



# ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

## *Διπλωματική Εργασία*

---

### **Προσομοίωση γραμμής παραγωγής αναψυκτικών κατά παρτίδες**

---

**Λουλβίλα Τσούνι**

**ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ**

*Βασίλειος Κουϊκόγλου: Επιβλέπων*

*Ευστράτιος Ιωαννίδης*

*Γεώργιος Τσιναράκης*

**Χανιά, Νοέμβριος 2022**

Copyright υπό Λουβίλα Τσούνι

*Η παρούσα διπλωματική εργασία αφιερώνεται στα παιδιά μου,*

*Γιούργκεν και Μαρκελιάν.*

*Την αφιερώνω στον εαυτό μου που μετά από 30 χρόνια*

*κατάφερα να πραγματοποιήσω το όνειρό μου.*

*Σε όλους τους φίλους που ήταν δίπλα μου!!!!.*

## ΠΕΡΙΛΗΨΗ

Στη παρούσα διπλωματική εργασία αναπτύσσεται ένα μοντέλο προσομοίωσης της γραμμής παραγωγής αναψυκτικών της εταιρίας «ΓΕΡΑΝΙ Α.Ε.». Στο συγκεκριμένο σύστημα, η παραγωγή γίνεται με βάση τις παραγγελίες αναψυκτικών διαφόρων τύπων και περιλαμβάνει την κυκλοφορία υγρών υλών (νερό, χυμοί, διαλύματα) και διακριτών κομματιών (φιάλες, συσκευασίες) και την παραγωγή κατά παρτίδες ανάλογα με τα είδη αναψυκτικών που έχει κάθε παραγγελία. Μελετώνται τα στάδια της παραγωγής και συσκευασίας από απόψεως κυκλοφορίας, αποθήκευσης και επεξεργασίας υλικών. Αναπτύσσονται μοντέλα προσομοίωσης με χρήση του ειδικού λογισμικού SIMIO και με πρόγραμμα στη Matlab. Διαπιστώθηκε ότι το δεύτερο προσφέρει στον χρήστη μεγαλύτερη ευκολία για την περιγραφή της παραγωγής κατά παρτίδες. Το πρόγραμμα προσομοίωσης χρησιμοποιείται στην εκτίμηση σημαντικών μέτρων απόδοσης του συστήματος, όπως χρόνοι υλοποίησης διάφορων προγραμμάτων παραγωγής, καθυστερήσεις διεκπεραίωσης παραγγελιών και το μέσο απόθεμα κάθε προϊόντος στην αποθήκη. Δοκιμάζοντας διάφορες τιμές παραμέτρων για μία ευρέως χρησιμοποιούμενη πολιτική παραγωγής παρτίδων προσδιορίζεται ο συνδυασμός εκείνος που ελαχιστοποιεί ένα συνολικό μέτρο κόστους του συστήματος. Το μοντέλο θα μπορούσε να αποτελέσει βάση για την ανάπτυξη σχεδίων βελτίωσης της παραγωγής, τον σχεδιασμό μίας νέας γραμμής παραγωγής ή την επέκταση της υπάρχουσας.

## **ABSTRACT**

This thesis develops a simulation model of the soft drinks production line of "Gerani S.A." company. In this system, the production is controlled based on customer orders involving soft drinks of different types, and it includes liquids (water, juices, and solutions) discrete items (bottles, packages) and production in batches (lots) depending on the types of soft drinks ordered. The production and packaging stages are studied from the point of view of circulation, storage, and material processing. Simulation models are developed using the specific simulation software SIMIO and by programming in MATLAB. It was found that the latter can describe lot sizing policies with greater ease. The simulation program is used to estimate important performance measures of the production system, such as the completion times of various production schedules, delays in handling customer orders, and the average stock of each product. By testing various values for the parameters of a commonly used lot scheduling policy the combination that minimizes a given cost criterion is found. The model could form the basis for developing production improvement plans, designing a new production line or expanding the existing

## ΕΥΧΑΡΙΣΤΙΕΣ

Τελειώνοντας τις σπουδές μου με το τελευταίο κομμάτι των υποχρεώσεων την διπλωματική εργασία θα ήθελα να ευχαριστήσω κάποιους ανθρώπους που βρέθηκαν στη ζωή μου και πραγματικά νιώθω πολύ τυχερή και ότι για κάποιο λόγο γίνονται όλα.

Η παρούσα διπλωματική εργασία υλοποιήθηκε με την ύπαρξη αυτών των ανθρώπων πάντα δίπλα μου, ενθαρρύνοντας και βοηθώντας με σε κάθε βήμα. Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή και επιβλέποντα της εργασίας, τον κ. Κουϊκόγλου Βασίλειο για την πολύτιμη βοήθεια και καθοδήγηση καθ' όλη την διάρκεια της εκπόνησής της εργασίας. Ήταν πάντα εκεί ανεξαρτήτως ωρών , διακοπών ή αργιών.

Ένα μεγάλο ευχαριστώ οφείλω στον σύζυγο, φίλο και πατέρα των παιδιών μου τον κ. Άρμπεν (Arben) που έδειξε υπομονή και κατανόηση όχι μόνο κατά την διάρκεια της εργασίας μου αλλά όλη την πενταετή φοίτηση μου. Η ενθάρρυνση και περηφάνεια των παιδιών μου Γιούργκεν (Jurgen) και Μαρκελιάν (Markelian) ήταν μια μεγάλη ευτυχία και χαρά για μένα. Χωρίς την στήριξη τους δεν θα μπορούσα να πραγματοποιήσω το όνειρό μου.

Δεν μπορώ να παραλείψω να ευχαριστήσω και κάποιους πολύ σημαντικούς, ανθρώπους που βρεθήκαν στο δρόμο μου κατά την διάρκεια των σπουδών μου. Συγκεκριμένα, την Δαράκη Μαρία-Στέλλα που ήταν πάντα έτοιμη για εμένα σε ό,τι χρειαζόμουν, την Καμπέρη Αγγελική με την φιλοξενία της και πρόθυμη να βρει χρόνο και χώρο για μας, τον Καπιδάκη Σαράντο ο πρώτος φίλος και αγαπημένος μου συμφοιτητής, τον Χατζηδάκη Αντρέα με την επιθυμία του να εξηγεί οτιδήποτε του ζητήσεις, και τέλος ένα εξαιρετικό παιδί των Αρώνη Στυλιανό που μοίραζε τις γνώσεις του απλόχερα. Είμαι πολύ τυχερή και νιώθω ευγνωμοσύνη που γνώρισα αυτούς τους υπέροχους φίλους και συναδέλφους πλέον.

Κλείνοντας θα ήθελα να ευχαριστήσω τον υπεύθυνο της εταιρίας Γεράνι τον κ. Μανίτη Ευάγγελο που με την πολύτιμη βοήθεια του αξιοποιήσαμε τα δεδομένα για την ανάλυση και διερεύνηση εργασίας.

# ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ .....	3
ABSTRACT.....	4
ΕΥΧΑΡΙΣΤΙΕΣ .....	5
ΠΕΡΙΕΧΟΜΕΝΑ.....	6
1 ΕΙΣΑΓΩΓΗ .....	8
1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ .....	8
1.2 ΔΟΜΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ.....	9
2 Η ΕΠΙΧΕΙΡΗΣΗ .....	10
2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΗΣ ΕΠΙΧΕΙΡΗΣΗΣ.....	10
2.2 ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ .....	11
3 ΜΟΝΤΕΛΑ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ.....	13
3.1 ΚΑΤΗΓΟΡΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ .....	13
3.2 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ ΜΕ ΜΟΝΤΕΛΑ ΟΥΡΩΝ ΑΝΑΜΟΝΗΣ .....	14
3.2.1 Σύστημα αναμονής M/M/1 .....	14
3.2.2 Σύστημα αναμονής M/M/1/K.....	14
3.2.3 Σύστημα αναμονής M/M/m/K.....	15
3.2.4 Επίλυση συστημάτων αναμονής.....	15
3.3 ΜΟΝΤΕΛΑ ΔΙΑΚΡΙΤΩΝ ΓΕΓΟΝΟΤΩΝ .....	16
3.4 ΣΥΣΤΗΜΑ ΠΑΡΑΓΩΓΗΣ ΚΑΤΑ ΠΑΡΤΙΔΕΣ (Lot sizing) .....	18
3.4.1 Μεγέθη παρτίδων για κυκλική παραγωγή με σταθερή ζήτηση.....	19
3.4.2 Εφαρμογές συστημάτων παραγωγής κατά παρτίδες .....	21
4 ΑΝΑΠΤΥΞΗ ΜΟΝΤΕΛΟΥ ΜΕ ΛΟΓΙΣΜΙΚΟ SIMIO .....	23
4.1 ΕΙΣΑΓΩΓΗ.....	23
4.2 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΤΟ ΛΟΓΙΜΙΚΟ SIMIO .....	23
4.3 ΔΙΑΔΙΚΑΣΙΑ ΕΙΣΑΓΩΓΗΣ ΠΑΡΑΜΕΤΡΩΝ .....	24
4.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ .....	24
5 ΜΟΝΤΕΛΟ ΡΟΗΣ ΜΕ ΧΡΗΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ MATLAB.....	25
5.1 ΕΙΣΑΓΩΓΗ.....	25
5.2 ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	25
5.2.1 Αναλυτική περιγραφή γεγονότων.....	26

5.2.2	Μεγέθη για την εξαγωγή συμπερασμάτων .....	27
6	ΥΠΟΛΟΓΙΣΜΟΙ ΠΑΡΑΜΕΤΡΩΝ .....	29
6.1	ΕΙΣΑΓΩΓΗ .....	29
6.2	ΑΡΧΙΚΗ ΕΚΤΙΜΗΣΗ ΠΑΡΑΜΕΤΡΩΝ .....	29
7	ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ .....	33
7.1	ΕΠΙΛΟΓΗ ΧΡΟΝΙΚΟΥ ΟΡΙΖΟΝΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ .....	33
7.2	ΒΕΛΤΙΣΤΕΣ ΠΑΡΑΜΕΤΡΟΙ ΕΛΕΓΧΟΥ ΑΠΟΘΕΜΑΤΩΝ .....	33
7.3	ΕΠΙΔΡΑΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΕΛΕΓΧΟΥ ΣΤΟΥΣ ΔΕΙΚΤΕΣ ΑΠΟΔΟΣΗΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....	36
7.3.1	Μέγιστο απόθεμα προϊόντος 1 .....	36
7.3.2	Στάθμη χαμηλού αποθέματος προϊόντος 1 .....	38
7.3.3	Μέγιστος αριθμός παραγγελιών σε προτεραιότητα (PRIORITY) .....	40
8	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	41
	ΒΙΒΛΙΟΓΡΑΦΙΑ .....	43
	ΠΑΡΑΡΤΗΜΑ 1 .....	44
	ΠΑΡΑΡΤΗΜΑ 2 .....	51



# 1 ΕΙΣΑΓΩΓΗ

Ένα σύστημα παραγωγής αποτελείται από τεχνολογικό εξοπλισμό (μηχανές, συστήματα μεταφοράς, αποθήκες) και ανθρώπινο δυναμικό (τεχνικό και επιστημονικό προσωπικό και στελέχη της διεύθυνσης) που επιτελούν ένα σύνολο εργασιών (επεξεργασίες, διαδικασίες) για την μετατροπή πρώτων υλών σε τελικά προϊόντα. Οι επιχειρήσεις που λειτουργούν τέτοια συστήματα ενδιαφέρονται για τους τρόπους βελτίωσης της παραγωγής, εξοικονόμησης πόρων και αύξησης της κερδοφορίας.

## 1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ

Σκοπός της διπλωματικής εργασίας είναι η μελέτη λειτουργίας της γραμμής παραγωγής και εμφιάλωσης εταιρείας αναψυκτικών Gerani Softdrinks S.A. Η μελέτη έγινε με την ανάπτυξη μοντέλων προσομοίωσης τα οποία:

- με βάση δεδομένα για τις λειτουργικές παραμέτρους του συστήματος όπως τα επίπεδα ζήτησης και αποθεμάτων και δυναμικότητες παραγωγής για κάθε προϊόν
- μιμούνται τη λειτουργία του συστήματος για έναν προκαθορισμένο χρονικό διάστημα (χρονικός ορίζοντας της προσομοίωσης)
- και εξάγουν δείκτες απόδοσης του συστήματος υπολογίζοντας τις μέσες τιμές ορισμένων μεταβλητών στο διάστημα αυτό όπως για παράδειγμα το μέσο απόθεμα προϊόντων και τη μέση καθυστέρηση διεκπεραίωσης παραγγελιών.

Η προσομοίωση χρησιμοποιείται για να μελετηθούν και να βελτιστοποιηθούν "συστήματα" ανθρώπων, εξοπλισμού, υλικών και διαδικασιών. Εκτός από αυτό, καθιστά εφικτή η αντιγραφή της συμπεριφοράς συστημάτων όπως εργοστάσια, αποθήκες, νοσοκομεία, τράπεζες, σούπερ μάρκετ, θεματικά πάρκα - σχεδόν οπουδήποτε παρέχεται μια υπηρεσία ή παράγεται ένα αντικείμενο. Επίσης αποτελεί ένα εργαλείο βελτίωσης της απόδοσης και λειτουργεί ως πειραματικό εργαστήριο, με τη διαφορά ότι δεν είναι φυσικό αλλά ένα εικονικό μοντέλο υπολογιστή.

Από τις παραγωγικές διαδικασίες της συγκεκριμένης επιχείρησης (προμήθεια υλών, παραγωγή, εμφιάλωση), η μελέτη εστιάστηκε στο τμήμα παραγωγής που παρουσιάζει τις μεγαλύτερες καθυστερήσεις. Καταγράφηκαν οι λειτουργικές παράμετροι του τμήματος (παραγωγή κατά παρτίδες, χρόνοι παραγωγής, χρόνοι προετοιμασίας) και αναπτύχθηκαν μοντέλα προσομοίωσης με το λογισμικό Simio και σε πρόγραμμα MATLAB. Το πρόγραμμα MATLAB προτιμήθηκε λόγω της ευκολίας στην περιγραφή της παραγωγής κατά παρτίδες.

## 1.2 ΔΟΜΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ

Η παρούσα Διπλωματική Εργασία αναφέρεται στη μοντελοποίηση και προσομοίωση της γραμμής παραγωγής και εμφιάλωσης αναψυκτικών της εταιρίας «ΓΕΡΑΝΙ Α.Ε.». Για τη παροχή των δεδομένων και την πρακτική εφαρμογή του μοντέλου υπήρξε συνεργασία με την Ελληνική εταιρία «ΓΕΡΑΝΙ Α.Ε.», η οποία διαθέτει παραγωγική μονάδα. Στο 1ο κεφάλαιο αναφέρεται το αντικείμενο της εργασίας και η ιστορική αναδρομή της επιχείρησης. Επιπλέον δίδονται κάποιες πληροφορίες για την εταιρία, περιγράφεται η παραγωγική διαδικασία αυτής, έπειτα διατυπώνεται και επιλύεται το πρόβλημα. Στο 2ο κεφάλαιο περιγράφονται τα είδη και τα μοντέλα συστημάτων παραγωγής, τα οποία θα είναι χρήσιμα για τη συγκεκριμένη εργασία. Επίσης παρουσιάζεται το βασικό μαθηματικό υπόβαθρο κάθε συστήματος. Τέλος το κεφάλαιο κλείνει με την ανασκόπηση ορισμένων εφαρμογών των αναφερόμενων συστημάτων σε βιομηχανίες αναψυκτικών. Στο 3ο κεφάλαιο, διερευνάται το υπό εξέταση μοντέλο το οποίο προσομοιώνεται σε λογισμικό SIMIO. Εκεί δίνονται μερικά βήματα και στιγμιότυπα κατά την εφαρμογή αριθμητικών πειραμάτων. Στο 4ο κεφάλαιο μοντελοποιείται το σύστημα, γράφοντας κώδικα σε περιβάλλον MATLAB. Ακόμη παρατίθενται και επεξηγούνται τα αποτελέσματα με τα συμπεράσματα που προκύπτουν από την πολιτική διαδικασία της παρούσας ανάλυσης. Κλείνοντας στο τελευταίο κεφάλαιο παρουσιάζονται κάποια σχόλια, προτάσεις χρήσης του εφαρμοζόμενου μοντέλου στην υπό εξέταση εταιρεία καθώς και μελλοντικές επεκτάσεις.

## 2 Η ΕΠΙΧΕΙΡΗΣΗ

### 2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΗΣ ΕΠΙΧΕΙΡΗΣΗΣ

Η εταιρεία παραγωγής και εμφιάλωσης αναψυκτικών «ΓΕΡΑΝΙ Α.Ε.» αποτελεί εξέλιξη της ομώνυμης οικογενειακής επιχείρησης που λειτουργεί εδώ και τρεις γενιές, από το 1928 μέχρι και σήμερα. Το πρώτο, σύγχρονο για την εποχή εργοστάσιο «Γεράνι», δημιουργήθηκε το 1966 στο κέντρο της πόλης των Χανίων, ενώ το 1984 οι εγκαταστάσεις μεταφέρθηκαν σε έναν ιδιόκτητο χώρο 8 στρεμμάτων, στην έξοδο της πόλης προς το λιμάνι της Σούδας [1]. Το εργοστάσιο αποτελείται σήμερα από τελευταίας τεχνολογίας μηχανολογικό εξοπλισμό και τηρεί άριστες προδιαγραφές. Η εταιρία διανέμει τα προϊόντα της σε όλη την Ελλάδα, ενώ εξάγει τα αναψυκτικά της στις ΗΠΑ, στον Καναδά και στην Αυστραλία. Η εταιρία ξεκίνησε το 1928 στο Γεράνι Κυδωνίας, Χανίων. Ιδρυτής ήταν ο Σοφοκλής Αναγνωστάκης όπου ξεκίνησε να δημιουργεί τα πρώτα αναψυκτικά στο καφενείο που λειτουργούσε στο πίσω μέρος της αυλής. Η εταιρία το 1984 έγινε Ανώνυμη Εταιρία (Α.Ε) αλλά δεν έχει χάσει την οικογενειακή της ταυτότητα γιατί, ακόμα και σήμερα, οι μέτοχοι της εταιρίας είναι ο κ. Μανώλης Αναγνωστάκης και ο κ. Σοφοκλής Αναγνωστάκης.

Τα προϊόντα της είναι η πορτοκαλάδα, η λεμονάδα, το μπυράλ, η γκαζόζα, το lime, το τόνικ, η σόδα και το ice tea σε γεύσεις ροδάκινο και λεμόνι. Επίσης, υπάρχει και γκαζόζα light, η οποία είναι για ανθρώπους που δεν καταναλώνουν ζάχαρη και απέχουν από αυτή. Μέχρι στιγμής εταιρία έχει καταφέρει να εξάγει προϊόντα σε 4 χώρες, όπως είναι οι Ηνωμένες Πολιτείες Αμερικής, η Γερμανία, Δανία και Αυστραλία.



**Εικόνα 1:** Αναψυκτικά Γεράνι [2]

Σύμφωνα με τον υπεύθυνο επικοινωνίας και marketing, «εμείς σαν τοπικοί παραγωγοί, πρέπει να είμαστε ευέλικτοι, πρέπει να είμαστε προσαρμοσμένοι στις ανάγκες του καταναλωτή, και, φυσικά, πρέπει να έχουμε μια εξαιρετική ποιότητα». Επίσης, για την εταιρία παίζει σημαντικό ρόλο ο ελληνισμός της ομογένειας: «όπου ζητά τα προϊόντα Γεράνι για να τα δοκιμάζει και στο τόπο που μένει. Επίσης, και οι τουρίστες, που έρχονται για να κάνουν διακοπές στην Κρήτη, δοκιμάζουν τα αναψυκτικά μας και μετά τα ζητάνε και πάλι στον τόπο διαμονής τους».

## 2.2 ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ

Ακολουθεί μια σύντομη περιγραφή για την πραγματική λειτουργία της επιχείρησης, για την διαδικασία τροφοδοσίας της και τι περιλαμβάνει, δηλαδή από ποια μέρη αποτελείται.

Η αρχή της διαδικασίας τροφοδοσίας είναι η παραγγελιοληψία μέσω πωλητών, τηλεφωνικής επικοινωνίας και γραπτής αλληλογραφίας (e-email) από τον Υπεύθυνο Παραγγελιοληψία, ο οποίος έχει την ευθύνη της καταχώρησης όλων των παραγγελιών των πελατών στο ηλεκτρονικό σύστημα. Μετά την καταχώρηση των παραγγελιών, γίνεται έλεγχος από τον Υπεύθυνο Αποθήκης, ο οποίος έχει πρόσβαση στο ηλεκτρονικό πρόγραμμα για τον έλεγχο αποθεμάτων του κάθε είδος ζητούμενου προϊόντος.

Η ζήτηση προϊόντων γίνεται κατά παραγγελίες που έρχονται σε τυχαίους χρόνους. Κάθε παραγγελία περιλαμβάνει ένα ή περισσότερα είδη προϊόντων σε διαφορετικές ποσότητες. Σε περίπτωση που τα αποθέματα δεν επαρκούν ειδοποιείται ο πελάτης και η παραγγελία καταχωρείται στη λίστα αναμονής (παραγγελία backorder). Μετά την ολοκλήρωση του ελέγχου γίνεται κατανομή των παραγγελιών σε δρομολόγιο και με ευθύνη του Υπευθύνου Γραφείου Κίνησης, εκτυπώνονται οι λίστες παραγγελιών (picking list) ανά δρομολόγιο και παραδίδονται στην αποθήκη για υλοποίηση. Σε περίπτωση που τα προϊόντα που ζητάει ο πελάτης δεν υπάρχουν σε επαρκή ποσότητα ενημερώνεται ο Διευθυντής Παραγωγής για την παραγωγή των απαιτούμενων ποσοτήτων.

Στο μεγαλύτερο ποσοστό οι πρώτες ύλες προέρχονται από ελληνικές εταιρείες καθώς η συγκεκριμένη έχει ως προτεραιότητα τη στήριξη των ελληνικών επιχειρήσεων. Οι προμηθευτές της επιχείρησης και οι πρώτες ύλες παραμένουν ίδιες και αξιολογούνται σύμφωνα με ένα πιστοποιημένο σύστημα ποιότητας. Όσον αφορά το κόστος αποθήκευσης της εταιρίας είναι μηδενικό, διότι όλες οι πρώτες ύλες εκτός από τους χυμούς που φυλάγονται σε κατάψυξη, δεν έχουν κάποιο κόστος στην αποθήκευση τους καθώς είναι αποθηκευμένες στις ιδιόκτητες εγκαταστάσεις της εταιρείας. Επιπλέον δεν υπάρχει κίνδυνος αλλοίωσης για οποιαδήποτε πρώτη ύλη, γιατί αποθηκεύονται με σωστό τρόπο ακόμα και στην περίπτωση διακοπής ρεύματος. Ωστόσο, οι πρώτες ύλες αναπληρώνονται σε συγκεκριμένο σημείο πριν τελειώσουν ώστε να μην υπάρξει ποτέ πρόβλημα από πιθανή έλλειψη τους και πάντα είναι σε ανάλογη ποσότητα σύμφωνα με τις ανάγκες που προκύπτουν από το προγραμματισμό και την οργάνωση της παραγωγής. Για τον έλεγχο των αποθεμάτων, έχει οριστεί από τον υπεύθυνο παραγωγής συγκεκριμένο άτομο ανάλογα με την πρώτη ύλη. Ένα εξίσου σημαντικό σημείο, είναι ότι η εταιρεία λαμβάνει υπόψη προσφορές πρώτων υλών σε ορισμένα είδη, αλλά μόνο στην περίπτωση που πληρούν τις προδιαγραφές με βάση το σύστημα ποιότητας που εφαρμόζει και είναι πιστοποιημένη η εταιρεία.

Από την εξέταση των τμημάτων του συστήματος, προέκυψε ότι τα τμήματα πρώτων υλών και εμφιάλωσης έχουν μικρές επιπτώσεις στους χρόνους διεκπεραίωσης παραγγελιών. Το τμήμα ανάμιξης και παραγωγής χυμών είναι υπεύθυνο τόσο για τις

καθυστερήσεις παραγγελιών όσο και για τα αποθέματα προϊόντων.

Στο τμήμα ανάμιξης κάθε προϊόν παράγεται κατά παρτίδες. Στο διάστημα που παράγεται μία παρτίδα, δεν μπορεί να παραχθεί άλλο προϊόν. Ο χρόνος έναρξης κάθε παρτίδας και το μέγεθός της αποφασίζονται από την διεύθυνση παραγωγής με βάση το τρέχον απόθεμα του αντίστοιχου προϊόντος και τη συνολική ζητούμενη ποσότητα που έχει συγκεντρωθεί στη λίστα παραγγελιών backorder. Αν ένα προϊόν παράγεται συχνά και σε μεγάλες ποσότητες τότε δεν υπάρχουν ελλείψεις, αλλά υπάρχουν μεγάλα αποθέματα. Αντίθετα, καθυστερήσεις στην παραγωγή ενός προϊόντος συνεπάγονται μικρά αποθέματα και ελλείψεις.

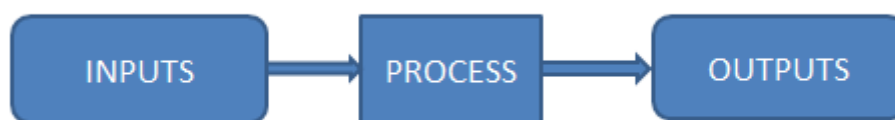
Οι αποφάσεις σχετικά με τα μεγέθη παρτίδων και τους χρόνους παραγωγής των είναι δυναμικές, δηλαδή αλλάζουν ανάλογα με το φόρτο παραγγελιών και τα τρέχοντα αποθέματα. Στην εργασία αυτή περιγράφεται και προτείνεται για εφαρμογή στην συγκεκριμένη εταιρεία μία πολιτική προγραμματισμού παρτίδων παραγωγής που χρησιμοποιείται συχνά σε συστήματα παραγωγής που εξυπηρετούν τυχαία ζήτηση.

### 3 ΜΟΝΤΕΛΑ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ

Στο κεφάλαιο αυτό παρουσιάζονται γενικές έννοιες των συστημάτων παραγωγής και ορισμένα από τα προβλήματα εκτίμησης μέτρων απόδοσης και βελτιστοποίησης συστημάτων παραγωγής τα οποία έχουν αναλυτική λύση.

#### 3.1 ΚΑΤΗΓΟΡΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ

Ένα στοιχειώδες στάδιο ή διαδικασία παραγωγής έχει ως εισόδους τις πρώτες ύλες και ως εξόδους τα παραγόμενα προϊόντα, όπως δείχνει το σχήμα.



Σχήμα 1: Σχηματικό διάγραμμα συστήματος

Ένα σύστημα παραγωγής μπορεί να χαρακτηριστεί από τις αλλαγές που υφίσταται ως συνεχές ή διακριτό.

**Συνεχές σύστημα:** Ένα σύστημα στο οποίο κατά κύριο λόγο οι μεταβολές της κατάστασης είναι ομαλές, δηλαδή οι λειτουργίες τροποποιούν το σύστημα συνεχώς και όχι μόνο όταν ολοκληρώνονται. Για παράδειγμα η ροή υγρού από μία δεξαμενή σε άλλη είναι συνεχής λειτουργία.

**Διακριτό σύστημα:** Το σύστημα στο οποίο οι αλλαγές είναι κατά κύριο λόγο ασυνεχείς. Για παράδειγμα η ολοκλήρωση μίας παρτίδας και το ξεκίνημα της παραγωγής παρτίδας άλλου προϊόντος είναι μία διακριτή μεταβολή.

Στο πλαίσιο της εργασίας θα ασχοληθούμε με **υβριδικά συστήματα**: οι παραγόμενες παρτίδες, οι αφίξεις παραγγελιών και οι εξυπηρέτησή τους είναι διακριτά μεγέθη αλλά όμως η ποσότητα χυμού που απαρτίζει μία παρτίδα παράγεται σε συνεχή χρόνο.

Οι μεταβλητές που περιγράφουν ένα σύστημα παραγωγής είναι οι ακόλουθες:

- Ο ρυθμός παραγωγής κάποια χρονική στιγμή (παραγωγή ανά μονάδα χρόνου).
- Ο ρυθμός παραγγελιών εκφράζει τον ρυθμό άφιξης παραγγελιών στη μονάδα χρόνου.
- Απόθεμα το οποίο χωρίζεται στις ακόλουθες υποκατηγορίες.
  - Απόθεμα πρώτων υλών
  - Απόθεμα παραγωγής
  - Απόθεμα ετοιμών προϊόντων

Οι παράμετροι είναι μεγέθη τα οποία χαρακτηρίζουν – συμβολίζουν τις παραπάνω μεταβλητές. Έχουμε παραμέτρους που χαρακτηρίζουν τους ρυθμούς άφιξης και παραγωγής και παραμέτρους που χαρακτηρίζουν τον αριθμό των αποθεμάτων.

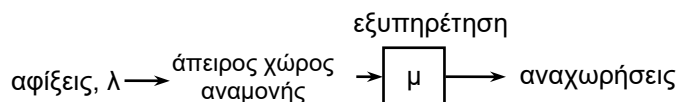
## 3.2 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ ΜΕ ΜΟΝΤΕΛΑ ΟΥΡΩΝ ΑΝΑΜΟΝΗΣ

Σε ένα απλό σύστημα αναμονής γίνονται αφίξεις πελατών σε έναν χώρο αναμονής όπου μένουν μέχρι να ξεκινήσει η εξυπηρέτησή τους. Κάποια συστήματα αναμονής λέγονται συστήματα γέννησης-θανάτου όταν κάθε άφιξη αυξάνει τον αριθμό πελατών κατά 1 (γέννηση) και κάθε εξυπηρέτηση τον μειώνει κατά 1 (θάνατος). Για τα συστήματα που περιγράφονται στην παράγραφο αυτή υπάρχουν τύποι υπολογισμού της παραγωγικότητας, των μέσων αποθεμάτων και των καθυστερήσεων.

### 3.2.1 Σύστημα αναμονής M/M/1

Το σύστημα M/M/1 είναι ένα σύστημα γέννησης-θανάτου στο οποίο ο αριθμός αφίξεων σε ορισμένο διάστημα έχει κατανομή Poisson με μέσο ρυθμό  $\lambda$  πελάτες ανά μονάδα χρόνου ή, ισοδύναμα, οι χρόνοι μεταξύ αφίξεων έχουν εκθετική κατανομή με μέση τιμή  $1/\lambda$  και οι διάρκειες εξυπηρέτησης έχουν επίσης εκθετική κατανομή με μέση τιμή  $1/\mu$ .

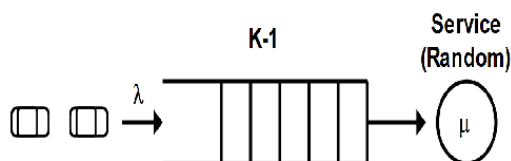
Το σύστημα έχει ένα εξυπηρετητή και άπειρη χωρητικότητα. Παρακάτω παρουσιάζεται η σχηματική μορφή του συστήματος.



Σχήμα 2: Διάγραμμα συστήματος M/M/1

### 3.2.2 Σύστημα αναμονής M/M/1/K

Το σύστημα M/M/1/K είναι το σύστημα γέννησης θανάτου, το οποίο έχει ίδια χαρακτηριστικά με το M/M/1 και η μόνη διαφορά είναι ότι έχει περιορισμένη χωρητικότητα που ισούται με K. Ένα τέτοιο σύστημα φαίνεται στο σχήμα 3.



Σχήμα 3: Διάγραμμα συστήματος M/M/1/K [4]

### 3.2.3 Σύστημα αναμονής M/M/m/K

Το σύστημα αυτό χαρακτηρίζεται ως ένα σύστημα M/M/1/K με m εξυπηρετητές, οι οποίοι λειτουργούν παράλληλα, και χωρίς χώρο αναμονής. Η διαφορά με το M/M/1/K είναι το γεγονός ότι μπορεί να εξυπηρετεί έως m πελάτες ταυτόχρονα. Εδώ

$$K = m + (\text{χωρητικότητα ουράς}) \geq m$$

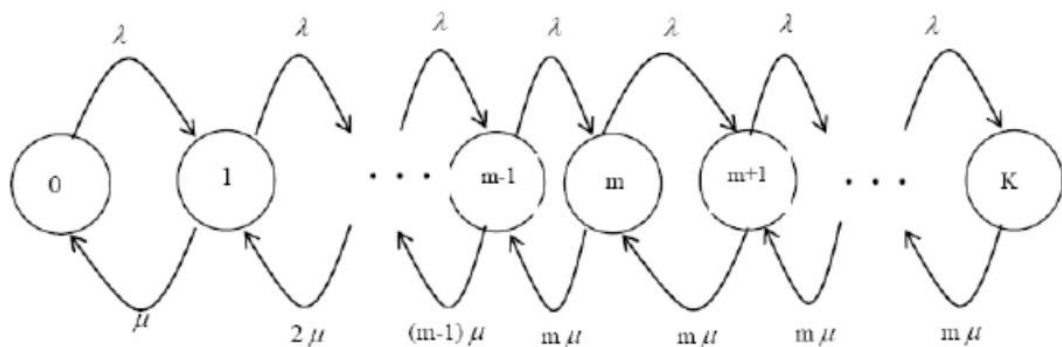
### 3.2.4 Επίλυση συστημάτων αναμονής

Η κατάσταση ενός συστήματος αναμονής για μία στιγμή δίνεται από τον αριθμό πελατών στο σύστημα (σε ουρά και εξυπηρέτηση) και συμβολίζεται n. Για την εκτίμηση της απόδοσης τέτοιων συστημάτων υπολογίζουμε τις πιθανότητες P(n) στη μόνιμη κατάσταση, δηλαδή μετά από άπειρο χρόνο λειτουργίας.

Για συστήματα γέννησης-θανάτου σχεδιάζουμε ένα διάγραμμα με τις καταστάσεις σε κύκλους. Από μία κατάσταση n μπορούμε να μεταβούμε είτε με άφιξη στην ανώτερη n+1 είτε με εξυπηρέτηση πελάτη στην n-1. Κάθε αλλαγή έχει έναν μέσο ρυθμό εμφάνισης. Οι αφίξεις έχουν ρυθμούς λ. Οι εξυπηρετήσεις γίνονται με μέσους ρυθμούς ανάλογους του αριθμού εξυπηρετητών που λειτουργούν κάθε φορά. Για παράδειγμα αν στο M/M/m/K υπάρχει n = 1 πελάτης, ο ρυθμός του συστήματος είναι 1μ επειδή ένας εξυπηρετών εργάζεται. Αν υπάρχουν n ≥ m, τότε ο ρυθμός είναι mμ. Γενικά ο ρυθμός παραγωγής είναι

$$\mu \min(n, m)$$

Στο παρακάτω σχήμα απεικονίζεται διαγραμματικά το διάγραμμα του M/M/m/m.



Σχήμα 4: Διάγραμμα κατάστασης του συστήματος M/M/m/m [5]

Οι πιθανότητες μόνιμης κατάστασης είναι

$$P(1) = (\lambda/\mu) P(0)$$

$$P(2) = (\lambda/2\mu) P(1) = (\lambda/\mu) (\lambda/2\mu) P(0)$$

...



και γενικά δύο γειτονικές καταστάσεις συνδέονται με έναν συντελεστή που προκύπτει διαιρώντας τους ρυθμούς επάνω στα βέλη που τις συνδέουν. Από αυτές τις εξισώσεις, όλες οι πιθανότητες  $P(n)$  εκφράζονται συναρτήσει της  $P(0)$ . Τελικά η άγνωστη  $P(0)$  ευρίσκεται λύνοντας την **εξίσωση κανονικοποίησης**:

$$P(0) + P(1) + P(2) \dots + P(K) = 1$$

όταν όλα τα  $P(n)$  αντικατασταθούν από τις εκφράσεις τους συναρτήσει της  $P(0)$ . Για το M/M/m/K έχουμε

$$P(n) = (\lambda/\mu) (\lambda/2\mu) \dots (\lambda/n\mu) P(0) \quad \text{για } n \leq m$$

$$P(n) = (\lambda/\mu) (\lambda/2\mu) \dots (\lambda/m\mu) (\lambda/m\mu)^{n-m-1} \quad \text{για } n = m+1, \dots, K$$

Από αυτές και την εξίσωση κανονικοποίησης υπολογίζονται όλες οι πιθανότητες.

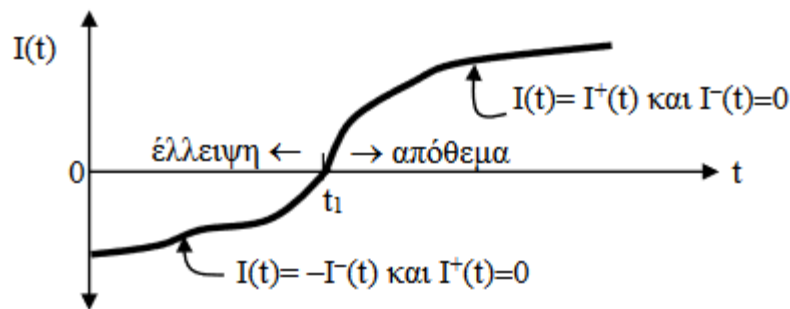
Δυστυχώς τέτοια μοντέλα δεν μπορούν να περιγράψουν την παραγωγή ροϊκού υλικού κατά παρτίδες.

### 3.3 ΜΟΝΤΕΛΑ ΔΙΑΚΡΙΤΩΝ ΓΕΓΟΝΟΤΩΝ

Τα μοντέλα αυτά είναι προγράμματα στον υπολογιστή που προσομοιώνουν ένα τυχαίο σενάριο ζήτησης και καταγράφουν ό,τι μπορεί να συμβεί σε ένα εργοστάσιο μέσα σε κάποιο χρονικό διάστημα λειτουργίας. Για την προσομοίωση ενός σεναρίου λειτουργίας χρησιμοποιείται η έννοια των **γεγονότων**, την οποία θα περιγράψουμε παρακάτω για ένα σύστημα που παράγει κατά παρτίδες, όπως αυτό που εξετάζουμε.

Για κάθε σύστημα παραγωγής που εξετάζεται, [6], όλες οι τυχαίες μεταβλητές οι οποίες περιγράφουν τυχαία φαινόμενα, όπως για παράδειγμα οι διαδοχικοί χρόνοι μεταξύ αφίξεων πελατών ή παραγγελιών, η διάρκεια εξυπηρέτησης στο εξής θεωρούνται ότι είναι μεταξύ τους ανεξάρτητα μεγέθη.

Ένα αποθεματικό σύστημα μπορεί να παρουσιαστεί ως εξής: Μια επιχείρηση προμηθεύεται ή παράγει ένα προϊόν, το αποθηκεύει και κατόπιν το πουλάει στους πελάτες τους που το ζητούν σε τυχαίους χρόνους. Υπάρχουν τρία πιθανά σενάρια που περιγράφουν την κατάσταση του αποθέματος. Αρχικά, το απόθεμα συμβολίζεται με  $I_t$ . Αν υπάρχει προϊόν στην αποθήκη τότε  $I_t > 0$  ενώ αν υπάρχει έλλειμμα τότε  $I_t < 0$ . Στο παρακάτω διάγραμμα απεικονίζεται η εξέλιξη ενός αποθεματικού συστήματος με συνεχή εισροή (παραγωγή) και εκροή (ζήτηση) όταν η εισροή είναι μεγαλύτερη.



Γράφημα 1: Εξέλιξη καθαρού αποθέματος [6]

Πιο αναλυτικά, όταν υπάρχει απόθεμα, τότε οι πωλήσεις καθορίζονται μόνο από την ζήτηση, επειδή όταν έρχεται μια καινούρια παραγγελία και εφόσον υπάρχει απόθεμα ικανοποιείται απευθείας. Όμως όταν υπάρχει έλλειψη αποθέματος, τότε οι πωλήσεις καθορίζονται από την παραγωγή. Επεξηγηματικά όταν παράγεται ένα καινούριο προϊόν και υπάρχουν εκκρεμείς παραγγελίες, το επόμενο είναι να γίνεται αμέσως η πώληση. Όταν  $I_t = 0$ , οι πωλήσεις καθορίζονται από τον ρυθμό παραγωγής ή ρυθμό άφιξης παραγγελιών που είναι μεγαλύτερος. Η κυκλοφορία υλικών σε ένα αποθεματικό σύστημα παραγωγής καθορίζεται από τα εξής **γεγονότα**:

- Ζήτηση προϊόντων
- Προετοιμασία Μηχανής, όπου περιλαμβάνονται τα παρακάτω στάδια:
  - Αρχή προετοιμασίας
  - Αρχή παραγωγής παρτίδας
  - Τέλος παραγωγής παρτίδας
- Παράδοση παραγγελίας

Η παραγωγική διαδικασία μπορεί να αποδοθεί ως εξής. Αρχικά, η εταιρεία (το σύστημα) δέχεται παραγγελίες από πελάτες. Όταν ο πελάτης ζητάει  $D_t$  μονάδες προϊόντων κάποιου τύπου και  $I_t \geq D_t$  η παραγγελία του ικανοποιείται αμέσως από το διαθέσιμο απόθεμα εκείνης της στιγμής. Αν  $I_t < D_t$  τότε το διαθέσιμο απόθεμα μηδενίζεται και το υπόλοιπο μέρος της παραγγελίας καταγράφεται ως εκκρεμής και υπολογίζεται ως έλλειμμα. Έχουμε τότε νέο  $I_t = I_t - D_t < 0$ . Είναι αδύνατον να έχουμε συγχρόνως και διαθέσιμο απόθεμα και έλλειμμα.

Ανάλογα πράγματα εφαρμόζονται για **όλα τα αναψυκτικά**. Το κάθε αναψυκτικό παράγεται κατά παρτίδες. Η εταιρεία εφαρμόζει την ακόλουθη πολιτική αναπλήρωσης αποθέματος:

- Το σύστημα εξυπηρετεί **πρώτα τις πιο παλιές εκκρεμείς παραγγελίες**. Μία παραγγελία που μόλις έφθασε δεν εξυπηρετείται αν πριν από αυτήν υπάρχουν άλλες με προτεραιότητα.
- Όταν η μηχανή ανάμιξης είναι ελεύθερη, επιλέγει για παραγωγή το προϊόν με το μεγαλύτερο έλλειμμα.

- Αν σε κάποιο ενδιάμεσο στάδιο παραγωγής μίας παρτίδας μπορεί να συμπληρωθεί εντελώς μία παραγγελία από εκείνες που είναι σε προτεραιότητα τότε η παραγγελία εξυπηρετείται αμέσως.

### 3.4 ΣΥΣΤΗΜΑ ΠΑΡΑΓΩΓΗΣ ΚΑΤΑ ΠΑΡΤΙΔΕΣ (Lot sizing)

Εξετάζουμε τώρα ένα άλλο αναλυτικό μοντέλο για την περιγραφή παραγωγής κατά παρτίδες. Το μοντέλο αυτό είναι κατάλληλο για συστήματα με σταθερή ζήτηση, οπότε δεν είναι άμεσα εφαρμόσιμο στην περίπτωση που εξετάζουμε, όμως καλύπτει την έννοια της **ευστάθειας** η οποία είναι χρήσιμη για την ανάλυση του συστήματος.

Αρκετά στοιχεία της παραγράφου αυτής έχουν ληφθεί από τις σημειώσεις του μαθήματος "Συστήματα Παραγωγής" που διδάσκεται στο Πολυτεχνείο Κρήτης [7]. Έχουμε ένα σύστημα παραγωγής που παράγει  $N$  προϊόντα, των οποίων οι ρυθμοί ζήτησης είναι γνωστοί και σταθεροί. Η παραγωγή γίνεται κατά παρτίδες. Τα προϊόντα παράγονται και αποθηκεύονται με τέτοιο τρόπο ώστε να μην υπάρχει έλλειψη. Το κόστος λειτουργίας του συστήματος περιλαμβάνει το κόστος παραγωγής, προετοιμασίας για κάθε παρτίδα, και το κόστος αποθέματος. Το συνολικό μέσο κόστος ανά μονάδα χρόνου είναι:

$$K = \sum_{i=1}^N K_i$$

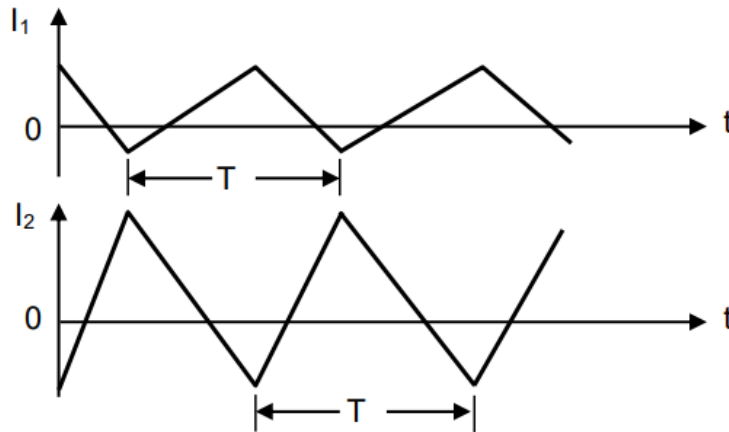
όπου το  $K_i$  είναι η συνιστώσα που έχει κόστος παραγωγής, κόστος προετοιμασίας, και κόστος αποθέματος που αντιστοιχούν στο προϊόν  $i$ . Το πρόβλημα είναι να υπολογιστούν τα μεγέθη παραγόμενων παρτίδων που ελαχιστοποιούν το  $K$ .

Επειδή η ζήτηση είναι σταθερή, στο σύστημα ορίζεται ένας σταθερός κύκλος παραγωγής διάρκειας, έστω,  $T$  μέσα στον οποίο προγραμματίζεται η παραγωγή όλων των προϊόντων και η διαδικασία επαναλαμβάνεται στον επόμενο κύκλο. Μία ειδική κατηγορία περιοδικών προγραμμάτων παραγωγής είναι τα κυκλικά, όπου κάθε είδος προϊόντος παράγεται μία μόνον φορά σε κάθε κύκλο και πάντοτε με την ίδια σειρά. Έστω ένα σύστημα με 5 προϊόντα. Τα ακόλουθα προγράμματα είναι κυκλικά:

..., (1, 2, 3, 4, 5), (1, 2, 3, 4, 5), ...

..., (2, 4, 3, 5, 1), (2, 4, 3, 5, 1), ...

Παρακάτω απεικονίζεται μια ενδεικτική κυκλική παραγωγή για δύο προϊόντα  $i = 1, 2$ .



Γράφημα 2: Τα αποθέματα για κυκλική παραγωγή δύο προϊόντων [7]

Κάθε κύκλος ξεκινά με την παραγωγή μίας παρτίδας προϊόντος 1 και ακολουθεί η παραγωγή παρτίδας του 2. Έχουμε δηλαδή το πρόγραμμα ... (1,2), (1,2) .... Θα μπορούσε ωστόσο κάποιος να θεωρήσει ως έναρξη κύκλου την έναρξη της παραγωγής του 2 και να έχει το **ισοδύναμο** πρόγραμμα ... (2,1), (2,1) .....

### 3.4.1 Μεγέθη παρτίδων για κυκλική παραγωγή με σταθερή ζήτηση

Στη συνέχεια εξετάζουμε κυκλικά προγράμματα παραγωγής. Δεν ενδιαφέρει η σειρά, αφού κάθε  $T$  μονάδες χρόνου επαναλαμβάνεται η παραγωγή του ίδιου προϊόντος. Έστω  $T_i$  ο χρόνος που αφιερώνεται για την παραγωγή του προϊόντος  $i$  σε κάθε κύκλο διάρκειας  $T$ .

Μπορεί, πριν ξεκινήσει η παραγωγή ενός προϊόντος,

- α) να πρέπει να γίνει προετοιμασία της εγκατάστασης παραγωγής για κάποιο χρόνο (καθαρισμός από προηγούμενο προϊόν, αλλαγή στις ρυθμίσεις της μηχανής),
- β) ή να συμφέρει να παρεμβάλλουμε έναν επί πλέον χρόνο αδράνειας στην παραγωγή
- γ) ή να γίνουν και το α) και το β).

Επομένως το  $T$  περιλαμβάνει πιθανόν και άλλους χρόνους πλην των  $T_i$ . Το πρόβλημα είναι να προσδιορίσουμε το  $T$  με το ελάχιστο κόστος. Έστω ότι η ζήτηση του  $i$  έχει ρυθμό  $D_i$  και αυτό μπορεί να παραχθεί με ρυθμό  $P_i$ . Λόγω περιοδικότητας, η παραγόμενη ποσότητα  $Q_i$  πρέπει να είναι ίση με τη ζήτηση σε χρόνο  $T$ . Το μέγεθος παρτίδας  $Q_i$  και η διάρκεια παραγωγής  $T_i$  του προϊόντος  $i$  σε έναν κύκλο  $T$  υπολογίζονται με τις εξισώσεις:

$$Q_i = T_i P_i = T D_i \rightarrow T_i = \rho_i T \Rightarrow \rho_i = \frac{D_i}{P_i}$$

όπου

$D_i$ : ρυθμός ζήτησης του προϊόντος  $i$   
 $Q_i$ : παραγόμενη ποσότητα  
 $T$ : χρόνος προσομοίωσης  
 $T_i$ : διάρκεια παραγωγής  
 $P_i$ : ρυθμός παραγωγής  
 $h_i$ : κόστος αποθέματος ανά μονάδα χρόνου

Η ποσότητα  $\rho_i = \frac{D_i}{P_i}$  είναι το κλάσμα του χρόνου που οι εγκαταστάσεις απασχολούνται για την παραγωγή του  $i$ . Αν για κάποιο προϊόν ισχυρε ότι  $D_i = P_i$  τότε θα έπρεπε να το παράγουμε συνεχώς και αποκλειστικά για να μην υπάρχει έλλειψη. Τότε όμως δεν θα μπορούσαν να παραχθούν τα άλλα προϊόντα και θα είχαν εκείνα έλλειψη και μάλιστα συνεχώς αυξανόμενη με την πάροδο του χρόνου. Υποθέτουμε ότι ισχύει  $P_i > D_i$  ή  $\rho_i < 1$ . Το άθροισμα  $\rho = \rho_1 + \dots + \rho_n$  δίνει την μέση απασχόληση της μηχανής ανά μονάδα χρόνου για όλα τα προϊόντα. Αυτή δεν πρέπει να υπερβαίνει το 100% του διαθέσιμου χρόνου. Άρα για να μην έχουμε έλλειψη σε κανένα προϊόν πρέπει :

$$\rho = \sum_{i=1}^N \rho_i < 1$$

Η συνθήκη αυτή λέγεται **συνθήκη ευστάθειας**. Είναι απαραίτητη ούτως ώστε η ζήτηση να μην "υπερφορτώνει συνεχώς" το σύστημα παραγωγής.

Έστω τώρα ότι πριν την παραγωγή κάθε παρτίδας απαιτείται και προετοιμασία και ότι

$A_i$ : κόστος προετοιμασίας για την παραγωγή παρτίδας προϊόντος  $i$

Ο συνολικός ρυθμός κόστους (κόστος ανά μονάδα χρόνου) είναι [7]:

$$K(T) = \sum_{i=1}^N \left[ \frac{A_i D_i}{Q_i} + C_i D_i + h_i \frac{Q_i}{2} (1 - \rho_i) \right]$$

Αντικαθιστώντας  $Q_i = T D_i$  προκύπτει

$$K(T) = \sum_{i=1}^N \left[ \frac{A_i}{T} + C_i D_i + h_i \frac{T D_i}{2} (1 - \rho_i) \right]$$

και λύνοντας ως προς  $T$

$$\frac{\partial K}{\partial T} = 0 \Rightarrow T^* = \sqrt{\frac{2 \sum_{i=1}^N A_i}{\sum_{i=1}^N h_i D_i (1 - \rho_i)}}$$

Έστω τώρα ότι πριν την παραγωγή κάθε παρτίδας απαιτείται και χρόνος προετοιμασίας των εγκαταστάσεων, η οποία για την παρτίδα των  $i$  παίρνει χρόνο  $S_i$ . Αν η παραγωγή διαρκεί  $T_i$  τότε πρέπει ο χρόνος  $T$  να είναι αρκετός για παραγωγή και προετοιμασίες:

$$\sum_{i=1}^N (S_i + T_i) \leq T$$

Είδαμε ότι  $T_i = \rho_i T$ . Με αντικατάσταση, η ανισότητα γίνεται:

$$\sum_{i=1}^N S_i \leq T \left( 1 - \sum_{i=1}^N \rho_i \right)$$

Διαιρώντας κατά μέλη με  $(1 - \sum \rho_i) > 0$ , προκύπτει τελικά η συνθήκη [7]:

$$T \geq \frac{\sum_{i=1}^N S_i}{1 - \sum_{i=1}^N \rho_i} \triangleq T_{min}$$

Εδώ έχουμε ένα πρόβλημα με περιορισμό ανισότητας. Σύμφωνα με τον αλγόριθμο της βελτιστοποίησης υπό περιορισμούς λύνουμε το πρόβλημα αγνοώντας την ανισότητα και βρίσκουμε λύση  $T = T^*$ . Αν η τιμή της είναι  $< T_{min}$  τότε θεωρούμε τον περιορισμό ως ισότητα, άρα θέτουμε  $T = T_{min}$ . Γενικά η βέλτιστη λύση είναι

$$T_{opt} = \min (T^*, T_{min}).$$

### 3.4.2 Εφαρμογές συστημάτων παραγωγής κατά παρτίδες

Στη βιβλιογραφία συνηθίζεται η χρήση συστημάτων μεγέθους παρτίδων. Οι κύριοι τομείς εφαρμογής είναι οι εξής:

- Παραγωγή
- Τρόφιμα και ποτά
- Βιομηχανίες
- Αλυσίδα εφοδιασμού
- Γεωργία

Η παρούσα εργασία επικεντρώνεται σε μελέτες για τον προγραμματισμό της παραγωγής αναψυκτικών. Ως εκ τούτου, αξίζει να δοθούν ορισμένες περαιτέρω εφαρμογές σε βιομηχανίες. Οι περισσότερες από τις εφαρμογές βασίζονται στο Σύγχρονο και Ολοκληρωμένο Πρόβλημα Μεγέθους και Προγραμματισμού Παρτίδων Δύο Επιπέδων (Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem-SITLSP). Ειδικότερα, το πρόβλημα αυτό συναντάται σε ορισμένα βιομηχανικά περιβάλλοντα, κυρίως σε εταιρείες αναψυκτικών, όπου η παραγωγική διαδικασία περιλαμβάνει δύο αλληλεξαρτώμενα επίπεδα με αποφάσεις που αφορούν την αποθήκευση πρώτων υλών και την εμφιάλωση αναψυκτικών. Η πρόκληση είναι ο ταυτόχρονος προσδιορισμός του μεγέθους της παρτίδας και του προγραμματισμού των πρώτων υλών στις δεξαμενές και των αναψυκτικών στις γραμμές εμφιάλωσης, όπου το κόστος και οι χρόνοι εγκατάστασης εξαρτώνται από τα προηγούμενα στοιχεία που αποθηκεύτηκαν και εμφιαλώθηκαν, [9]. Σε άλλη εργασία χρησιμοποιείται η ίδια μέθοδος για τον προσδιορισμό της ποσότητας και του χρόνου που πρέπει να

αποθηκευτούν και να παραχθούν οι πρώτες ύλες και τα προϊόντα, [10]. Μια ακόμη εφαρμογή που βασίζεται σε μαθηματικά μοντέλα. Στην εργασία των Maldonado, [11] η παραγωγή αναψυκτικών περιλαμβάνει δύο βασικά στάδια: την παρασκευή σιροπιού και την εμφιάλωση. Για να προκύψει η ακολουθία των παρτίδων στο στάδιο της εμφιάλωσης, μελετώνται τρεις προσεγγίσεις. Βασίζονται στους περιορισμούς εξάλειψης των επιμέρους διαδρομών που χρησιμοποιούνται στα μαθηματικά μοντέλα για το πρόβλημα του πλανόδιου πωλητή. Δύο από τα μαθηματικά μοντέλα προέρχονται από τη βιβλιογραφία και χρησιμοποιούν κλασικούς περιορισμούς. Το τρίτο μοντέλο περιλαμβάνει περιορισμούς ροής πολλαπλών εμπορευμάτων για την εξάλειψη ασύνδετων υπό διαδρομών.

## 4 ΑΝΑΠΤΥΞΗ ΜΟΝΤΕΛΟΥ ΜΕ ΛΟΓΙΣΜΙΚΟ SIMIO

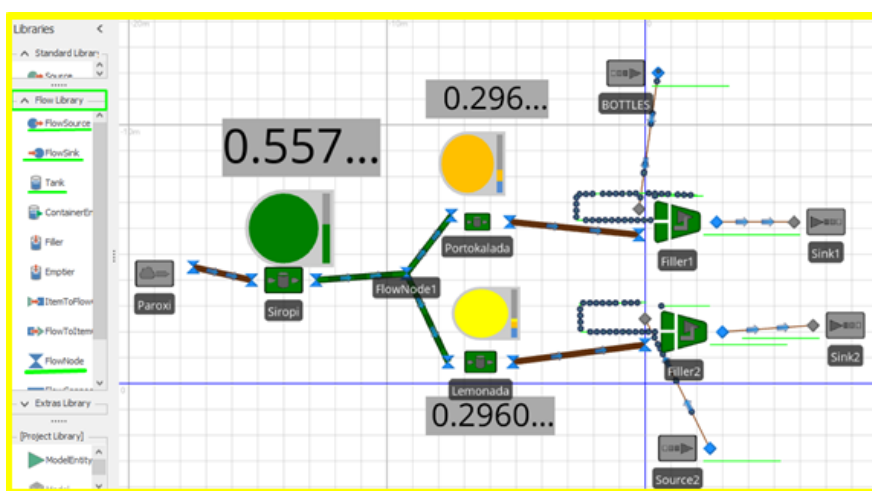
### 4.1 ΕΙΣΑΓΩΓΗ

Το SIMIO είναι λογισμικό που διευκολύνει την ανάπτυξη μοντέλων προσομοίωσης γιατί χρησιμοποιεί έτοιμα αντικείμενα για αποθήκες, μηχανές και ουρές αναμονής. Συνήθως ο χρήστης εργάζεται σε δισδιάστατη λειτουργία αλλά πλέον είναι εφικτή η τρισδιάστατη σχεδίαση αφού είναι πλήρως αυτοματοποιημένη και μπορεί να λειτουργήσει μόλις με ένα πάτημα κουμπιού από το πληκτρολόγιο.

Η διαδικασία κατασκευής ενός αντικειμένου στο SIMIO είναι πανομοιότυπη με τη διαδικασία κατασκευής ενός μοντέλου. Στην πραγματικότητα δεν υπάρχει καμία διαφοροποίηση μεταξύ ενός αντικειμένου και ενός μοντέλου. Αυτή η έννοια αναφέρεται ως αρχή της ισοδυναμίας και έχει ιδιαίτερη σημασία για το σχεδιασμό στο SIMIO. Κάθε φορά που κατασκευάζεται ένα μοντέλο είναι εξ ορισμού ένα αντικείμενο που μπορεί να ενσωματωθεί σε ένα άλλο μοντέλο. Η εκτέλεση ενός πειράματος μέσω SIMIO, αναπαριστώντας γραφικά τρισδιάστατα αντικείμενα, δείχνει την εκτέλεση σε πραγματικό χρόνο της μεταβολής των μεγεθών που εξετάζουμε μαζί με κάποια γραφικά αντικείμενα και πως αυτά μεταβάλλονται με την πάροδο του χρόνου. Έτσι είναι δυνατόν να αντιληφθεί πλήρως το μοντέλο και πώς αυτό εξελίσσεται χρονικά ακόμη και κάποιος που δεν έχει πλήρη γνώση της λειτουργίας του συστήματος.

### 4.2 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΤΟ ΛΟΓΙΣΜΙΚΟ SIMIO

Για την μοντελοποίηση του συστήματος χρησιμοποιήσαμε την βιβλιοθήκη Flow Library,[12], η οποία έχει αντικείμενα σχετικά με το μοντέλο μας: επεξεργασία υγρού και ροές μεταξύ αντικειμένων όπως δεξαμενές, αναδευτήρες, πηγές παραγωγής κλπ. Για απλούστευση περιγράφουμε ένα σύστημα με μόνο δυο τύπους παραγωγής προϊόντων (πορτοκαλάδες και λεμονάδες). Η εικόνα του συστήματος είναι η ακόλουθη:



Εικόνα 2: Μοντελοποίηση παραγωγής δύο προϊόντων στο λογισμικό SIMIO



Στην εικόνα φαίνονται τα στάδια παραγωγής και η δρομολόγηση για κάθε τύπο προϊόντος. Υπάρχουν δεξαμενές οι οποίες περιέχουν την πρώτη ύλη του αναψυκτικού που παράγεται και την κατάλληλη ποσότητα νερού. Στη συνέχεια και αφού περάσει ο κατάλληλος χρόνος που χρειάζεται το παραγόμενο αναψυκτικό είναι έτοιμο και γεμίζει η συσκευασία. Πιο συγκεκριμένα, από το σχήμα μπορούμε να διακρίνουμε τα εξής:

Την βασική παροχή, η οποία μπορεί να λειτουργεί σύμφωνα με τον ρυθμό πλήρωσης, και όπου αρχικά γεμίζει με νερό. Εκτός από αυτό περιλαμβάνεται η βασική δεξαμενή που περιέχει το σιρόπι. Η κύρια δεξαμενή έχει και μια ένδειξη που μας πληροφορεί για την ποσότητα που περιέχει σε λίτρα και αυτό γίνεται για πιο εύκολη επισκόπηση και αναπαράσταση των δεδομένων μας. Μόλις η δεξαμενή αυτή γεμίσει με την επιθυμητή ποσότητα που έχει καθοριστεί εξ αρχής και ανακατευτεί ώστε το σιρόπι να είναι έτοιμο, διοχετεύεται σε ένα σωλήνα και στη συνέχεια το υγρό περνά σε δύο επιμέρους σωλήνες με σκοπό την πλήρωση των δεξαμενών. Οι δύο νέες δεξαμενές έχουν ενδείξεις που μας πληροφορούν το ποσό του κάθε έτοιμου υλικού σε σχέση με την ποσότητα, χωρητικότητα τους σε λίτρα. Επίσης υπάρχει και κατάλληλη ποσότητα μπουκαλιών που εμείς καθορίζουμε την χωρητικότητα τους, με βάση τις ανάγκες της παραγγελίας και στη συνέχεια αρχίζουν να γεμίζουν μόλις ετοιμαστεί το υλικό του κάθε τύπου.

#### **4.3 ΔΙΑΔΙΚΑΣΙΑ ΕΙΣΑΓΩΓΗΣ ΠΑΡΑΜΕΤΡΩΝ**

Για τη συλλογή των απαραίτητων αποτελεσμάτων του συστήματος, μεταβάλλεται μια σειρά από παράμετρος. Η πρώτη υπό μεταβολή παράμετρος, είναι η κατανομή που ακολουθεί η αρχική ροή και με βάση αυτή γεμίζει η βασική δεξαμενή. Γίνονται υπολογιστικά τρεξίματα με διαφορετικές κατανομές και χρόνους άφιξης, αλλά και επεξεργασίας των παραγγελιών. Το δεύτερο στοιχείο που αλλάζουμε είναι οι αρχικές ποσότητες των υλικών που χρειάζονται όπως επίσης και οι ποσότητες του νερού σε κάθε δεξαμενή. Από τις μεταβολές που κάνουμε προκύπτει μια σειρά αποτελεσμάτων για επεξεργασία και εκτενέστερη μελέτη.

#### **4.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ**

Μετά από αρκετές εκτελέσεις που υλοποιήθηκαν, διαπιστώσαμε ότι σύμφωνα με τις παραγγελίες καθορίζονται οι ποσότητες που θα αναμειχθούν τα υλικά. Οπότε οι εκτελέσεις που γίνονται ικανοποιούν όλες τις παραγγελίες χωρίς να αφήνουν εκκρεμότητες. Επίσης, καταγράψαμε τα αποτελέσματα προσομοίωσης κατά την ολοκλήρωση της παραγωγής χωρίς όμως να τις παρουσιάσουμε, αφού δεν υλοποιήθηκε το ακριβές σενάριο παραγωγής κατά παρτίδες. Η SIMIO ήταν η πρώτη προσπάθεια αναπαράστασης της λειτουργίας του συστήματος αλλά χωρίς μεγάλη επιτυχία. Μία ακριβέστερη μοντελοποίηση περιγράφεται στο επόμενο κεφάλαιο.

## 5 ΜΟΝΤΕΛΟ ΡΟΗΣ ΜΕ ΧΡΗΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ MATLAB

### 5.1 ΕΙΣΑΓΩΓΗ

Η MATLAB είναι μία γλώσσα προγραμματισμού υψηλού επιπέδου που δίνει στον χρήστη δυνατότητα να περιγράψει μεταξύ άλλων και μοντέλα διακριτών γεγονότων με μαθηματικές εκφράσεις και λογικές σχέσεις. Το συγκεκριμένο περιβάλλον χρησιμοποιήθηκε για την ανάπτυξη

- ενός προγράμματος που βρίσκει διάφορους συνδυασμούς εναλλακτικών πολιτικών παραγωγής κατά παρτίδες
- και ενός προγράμματος που προσομοιώνει τη λειτουργία του εργοστασίου για δεδομένη πολιτική παραγωγής.

Ξεκινάμε με την περιγραφή του αλγορίθμου προσομοίωσης.

### 5.2 ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

Το συγκεκριμένο σύστημα παραγωγής χρησιμοποιεί τον ακόλουθο αλγόριθμο, ο οποίος προσομοιώνει την λειτουργία του συστήματος και στο οποίο είναι εφικτή η εφαρμογή μιας βέλτιστης πολιτικής. Στη συνέχεια, ακολουθεί αναλυτική περιγραφή του τρόπου λειτουργίας της επιχείρησης μέσω του αλγορίθμου.

Το εργοστάσιο δέχεται παραγγελίες και κάθε παραγγελία διαθέτει πολλούς διαφορετικούς τύπους προϊόντων. Για να ικανοποιήσει αυτές τις παραγγελίες θα πρέπει να διαθέτει τις ζητούμενες ποσότητες από το κάθε τύπο προϊόντος. Στην περίπτωση που κάποιος τύπος προϊόντος δεν υπάρχει ή υπάρχει σε ανεπαρκή ποσότητα, θα πρέπει να δοθεί εντολή στη μηχανή ανάδευσης να παράγει τον συγκεκριμένο τύπο προϊόντος. Η παραγωγή γίνεται κατά παρτίδες και κάθε χρονική στιγμή μπορεί να παράγεται ένας μόνο τύπος προϊόντος.

Η λειτουργία του συστήματος καθορίζεται από την εμφάνιση γεγονότων που περιεγράφηκαν στην Παράγραφο 3.3 τα οποία έχουν τις ακόλουθες συνέπειες:

Όταν φθάνει μια παραγγελία μπαίνει σε ουρά αναμονής και το εργοστάσιο ενημερώνεται για τις υποχρεώσεις του. Αν υπάρχει έλλειψη και η μηχανή είναι αδρανής, τότε ξεκινάει την παραγωγή ενός τύπου προϊόντος με τη συγκεκριμένη πολιτική. Επιλέγεται το προϊόν το οποίο είναι πιο μακριά από το στόχο ή πιο απλά με την μικρότερη διαθέσιμη ποσότητα.

Όταν υπάρχουν πελάτες (παραγγελίες) σε αναμονή, τότε **οι πρώτοι πελάτες στην ουρά** είναι εκείνοι που είχαν παραγγείλει νωρίτερα και **έχουν προτεραιότητα**. Όταν για αυτούς τους πελάτες υπάρχουν επαρκείς διαθέσιμες ποσότητες σε απόθεμα για κάθε τύπο προϊόντος που ζητούν, τότε εξυπηρετούνται και αποχωρούν από το σύστημα.

Στο μοντέλο που εξετάζουμε υπάρχουν δύο κατηγορίες γεγονότων. Η πρώτη αναφέρεται στους πελάτες και η δεύτερη στις μηχανές. Η καθεμιά από τις προηγούμενες κατηγορίες χωρίζεται σε υποκατηγορίες.

Στην κατηγορία με τους πελάτες έχουμε τις παρακάτω υποκατηγορίες:

- 1) άφιξη πελατών και
- 2) αναχώρηση πελατών.

Για τις κατηγορίες με τις μηχανές έχουμε τις υποκατηγορίες:

- 1) προετοιμασία μηχανής (set up)
- 2) η μηχανή ξεκινάει την παραγωγή ενός συγκεκριμένου τύπου προϊόντος,
- 3) η μηχανή τελειώνει την παραγωγή ενός συγκεκριμένου τύπου προϊόντος.

Κάθε ένα από αυτά τα γεγονότα έχει έναν **επικείμενο χρόνο εμφάνισης** (σύντομο χρονικό διάστημα). Ο αλγόριθμος βρίσκει ποιο γεγονός προηγείται χρονικά και το εκτελεί. Το πρόγραμμα ολοκληρώνεται όταν εξαντληθεί ο χρονικός ορίζοντας της προσομοίωσης που συμβολίζεται TSIM.

### 5.2.1 Αναλυτική περιγραφή γεγονότων

Η πολιτική παραγωγής παρτίδων έχει τις ακόλουθες παραμέτρους ελέγχου:

- Μέγιστος αριθμός πελατών που μπορεί να έχουν προτεραιότητα: **PRIORITY**
- Απόθεμα ασφαλείας προϊόντος  $i$ :  **$I\_high(i)$**
- Στάθμη συναγερμού αποθέματος προϊόντος  $i$ :  **$I\_low(i)$**

Η πολιτική αυτή όταν έχουμε μόνο ένα είδος προϊόντος είναι διεθνώς γνωστή με τον συμβολισμό ( $s, S$ ). Εδώ χρησιμοποιούμε διαφορετικό συμβολισμό όπου  $s = I\_low$  και  $S = I\_high$ . Εφαρμόζεται πολύ συχνά για τον έλεγχο αποθεμάτων όταν η ζήτηση δεν είναι σταθερή.

**Άφιξη πελάτη:** Μόλις φτάσει νέα παραγγελία, εξετάζεται αν θα εκτελεστεί κατά προτεραιότητα. Έχουμε ορίσει οι χρόνοι ανάμεσα σε διαδοχικές αφίξεις των πελατών να ακολουθούν την ομοιόμορφη κατανομή σε ένα διάστημα τιμών ( $min, max$ ). Ο χρήστης πρέπει να δώσει τις τιμές αυτές σε ένα αρχείο δεδομένων και μπορεί να τις μεταβάλει ώστε να αναπαραστήσει διαφορετικά εποχικά σενάρια ζήτησης (πχ. το καλοκαίρι οι χρόνοι  $min$  και  $max$  είναι σύντομοι). Αν μια παραγγελία είναι σε προτεραιότητα, γίνεται έλεγχος με τον οποίο τσεκάρουμε την επάρκεια κάθε τύπου προϊόντος της παραγγελίας. Σε μια παραγγελία δεχόμαστε μία πιθανότητα ενδεικτικά 0.05 ή ποσοστό 5% ώστε να **μην περιλαμβάνεται** κάποιος τύπος προϊόντος. Στη συγκεκριμένη εφαρμογή προσομοιώνουμε το σύστημα με 4 τύπους προϊόντων. Αν υπάρχει επάρκεια όλων των τύπων προϊόντων σε μια παραγγελία με προτεραιότητα τότε η παραγγελία εκτελείται άμεσα και βγαίνει από το σύστημα αφαιρώντας τις ποσότητες κάθε τύπου που περιέχει από το συνολικό απόθεμα κάθε τύπου προϊόντος. Σε διαφορετική περίπτωση μπαίνει στην ουρά με ενημέρωση και του χρόνου

καθυστερήσης της παραγγελίας. Επεξηγηματικά σε μια μεταβλητή κρατάμε τον συνολικό χρόνο καθυστέρησης όλων των εκκρεμών παραγγελιών, αφαιρώντας τον χρόνο στον οποίο μπαίνει η παραγγελία στο σύστημα. Αν μια παραγγελία μπει μέσα στην ουρά τότε θα εκτελεστεί όταν θα έρθει η σειρά της και εξαρτάται από τον χρόνο άφιξης της και του τρέχοντος χρόνου. Σε περίπτωση που βρεθεί μια παραγγελία που πρέπει να εκτελεστεί, αλλά δεν επαρκούν οι ποσότητες που απαιτεί τότε η μηχανή θα πρέπει να ξεκινήσει την παραγωγή του τύπου προϊόντος που είναι στο μικρότερο απόθεμα, βάσει της ποσότητας που υπάρχει εκείνη την στιγμή αλλά και των συνολικών απαιτήσεων στις εκκρεμείς παραγγελίες μέχρι τότε και έχει να κάνει με το μέγιστο απόθεμα που υπάρχει.

**Αναχώρηση πελάτη:** Εφόσον η παραγγελία του συγκεκριμένου πελάτη ικανοποιείται τότε φεύγει από το σύστημα. Σε αυτό το στάδιο αφαιρούνται οι ποσότητες που ζήτησε από το διαθέσιμο συνολικό απόθεμα κάθε τύπου και προστίθεται ο χρόνος όπου ολοκληρώνεται η παραγγελία στη μεταβλητή που αποθηκεύει την συνολική καθυστέρηση του συστήματος.

**Προετοιμασία μηχανής:** Είναι η έναρξη ενός χρονικού διαστήματος που απαιτείται έτσι ώστε η μηχανή να προετοιμαστεί για την παραγωγή ενός νέου τύπου προϊόντος. Αυτή η διαδικασία γίνεται σε περίπτωση που θέλει να παράγει ξανά τον ίδιο τύπο που παρήγαγε την τελευταία φορά λειτουργίας της και χωρίς να χρειάζεται εκ νέου προετοιμασία.

**Έναρξη παραγωγής μίας παρτίδας προϊόντος:** Είναι η χρονική στιγμή που ξεκινάει η παραγωγή ποσότητας ενός συγκεκριμένου τύπου προϊόντος που καθορίζεται από τον ρυθμό παραγωγής της μηχανής και διαβάζεται σαν παράμετρος εισόδου στην αρχή του προγράμματος. Επίσης οι μέγιστες τιμές αποθέματος (απόθεμα ασφαλείας), αλλά και οι ελάχιστες (στάθμη συναγερμού), που έχουν αποθηκευτεί ως παράμετροι του συστήματος, καθορίζουν πώς θα εκτελεστεί το πρόγραμμα σε σχέση με την διαδοχική σειρά εκτέλεσης των γεγονότων.

**Ολοκλήρωση της παραγωγής παρτίδας:** Η χρονική στιγμή στην οποία ολοκληρώνεται η παραγωγή του συγκεκριμένου τύπου προϊόντος με ταυτόχρονη ενημέρωση του νέου αποθέματος.

Σε κάθε εμφάνιση γεγονότος υπολογίζεται το εμβαδόν όλων των αποθεμάτων κάνοντας χρήση του εμβαδού του τραpezίου. Στο τέλος της προσομοίωσης, το πηλίκο εμβαδόν/TSIM δίνει το μέσο απόθεμα κάθε προϊόντος.

## 5.2.2 Μεγέθη για την εξαγωγή συμπερασμάτων

Τα αποτελέσματα που συλλέγουμε για την εξαγωγή συμπερασμάτων στην επόμενη παράγραφο γίνονται με βάση το κόστος ανά μονάδα χρόνου. Για την εκτέλεση πειραμάτων υποθέτουμε ότι το κόστος ανά ώρα δίνεται από τον τύπο

$$\text{Κόστος/ώρα} = 1 \times (\text{μέσο συνολικό απόθεμα}) + 10 \times (\text{μέση καθυστέρηση παραγγελίας})$$

Οι συντελεστές 1 και 10 είναι υποκειμενικοί διότι δεν υπήρχε υπόδειξη από την εταιρεία. Βασίζονται μία παραδοχή ότι μια αύξηση της καθυστέρησης πελατών κατά 1 ώρα είναι πολύ μεγαλύτερη βαρύτητας σε σχέση με την αύξηση του μέσου αποθέματος κατά 1 παλέτα που είναι η μονάδα μέτρησης προϊόντων.

Το μέσο απόθεμα για κάθε τύπο προϊόντος υπολογίζεται από το εμβαδόν αποθέματος σύμφωνα με τον τύπο της προηγούμενης παραγράφου.

Τέλος ο υπολογισμός της μέσης συνολικής καθυστέρησης γίνεται με βάση το άθροισμα των χρόνων καθυστέρησης των παραγγελιών διαιρούμενο με το συνολικό πλήθος των παραγγελιών που έχουν εισέλθει στο σύστημα.

## 6 ΥΠΟΛΟΓΙΣΜΟΙ ΠΑΡΑΜΕΤΡΩΝ

### 6.1 ΕΙΣΑΓΩΓΗ

Για την συλλογή των αποτελεσμάτων βασιστήκαμε στην εκτέλεση του προγράμματος σε περιβάλλον MATLAB, ο κώδικας επισυνάπτεται στο τέλος της εργασίας, και μάλιστα με διαδοχική εκτέλεση πολλών επαναλήψεων στις οποίες όπως αλλάζουμε κάποιες παραμέτρους δίνοντας τους τιμές που χρησιμοποιούνται στην πράξη.

Οι παράμετροι τους οποίους μεταβάλαμε είναι οι ακόλουθοι:

- Μέγιστος αριθμός πελατών που μπορεί να έχουν προτεραιότητα: **PRIORITY**
- Απόθεμα ασφαλείας προϊόντος  $i$ :  **$l\_high(i)$**
- Στάθμη συναγερμού αποθέματος προϊόντος  $i$ :  **$l\_low(i)$**

Για την εύρεση της βέλτιστης πολιτικής παραγωγής δοκιμάσαμε τιμές **PRIORITY** = 1, 2 και 3. Επίσης αλλάζουμε τις μέγιστες και ελάχιστες τιμές των αποθεμάτων ( **$l\_high$** ,  **$l\_low$** ). Ξεκινάμε τις εκτελέσεις αλλάζοντας τα άνω όρια αποθέματος από μια τιμή σε μια μέγιστη αλλά και ελάχιστα από 0 μέχρι την μέγιστη τιμή μείον κάποιο βήμα (step).

### 6.2 ΑΡΧΙΚΗ ΕΚΤΙΜΗΣΗ ΠΑΡΑΜΕΤΡΩΝ

Ευσταθές είναι ένα σύστημα παραγωγής στο οποίο οι μέσοι χρόνοι καθυστέρησης στη διεκπεραίωση παραγγελιών είναι φραγμένες ποσότητες, δηλαδή δεν αυξάνονται με τον χρόνο. Όπως είδαμε στην Παράγραφο 3.4, αναγκαία συνθήκη ευστάθειας του συστήματος είναι:

$$\rho = \frac{D_1}{P_1} + \frac{D_2}{P_2} + \frac{D_3}{P_3} + \frac{D_4}{P_4} < 1,$$

όπου  $D_i$  είναι μέσος ρυθμός ζήτησης προϊόντος  $i$  και  $P_i$  είναι η δυναμικότητα της μηχανής για το προϊόν  $i$ , δηλαδή μέσος ρυθμός παραγωγής αν το σύστημα παρήγαγε το  $i$  συνεχώς χωρίς διακοπές. Υπολογίζουμε παρακάτω τα  $D_i$  και  $P_i$ . Ο ημερήσιος αριθμός παραγγελιών είναι από 20 έως 40.

Άρα : (μέσες παραγγελίες/ημέρα) = 30

Στο εξής μία ημέρα θεωρείται ως το διάστημα 8 ωρών παραγωγής διότι το εργοστάσιο λειτουργεί μία οκτάωρη βάρδια. Ο μέσος χρόνος διαδοχικών αφίξεων θα υπολογιστεί ως ο μέσος όρος ελάχιστων και μέγιστων χρόνων. Ο μέγιστος χρόνος μεταξύ αφίξεων είναι  $(8 \text{ ώρες})/(\text{λιγότερες αφίξεις}) = 8/20 = 0.4$  και ο ελάχιστος χρόνος  $(8 \text{ ώρες})/(\text{περισσότερες αφίξεις}) = 8/40 = 0.2$ . Ο μέσος αριθμός παραγγελιών ανά ώρα είναι το αντίστροφο του μέσου χρόνου ενδοαφίξεων:

(μέσος αριθμός παραγγελιών ανά ώρα) =  $2/(0.2+0.4) = 3.33$ .

Για κάθε προϊόν  $i$  υπολογίζουμε:  $D_i = (\text{μέσος αριθμός παραγγελιών ανά ώρα}) \times (\text{μέση}$

ζήτηση  $i$  ανά παραγγελία).

Από τα δεδομένα της επιχείρησης, έχουμε τους ακόλουθους ελάχιστους και μέγιστους αριθμούς παλετών ανά ημέρα κατά την καλοκαιρινή περίοδο.

**Πίνακας 1:** Δεδομένες τιμές ζήτησης της εταιρείας κατά την καλοκαιρινή περίοδο

Προϊόντα	Min παλέτες	Max παλέτες
Γκαζόζα	2	8
Λεμονάδα	2	6
Πορτοκαλάδα	2	6
Ice Tea	1	3

Η μέση ζήτηση ανά παραγγελία για κάθε προϊόν είναι:

- Γκαζόζα: μέση ζήτηση είναι  $\frac{2+8}{2} = 5$  παλέτες σε 30 (μέσες παραγγελίες) άρα σε μια παραγγελία θα είναι  $x = \frac{5}{30}$
- Λεμονάδα: μέση ζήτηση είναι  $\frac{2+6}{2} = 4$  παλέτες σε 30 (μέσες παραγγελίες) άρα σε μια παραγγελία θα είναι  $x = \frac{4}{30}$
- Πορτοκαλάδα: μέση ζήτηση είναι  $\frac{2+6}{2} = 4$  παλέτες σε 30 (μέσες παραγγελίες) άρα σε μια παραγγελία θα είναι  $x = \frac{4}{30}$
- Ice Tea: μέση ζήτηση είναι  $\frac{1+3}{2} = 2$  παλέτες σε 30 (μέσες παραγγελίες) άρα σε μια παραγγελία θα είναι  $x = \frac{2}{30}$

Άρα

$$D_1 = 3.33 \times \frac{5}{30} = 0.555$$

$$D_2 = 3.33 \times \frac{4}{30} = 0.444$$

$$D_3 = 3.33 \times \frac{4}{30} = 0.444$$

$$D_4 = 3.33 \times \frac{2}{30} = 0.222$$

Οι ρυθμοί παραγωγής όλων των προϊόντων είναι ίδιοι. Από τα δεδομένα έχουμε ότι το σύστημα παράγει μείγμα που αντιστοιχεί σε 10000 μπουκάλια την ώρα και μία παλέτα περιέχει 1584 μπουκάλια. Επομένως οι ρυθμοί παραγωγής σε παλέτες ανά ώρα είναι

$$P_i = 10000/1584 = 6.3.$$

Με αντικατάσταση στη συνθήκη έχουμε

$$\rho = \frac{D_1}{P_1} + \frac{D_2}{P_2} + \frac{D_3}{P_3} + \frac{D_4}{P_4} = \frac{0.555}{6.3} + \frac{0.444}{6.3} + \frac{0.444}{6.3} + \frac{0.222}{6.3} = 0.264 < 1$$

που σημαίνει ότι το σύστημα είναι ευσταθές.

Στις σημειώσεις του μαθήματος Συστήματα Παραγωγής (2021), [7], εξετάζεται ίδιο πρόβλημα με κόστος προετοιμασίας  $A_i$ , κόστος αποθέματος ανά μονάδα προϊόντος και μονάδα χρόνου  $h_i$ , κόστος παραγωγής μονάδας προϊόντος  $C_i$  και μέγεθος παρτίδας  $Q_i$ . Σύμφωνα με τους τύπους της Παραγράφου 3.4, το κόστος ανά μονάδα χρόνου δίνεται από την σχέση:

$$K = \sum_{i=1}^4 \left[ \frac{A_i D_i}{Q_i} + C_i D_i + h_i \frac{T D_i}{2} (1 - \rho_i) \right], \rho_i = \frac{D_i}{P_i}$$

όπου  $T$  είναι ο κύκλος παραγωγής όλων των προϊόντων από μία φορά, ώστε η παρτίδα που παράγεται να ικανοποιεί όλη τη ζήτηση κάθε  $i$  στο ίδιο διάστημα, δηλαδή  $Q_i = T D_i$ . Η βέλτιστη λύση  $T^*$  που ελαχιστοποιεί  $K$  υπολογίζεται:

$$T^* = \sqrt{\frac{2 \sum_{i=1}^4 A_i}{\sum_{i=1}^4 h_i D_i (1 - \rho_i)}}$$

ελέγχοντας αν ικανοποιεί την σχέση:

$$T^* - T^* \rho \geq (S_1 + S_2 + S_3 + S_4)$$

όπου  $S_i$  οι χρόνοι προετοιμασίας για κάθε παρτίδα. Κάνοντας πράξεις καταλήγουμε στην

$$T^* \geq \frac{S_1 + S_2 + S_3 + S_4}{1 - \rho} = T_{\min}$$

Ο χρόνος προετοιμασίας της μηχανής για κάθε τύπο προϊόντος είναι μισή ώρα, άρα  $S_i = 0.5$

$$T_{\text{opt}} = T_{\min} = \frac{\sum S_i}{1 - \sum \rho_i} = \frac{\sum S_i}{1 - \left( \frac{D_1}{P_1} + \frac{D_2}{P_2} + \frac{D_3}{P_3} + \frac{D_4}{P_4} \right)}$$

**Πίνακας 2:** Τιμές παραμέτρων για κάθε προϊόν

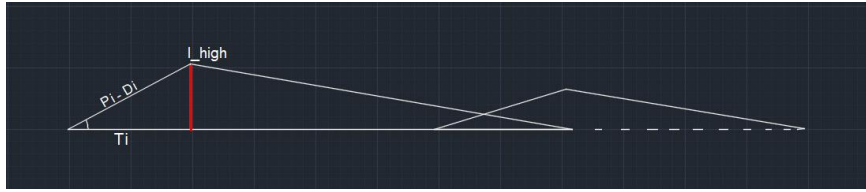
Παράμετροι	Γκαζόζα	Λεμονάδα	Πορτοκαλάδα	Ice Tea
$D_i$	0.555	0.444	0.444	0.222
$P_i$	6.3	6.3	6.3	6.3
$S_i$	0.5	0.5	0.5	0.5

Έτσι

$$T_{\text{opt}} = \frac{4 \cdot 0.5}{1 - \left( \frac{0.555}{6.3} + \frac{0.444}{6.3} + \frac{0.444}{6.3} + \frac{0.222}{6.3} \right)} = \frac{2}{1 - 0.264} = 2.717$$

Παρακάτω παρουσιάζεται γεωμετρική εύρεση της ελάχιστης τιμής του μέγιστου αποθέματος  $I_{\text{high}}$ .





**Γράφημα 3:**Απεικόνιση της εύρεσης της τιμής  $l\_high$

Το μέγεθος παρτίδας του  $i$  που παράγεται σε έναν κύκλο είναι  $T_i P_i = Q_i$  άρα ο χρόνος που χρειάζεται για να παραχθεί είναι  $T_i = Q_i / P_i$ . Επίσης η κλίση  $P_i - D_i$  ισούται με την εφαπτομένη της αριστερής γωνίας ή  $P_i - D_i = l\_high / T_i$ . Συνεπώς

$$l\_high = T_i (P_i - D_i) = \frac{Q_i}{P_i} (P_i - D_i)$$

Άρα τα βέλτιστα  $l\_high$  όταν η ζήτηση είναι σταθερή είναι:

- Γκαζόζα  $l\_high = T_1 \times (P_1 - D_1) = 2.717 \times (6.3 - 0.555) = \mathbf{15.60}$
- Λεμονάδα  $l\_high = T_1 \times (P_1 - D_1) = 2.717 \times (6.3 - 0.444) = \mathbf{15.91}$
- Πορτ./δα  $l\_high = T_1 \times (P_1 - D_1) = 2.717 \times (6.3 - 0.444) = \mathbf{15.91}$
- Iced Tea  $l\_high = T_1 \times (P_1 - D_1) = 2.717 \times (6.3 - 0.222) = \mathbf{16.51}$

Όταν η ζήτηση μεταβάλλεται, αυτά μπορούν να ληφθούν ως **τα κάτω όρια** του  $l\_high$ . Θα δοκιμάσουμε αρχικά πολλαπλάσια αυτών των τιμών:

$$\begin{aligned} l\_high1 &= 15.60 \text{ μέχρι } 3 \times 15.60 = 46.80 \\ l\_high2 &= 15.91 \text{ μέχρι } 3 \times 15.91 = 47.73 \\ l\_high3 &= 15.91 \text{ μέχρι } 3 \times 15.91 = 47.73 \\ l\_high4 &= 16.51 \text{ μέχρι } 3 \times 16.51 = 49.53 \end{aligned}$$

Συνεπώς:

$$\begin{aligned} l\_low1 &= 0 \text{ μέχρι } l\_high1 - 15.60 \text{ με βήμα } 15.60 \\ l\_low2 &= 0 \text{ μέχρι } l\_high2 - 15.91 \text{ με βήμα } 15.91 \\ l\_low3 &= 0 \text{ μέχρι } l\_high3 - 15.91 \text{ με βήμα } 15.91 \\ l\_low4 &= 0 \text{ μέχρι } l\_high3 - 16.51 \text{ με βήμα } 16.51 \end{aligned}$$

Στο επόμενο κεφάλαιο δοκιμάζουμε συνδυασμούς με όλες αυτές τις τιμές. Για κάθε συνδυασμό προσομοιώνουμε το σύστημα και επιλέγουμε εκείνον που εμφανίζει το μικρότερο κόστος.

## 7 ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ

Σε αυτή την ενότητα περιγράφουμε τα αποτελέσματα μιας σειράς αριθμητικών πειραμάτων. Τα πειράματα αυτά αποδεικνύουν ότι το μοντέλο συμπεριφέρεται λογικά, δηλαδή όπως θα περίμενε κανείς να συμπεριφέρεται μια πραγματική γραμμή παραγωγής καθώς μεταβάλλονται οι παράμετροί της.

### 7.1 ΕΠΙΛΟΓΗ ΧΡΟΝΙΚΟΥ ΟΡΙΖΟΝΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ

Είναι σημαντικό στα πειράματα να επιλέγουμε έναν χρονικό ορίζοντα προσομοίωσης TSIM τέτοιο ώστε τα αποτελέσματα να είναι κοντά στις μέσες τιμές της μόνιμης κατάστασης λειτουργίας του συστήματος αλλά και να προκύπτουν από σχετικά σύντομους χρόνους εκτέλεσης προσομοιώσεων στον υπολογιστή. Εξετάστηκαν διάφορες τιμές TSIM = 10000, 20000, 40000, 60000. Εκτελέσαμε δύο συνδυασμούς μέγιστων αποθεμάτων,  $l\_high$  για τα τέσσερα προϊόντα. Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα. Όλες οι μονάδες χρόνου είναι ώρες.

Πίνακας 3: Τιμές εύρεσης καλύτερου χρόνου εκτέλεσης κώδικα		
Χρονικός ορίζοντας	$l\_high = 15.60, 15.91, 15.91, 16.91$	$l\_High = 13.60, 13.91, 13.91, 13.91$
TSIM	Μέση καθυστέρηση παραγγελιών	Μέση καθυστέρηση παραγγελιών
10000	0.1004	0.0897
20000	0.0982	0.0885
40000	0.0995	0.0892
60000	0.1006	0.0898

Παρατηρούμε ότι για μέγιστο ορίζοντα προσομοίωσης, TSIM = 60000, η μέση καθυστέρηση παραγγελιών φαίνεται να ομαλοποιείται γύρω από μία τιμή. Για μικρότερο χρονικό ορίζοντα, TSIM = 20000, έχουμε μικρότερους χρόνους εκτέλεσης προσομοιώσεων στον υπολογιστή αλλά όμοια αποτελέσματα. Πράγματι οι αποκλίσεις στην εκτίμηση της μέσης καθυστέρησης είναι  $(0.1006 - 0.0982)/0.1006 \approx 2.4\%$  και  $(0.0898 - 0.0885)/0.0898 \approx 1.4\%$ . Επομένως η επιλογή TSIM = 20000 εγγυάται ακρίβεια στην εκτίμηση της μόνιμης κατάστασης.

### 7.2 ΒΕΛΤΙΣΤΕΣ ΠΑΡΑΜΕΤΡΟΙ ΕΛΕΓΧΟΥ ΑΠΟΘΕΜΑΤΩΝ

Ακολούθως έγινε αναζήτηση των βέλτιστων τιμών των παραμέτρων των αποθεμάτων:

- μέγιστα αποθέματα,  $l\_high(i)$ ,  $i = 1, \dots, 4$
- στάθμες χαμηλών αποθεμάτων,  $l\_low(i)$ ,  $i = 1, \dots, 4$
- πλήθος ενωρίτερων παραγγελιών οι οποίες μπορούν να εξυπηρετηθούν από το διαθέσιμο απόθεμα, PRIORITY.

Για κάθε παράμετρο επιλέγονται διάφορες τιμές από μία ελάχιστη έως μία μέγιστη με δεδομένο βήμα και δοκιμάζονται όλοι οι δυνατοί συνδυασμοί παραμέτρων. Η επιλογή των μέγιστων και ελάχιστων τιμών των αποθεμάτων  $l\_high$  έγινε με μαθηματικούς

υπολογισμούς που παρουσιάστηκαν στην Παράγραφο 6.2. Οι τιμές που επιλέχθηκαν είναι οι εξής: Για  $l\_high$  οι ελάχιστες τιμές των τεσσάρων προϊόντων είναι

$$15.60, 15.91, 15.91, 16.51$$

τα βήματα είναι ίσα με τις ελάχιστες τιμές, δηλαδή 15.60, 15.91, 15.91, 16.51 και οι μέγιστες τιμές των  $l\_high$  είναι **τριπλάσιες** των ανωτέρω. Για τα  $l\_low$  επιλέγονται ελάχιστες τιμές 0 και για τα τέσσερα προϊόντα, ίδια βήματα, και μέγιστες τιμές

$$l\_low \leq l\_high - \text{βήμα}$$

όπου, ανάλογα με το προϊόν, βήμα = 15.60, 15.91, 15.91, 16.51.

Τελικά, δοκιμάστηκαν οι συνδυασμοί:

PRIORITY	$l\_high(1)$	$l\_high(2)$	$l\_high(3)$	$l\_high(4)$	$l\_low(1)$	$l\_low(2)$	$l\_low(3)$	$l\_low(4)$
1	15.60	15.91	15.91	16.51	0	0	0	0
2	15.60	15.91	15.91	16.51	0	0	0	0
...								
3	46.80	47.73	47.73	49.53	31.20	31.82	31.82	33.02

Ενδεικτικά αποτελέσματα μετά την εκτέλεση του προγράμματος φαίνονται στον επόμενο πίνακα, ο οποίος δείχνει το συνολικό κόστος των διαφόρων συνδυασμών για  $PRIORITY = 2$ , που είναι και η τιμή με την οποία το σύστημα έχει χαμηλότερο κόστος λειτουργίας ανά ώρα.

**Πίνακας 4:** Ενδεικτικά αποτελέσματα προσομοιώσεων για  $PRIORITY = 2$

$l\_high(1)$	$l\_high(2)$	$l\_high(3)$	$l\_high(4)$	$l\_low(1)$	$l\_low(2)$	$l\_low(3)$	$l\_low(4)$	Κόστος/ώρα
<b>15.60</b>	<b>15.91</b>	<b>15.91</b>	<b>16.51</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>32.0239 (βέλτιστο)</b>
15.60	15.91	15.91	33.02	0.00	0.00	0.00	0.00	40.3407
15.60	15.91	15.91	33.02	0.00	0.00	0.00	16.51	48.5138
15.60	15.91	15.91	49.53	0.00	0.00	0.00	0.00	48.6314
31.20	47.73	31.82	16.51	0.00	0.00	0.00	0.00	63.5425
31.20	47.73	31.82	16.51	15.60	31.82	0.00	0.00	86.8726
31.20	47.73	47.73	49.53	0.00	15.91	0.00	33.02	112.0681
46.80	15.91	15.91	16.51	0.00	0.00	0.00	0.00	47.9051
46.80	15.91	15.91	16.51	15.60	0.00	0.00	0.00	55.3467
46.80	31.82	15.91	16.51	15.60	15.91	0.00	0.00	70.9508
46.80	47.73	15.91	16.51	31.20	31.82	0.00	0.00	94.6428
46.80	47.73	15.91	33.02	0.00	0.00	0.00	0.00	71.8242
31.20	47.73	15.91	33.02	15.60	0.00	0.00	0.00	79.2418
46.80	47.73	47.73	49.53	31.20	15.91	15.91	33.02	143.1425
46.80	47.73	47.73	49.53	31.20	15.91	31.82	0.00	134.5090
46.80	47.73	47.73	49.53	31.20	15.91	31.82	33.02	151.1374

Την καλύτερη λύση σε αυτή την εκτέλεση φαίνεται ότι έχουν οι συνδυασμοί προϊόντων για  $l\_high = [15.60, 15.91, 15.91, 16.51]$  και  $l\_low = [0.00, 0.00, 0.00, 0.00]$  με βέλτιστο κόστος 32.0239 ανά μονάδα χρόνου.

Για πιο λεπτομερή αναζήτηση γύρω από το αρχικό βέλτιστο μειώνουμε το βήμα του

$l\_high$  και το κάνουμε ίσο με 1 δοκιμάζοντας 3 διαφορετικές τιμές  $l\_high$  για κάθε προϊόν, όλες με  $PRIORITY = 2$ . Άρα οι συνδυασμοί τιμών  $l\_high$  περιλαμβάνουν εκείνον με τις ελάχιστες τιμές [14.60, 14.91, 14.91, 15.51] και καλύπτουν όλο το πλέγμα μέχρι τον συνδυασμό με τις μέγιστες [16.60, 16.91, 16.91, 17.51]. Στην προηγούμενη αναζήτηση τα βέλτιστα  $l\_low$  βρέθηκαν όλα 0 αλλά το βήμα ήταν μεγάλο. Τώρα δοκιμάζουμε τιμές  $l\_low = 0, 1$  και 2 για κάθε προϊόν και με όλους τους συνδυασμούς. Εκτελώντας τον κώδικα προκύπτουν αποτελέσματα εκ των οποίων κάποια φαίνονται στον παρακάτω πίνακα.

**Πίνακας 5:** Ενδεικτικά αποτελέσματα προσομοιώσεων από πιο λεπτομερή αναζήτηση

$l\_high(1)$	$l\_high(2)$	$l\_high(3)$	$l\_high(4)$	$l\_low(1)$	$l\_low(2)$	$l\_low(3)$	$l\_low(4)$	Κόστος/ώρα
<b>14.60</b>	<b>14.91</b>	<b>14.91</b>	<b>15.51</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>30.0819 (βέλτιστο)</b>
14.60	14.91	14.91	16.51	2.00	2.00	1.00	1.00	33.2370
14.60	15.91	15.91	17.51	1.00	0.00	2.00	1.00	33.5261
15.60	14.91	14.91	15.51	0.00	0.00	0.00	0.00	30.5826
15.60	14.91	15.91	15.51	1.00	0.00	1.00	1.00	32.0170
14.60	16.91	15.91	17.51	2.00	1.00	1.00	0.00	33.8716
15.60	14.91	15.91	17.51	1.00	0.00	1.00	1.00	33.0640
15.60	15.91	14.91	15.51	0.00	2.00	0.00	2.00	32.6846
14.60	16.91	15.91	17.51	1.00	2.00	2.00	1.00	34.7892
14.60	16.91	16.91	16.51	2.00	0.00	1.00	1.00	33.8837
15.60	14.91	14.91	15.51	2.00	2.00	2.00	0.00	32.8737
15.60	14.91	15.91	17.51	2.00	2.00	0.00	2.00	34.3813
15.60	15.91	14.91	15.51	0.00	0.00	1.00	1.00	31.7649
15.60	15.91	15.91	15.51	1.00	2.00	0.00	0.00	32.5399
15.60	16.91	14.91	17.51	0.00	1.00	0.00	0.00	32.9639
15.60	16.91	15.91	15.51	1.00	0.00	1.00	1.00	32.9526

Καλύτερη λύση σε αυτή την εκτέλεση φαίνεται ότι έχουν οι συνδυασμοί προϊόντων για  $l\_high = [15.60, 14.91, 14.91, 15.51]$  και  $l\_low = [0.00, 0.00, 0.00, 0.00]$  με κόστος 30.081999 ανά μονάδα χρόνου.

Η μείωση του κόστους σε σχέση με την προηγούμενη δοκιμή είναι :

$$(32.0239 - 30.081999) / 32.0239 \approx 6\%$$

Ενδεχομένως μία αναζήτηση με ακόμη μικρότερο βήμα να κατέληγε σε κάποια καλύτερη πολιτική όμως το περιθώριο μείωσης κόστους θα ήταν πολύ μικρότερο, εκτός εάν η συνάρτηση κόστους έχει μεγάλες αυξομειώσεις και το ολικό ελάχιστο ευρίσκεται σε μη κυρτή περιοχή. Από τα πειράματα που έγιναν προκύπτει ότι η συνάρτηση αυτή έχει κυρτή μορφή, παρόλο που δεν υπάρχει απόδειξη για κάτι τέτοιο.

Στο εξής θα εξετάσουμε κάποιες περιπτώσεις γύρο από την καλύτερη λύση της εκτέλεσης του κώδικα, μεταβάλλοντας τις τιμές του πρώτου προϊόντος με διαφορετικά σενάρια εκτέλεσης.

### 7.3 ΕΠΙΔΡΑΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΕΛΕΓΧΟΥ ΣΤΟΥΣ ΔΕΙΚΤΕΣ ΑΠΟΔΟΣΗΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Μεταβάλλοντας την τιμή μίας παραμέτρου και κρατώντας τις άλλες παραμέτρους σταθερές, υπολογίζονται οι αντίστοιχες μεταβολές σε μέτρα απόδοσης όπως η μέση καθυστέρηση διεκπεραίωσης παραγγελιών, το μέσο απόθεμα και το μέσο συνολικό κόστος λειτουργίας του συστήματος ανά ώρα. Οι **σταθερές τιμές** των παραμέτρων είναι:

PRIORITY = 2

I\_high (1) = 15.60      I\_low (1) = 0

I\_high (2) = 15.91      I\_low (2) = 0

I\_high (3) = 15.91      I\_low (3) = 0

I\_high (4) = 16.51      I\_low (4) = 0

Θα εξετάσουμε μεταβολές σε παραμέτρους του προϊόντος 1 μόνον. Επειδή τα προϊόντα έχουν παρόμοιους χρόνους παραγωγής είναι αναμενόμενο οι μεταβολές παραμέτρων των άλλων προϊόντων να έχουν όμοιες επιπτώσεις στα μέτρα απόδοσης του συστήματος.

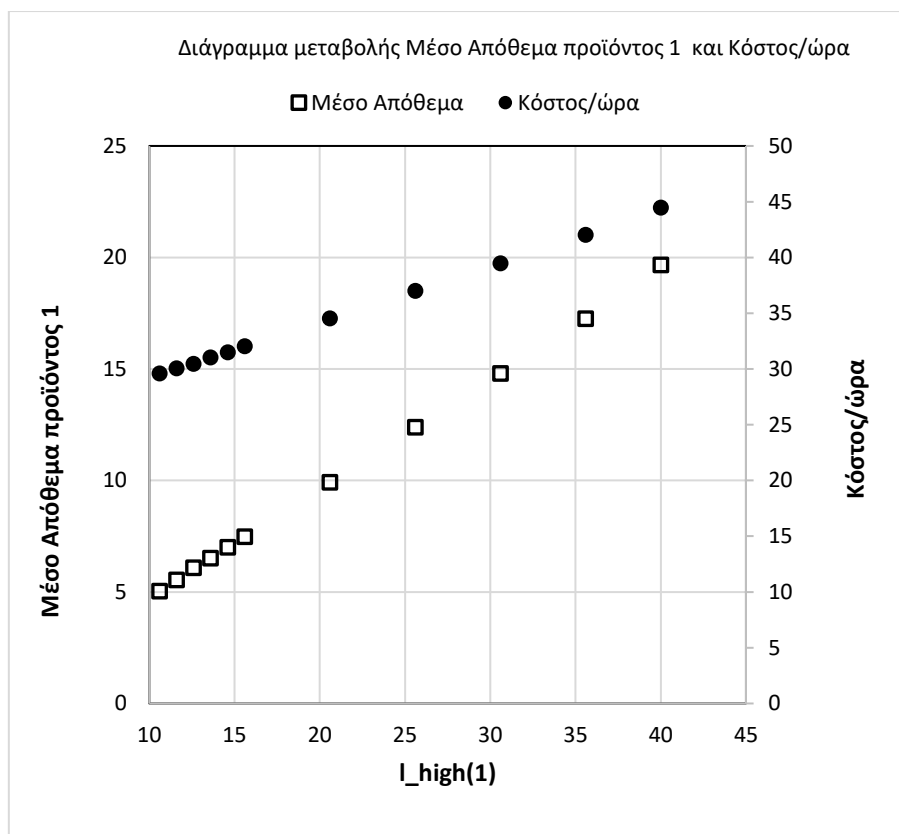
#### 7.3.1 Μέγιστο απόθεμα προϊόντος 1

Για διάφορες τιμές του μέγιστου αποθέματος **I\_high** του πρώτου προϊόντος υπολογίζουμε τις αντίστοιχες μεταβολές στο μέσο απόθεμα του πρώτου προϊόντος (mean\_ stock(1)), τη μέση καθυστέρηση διεκπεραίωσης παραγγελιών (delay) και το μέσο κόστος λειτουργίας του συστήματος ανά ώρα. Δοκιμάζουμε τις ακόλουθες τιμές I\_high(1) = 10.60, 11.60, 12.60, 13.60, 14.60, 15.60, 20.60, 25.60 30.60, 35.60, 40.

**Πίνακας 6:** Αποτελέσματα προσομοίωσης για διάφορες τιμές I\_high

I_high(1)	Μέσο απόθεμα προϊόντος 1	Μέση καθυστέρηση	Κόστος/ώρα
10.6	5.040	0.0961	29.5995
11.6	5.535	0.0901	30.0650
12.6	6.080	0.0696	30.4728
13.6	6.523	0.0830	31.0232
14.6	7.009	0.0868	31.4979
15.6	7.473	0.0966	32.0377
20.6	9.917	0.1116	34.5345
25.6	12.394	0.1030	36.9943
30.6	14.801	0.1103	39.4685
35.6	17.252	0.1228	42.0399
40.00	19.660	0.1355	44.4767

Το επόμενο διάγραμμα δείχνει την εξάρτηση του μέσου αποθέματος του προϊόντος 1 και του κόστους/ώρα από την παράμετρο I\_high(1).



**Γράφημα 4:** Διάγραμμα μεταβολής αποθέματος και κόστος προϊόντος 1

Και οι δύο δείκτες απόδοσης του συστήματος φαίνεται να έχουν ομαλή εξάρτηση από την παράμετρο  $I_{high}$ . Παρατηρούμε ότι το μέσο απόθεμα του προϊόντος 1 αυξάνεται όταν αυξάνεται το απόθεμα ασφαλείας. Αυτό εξηγείται από το γεγονός ότι, με μεγάλο απόθεμα ασφαλείας το εργοστάσιο παράγει μεγαλύτερες ποσότητες του προϊόντος 1 και τις αποθηκεύει. Επομένως σχηματίζονται μεγαλύτερα αποθέματα και συνεπώς υπάρχει μεγαλύτερο κόστος για το συγκεκριμένο προϊόν.

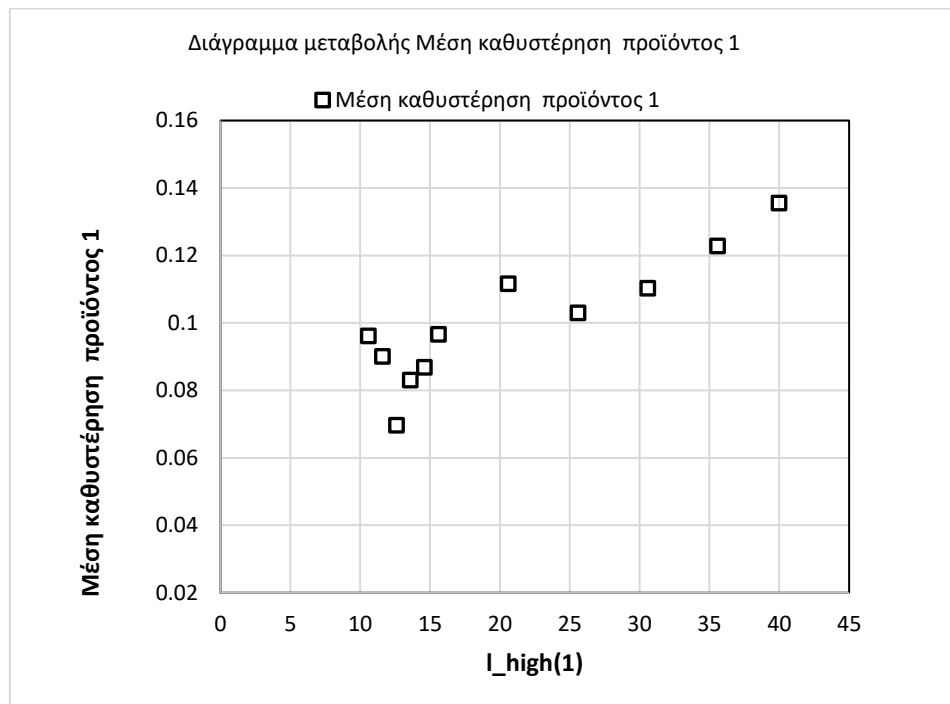
Το επόμενο διάγραμμα δείχνει τη μεταβολή της μέσης καθυστέρησης στη διεκπεραίωση παραγγελιών συναρτήσει του  $I_{high}(1)$ .

Εδώ παρατηρούμε κάποιες αυξομειώσεις στα σημεία  $I_{high} = 12.6$ ,  $20.6$  και  $25.6$  που υποδηλώνουν ότι η μέση καθυστέρηση δεν έχει κυρτή μορφή. Αυτό μπορεί να οφείλεται σε κάποιον από τους ακόλουθους λόγους:

- 1) είτε η συνάρτηση όντως δεν είναι κυρτή
- 2) είτε η διάρκεια προσομοίωσης  $TSIM = 20000$  δεν είναι αρκετή ώστε να έχουμε τη μόνιμη κατάσταση λειτουργίας όπου, πιθανόν, η συνάρτηση αυτή να εμφανίσει πράγματι την κυρτή της μορφή.

Τέτοια διερεύνηση θα μπορούσε ενδεχομένως να γίνει στα πλαίσια άλλης ερευνητικής εργασίας. Η διπλωματική αυτή έχει ως στόχο την ανάπτυξη του μοντέλου και την επίδειξη των δυνατοτήτων του στη βελτιστοποίηση του συστήματος. Το επόμενο

διάγραμμα δείχνει την εξάρτηση της μέσης καθυστέρησης του προϊόντος 1 από την παράμετρο  $I_{high}(1)$ .



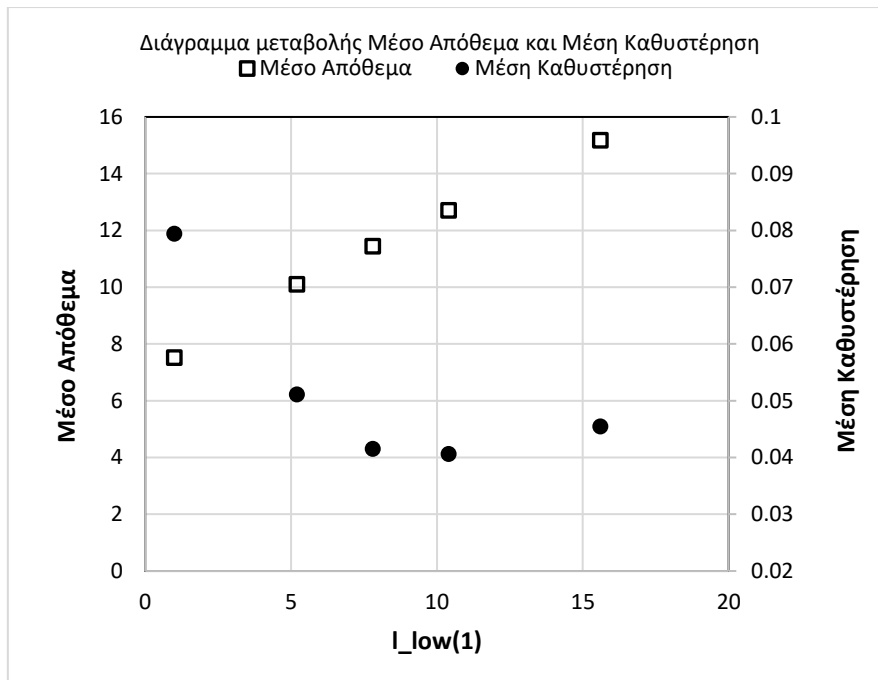
**Γράφημα 5:** Μεταβολή μέσης καθυστέρησης σε συνάρτηση με το απόθεμα προϊόντος 1

### 7.3.2 Στάθμη χαμηλού αποθέματος προϊόντος 1

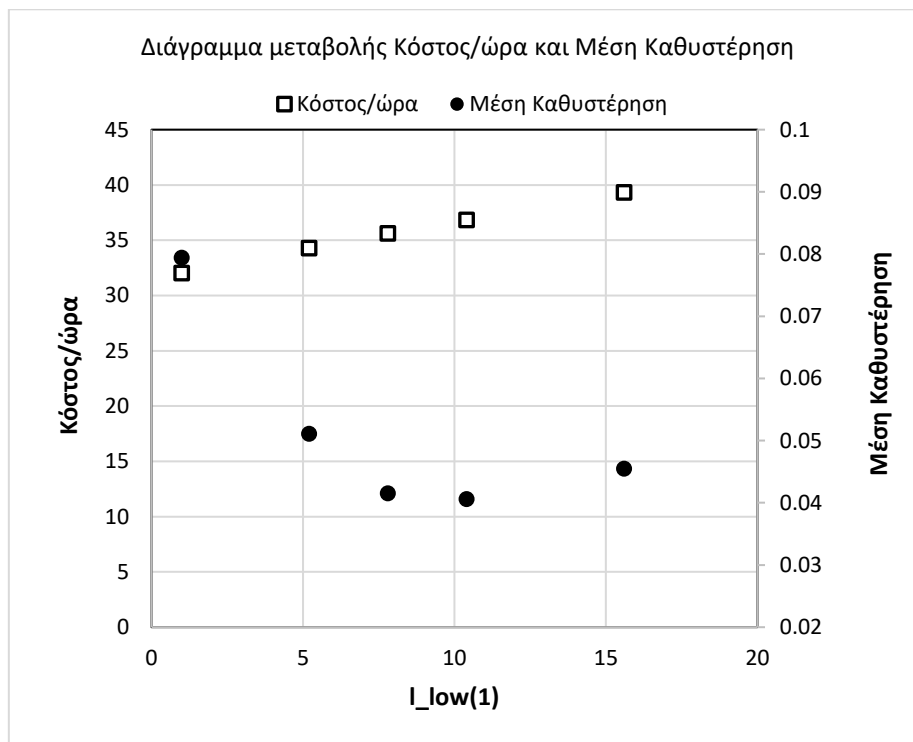
Μεταβάλλουμε τώρα την τιμή  $I_{low}$  του πρώτου προϊόντος διατηρώντας τις υπόλοιπες τιμές των παραμέτρων σταθερές. Τα αποτελέσματα φαίνονται στον πίνακα και τα διαγράμματα που ακολουθούν. Παρατηρούμε ότι το κόστος/ώρα και το μέσο απόθεμα του προϊόντος 1 αυξάνουν καθώς αυξάνεται η στάθμη χαμηλού αποθέματος. Η αύξηση του μέσου αποθέματος οφείλεται στο γεγονός για μεγάλο  $I_{low}(1)$ , ακόμη και μεγάλα αποθέματα θεωρούνται χαμηλά οπότε δίνεται εντολή στο σύστημα να παραγάγει περισσότερα προϊόντα τύπου 1 κι έτσι αυξάνεται το μέσο απόθεμά του.

**Πίνακας 7:** Αποτελέσματα προσομοίωσης για διάφορες τιμές  $I_{low}$

$I_{low}(1)$	Μέσο απόθεμα προϊόντος 1	Μέση καθυστέρηση	Κόστος/ώρα
0	7.522	0.0794	32.0191
5.4	10.106	0.0511	34.2892
7.8	11.447	0.0415	35.6009
10.4	12.710	0.0406	36.8367
15.6	15.183	0.0455	39.3083



Γράφημα 6: Διάγραμμα μεταβολής μέσο απόθεμα και μέσης καθυστέρηση



Γράφημα 7: Διάγραμμα μεταβολής κόστος και μέσης καθυστέρησης

Αντίθετα από τις αυξομειώσεις που εμφάνιζε η μέση καθυστέρηση προηγουμένως, εδώ το μέγεθος αυτό φαίνεται να είναι κυρτή συνάρτηση του  $I_{low}(1)$ .

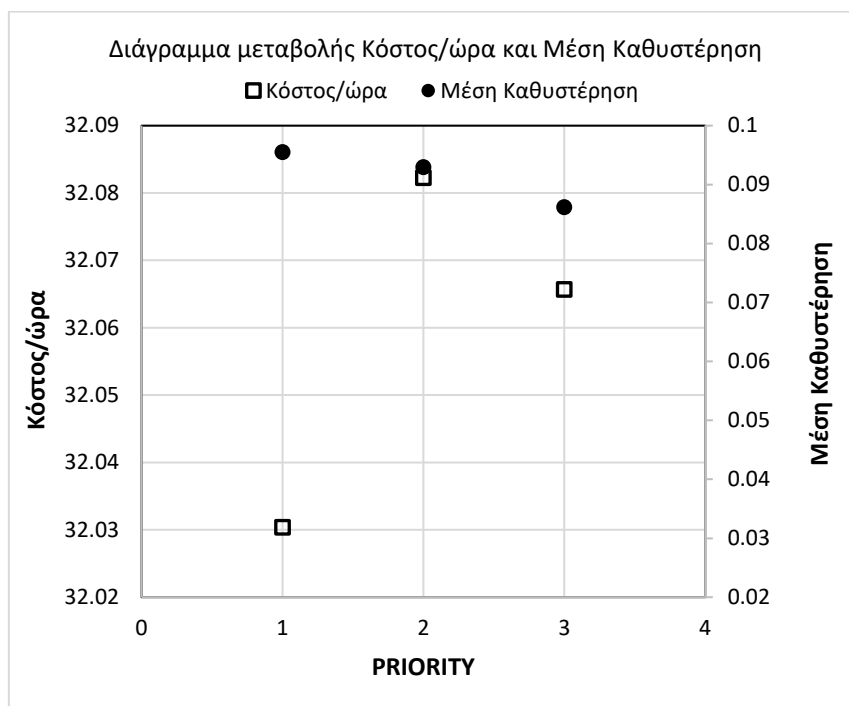


### 7.3.3 Μέγιστος αριθμός παραγγελιών σε προτεραιότητα (PRIORITY)

Στη συνέχεια εξετάσαμε την παράμετρο PRIORITY, η οποία εκφράζει των αριθμό των παραγγελιών που είναι σε προτεραιότητα έναντι των υπολοίπων. Κάναμε εκτελέσεις του κώδικα αλλάζοντας μόνο την τιμή της συγκεκριμένης παραμέτρου σε 1, 2 και 3, με τους υπολοίπους παραμέτρους να παραμένουν σταθερά. Τα αποτελέσματα φαίνονται στον πίνακα και το διάγραμμα που ακολουθούν.

**Πίνακας 8:** Αποτελέσματα προσομοίωσης για διάφορες τιμές προτεραιότητας

PRIORITY	Μέση καθυστέρηση	Κόστος/ώρα
1	0.0955	32.0304
2	0.0930	32.0823
3	0.0862	32.0657



**Γράφημα 8:** Διάγραμμα μεταβολής κόστος και μέσης καθυστέρησης σε συνάρτηση με την προτεραιότητας

Για την επιλογή των συγκεκριμένων σταθερών τιμών που δόθηκαν στην αρχή της παραγράφου 1.3, το ελάχιστο κόστος προκύπτει για PRIORITY = 2. Θα πρέπει ωστόσο να τονισθεί ότι οι τιμές αυτές όπως και το PRIORITY = 2 δεν είναι βέλτιστες, αλλά χρησιμοποιούνται απλώς για πειραματισμούς. Εδώ βλέπουμε ότι το κόστος λειτουργίας του συστήματος ανά ώρα ελαχιστοποιείται για PRIORITY = 1 και η μέση καθυστέρηση παραγγελιών ελαχιστοποιείται με το PRIORITY = 3.

## 8 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε η μοντελοποίηση και προσομοίωση της γραμμής παραγωγής αναψυκτικών της εταιρείας Γεράνι. Έγινε μια αναλυτική καταγραφή του τρόπου λειτουργίας της επιχείρησης με δεδομένα από την ίδια την εταιρεία. Για την υλοποίηση του μοντέλου εργαστήκαμε με δυο λογισμικά πρώτον το SIMIO και δεύτερον το MATLAB. Επομένως στα πλαίσια αυτής της εργασίας, είχαμε ως στόχο την ανάπτυξη του μοντέλου και την επίδειξη των δυνατοτήτων του στη βελτιστοποίηση του συστήματος μιας γραμμής παραγωγής αναψυκτικών. Πρόκειται για ένα σύστημα παραγωγής πολλών προϊόντων.

Αρχικά, επιχειρήσαμε να γίνει η μοντελοποίηση του συστήματος που εφαρμόζει η εταιρεία, μόνο για δυο τύπους παραγωγής προϊόντων (πορτοκαλάδες και λεμονάδες) σε περιβάλλον SIMIO, σχεδιάζοντας την γραμμή παραγωγής από την αρχή της εισαγωγής πρώτης ύλης (σιρόπι, γεύση, ανθρακούχο νερό) έως την παραγωγή του τελικού προϊόντος αναψυκτικού. Η προσομοίωση αυτή δεν υλοποιήθηκε για το ακριβές σενάριο του μοντέλου μας, αφού δεν μπορούσαμε να προχωρήσουμε σε χρήση γεννητριών για την άφιξη παραγγελιών όπως και του χρόνου εξυπηρέτησης που χρειάζονταν οι πελάτες της επιχείρησης.

Έτσι προχωρήσαμε στην δεύτερη παραλλαγή της προσομοίωσης μέσω του λογισμικού MATLAB. Σε αυτό το λογισμικό, αναπτύχθηκε ο αλγόριθμος ο οποίος υπολογίζει αναλυτικά διάφορα μέτρα απόδοσης του συστήματος όπως: αριθμός παραγγελιών που έφθασαν στο σύστημα, μέσο απόθεμα, μέσο χρόνο καθυστέρησης των παραγγελιών και κόστος ανά μονάδα χρόνου. Για την υλοποίηση του κώδικα έπρεπε να υπολογίσουμε τις παραμέτρους που χρησιμοποιήσαμε με βάση τα δεδομένα της εταιρείας. Συγκεκριμένα, ορίσαμε μέγιστο αριθμό παραγγελιών προτεραιότητας, αριθμός διαφορετικών προϊόντων, συνολικός χρόνος προσομοίωσης και σπόρος για τυχαίους αριθμούς (seed). Ακολουθώντας, εισήχθησαν ελάχιστοι και μέγιστοι χρόνοι μεταξύ των αφίξεων, μέγιστη και ελάχιστη ζήτηση ανά παραγγελία, ρυθμός παραγωγής κάθε προϊόντος και χρόνοι εγκατάστασης μηχανής. Τα στοιχεία που αναφέρθηκαν προηγουμένως, καταχωρήθηκαν σε αρχείο δεδομένων τα οποία χρησιμοποιήθηκαν για την εκτέλεση του κώδικα προσομοίωσης.

Έπειτα, ελέγξαμε την συνθήκη ευστάθειας του συστήματος, στοχεύοντας στην απόδειξη της ορθότητας ότι ο αλγόριθμος που εφαρμόστηκε για διάφορα σενάρια εξυπηρέτησης πελατών (παραγγελιών) οδηγεί σε ομαλή κατάσταση το σύστημα (ισορροπία).

Εξετάστηκαν όλα τα πιθανά σενάρια αλλάζοντας τις μέγιστες και ελάχιστες τιμές των αποθεμάτων (ενδεικτικά τρεις τιμές για κάθε προϊόν) και την τιμή της παραμέτρου προτεραιότητας (PRIORITY) σε 1, 2 και 3, οπότε προέκυψαν περίπου 4000 διαφορετικοί συνδυασμοί.

Περαιτέρω ανάλυση και εμβάθυνση αυτής της εργασίας θα μπορούσε να οδηγήσει σε

αλλαγή της πολιτικής που ακολουθεί το μοντέλο μας ώστε να μειώσει περαιτέρω το κόστος ή τις μέσες καθυστερήσεις.

Επίσης μια άλλη κατεύθυνση που θα μπορούσε να εξετασθεί, θα ήταν ένα ακόμα πιο πολύπλοκο σύστημα άλλης εταιρείας με αρκετές επιπλέον παραμέτρους όσο αφορά το κόστος αποθήκευσης ή συντήρησης προϊόντων.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] “Γεράνι: Η εταιρεία αναψυκτικών που διατηρεί την οικογενειακή της ταυτότητα [video],” *epixeiro.gr* || *Η επιχειρηματικότητα στο προσκήνιο*. <https://www.epixeiro.gr/article/253612> (accessed Oct. 09, 2022).
- [2] “Τα χανιώτικα αναψυκτικά που αγαπάμε στην τρίτη θέση σε πωλήσεις στην Κρήτη! | Photos.” <https://www.zarpanews.gr/ta-chaniotika-anaψυkτικά-pou-agapάme-σ> (accessed Nov. 04, 2022).
- [3] Γ. Α. Φίλης, “ΔΙΚΤΥΑ ΠΑΡΑΓΩΓΗΣ.” [Online]. Available: <https://www.eclass.tuc.gr/modules/document/file.php/MPD143/cam%20Feb2020.pdf>
- [4] A. Boualouache, S.-M. Senouci, and S. Moussaoui, “VLPZ: The Vehicular Location Privacy Zone,” *Procedia Comput. Sci.*, vol. 83, pp. 369–376, 2016, doi: 10.1016/j.procs.2016.04.198.
- [5] S. Mohammadreza, H. Aziz, M. Jahangir, and J. Chimeh, “Delay Calculation of Internet Traffic in Mobile Systems,” Jun. 2011.
- [6] Β. ΚΟΥΙΚΟΓΛΟΥ and Δ. ΚΩΝΣΤΑΝΤΑΣ, *ΠΡΟΣΟΜΟΙΩΣΗ ΣΥΣΤΗΜΑΤΩΝ ΔΙΑΚΡΙΤΩΝ ΓΕΓΟΝΟΤΩΝ*. ΔΙΣΙΓΜΑ, 2016. [Online]. Available: <https://disigma.gr/products/prosomoiosi-systimatwn-diakritwn-gegonotwn>
- [7] Γ. Α. Φίλης and Β. Σ. Κουϊκόγλου, “ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΓΩΓΗΣ.” [Online]. Available: <https://www.eclass.tuc.gr/modules/document/file.php/MPD145/production%20systems%20Jun2022.pdf>
- [8] “Flow production - Methods of production - National 5 Business management Revision,” *BBC* *Bitesize*. <https://www.bbc.co.uk/bitesize/guides/zth78mn/revision/4> (accessed Oct. 09, 2022).
- [9] C. F. M. Toledo, P. M. França, R. Morabito, and A. Kimms, “Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem,” *Int. J. Prod. Res.*, vol. 47, no. 11, pp. 3097–3119, Jun. 2009, doi: 10.1080/00207540701675833.
- [10] C. F. M. Toledo, A. Kimms, P. M. França, and R. Morabito, “The Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem: Evaluating the Generalized Mathematical Model,” *Math. Probl. Eng.*, vol. 2015, pp. 1–18, 2015, doi: 10.1155/2015/182781.
- [11] M. Maldonado, S. Rangel, and D. Ferreira, “A Study of Different Subsequence Elimination Strategies for the Soft Drink Production Planning,” *J. Appl. Res. Technol.*, vol. 12, no. 4, pp. 631–641, Aug. 2014, doi: 10.1016/S1665-6423(14)70080-X.
- [12] E. Mendez, “Simulation Modeling with SIMIO: A Workbook”, Accessed: Nov. 03, 2022. [Online]. Available: [https://www.academia.edu/40451739/Simulation\\_Modeling\\_with\\_SIMIO\\_A\\_Workbook](https://www.academia.edu/40451739/Simulation_Modeling_with_SIMIO_A_Workbook)

# ΠΑΡΑΡΤΗΜΑ 1

ΣΥΝΑΡΤΗΣΗ ΣΕ MATLAB  (FUNCTION)

```
function [kostos]=Simulation5(A,B,p_nodemand,min_demand,max_demand,  
r,s,C,D,priority,l_high,l_low)  
  
clc  
np=A(1);  
tsim=A(2);  
iseed=A(3);  
  
min_interarr_t=B(1);  
max_interarr_t=B(2);  
  
% outfile will keep the FINAL results at time t=tsim;  
outfile = fopen('Simul_Gerani_out.txt','a'); %'w'=overwrite old file  
  
% At time 0: total number of orders arrived tot_orders,  
% number of outstanding orders nq,  
% mean delay of orders, stock levels, average levels: ALL=0  
tot_orders=0;  
nq=0;  
tprev=0;
```

```
for j=1:np  
    distance=l_high(j)-l(j);  
    % We adopt the following heuristic POLICY:  
    % We find the product i which has the MAX distance from target.  
    % This MAX distance is the lot_size of the production cycle,after setup  
    if distance>lot_size  
        i=j;  
        isetup=1;  
        lot_size=distance;  
    end  
end  
  
%Event types 1=order arrival, 2=machine-related, 3=order delivery(ord_deliv  
%Subtypes of 2: 1=setup begins, 2=production begins, 3=production ends  
evtime(1)=t+min_interarr_t+rand*(max_interarr_t-min_interarr_t);  
if i==0  
    evtime(2)=tsim+1;
```

```

else
    evtime(2)=0;
    evsubtype=1;
end
evtime(3)=tsim+1;
ord_deliv=0; % index of priority order in the queue to be delivered next

while t<=tsim
    % Find time and type of next event
    tprev=t;
    t=tsim+1;
    for j=1:3
        if evtime(j)<t
            t=evtime(j);
            event=j;
        end
    end
end

```

```

% Update system and execute event ONLY if it takes place at time t <=tsim
if t<=tsim

    % Time has advanced to t; update stock level of j=i if machine is w
    % and update ALL average stock levels j
    for j=1:np
        lprev=l(j); % the level at the time of previous event--for use
        if (j==i)&&(isetup==0) % machine produces j=i so l(i) changes
            l(i)= l(i) + r(i)*(t-tprev); % level at time t
        end
        % (*) Update average stock of j by ADDING area of trapezoid: height
        ave_l(j)= ave_l(j) + (t-tprev)*(lprev+l(j))/2;
    end

    % Execute event at time t
    if event==1
        % EVENT=New order arrives; demand not yet specified
        yesdemand=0;
    end
end

```

```

for j=1:np
    p=rand;
    if p>p_nodemand(j)
        % At least one product (=j) is included in this order, so nq is going to
        % increase
        yesdemand=1;
        q(j,nq+1)=min_demand(j)+(max_demand(j)-min_demand(j))*rand;
    else
        q(j,nq+1)=0;
    end
end
if yesdemand==1
    tot_orders=tot_orders+1;
    nq=nq+1;
    % If new order is in the priority list, then:
    if nq<=priority
        % EITHER new order will receive immediate service (and nq=nq-1)
        % OR it will join the queue.
        % Under certain conditions, this new order may trigger a new setup.
    end
end

```

```

    enoughstock=1;
    j=0;
    while (j<np) && (enoughstock==1)
        j=j+1;
        if l(j)<q(j,nq)
            enoughstock=0;
        end
    end
    if enoughstock==1
        % New order is served immediately; reduce stocks and remove customer
        % from queue
        for j=1:np
            l(j)=l(j)-q(j,nq);
        end
        nq=nq-1;
    end

```

```

    else
        % New order is in priority list BUT there is not enough stock and
        % it MUST join the queue; its delay (departure time - arrival time)
        % will be added to sumofdelays;
        % NOW we subtract arrival time=t; LATER we shall add the departure time
        sumofdelays=sumofdelays-t;
    end
    % The new order is in the priority list (see above: nq<=priority). Now
    % a) if machine is idle, perhaps a new setup must commence
    % b) if machine is EITHER being set up for i OR has already started
    % producing i, then its "lot_size" may increase (if new order needs i)
    % and the production cycle may have to be prolonged.
    % "lot_size" is the estimated amount to be produced during either the
    % CURRENT cycle or the NEXT cycle if machine is already set up
    % or is idle but will commence a setup.
    if i==0 % a) The machine is idle:
        % Find distance of l(j) from target of each product j, and find WHICH
        % product has max distance from target where [Distance from target]=
        % [base stock, l_high - [current stock, l] + [demand of priority only]
    end

```

```

% Initially max distance=0
lot_size=0;
for j=1:np
    % Calculate needs of product j for priority customers
    sumqj=0;
    for k=1:min(nq,priority) % if nq<priority then only nq orders exist
        sumqj=sumqj+q(j,k);
    end
    distance=l_high(j)-l(j)+sumqj;
    % if net level is lower than emergency threshold and distance from target
    % higher than current lot_size then set up for product j
    if (l(j)-sumqj<l_low(j)) && (distance>lot_size)
        i=j;
        isetup=1;
        lot_size=distance;
        % lot_size=estimated quantity to be produced during the NEXT cycle, after
        % if the final choice is made for product i.
    end
end

```

```

    % if i=0, then all net inventories are in safe zone (l_low, l_high);
    % otherwise:
    if (i>0)
        evtime(2)=t; % schedule immediate setup to begin producing i
        evsubtype=1;
        end
    else% b) The machine is EITHER currently being set up for some product i
    %OR already produces i. Index i has been determined previously
    % Lot size=distance from target for product i
        lot_size=l_high(i)-l(i);
        for k=1:min(nq,priority) % if nq<priority then only nq orders exist
            lot_size=lot_size+q(i,k);
        end
    if (isetaup==0)% Machine already producing i; schedule end of production
        evtime(2)=t+lot_size/r(i);
        evsubtype=3;
    end
end

```

```

% If isetaup=1, then only lot_size is increased by the new arrival,
% but evtime (=end of setup) doesn't need to be changed.
    end
else
    % New order NOT in priority list and MUST join the queue;
    % its delay (departure time - arrival time) will be added to sumofdelays;
    % NOW we subtract arrival time=t; LATER we shall add the departure time.
    sumofdelays=sumofdelays-t;
    end
end
% Schedule new order arrival
evtime(1)=t+min_interarr_t+rand * (max_interarr_t-min_interarr_t);
elseif event==3 % A priority order is delivered; its index is 'ord_deliv'
    for j=1:np
        l(j)=l(j)-q(j,ord_deliv); % stocks are reduced by the amounts delivered
    end
    sumofdelays=sumofdelays+t; % update delay
    nq=nq-1; % reduce queue
    for k=min(ord_deliv,np):nq

```

```

        for j=1:np
            q(j,k)=q(j,k+1);
        end
    end
elseif evsubtype==1 % From this point on all events are machine-related,
% subtype=Machine is being set up for product i; next event = start prod.
    numberofsetups=numberofsetups+1;
    evtime(2)=t+s(i);
    evsubtype=2;
    elseif evsubtype==2
% subtype=Machine completes a setup and starts making product i
    isetaup=0;
% Schedule end of production using distance of net inventory from target
    lot_size=l_high(i)-l(i);
    for k=1:min(nq,priority)
        lot_size=lot_size+q(i,k);
    end

```



```

% next event = end of production cycle
    evttime(2)=t+lot_size/r(i);
    evsubtype=3;
    else
%   subtype=Machine completes a production cycle; check if it must start
%   making another product
        i=0;
        lot_size=0;
    for j=1:np
%   Calculate needs of product j for priority customers
        sumqj=0;
        for k=1:min(nq,priority) % if nq<priority then only nq orders exist
            sumqj=sumqj+q(j,k);
        end
        distance=l_high(j)-l(j)+sumqj;
%   if net level is lower than emergency threshold & distance from target is
%   higher than current lot_size then set up for product j

```

```

    if (l(j)-sumqj< l_low(j))&&(distance>lot_size)
        i=j;
        isetup=1;
        lot_size=distance;
%   lot_size=estimated quantity to be produced during the NEXT cycle, after :
%   if the final choice is made for product i.
    end
    end
    if i>0
        evttime(2)=t; % schedule immediate setup
        evsubtype=1;
    else
%   if i=0, then all net inventories are in safe zone (l_low, l_high]
        evttime(2)=tsim+1;
    end
    end
%   Find priority order to be delivered next; this is possible only if the
%   machine is producing
    evttime(3)=tsim+1;

```

```

    ord_deliv=0;
    if (nq>0)&&(i>0)&&(isetup==0) % If machine is producing
        for k=1:min(nq,priority)
            possible=1; % Initialization: possible=yes
            for j=1:np
                if (possible==1)&&(j==i)
%   Time when ALL products i required by order k will be produced
                    if q(i,k)>l(i)
                        tentative_time=t+(q(i,k)-l(i))/r(i);
                    else
                        tentative_time=t; % Perhaps "else", i.e., q(i,k)<=l(i),
%   should never happen here but only upon the arrival of order k
                    end
                    elseif (possible==1) % Here j NOT= i
                        if q(j,k)>l(j) % q(j,k) must be <= l(j),
%   otherwise it is IMPOSSIBLE for order k to be filled during the current
%   production cycle because there is shortage of ANOTHER product j.
                            possible=0;
                        end

```

```

        end
    end
    if (possible==1)&&(tentative_time<evtime(3))
        evtime(3)=tentative_time;
        ord_deliv=k;
    end
end
end

end

end

% Advance time to tsim; update stock level j=i if machine is working on i
% and update ALL average stock levels j
t=tsim;
for j=1:np
    lprev=l(j);
    if (j==i)&&(isetaup==0) % machine produces j=i so l(j) changes
        l(i)=l(i)+r(i)*(t-tprev);
    end
end

```

```

% Update average stock of j by ADDING area of trapezoid=
% height * (base_a+base_b)/2
ave_l(j)= ave_l(j) + (t-tprev)*(lprev+l(j))/2;
end

for j=1:np
    ave_l(j)=ave_l(j)/tsim;
end
delay=(sumofdelays+nq*tsim)/tot_orders;

outfile = fopen('Simul_Gerani_out.txt','a');
fprintf(outfile,'Priority %d\n',priority);
fprintf(outfile,'high level j=\t');
fprintf(outfile,'%f\t',l_high);
fprintf(outfile,'\n');

fprintf(outfile,'low level j=\t');
fprintf(outfile,'%f\t',l_low);
fprintf(outfile,'\n');

```

```

fprintf(outfile, '\n**--END OF SIMULATION   Random Number Seed= %d\n', iseed);
fprintf(outfile, '      Time= %.4f \t N. of setups= %d\n', t, numberofsetups);
fprintf(outfile, ' N.orders arrived=%d \t Now waiting=%d \t Meandelay=%.4f\n', tot_orders, nq,
for j=1:np
    fprintf(outfile, 'stock(%d)= %.3f \t   mean_stock(%d)= %.3f\n', j, l(j), j, ave_l(j));
end

suml =0;
for j=1:np
    suml=suml+ave_l(j);
end
kostos = C*suml+D*delay;
fprintf(outfile, 'Kostos =%.4f\n', kostos);
fprintf(outfile, '\n\n');
fclose('all');
end

```

## ΠΑΡΑΡΤΗΜΑ 2

ΚΩΔΙΚΑΣ ΣΕ ΛΟΓΙΣΜΙΚΟ MATLAB  (CODE)

```
clear all
x=1;
y=10;
min_cost = realmax;

% outfile will keep the FINAL results at time t=tsim;
outfile = fopen('Simul_Gerani_out.txt','w'); %'w'=overwrite old file

infile=fopen('Simul_Gerani_in.txt');
% First skip two lines and then read #priority customers, #products, tsim
nextline=fgetl(infile);
nextline=fgetl(infile);
nextline=fgetl(infile);
A=sscanf(nextline,'%d %f %d');

np=A(1);
tsim=A(2);
iseed=A(3);
```

```
rand('twister',iseed);

nextline=fgetl(infile);
nextline=fgetl(infile);
B=sscanf(nextline,'%f %f');
min_interarr_t=B(1);
max_interarr_t=B(2);

nextline=fgetl(infile);
nextline=fgetl(infile);
p_nodemand=sscanf(nextline,'%f',np);

% Read min demand of product, IF INCLUDED in one order
nextline=fgetl(infile);
nextline=fgetl(infile);
min_demand=sscanf(nextline,'%f',np);
```

```

% Read max demand of product, IF INCLUDED in one order
nextline=fgetl(infile);
nextline=fgetl(infile);
max_demand=sscanf(nextline,'%f',np);

% Read production rate for each product
nextline=fgetl(infile);
nextline=fgetl(infile);
r=sscanf(nextline,'%f',np);

% Read setup time for each product
nextline=fgetl(infile);
nextline=fgetl(infile);
s=sscanf(nextline,'%f',np);

```

```

for priority=1:3
    for l_high1=15.60:15.60:3*15.60
        for l_high2=15.91:15.91:3*15.91
            for l_high3=15.91:15.91:3*15.91
                for l_high4=16.51:16.51:3*16.51
                    for l_low1=0:15.60:l_high1-15.60
                        for l_low2=0:15.91:l_high2-15.91
                            for l_low3=0:15.91:l_high3-15.91
                                for l_low4=0:16.51:l_high4-16.51

                                    l_high(1) = l_high1;
                                    l_high(2) = l_high2;
                                    l_high(3) = l_high3;
                                    l_high(4) = l_high4;
                                    l_low(1)= l_low1;
                                    l_low(2)= l_low2;
                                    l_low(3)= l_low3;
                                    l_low(4)= l_low4;
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```

```

[cost] = Simulation5(A,B,p_nodemand,min_demand,max_demand,r,s,x,y,priority,l_high,l_low);

    if (cost < min_cost)
        min_cost = cost;
        optimal = priority;
        for i =1:4
            optimal_lhigh(i)=l_high(i);
            optimal_low(i)=l_low(i);
        end
    end
end
end
end
end
end
end

```

```

end
end
end
end

outfile = fopen('Simul_Gerani_out.txt','a');
fprintf(outfile,'The best solution is: with cost =%f\n\tpriority %d\t
l_high(1)=%f\t l_high(2)=%f\t l_high(3)=%f\t l_high(4)=%f\n',
min_cost,optimal,optimal_lhigh(1),optimal_lhigh(2),
optimal_lhigh(3),optimal_lhigh(4));

fprintf(outfile,'\n\t l_low(1)=%f\t l_low(2)=%f\t l_low(3)=%f\t
l_low(4)=%f\t',optimal_low(1),optimal_low(2),optimal_low(3),optimal_low(4));

```