη γενιά. Έτσι, τα άτομα που είναι τα πιο ικανά, έστω και με ελάχιστη διαφορά, δεν κατορθώνουν -κατά πάσα πιθανότητα- να επιβληθούν μέσα στο περιβάλλον τους και η πρόοδος του αλγορίθμου από ένα σημείο και πέρα γίνεται με πολύ αργούς ρυθμούς, χωρίς όμως η σύγκλιση να γίνεται προς το βέλτιστο σημείο, αλλά προς ένα τοπικό ακρότατο. Το φαινόμενο αυτό ονομάζεται πρόωρη σύγκλιση (premature convergence).

Η πρόωρη σύγκλιση μπορεί να αποφευχθεί με χρήση μηχανισμών διαβάθμισης (scaling mechanisms). Πρόκειται για τεχνικές που έχουν στόχο την επαύξηση των διαφορών - όσο μικρές κι αν είναι - των ικανοτήτων των ατόμων και την πιο εύκολη ανάδειξη των καλύτερων μέσα από την λειτουργία της επιλογής. Με τους μηχανισμούς διαβάθμισης εγκαταλείπεται το κλασικό μοντέλο της ρουλέτας και πλέον οι ευκαιρίες των ατόμων στην επιλογή δεν είναι ανάλογες της ικανότητάς τους. Οι κυριότεροι μηχανισμοί που σήμερα επικρατούν είναι:

- Γραμμική διαβάθμιση (Linear scaling).
- Εκθετική διαβάθμιση (proportional scaling).
- Διαβάθμιση με βάση τη σειρά (Fitness ranking).
- Παραθυροποίηση (Windowing).

Στη γραμμική διαβάθμιση οι τιμές των ικανοτήτων του πληθυσμού μετασχηματίζονται σύμφωνα με μια γραμμική σχέση της μορφής: $s = a \cdot f + \beta$ όπου τα a και β υπολογίζονται με βάση τις εξής δύο συνθήκες:

- ο μέσος όρος των ικανοτήτων πριν και μετά τον μετασχηματισμό να μην αλλάζει και
- οι μεγαλύτερες ικανότητες του πληθυσμού να μην υπερβαίνουν ένα προκαθορισμένο πολλαπλάσιο του μέσου όρου (συνήθως διπλάσιο).

Ανάλογος είναι και ο τρόπος λειτουργίας της εκθετικής διαβάθμισης. Τα άτομα σχηματίζουν μια ταξινομημένη λίστα με βάση τις ικανότητές τους και η νέα τους ικανότητα, η εκθετικά διαβαθμισμένη ικανότητα s , προκύπτει εφαρμόζοντας τον μετασχηματισμό $s = f^k$ όπου k συνήθως επιλέγεται να είναι η σειρά του κάθε ατόμου στη ταξινομημένη λίστα. Γενικά το k μπορεί να ποικίλει από πρόβλημα σε πρόβλημα. Μπορεί ακόμη και να αλλάξει κατά το χρόνο εκτέλεσης για να περιορίζει ή να διευρύνει την ποικιλία ανάλογα με τις ανάγκες.

Στη διαβάθμιση με βάση τη σειρά, στους μετασχηματισμούς που λαμβάνουν χώρα, δε χρησιμοποιείται καμία παράμετρος. Τα άτομα ταξινομούνται με βάση τις ικανότητές τους από το μικρότερο προς το μεγαλύτερο. Αποδίδεται στο μικρότερο μια νέα ικανότητα και συνεχίζοντας διαδοχικά προς το μεγαλύτερο, οι νέες ικανότητες αυξάνουν κατά μία σταθερή τιμή.

Η παραθυροποίηση είναι τεχνική σύμφωνα με την οποία οι τιμές των ικανοτήτων τροποποιούνται πριν από την επιλογή ως εξής: Αφαιρείται από την ικανότητα κάθε ατόμου ποσό ίσο με την μικρότερη ικανότητα που υπάρχει στον πληθυσμό. Έτσι, οι διαφορές ικανοτήτων των ατόμων γίνονται πιο εμφανείς. Επειδή όμως κατ' αυτόν τον τρόπο, τα λιγότερο ικανά άτομα έχουν μηδαμινές ευκαιρίες επιλογής, μπορεί να τεθεί ένα κάτω όριο των ικανοτήτων μεγαλύτερου του μηδενός.

Με την επιλογή, βρίσκει εφαρμογή στα πλαίσια του αλγόριθμου, ο νόμος της επιβίωσης του ικανότερου. Μέσω αυτής της διαδικασίας, καθορίζεται ποια άτομα από τον υπάρχοντα πληθυσμό θα έχουν την ευκαιρία να λάβουν μέρος στην αναπαραγωγή και να κληροδοτήσουν στην επόμενη γενιά μέρος ή το σύνολο των χαρακτηριστικών τους. Στόχος της λειτουργίας της επιλογής είναι να επιτρέπει εκθετική αύξηση των ικανότερων ατόμων και τελικά, μετά από αναπαραγωγή αρκετών γενεών, την επικράτησή τους. Ένας Γ.Α. χωρίς επιλογή στην αναπαραγωγική του διαδικασία ισοδυναμεί με τυχαίο ψάξιμο.

Υπάρχουν διάφοροι τρόποι υλοποίησης της επιλογής στα πλαίσια ενός Γ.Α. Δεδομένου όμως, ότι στη βασική μορφή του αλγόριθμου το μέγεθος του πληθυσμού από γενιά σε γενιά δεν αλλάζει, κάθε τεχνική επιλογής, για να δικαιώνει τον τίτλο της, οφείλει να δίνει με κάποιο τρόπο μεγαλύτερες πιθανότητες αναπαραγωγής σε άτομα που αξιολογούνται μέσα στο τεχνητό περιβάλλον ως τα πιο ικανά.

Ο τελεστής αναπαραγωγής μπορεί να εκφραστεί σε αλγοριθμική βάση, με πολλούς τρόπους. Ίσως ο ευκολότερος από αυτούς είναι η έκφραση μέσω μιας εξαναγκασμένης ρουλέτας, στην οποία κάθε συμβολοσειρά ενός πληθυσμού αντιπροσωπεύεται σε ένα μέρος της ρουλέτας, σε αναλογία με την απόδοσή της. Για να εξηγήσουμε τη χρήση της εξαναγκασμένης ρουλέτας, θεωρούμε τον πληθυσμό των τεσσάρων συμβολοσειρών, που έχουμε δημιουργήσει με τη ρίψη ενός νομίσματος 20 φορές. Έστω, ότι έχουμε μετρήσει την απόδοση (δηλαδή την τιμή της αντικειμενικής συνάρτησης), για κάθε συμβολοσειρά, όπως φαίνεται στον παρακάτω πίνακα:

Αριθμός συμβολοσειράς	Συμβολοσειρά	Απόδοση	Απόδοση %
1	01101	169	14,4
2	11000	576	49,2
3	01000	64	5,5
4	10011	361	30,9
Σύνολο		1170	100,0

Αθροίζοντας την απόδοση των τεσσάρων συμβολοσειρών παίρνουμε άθροισμα 1170. Το ποσοστό κάθε συμβολοσειράς στη συνολική απόδοση του πληθυσμού φαίνεται στην τελευταία στήλη του πίνακα. Αυτή η αντιστοιχία στην εξαναγκασμένη ρουλέτα γι' αυτή τη γενιά αναπαραγωγής φαίνεται στο παρακάτω σχήμα.

Για να γίνει η αναπαραγωγή, στρίβουμε τη ρουλέτα τέσσερις φορές. Για το συγκεκριμένο πρόβλημα η συμβολοσειρά No.1 έχει απόδοση 169, η οποία αντιπροσωπεύει το 14.4% της συνολικής απόδοσης. Σαν αποτέλεσμα η συμβολοσειρά No.1 αντιστοιχεί στο 14.4 της επιφάνειας της ρουλέτας και σε κάθε στρίψιμο της ρουλέτας θα δώσει ως αποτέλεσμα αυτή τη συμβολοσειρά με πιθανότητα 0.144. Με αυτόν τον τρόπο, οι συμβολοσειρές που έχουν τη μεγαλύτερη απόδοση, θα έχουν μεγαλύτερο αριθμό αντιγράφων (απογόνων) στην επόμενη γενιά, ενώ αυτές που έχουν χαμηλή απόδοση δεν θα υπάρχουν. Όταν μια συμβολοσειρά επιλεγεί, δημιουργείται ένα ακριβές αντίγραφο της και μαζί με τα αντίγραφα άλλων συμβολοσειρών, που παράγονται με τον ίδιο τρόπο, δημιουργείται ένας νέος δοκιμαστικός πληθυσμός, ο οποίος θα υποστεί περισσότερες γενετικές διαδικασίες. Αυτός ο νέος πληθυσμός αναφέρεται στη διεθνή βιβλιογραφία και σαν δεξαμενή ζευγαρώματος (mating pool).

Κατά τη διαδικασία της επιλογής ένα απευκταίο, αλλά όχι απίθανο ενδεχόμενο είναι να μην επιλεγεί το ικανότερο άτομο του πληθυσμού. Προκειμένου να αποφευχθεί κάτι τέτοιο εφαρμόζεται μια τεχνική που ονομάζεται *ελιτισμός* (elitism). Με τον ελιτισμό αντιγράφεται “χαριστικά” το ικανότερο άτομο στην επόμενη γενιά, προτού καν αρχίσει η επιλογή και εξασφαλίζεται έτσι η επιβίωσή του.

Η τεχνική του ελιτισμού ενέχει τον κίνδυνο της γρήγορης κυριαρχίας του πληθυσμού από ένα άτομο, το οποίο ναι μεν έχει συνήθως μεγάλη ικανότητα, αλλά δεν αποτελεί προϊόν εξαντλητικού ψαξίματος και αποδοτικής εργασίας του αλγορίθμου και τις περισσότερες φορές δεν είναι το βέλτιστο σημείο του χώρου αναζήτησης. Γι' αυτόν ακριβώς το λόγο, και σύμφωνα με τη διδακτορική εργασία του De Jong, ο ελιτισμός είναι προτιμότερο να εφαρμόζεται σε περιπτώσεις τοπικής αναζήτησης, όπου όντως μπορεί να βελτιώσει αξιοσημείωτα την απόδοση.

Ο προσωρινός πληθυσμός που προέκυψε από τη διαδικασία της επιλογής πρέπει να περάσει από τη διαδικασία ζευγαρώματος για να πραγματοποιηθεί ένα είδος γονιμοποίησης, όπως συμβαίνει και στη φύση. Η νέα λοιπόν ομάδα ατόμων, που προέκυψε από την επιλογή, σχηματίζει με τυχαίο τρόπο ομάδες των δύο. Σε κάθε ομάδα, τα δύο μέλη παίρνουν μέρος σε μια απλή λειτουργία ανταλλαγής γενετικού υλικού που ονομάζεται διασταύρωση. Η διασταύρωση είναι μια απαραίτητη λειτουργία που συμβάλει αποφασιστικά στην επίδοση ενός Γ.Α. Εξ' αιτίας αυτής της σπουδαιότητας, έχει γίνει αρκετή έρευνα και έχουν επινοηθεί πολλοί τρόποι υλοποίησης της. Μερικοί μπορούν να εφαρμοστούν σε κάθε τύπο προβλήματος, ενώ άλλοι είναι πιο κατάλληλοι και εξειδικευμένοι για ειδικές περιπτώσεις. Στόχος της διασταύρωσης είναι η νέα γενιά που θα προκύψει μετά την εφαρμογή της να περιλαμβάνει άτομα που θα διαφέρουν από τους γονείς τους και θα φέρουν συνδυασμό των καλύτερων χαρακτηριστικών τους. Ερευνητές που ασχολούνται χρόνια με τους Γ.Α. υποστηρίζουν ότι αν αφαιρεθεί η διασταύρωση από έναν Γ.Α., τότε μειώνεται σημαντικά η απόδοσή του, αλλά αυτή δεν είναι μια άποψη με καθολική αποδοχή.

Ένα ενδεικτικό της χρησιμότητας της διασταύρωσης είναι η ανακατεύθυνση του ψαξίματος σε νέες "απάτητες" περιοχές του χώρου αναζήτησης. Έτσι διευρύνεται το πεδίο δράσης του αλγορίθμου και αυξάνουν οι πιθανότητες επιτυχίας του. Επίσης, τα νέα άτομα περιλαμβάνουν συνδυασμούς χαρακτηριστικών των γονέων τους και με αυτό τον τρόπο μπορούν να προκύψουν επιτυχημένοι συνδυασμοί υψηλής ικανότητας. Υπάρχει βέβαια το ενδεχόμενο, η διασταύρωση να δώσει χειρότερα παιδιά από τους γονείς, αλλά αυτά δεν θα έχουν μεγάλη πιθανότητα πολλαπλασιασμού στον επόμενο αναπαραγωγικό κύκλο, λόγω μικρής απόδοσης.

Στην πράξη, η διασταύρωση χρησιμοποιείται με παραμετροποιημένη μορφή, δηλαδή λαμβάνει χώρα με πιθανότητα που καθορίζεται από το σχεδιαστή του Γ.Α. Συνήθως, αυτή η πιθανότητα ποικίλει από πρόβλημα σε πρόβλημα, ενώ είναι δυνατό και να αλλάζει κατά το χρόνο τρεξίματος.

Τελευταία στον κύκλο αναπαραγωγικής διαδικασίας και ίσως λιγότερο σημαντική, αλλά πάντως χρήσιμη, είναι η μετάλλαξη. Είναι μια λειτουργία που όταν συμβαίνει αραιά στη

φύση δρα βελτιωτικά για τους οργανισμούς και γενικά για την εξέλιξη της ζωής. Ανάλογος είναι ο ρόλος της και στα τεχνικά περιβάλλοντα. Η λειτουργία της είναι απλή: Ενεργεί σε ένα μόνο οργανισμό κάθε φορά και αφού επιλεγθεί τυχαία ένα ψηφίο, αυτό αντιστρέφεται (από 0 σε 1 ή το αντίστροφο) και προκύπτει ένας απόγονος. Είναι πολύ σημαντικό η πιθανότητα να πραγματοποιηθεί η μετάλλαξη να είναι αρκετά μικρή (περίπου μία μετάλλαξη σε κάθε χίλια ψηφία που αντιγράφονται), γιατί σε αντίθετη περίπτωση ο Γ.Α. εκφυλίζεται σε τυχαίο ψάξιμο.

Αν και υπάρχει κάποια σύγχυση για το ρόλο της μετάλλαξης, τόσο της φυσικής όσο και της τεχνητής, το σίγουρο είναι πως είναι απαραίτητη. Η μετάλλαξη λειτουργεί ως ασφαλιστική δικλείδα για τις περιπτώσεις όπου η επιλογή και η διασταύρωση χάσουν κάποιες πολύτιμες γενετικές πληροφορίες. Όταν συμβαίνει, επιφέρει ποικιλία στον πληθυσμό, ανακατευθύνει την αναζήτηση και εξασφαλίζει ότι κανένα σημείο του χώρου αναζήτησης δεν αποκλείεται από τη διαδικασία του ψαξίματος.

2.6 ΠΑΡΑΜΕΤΡΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Οι διάφορες φάσεις που ανήκουν στο επαναληπτικό μέρος ενός γενετικού αλγόριθμου υλοποιούνται με ποικίλες συναρτήσεις και μαθηματικές μεθόδους. Στην παράγραφο αυτή θα περιγραφούν οι κυριότερες από αυτές καθώς και οι παράγοντες που καθορίζουν τη σύσταση του πληθυσμού.

Συναρτήσεις διαβάθμισης ικανοτήτων (fitness scaling functions)

1. Shift linear (γραμμική διαβάθμιση)

2. Proportional (εκθετική διαβάθμιση)

Οι τρεις πρώτες μέθοδοι έχουν αναλυθεί στην προηγούμενη παράγραφο

3. Rank (διαβάθμιση με βάση τη σειρά)

4. Top: Τα άτομα με την καλύτερη απόδοση διαβαθμίζονται ισότιμα. Το πλήθος τους καθορίζεται από τον χρήστη. Τα άτομα αυτά έχουν ίσες πιθανότητες αναπαραγωγής σε αντίθεση με όλα τα υπόλοιπα άτομα του πληθυσμού που έχουν μηδενικές πιθανότητες.

Συναρτήσεις επιλογής (selection functions)

1. Roulette (έχει αναλυθεί στην προηγούμενη παράγραφο).
2. Stochastic uniform: Δημιουργείται μία ευθεία και κάθε άτομο αντιστοιχεί σε μήκος αυτής ανάλογο της διαβαθμισμένης ικανότητάς του. Ο αλγόριθμος μετακινείται πάνω στην ευθεία με ίσα βήματα, ένα βήμα για κάθε γονέα. Ο γονέας επιλέγεται ανάλογα με την περιοχή στην οποία κατέληξε το κάθε βήμα.
3. Tournament: Επιλέγονται τυχαία άτομα που συγκροτούν μία ομάδα (το πλήθος των ατόμων έχει οριστεί από το χρήστη). Γονέας είναι το άτομο με την καλύτερη απόδοση στην κάθε ομάδα.

Συναρτήσεις διασταύρωσης (crossover functions)

1. Single point (διασταύρωση μονού σημείου): Επιλέγεται στη τύχη ένας ακέραιος n από το 1 έως και το πλήθος των μεταβλητών. Στον απόγονο τα γονίδια 1 έως n προέρχονται από τον πρώτο γονέα και τα γονίδια $n+1$ έως (πλήθος μεταβλητών) από τον δεύτερο. Για παράδειγμα $p_1=[a b c d e f g h]$, $p_2=[1 2 3 4 5 6 7 8]$ και $n=3$ οπότε $child=[a b c 4 5 6 7 8]$.
2. Two point (διασταύρωση διπλού σημείου): Πρόκειται για την εφαρμογή της διασταύρωσης μονού σημείου δύο φορές, αφού πρώτα επιλεγούν με τυχαίο τρόπο δύο θέσεις κοπής. Για παράδειγμα $p_1=[a b c d e f g h]$, $p_2=[1 2 3 4 5 6 7 8]$, $n=3$ και $m=6$ οπότε $child=[a b c 4 5 6 g h]$.
3. Scattered: Δημιουργείται με τυχαίο τρόπο μία φόρμα δυαδικών αριθμών. Στις θέσεις της φόρμας με τιμή 1 επιλέγεται το αντίστοιχο γονίδιο από τον πρώτο γονέα και στις θέσεις με τιμή 0 από τον δεύτερο. Για παράδειγμα $p_1=[a b c d e f g h]$, $p_2=[1 2 3 4 5 6 7 8]$ και $RandomCrossoverVector=[1 1 0 0 1 0 0 0]$ οπότε $child=[a b 3 4 e 6 7 8]$.
4. Arithmetic (διασταύρωση μέσου όρου): Ενεργεί πάνω σε δυο γονείς και παράγει ένα νέο άτομο, που είναι το αποτέλεσμα υπολογισμού του αριθμητικού μέσου όρου των γονέων του. Μερικές φορές μάλιστα, οι δυο γονείς δεν μετέχουν ισότιμα, αλλά με

βάρη στον υπολογισμό. Για την πραγματική κωδικοποίηση η διασταύρωση μέσου όρου δουλεύει πολύ καλά, πέρα από το γεγονός ότι ταιριάζει διαισθητικά με την μορφή των ατόμων, και είναι αρκετά δημοφιλής στις εφαρμογές.

Συναρτήσεις Μετάλλαξης (mutation functions)

1. Gaussian: Αφορά χρωμοσώματα πραγματικών ακεραίων. Σε κάθε γονίδιο ενός επιλεγμένου ατόμου προστίθεται ένας τυχαίος αριθμός που προκύπτει από μια κατανομή Gauss. Η κατανομή αυτή έχει κεντρική τιμή το 0 και απόκλιση η οποία καθορίζεται από το χρήστη και φθίνει από γενιά σε γενιά με συγκεκριμένο ρυθμό.
2. Uniform: Επιλέγονται από κάποιο άτομο τα γονίδια προς μετάλλαξη. Κάθε γονίδιο έχει συγκεκριμένη μικρή πιθανότητα να επιλεγεί. Στα επιλεγμένα γονίδια δίνεται μία τυχαία τιμή από το πεδίο ορισμού τους.

Σύνθεση πληθυσμού (παράμετροι elite count και crossover fraction)

Μπορούμε να καθορίσουμε τη σύνθεση ενός πληθυσμού μεταβάλλοντας τους παράγοντες elite count και crossover fraction. Συγκεκριμένα, τα άτομα ενός πληθυσμού προκύπτουν από αυτά της προηγούμενης γενιάς είτε με διασταύρωση, είτε με μετάλλαξη, είτε από τα καλύτερα μέλη της που αντιγράφονται χαριστικά (ελιτισμός).

Σε ένα πληθυσμό 300 ατόμων με παράμετρο ελιτισμού (elite count) 8 και ποσοστό διασταύρωσης (crossover fraction) 0.85 κάθε νέα γενιά προκύπτει ως εξής από της προηγούμενη:

- τα 8 πιο πετυχημένα άτομα αντιγράφονται κατευθείαν στη νέα γενιά.
- από τα υπόλοιπα 292 άτομα τα 248 (292×0.85) θα προκύψουν από διασταύρωση μελών της προηγούμενης γενιάς και τα 44 (292×0.15) από μετάλλαξη.

2.7 ΤΕΧΝΙΚΕΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ

Η δυαδική κωδικοποίηση που χρησιμοποιήθηκε στο βασικό σχήμα των Γ.Α, αν και αρκετά ευέλικτη, δεν είναι μοναδική και κατάλληλη για όλες τις περιπτώσεις προβλημάτων. Η έρευνα και ο προγραμματισμός έχουν οδηγήσει και σε άλλες τεχνικές κωδικοποίησης (πχ. πραγματική) που αποδεικνύονται αποδοτικές για ορισμένους τύπους προβλημάτων. Πολλά πεδία εφαρμογής των Γ.Α, όπως η βελτιστοποίηση μηχανικού σχεδιασμού (engineering design optimization) ή η συνδυαστική βελτιστοποίηση (combinatorial optimization) επιβάλλουν από τη φύση τους διαφορετικές κωδικοποιήσεις.

Μέσα από το πλήθος και την ποικιλία των κωδικοποιήσεων που έχουν προταθεί, ξεχωρίζουν δύο κανόνες-αρχές, που αποτελούν καλούς οδηγούς για την κατασκευή μιας επιτυχημένης κωδικοποίησης:

1. Η αρχή της Σημασίας των Δομικών Φορμών (Meaningful Building Blocks principle)
2. Η αρχή του Ελάχιστου Αλφάβητου (Minimal Alphabet principle).

Η αρχή της Σημασίας των Δομικών Φορμών συνοψίζεται ως εξής: Κύριος στόχος της κωδικοποίησης πρέπει να είναι οι ομοιότητες ανάμεσα σε άτομα μέσα στο περιβάλλον του προβλήματος να συνεπάγονται παραπλήσιες επιδόσεις. Βασικός εκφραστής αυτής της τάσης είναι η δυαδική κωδικοποίηση που είναι και η επικρατέστερη γενικά στους Γ.Α. Η ύπαρξη ομοιοτήτων και κατά συνέπεια δομικών φορμών είναι το χαρακτηριστικό πάνω στο οποίο οικοδομείται όλη η φιλοσοφία της αρχής. Η θεωρία των σχημάτων και, βεβαίως το θεμελιώδες συμπέρασμα των Γ.Α., αναπτύχθηκε με κωδικοποιήσεις αυτής της κατηγορίας. Πάντως, αν κάποιος επιθυμεί να αξιοποιήσει τα πλεονεκτήματα της θεωρίας των σχημάτων και επιλέξει κωδικοποίηση που κατ' εξοχήν την υποστηρίζει, οφείλει να γνωρίζει ότι πέρα από την δυαδική, οποιαδήποτε άλλη κωδικοποίηση απαιτεί συνήθως φαντασία και λίγη "τέχνη" από τον εμπνευστή της για να αποδειχτεί επιτυχημένη.

Στην αρχή του Ελάχιστου Αλφάβητου κύριο μέλημα, όπως εύκολα γίνεται αντιληπτό, είναι η υιοθέτηση του μικρότερου δυνατού αλφάβητου. Αυτό οδηγεί πιο εύκολα σε ανάδειξη των ιδιαίτερων χαρακτηριστικών των δυαδικών συμβολοσειρών και καλύτερη εκμετάλλευση

των ομοιοτήτων τους. Το ελάχιστο αλφάβητο που μπορεί να υπάρξει είναι το δυαδικό, επομένως για άλλη μια φορά επιβεβαιώνεται η μεγάλη αξία της δυαδικής κωδικοποίησης. Είναι εύκολο να δειχθεί με μαθηματικό τρόπο, ότι με το δυαδικό αλφάβητο επιτυγχάνεται ο μέγιστος αριθμός σχημάτων ανά δυαδικό ψηφίο ανεξάρτητα με τις όποιες επιμέρους σχεδιαστικές λεπτομέρειες της κωδικοποίησης, που μπορεί να υπάρχουν.

Εξετάζοντας άλλες μορφές κωδικοποίησης που χρησιμοποιούνται συχνά θα ήταν παράλειψη να μην αναφερθούν η πραγματική (real) και η κωδικοποίηση ταξινομημένης λίστας (ordinal ή order-based list).

Η πρώτη χρησιμοποιείται περισσότερο στο μηχανικό σχεδιασμό (engineering design), κάνοντας μ' αυτό τον τρόπο πιο κατανοητό και ελκυστικό το Γ.Α. στους μηχανικούς-χρήστες. Η κωδικοποίηση συνίσταται στη χρησιμοποίηση ενός αριθμού πραγματικών παραμέτρων που περιγράφουν τις διάφορες σχεδιαστικές επιλογές. Σημαντικό πλεονέκτημα είναι ότι έχει καλύτερη απόδοση σε περιπτώσεις που ο χώρος αναζήτησης είναι τεράστιος και η δυαδική κωδικοποίηση δεν παράγει ικανοποιητικά αποτελέσματα. Λόγω της απλότητας που τη χαρακτηρίζει επιτρέπει με μεγαλύτερη άνεση συμμετοχή του αλγόριθμου σε υβριδικά σχήματα.

Η κωδικοποίηση ταξινομημένης λίστας βρίσκει μεγάλο πεδίο εφαρμογής στα συνδυαστικά προβλήματα (πχ. Traveling Salesman Problem, Graph Coloring Problem, κτλ.), των οποίων η φύση απαγορεύει τις προηγούμενες κωδικοποιήσεις. Κάθε πιθανή λύση αναπαρίσταται με διατάξεις κόμβων που περιγράφουν πλήρως τον τρόπο επίλυσης (πχ. για το TSP με τρεις πόλεις μια πιθανή λύση θα μπορούσε να είναι η ΑΒΓ, δηλώνοντας ότι η σειρά επίσκεψης των πόλεων πρέπει να είναι: πρώτη η Α, δεύτερη η Β και τρίτη η Γ). Ιδιαίτερη προσοχή πρέπει να δοθεί στις άλλες λειτουργίες του Γ.Α. (διασταύρωση, μετάλλαξη) που πλέον δε μπορούν να δράσουν με τη γνωστή τους μορφή και πρέπει να προσαρμοστούν στα δεδομένα της κωδικοποίησης (έχουν γίνει πολλές προτάσεις για το συγκεκριμένο θέμα).

Μεγάλο ενδιαφέρον για την Τεχνητή Νοημοσύνη παρουσιάζουν κωδικοποιήσεις στις οποίες τα άτομα είναι Νευρωνικά Δίκτυα.

Πάντως, γενικά αν σε κάποιο πρόβλημα δεν είναι προφανές το είδος της κωδικοποίησης και γίνει μια κακή επιλογή δε σημαίνει ότι ο αλγόριθμος θα οδηγηθεί οπωσδήποτε σε αποτυχία. Οι Γ.Α. είναι από τη φύση τους αρκετά εύρωστοι, αντέχουν σε αλλαγές και "συγχωρούν" λάθη που δε ξεφεύγουν κατά πολύ από τα όριά τους. Επιπλέον, έχουν το πλεονέκτημα της χωρίς ιδιαίτερο κόπο αλλαγής της κωδικοποίησης.

Σε περιπτώσεις που οι μεταβλητές του προβλήματος είναι παραπάνω από μία εφαρμόζεται συνένωση (concatenation) τους σε μια συμβολοσειρά. Όμως, το ποια ακριβώς θα είναι η θέση κάθε μεταβλητής μέσα στη συμβολοσειρά είναι ένα θέμα που απαιτεί ιδιαίτερη προσοχή. Το είδος των αλληλεπιδράσεων ανάμεσα στις μεταβλητές παίζει σημαντικό ρόλο: Αν οι μεταβλητές είναι γραμμικά ανεξάρτητες δεν ενδιαφέρει η σχετική θέση τους μέσα στη συμβολοσειρά. Αν όμως υπάρχει εξάρτηση, τότε είναι καλύτερα να τοποθετούνται οι ανεξάρτητες μεταβλητές κοντά η μία στην άλλη, ώστε να είναι σε θέση ο Γ.Α. να ανακαλύψει τις δομικές φόρμες και κατά συνέπεια να αναδείξει τα καλύτερα άτομα. Αυτό γίνεται πιο εύκολα αντιληπτό αν σκεφτεί κανείς ότι η διασταύρωση θα χώριζε με μεγαλύτερες πιθανότητες μεταβλητές που απέχουν αρκετά μέσα στην δομή της συμβολοσειράς.

2.8 ΕΝΣΩΜΑΤΩΣΗ ΠΕΡΙΟΡΙΣΜΩΝ

Πολύ συχνά στα προβλήματα βελτιστοποίησης που καλούνται να δώσουν λύσεις οι Γ.Α., οι αντικειμενικές συναρτήσεις έχουν κάποιους περιορισμούς, όσον αφορά το πεδίο ορισμού τους, που πρέπει αυστηρά να ικανοποιούνται. Το γεγονός αυτό πρέπει να λαμβάνεται υπ' όψη στο σχεδιασμό ενός Γ.Α., ώστε να είναι δυνατός ο εντοπισμός νόμιμων λύσεων.

Οι περιορισμοί στις αντικειμενικές συναρτήσεις εκφράζονται συνήθως με δύο τρόπους: με ισότητες και ανισότητες (ή άνισο-ισότητες). Ωστόσο, αυτές οι δύο κατηγορίες περιορισμών μπορούν να αναχθούν σε μία, αφού κάθε ισότητα μπορεί να μετατραπεί σε δύο ανισότητες (πχ. η ικανοποίηση της ισότητας $g(x)=0$ μετατρέπεται στην ταυτόχρονη ικανοποίηση των ανισοτήτων $g(x)>0$ και $g(x)<0$). Συνεπώς, μιλώντας για περιορισμούς εννοούμε ανισότητες.

Εκ πρώτης όψεως, ίσως να δημιουργείται η απορία γιατί προκαλεί αναστάτωση ο χειρισμός των περιορισμών. Θα μπορούσε κάποιος να υποστηρίξει ότι, όταν προκύπτουν λύσεις που είναι έξω από το πεδίο ορισμού της συνάρτησης ικανότητας, να αγνοούνται από τον αλγόριθμο (αφού δεν ορίζεται γι' αυτές τιμή απόδοσης) και στη θέση τους να παράγονται άλλες που είναι αποδεκτές.

Η ιδέα αυτή δεν είναι άσχημη, αλλά αποδεικνύεται ανεπαρκής για περιπτώσεις προβλημάτων με πολύ μεγάλο αριθμό περιορισμών, στα οποία ο εντοπισμός ενός νόμιμου σημείου μπορεί να είναι ανάλογης δυσκολίας με τον εντοπισμό του ζητούμενου βέλτιστου σημείου. Σε τέτοια προβλήματα συμφέρει τα μη νόμιμα σημεία να αντιμετωπίζονται ως σημεία χαμηλής απόδοσης και να μην αγνοούνται εντελώς. Αντιμετωπίζονται δηλαδή τα μη νόμιμα σημεία ως φορείς πληροφορίας που αξιοποιείται αποδοτικά από τον Γ.Α. Αυτό υλοποιείται με την εκχώρηση σε αυτά αποδόσεων χαμηλής απόδοσης που είναι ανάλογη του βαθμού παραβίασης των περιορισμών. Η μέθοδος αυτή ενσωμάτωσης των περιορισμών σε ένα Γ.Α. ονομάζεται μέθοδος της ποινής (penalty method).

Με την εφαρμογή αυτής της μεθόδου ένα πρόβλημα με περιορισμούς εύκολα μετατρέπεται σε ένα ισοδύναμο χωρίς περιορισμούς: απλά κάθε παραβίαση περιορισμού συσχετίζεται με ένα κόστος (μια ποινή).

2.9 ΓΝΩΣΤΕΣ ΕΦΑΡΜΟΓΕΣ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Στα πρώτα τους βήματα οι Γ.Α. αποτέλεσαν αντικείμενο μελέτης και ανάπτυξης σε πανεπιστήμια και ερευνητικά κέντρα. Τα τελευταία χρόνια όμως, οι μεγάλες ανάγκες για δημιουργία αποδοτικών εφαρμογών στο χώρο της βελτιστοποίησης, σε συνδυασμό με την πολλά υποσχόμενη τεχνολογία των Γ.Α. ώθησαν στη χρήση του Γενετικού Προγραμματισμού (Genetic Programming) σε πολλές εφαρμογές ενός ευρύτατου φάσματος πεδίων με εντυπωσιακά αποτελέσματα. Σήμερα υπάρχουν και λειτουργούν με επιτυχία πολλά συστήματα βασισμένα σε Γ.Α. σε τομείς όπως η Επεξεργασία Εικόνας (Image Processing), ο Computer Aided Design (CAD), η Οικονομία, οι Τηλεπικοινωνίες, η Τεχνολογία Λογισμικού (Software Engineering), ο Χρονοπρογραμματισμός (Scheduling), τα Γραφικά Υπολογιστών (Computer Graphics) και πολλοί άλλοι.

2.10 ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι Γ.Α. αποτελούν μια πρωτότυπη μεταφορά ενός μοντέλου που λειτουργεί με επιτυχία για εκατομμύρια χρόνια στη φύση. Από αυτή τη σκοπιά, δημιουργείται τουλάχιστον έκπληξη, αν όχι θαυμασμός, στον απλό χρήστη για την πρωτοτυπία της ιδέας. Επιβεβαιώνεται, και σε αυτή την περίπτωση, η τάση της Επιστήμης να εμπνέεται από την ανθρώπινη ζωή. Από λειτουργική άποψη, οι Γ.Α. αποτελούν ένα ισχυρό και εύρωστο εργαλείο βελτιστοποίησης. Είναι σε θέση να αντιμετωπίζουν μεγάλη ποικιλία προβλημάτων μεγάλης δυσκολίας και να προσαρμόζονται σε πολλά περιβάλλοντα υλοποίησης. Παρ' όλα αυτά, για προβλήματα όχι μεγάλης πολυπλοκότητας και, όπου υπάρχουν εξειδικευμένες μέθοδοι βελτιστοποίησης, ίσως οι Γ.Α. να μην είναι η καλύτερη επιλογή, γιατί είναι εργαλείο γενικού σκοπού. Πολύ δημοφιλής και αποδοτικές είναι οι εφαρμογές που συνδυάζουν Γ.Α. με άλλες μεθόδους (υβριδικοί Γ.Α.), γιατί έτσι εξουδετερώνονται αμοιβαία τα μειονεκτήματά τους.

Σήμερα ο αριθμός των εφαρμογών που χρησιμοποιούν Γ.Α. αυξάνει με γρήγορους ρυθμούς, πράγμα που δείχνει ότι το μέλλον των Γ.Α. είναι ευοίωνο. Ο επιστημονικός και επιχειρηματικός κόσμος έχει πειστεί ότι η δύναμη που προσφέρουν οι Γ.Α. είναι ικανή να δίνει γρήγορες και αξιόπιστες λύσεις σε προβλήματα πολύ μεγάλης δυσκολίας και πολυπλοκότητας, γι' αυτό και αναμένονται σοβαρές επενδύσεις σε κόπο και χρήματα για την παραπέρα εξέλιξή τους.

Μέσα στο ευρύτερο πλαίσιο της Τεχνητής Νοημοσύνης, οι Γ.Α. αποτελούν ένα πολλά υποσχόμενο πεδίο. Υπάρχουν προβλέψεις που υποστηρίζουν ότι το τοπίο της Τεχνητής Νοημοσύνης ίσως αλλάξει ριζικά με την χρήση Γ.Α. Τα Νευρωνικά Δίκτυα επίσης αναμένεται να γνωρίσουν μεγάλη ανάπτυξη συνδυαζόμενα με Γ.Α., ενώ γίνεται λόγος ακόμη και για μελλοντική τους ενοποίηση.

Ένα καυτό σημείο έρευνας αποτελεί σήμερα η υλοποίηση Γ.Α. σε παράλληλες μηχανές. Σε προηγούμενο κεφάλαιο επισημάνθηκε ότι οι Γ.Α. έχουν έντονα στοιχεία παραλληλισμού, ενώ κάποιοι ερευνητές τους χαρακτηρίζουν υψηλού επιπέδου παράλληλους αλγόριθμους

(highly parallel algorithms). Από την άλλη μεριά, οι παράλληλοι υπολογιστές έχουν αρχίσει να κάνουν δειλά βήματα στην αγορά με συστήματα που περιέχουν από λίγες δεκάδες μέχρι λίγες χιλιάδες επεξεργαστές. Όλη αυτή η επεξεργαστική ισχύς μπορεί να γίνει πολύ καλά εκμεταλλεύσιμη από τους Γ.Α. Αυτό βασίζεται κυρίως στο γεγονός ότι οι Γ.Α. λειτουργούν πάνω σε πληθυσμό ατόμων, ο οποίος πλέον μπορεί να διαμοιραστεί σε αρκετούς επεξεργαστές. Έτσι, ίσως να εμπεριέχει αρκετή αλήθεια η άποψη ότι οι Γ.Α. θα βοηθήσουν την εξάπλωση και βελτίωση των επιδόσεων των παράλληλων μηχανών και το αντίστροφο.

Το μέλλον θα δείξει, αλλά το σίγουρο είναι ότι οι Γ.Α. δεν πρόκειται να εξαφανιστούν, τουλάχιστον για όσο θα μπορούν να δίνουν καλύτερες λύσεις από τους ανταγωνιστές τους. Κι αυτό μάλλον θα συμβαίνει για αρκετό καιρό, αφού είναι ανοικτά για έρευνα πολλά θέματα που αναμένεται να δώσουν ακόμη πιο εντυπωσιακά αποτελέσματα και να καθιερώσουν τους Γ.Α. ως καθοριστικό εργαλείο για την περαιτέρω εξέλιξη της Επιστήμης των Υπολογιστών.

ΚΕΦΑΛΑΙΟ 3^ο

Η ΑΝΤΙΚΕΙΜΕΝΙΚΗ ΣΥΝΑΡΤΗΣΗ

Το κεφάλαιο αναφέρεται στη βέλτιστη επιλογή εξοπλισμού κατά τη δημιουργία του εικονικού συστήματος. Παρουσιάζεται η κωδικοποίηση του προβλήματος στα πλαίσια της χρήσης της μεθοδολογίας των γενετικών αλγορίθμων (παράγραφος 3.1) και τα κριτήρια πάνω στα οποία θα βασιστεί η βελτιστοποίηση (παράγραφος 3.2).

Η βελτιστοποίηση αφορά κυρίως στην ελαχιστοποίηση του χρόνου ολοκλήρωσης του συστήματος, επομένως προσεγγίζεται το πρόβλημα θεωρητικού υπολογισμού του χρόνου αυτού (παράγραφος 3.3).

Περιγράφονται, η δομή και λειτουργία της αντικειμενικής συνάρτησης που χρησιμοποιείται από το γενετικό αλγόριθμο κατά τη βελτιστοποίηση (παράγραφος 3.4).

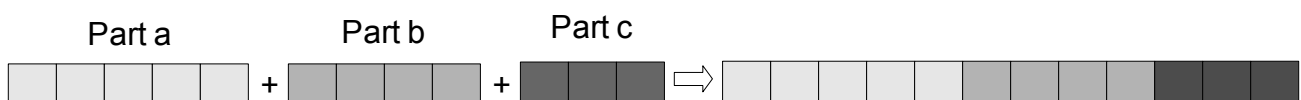
Στο τέλος του κεφαλαίου παρουσιάζεται η υπολογιστική υλοποίηση όλων των παραπάνω σε κώδικα στη Matlab (παράγραφοι 3.5 και 3.6).

3.1 ΚΩΔΙΚΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

Πρόκειται για τον πίνακα, οι διαδοχικές τιμές του οποίου συμβολίζουν την αλληλουχία των μηχανών από τις οποίες θα περάσουν τα τεμάχια. Επιλέξαμε λόγω του μικρού διαστήματος μεταβολής της λύσης να εργαστούμε με πραγματικές τιμές. Εξάλλου στα προβλήματα χρονοπρογραμματισμού, όπως στην περίπτωση μας, είναι συνηθισμένη η κωδικοποίηση αυτή.

Συγκεκριμένα, τα φασεολόγια των μερίδων συμβολίζονται με διαδοχικά γονίδια. Κάθε γονίδιο αντιστοιχεί σε μία κατεργασία και περιέχει τον αύξοντα αριθμό της μηχανής στην οποία αυτή θα εκτελεστεί. Το πεδίο ορισμού από το οποίο κάθε κατεργασία (γονίδιο) παίρνει τιμές είναι το σύνολο των μηχανών (αριθμών) που μπορούν να την εκτελέσουν.

Το συνολικό χρωμόσωμα αποτελείται από τα γονίδια όλων των μερίδων μαζί.



Στο τέλος του χρωμοσώματος τοποθετείται ένα extra γονίδιο με τον αύξοντα αριθμό της εβδομάδας μέσα στην οποία θα αρχίσει η παραγωγή. Ο φόρτος εργασίας των μηχανών υποθέτουμε πως αλλάζει ανά εβδομάδα και μας ενδιαφέρει η εύρεση του βέλτιστου χρόνου έναρξης των εργασιών.



Το δεύτερο σημαντικό στοιχείο του κώδικα βελτιστοποίησης, η αντικειμενική συνάρτηση, παρουσιάζεται αναλυτικά σε επόμενο κεφάλαιο μαζί με τα κριτήρια βελτιστοποίησης που χρησιμοποιούμε.

3.2 ΚΡΙΤΗΡΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Η επιλογή εξοπλισμού από επιμέρους πραγματικά συστήματα κατεργασιών για τη δημιουργία ενός εικονικού αντιμετωπίζεται ως ένα πρόβλημα βελτιστοποίησης. Η βελτιστοποίηση ορίζεται από τις τρεις ακόλουθες απαιτήσεις:

1. Ελαχιστοποίηση του χρόνου ολοκλήρωσης των κατεργασιών.
2. Συγκέντρωση διαφορετικών μερίδων, που βρίσκονται στον ίδιο κατασκευαστή - αφετηρία και μεταφορά τους στον κοινό κατασκευαστή – προορισμό. Κάθε φορά που επιτυγχάνεται ο στόχος αυτός δίνεται bonus στο χρωμόσωμα.
3. Ελαχιστοποίηση της χρήσης μηχανών με υψηλό φόρτο εργασίας. Κάθε φορά που χρησιμοποιείται μια τέτοια μηχανή δίνεται penalty στο χρωμόσωμα.
4. Ολοκλήρωση όλων των κατεργασιών μέσα στην χρονική περίοδο που έχει οριστεί από την παραγγελία. Για κάθε μία εβδομάδα καθυστέρησης δίνεται penalty στο χρωμόσωμα.

Η βελτιστοποίηση λοιπόν, ισοδυναμεί με την ελαχιστοποίηση της αντικειμενικής συνάρτησης: “Χρόνος ολοκλήρωσης κατεργασιών” - “Bonuses” + “Penalties”

Τα bonuses και τα penalties μετατρέπονται πρώτα σε χρονικές μονάδες βάσει μιας ισοτιμίας, ώστε να προστεθούν στο “Χρόνο ολοκλήρωσης” (έχουμε εκ των προτέρων ορίσει ότι κάθε bonus θα ισοδυναμεί με κέρδος “x” ωρών. Αντίστοιχα για τις ποινές).

Τα κριτήρια 1, 3 και 4 αφορούν στην χρονική βελτιστοποίηση του συστήματος, ενώ το κριτήριο 2 αφορά στο κόστος.

3.3 ΧΡΟΝΟΣ ΟΛΟΚΛΗΡΩΣΗΣ ΚΑΤΕΡΓΑΣΙΩΝ

Σύστημα με μία μερίδα παραγωγής:

Για τον υπολογισμό της χρονικής διάρκειας ενός φασεολογίου θεωρούμε το ακόλουθο παράδειγμα μίας μερίδας που η παραγωγή της απαιτεί 3 κατεργασίες (έστω ότι η μερίδα αποτελείται από “n” τεμάχια):

Φάση 1 ^η	Κατεργασία 1 ^η	Μηχανή 1 ^η
Φάση 2 ^η	Κατεργασία 2 ^η	Μηχανή 2 ^η
Φάση 3 ^η	Κατεργασία 3 ^η	Μηχανή 3 ^η

Πριν από κάθε κατεργασία η μηχανή σετάρεται. Τα τεμάχια με τη σειρά δένονται, ακολουθεί η κατεργασία τους και τελικά απομακρύνονται από τη μηχανή. Η διαδικασία επαναλαμβάνεται μέχρι και το τελευταίο τεμάχιο. Στη συνέχεια, η μερίδα μεταφέρεται στην επόμενη μηχανή. Όπως φαίνεται, μεταξύ των κατεργασιών παρεμβάλλονται ενδιάμεσες φάσεις μη μηδενικής διάρκειας. Το φασεολόγιο στην πλήρη του μορφή γίνεται:

Φάση 1 ^η	Store -> 1	Μηχανή 1 ^η
	Setup 1	
	Load 1	
	Process 1	
	Unload 1	
Φάση 2 ^η	Trans 1 -> 2	Μηχανή 2 ^η
	Setup 2	
	Load 2	
	Process 2	
	Unload 2	
Φάση 3 ^η	Trans 2 -> 3	Μηχανή 3 ^η
	Setup 3	
	Load 3	
	Process 3	
	Unload 3	
	3 -> Store	

Store

Μεταφορά της πρώτης ύλης από το χώρο αποθήκευσης στην πρώτη μηχανή και μεταφορά των τελικών προϊόντων από την τελευταία μηχανή προς το χώρο αποθήκευσης. Ο χρόνος αυτός εξαρτάται από την εκάστοτε μηχανή.

Setup [1 2 3]

Η ρύθμιση και προετοιμασία της μηχανής για την κατεργασία (αλλαγή εργαλείου, φόρτωση και έλεγχος του προγράμματος σε περίπτωση μηχανής CNC, ρύθμιση ιδιοσυσκευών συγκράτησης κλπ). Η χρονική της διάρκεια εξαρτάται από την εκάστοτε κατεργασία - φάση και διαφέρει από μηχανή σε μηχανή. Πολλές φορές ο χρόνος του setup είναι μεγαλύτερος από τον χρόνο κατεργασίας των τεμαχίων.

Load [1 2 3]

Χρόνος για την πρόσδεση του κάθε τεμαχίου στη μηχανή. Η τιμή του για κάθε μερίδα εξαρτάται από την εκάστοτε μηχανή.

Process [1 2 3]

Ο καθαρός χρόνος που απαιτείται για την κατεργασία.

Unload [1 2 3]

Χρόνος για την απομάκρυνση των τεμαχίων από τη μηχανή. Η τιμή του για κάθε μερίδα εξαρτάται από την εκάστοτε μηχανή.

Trans [1->2->3]

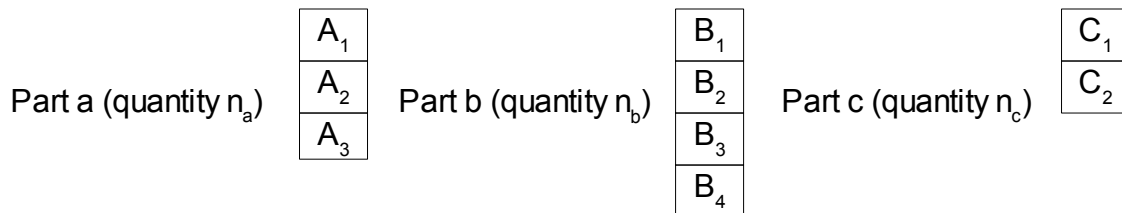
Χρόνος για τη μεταφορά της μερίδας από μηχανή σε μηχανή. Η τιμή του εξαρτάται από τους κατασκευαστές στους οποίους ανήκουν οι μηχανές, και συγκεκριμένα την μεταξύ τους απόσταση.

Ο συνολικός χρόνος για την ολοκλήρωση του φασεολογίου προκύπτει ως το άθροισμα των επιμέρους χρόνων:

$$\text{Συνολικός Χρόνος} = \sum \text{Store} + \sum \text{Trans} + \sum \text{Setup} + n * (\sum \text{Load} + \sum \text{Process} + \sum \text{Unload})$$

Σύστημα με περισσότερες μερίδες παραγωγής:

Έστω λοιπόν τώρα ότι έχουμε ένα σύστημα 3 μερίδων παραγωγής καθεμία από τις οποίες αποτελείται από n_i τεμάχια. Παρακάτω φαίνονται οι φάσεις κατεργασίας για τα προϊόντα a, a, c:



Για την κάθε μερίδα ο συνολικός χρόνος ολοκλήρωσης των κατεργασιών είναι:

Συνολικός Χρόνος = Καθαρός Χρόνος + Καθυστερήσεις

Υπολογίζεται σαν οι μηχανές να είναι πάντα διαθέσιμες. Σύμφωνα λοιπόν με την προηγούμενη παράγραφο ο χρόνος είναι:

$$\text{Συνολικός Χρόνος} = \sum \text{Store} + \sum \text{Trans} + \sum \text{Setup} + n * (\sum \text{Load} + \sum \text{Process} + \sum \text{Unload})$$

Οι μηχανές δεν είναι πάντα διαθέσιμες, καθώς στο σύστημα κυκλοφορούν πολλές μερίδες. Επομένως είναι πιθανό κάποια μερίδα να καθυστερήσει να μπει για κατεργασία σε μία μηχανή έως ότου αυτή να ελευθερωθεί. Επίσης είναι δυνατό δύο μερίδες να φτάσουν ταυτόχρονα σε μία μηχανή και να προηγηθεί αυτή στην οποία έχουμε δώσει προτεραιότητα.

Υπάρχουν όμως και εξωγενείς παράγοντες που πιθανώς να καθυστερήσουν μία μερίδα. Ένας απ αυτούς είναι ο φόρτος εργασίας των μηχανών εξαιτίας προϊόντων που δεν ανήκουν στο σύστημά μας. Φυσικά, δεν μπορούμε να γνωρίζουμε το πλήρες πρόγραμμα εργασίας μιας μηχανής αφού αυτές δεν μας ανήκουν. Μπορούμε όμως να γνωρίζουμε το φόρτο εργασίας της. Έτσι πρέπει να αποφεύγουμε να στέλνουμε τα parts σε μηχανές με υψηλό φόρτο εργασίας ώστε να αποφύγουμε μια πιθανή καθυστέρηση.

Ο υπολογισμός των καθυστερήσεων στο σύστημά είναι η μεγαλύτερη δυσκολία κατά τη δημιουργία της αντικειμενικής συνάρτησης και θα αντιμετωπιστεί κατασκευάζοντας διάγραμμα Gantt των μηχανών.

Ο συνολικός χρόνος για την ολοκλήρωση του συστήματος (εφόσον γνωρίζουμε τις καθυστερήσεις) είναι ο μέγιστος από τους χρόνους των επιμέρους μερίδων.

3.4 ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ ΑΝΤΙΚΕΙΜΕΝΙΚΗΣ ΣΥΝΑΡΤΗΣΗΣ

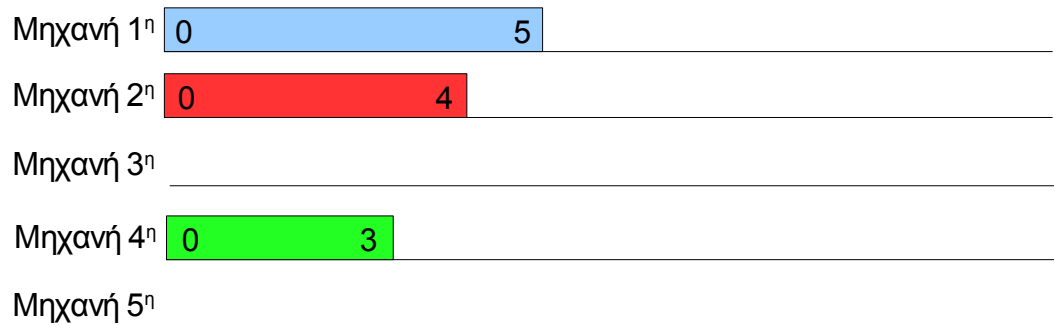
Για κάθε χρωμόσωμα η αντικειμενική συνάρτηση δημιουργεί μοναδικό διάγραμμα Gantt των μηχανών. Το διάγραμμα αυτό κατασκευάζεται ως εξής:

Κάθε μερίδα της οποίας η κατεργασία σε μια μηχανή τελειώνει, μεταφέρεται στην επόμενη βάση του φασεολογίου της. Αν η νέα μηχανή είναι άδεια τότε η κατεργασία αρχίζει άμεσα, ενώ αν είναι κατειλημμένη η μερίδα περιμένει και παίρνει σειρά να μπει αμέσως μετά. Αν δύο μερίδες φτάσουν ταυτόχρονα σε μία μηχανή, τότε προηγείται αυτή της οποίας η κατεργασία είναι βραδύτερη.

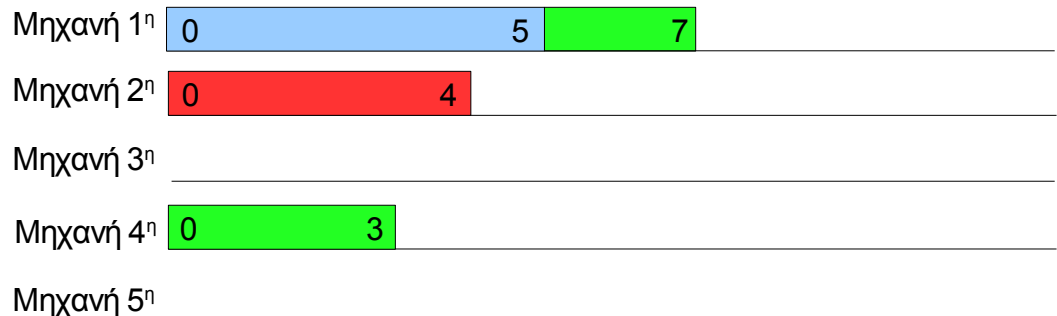
Η διαδικασία αυτή παρουσιάζεται επεκτείνοντας το παράδειγμα με τα 3 τεμάχια a, b και c:

Μηχανή (Διάρκεια)	Κατεργασίες			
	1 ^η	2 ^η	3 ^η	4 ^η
Part a	4 (3)	1 (2)	2 (2)	-
Part b	1 (5)	5 (1)	4 (1)	2 (3)
Part c	2 (4)	3 (5)	-	-

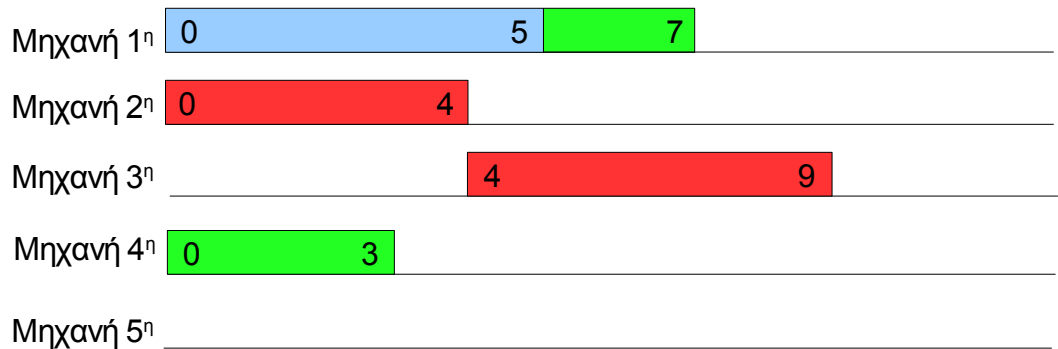
Τα τεμάχια μπαίνουν για κατεργασία στις αρχικές μηχανές:



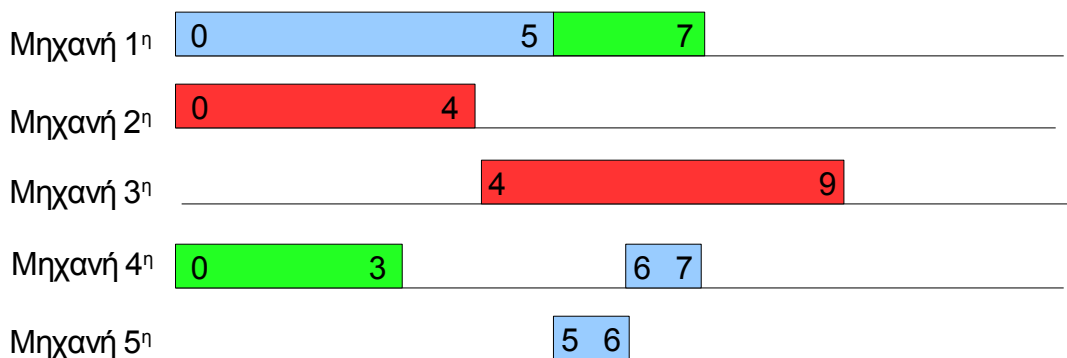
Πρώτη τελειώνει η μερίδα a και μεταφέρεται από τη μηχανή 4 στη μηχανή 1. Η 1 είναι όμως κατειλημμένη από τη μερίδα b. Επομένως η μερίδα a περιμένει να ελευθερωθεί και μπαίνει αμέσως μετά στη μηχανή:



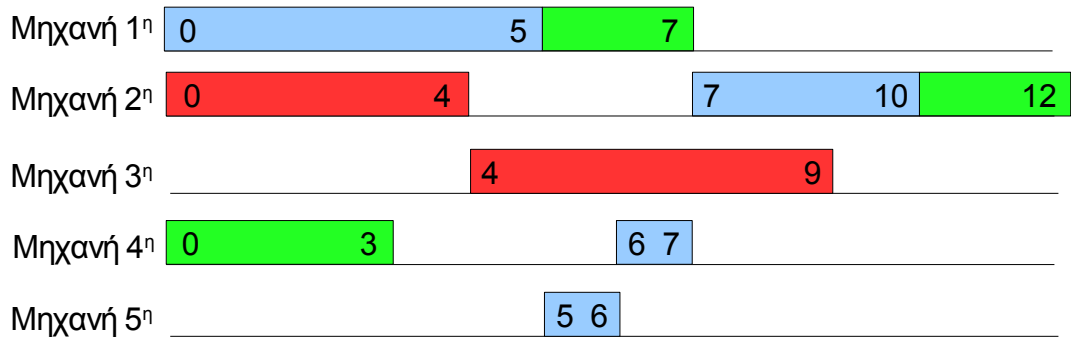
Στη συνέχεια τελειώνει η μερίδα c και μεταφέρεται στην επόμενη μηχανή κατά το φασεολόγιο της. Η μηχανή 3 είναι εκείνη τη στιγμή ελεύθερη οπότε η μερίδα μπαίνει κατευθείαν για κατεργασία:



Τελειώνει και η μερίδα b την κατεργασία της στη μηχανή 1 και μεταφέρεται διαδοχικά στις μηχανές 5 και 4 για τις επόμενες κατεργασίες της:



Οι μερίδες a και b τελειώνουν ταυτόχρονα και κατευθύνονται και οι δύο στη μηχανή 2. Θα προηγηθεί η μερίδα b στην οποία δίνουμε προτεραιότητα και θα ακολουθήσει η μερίδα a:



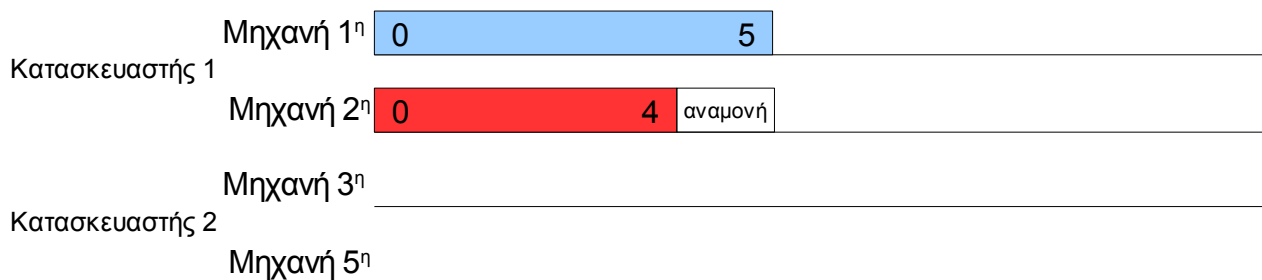
Ο τελικός χρόνος για την ολοκλήρωση των κατεργασιών του συστήματος καθορίζεται από τη μερίδα που τελειώνει τελευταία (μερίδα a και συνολικός χρόνος 12). Στο παράδειγμα αυτό θεωρήσαμε για ευκολία μηδενικούς τους υπόλοιπους χρόνους (SetupTime, LoadTime, UnloadTime, TransTime).

Πριν από κάθε μετακίνηση μερίδας πρέπει να παρεμβάλλεται ο εξής έλεγχος:

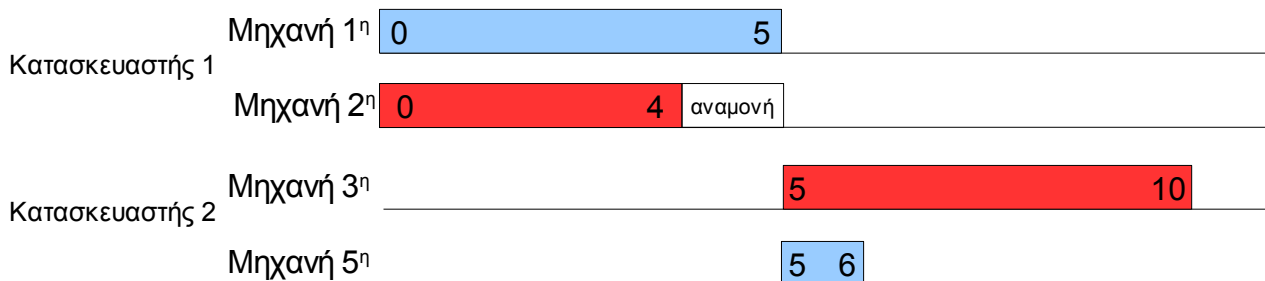
Για κάθε μερίδα που είναι έτοιμη να εγκαταλείψει ένα κατασκευαστή αναζητούνται άλλες μερίδες που βρίσκονται την ίδια στιγμή στον ίδιο κατασκευαστή και με τις οποίες έχει κοινό προορισμό - κατασκευαστή. Αν βρεθούν τέτοιες, η μερίδα περιμένει ώστε να μεταφερθούν όλες μαζί στο νέο κατασκευαστή. Αν δεν βρεθούν, τότε η μερίδα μεταφέρεται αμέσως στην επόμενη μηχανή.

Έστω λοιπόν ότι οι μερίδες b, c βρίσκονται σε μηχανές που ανήκουν στον ίδιο κατασκευαστή. Μόλις οι κατεργασίες τους τελειώσουν θα μεταφερθούν μαζί στον κοινό προορισμό – κατασκευαστή για τη συνέχιση των κατεργασιών τους. Το διάγραμμα Gantt μεταβάλλεται ως εξής στο σημείο αυτό:

Η μερίδα c τελειώνει πρώτη, δεν μεταφέρεται όμως άμεσα στη νέα μηχανή, αφού περιμένει και τη μερίδα b να τελειώσει:



Στη συνέχεια μεταφέρονται μαζί στο νέο κατασκευαστή:



3.5 ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ

Επιλέγουμε το part του οποίου η κατεργασία (έστω κ) τέλειωσε
ώστε να μεταβεί στην επόμενη (κ+1)



Έλεγχος για ταυτόχρονη μετακίνηση με άλλη μερίδα
(έστω ότι αυτή βρίσκεται στην κατεργασία λ)

ναι όχι

αν $fin_k < fin_\lambda \Rightarrow fin_k = fin_\lambda$
(η κατεργασία κ τελειώνει "εικονικά" ταυτόχρονα με τη
κατεργασία λ)
και κατάσταση αναμονής του part από 0- \rightarrow 1
επανάληψη (πίσω για την επιλογή επόμενου part)

αν κατάσταση αναμονής του part== 1 τότε
bonus=bonus+1
κατάσταση αναμονής 1- \rightarrow 0 (μηδενίζεται)
κ- \rightarrow κ+1 (επόμενη κατεργασία)
 $sta_{k+1} = fin_k + fratetime$ (έναρξη της επόμενης
κατεργασίας)



ελέγχουμε τη διαθεσιμότητα της νέας μηχανής
(έστω ότι η κατεργασία κ+1 αντιστοιχεί στη μηχανή 3)

Μηχανή διαθέσιμη
 $sta_{k+1} \geq fin_3$

Μηχανή κατειλημμένη
 $sta_{k+1} < fin_3$

$fin_{k+1} = sta_{k+1} + fgentime$ (λήξη κατεργασίας)
 $sta_3 = sta_{k+1}$ (έναρξη εργασίας μηχανής)
 $fin_3 = fin_{k+1}$ (λήξη εργασίας μηχανής)
επανάληψη

$sta_{k+1} = fin_3$ (η κατεργασία θα αρχίσει όταν η μηχανή
ελευθερωθεί)
 $fin_{k+1} = sta_{k+1} + fgentime$
 $sta_3 = sta_{k+1}$
 $fin_3 = fin_{k+1}$
επανάληψη

Η διαδικασία επαναλαμβάνεται για όλα τα γονίδια. Ο τελικός χρόνος του συστήματος είναι
το μέγιστο από τα τελικά fin των μερίδων.



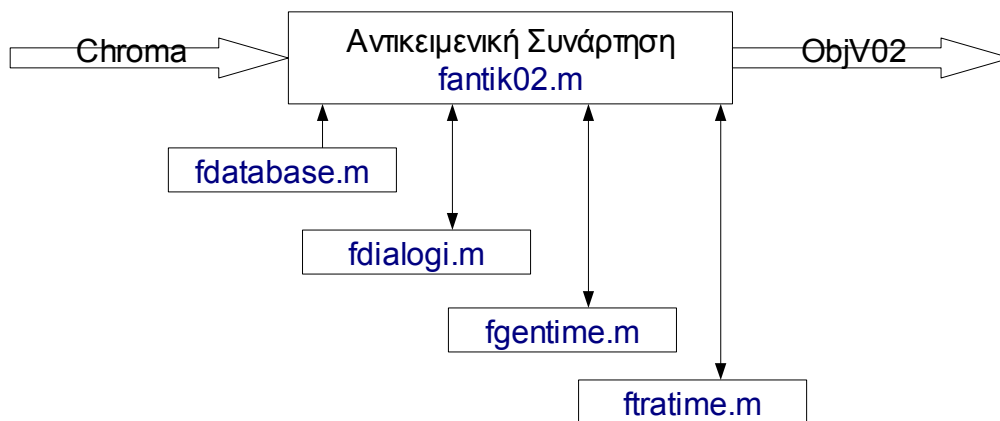
Επιπλέον bonuses για συγκεντρωτικές μεταφορές μερίδων από την αποθήκη προς τους
κατασκευαστές και αντίστροφα. Έλεγχος στους φόρτους εργασίας των μηχανών και στο
χρόνο ολοκλήρωσης του συστήματος για την επιβολή ποινών στο χρωμόσωμα.

Παρατηρήσεις:

1. Αν δύο μερίδες τελειώνουν ταυτόχρονα τις κατεργασίες τους, επιλέγεται για μετακίνηση αυτή της οποίας η επόμενη κατεργασία είναι βραδύτερη.
2. Ο έλεγχος για ταυτόχρονη μετακίνηση με άλλη μερίδα περιλαμβάνει τα εξής:
 - a) Η προς εξέταση μερίδα δεν θα πρέπει να βρίσκεται ήδη σε αναμονή.
 - b) Οι μερίδες δεν πρέπει να βρίσκονται στην τελευταία τους κατεργασία (αφού μετά δεν υπάρχει επόμενος κατασκευαστής)
 - c) Θα πρέπει να βρίσκονται στον ίδιο κατασκευαστή, να κατευθύνονται σε κοινό κατασκευαστή και ο κατασκευαστής αφητηρία να διαφέρει από τον κατασκευαστή προορισμό.
 - d) Η προβλεπόμενη αναμονή του τεμαχίου πρέπει να είναι εντός κάποιων χρονικών ορίων και όχι μεγαλύτερη.

3.6 ΚΩΔΙΚΑΣ Matlab

Η αντικειμενική συνάρτηση (fantik02) δέχεται ως είσοδο το χρωμόσωμα (Chroma) και δίνει στην έξοδο την αντικειμενική του τιμή (ObjV02). Στους υπολογισμούς χρησιμοποιεί πληροφορίες από τη βάση δεδομένων. Κάποιοι από τους υπολογισμούς γίνονται σε ειδικές υπορουτίνες (fdialogi, fgentime, ftritime). Ο συγκεκριμένος κώδικας, στη μορφή που είναι, μπορεί να χρησιμοποιηθεί από το Genetic Algorithm Toolbox της Matlab για τη βελτιστοποίηση του προβλήματός. Σχηματικά:



Παρουσίαση της συνάρτησης fantik02

Αρχικά φορτώνουμε τη βάση δεδομένων και δημιουργούμε τις μεταβλητές και τους πίνακες που θα χρησιμοποιηθούν (ο κώδικας υλοποιείται κυρίως μέσω της ανανέωσης των στοιχείων των πινάκων LivePar και LiveMac). Οι χρόνοι που φορτώνονται από τη βάση δεδομένων πρέπει να είναι σε λεπτά.

```
function ObjV02=fantik02(Chroma)

rChroma=round(Chroma);
%...load database
fdatabase

%-----

%...Format
Chrom=zeros(1,kat);
for xa=1:kat, Chrom(1,xa)=rChroma(1,xa); end
```



```

Week=rChroma(1,kat+1);
LivePar=zeros(par,4);
for xb=1:par, LivePar(xb,1)=orio(xb,1)-1; end
LiveMac=zeros(mac,2);
Gantt=zeros(kat,5);
fincheck=0;
bonus1=0;
bonus2=0;
penalty3=0;
penalty4=0;

```

Αρχίζει η αλγοριθμική διαδικασία με την επιλογή της πρώτης μερίδας.

```

%...Gantt
for xc=1:1000

p=fdialogi(LivePar,Chrom(1,:));
pass=0;

```

Γίνεται έλεγχος για ομαδοποίηση μερίδων κατά τη μεταφορά. Ο έλεγχος ξεκινά εφόσον η μερίδα δεν βρίσκεται ήδη σε αναμονή. Εξετάζονται λοιπόν τα ενεργά γονίδια όλων των άλλων μερίδων σε σχέση με αυτό της επιλαχούσας μερίδας. Πρέπει οι μερίδες να βρίσκονται στον ίδιο κατασκευαστή, να πηγαίνουν σε κοινό προορισμό και η αφετηρία να διαφέρει από τον προορισμό. Η μερίδα θα περιμένει μόνο στην περίπτωση που η προβλεπόμενη αναμονή είναι μικρότερη του τετραπλάσιου της διάρκειας της επόμενης κατεργασίας της (η επιλογή αυτή είναι θέμα στρατηγικής και μπορεί να μεταβληθεί ανά πάσα στιγμή). Σε περίπτωση που οι έλεγχοι προκύψουν θετικοί η μερίδα καθυστερεί να ξεκινήσει για τον επόμενο κατασκευαστή.

```

if LivePar(p,4)==0
    for r=1:par
        if r~=p && LivePar(p,1)>=orio(p,1) && LivePar(p,1)~=orio(p,2) &&
            LivePar(r,1)~=orio(r,2)
            if TransTime(Chrom(1,LivePar(p,1)),Chrom(1,LivePar(p,1)+1))~=0 &&
                TransTime(Chrom(1,LivePar(p,1)),Chrom(1,LivePar(r,1)))==0 &&
                TransTime(Chrom(1,LivePar(p,1)+1),Chrom(1,LivePar(r,1)+1))==0
                if LivePar(p,3)<LivePar(r,3)
                    Anam=LivePar(r,3)-LivePar(p,3);

```

```

        if Anam<fgentime (Chrom(1, :), p, LivePar(p, 1)+1) *4;
            Gantt (LivePar(p, 1), 5)=Gantt (LivePar(p, 1), 5)+Anam;
            LivePar(p, 3)=LivePar(r, 3);
            LivePar(p, 4)=1;
            pass=1;
        end
    end
end
end
end
end
end
end

```

Η επιλαχούσα μερίδα μεταβαίνει στην επόμενη φάση – κατεργασία αφού πρώτα δοθεί bonus σε περίπτωση που αυτή βρισκόταν σε αναμονή. Ανανεώνονται οι πίνακες LivePar, LiveMac. Συμπληρώνεται στο τέλος και ο πίνακας Gantt.

```

if pass==0
    if LivePar(p, 4)==1
        bonus1=bonus1+1;
        LivePar(p, 4)=0;
    end
    LivePar(p, 1)=LivePar(p, 1)+1;
    LivePar(p, 2)=LivePar(p, 3)+ftratime(Chrom(1, :), orio(p, 1), LivePar(p, 1));
    if LivePar(p, 2)>=LiveMac(Chrom(1, LivePar(p, 1)), 2)
        Kath=0;
        LivePar(p, 3)=LivePar(p, 2)+fgentime(Chrom(1, :), p, LivePar(p, 1));
        LiveMac(Chrom(1, LivePar(p, 1)), 1)=LivePar(p, 2);
        LiveMac(Chrom(1, LivePar(p, 1)), 2)=LivePar(p, 3);
    else
        Kath=LiveMac(Chrom(1, LivePar(p, 1)), 2)-LivePar(p, 2);
        LivePar(p, 2)=LiveMac(Chrom(1, LivePar(p, 1)), 2);
        LivePar(p, 3)=LivePar(p, 2)+fgentime(Chrom(1, :), p, LivePar(p, 1));
        LiveMac(Chrom(1, LivePar(p, 1)), 1)=LivePar(p, 2);
        LiveMac(Chrom(1, LivePar(p, 1)), 2)=LivePar(p, 3);
    end
    Gantt(LivePar(p, 1), 1)=Chrom(1, LivePar(p, 1));
    Gantt(LivePar(p, 1), 2)=Kath;
    Gantt(LivePar(p, 1), 3)=LivePar(p, 2);
    Gantt(LivePar(p, 1), 4)=LivePar(p, 3);
end
end

```

Ελέγχεται αν κάποια μερίδα έχει φτάσει στην τελευταία κατεργασία της ώστε να προστεθεί στο συνολικό χρόνο και η μεταφοράς της πίσω στην αποθήκη. Αν όλες οι μερίδες έχουν ολοκληρώσει τις κατεργασίες τους τότε η επαναληπτική διαδικασία τερματίζεται.

```
if LivePar(p,1)==orio(p,2)
    LivePar(p,3)=LivePar(p,3)+StoreTime(Chrom(1,orio(p,2)),1);
    fincheck=fincheck+1;
end

if fincheck==par, break, end

end
```

Τελικά προστίθενται τα επιπλέον bonuses (για ομαδοποιημένη μεταφορά μερίδων από την αποθήκη πρώτων υλών στους αρχικούς κατασκευαστές) και τα penalties. Το penalty (1^{ου} είδους) καθορίζεται ανάλογα με το φόρτο εργασίας της μηχανή και υπάρχουν τρία επίπεδα ποινής:

penalty=1 για φόρτο εργασίας μεγαλύτερο από 90%

penalty=0.5 για φόρτο εργασίας από 70%-90%

penalty=0.1 για φόρτο εργασίας από 50%-70%

(οι επιλογές αυτές μπορούν να μεταβάλλονται ανάλογα με τη στρατηγική που ακολουθούμε)

Θεωρούμε πως η εβδομάδα έχει 6 εργάσιμες μέρες, με 8 ώρες εργασίας ανά ημέρα (σύνολο 2880 λεπτά). Αν προσθέσουμε στην εβδομάδα έναρξης όλων των κατεργασιών (Week) τον αριθμό “Χρονική στιγμή έναρξης κατεργασίας/2880” στρογγυλοποιημένο στον πλησιέστερο ακέραιο προς τα κάτω, παίρνουμε την εβδομάδα έναρξης της εκάστοτε κατεργασίας. Αν αυτή ξεπερνάει την εβδομάδα λήξης των κατεργασιών που ορίστηκε κατά την παραγγελία τότε δίνεται penalty (2^{ου} είδους) στο χρωμόσωμα.

```

%...Bonuses for Store->First machine
for p=1:(par-1)
    for r=(p+1):par
        if TransTime(Chrom(1,orio(p,1)),Chrom(1,orio(r,1)))==0, bonus2=bonus2+1; break, end
    end
end

%...Bonuses for Last machine->Store
for p=1:(par-1)
    for r=(p+1):par
        if TransTime(Chrom(1,orio(p,2)),Chrom(1,orio(r,2)))==0, bonus2=bonus2+1; break, end
    end
end

%...Penalties
EndingWeek=Week+floor(ObjV01/2880);
if EndingWeek>duration, penalty3=EndingWeek-duration; end
for xd=1:kat
    StartingWeek=Week+floor(Gantt(xd,3)/2880);
    if StartingWeek<=duration
        if WrkLoa(StartingWeek,Chrom(1,xd))>=0.9, penalty4=penalty4+1; end
        if WrkLoa(StartingWeek,Chrom(1,xd))>=0.7 &&
            WrkLoa(Week,Chrom(1,xd))<0.9, penalty4=penalty4+0.5; end
        if WrkLoa(StartingWeek,Chrom(1,xd))>=0.5 &&
            WrkLoa(Week,Chrom(1,xd))<0.7, penalty4=penalty4+0.1; end
    end
end
end

```

Αμέσως μετά υπολογίζεται η τελική τιμή της αντικειμενικής συνάρτησης.

```

%...Max system time+bonuses+penalties
ObjV02=ObjV01+bonus1*isotimia(1)+bonus2*isotimia(2)+penalty3*isotimia(3)+penalty
4*isotimia(4);

```

επεξήγηση της μεταβλητής Week

Πρόκειται για την εβδομάδα μέσα στην οποία θα ξεκινήσουν οι κατεργασίες. Είναι το τελευταίο γονίδιο του χρωμοσώματος.

επεξήγηση του πίνακα LivePar

Κάθε σειρά του πίνακα αντιστοιχεί σε κάποια μερίδα. Η πρώτη στήλη δείχνει το ενεργό γονίδιο, η δεύτερη την έναρξη της κατεργασίας και η τρίτη τη λήξη της. Η τελευταία στήλη παίρνει την τιμή 1, όταν η μερίδα βρίσκεται σε αναμονή και τη τιμή 0 όταν είναι ελεύθερη να κινηθεί.

πχ για ένα σύστημα με 3 μερίδες δημιουργείται ο πίνακας:

Part 1	κ	sta _κ	fin _κ	-
Part 2	λ	sta _λ	fin _λ	-
Part 3	μ	sta _μ	fin _μ	-

επεξήγηση του πίνακα LiveMac

Κάθε σειρά του πίνακα αντιστοιχεί σε κάποια μηχανή. Η πρώτη στήλη δείχνει την έναρξη της εργασίας και η δεύτερη τη λήξη της.

πχ για ένα σύστημα με 3 μηχανές δημιουργείται ο πίνακας:

Machine 1	sta ₁	fin ₁
Machine 2	sta ₂	fin ₂
Machine 3	sta ₃	fin ₃

επεξήγηση του πίνακα Gantt

Κάθε σειρά του πίνακα αντιστοιχεί σε κάποια φάση – κατεργασία. Η πρώτη στήλη περιέχει τον αύξοντα αριθμό της μηχανής στην οποία γίνεται η κατεργασία, η δεύτερη στήλη την χρονική καθυστέρηση εισόδου της μερίδας στη μηχανή, λόγω του ότι αυτή ήταν κατειλημμένη, η τρίτη και η τέταρτη στήλη δίνουν τις χρονικές στιγμές έναρξης και λήξης της κατεργασίας και η πέμπτη στήλη δίνει το χρόνο αναμονής κάποιας μερίδας σε μια μηχανή πριν μεταφερθεί στην επόμενη (βλέπε ομαδοποίηση τεμαχίων).

Παρουσίαση της υπορουτίνας fdialogi

Υπορουτίνα για την επιλογή της μερίδας, της οποίας η κατεργασία τελειώνει πριν από τις υπόλοιπες, ώστε να μετακινηθεί στην επόμενη μηχανή.

```
function wp=fdialogi(LivePar,Chrom)

global par
global orio

for xp=1:par
    if LivePar(xp,1)==orio(xp,2)
        LivePar(xp,3)=LivePar(xp,3)*(10^12);
    end
end

wp=1;
for xp=2:par
    if LivePar(xp,3)<LivePar(wp,3)
        wp=xp;
    elseif LivePar(xp,3)==LivePar(wp,3)
        A=fgentime(Chrom,xp,LivePar(xp,1)+1);
        B=fgentime(Chrom,wp,LivePar(wp,1)+1);
        if A>B, wp=xp; end
    end
end
```

Αν κάποια μερίδα βρίσκεται ήδη στο τελικό γονίδιο πολλαπλασιάζεται με μία μεγάλη τιμή. Αποφεύγουμε έτσι την πιθανότητα επιλογής της.

Παρουσίαση της υπορουτίνας fgentime

Υπορουτίνα για τον υπολογισμό του χρόνου κατεργασίας σε μια μηχανή.

```
function GenTime=fgentime (Chrom,cp,cg)

global Qua
global SetupTime
global LoadTime
global ProcessTime
global UnloadTime

GenTime=0;

GenTime=GenTime+SetupTime (Chrom (1,cg) ,cg) ;
GenTime=GenTime+Qua (1,cp) *LoadTime (Chrom (1,cg) ,cg) ;
GenTime=GenTime+Qua (1,cp) *ProcessTime (1,cg) ;
GenTime=GenTime+Qua (1,cp) *UnloadTime (Chrom (1,cg) ,cg) ;
```

Παρουσίαση της υπορουτίνας ftritime

Υπορουτίνα υπολογισμού του χρόνου μεταφοράς των μερίδων από μηχανή σε μηχανή.

```
function TraTime=ftritime (Chrom,sta,cg)

global TransTime
global StoreTime

if cg==sta
    TraTime=StoreTime (Chrom (1,cg) ,1) ;
else
    TraTime=TransTime (Chrom (1,cg) ,Chrom (1,cg-1) ) ;
end
```

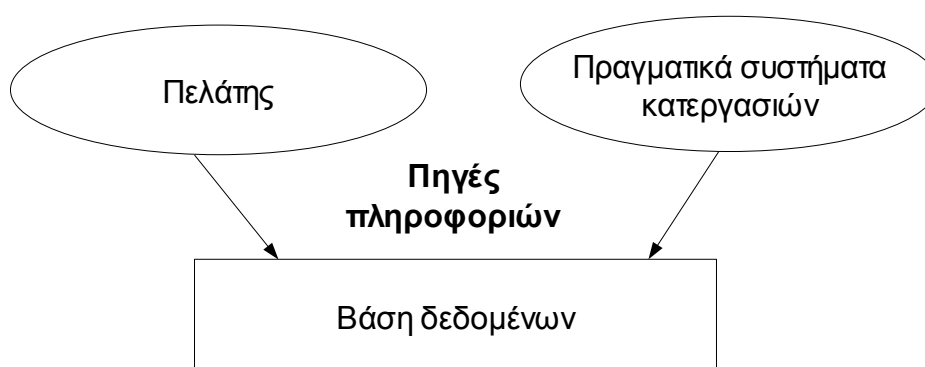
fdatabase (υπορουτίνα)

Η βάση δεδομένων παρουσιάζεται αναλυτικά στο επόμενο κεφάλαιο.

ΚΕΦΑΛΑΙΟ 4^ο

Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Το σύστημά μας αποτελείται από τον πελάτη - παραγγελία και από τα πραγματικά συστήματα κατεργασιών. Η παραγγελία αφορά πληροφορίες σχετικές με εμπορικά και γεωμετρικά μεγέθη των τεμαχίων, φασεολόγια, χρόνους και απαιτήσεις κατεργασιών. Για τα συστήματα κατεργασιών είναι απαραίτητη η αποθήκευση πληροφοριών σχετικών με τις μηχανές και τις προδιαγραφές τους.



Ποιο συγκεκριμένα η αντικειμενική συνάρτηση θα τραβάει τις ακόλουθες πληροφορίες από τη βάση δεδομένων:

1. Αριθμό μερίδων και πλήθος τεμαχίων ανά μερίδα.
2. Εναλλακτικά φασεολόγια για κάθε μερίδα, όπου φαίνονται η σειρά και το είδος των κατεργασιών που απαιτούνται (φασεολόγια στη 1^η μορφή)
3. Για κάθε επιμέρους κατεργασία πρέπει να ξέρουμε το χρόνο και το κόστος της, την αρχική και τελική γεωμετρία του τεμαχίου, τους βαθμούς ελευθερίας που απαιτούνται (άξονες) και τις τεχνικές απαιτήσεις της.
4. Χρόνους των ενδιάμεσων φάσεων (φόρτωσης τεμαχίου στη μηχανή, setup της μηχανής, εκφόρτωσης του τεμαχίου).
5. Πληροφορίες για τον αριθμό και το είδος των μηχανών που υπάρχουν στο σύστημά μας καθώς και τα είδη των κατεργασιών για τα οποία είναι ικανή η κάθε μία.
6. Οι τεχνικές προδιαγραφές της κάθε μηχανή, η οριακή γεωμετρία τεμαχίου που μπορεί να δεχτεί η κάθε μία και οι άξονες στους οποίους δουλεύει.

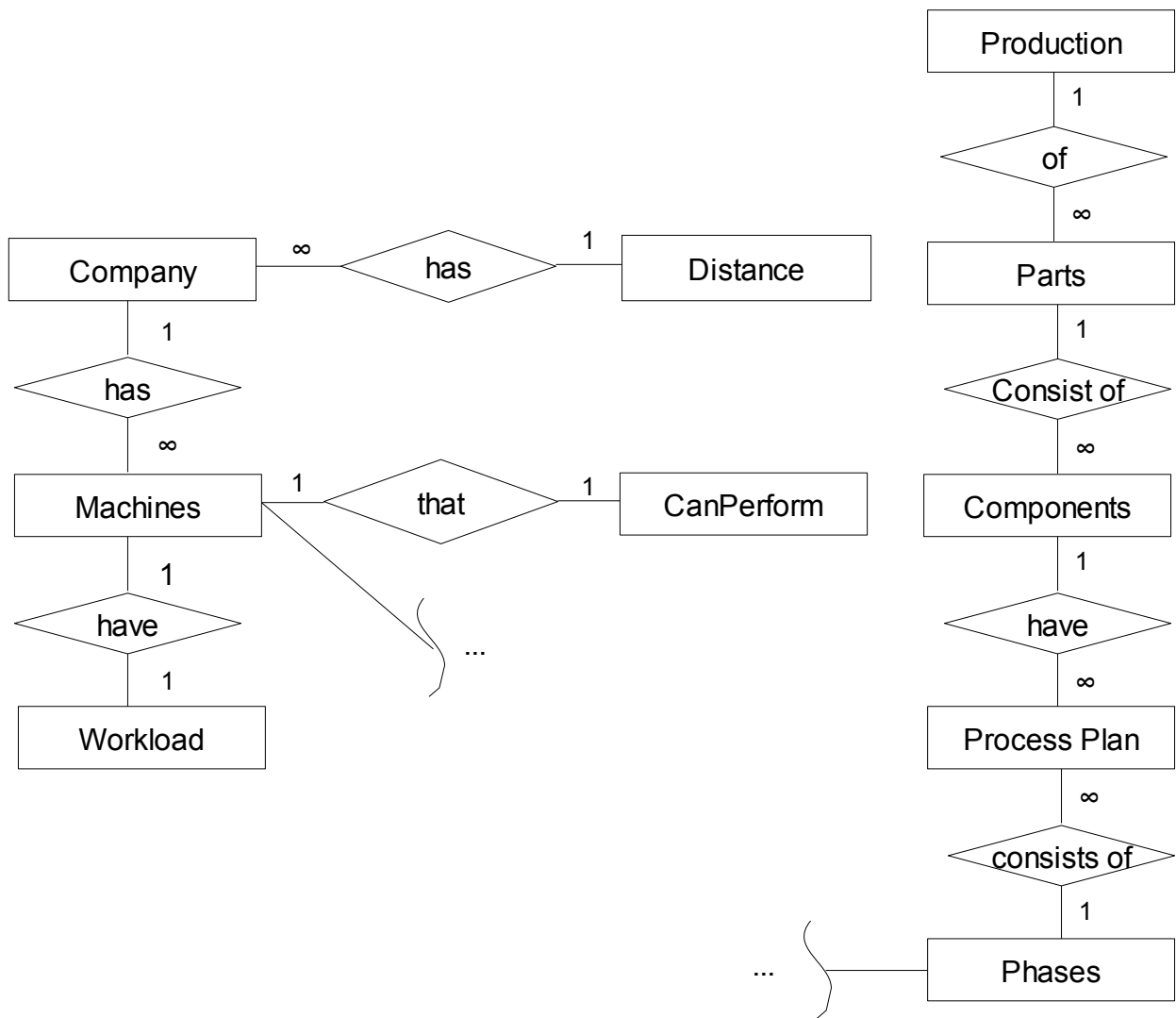
Στο κεφάλαιο αυτό παρουσιάζονται η δομή και τα χαρακτηριστικά της βάσης δεδομένων που σχεδιάστηκε στα πλαίσια της υλοποίησης ενός εικονικού συστήματος κατεργασιών (παράγραφοι 4.1 και 4.2). Παρουσιάζεται επίσης η σύνδεση της βάσης δεδομένων με την αντικειμενική συνάρτηση (παράγραφος 4.3).

4.1 ΚΕΝΤΡΙΚΟ ΣΧΕΣΙΑΚΟ ΔΙΑΓΡΑΜΜΑ – ΒΑΣΙΚΕΣ ΟΝΤΟΤΗΤΕΣ

Στο σχεσιακό διάγραμμα φαίνονται οι βασικές οντότητες της βάσης δεδομένων και οι μεταξύ τους σχέσεις.

Εμφανίζονται δύο ειδών σχέσεις:

- Σχέση ένα (1) προς πολλά (∞) που σημαίνει ότι κάθε εγγραφή του πρώτου πίνακα μπορεί να αντιστοιχεί σε περισσότερες εγγραφές του δεύτερου. Αυτό επιτυγχάνεται τοποθετώντας το **πρωτεύον κλειδί** του πρώτου σαν **δευτερεύον κλειδί** στο δεύτερο.
- Σχέση ένα (1) προς ένα (1). Κάθε εγγραφή του πρώτου πίνακα αντιστοιχεί το πολύ σε μία εγγραφή του δεύτερου. Στην πραγματικότητα είναι δυνατόν όλα τα χαρακτηριστικά να περιληφθούν σε έναν πίνακα, παρ' όλα αυτά για λόγους που θα φανούν παρακάτω επιλέξαμε να δημιουργήσουμε δύο διαφορετικούς. Η σχέση υλοποιείται τοποθετώντας το **πρωτεύον κλειδί** του ενός σαν **δευτερεύον κλειδί** στον άλλο.



Παρακάτω, παρουσιάζονται τα χαρακτηριστικά (attributes) της κάθε οντότητας (entity) **με το πρωτεύον κλειδί να είναι με κόκκινη γραφή**. Πρόκειται για τις πληροφορίες που πρέπει να αποθηκευτούν για την κάθε οντότητα και ισοδυναμούν με τις επικεφαλίδες των στηλών των αντίστοιχων πινάκων της βάσης δεδομένων.

Production

Production ID	Production description	Starting date	Max Duration (weeks)
----------------------	------------------------	---------------	----------------------

Στη κορυφή της βάσης δεδομένων βρίσκεται η οντότητα Production (παραγωγή). Εκτός από τον κωδικό της και μια σύντομη περιγραφή, αποθηκεύονται η ημερομηνία έναρξης και η μέγιστη διάρκειά της σε εβδομάδες.

Μια παραγωγή αφορά στην κατασκευή πολλών κομματιών, ενώ κάθε κομμάτι ανήκει σε κάποια προγραμματισμένη παραγωγή. Η σχέση μεταξύ της οντότητας Production με την οντότητα Part, είναι μία σχέση της μορφής ένα (1) προς πολλά (∞).

Parts

Part ID	Part description	Number of components	Production ID
---------	------------------	----------------------	---------------

Ο πίνακας (Parts) αναφέρεται στα χαρακτηριστικά των κομματιών που θα παραχθούν και συγκεκριμένα στο πλήθος των τεμαχίων από τα οποία αποτελούνται.

Components

Component ID	Component description	Material	Quantity per part	Total Quantity
--------------	-----------------------	----------	-------------------	----------------

Initial length (mm)	Initial width (mm)	Initial height (mm)	Initial Diameter (mm)	Initial weight (kg)
---------------------	--------------------	---------------------	-----------------------	---------------------

CAD	Part ID
-----	---------

Είναι τα μηχανολογικά υποσύνολα από τα οποία απαρτίζεται ένα κομμάτι. Χαρακτηρίζονται από κάποιον κωδικό (ID) και μία περιγραφή (description), ενώ σημαντικές πληροφορίες είναι το υλικό (material), το πλήθος τους ανά κομμάτι (Quantity per part), το μέγεθος της μερίδας (Total Quantity), η αρχική γεωμετρία, το βάρος του ακατέργαστου υλικού (initial geometry and weight) και τα μηχανολογικά τους σχέδια (CAD).

Οι διαστάσεις Initial Length, Initial Width και Initial Height αφορούν ένα ακατέργαστο τεμάχιο σε μορφή παραλληλεπιπέδου, ενώ οι διαστάσεις Initial Diameter και Initial Length αφορούν ακατέργαστο τεμάχιο σε μορφή κυλίνδρου.

Στο κάθε τεμάχιο αντιστοιχούν ένα ή περισσότερα εναλλακτικά φασεολόγια. Αυτά είναι που θα αξιολογήσουμε στη συνέχεια με τη μέθοδο των γενετικών αλγορίθμων ώστε να καταλήξουμε στο βέλτιστο.

Process Plan

Process Plan ID	Process Plan description	Number of phases	Phase 1	
Phase 2	Phase 3	Phase 4	Phase 5	Component ID

Κάθε φασεολόγιο χαρακτηρίζεται από τον κωδικό (ID), την περιγραφή (description) και των αριθμό των φάσεων που περιλαμβάνει (number of phases). Πρόκειται για φασεολόγιο στην αρχική μορφή όπου εμφανίζονται οι κωδικοί των απαραίτητων κατεργασιών στη σειρά που πρέπει να γίνουν.

Στις επόμενες στήλες αποθηκεύονται οι κωδικοί των φάσεων και παραπέμπουν στους αντίστοιχους πίνακες με τα ιδιαίτερα χαρακτηριστικά τους.

Phase

Phase ID	Process Description	Process Type	Geometrical Data Pro	Geometrical Data Post
Time (min)	Cost			

Κάθε φάση, αντιστοιχεί σε κατεργασία συγκεκριμένου είδους (Process Type) ενός γεωμετρικού χαρακτηριστικού πάνω στο τεμάχιο (Geometrical Data pro, post). Πολύ σημαντικές πληροφορίες είναι η διάρκεια (Time) και το κόστος της (Cost), που απαιτούνται για τη χρονική και κοστολογική βελτιστοποίηση του προβλήματος.

Company

Company ID	Company Description	Number of machines	Other Company Data	Distance
------------	------------------------	-----------------------	-----------------------	----------

Κάθε κατασκευαστής διαθέτει έναν αριθμό μηχανών (number of machines). Στις επόμενες στήλες (Other Company Data) αποθηκεύονται προαιρετικές πληροφορίες όπως η διεύθυνση και το τηλέφωνο του κατασκευαστή. Η τελευταία στήλη παραπέμπει στον πίνακα με τις χρονικές αποστάσεις του κάθε κατασκευαστή από τους υπόλοιπους. Οι

χρόνοι αυτοί απαιτούνται για τον υπολογισμό των μεταφορών των μερίδων από μηχανή σε μηχανή.

Machines

Machine ID	Machine Description	Manufacturer	Machine type
X3D model	Workload	Company ID	

Η μηχανή είναι από τις πιο σημαντικές οντότητες. Χαρακτηρίζεται από το όνομά της (description), τον κατασκευαστή της (manufacturer), τον τύπο της (type) ενώ ίσως διαθέτει και ένα τρισδιάστατο μοντέλο (X3D model).

Κάθε μηχανή αντιστοιχεί σε κάποιον φόρτο εργασίας (οι πληροφορίες σχετικά με αυτόν αποθηκεύονται σε ξεχωριστό πίνακα).

Workload

Workload ID	Week 1	Week 2	Week 3	Week 4
-------------	--------	--------	--------	--------

Στον πίνακα αυτό εμφανίζεται ο εβδομαδιαίος φόρτος εργασίας που αντιστοιχεί σε κάθε μηχανή. Η εβδομάδες, για τις οποίες συγκεντρώνουμε τα δεδομένα, πρέπει να αντιστοιχούν στις εβδομάδες κατά τις οποίες θα πραγματοποιηθεί η παραγωγή (βλ. εγγραφές *starting date* και *duration* του πίνακα Production).

CanPerform

CanPerform ID	Machine ID	Milling	Drilling	Reaming
Turning	Boring	Cylindrical grinding	Flat grinding	EDM wire cutting
EDM die sinking	Consumable mould casting	Die casting	Injection moulding	

Η ικανότητα μιας μηχανής για κατεργασία συγκεκριμένου είδους εξαρτάται από τη φύση της καθώς και από τα εργαλεία και τις ιδιοσκευές που διαθέτει. Είναι δυνατόν μία

μηχανή να είναι ικανή για περισσότερους ή λιγότερους τύπους κατεργασιών από αυτούς που είναι χαρακτηριστικοί του είδους της.

Ταυτόχρονα με τη προσέγγιση αυτή μπορούμε να εντάξουμε στη βάση δεδομένων μας και διάφορες πολυμηχανές. Μια μηχανή τύπου “Multi tasking lathe” μπορεί να εκτελέσει εκτός από τις κλασσικές κατεργασίες ενός τόρνου (turning, boring) και κατεργασίες φρεζομηχανής (milling, drilling, reaming). Αντίστοιχα συμβαίνει και για μια φρεζομηχανή τύπου “Machining center with turning tools” που μπορεί να εκτελέσει και βασικές κατεργασίες τórνευσης.

Αρκεί λοιπόν για κάθε μηχανή να συμπληρώνονται οι αντίστοιχες στήλες των ειδών κατεργασιών που δύνανται να εκτελέσουν με ένα yes.

Distance

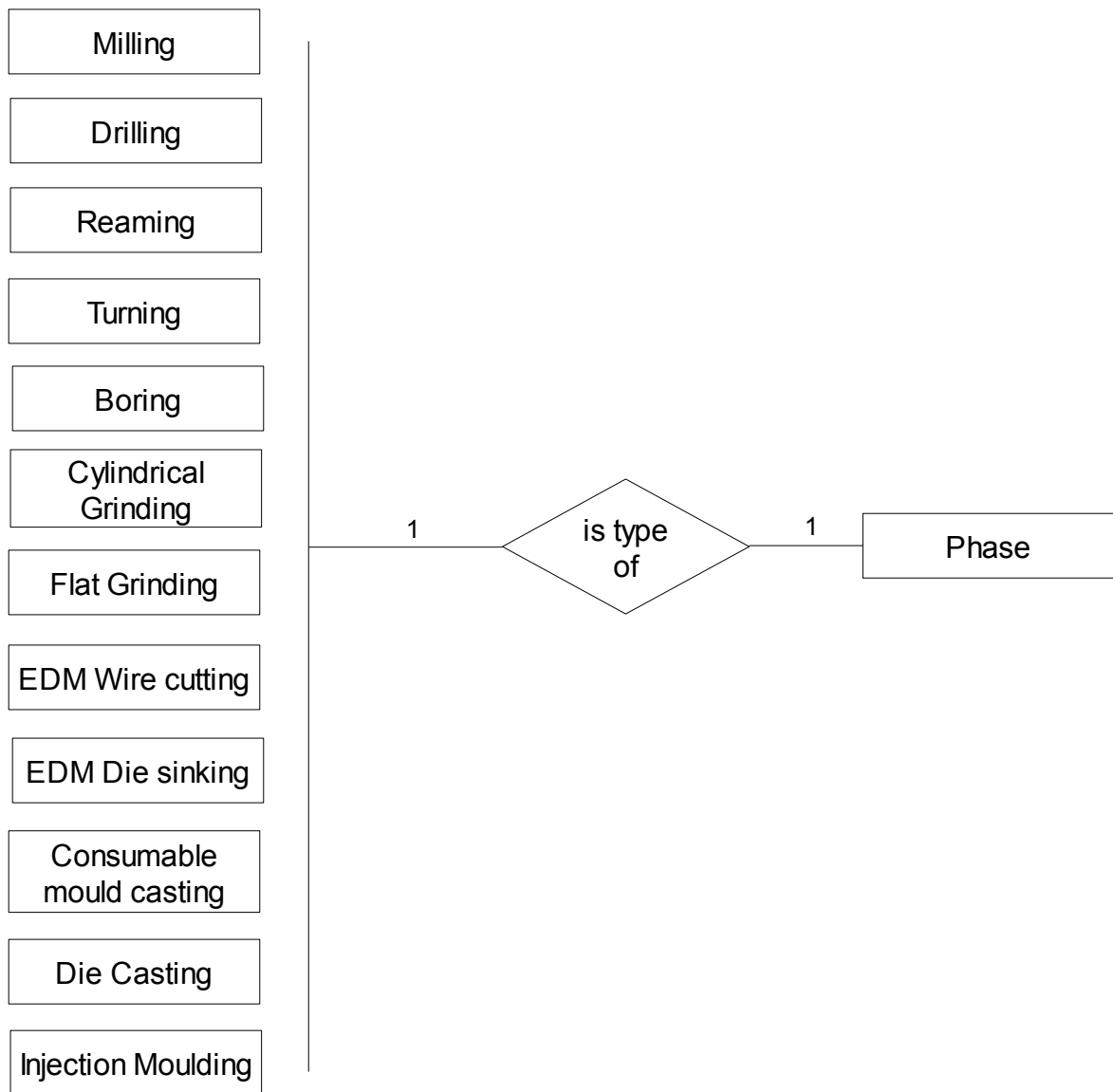
Distance ID	Distance from Company 01 (min)	Distance from Company 02 (min)	...	Distance for Store (min)
-------------	--------------------------------	--------------------------------	-----	--------------------------

Για κάθε εταιρεία συμπληρώνονται οι αποστάσεις της από τις υπόλοιπες (κάθε στήλη αντιστοιχεί σε διαφορετική εταιρεία, απόσταση 0 από τον εαυτό της). Στην τελευταία στήλη αποθηκεύεται η απόσταση από την αποθήκη, απ όπου ξεκινούν οι πρώτες ύλες και στην οποία καταλήγουν τα τελικά προϊόντα.

4.2 ΣΥΜΠΛΗΡΩΜΑΤΙΚΟ ΣΧΕΣΙΑΚΟ ΔΙΑΓΡΑΜΜΑ – ΔΕΥΤΕΡΕΥΟΥΣΕΣ ΟΝΤΟΤΗΤΕΣ

Παρακάτω φαίνεται ο διαχωρισμός της οντότητας Phase σε υποκατηγορίες – είδη κατεργασιών. Δημιουργήσαμε νέους πίνακες ώστε να αποθηκεύσουμε τα ιδιαίτερα χαρακτηριστικά κάθε κατεργασίας ανάλογα με το είδος της. Τα χαρακτηριστικά αυτά αφορούν στους άξονες που πιθανώς απαιτούνται για τη συγκεκριμένη κατεργασία (βαθμοί ελευθερίας) και στις τεχνικές της προδιαγραφής.

Από τη στιγμή μας ενδιαφέρει αποκλειστικά η σύγκριση των απαιτήσεων μιας κατεργασίας με τις προδιαγραφές της μηχανής (έλεγχος επάρκειας της μηχανής), αρκεί για τις διάφορες τιμές των χαρακτηριστικών να συμπληρώσουμε τη μέγιστη τιμή τους. Την τελική τους τιμή θα την καθορίσουμε πάνω στη μηχανή.



Milling

Milling ID	Number of axes	Number of rotary axes	X travel (mm)	Y travel (mm)
Z travel (mm)	Spindle speed (rpm)	Spindle power (kW)	Torgue (Nm)	Maximum cutting feedrate (mm/min)
Maximum rapid traverse (mm/min)	Feed motors power (kW)	Accuracy (μm)	Surface quality Ra (μm)	Tool diameter (mm)
Tool length (mm)	Phase ID			

ομοίως για Drilling και Reaming.

Turning

Turning ID	Number of axes	Number of rotary axes	Machining diameter (mm)	X travel (mm)
------------	----------------	-----------------------	-------------------------	---------------

Z travel (mm)	Spindle bore (mm)	Spindle speed (rpm)	Spindle power (mm)	Torgue (Nm)
---------------	-------------------	---------------------	--------------------	-------------

Maximum cutting feedrate (mm/min)	Maximum rapid traverse (mm/min)	Feed motors power (kW)	Accuracy (μm)	Surface quality (μm)
-----------------------------------	---------------------------------	------------------------	----------------------------	-----------------------------------

Phase ID	
----------	--

ομοίως για Boring

Cylindrical Grinding

Cylindrical Grinding ID	Number of axes	Number of rotary axes	Grinding wheel diameter (mm)	Grinding diameter (mm)
-------------------------	----------------	-----------------------	------------------------------	------------------------

X travel (mm)	Z travel (mm)	Spindle speed (rpm)	Spindle power (kW)	Maximum grinding feedrate (mm/min)
---------------	---------------	---------------------	--------------------	------------------------------------

Maximum rapid traverse (mm/min)	Accuracy (μm)	Surface quality Ra (μm)	Phase ID	
---------------------------------	----------------------------	--------------------------------------	----------	--

Flat Grinding

Flat Grinding ID	Number of axes	Number of rotary axes	Grinding wheel diameter (mm)	X travel (mm)
------------------	----------------	-----------------------	------------------------------	---------------

Y travel (mm)	Z travel (mm)	Spindle speed (rpm)	Spindle power (kW)	Maximum cutting feedrate (mm/min)
---------------	---------------	---------------------	--------------------	-----------------------------------

Maximum rapid traverse (mm/min)	Accuracy (μm)	Surface quality (μm)	Phase ID	
---------------------------------	----------------------------	-----------------------------------	----------	--

EDM Wire cutting

EDM Wire cutting ID	Wire diameter (mm)	Current (A)	Power supply (kW)	X travel (mm)
Y travel (mm)	Maximum cutting rate (mm/min)	Accuracy (μm)	Surface quality (μm)	Phase ID

EDM Die Sinking

EDM Die Sinking ID	Current (A)	Power supply (kW)	X travel (mm)	Y travel (mm)
Z travel (mm)	Maximum advance rate (mm/min)	Accuracy (μm)	Surface quality (μm)	Phase ID

Consumable Mould Casting

Consumable Mould Casting ID	Type (sand, croning, investment, plaster etc.)	Workpiece weight (kg)	Capacity (kg/hr)	Best average roughness (μm)
Accuracy (μm)	Phase ID			

Die Casting

Die Casting ID	Clamping force (tn)	Clamp stroke (mm)	Platen size horizontal (mm)	Platen size vertical (mm)
Maximum die height (mm)	Minimum die height (mm)	Tiebar space horizontal (mm)	Tiebar space vertical (mm)	Maximum shot size (kg)
Maximum ram velocity (m/min)	Ejector force (tn)	Shot stroke (mm)	Max casting pressure (bar)	Pot capacity (kg)
Maximum casting area (cm^2)	Dry cycles per minute	Phase ID		

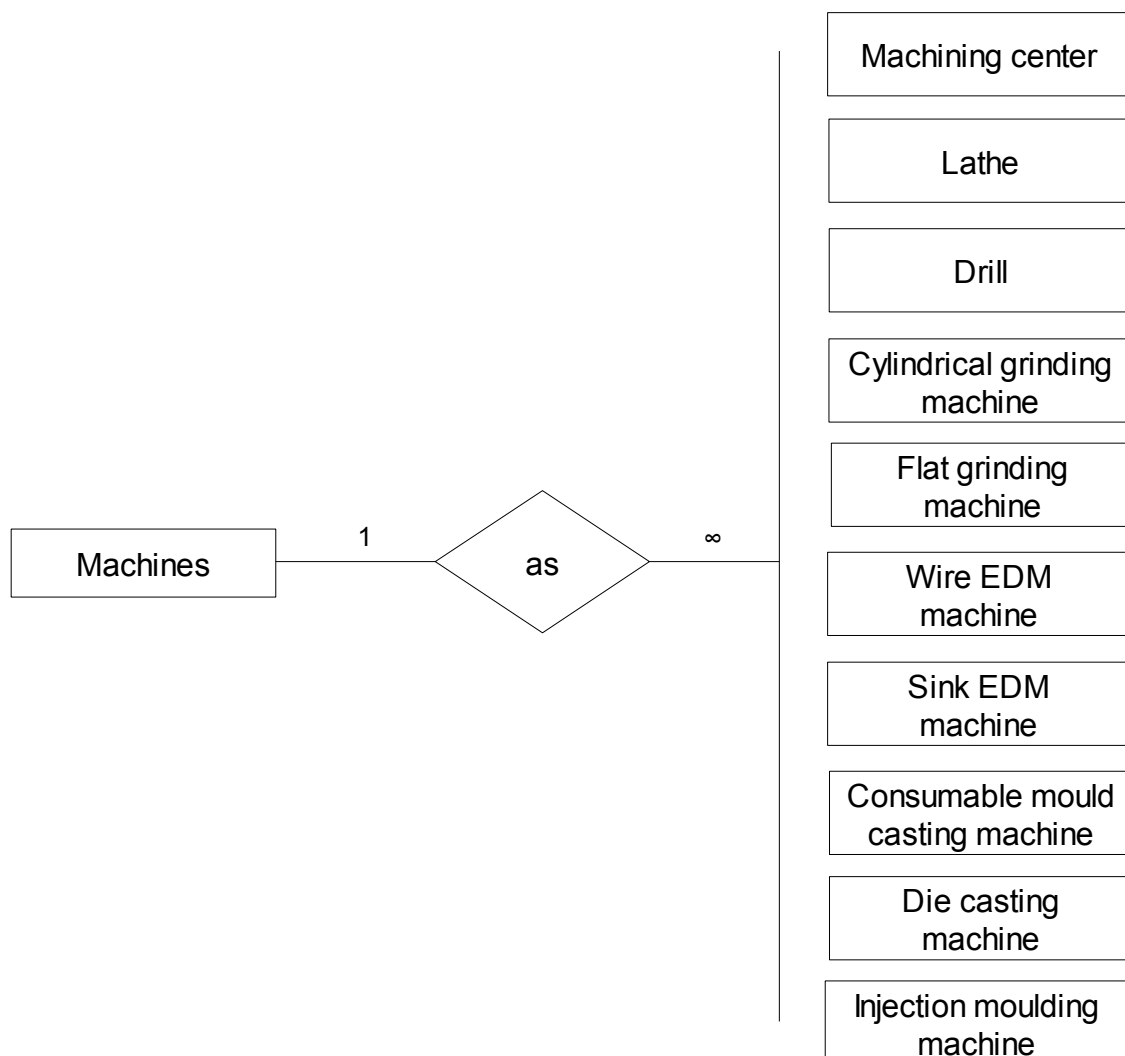
Injection Moulding

Injection Moulding ID	Clamping force (tn)	Clamp stroke (mm)	Max daylight (mm)	Platen size horizontal (mm)
-----------------------	---------------------	-------------------	-------------------	-----------------------------

Platen size vertical (mm)	Maximum die height (mm)	Minimum die height (mm)	Tiebar space horizontal (mm)	Tiebar space vertical (mm)
---------------------------	-------------------------	-------------------------	------------------------------	----------------------------

Maximum injection shot (kg)	Maximum injection pressure (bar)	Maximum injection rate (cm ³ /min)	Phase ID	
-----------------------------	----------------------------------	-----------------------------------------------	----------	--

Όσον αφορά στις μηχανές, αυτές διακρίνονται σε κάποιες βασικές κατηγορίες:



Παρουσιάζονται μόνο οι βασικοί τύποι μηχανών.

Η αντιστοιχία βασικών τύπων κατεργασιών με τους τύπους μηχανών που δύνανται να τις εκτελέσουν φαίνεται στον παρακάτω πίνακα. Πρέπει να φροντίσουμε ώστε οι τεχνικές προδιαγραφές των μηχανών να συμπίπτουν με τις προδιαγραφές των αντίστοιχων κατεργασιών. Τα χαρακτηριστικά των μηχανών συμπληρώνονται, όπου απαιτείται, με τον μέγιστο όγκο και βάρος τεμαχίου που η κάθε μηχανή μπορεί να δεχτεί και με τους μέσους χρόνους setup, φόρτωσης και εκφόρτωσης τεμαχίων που τις χαρακτηρίζουν:

	Lathe	Machining center	Drill	Cylindrical grinding machine	Flat grinding machine	Wire EDM machine	Sink EDM machine	Consumable mould casting machine	Die casting machine	Injection moulding machine
Turning	x									
Boring	x									
Milling		x								
Drilling		x	x							
Reaming		x	x							
Cylindrical grinding				x						
Flat grinding					x					
EDM wire cutting						x				
EDM die sinking							x			
Consumable mould casting								x		
Die casting									x	
Injection moulding										x

Σε κάθε μηχανή, όπως αναφέρθηκε παραπάνω, αντιστοιχούν χρόνοι φόρτωσης, προετοιμασίας και εκφόρτωσης που εξαρτώνται από το τεμάχιο και το είδος της κατεργασίας που θα εκτελεσθεί. Για την ίδια μηχανή αυτοί διαφέρουν από τεμάχιο σε τεμάχιο, αλλά και μεταξύ διαφορετικών κατεργασιών του ίδιου τύπου. Πρακτικά είναι πολύ δύσκολο να υπολογίζουμε τους χρόνους αυτούς και να τους αποθηκεύουμε στη βάση δεδομένων για κάθε διαφορετικό τεμάχιο και κατεργασία της παραγγελίας. Επίσης, η κατασκευή της βάσης δεδομένων προηγείται του ελέγχου συμβατότητας μεταξύ μηχανών και κατεργασιών ώστε να μην είμαστε, στη φάση αυτή, σίγουροι για τις κατεργασίες για τις οποίες είναι ικανή η κάθε μηχανή. Προσεγγίζουμε το πρόβλημα τοποθετώντας τους χρόνους Loading time, Setup time και Unloading time στα χαρακτηριστικά της κάθε μηχανής και υποθέτουμε μια μέση τιμή γι' αυτούς για όλες τις κατεργασίες που εκτελούν και τα τεμάχια που κατεργάζονται. Παρά την απλοποίηση αυτή, ο κώδικας βελτιστοποίησης

μπορεί να δεχτεί σε κάθε μηχανή διαφορετικές τιμές των παραπάνω χρόνων ανάλογα με την κατεργασία και το τεμάχιο.

Machining Center (ομοίως για Drill)

Machining center ID	Number of axes	Number of rotary axes	Max X travel (mm)	Max Y travel (mm)
---------------------	----------------	-----------------------	-------------------	-------------------

Max Z travel (mm)	Max spindle speed (rpm)	Spindle power (kW)	Torgue (Nm)	Lowest maximum cutting feedrate (mm/min)
-------------------	-------------------------	--------------------	-------------	------------------------------------------

Lowest maximum rapid traverse (mm/min)	Lowest feed motors power (kW)	Accuracy (μm)	Surface quality Ra (μm)	Max tool diameter (mm)
----------------------------------------	-------------------------------	----------------------------	--------------------------------------	------------------------

Max tool length (mm)	Table length (mm)	Table width (mm)	Table diameter (mm)	Height clearance (mm)
----------------------	-------------------	------------------	---------------------	-----------------------

Max load capacity (kg)	Setup time (min)	Loading time (min)	Unloading time (min)	Machine ID
------------------------	------------------	--------------------	----------------------	------------

Lathe

Turning ID	Number of axes	Number of rotary axes	Max machining diameter (mm)	Max X travel (mm)
------------	----------------	-----------------------	-----------------------------	-------------------

Max Z travel (mm)	Spindle bore (mm)	Max spindle speed (rpm)	Spindle power (mm)	Torgue (Nm)
-------------------	-------------------	-------------------------	--------------------	-------------

Lowest maximum cutting feedrate (mm/min)	Lowest maximum rapid traverse (mm/min)	Lowest feed motors power (kW)	Accuracy (μm)	Surface quality (μm)
------------------------------------------	----------------------------------------	-------------------------------	----------------------------	-----------------------------------

Swing over bed (mm)	Swing over carriage (mm)	Max workpiece length (mm)	Max load capacity (kg)	Setup time (min)
---------------------	--------------------------	---------------------------	------------------------	------------------

Loading time (min)	Unloading time (min)	Machine ID		
--------------------	----------------------	------------	--	--

Cylindrical grinding machine

Cylindrical grinding machine ID	Number of axes	Number of rotary axes	Max grinding wheel diameter (mm)	Max grinding diameter (mm)
---------------------------------	----------------	-----------------------	----------------------------------	----------------------------

Max X travel (mm)	Max Z travel (mm)	Max spindle speed (rpm)	Spindle power (kW)	Lowest maximum grinding feedrate (mm/min)
-------------------	-------------------	-------------------------	--------------------	-------------------------------------------

Lowest maximum rapid traverse (mm/min)	Accuracy (μm)	Surface quality Ra (μm)	Swing over table (mm)	Max workpiece length (mm)
----------------------------------------	----------------------------	--------------------------------------	-----------------------	---------------------------

Max load capacity (kg)	Setup time (min)	Loading time (min)	Unloading time (min)	Machine ID
------------------------	------------------	--------------------	----------------------	------------

Flat grinding machine

Flat grinding machine ID	Number of axes	Number of rotary axes	Max grinding wheel diameter (mm)	Max X travel (mm)
--------------------------	----------------	-----------------------	----------------------------------	-------------------

Max Y travel (mm)	Max Z travel (mm)	Max spindle speed (rpm)	Spindle power (kW)	Lowest maximum cutting feedrate (mm/min)
-------------------	-------------------	-------------------------	--------------------	------------------------------------------

Lowest maximum rapid traverse (mm/min)	Accuracy (μm)	Surface quality (μm)	Max workpiece length (mm)	Max workpiece width (mm)
----------------------------------------	----------------------------	-----------------------------------	---------------------------	--------------------------

Max workpiece height (mm)	Max load capacity (kg)	Setup time (min)	Loading time (min)	Unloading time (min)
---------------------------	------------------------	------------------	--------------------	----------------------

Machine ID	
------------	--

Wire EDM machine

Wire EDM machine ID	Max wire diameter (mm)	Min wire diameter (mm)	Max current (A)	Min current (A)
Power supply (kW)	Max X travel (mm)	Max Y travel (mm)	Lowest maximum cutting rate (mm/min)	Accuracy (μm)
Surface quality (μm)	Max workpiece length (mm)	Max workpiece width (mm)	Max workpiece height (mm)	Max load capacity (kg)
Setup time (min)	Loading time (min)	Unloading time (min)	Machine ID	

Sink EDM machine

Sink EDM machine ID	Min current (A)	Max current (A)	Power supply (kW)	Max X travel (mm)
Max Y travel (mm)	Max Z travel (mm)	Lowest maximum advance rate (mm/min)	Accuracy (μm)	Surface quality Ra (μm)
Max workpiece length (mm)	Max workpiece width (mm)	Max workpiece height (mm)	Max load capacity (kg)	Setup Time (min)
Loading time (min)	Unloading time (min)	Machine ID		

Consumable mould casting machine

Consumable mould casting machine ID	Type (sand, croning, investment, plaster etc.)	Minimum workpiece weight (kg)	Capacity (kg/hr)	Best average roughness (μm)
Accuracy (μm)	Setup time (min)	Loading time= 0	Unloading time (min)	Machine ID

Die casting machine

Die casting machine ID	Clamping force (tn)	Clamp stroke (mm)	Platen size horizontal (mm)	Platen size vertical (mm)
------------------------	---------------------	-------------------	-----------------------------	---------------------------

Maximum die height (mm)	Minimum die height (mm)	Tiebar space horizontal (mm)	Tiebar space vertical (mm)	Maximum shot size (kg)
-------------------------	-------------------------	------------------------------	----------------------------	------------------------

Maximum ram velocity (m/min)	Ejector force (tn)	Shot stroke (mm)	Max casting pressure (bar)	Pot capacity (kg)
------------------------------	--------------------	------------------	----------------------------	-------------------

Maximum casting area (cm ²)	Dry cycles per minute	Setup time (min)	Loading time= 0	Unloading time (min)
-----------------------------------------	-----------------------	------------------	-----------------	----------------------

Machine ID	
------------	--

Injection moulding machine

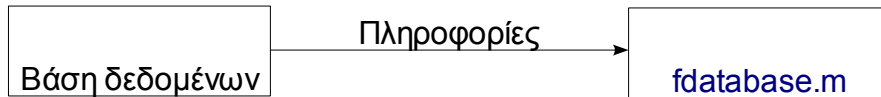
Injection moulding machine ID	Clamping force (tn)	Clamp stroke (mm)	Max daylight (mm)	Platen size horizontal (mm)
-------------------------------	---------------------	-------------------	-------------------	-----------------------------

Platen size vertical (mm)	Maximum die height (mm)	Minimum die height (mm)	Tiebar space horizontal (mm)	Tiebar space vertical (mm)
---------------------------	-------------------------	-------------------------	------------------------------	----------------------------

Maximum injection shot (kg)	Maximum injection pressure (bar)	Maximum injection rate (cm ³ /min)	Setup time (min)	Loading time= 0
-----------------------------	----------------------------------	-----------------------------------------------	------------------	-----------------

Unloading time (min)	Machine ID	
----------------------	------------	--

4.3 ΣΥΝΔΕΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΜΕ ΑΝΤΙΚΕΙΜΕΝΙΚΗ ΣΥΝΑΡΤΗΣΗ



Από τη βάση δεδομένων, όλες οι απαιτούμενες πληροφορίες για τη λειτουργία του γενετικού αλγόριθμου μεταφέρονται και αποθηκεύονται στο αρχείο `fdatabase` (m-file της `Matlab`) μέσω SQL προγραμματισμού. Το αρχείο αυτό είναι ο μοναδικός σύνδεσμος του κώδικα βελτιστοποίησης με το πραγματικό σύστημα ώστε κάθε διαφορετικό πρόβλημα να αντιστοιχεί σε διαφορετικό αρχείο. Στη συνέχεια της παραγράφου αυτής επεξηγούνται οι μεταβλητές και οι πίνακες του `fdatabase`.

Σε μία δοσμένη παραγωγή (Production) μας ενδιαφέρει η κατασκευή των τεμαχίων (Components) που περιλαμβάνει και μάλιστα στις ποσότητες που προδιαγράφεται. Σε κάθε Component αντιστοιχεί ένας αριθμός εναλλακτικών φασεολογίων. Από τη στιγμή που μας ενδιαφέρει η βελτιστοποίηση της παραγωγής για το σύνολο των τεμαχίων οφείλουμε να διαλέξουμε τον καλύτερο συνδυασμό φασεολογίων. Για παράδειγμα αν το σύστημα περιλαμβάνει 3 τεμάχια που το καθένα έχει αντίστοιχα 2,3 και 4 εναλλακτικά φασεολόγια τότε το πλήθος των διαφορετικών συνδυασμών φασεολογίων που μπορούμε να ακολουθήσουμε είναι $2 \times 3 \times 4 = 24$. Αφού τρέξουμε τον γενετικό αλγόριθμο για όλους τους συνδυασμούς θα τους συγκρίνουμε και θα επιλέξουμε τον καλύτερο.

Για ένα συγκεκριμένο συνδυασμό φασεολογίων γνωρίζουμε το πλήθος των τεμαχίων (εγγραφές *Number of components* του πίνακα *Parts*), το μέγεθος των μερίδων (εγγραφές *Total Quantity* του πίνακα *Components*) και τον αριθμό των φάσεων κατεργασίας για το κάθε τεμάχιο (εγγραφές *Number of phases* στον πίνακα *Process Plan*). Γνωρίζουμε επίσης το χρόνο μέσα στον οποίο πρέπει να έχει παραδοθεί η παραγγελία (εγγραφή *Max Duration* του πίνακα *Production*).

```
%Max project duration (weeks)
duration
%Number of parts
par
%Arhi kai telos gonidiou gia kathe part (pinakas par x 2)
orio
%Synolo katergasiwn
kat
%Megethos meridas tou kathe part (pinakas 1 x par)
Qua
```


Κάθε φάση αντιστοιχεί σε κάποιον καθαρό χρόνο κατεργασίας (εγγραφή *Time* στον πίνακα *Phase*).

```
%katharoi xronoi katergasias (minutes) gia kathe fasi (pinakas 1 x kat)
ProcessTime
```

Στη συνέχεια πρέπει να ορίσουμε το πεδίο ορισμού των γονιδίων, δηλαδή το σύνολο από το οποίο ο γενετικός αλγόριθμος θα διαλέγει μηχανές για το χρωμόσωμα. Για κάθε φάση, που είναι ορισμένου τύπου (βλέπε εγγραφή *Process type* στον πίνακα *Phases*), πηγαίνουμε στον πίνακα *CanPerform* και επιλέγουμε τις μηχανές εκείνες που είναι ικανές για τη συγκεκριμένου τύπου κατεργασία (συμπληρωμένο *yes* στην αντίστοιχη στήλη). Η διαδικασία επαναλαμβάνεται για όλες τις κατεργασίες. Δημιουργείται έτσι ο βοηθητικός πίνακας *Phases/Machines* (κάθε φάση αντιστοιχείται στις μηχανές που δύνανται να την εκτελέσουν) και προκύπτει το σύνολο των μηχανών που μπορούν να εργαστούν στο χρωμόσωμα.

Παράδειγμα:

Έστω χρωμόσωμα με 5 γονίδια – φάσεις:

ΦΑΣΕΙΣ	ΜΗΧΑΝΕΣ (CanPerform)	Ικανότητα
Φάση 1 ^η	a	
	b	
	c	
Φάση 2 ^η	a	
	b	
Φάση 3 ^η	e	
Φάση 4 ^η	g	
Φάση 5 ^η	a	
	h	

=> Σύνολο μηχανών= [a, b, c, e, h, g]

Στη συνέχεια, αντιστοιχούμε τους κωδικούς τους αριθμούς σε συνεχόμενους ακεραίους, ξεκινώντας από το 1. Πλέον οι μηχανές θα αντιπροσωπεύονται στο πρόβλημα από τους ακεραίους αυτούς.

=> Σύνολο μηχανών= [a, b, c, e, h, g]= [1, 2, 3, 4, 5, 6]

ΦΑΣΕΙΣ	ΜΗΧΑΝΕΣ (CanPerform)	Ικανότητα
Φάση 1 ^η	1 (a)	
	2 (b)	
	3 (c)	
Φάση 2 ^η	1 (a)	
	2 (b)	
Φάση 3 ^η	4 (e)	
Φάση 4 ^η	6 (g)	
Φάση 5 ^η	1 (a)	
	5 (h)	

Το πεδίο ορισμού των γονιδίων προκύπτει από το 1 έως και το πλήθος των μηχανών δηλαδή είναι το σύνολο [1 6].

Είναι επιθυμητό να γίνεται πρώτα ομαδοποίηση των μηχανών κατά είδος και να αντιστοιχούνται στις μηχανές κάθε είδους συνεχόμενοι ακέραιοι. Έτσι είναι δυνατόν για κάθε γονίδιο (κατεργασία κάποιου είδους) να ορίσουμε ένα αρχικό πεδίο αριθμών μέσα στο οποίο θα κατασκευαστεί ο αρχικός πληθυσμός, το οποίο θα είναι μικρότερο του συνολικού πεδίου ορισμού. Με αυτήν την αρχική συνθήκη αναμένουμε ταχύτερη εύρεση της βέλτιστης λύσης. Αναλυτικότερα, τα πλεονεκτήματα της μεθόδου αυτής φαίνονται στο κεφάλαιο “Βελτιστοποίηση ενός προβλήματος”.

```
%Number of machines
mac
%pedio orismou gonidiwn (pinakas 2 x kat+1)
mul
%Arhiko diastima anazitisis lysis
Mih
```

Μας ενδιαφέρουν και οι χρονικές αποστάσεις μεταξύ των μηχανών ώστε να υπολογίσουμε τους χρόνους μεταφοράς των τεμαχίων από μηχανή σε μηχανή (πίνακας *Distance*).

```
%pinakas xronwn metaforas (minutes) apo mihani se mihani (mac x mac)
TransTime
%pinakas xronwn metaforas (minutes) apo/pros store (mac x 1)
StoreTime
```

Απομένει να αποθηκεύσουμε τους χρόνους των ενδιάμεσων φάσεων φόρτωσης,

προετοιμασίας και εκφόρτωσης των μηχανών (εγγραφές *Setup time*, *Loading Time* και *Unloading Time* από τους πίνακες των ιδιαίτερων χαρακτηριστικών των μηχανών ανάλογα με τη φύση της κατεργασίας που εκτελούν).

Για τη συμπλήρωση των αντίστοιχων πινάκων ελέγχουμε κάθε γονίδιο – κατεργασία με τις μηχανές που μπορούν να την αναλάβουν. Συγκεκριμένα:

- Συγκρίνουμε το μέγεθος και βάρος του τεμαχίου (εγγραφές *Initial geometrical data* και *Initial weight* του πίνακα *Components*) με το χώρο υποδοχής τεμαχίων της μηχανής (*Table length*, *Table width*, *Table Diameter*, *Height clearance*, *Max load capacity*).
- Συγκρίνουμε τις τεχνικές απαιτήσεις της κατεργασίας με τις προδιαγραφές της μηχανής.

Σε περίπτωση που οι παραπάνω έλεγχοι προκύψουν θετικοί τότε στην τρίτη στήλη του πίνακα *Phases/Machines* αποθηκεύεται η τιμή 1, αλλιώς η τιμή 0.

Παράδειγμα:

ΦΑΣΕΙΣ	ΜΗΧΑΝΕΣ (CanPerform)	Capability
Φάση 1 ^η	1	1
	2	1
	3	0
Φάση 2 ^η	1	1
	2	1
Φάση 3 ^η	4	1
Φάση 4 ^η	6	1
Φάση 5 ^η	1	1
	5	1

Για κάθε ζευγάρι κατεργασίας - μηχανής που παίρνει την τιμή “1” συμπληρώνουμε τους χρόνους *Loading Time*, *Setup Time*, *Unloading Time* στους αντίστοιχους πίνακες. Σε κάθε άλλη θέση του πίνακα τοποθετούμε την τιμή *Not Available / na*.

```

%Time value if machine doesn't meet with the technological requirements
%stands for 100 minutes
na=100;
%pinakas xronwn setup (minutes) opou rows->mihanes cols->katergasies (mac x kat)
SetupTime
%pinakas xronwn loading (minutes) opou rows->mihanes cols->katergasies (mac x
kat)
LoadTime
%pinakas xronwn unloading (minutes) opou rows->mihanes cols->katergasies (mac x
kat)
UnloadTime

```

Ο γενετικός αλγόριθμος διαλέγει τις μηχανές του χρωμοσώματος με τυχαίο τρόπο. Σε περίπτωση που για κάποια κατεργασία διαλέξει μια μηχανή η οποία δεν μπορεί να εργαστεί το μέγεθος $N_a=100$ θα δώσει μεγάλη τιμή στην αντικειμενική συνάρτηση του χρωμοσώματος αυτού και ο αλγόριθμος θα το απορρίψει.

Ο φόρτος εργασίας των μηχανών αποθηκεύεται στον αντίστοιχο πίνακα (από την εγγραφές του πίνακα *Workload*).

```

%pinakas fortou ergasias opou rows->weeks cols->mihanes (duration x mac)
WrkLoa

```

ΚΕΦΑΛΑΙΟ 5^ο

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΝΟΣ ΠΡΟΒΛΗΜΑΤΟΣ

Στο κεφάλαιο αυτό θα παρουσιάσουμε τη βελτιστοποίηση ενός προβλήματος με τη βοήθεια των γενετικών αλγορίθμων. Η παρουσίαση της διαδικασίας ξεκινάει από τη δημιουργία της βάσης δεδομένων, ακολουθεί η σύνδεσή της με τον κώδικα βελτιστοποίησης στη Matlab και ολοκληρώνεται με την βελτιστοποίηση του προβλήματος με τη μέθοδο των γενετικών αλγορίθμων. Το πρόβλημα που επιλέξαμε να βελτιστοποιήσουμε δεν αφορά την κατασκευή κάποιου συγκεκριμένου τεμαχίου αλλά είναι υποθετικό και περιλαμβάνει μόνο τις απαραίτητες πληροφορίες για την επίδειξη της διαδικασίας. Κατά αυτόν τον τρόπο έχουμε την δυνατότητα να παρουσιάσουμε πολλές πτυχές της δημιουργίας ενός εικονικού συστήματος από τα αρχικά στάδια, χωρίς περιορισμούς.

Στην παράγραφο 5.1 παρουσιάζεται το πρόβλημα και η βάση δεδομένων που δημιουργήθηκε γι αυτό.

Η παράγραφος 5.2 αφορά στον ενδιάμεσο πίνακα πριν την μεταφορά των δεδομένων σε κατάλληλη μορφή στη Matlab στην παράγραφο 5.3.

Στην 5.4 παρουσιάζονται και σχολιάζονται τα αποτελέσματα της βελτιστοποίησης του προβλήματος.

Στην παράγραφο 5.5 σχολιάζονται οι μεταβολές στην απόδοση του κώδικα βελτιστοποίησης έπειτα από αλλαγές που έγιναν στις παραμέτρους του.

5.1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Πρόκειται για τη βάση δεδομένων του προβλήματος που θα επιλύσουμε. Οι βάση δεδομένων δεν είναι ολοκληρωμένη. Εμφανίζονται μόνο οι απαραίτητες πληροφορίες που απαιτούνται από τον κώδικα βελτιστοποίησης. Αρκετές από τις στήλες που μένουν κενές, παραλείπονται. Οι πληροφορίες της παραγγελίας από τον πελάτη αποθηκεύονται στους πίνακες Production, Part, Components, Process Plan και Phases. Πληροφορίες που αφορούν στα πραγματικά συστήματα κατεργασιών αποθηκεύονται στους πίνακες Company, Machines, Distance, CanPerform και Workload.

Στο παράδειγμα αυτό οφείλουμε να τονίσουμε τα δύο σημαντικότερα σημεία:

1. Την ύπαρξη δύο διαφορετικών σχεδιασμών παραγωγής τεμαχίων (εναλλακτικά φασεολόγια).
2. Την ύπαρξη μηχανών με ικανότητες τόρνου και φρέζας μαζί, καθώς και άλλων μηχανών με δυνατότητες λιγότερες από τις συνηθισμένες για το είδος τους.

Production

Επιθυμούμε τη βελτιστοποίηση μίας παραγγελίας με χρονικό περιθώριο 6 εβδομάδων.

Production ID	Production description	Starting date	Max Duration (weeks)
Παραγωγή 1			6

Part

Οι παραγγελία αφορά στην παραγωγή δύο κομματιών, που αποτελούνται από 5 και 3 υποσύνολα αντίστοιχα.

Part ID	Part description	Number of components	Production ID
Κομμάτι 1.1		5	Παραγωγή 1
Κομμάτι 1.2		3	Παραγωγή 1

Components

Στον πίνακα εμφανίζονται τα τεμάχια - υποσύνολα που πρέπει να κατασκευαστούν και το μέγεθος της μερίδας για το καθένα από αυτά.

Component ID	Component description	Quantity per part	Total Quantity	Part ID
Τεμάχιο 1.1.1		1	50	Κομμάτι 1.1
Τεμάχιο 1.1.2		1	50	Κομμάτι 1.1
Τεμάχιο 1.1.3		1	50	Κομμάτι 1.1
Τεμάχιο 1.1.4		3	150	Κομμάτι 1.1
Τεμάχιο 1.1.5		3	150	Κομμάτι 1.1
Τεμάχιο 1.2.1		1	20	Κομμάτι 1.2
Τεμάχιο 1.2.2		1	20	Κομμάτι 1.2
Τεμάχιο 1.2.3		2	40	Κομμάτι 1.2

Process Plan - Phases

1^{ος} συνδυασμός φασεολογιών

Process Plan ID	Process Plan description	Number of phases
Φασεολόγιο 1.1.1.1	Τόρνευση, φρεζάρισμα	3
Φασεολόγιο 1.1.2.1	Τόρνευση, φρεζάρισμα	3
Φασεολόγιο 1.1.3.1	Τόρνευση, φρεζάρισμα, διάτρηση, γλύφανση	4
Φασεολόγιο 1.1.4.1	Φρεζάρισμα, διάτρηση	2
Φασεολόγιο 1.1.5.1	Χύτευση	1
Φασεολόγιο 1.2.1.1	Τόρνευση, φρεζάρισμα, διάτρηση	5
Φασεολόγιο 1.2.2.1	Τόρνευση, φρεζάρισμα, διάτρηση, γλύφανση	5
Φασεολόγιο 1.2.3.1	Φρεζάρισμα, διάτρηση, EDM	4

(συνέχεια)

Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Component ID
Φάση 01	Φάση 02	Φάση 03			Τεμάχιο 1.1.1
Φάση 04	Φάση 05	Φάση 06			Τεμάχιο 1.1.2
Φάση 07	Φάση 08	Φάση 09	Φάση 10		Τεμάχιο 1.1.3
Φάση 11	Φάση 12				Τεμάχιο 1.1.4
Φάση 13					Τεμάχιο 1.1.5
Φάση 14	Φάση 15	Φάση 16	Φάση 17	Φάση 18	Τεμάχιο 1.2.1
Φάση 19	Φάση 20	Φάση 21	Φάση 22	Φάση 23	Τεμάχιο 1.2.2
Φάση 24	Φάση 25	Φάση 26	Φάση 27		Τεμάχιο 1.2.3

Κάθε φάση είναι συγκεκριμένου είδους και χαρακτηρίζεται από κάποιον χρόνο ολοκλήρωσης.

Phase ID	Process Description	Process Type	Time	Cost
Φάση 01		Turning	3	
Φάση 02		Turning	5	
Φάση 03		Milling	6	
Φάση 04		Turning	5	
Φάση 05		Milling	3	
Φάση 06		Drilling	1	
Φάση 07		Turning	5	
Φάση 08		Milling	2	
Φάση 09		Drilling	2	
Φάση 10		Reaming	1	
Φάση 11		Milling	10	
Φάση 12		Drilling	1	
Φάση 13		Die Casting	3	
Φάση 14		Turning	2	
Φάση 15		Turning	8	
Φάση 16		Cylindrical Grinding	3	
Φάση 17		Milling	6	
Φάση 18		Drilling	2	
Φάση 19		Turning	5	
Φάση 20		Boring	2	
Φάση 21		Milling	5	
Φάση 22		Drilling	2	
Φάση 23		Reaming	2	
Φάση 24		Milling	5	
Φάση 25		Milling	1	
Φάση 26		Drilling	1	
Φάση 27		EDM die sinking	15	

Τα τεμάχια 1.1.2 και 1.2.1 έχουν εναλλακτικούς τρόπους παραγωγής. Ένα από τα εναλλακτικά σχέδια παραγωγής που δημιουργούνται είναι το ακόλουθο. Οι μοναδικές αλλαγές βρίσκονται στα φασεολόγια παραγωγής των τεμαχίων αυτών και στην αρίθμηση των κατεργασιών.

2ος εναλλακτικός συνδυασμός φασεολογίων

Process Plan ID	Process Plan description	Number of phases
Φασεολόγιο 1.1.1.1	Τόρνευση, φρεζάρισμα	3
Φασεολόγιο 1.1.2.2	Χύτευση, λείανση	2
Φασεολόγιο 1.1.3.1	Τόρνευση, φρεζάρισμα, διάτρηση, γλύφανση	4
Φασεολόγιο 1.1.4.1	Φρεζάρισμα, διάτρηση	2
Φασεολόγιο 1.1.5.1	Χύτευση	1
Φασεολόγιο 1.2.1.2	Χύτευση, EDM	2
Φασεολόγιο 1.2.2.1	Τόρνευση, φρεζάρισμα, διάτρηση, γλύφανση	5
Φασεολόγιο 1.2.3.1	Φρεζάρισμα, διάτρηση, EDM	4

(συνέχεια)

Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Component ID
Φάση 01	Φάση 02	Φάση 03			Τεμάχιο 1.1.1
Φάση 04	Φάση 05				Τεμάχιο 1.1.2
Φάση 06	Φάση 07	Φάση 08	Φάση 09		Τεμάχιο 1.1.3
Φάση 10	Φάση 11				Τεμάχιο 1.1.4
Φάση 12					Τεμάχιο 1.1.5
Φάση 13	Φάση 14				Τεμάχιο 1.2.1
Φάση 15	Φάση 16	Φάση 17	Φάση 18	Φάση 19	Τεμάχιο 1.2.2
Φάση 20	Φάση 21	Φάση 22	Φάση 23		Τεμάχιο 1.2.3

Σε κάθε φάση αντιστοιχούν οι εξής πληροφορίες.

Phase ID	Process Description	Process Type	Time	Cost
Φάση 01		Turning	3	
Φάση 02		Turning	5	
Φάση 03		Milling	6	
Φάση 04		Die casting	5	
Φάση 05		Cylindrical grinding	8	
Φάση 06		Turning	5	
Φάση 07		Milling	2	
Φάση 08		Drilling	2	

Φάση 09		Reaming	1	
Φάση 10		Milling	10	
Φάση 11		Drilling	1	
Φάση 12		Die Casting	3	
Φάση 13		Die casting	10	
Φάση 14		EDM die sinking	8	
Φάση 15		Turning	5	
Φάση 16		Boring	2	
Φάση 17		Milling	5	
Φάση 18		Drilling	2	
Φάση 19		Reaming	2	
Φάση 20		Milling	5	
Φάση 21		Milling	1	
Φάση 22		Drilling	1	
Φάση 23		EDM die sinking	15	

Company

Τα πραγματικά συστήματα κατεργασιών αποτελούνται από μηχανές που ανήκουν σε 5 κατασκευαστές.

Company ID	Company Description	Number of machines	Distance
Κατασκευαστής 1		5	Αποστάσεις 1
Κατασκευαστής 2		4	Αποστάσεις 2
Κατασκευαστής 3		3	Αποστάσεις 3
Κατασκευαστής 4		3	Αποστάσεις 4
Κατασκευαστής 5		3	Αποστάσεις 5

Distance

Οι χρονικές αποστάσεις μεταξύ των κατασκευαστών καθώς και από την αποθήκη πρώτων υλών είναι αποθηκευμένες στον πίνακα αυτό.

Distance ID	Κατ 1 (min)	Κατ 2 (min)	Κατ 3 (min)	Κατ 4 (min)	Κατ 5 (min)	Distance from Store (min)
Αποστάσεις 1	0	120	180	360	540	60
Αποστάσεις 2	120	0	240	420	420	60
Αποστάσεις 3	180	240	0	120	360	240
Αποστάσεις 4	360	420	120	0	120	420
Αποστάσεις 5	540	420	360	120	0	540

Workload

Αρχικά θεωρούμε χαμηλό φόρτο εργασίας (40%) για όλες τις μηχανές, ώστε η βελτιστοποίηση του χρωμοσώματος να μην επηρεάζεται από τυχόν ποινές. Στη συνέχεια θα επιλύσουμε πάλι το πρόβλημα με υψηλό φόρτο εργασίας (95%) για επιλεγμένες μηχανές ώστε να μελετήσουμε την μεταβολή στη βέλτιστη λύση.

Machine

18 μηχανές συνολικά (όπου 4 τόρνοι, 1 τόννος με ικανότητες φρέζας, 3 φρέζες, 1 φρέζα με ικανότητες τόννου, 2 δράπανα, 2 μηχανές για κυλινδρική λείανση, 3 μηχανές χύτευσης και 2 μηχανές EDM).

Οι κωδικοί των μηχανών έχουν δοθεί έτσι ώστε συνεχόμενοι ακέραιοι αριθμοί να αντιστοιχούν σε μηχανές του ίδιου τύπου. Συγκεκριμένα δόθηκαν οι τιμές, 1-3 στις φρέζες, 4 στη φρέζα με δυνατότητες τόννου, 5-6 στα δράπανα, 7-10 στους τόννους, 11 στον τόννο με δυνατότητες φρέζας, 12-13 στις μηχανές κυλινδρικής λείανσης, 14-15 στις μηχανές EDM και 16-18 στις μηχανές χύτευσης. Ο τρόπος αυτός μας επιτρέπει να ορίσουμε κατευθείαν ένα πεδίο, μικρότερο του πεδίου ορισμού, μέσα στο οποίο θα κατασκευαστεί ο αρχικός πληθυσμός. Σε κάθε άλλη περίπτωση θα έπρεπε να κατασκευάσουμε έναν πίνακα αντιστοιχίας κωδικών – ακεραίων.

Machine ID	Machine type	Workload	Company ID
Μηχ 7	Lathe	Workload 7	Κατασκευαστής 1
Μηχ 8	Lathe	Workload 8	Κατασκευαστής 1
Μηχ 1	Machining center	Workload 1	Κατασκευαστής 1
Μηχ 5	Drill	Workload 5	Κατασκευαστής 1
Μηχ 16	Die casting machine	Workload 16	Κατασκευαστής 1
Μηχ 9	Lathe	Workload 9	Κατασκευαστής 2
Μηχ 2	Machining center	Workload 2	Κατασκευαστής 2
Μηχ 3	Machining center	Workload 3	Κατασκευαστής 2
Μηχ 12	Cylindrical grinding machine	Workload 12	Κατασκευαστής 2
Μηχ 10	Lathe	Workload 10	Κατασκευαστής 3
Μηχ 6	Drill	Workload 6	Κατασκευαστής 3
Μηχ 13	Cylindrical grinding machine	Workload 13	Κατασκευαστής 3
Μηχ 17	Die casting machine	Workload 17	Κατασκευαστής 4

Μηχ 18	Die casting Machine	Workload 18	Κατασκευαστής 4
Μηχ 14	Sink EDM machine	Workload 14	Κατασκευαστής 4
Μηχ 11	Multi tasking lathe	Workload 11	Κατασκευαστής 5
Μηχ 4	Machining center with turning tools	Workload 4	Κατασκευαστής 5
Μηχ 15	Sink EDM machine	Workload 15	Κατασκευαστής 5

CanPerform

Τα σημεία που πρέπει να δώσουμε προσοχή είναι υπογραμμισμένα. Πρόκειται για τον τόρνο (μηχανή 9) που αδυνατεί να εκτελέσει κατεργασία Boring, για την πολυμηχανή τύπου Multi tasking lathe (μηχανή 11) που είναι ικανή για κατεργασίες Milling, Drilling, Reaming, Turning και Boring και για την πολυμηχανή τύπου Machining center with turning tools (μηχανή 4) που είναι ικανή για κατεργασίες τύπου Milling, Drilling, Reaming και Turning.

CanPerform ID	Machine ID	Milling	Drilling	Reaming	Turning	Boring
CanPerform 7	Μηχ 7				yes	yes
CanPerform 8	Μηχ 8				yes	yes
CanPerform 1	Μηχ 1	yes	yes	yes		
CanPerform 5	Μηχ 5		yes	yes		
CanPerform 16	Μηχ 16					
CanPerform 9	Μηχ 9				yes	no
CanPerform 2	Μηχ 2	yes	yes	yes		
CanPerform 3	Μηχ 3	yes	yes	yes		
CanPerform 12	Μηχ 12					
CanPerform 10	Μηχ 10				yes	yes
CanPerform 6	Μηχ 6		yes	yes		
CanPerform 13	Μηχ 13					
CanPerform 17	Μηχ 17					
CanPerform 18	Μηχ 18					
CanPerform 14	Μηχ 14					
CanPerform 11	Μηχ 11	yes	yes	yes	yes	yes
CanPerform 4	Μηχ 4	yes	yes	yes	yes	
CanPerform 15	Μηχ 15					

(συνέχεια)

Machine ID	...	Cylindrical grinding	EDM die sinking	Die casting
Μηχ 7				
Μηχ 8				
Μηχ 1				
Μηχ 5				
Μηχ 16				yes
Μηχ 9				
Μηχ 2				
Μηχ 3				
Μηχ 12		yes		
Μηχ 10				
Μηχ 6				
Μηχ 13		yes		
Μηχ 17				yes
Μηχ 18				yes
Μηχ 14			yes	
Μηχ 11				
Μηχ 4				
Μηχ 15			yes	

Χαρακτηριστικά μηχανών

Machining center ID	Machine ID	Setup time (min)	Load time (min)	Unload time (min)
1	Μηχ 1	15	1	1
2	Μηχ 2	25	3	3
3	Μηχ 3	17	1	1
4	Μηχ 4	12	0.5	0.5
5	Μηχ 11	10	0.5	0.5

Drill ID	Machine ID	Setup time (min)	Load time (min)	Unload time (min)
1	Μηχ 5	5	0.5	0.5
2	Μηχ 6	6	0.6	0.6

Lathe ID	Machine ID	Setup time (min)	Load time (min)	Unload time (min)
1	Μηχ 7	15	1	1
2	Μηχ 8	20	1.5	1.5
3	Μηχ 9	18	1.5	1.5
4	Μηχ 10	25	3	3
5	Μηχ 11	10	0.5	0.5
6	Μηχ 4	12	0.5	0.5

Cylindrical grinding machine ID	Machine ID	Setup time (min)	Load time (min)	Unload time (min)
1	Mηχ 12	20	4	4
2	Mηχ 13	15	2	2

Sink EDM machine ID	Machine ID	Setup time (min)	Load time (min)	Unload time (min)
1	Mηχ 14	30	2	2
2	Mηχ 15	40	2	2

Die casting machine ID	Machine ID	Setup time (min)	Load time (min)	Unload time (min)
1	Mηχ 16	120	0	1
2	Mηχ 17	150	0	1.5
3	Mηχ 18	80	0	2

Οι κατασκευαστές με τους τύπους μηχανών που διαθέτουν σχηματικά:

Store

Company 1

2 lathes
1 machining center
1 drill
1 die casting machine

Company 2

1 lathe
2 machining centers
1 cylindrical grinding machine

Company 3

1 lathe
1 drill
1 cylindrical grinding machine

Company 4

2 die casting machines
1 Sink EDM machine

Company 5

1 multi tasking lathe
1 machining center with turning tools
1 Sink EDM machine

5.2 ΠΙΝΑΚΑΣ PHASES/MACHINES

Πρόκειται για τον βοηθητικό πίνακα που συνδέει τη βάση δεδομένων με το αρχείο fdatabase.m του κώδικα βελτιστοποίησης. Στον πίνακα αντιστοιχίζεται κάθε κατεργασία του φασεολογίου στις μηχανές που δύνανται να την εκτελέσουν αφού συγκριθούν οι απαιτήσεις χώρου και οι τεχνικές απαιτήσεις της κάθε φάσης με τις προδιαγραφές των μηχανών.

Υποθέτουμε ότι:

1. Το τεμάχιο 1.1.1 (φάσεις 01, 02, 03) δεν χωράει στην μηχανή 8. Η αντίστοιχη στήλη ικανότητας συμπληρώθηκε με 0.
2. Οι τεχνικές απαιτήσεις της κατεργασίας 11 είναι ανώτερες των προδιαγραφών της μηχανής 3. Η στήλη της ικανότητας συμπληρώθηκε με 0.

1ος συνδυασμός φασεολογιών

Φάσεις	Είδος	Μηχανές	Ικανότητα
Φάση 01	turning	4	1
		7	1
		8	0
		9	1
		10	1
		11	1
Φάση 02	turning	4	1
		7	1
		8	0
		9	1
		10	1
		11	1
Φάση 03	milling	1	1
		2	1
		3	1
		4	1
		11	1

Φάση 04	turning	4	1
		7	1
		8	1
		9	1
		10	1
		11	1
Φάση 05	milling	1	1
		2	1
		3	1
		4	1
		11	1
Φάση 06	drilling	1	1
		2	1
		3	1
		4	1
		5	1
		6	1
		11	1
Φάση 07	turning	4	1
		7	1
		8	1
		9	1
		10	1
		11	1
Φάση 08	milling	1	1
		2	1
		3	1
		4	1
		11	1
Φάση 09	drilling	1	1
		2	1
		3	1
		4	1
		5	1
		6	1
		11	1
Φάση 10	reaming	1	1
		2	1
		3	1
		4	1
		5	1

		6	1
		11	1
Φάση 11	milling	1	1
		2	1
		3	0
		4	1
		11	1
Φάση 12	drilling	2	1
		3	1
		4	1
		5	1
		6	1
		11	1
Φάση 13	Die casting	17	1
		18	1
Φάση 14	turning	7	1
		8	1
		9	1
		10	1
		11	1
Φάση 15	turning	7	1
		8	1
		9	1
		10	1
		11	1
Φάση 16	Cylindrical grinding	13	1
Φάση 17	milling	2	1
		3	1
		4	1
		11	1
Φάση 18	drilling	2	1
		3	1
		4	1
		5	1

		6	1
		11	1
Φάση 19	turning	4	1
		7	1
		8	1
		9	1
		10	1
		11	1
		Φάση 20	boring
8	1		
10	1		
11	1		
Φάση 21	milling	1	1
		2	1
		3	1
		4	1
		11	1
Φάση 22	drilling	1	1
		2	1
		3	1
		4	1
		5	1
		6	1
		11	1
Φάση 23	reaming	1	1
		2	1
		3	1
		4	1
		5	1
		6	1
		11	1
Φάση 24	milling	1	1
		2	1
		3	1
		4	1
		11	1
Φάση 25	milling	1	1
		2	1
		3	1
		4	1
		11	1

Φάση 26	drilling	1	1
		2	1
		3	1
		4	1
		5	1
		6	1
		11	1
Φάση 27	EDM die sinking	14	1
		15	1

Στον πίνακα αυτόν φαίνεται καθαρά το πεδίο αριθμών - μηχανών μέσα στο οποίο θα κατασκευαστεί ο αρχικός πληθυσμός. Συγκεκριμένα:

turning -> 7-11

milling-> 1-4

drilling -> 1-6

reaming -> 1-6

boring -> 7-11

cylindrical grinding -> 12-13

EDM die sinking -> 14-15

die casting -> 16-18

Φυσικά, η αναζήτηση της λύσης από τον γενετικό δε θα περιοριστεί στο πεδίο αυτό, αλλά θα εξαπλωθεί σε ολόκληρο το πεδίο ορισμού. Τα πλεονεκτήματα ορισμού ενός αρχικού πεδίου θα φανούν στη συνέχεια (παράγραφος “Πειραματισμοί με παραμέτρους των γενετικών” - Πείραμα 1°).

2ος συνδυασμός φασεολογίων

κατασκευάζεται στην ίδια λογική με τον προηγούμενο

5.3 fdatabase.m

Από την βάση δεδομένων, οι απαραίτητες πληροφορίες για τη βελτιστοποίηση του προβλήματος αποθηκεύονται σε πίνακες στη Matlab.

Τα bonus και τα penalty των χρωμοσωμάτων ορίζονται εδώ (πίνακας ισοτιμίας). Συγκεκριμένα:

1. Bonus 300' για κάθε μερίδα που περιμένει ώστε να μεταφερθεί μαζί με κάποια άλλη στον επόμενο κατασκευαστή.
2. Bonus 150' για κάθε μεταφορά μερίδας που γλιτώνουμε από την αποθήκη ή προς την αποθήκη.
3. Penalty 2880 λεπτών για κάθε εβδομάδα που καθυστερούμε την παράδοση.
4. Penalty 480 λεπτών όταν χρησιμοποιούμε μηχανή με υψηλό φόρτο εργασίας.

Οι χρόνοι Setup, Loading και Unloading μιας μηχανής που δε δύναται να εκτελέσει κάποια κατεργασία ορίζονται στα $pa=100$ λεπτά (χρόνος πολύ μεγαλύτερος από τον φυσιολογικό. Είναι αρκετός για την απώριψη του χρωμοσώματος από τον κώδικα).

1^{ος} συνδυασμός φασεολογιών

```
%Kathe bonus - penalty idodynamei me xrono (se lepta)
%   bonus kathe fora pou perimenoume gia metafora
%   bonus gia kathe metafora pou glitwnoume apo kai pros Store
%   penalty gia kathe evdomada pou kathisteroume tin paradosi
%   penalty gia kathe xrisi mihanis me ipsilo forto ergasias
isotimia=[-300 -150 2880 480];
%Max project duration (weeks)
duration=6;
%Number of parts
par=8;
%Arhi kai telos gonidiou gia kathe part (pinakas par x 2)
orio=[1 3; 4 6; 7 10; 11 12; 13 13; 14 18; 19 23; 24 27];
%Synolo katergasiwn
kat=27;
%Megethos meridas tou kathe part (pinakas 1 x par)
Qua=[50 50 50 150 150 20 20 40];
%Number of machines
```

```

mac=18;
%Time value if machine doesn't meet with the technological requirements
%stands for 100 minutes
na=100;

%katharoi xronoi katergasias (minutes) gia kathe fasi (pinakas 1 x kat)
ProcessTime=[3 5 6 5 3 1 5 2 2 1 10 1 3 2 8 3 6 2 5 2 5 2 2 5 1 1 15];

%pinakas xronwn setup (minutes) rows->mihanes cols->katergasies (mac x kat)
SetupTime=
[na na 15 na 15 15 na 15 15 15 15 15 na na na na 15 15 na na 15 15 15 15 15 15 na;
na na 25 na 25 25 na 25 25 25 25 25 na na na na 25 25 na na 25 25 25 25 25 25 na;
na na 17 na 17 17 na 17 17 17 na 17 na na na na 17 17 na na 17 17 17 17 17 17 na;
12 12 12 12 12 12 12 12 12 12 12 na 12 12 na 12 12 12 na 12 12 12 12 12 12 na;
na na na na na 5 na na 5 5 na 5 na na na na na 5 na na na 5 5 na na 5 na;
na na na na na 6 na na 6 6 na 6 na na na na na 6 na na na 6 6 na na 6 na;
15 15 na 15 na na 15 na na na na na na 15 15 na na na 15 15 na na na na na na;
na na na 20 na na 20 na na na na na na 20 20 na na na 20 20 na na na na na na;
18 18 na 18 na na 18 na na na na na na 18 18 na na na 18 na na na na na na na;
25 25 na 25 na na 25 na na na na na na 25 25 na na na 25 25 na na na na na na;
10 10 10 10 10 10 10 10 10 10 10 na 10 10 na 10 10 10 10 10 10 10 10 10 na;
na na na na na na na na na na na na na na na na 20 na na na na na na na na na;
na na na na na na na na na na na na na na na na na 15 na na na na na na na na na;
na na na na na na na na na na na na na na na na na na na na na na na na 30;
na na na na na na na na na na na na na na na na na na na na na na na na 40;
na na na na na na na na na na na na na 120 na na na na na na na na na na na na;
na na na na na na na na na na na na na 150 na na na na na na na na na na na na;
na na na na na na na na na na na na na 80 na na na na na na na na na na na na];

%pinakas xronwn loading (minutes) rows->mihanes cols->katergasies (mac x kat)
LoadTime=
[na na 1 na 1 1 na 1 1 1 1 1 na na na na 1 1 na na 1 1 1 1 1 1 na;
na na 3 na 3 3 na 3 3 3 3 3 na na na na 3 3 na na 3 3 3 3 3 3 na;
na na 1 na 1 1 na 1 1 1 na 1 na na na na 1 1 na na 1 1 1 1 1 1 na;
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 na 0.5 0.5 na 0.5 0.5 0.5 na 0.5 0.5 0.5
0.5 0.5 0.5 na;
na na na na na 0.5 na na 0.5 0.5 na 0.5 na na na na na 0.5 na na na 0.5 0.5 na na 0.5 na;
na na na na na 0.6 na na 0.6 0.6 na 0.6 na na na na na 0.6 na na na 0.6 0.6 na na 0.6 na;
1 1 na 1 na na 1 na na na na na na 1 1 na na na 1 1 na na na na na na na;
na na na 1.5 na na 1.5 na na na na na na 1.5 1.5 na na na 1.5 1.5 na na na na na na;
1.5 1.5 na 1.5 na na 1.5 na na na na na na 1.5 1.5 na na na 1.5 na na na na na na;
3 3 na 3 na na 3 na na na na na na 3 3 na na na 3 3 na na na na na na na;
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 na 0.5 0.5 na 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 na;
na na na na na na na na na na na na na na na 4 na na na na na na na na na na;
na na na na na na na na na na na na na na na 2 na na na na na na na na na na na];

```

```

na na na na na na na na na na na na na na na na na na na na na na na na na na 2;
na na na na na na na na na na na na na na na na na na na na na na na na na na 2;
na na na na na na na na na na na na na 0 na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na 0 na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na 0 na na na na na na na na na na na na na na na];

%pinakas xronwn unloading (minutes) rows->mihanes cols->katergasies (mac x kat)
UnloadTime=
[na na 1 na 1 1 na 1 1 1 1 1 na na na na 1 1 na na 1 1 1 1 1 1 na;
na na 3 na 3 3 na 3 3 3 3 3 na na na na 3 3 na na 3 3 3 3 3 3 na;
na na 1 na 1 1 na 1 1 1 na 1 na na na na 1 1 na na 1 1 1 1 1 1 na;
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 na 0.5 0.5 na 0.5 0.5 0.5 na 0.5 0.5 0.5
0.5 0.5 0.5 na;
na na na na na na 0.5 na na 0.5 0.5 na 0.5 na na na na na na 0.5 na na na 0.5 0.5 na na 0.5 na;
na na na na na na 0.6 na na 0.6 0.6 na 0.6 na na na na na na 0.6 na na na 0.6 0.6 na na 0.6 na;
1 1 na 1 na na 1 na na na na na na na 1 1 na na na 1 1 na na na na na na na;
na na na 1.5 na na 1.5 na na na na na na na 1.5 1.5 na na na 1.5 1.5 na na na na na na na;
1.5 1.5 na 1.5 na na 1.5 na na na na na na na 1.5 1.5 na na na 1.5 na na na na na na na;
3 3 na 3 na na 3 na na na na na na na 3 3 na na na 3 3 na na na na na na na;
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 na 0.5 0.5 na 0.5 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 na;
na na na na na na na na na na na na na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na na na na na na na na na na na na na na na];

%pinakas xronwn metaforas (minutes) apo mihani se mihani (mac x mac)
TransTime=[0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
120 0 0 420 120 240 120 120 0 240 420 0 240 420 420 120 420 420;
120 0 0 420 120 240 120 120 0 240 420 0 240 420 420 120 420 420;
540 420 420 0 540 360 540 540 420 360 0 420 360 120 120 540 120 120;
0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
180 240 240 360 180 0 180 180 240 0 360 240 0 120 360 180 120 120;
0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
120 0 0 420 120 240 120 120 0 240 420 0 240 420 420 120 420 420;
180 240 240 360 180 0 180 180 240 0 360 240 0 120 360 180 120 120;
540 420 420 0 540 360 540 540 420 360 0 420 360 120 120 540 120 120;
120 0 0 420 120 240 120 120 0 240 420 0 240 420 420 120 420 420;
180 240 240 360 180 0 180 180 240 0 360 240 0 120 360 180 120 120;
360 420 420 120 360 120 360 360 420 120 120 420 120 0 120 360 0 0;
540 420 420 0 540 360 540 540 420 360 0 420 360 120 120 540 120 120;
0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;

```

```

360 420 420 120 360 120 360 360 420 120 120 420 120 0 120 360 0 0;
360 420 420 120 360 120 360 360 420 120 120 420 120 0 120 360 0 0];

%pinakas xronwn metaforas (minutes) apo/pros store (mac x 1)
StoreTime=
[60; 60; 60; 540; 60; 240; 60; 60; 60; 240; 540; 60; 240; 420; 540; 60; 420; 420];

%pinakas fortou ergasias rows->weeks cols->mihanes (duration x mac)
WrkLoa=[0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4];

```

2^{ος} συνδυασμός φασεολογίων

```

%Kathe bonus - penalty idodynamei me xrono (se lepta):
isotimia=[-300 -150 2880 480];
%Max project duration (weeks)
duration=6;
%Number of parts
par=8;
%Arhi kai telos gonidiou gia kathe part (pinakas par x 2)
orio=[1 3; 4 5; 6 9; 10 11; 12 12; 13 14; 15 19; 20 23];
%Synolo katergasiwn
kat=23;
%Megethos meridas tou kathe part (pinakas 1 x par)
Qua=[50 50 50 150 150 20 20 40];
%Number of machines
mac=18;
%Time value if machine doesn't meet with the technological requirements
%stands for 100 minutes
na=100;

%katharoi xronoi katergasias (minutes) gia kathe fasi (pinakas 1 x kat)
ProcessTime=[3 5 6 5 8 5 2 2 1 10 1 3 10 8 5 2 5 2 2 5 1 1 15];

%pinakas xronwn setup (minutes) rows->mihanes cols->katergasies (mac x kat)
SetupTime=[na na 15 na na na 15 15 15 15 15 na na na na na 15 15 15 15 15 15 na;
na na 25 na na na 25 25 25 25 25 na na na na na 25 25 25 25 25 25 na;
na na 17 na na na 17 17 17 na 17 na na na na na 17 17 17 17 17 17 na;
12 12 12 na na 12 12 12 12 12 12 na na na 12 12 12 12 12 12 12 12 na;
na na na na na na na 5 5 na 5 na na na na na 5 5 na na 5 na;
na na na na na na na 6 6 na 6 na na na na na na 6 6 na na 6 na;

```



```

15 15 na na na 15 na na na na na na na na 15 15 na na na na na na na;
na na na na na 20 na na na na na na na na na 20 20 na na na na na na na;
18 18 na na na 18 na na na na na na na na na 18 na na na na na na na na;
25 25 na na na 25 na na na na na na na na na 25 25 na na na na na na na;
10 10 10 na na 10 10 10 10 10 10 na na na 10 10 10 10 10 10 10 na;
na na na na na 20 na na na na na na na na na na na na na na na na na;
na na na na na 15 na na na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na na na 30 na na na na na na na na 30;
na na na na na na na na na na na na na na na 40 na na na na na na na na 40;
na na na 120 na na na na na na na na 120 120 na na na na na na na na na;
na na na 150 na na na na na na na na 150 150 na na na na na na na na na;
na na na 80 na na na na na na na na 80 80 na na na na na na na na na];

```

%pinakas xronwn loading (minutes) rows->mihanes cols->katergasies (mac x kat)

LoadTime=

```

[na na 1 na na na 1 1 1 1 1 na na na na na 1 1 1 1 1 1 na;
na na 3 na na na 3 3 3 3 3 na na na na na 3 3 3 3 3 3 na;
na na 1 na na na 1 1 1 na 1 na na na na na 1 1 1 1 1 1 na;
0.5 0.5 0.5 na na 0.5 0.5 0.5 0.5 0.5 0.5 na na na 0.5 0.5 0.5 0.5 0.5 0.5 0.5 na;
na na na na na na na 0.5 0.5 na 0.5 na na na na na na 0.5 0.5 na na 0.5 na;
na na na na na na na 0.6 0.6 na 0.6 na na na na na na 0.6 0.6 na na 0.6 na;
1 1 na na na 1 na na na na na na na na 1 1 na na na na na na na;
na na na na na 1.5 na na na na na na na na 1.5 1.5 na na na na na na na;
1.5 1.5 na na na 1.5 na na na na na na na na 1.5 na na na na na na na;
3 3 na na na 3 na na na na na na na na 3 3 na na na na na na na;
0.5 0.5 0.5 na na 0.5 0.5 0.5 0.5 0.5 0.5 na na na 0.5 0.5 0.5 0.5 0.5 0.5 0.5 na;
na na na na 4 na na na na na na na na na na na na na na na na na;
na na na na 2 na na na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na na 2 na na na na na na na na 2;
na na na na na na na na na na na na na na 2 na na na na na na na na 2;
na na na 0 na na na na na na na 0 0 na na na na na na na na na na;
na na na 0 na na na na na na na 0 0 na na na na na na na na na na;
na na na 0 na na na na na na na 0 0 na na na na na na na na na na];

```

%pinakas xronwn unloading (minutes) rows->mihanes cols->katergasies (mac x kat)

UnloadTime=

```

[na na 1 na na na 1 1 1 1 1 na na na na na 1 1 1 1 1 1 na;
na na 3 na na na 3 3 3 3 3 na na na na na 3 3 3 3 3 3 na;
na na 1 na na na 1 1 1 na 1 na na na na na 1 1 1 1 1 1 na;
0.5 0.5 0.5 na na 0.5 0.5 0.5 0.5 0.5 0.5 na na na 0.5 0.5 0.5 0.5 0.5 0.5 0.5 na;
na na na na na na na 0.5 0.5 na 0.5 na na na na na na 0.5 0.5 na na 0.5 na;
na na na na na na na 0.6 0.6 na 0.6 na na na na na na 0.6 0.6 na na 0.6 na;
1 1 na na na 1 na na na na na na na na 1 1 na na na na na na na;
na na na na na 1.5 na na na na na na na na 1.5 1.5 na na na na na na na;
1.5 1.5 na na na 1.5 na na na na na na na na 1.5 na na na na na na na;
3 3 na na na 3 na na na na na na na na 3 3 na na na na na na na;

```

```
0.5 0.5 0.5 na na 0.5 0.5 0.5 0.5 0.5 0.5 na na na 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 na;
na na na na 4 na na na na na na na na na na na na na na na na na;
na na na na 2 na na na na na na na na na na na na na na na na na;
na na na na na na na na na na na na na 2 na na na na na na na na 2;
na na na na na na na na na na na na na 2 na na na na na na na na 2;
na na na 1 na na na na na na na na 1 1 na na na na na na na na na;
na na na 1.5 na na na na na na na 1.5 1.5 na na na na na na na na na;
na na na 1 na na na na na na na 1 1 na na na na na na na na na];
```

```
%pinakas xronwn metaforas (minutes) apo mihani se mihani (mac x mac)
```

```
TransTime=[0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
            120 0 0 420 120 240 120 120 0 240 420 0 240 420 420 120 420 420;
            120 0 0 420 120 240 120 120 0 240 420 0 240 420 420 120 420 420;
            540 420 420 0 540 360 540 540 420 360 0 420 360 120 120 540 120 120;
            0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
            180 240 240 360 180 0 180 180 240 0 360 240 0 120 360 180 120 120;
            0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
            0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
            120 0 0 420 120 240 120 120 0 240 420 0 240 420 420 120 420 420;
            180 240 240 360 180 0 180 180 240 0 360 240 0 120 360 180 120 120;
            540 420 420 0 540 360 540 540 420 360 0 420 360 120 120 540 120 120;
            120 0 0 420 120 240 120 120 0 240 420 0 240 420 420 120 420 420;
            180 240 240 360 180 0 180 180 240 0 360 240 0 120 360 180 120 120;
            360 420 420 120 360 120 360 360 420 120 120 420 120 0 120 360 0 0;
            540 420 420 0 540 360 540 540 420 360 0 420 360 120 120 540 120 120;
            0 120 120 540 0 180 0 0 120 180 540 120 180 360 540 0 360 360;
            360 420 420 120 360 120 360 360 420 120 120 420 120 0 120 360 0 0;
            360 420 420 120 360 120 360 360 420 120 120 420 120 0 120 360 0 0];
```

```
%pinakas xronwn metaforas (minutes) apo/pros store (mac x 1)
```

```
StoreTime=
[60; 60; 60; 540; 60; 240; 60; 60; 60; 240; 540; 60; 240; 420; 540; 60; 420; 420];
```

```
%pinakas fortou ergasias rows->weeks cols->mihanes (duration x mac)
```

```
WrkLoa=[0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4];
```

5.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

1^{ος} συνδυασμός φασεολογιών

Παράμετροι βελτιστοποίησης:

Number of variables	28
Constraints - Lower bound	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Constraints - Upper bound	[18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 6]
Stopping Criteria	Stall generations= 50
Population	300
Initial range	[7 7 1 7 1 1 7 1 1 1 1 1 1 16 7 7 12 1 1 7 7 1 1 1 1 1 1 14 1; 11 11 4 11 4 6 11 4 6 6 4 6 18 11 11 13 4 6 11 11 4 6 6 4 4 6 15 6]
Creation function	Uniform
Scaling function	Rank
Selection function	Stochastic uniform
Elite count	16
Crossover function	0.85
Mutation function	Gaussian
Crossover function	Scattered

Βέλτιστο αποτέλεσμα:

Χρωμόσωμα	[9 7 2 7 2 1 7 3 5 3 1 5 16 9 8 12 1 2 9 8 3 5 3 3 3 5 14 4]
Τιμή αντικειμενικής συνάρτησης	-2756'
Χρόνος ολοκλήρωσης κατεργασιών	2494'
Bonus των 300'	12
Bonus των 150'	11
Penalties των 2880'	0
Penalties των 480'	0

Η τελευταία τιμή του βέλτιστου χρωμοσώματος (τιμή 4) αφορά στην εβδομάδα μέσα στην οποία θα αρχίσει η παραγωγή. Ο αλγόριθμος επέλεξε τυχαία την τιμή αυτή, ενώ από τη στιγμή που όλες οι μηχανές έχουν τον ίδιο χαμηλό φόρτο εργασίας, δε θα υπήρχε κάποια ουσιαστική διαφορά αν επέλεγε κάποια άλλη.

Στην πραγματικότητα η βέλτιστη αυτή τιμή προέκυψε αφού τρέξαμε τον αλγόριθμο αρκετές φορές. Ο αλγόριθμος, για το πρόβλημα αυτό, δε καταλήγει πάντα στην ίδια βέλτιστη τιμή, αλλά σε τιμές που κυμαίνονται από -1500 έως και -2756. Μία τιμή όπως η -1500 δεν θα πρέπει να σημαίνει την αποτυχία της βελτιστοποίησης. Ο χρονοπρογραμματισμός είναι και εδώ ικανοποιητικός με διαφορά που ισοδυναμεί περίπου με 4 bonuses λιγότερα (4 x 300= 1200).

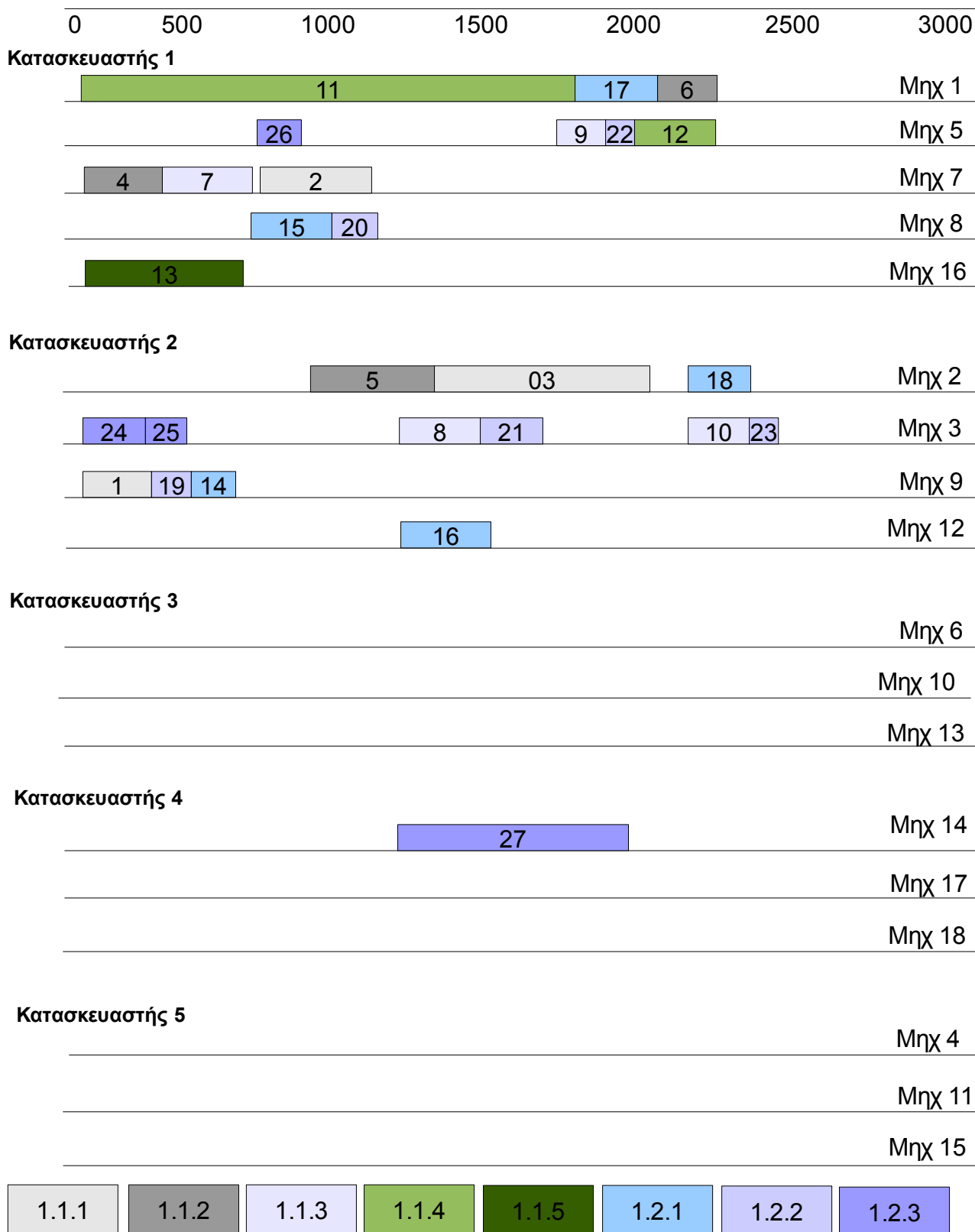
Το βέλτιστο χρωμόσωμα αντιστοιχεί στον ακόλουθο χρονικό προγραμματισμό (όλοι οι χρόνοι αντιστοιχούν σε λεπτά):

Κατεργασία	Μηχανή	Καθυστέρηση λόγω κατειλημμένης μηχανής	Στιγμή έναρξη κατεργασίας	Στιγμή λήξης κατεργασίας	Καθυστέρηση πριν την μεταφορά στον επόμενο κατασκευαστή	
Τεμάχιο 1.1.1	01	9	0	60	378	296
	02	7	0	794	1159	0
	03	2	106	1385	2010	0
Τεμάχιο 1.1.2	04	7	0	60	425	365
	05	2	0	910	1385	268
	06	1	277	2050	2215	0
Τεμάχιο 1.1.3	07	7	365	425	790	369
	08	3	0	1279	1496	157
	09	5	0	1773	1928	122
	10	3	0	2170	2337	0
Τεμάχιο 1.1.4	11	1	0	60	1875	0
	12	5	118	1993	2298	0
Τεμάχιο 1.1.5	13	16	0	60	780	0
Τεμάχιο 1.2.1	14	9	496	556	674	0
	15	8	0	794	1034	125
	16	12	0	1279	1519	134
	17	1	102	1875	2050	0
	18	2	0	2170	2355	0
Τεμάχιο 1.2.2	19	9	318	378	556	118
	20	8	240	1034	1154	5
	21	3	217	1496	1653	0
	22	5	155	1928	1993	57
	23	3	167	2337	2434	0
Τεμάχιο 1.2.3	24	3	0	60	357	0
	25	3	0	357	494	180
	26	5	0	794	879	0
	27	14	0	1239	2029	0

Παρατηρούμε πως 26 από τις 27 κατεργασίες αναλαμβάνονται από μηχανές των δύο πρώτων κατασκευαστών. Πρόκειται για τους δύο μεγαλύτερους κατασκευαστές (έχουν μαζί τις 9 από τις 18 συνολικά μηχανές), βρίσκονται στην μικρότερη απόσταση από την αποθήκη σε σύγκριση με τους υπόλοιπους, αλλά και σε μικρή απόσταση μεταξύ τους. Υπό αυτό το πρίσμα θεωρείται λογική αυτή η κατανομή κατεργασιών από τον αλγόριθμο βελτιστοποίησης. Όλα αυτά φαίνονται εντονότερα σε μία γραφική αναπαράσταση του διαγράμματος Gantt.

Διάγραμμα Gantt των μηχανών

Αξιοποιώντας της πληροφορίες για τους χρόνους μπορούμε να κατασκευάσουμε, γραφικά πλέον, το διάγραμμα Gantt.



1^{ος} συνδυασμός φασεολογιών (μηχανές με υψηλό φόρτο εργασίας)

Παρατηρούμε ότι οι μηχανές του κατασκευαστή 2 αναλαμβάνουν πολλές κατεργασίες (13 κατεργασίες από τις 27). Μάλιστα η μηχανή 3 αναλαμβάνει τις περισσότερες κατεργασίες από κάθε άλλη μηχανή (σύνολο 6).

Θα αυξήσουμε το φόρτο εργασίας των μηχανών του κατασκευαστή αυτού στο 95% για τις 3 πρώτες εβδομάδες και τις 2 τελευταίες. Στην 4 εβδομάδα θα θεωρήσουμε μέτριο φόρτο εργασίας 70% που είναι και πάλι υψηλότερος από αυτούς των μηχανών στους άλλους κατασκευαστές.

Στο αρχείο `fdatabase.m` ο πίνακας `workload` παίρνει πλέον την μορφή:

```
%pinakas fortou ergasias opou rows->weeks cols->mihanes (duration x mac)
WrkLoa=[0.4 0.95 0.95 0.4 0.4 0.4 0.4 0.4 0.95 0.4 0.4 0.95 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.95 0.95 0.4 0.4 0.4 0.4 0.4 0.95 0.4 0.4 0.95 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.95 0.95 0.4 0.4 0.4 0.4 0.4 0.95 0.4 0.4 0.95 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.70 0.70 0.4 0.4 0.4 0.4 0.4 0.70 0.4 0.4 0.70 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.95 0.95 0.4 0.4 0.4 0.4 0.4 0.95 0.4 0.4 0.95 0.4 0.4 0.4 0.4 0.4 0.4;
        0.4 0.95 0.95 0.4 0.4 0.4 0.4 0.4 0.95 0.4 0.4 0.95 0.4 0.4 0.4 0.4 0.4 0.4];
```

Η βελτιστοποίηση του συστήματος θα δείξει αν οι κατεργασίες θα μοιραστούν σε άλλες μηχανές ή αν η παραγωγή θα διατηρηθεί στις μηχανές του κατασκευαστή 2. Η εβδομάδα έναρξης των κατεργασιών έχει πλέον σημασία.

Βέλτιστο αποτέλεσμα:

Χρωμόσωμα	[7 11 1 7 11 5 7 1 6 4 4 5 16 11 7 13 4 5 11 8 4 4 5 1 1 5 15 4]
Τιμή αντικειμενικής συνάρτησης	-1129'
Χρόνος ολοκλήρωσης κατεργασιών	3521
Bonus των 300'	10
Bonus των 150'	11
Penalties των 2880'	0
Penalties των 480'	0

Παρατηρούμε ότι ο αλγόριθμος δεν επέλεξε καμία μηχανή του κατασκευαστή 2 (μηχ 9, 2, 3, 12) οπότε και οι ποινές είναι μηδενικές. Επέλεξε να μεταφέρει την παραγωγή σε άλλους κατασκευαστές με κόστος στη διάρκεια ολοκλήρωσης των κατεργασιών (3521' σε σύγκριση με το 2494' του προηγούμενου προβλήματος) αλλά και μεγάλο κέρδος από συγκεντρωτικές μεταφορές τεμαχίων.

Ο κώδικας, σε διαδοχικές επαναλήψεις του, κατέληγε πάντα σε χρωμοσώματα με εβδομάδα έναρξης των κατεργασιών την 4^η. Είναι λογικό αν αναλογιστούμε ότι συνολικά οι μηχανές στην εβδομάδα αυτή έχουν τον μικρότερο φόρτο εργασίας (όλες 40% και αυτές του 2ου κατασκευαστή 70%).

Χρονικός προγραμματισμός:

Κατεργασία	Μηχανή	Καθυστέρηση λόγω κατειλημμένης μηχανής	Στιγμή έναρξη κατεργασίας	Στιγμή λήξης κατεργασίας	Καθυστέρηση πριν την μεταφορά στον επόμενο κατασκευαστή	
Τεμάχιο 1.1.1	01	7	730	790	1055	345
	02	11	0	1940	2250	236
	03	1	0	3026	3441	0
Τεμάχιο 1.1.2	04	7	0	60	425	630
	05	11	0	1595	1805	397
	06	5	0	2742	2847	0
Τεμάχιο 1.1.3	07	7	365	425	790	0
	08	1	0	790	1005	490
	09	6	0	1675	1841	0
	10	4	285	2486	2598	0
Τεμάχιο 1.1.4	11	4	0	540	2202	284
	12	5	0	3026	3331	0
Τεμάχιο 1.1.5	13	16	0	60	780	0
Τεμάχιο 1.2.1	14	11	130	670	740	0
	15	7	0	1280	1495	0
	16	13	0	1675	1830	11
	17	4	133	2334	2486	184
	18	5	121	3331	3396	0
Τεμάχιο 1.2.2	19	11	0	540	670	70
	20	8	0	1280	1400	0
	21	4	262	2202	2334	0
	22	4	264	2598	2670	0
	23	5	186	3396	3461	0
Τεμάχιο 1.2.3	24	1	0	60	355	0
	25	1	0	355	490	0
	26	5	0	490	575	480
	27	15	0	1595	2395	0

Ο αλγόριθμος για το πρόβλημα αυτό επέδειξε αρκετή σταθερότητα καθώς έδινε στις περισσότερες των περιπτώσεων χρωμοσώματα που άφηναν εκτός της **μηχανές του 2^{ου} κατασκευαστή** με τα αποτελέσματα της αντικειμενικής συνάρτησης να κυμαίνονται από -1000 έως -1129. Σε κάποιες από τις περιπτώσεις που συνέκλινε σε αποτέλεσμα με χρήση των **μηχανών του κατασκευαστή 2** η τιμή της αντικειμενικής συνάρτησης κυμαινόταν στο διάστημα -300 με -500.

Για παράδειγμα:

Χρωμόσωμα	[7 9 1 7 3 1 7 3 4 1 4 5 16 8 10 12 4 1 8 10 2 4 5 1 2 5 14 4]
Τιμή αντικειμενικής συνάρτησης	-365'
Χρόνος ολοκλήρωσης κατεργασιών	3595'
Bonus των 300'	12
Bonus των 150'	12
Penalties των 2880'	0
Penalties των 480'	3

Στο αποτέλεσμα αυτό γίνεται χρήση 6 μηχανών με φόρτο εργασίας 70% που ισοδυναμούν με μισή ποινή ο καθένας. Σωστά λοιπόν υπολογίστηκαν 3 ποινές.

2^{ος} συνδυασμός φασεολογιών

Παράμετροι βελτιστοποίησης:

Number of variables	24
Constraints - Lower bound	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Constraints - Upper bound	[18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 6]
Stopping Criteria	Stall generations 50
Population	300
Initial range	[7 7 1 16 12 7 1 1 1 1 1 16 16 14 7 7 1 1 1 1 1 1 14 1;11 11 4 18 13 11 4 6 6 4 6 18 18 15 11 11 4 6 6 4 4 6 15 6]
Creation function	Uniform
Scaling function	Rank
Selection function	Stochastic uniform
Elite count	12
Crossover function	0.85
Mutation function	Gaussian
Crossover function	Scattered

Βέλτιστο αποτέλεσμα:

Χρωμόσωμα	[9 7 3 16 12 7 3 5 2 1 3 18 16 14 9 8 3 5 2 3 2 5 14 5]
Τιμή αντικειμενικής συνάρτησης	-1633'
Χρόνος ολοκλήρωσης κατεργασιών	3017
Bonus των 300'	10
Bonus των 150'	11
Penalties των 2880'	0
Penalties των 480'	0

Στο συγκεκριμένο πρόβλημα ο αλγόριθμος βελτιστοποίησης επέδειξε εξαιρετική σταθερότητα συγκλίνοντας τις περισσότερες φορές στο αποτέλεσμα αυτό.

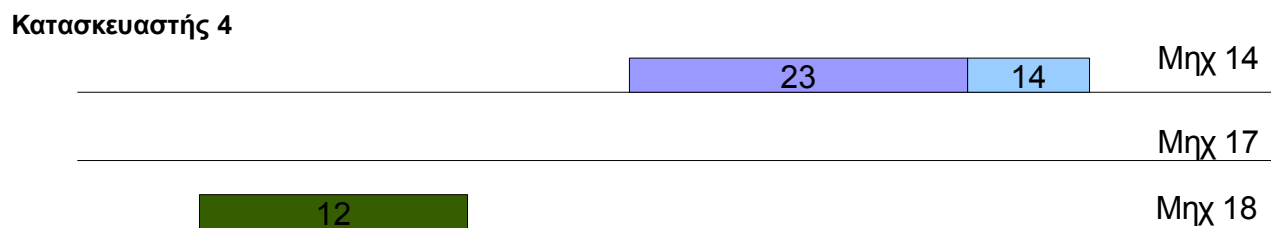
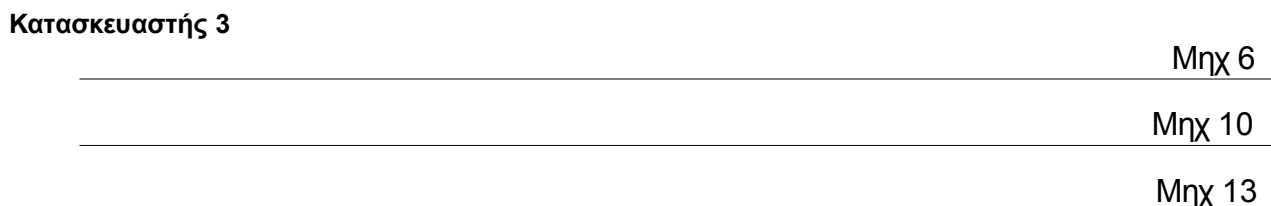
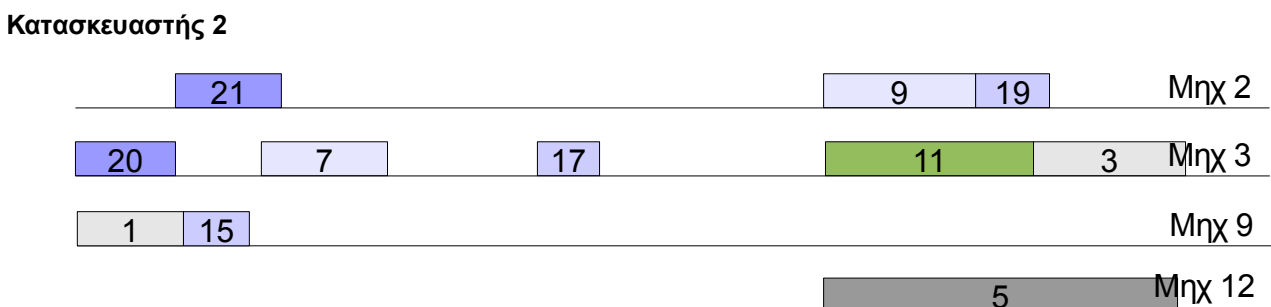
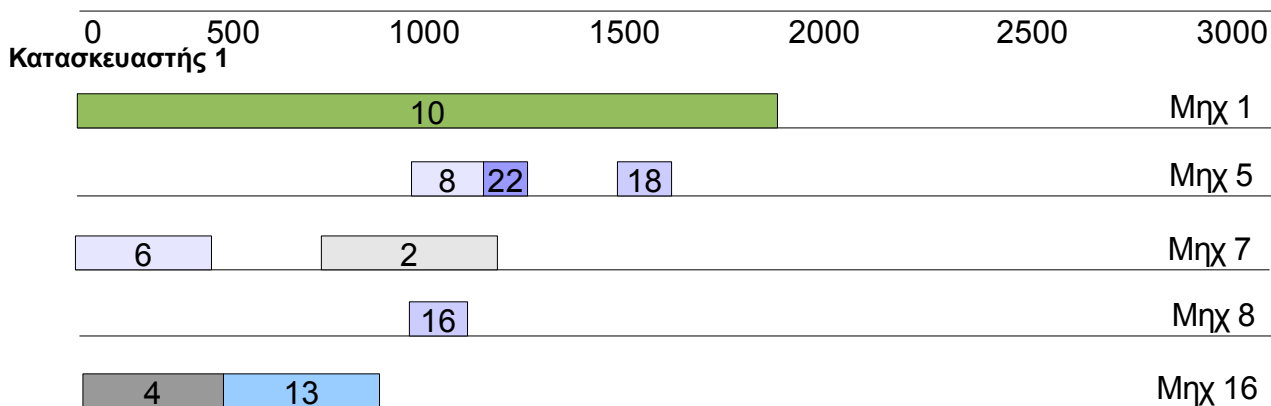
Το βέλτιστο χρωμόσωμα αντιστοιχεί στον ακόλουθο χρονικό προγραμματισμό (όλοι οι χρόνοι αντιστοιχούν σε λεπτά):

Κατεργασία	Μηχανή	Καθυστέρηση λόγω κατειλημμένης μηχανής	Στιγμή έναρξη κατεργασίας	Στιγμή λήξης κατεργασίας	Καθυστέρηση πριν την μεταφορά στον επόμενο κατασκευαστή	
Τεμάχιο 1.1.1	01	9	0	60	378	284
	02	7	0	782	1147	728
	03	3	467	2462	2879	0
Τεμάχιο 1.1.2	04	16	0	60	480	1395
	05	12	0	1995	2815	0
Τεμάχιο 1.1.3	06	7	0	60	425	55
	07	3	0	600	817	0
	08	5	0	937	1092	783
	09	2	0	1995	2370	0
Τεμάχιο 1.1.4	10	1	0	60	1875	0
	11	3	0	1995	2462	0
Τεμάχιο 1.1.5	12	18	0	420	1100	0
Τεμάχιο 1.2.1	13	16	420	480	820	357
	14	14	790	2327	2597	0
Τεμάχιο 1.2.2	15	9	318	378	556	261
	16	8	0	937	1057	90
	17	3	0	1267	1424	0
	18	5	0	1544	1609	266
	19	2	375	2370	2555	0
Τεμάχιο 1.2.3	20	3	0	60	357	0
	21	2	0	357	662	155
	22	5	155	1092	1177	0
	23	14	0	1537	2327	0

Και στο πρόβλημα αυτό ο αλγόριθμος ανέθεσε τις περισσότερες κατεργασίες (20/23) στους δύο πρώτους κατασκευαστές. Η κατεργασία 12 (χύτευση) ανατέθηκε στη μηχανή 18 του 4^{ου} κατασκευαστή επειδή την ίδια στιγμή η αντίστοιχη μηχανή του 1^{ου} είναι κατειλημμένη. Οι κατεργασίες 14 και 23 (EDM) δόθηκαν στις μηχανές του κατασκευαστή 4 αντί στις αντίστοιχες του κατασκευαστή 5, μάλλον λόγω της μικρότερης απόστασης του πρώτου από την αποθήκη σε σύγκριση με τον δεύτερο.

Συγκριτικά με το προηγούμενο πρόβλημα, σε αυτό έχουμε 2 bonus λιγότερα και αρκετά μεγαλύτερο χρόνο ολοκλήρωσης των κατεργασιών (3017' > 2494'). Θα προτιμήσουμε επομένως τον 1^ο συνδυασμό φασεολογιών.

Διάγραμμα Gantt των μηχανών



5.5 ΠΕΙΡΑΜΑΤΙΣΜΟΙ ΜΕ ΠΑΡΑΜΕΤΡΟΥΣ ΤΩΝ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Πείραμα 1^ο

Υποθέτουμε ότι ο ορισμός ενός αρχικού πεδίου αριθμών μέσα στο οποίο θα κατασκευαστεί ο αρχικός πληθυσμός δρα ευεργετικά στη σύγκλιση του αλγορίθμου σε μια καλή λύση, σε αντίθεση με την περίπτωση που ο αρχικός πληθυσμός κατασκευάζεται με τιμές από ολόκληρο το πεδίο ορισμού (π.ο 1-18).

Στην πρώτη περίπτωση είναι:

	Κατ 1	Κατ 2	Κατ 3	Κατ 4	Κατ 5	Κατ 6	Κατ 7	Κατ 8	Κατ 9
από	7	7	1	7	1	1	7	1	1
έως	11	11	4	11	4	6	11	4	6

	Κατ 10	Κατ 11	Κατ 12	Κατ 13	Κατ 14	Κατ 15	Κατ 16	Κατ 17	Κατ 18
από	1	1	1	16	7	7	12	1	1
έως	6	4	6	18	11	11	13	4	6

	Κατ 19	Κατ 20	Κατ 21	Κατ 22	Κατ 23	Κατ 24	Κατ 25	Κατ 26	Κατ 27
από	7	7	1	1	1	1	1	1	14
έως	11	11	4	6	6	4	4	6	15

Για παράδειγμα, αυτό σημαίνει ότι στην περίπτωση μιας τόννευσης και κατά την κατασκευή του αρχικού πληθυσμού δοκιμάζονται λύσεις μόνο μεταξύ των διαθέσιμων τόννων του συστήματος (διάστημα 7-11). Φυσικά έτσι μένει εκτός η μηχανή 4, η οποία είναι και αυτή ικανή για τόννευση, σε κάθε όμως περίπτωση η λύση αυτή είναι πιθανό να προκύψει και να δοκιμαστεί σε μία επόμενη γενιά.

Λύνουμε το πρόβλημα του 1^{ου} συνδυασμού φασεολογιών στις ακόλουθες συνθήκες:

Stopping criteria	Stall generations= 30
Number of variables	28
Constraints - Lower bound	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Constraints - Upper bound	[18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 6]

Creation function	Uniform
Scaling function	Rank
Selection function	Stochastic uniform
Elite count	8
Crossover function	0.85
Mutation function	Gaussian
Crossover function	Scattered

Αποτελέσματα τιμών αντικειμενικής συνάρτησης (without initial range)

Population	Run 1	Run 2	Run 3
200	4485	826	7009
300	-583	2999	-482
600	-205	2745	864

Αποτελέσματα τιμών αντικειμενικής συνάρτησης (with initial range)

Population	Run 1	Run 2	Run 3
200	-1755	-2421	-2000
300	-2000	-2156	-1675
600	-2450	-2708	-2613

7 στα 9 αποτελέσματα της δεύτερης περίπτωσης είναι καλές τιμές μικρότερες του -2000 σε αντίθεση με την πρώτη περίπτωση όπου μόνο 3 στις 9 τιμές είναι αρνητικές και μάλιστα μεγαλύτερες του -1000.

Λύνουμε ξανά το πρόβλημα με αλλαγή στο κριτήριο τερματισμού του αλγορίθμου:

Stopping criteria	Time limit= 60"
--------------------------	------------------------

Αποτελέσματα τιμών αντικειμενικής συνάρτησης (without initial range)

Population	Run 1	Run 2	Run 3
200	916	3094	3286
300	901	3985	7124
600	1830	4164	2145

Αποτελέσματα τιμών αντικειμενικής συνάρτησης (with initial range)

Population	Run 1	Run 2	Run 3
200	-1975	-1993	-1820
300	-1856	-2718	-1926
600	-1689	-1853	-1985

Και εδώ φαίνεται πως τα αποτελέσματα στη δεύτερη περίπτωση είναι πολύ καλύτερα σε σύγκριση με αυτά της πρώτης.

Όλα αυτά αποτελούν ένδειξη ότι ο αλγόριθμος βελτιστοποίησης δίνει καλύτερα αποτελέσματα στη περίπτωση περιορισμού του πεδίου μέσα στο οποίο θα κατασκευαστεί ο αρχικός πληθυσμός.

Πείραμα 2°

Θα πειραματιστούμε με διάφορα μεγέθη πληθυσμών και κριτήρια τερματισμού του αλγορίθμου.

Παράμετροι βελτιστοποίησης:

Number of variables	28
Constraints - Lower bound	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Constraints - Upper bound	[18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 6]
Initial range	[7 7 1 7 1 1 7 1 1 1 1 1 16 7 7 12 1 1 7 7 1 1 1 1 1 1 14 1; 11 11 4 11 4 6 11 4 6 6 4 6 18 11 11 13 4 6 11 11 4 6 6 4 4 6 15 6]
Creation function	Uniform
Scaling function	Rank
Selection function	Stochastic uniform
Elite count	2% του πληθυσμού
Crossover function	0.85
Mutation function	Gaussian
Crossover function	Scattered

Κριτήριο τερματισμού τα 60"

Τελικές τιμές αντικειμενικής συνάρτησης (και πλήθος γενεών που έτρεξαν στο χρονικό διάστημα των 60"):

Population	Run 1	Run 2	Run 3	Run 4	Run 5	Average	Good results under -2000
200	-2156 (37)	-1587 (36)	-2055 (38)	-1758 (39)	-1693 (37)	-1850	2/5
300	-2300 (24)	-1493 (26)	-1856 (25)	-1655 (25)	-1993 (25)	-1860	1/5
600	-2250 (13)	-1901 (13)	-1884 (13)	-2300 (13)	-1503 (13)	-1968	2/5

Παρατηρούμε ότι αυξανόμενου του πληθυσμού ο μέσος όρος των αποτελεσμάτων βελτιώνεται. Για πληθυσμό 200 ολοκληρώθηκαν κατά μέσο όρο 37.4 γενιές (1.6" ανά γενιά), για πληθυσμό 300 ολοκληρώθηκαν 25 γενιές (2.4" ανά γενιά) και για πληθυσμό 600 ολοκληρώθηκαν 13 γενιές (4.6" ανά γενιά).

Κριτήριο τερματισμού η σταθεροποίηση της τιμής της αντικειμενικής συνάρτησης για 20 γενιές

Τελικές τιμές αντικειμενικής συνάρτησης:

Population	Run 1	Run 2	Run 3	Run 4	Run 5	Average	Good results under -2000
200	-1854	-2718	-1591	-1810	-2422	-2079	2/5
300	-1771	-2712	-1900	-1614	-2275	-2054	2/5
600	-2055	-2645	-2645	-2300	-2300	-2389	5/5

Ο αλγόριθμος για πληθυσμούς 200 και 300 ατόμων έδωσε περίπου τα ίδια αποτελέσματα όσον αφορά τον μέσο όρο και τις καλές τιμές (μικρότερες του -2000) ενώ για πληθυσμό 600 ατόμων τα αποτελέσματα είναι εμφανώς καλύτερα.

Κριτήριο τερματισμού η ολοκλήρωση της 35^{ης} γενιάς

Τελικές τιμές αντικειμενικής συνάρτησης:

Population	Run 1	Run 2	Run 3	Run 4	Run 5	Average	Good results under -2000
200	-1543	-1975	-1543	-2122	1550	-1747	1/5
300	-1675	-1991	-1901	-2102	1951	-1924	1/5
600	-2300	-2456	-2055	-2538	2500	-2370	5/5

Και εδώ τα αποτελέσματα βελτιώνονται με την αύξηση του πληθυσμού.

Συμπεράσματα:

1. Όσο μεγαλύτερος ο πληθυσμός τόσο καλύτερα τα αποτελέσματα για όλα τα κριτήρια τερματισμού του αλγορίθμου. Κερδίζουμε σε ποιότητα αποτελεσμάτων χάνουμε όμως σε ταχύτητα καθώς ο χρόνος για την ολοκλήρωση κάθε γενιάς αυξάνει.
2. Καλύτερα αποτελέσματα έχουμε όταν κριτήριο τερματισμού είναι η σταθεροποίηση της τιμής της αντικειμενικής συνάρτησης μετά από κάποιον αριθμό γενεών. Αποτελεί ένδειξη της αξιοπιστίας του κριτηρίου αυτού.

Πείραμα 3^ο

Υποθέτουμε πως μεταβάλλοντας τη σύνθεση του πληθυσμού που προκύπτει από μια αναπαραγωγική διαδικασία θα καταλήξουμε στο συνδυασμό εκείνο που δίνει τα καλύτερα αποτελέσματα.

Παράμετροι βελτιστοποίησης:

Number of variables	28
Constraints - Lower bound	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Constraints - Upper bound	[18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 6]
Stopping Criteria	Stall generations= 30
Population	300
Initial range	[7 7 1 7 1 1 7 1 1 1 1 1 16 7 7 12 1 1 7 7 1 1 1 1 1 1 14 1; 11 11 4 11 4 6 11 4 6 6 4 6 18 11 11 13 4 6 11 11 4 6 6 4 4 6 15 6]
Creation function	Uniform
Scaling function	Rank
Selection function	Stochastic uniform
Mutation function	Gaussian
Crossover function	Scattered

Elite count= 8

Αποτελέσματα τιμών αντικειμενικής συνάρτησης:

Crossover fraction	Run 1	Run 2	Run 3	Average	Good results under -2000
0.55	-1711	-1393	-1424	-1509	0/3
0.65	-1296	-2250	-1544	-1697	1/3
0.75	-1886	-2090	-2156	-2044	2/3
0.85	-1942	-2645	-2613	-2400	2/3
0.95	-2661	-2645	-1833	-2380	2/3

Τα καλύτερα αποτελέσματα ήρθαν για την τιμή 0.85. Καλά αποτελέσματα έδωσαν και οι υψηλές τιμές 0.75 – 0.95 του crossover fraction σε αντίθεση με τις χαμηλές τιμές 0.55 – 0.65. Αυτό αποτελεί ένδειξη ότι χαμηλά ποσοστά μετάλλαξης ίσως βελτιώνουν την απόδοση του αλγόριθμου.

Θα θεωρήσουμε μία υψηλή τιμή του crossover fraction (0.90) για να συνεχίσουμε.

Crossover fraction= 0.90

Αποτελέσματα τιμών αντικειμενικής συνάρτησης:

Elite count	Run 1	Run 2	Run 3	Average	Good results under -2000
2	-2156	-1941	-2213	-2103	2/3
4	-2122	-2035	-2150	-2102	3/3
8	-1480	-2661	-1693	-1945	1/3
10	-1993	-2290	-1759	-2014	1/3
16	-2712	-2712	-2166	-2530	3/3

Ένας πληθυσμός 300 ατόμων με σύνθεση 16 ατόμων προϊόντων ελιτισμού, 256 ατόμων προερχόμενων από διασταύρωση και 28 ατόμων από μετάλλαξη είναι πιθανό να δίνει τα καλύτερα αποτελέσματα σε συνεχόμενες επαναλήψεις του αλγόριθμου.

Με παρόμοιες δοκιμές μπορούμε να καταλήγουμε στις βέλτιστες παραμέτρους elite count και crossover fraction για οποιονδήποτε πληθυσμό.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα πλαίσια της διπλωματικής εργασίας δείξαμε ότι οι γενετικοί αλγόριθμοι αποτελούν ένα πολύ καλό εργαλείο για τη βελτιστοποίηση της σύνθεσης συστημάτων κατεργασιών σε προβλήματα με εναλλακτικά φασεολόγια. Ο συμβολισμός μηχανών με ακεραίους αριθμούς και η αντιστοίχιση των γονιδίων του χρωμοσώματος στις κατεργασίες των φασεολογίων είναι η κωδικοποίηση που προτείνουμε.

Αντιμετωπίσαμε το δύσκολο πρόβλημα υπολογισμού του χρόνου ολοκλήρωσης των κατεργασιών και των καθυστερήσεων των μερίδων δημιουργώντας διάγραμμα Gantt των μηχανών.

Προσεγγίσαμε τη δομή που θα πρέπει να έχει μία βάση δεδομένων για την υποστήριξη ενός τέτοιου συστήματος βελτιστοποίησης. Παράλληλα, αναπτύξαμε μια μεθοδολογία ανταλλαγής δεδομένων μεταξύ της βάσης δεδομένων και του κώδικα βελτιστοποίησης.

Πειραματιστήκαμε με παραμέτρους των γενετικών αλγορίθμων. Καταλήξαμε στην χρησιμότητα της ομαδοποίησης των μηχανών κατά είδος και τη δημιουργία ενός πεδίου αριθμών, μικρότερου του πεδίου ορισμού, όπου θα περιορίζεται η παραγωγή του αρχικού πληθυσμού. Σε αυτή την περίπτωση η βελτιστοποίηση είναι ταχύτερη και συγκλίνει σε καλύτερα αποτελέσματα σε διαδοχικές επαναλήψεις της.

Μια μελλοντική επέκταση της εργασίας ίσως θα πρέπει να αφορά στην κοστολογική, πέρα από τη χρονική, βελτιστοποίηση τέτοιων συστημάτων.

Σημαντική είναι η δημιουργία κώδικα με SQL προγραμματισμό που θα αυτοματοποιεί τη διαδικασία μεταφοράς των πληροφοριών από τη βάση δεδομένων στο σύστημα βελτιστοποίησης. Περαιτέρω έρευνα χρειάζεται και για τις συναρτήσεις – παραμέτρους των γενετικών αλγορίθμων που επιλύουν καλύτερα τα προβλήματα που μελετάμε.

Η βάση δεδομένων και ο κώδικας βελτιστοποίησης που προτείνουμε μπορούν να χρησιμοποιηθούν ως μέρος ενός ευρύτερου συστήματος e-manufacturing, το οποίο θα περιλαμβάνει διαδικτυακό interface και επιπλέον στοιχεία Cad online, features recognition, process attribution, time evaluation και cost evaluation.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Rajamani, D., Singh, N., Aneja, Y.P., Integrated design of cellular manufacturing systems in the presence of alternative process plans, Int Jnl Production Research, Aug 1990, 28 (8), 1541-1554
2. Wu, T.-H., Chen, J.-F., Yeh, J.-Y., A decomposition approach to the cell formation problem with alternative process plans, Int Jnl Advanced Manufacturing Technology , Dec 2004, 24 (11-12), 834-840
3. Logendran, R., Ramakrishna, P., Srisankarajah, C., Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans, Int Jnl Production Research, Feb 1994, 32 (2), 273-297
4. Hwang, H., Ree, P., Routes selection for the cell formation problem with alternative part process plans, Computers & Industrial Engineering, 3 1996, 30 (3),423-431
5. Saygin, C., Kilie, S.E., Integrating flexible process plans with scheduling in flexible manufacturing systems, Int Jnl Advanced Manufacturing Technology, Jul 1999, 15 (4), 268-180
6. Ferreira, J.C.E., Wysk, R.A., An investigation of the influence of alternative process plans on equipment control, Jnl Manufacturing Systems, 2000, 19 (6), 393-406
7. Rajamani, D.R., Singh, N., Aneja, Y.P., Design of cellular manufacturing systems, Int Jnl Production Research, 7 1996, 34 (7), 1917-1928
8. Sofianopoulou, S., Manufacturing cells design with alternative process plans and/or replicate machines, Int Jnl Production Research, Feb 1999, 37 (3), 707-720
9. Candido, M.A.B., Khator, S.K., Barcia, P.M., A genetic algorithm based procedure for more realistic job shop scheduling problems, Int Jnl Production Research, Dec 1998, 36 (12), 3437-3457

10. Rossi, A., Dini, G., Dynamic scheduling of FMS using a real-time genetic algorithm, *Int Jnl Production Research*, Jan 2000, 38 (1), 1-20
11. Giovanni Perrone, *Manufacturing e-marketplaces: Innovative tools for the extended enterprise. Supporting manufacturing enterprise networks, Designing and evaluating value added services in manufacturing e-marketplaces*, Springer, 2005, Chapter 1, 1-21
12. Giovanni Perrone, Giovanni Montana, *An agent based architecture for manufacturing e-marketplaces – The agent based architecture*, *Designing and evaluating value added services in manufacturing e-marketplaces*, Springer, 2005, Chapter 2, 23-43
13. Lanfranco Imperti, Tulio Tolio, *Manufacturability models for manufacturing e-marketplaces. The Manufacturing Planner agent*, *Designing and evaluating value added services in manufacturing e-marketplaces*, Springer, 2005, Chapter 4, 67-95
14. Σημειώσεις μαθήματος Υπολογιστικής Νοημοσύνης II, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Πατρών
15. Mathworks Matlab Genetic Algorithm and Direct Search Toolbox User's Guide
16. Steven Roman, *Access Database Design and Programming*, O'Reilly, 3rd edition

ΠΑΡΑΡΤΗΜΑ

Στο cdrom που συνοδεύει την εργασία περιέχονται τα αρχεία της Matlab με τον κώδικα βελτιστοποίησης και τη βάση δεδομένων για τα προβλήματα που παρουσιάστηκαν στο 5^ο κεφάλαιο.

Συγκεκριμένα, στο φάκελο version 1 περιλαμβάνονται όλες οι απαραίτητες συναρτήσεις ([fantik02](#), [fdialogi](#), [fgentime](#), [fratime](#)) για την επίλυση προβλημάτων με το Genetic Algorithm Toolbox της Matlab.

Το αρχείο [fdatabase_1](#) αντιστοιχεί στο πρόβλημα 1^ο συνδυασμού φασεολογίων, το [fdatabase_1b](#) στο πρόβλημα του αυξημένου φόρτου εργασίας και το [fdatabase_2](#) στο πρόβλημα του 2^ο συνδυασμού φασεολογίων. Αρκεί να μετονομαστεί το αρχείο [fdatabase_*](#) σε [fdatabase](#).

Φυσικά ο χρήστης μπορεί να δημιουργήσει τα δικά του προβλήματα, να τα μεταφέρει σε ένα αρχείο με το όνομα [fdatabase](#) και να τα επιλύσει χρησιμοποιώντας τις ίδιες συναρτήσεις βελτιστοποίησης.

Στον φάκελο version 2 περιέχεται η αντικειμενική συνάρτηση τροποποιημένη ([fantik01](#)) ώστε η βελτιστοποίηση να τρέχει εκτός του περιβάλλοντος Genetic Algorithm Toolbox. Περιλαμβάνονται οι απαραίτητες συναρτήσεις δημιουργίας αρχικού πληθυσμού, αξιολόγησης λύσεων, επιλογής γονέων, διασταύρωσης και μετάλλαξης. Τα προβλήματα επιλύονται τρέχοντας την κεντρική συνάρτηση [fmain](#).

