



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΤΟΜΕΑΣ ΟΡΓΑΝΩΣΗΣ ΠΑΡΑΓΩΓΗΣ &
ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ**

ΕΥΣΤΡΑΤΙΟΥ Π. ΑΘΑΝΑΣΙΟΥ
Διπλωματούχου Μηχανολόγου Μηχανικού Α.Π.Θ.

ΜΕΤΑΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Προγραμματισμός και Έλεγχος Παραγωγής Χημικής Βιομηχανίας σε Καθοριστικό
και Στοχαστικό Περιβάλλον**

ΒΟΛΟΣ, 2010



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΒΙΒΛΙΟΘΗΚΗ & ΚΕΝΤΡΟ ΠΛΗΡΟΦΟΡΗΣΗΣ
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 8204/1
Ημερ. Εισ.: 11-03-2010
Δωρεά: Συγγραφέα
Ταξιθετικός Κωδικός: Δ
519.72
ΑΘΑ

ΕΥΣΤΡΑΤΙΟΥ Π. ΑΘΑΝΑΣΙΟΥ
Διπλωματούχου Μηχανολόγου Μηχανικού Α.Π.Θ.

**Προγραμματισμός και Έλεγχος Παραγωγής Χημικής Βιομηχανίας σε Καθοριστικό
και Στοχαστικό Περιβάλλον: Σύγκριση των Αποτελεσμάτων**

ΜΕΤΑΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Υποβλήθηκε στο Τμήμα Μηχανολόγων Μηχανικών
Πανεπιστημίου Θεσσαλίας**

Ημερομηνία Παράδοσης: 23 Φεβρουαρίου 2010

Εξεταστές:

Δρ. Δημήτριος Παντελής, Επίκουρος Καθηγητής

Δρ. Γεώργιος Λυμπερόπουλος, Καθηγητής

Δρ. Αθανάσιος Ζηλιασκόπουλος, Καθηγητής

Στους γονείς μου για την ηθική και υλική υποστήριξή τους στα έτη των σπουδών μου και για την αμέριστη συμπαράσταση και εμπιστοσύνη που μου επιδεικνύουν.

ΠΡΟΛΟΓΟΣ

Ο Τομέας Οργάνωσης Παραγωγής & Βιομηχανικής Διοίκησης του τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας καλύπτει ένα ευρύ πεδίο επιστημονικών εφαρμογών και διερεύνησης προς την κατεύθυνση βελτίωσης ή/και βελτιστοποίησης της συνολικής λειτουργίας των παραγωγικών συστημάτων. Η ανάπτυξη μεθοδολογιών βασίζεται σε ένα ευρύτερο επιστημονικό υπόβαθρο το οποίο εμπεριέχει γνώσεις των θετικών επιστημών, προσφέροντας στον Διπλωματούχο Μηχανολόγο Μηχανικό τη δυνατότητα απεικόνισης ενός πραγματικού παραγωγικού συστήματος σε ένα ή περισσότερα συνεργαζόμενα μαθηματικά μοντέλα. Εφάμιλλο αντικείμενο επιστημονικής εργασίας αποτελεί και η διαμόρφωση αναλυτικών μοντέλων για την απεικόνιση και βελτιστοποίηση του τρόπου λειτουργίας διαφόρων βιομηχανιών, ώστε να ανευρεθεί η βέλτιστη πολιτική λειτουργίας τους και η βέλτιστη στρατηγική προς την κατεύθυνση της διαδικασίας λήψης αποφάσεων. Ταυτόχρονα με τις αναλυτικές μεθόδους βελτιστοποίησης μπορούν να χρησιμοποιηθούν και μέθοδοι προσομοίωσης προς την κατεύθυνση δημιουργίας μοντέλων υψηλής ευελιξίας τα οποία αφενός να ανταποκρίνονται στις δυναμικές απαιτήσεις των πραγματικών συστημάτων και αφετέρου να αποδίδουν πάντοτε μία εφικτή λύση, άλλοτε λιγότερο και άλλοτε περισσότερο αποδοτική. Τα μοντέλα προσομοίωσης μπορούν να χρησιμοποιηθούν είτε σε συνεργασία είτε παράλληλα με αναλυτικά μοντέλα, και ιδιαίτερα σε περιπτώσεις όπου τα αναλυτικά μοντέλα αδυνατούν να δώσουν εφικτή λύση λόγω παραβίασης των περιορισμών. Σε αυτό το πλαίσιο εκπονήθηκε και η παρούσα μεταδιπλωματική εργασία, προσεγγίζοντας τη λειτουργία μίας τυπικής χημικής βιομηχανίας παραγωγής ρητίνης PET.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όλους όσους συνετέλεσαν στην προσπάθεια αυτή. Συγκεκριμένα ευχαριστώ :

Τον Επίκουρο Καθηγητή του Τμήματος κ. Δημήτριο Παντελή για την εποικοδομητική και συντονισμένη καθοδήγησή του, καθώς ακόμη και για τις χρήσιμες υποδείξεις και διορθώσεις του.

Επίσης θα ήθελα να ευχαριστήσω τη Διδάκτορα του Τμήματος κα. Χατζηκωνσταντίνου Ολυμπία για τις συμβουλές της κατά την πορεία ολοκλήρωσης της παρούσας μεταδιπλωματικής εργασίας.

Τέλος θα ήθελα να ευχαριστήσω όλους τους διδάσκοντες του Τομέα για τις γνώσεις που αποκόμισα καθώς και τους γονείς μου για την αμέριστη συμπαράστασή τους σε ολόκληρη τη διάρκεια των σπουδών μου.

Βόλος, Φεβρουάριος 2010

Αθανασίου Π. Ευστράτιος

Περιεχόμενα

1	Εισαγωγή.....	1
1.1	Γενικά.....	1
1.2	Σκοπός - μεθοδολογία.....	2
1.3	Δομή - περιεχόμενα.....	3
2	Περιγραφή της χημικής βιομηχανίας.....	5
2.1	Εισαγωγή.....	5
2.2	Περιγραφή παραγωγικής διαδικασίας.....	6
2.3	Ποιοτικά δεδομένα εισόδου κατά τη μοντελοποίηση.....	10
2.4	Ποσοτικά δεδομένα εισόδου κατά τη μοντελοποίηση.....	12
3	Γραμμικός προγραμματισμός.....	17
3.1	Γενικά.....	17
3.2	Αναγκαιότητα ύπαρξης μεθόδων βελτιστοποίησης.....	17
3.3	Μαθηματικός προγραμματισμός.....	19
3.4	Βασικά χαρακτηριστικά των προβλημάτων μαθηματικού προγραμματισμού.....	20
3.5	Μορφοποίηση προβλημάτων MILP.....	22
3.5.1	Μορφοποίηση με τη μέθοδο της άμεσης προσέγγισης.....	22
3.5.2	Μορφοποίηση με την προσέγγιση των εισροών-εκροών.....	23
4	Συνοπτική περιγραφή μοντέλου MILP.....	25
4.1	Μαθηματική μορφοποίηση.....	25
4.1.1	Ορισμός παραμέτρων και μεταβλητών.....	25
4.1.2	Μοντέλο MILP.....	28
4.2	Κυριότερες υποθέσεις του γραμμικού μοντέλου.....	30
5	Δυναμική Συστημάτων.....	33
5.1	Μεθοδολογία Δυναμικής Συστημάτων (System Dynamics).....	33
5.2	Δομή των διαγραμμάτων επιρροής.....	35
5.3	Δομή των διαγραμμάτων ροής.....	36
5.4	Λογισμικά προγραμματισμού Δυναμικής Συστημάτων.....	37
5.5	Η λειτουργία της γλώσσας προγραμματισμού Powersim® 2.5c.....	38
6	Περιγραφή του μοντέλου Δυναμικής Συστημάτων.....	41
6.1	Εσωτερικό και εξωτερικό περιβάλλον του συστήματος.....	41
6.2	Υποθέσεις του δυναμικού μοντέλου.....	42
6.3	Διάγραμμα επιρροής.....	43
6.4	Διάγραμμα ροής στην Powersim® 2.5c.....	45
6.4.1	Γενικά.....	45

6.4.2	Διαχείριση αποθεμάτων του LFSS και επιλογή χρώματος του POLY	47
6.4.3	Παραγωγικό στάδιο πολυμερισμού POLY	49
6.4.4	Επιλογή σιλό και slot αποθήκευσης ημιεπίμων προϊόντων στο TSS.....	50
6.4.5	Στάδιο αποθήκευσης ημιεπίμων προϊόντων TSS.....	51
6.4.6	Παραγωγικό στάδιο απόδοσης ιξώδους SSP.....	53
6.4.7	Υπολογισμός βέλτιστου σιλό αποθήκευσης στο LFSS	55
6.4.8	Υπολογισμός προϊόντων που απορρίπτονται.....	57
6.4.9	Ζήτηση τελικών προϊόντων	58
6.4.10	Εκροή των σιλό του LFSS.....	59
6.4.11	Διαχείριση αποθεμάτων αποθήκης.....	62
6.4.12	Υπολογισμός της ανικανοποίητης ζήτησης και του fill rate.....	63
6.4.13	Υπολογισμός του συνόλου των μεταβάσεων.....	65
6.4.14	Διαμόρφωση μαθηματικού μοντέλου.	66
7	Συγκριτική ανάλυση των αποτελεσμάτων - Σύνοψη	67
7.1	Αποτελέσματα μοντέλου ΔΣ	67
7.2	Ανάλυση των αποτελεσμάτων του μοντέλου ΔΣ	69
7.3	Σύγκριση των αποτελεσμάτων των μοντέλων ΔΣ και ΓΠ.....	71
7.4	Συμπεράσματα από τη χρήση των δύο μεθοδολογιών προσέγγισης	73
7.5	Προτάσεις για περαιτέρω διερεύνηση.	74
	Βιβλιογραφία.....	77
	Παράρτημα Α.....	79

1 ΕΙΣΑΓΩΓΗ

1.1 Γενικά

Μία από τις σχετικά σύγχρονες μεθόδους η οποία χρησιμοποιείται ευρέως σήμερα στη βελτιστοποίηση της λειτουργίας πραγματικών συστημάτων είναι ο γραμμικός προγραμματισμός (ΓΠ). Σύμφωνα με τη θεωρία του ΓΠ κάθε φυσικό μέγεθος μετατρέπεται σε μαθηματική μεταβλητή μίας ή πολλών διαστάσεων ενώ το σύνολο των συναρτησιακών σχέσεων που τις συνδέει αποτελεί το γραμμικό μοντέλο. Έπειτα από την επίλυση του μοντέλου, κάθε μεταβλητή λαμβάνει μία μοναδική τιμή η οποία και θεωρείται βέλτιστη. Επειδή όμως τα φυσικά μεγέθη δεν είναι στατικά αλλά εξελίσσονται δυναμικά και τις περισσότερες φορές απρόβλεπτα συναρτήσκει του χρόνου, η μέθοδος του ΓΠ αποδείχτηκε ανεπαρκής στις περιπτώσεις περιγραφής και επίλυσης μη καθοριστικών συστημάτων αυξημένης μεταβλητότητας. Προκειμένου να αντιμετωπιστεί αυτό το πρόβλημα αναζητήθηκαν εναλλακτικές μέθοδοι περιγραφής και επίλυσης οι οποίες οδήγησαν στην ανάπτυξη της προσομοίωσης και ειδικότερα της Δυναμικής Συστημάτων (ΔΣ). Οι δύο αυτές διαφορετικές μεθοδολογίες προσέγγισης πραγματικών προβλημάτων παραγωγής, έχουν διαφορετικές απαιτήσεις προκειμένου το αποτέλεσμα το οποίο εξάγουν να είναι μαθηματικά αλλά και ρεαλιστικά αποδεκτό, με αποτέλεσμα η μία όχι μόνο να μην παραγκωνίζει την άλλη αλλά αντιθέτως να αλληλοσυμπληρώνονται ως επιστημονικά εργαλεία μελέτης στη διάθεση του Υπευθύνου Παραγωγής. Στα κεφάλαια τα οποία ακολουθούν γίνεται εκτενής αναφορά σε κάθε μία από τις δύο αυτές μεθοδολογίες με στόχο να καταστεί προφανής η μεταξύ τους σχέση αλλά και μέσω της σύγκρισής τους, να γίνουν αντιληπτά τα πλεονεκτήματα και τα μειονεκτήματα της μίας έναντι της άλλης.

Προκειμένου να εξυπηρετηθεί αυτός ο σκοπός, εξετάζεται στην παρούσα μεταδιπλωματική εργασία ο προγραμματισμός παραγωγής μίας πραγματικής χημικής βιομηχανίας που παράγει ρητίνη PET. Η ρητίνη PET χρησιμοποιείται στην παραγωγή μπουκαλιών εμφιάλωσης λόγω των σύγχρονων τάσεων χρησιμοποίησης πλαστικών υλικών προς την αντικατάσταση των ενεργειοβόρων και δύσχρηστων υλικών εμφιάλωσης (αλουμίνιο, γυαλί). Πέρα από την οικονομική διάσταση του προβλήματος, τόσο το νομικό πλαίσιο προστασίας του φυσικού περιβάλλοντος όσο και το πράσινο προφίλ που επιθυμούν οι εταιρείες να προβάλλουν, οδηγεί στην έντονη τάση για χρήση ανακυκλώσιμων υλικών. Προς την κατεύθυνση αυτή μελετάται στην παρούσα εργασία η βελτιστοποίηση του προγράμματος παραγωγής μίας τυπικής χημικής βιομηχανίας παραγωγής ρητίνης PET μέσω ενός προϋπάρχοντος γραμμικού μοντέλου, όποτε αυτό είναι εφικτό (μη παραβίαση των περιορισμών) ή της βελτίωσής του μέσω ενός προσομοιωτικού μοντέλου, όταν η χρήση του γραμμικού μοντέλου δεν είναι εφικτή (παραβίαση των περιορισμών του γραμμικού μοντέλου).

1.2 Σκοπός - μεθοδολογία

Στην παρούσα εργασία περιγράφεται το προϋπάρχον γραμμικό μοντέλο και διαμορφώνεται ένα προσομοιωτικό μοντέλο περιγραφής μίας τυπικής χημικής βιομηχανίας παραγωγής ρητίνης PET. Η εταιρεία δραστηριοποιείται στον κλάδο των ανακυκλώσιμων υλικών εμφιάλωσης PET διαφορετικών τύπων και απαιτήσεων. Η πρώτη ύλη είναι η ίδια για όλα τα προϊόντα ενώ κατά την παραγωγική διαδικασία προσδίδονται δύο χαρακτηριστικά στο τελικό προϊόν, το χρώμα και το ιξώδες σε δύο διαδοχικά στάδια παραγωγής. Το σύστημα περιπλέκεται τόσο από την περιορισμένη χωρητικότητα των διακριτών θέσεων ενδιάμεσων χώρων αποθήκευσης όσο και από τη μορφή της ζήτησης για ένα τελικό προϊόν η οποία μπορεί να είναι είτε για σακευμένο προϊόν είτε για προϊόν σε μορφή χύδην το οποίο μεταφέρεται με κοντέινερ ή με σιλοφόρα φορτηγά. Σημαντικό παράγοντα περιπλοκής της διαδικασίας λήψης αποφάσεων αποτελούν και οι σχετικά μεγάλοι χρόνοι μετάβασης που απαιτούνται προκειμένου να ολοκληρωθούν οι απαραίτητες ρυθμίσεις κατά την αλλαγή της παραγωγικής διαδικασίας από ένα προϊόν σε κάποιο άλλο. Ακόμη, κατά την διάρκεια των μεταβάσεων παράγονται προϊόντα εκτός των απαιτούμενων προδιαγραφών ποιότητας τα οποία η βιομηχανία είτε τα πουλάει σε χαμηλότερη τιμή δημιουργώντας διαφυγόντα κέρδη, είτε τα αναμιγνύει με ποιοτικά ανώτερα προϊόντα μειώνοντας την ποιότητά τους. Επομένως, οι συνεχείς μεταβάσεις είναι ανεπιθύμητες τόσο από απόψεως μείωσης του καθαρού παραγωγικού χρόνου, όσο και από απόψεως παραγωγής παραπροϊόντων.

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι διπλός: αφενός, η ικανοποίηση όσο το δυνατόν περισσότερης ζήτησης στο συμφωνημένο χρόνο (δεν επιτρέπεται καθυστερημένη ικανοποίηση της ζήτησης) με ταυτόχρονη ελαχιστοποίηση των ανεπιθύμητων απαιτούμενων αλλαγών της παραγωγικής διαδικασίας. Αφετέρου, ιδιαίτερο ενδιαφέρον παρουσιάζει η μελέτη της αποδοτικότητας των δύο εναλλακτικών μεθοδολογιών προσέγγισης του ίδιου προβλήματος, του Γραμμικού Προγραμματισμού και της Δυναμικής Συστημάτων αλλά υπό διαφορετικές συνθήκες λειτουργίας (καθοριστικό και μη καθοριστικό περιβάλλον λειτουργίας αντίστοιχα).

Όπως αναφέρθηκε, στην προσπάθεια απεικόνισης του υπό εξέταση συστήματος παραγωγής διαμορφώθηκαν δύο μοντέλα: ένα γραμμικό (που προϋπήρχε της παρούσας μεταδιπλωματικής εργασίας) και ένα προσομοίωσης. Το μοντέλο προσομοίωσης δημιουργήθηκε με χρήση της Θεωρίας $\Delta\Sigma$ και ενσωματώνει όλους τους κανόνες του πραγματικού συστήματος που περιγράφονται από το γραμμικό μοντέλο. Κατά τη διαμόρφωση αφαιρέθηκαν όλες οι διαδικασίες βελτιστοποίησης διότι κατά την προσομοίωση μοντέλων δεν υφίσταται ενδογενής διαδικασία βελτιστοποίησης αλλά διαδικασία περιγραφής, απεικόνισης, προσομοίωσης και τέλος εξωγενούς βελτίωσης. Το γραμμικό μοντέλο που περιγράφει και βελτιστοποιεί τη λειτουργία του πραγματικού συστήματος ελήφθη από τη δημοσιευμένη επιστημονική εργασία των Liberopoulos et al.³ (2009).

Η ανάγκη χρησιμοποίησης των δύο μεθοδολογιών προκύπτει από τις εναλλακτικές περιπτώσεις υπό τις οποίες μπορεί να λειτουργεί ένα πραγματικό σύστημα παραγωγής. Ειδικότερα, το γραμμικό μοντέλο χρησιμοποιείται οποτεδήποτε τα δεδομένα της βιομηχανίας είναι καθοριστικά και δεν παραβιάζονται οι περιορισμοί του συστήματος. Αντιθέτως, το προσομοιωτικό μοντέλο χρησιμοποιείται όταν είτε υπάρχει αβεβαιότητα στο σύστημα, είτε παραβιάζονται οι περιορισμοί του γραμμικού μοντέλου οπότε υπεισέρχονται στη διαδικασία ικανοποίησης της ζήτησης και ποιοτικά χαρακτηριστικά που δε μπορούν να απεικονιστούν με μαθηματικές εξισώσεις, (π.χ. η ταξινόμηση πελατών), είτε και τα δύο μαζί.

Τα δύο μοντέλα έχουν ως στόχο την ικανοποίηση της ζήτησης που προκύπτει ανά περίοδο με ταυτόχρονη ελαχιστοποίηση (για το μοντέλο ΓΠ) ή μείωση (για το μοντέλο ΔΣ) του συνολικού αριθμού των αλλαγών που απαιτούνται στην παραγωγική διαδικασία. Όπως γίνεται εύκολα αντιληπτό μία τέτοια σύγκριση παρουσιάζει ιδιαίτερο ενδιαφέρον αφού η σχετικά ταχύτερη και ευκολότερη δημιουργία και χρησιμοποίηση ενός προσομοιωτικού μοντέλου με σύγχρονες γλώσσες προγραμματισμού, είναι άγνωστο τι βαθμό προσέγγισης της πραγματικά βέλτιστης λύσης παρουσιάζει έτσι ώστε ο Υπεύθυνος Παραγωγής να γνωρίζει πότε είναι σκόπιμο να τη χρησιμοποιήσει αποδεχόμενος πιθανή μειωμένη ακρίβεια, και πότε απαιτείται η πιο σύνθετη και χρονοβόρος μέθοδος του ΓΠ η οποία παρουσιάζει υψηλή ακρίβεια αποτελεσμάτων.

1.3 Δομή - περιεχόμενα

Η παρούσα διπλωματική εργασία αποτελείται από επτά κεφάλαια. Στο 2^ο γίνεται μία γενική περιγραφή της χημικής βιομηχανίας. Ειδικότερα περιγράφεται η διαδικασία παραγωγής και τα ποιοτικά/ποσοτικά δεδομένα εισόδου στα δύο πρότυπα ΓΠ και ΔΣ. Στο 3^ο κεφάλαιο παρουσιάζεται το γενικό θεωρητικό υπόβαθρο και ο γενικός τρόπος μοντελοποίησης της μεθοδολογίας του ΓΠ. Στο 4^ο κεφάλαιο παρουσιάζεται το μοντέλο Μεικτού Ακέραιου ΓΠ (Mixed Integer Linear Programming - MILP) που δημιουργήθηκε από τους Liberopoulos et al.^a (2009). Ειδικότερα παρουσιάζεται η μαθηματική μορφοποίηση, ο ορισμός των παραμέτρων και των μεταβλητών του προβλήματος, το μοντέλο MILP και οι κυριότερες υποθέσεις του. Στο 5^ο κεφάλαιο παρουσιάζεται εν συντομία το γενικό θεωρητικό υπόβαθρο της Θεωρίας ΔΣ και γίνεται μία σύντομη περιγραφή των μεταβλητών του λογισμικού Powersim[®] 2.5c το οποίο χρησιμοποιήθηκε για τη δημιουργία του προσομοιωτικού μοντέλου. Στο 6^ο κεφάλαιο παρουσιάζεται το προσομοιωτικό μοντέλο. Ειδικότερα η περιγραφή του γίνεται ανά στάδιο παραγωγής ή αποθήκευσης ώστε η κατανόησή του από τον αναγνώστη να είναι όσο το δυνατόν πληρέστερη. Τέλος, στο 7^ο κεφάλαιο παρουσιάζονται και σχολιάζονται συγκριτικά τα αποτελέσματα των μοντέλων MILP και ΔΣ, καθώς και τα συμπεράσματα τα οποία προέκυψαν από τη χρήση των δύο διαφορετικών μεθοδολογικών προσεγγίσεων. Το κεφάλαιο και τη μεταδιπλωματική κλείνουν κάποιες προτάσεις για περαιτέρω διερεύνηση.

2 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΧΗΜΙΚΗΣ ΒΙΟΜΗΧΑΝΙΑΣ

2.1 Εισαγωγή

Η παραγωγή χημικών και πλαστικών προϊόντων είναι βασισμένη στην επεξεργασία του πετρελαίου και των παραγώγων του, του φυσικού αερίου και των προϊόντων άνθρακα. Τα προϊόντα αυτά αποτελούνται από ομαδοποιημένες οργανικές μονομερής χημικές ενώσεις, όπως είναι το αιθυλένιο και το προπυλένιο από τη σύνθεση των οποίων παράγονται πολλά πολυμερή που αποτελούν την πρώτη ύλη για την παρασκευή ενδιάμεσων και τελικών προϊόντων πλαστικού. Μία από τις πιο σημαντικές κατηγορίες πολυμερών που χρησιμοποιούνται σήμερα είναι οι πολυεστέρες, οι οποίοι περιλαμβάνουν τον εστέρα στην βασική τους αλυσίδα. Η ευρεία χρήση των πολυεστέρων εκτείνεται από τα μπουκάλια ανθρακούχων ποτών και νερού έως τις ίνες για πουκάμισα και άλλα προϊόντα ενδυμασίας, τα φωτογραφικά φιλμ και τις κασέτες εγγραφής. Παρόλο που υπάρχουν πολλοί πολυεστέρες, ο όρος «πολυεστέρας» χρησιμοποιείται συνήθως για το υλικό *Polyethylene Terephthalate* (PET), το οποίο χρησιμοποιείται κατά κόρον στη συσκευασία, στις φιάλες εμφύσησης και στη παραγωγή ινών για υφάσματα. Η παρούσα μεταδιπλωματική εργασία εστιάζει στον προγραμματισμό της παραγωγής ρητίνης PET η οποία χρησιμοποιείται για την παρασκευή φιαλών PET.

Παρόλο που η διαδικασία του πολυμερισμού αποτελείται από μία πληθώρα μηχανισμών αντιδράσεων με διαφορετικούς βαθμούς πολυπλοκότητας, από απόψεως οργάνωσης της παραγωγής η βιομηχανική παραγωγή των περισσότερων πολυμερών είναι περίπου ίδια. Συγκεκριμένα, είναι συνηθισμένη πρακτική στις βιομηχανίες πολυμερισμού η λειτουργία σε συνεχή βάση, κατά την οποία παράγονται διάφορα προϊόντα χρησιμοποιώντας τον ίδιο εξοπλισμό. Τα διαφορετικά παραγόμενα προϊόντα τα οποία παράγονται από το ίδιο πολυμερές αλλά έχουν διαφορετικές τελικές ιδιότητες (όπως διαύγεια, χρώμα, μηχανική αντοχή κτλ) ονομάζονται βαθμίδες (grades). Οι τελικές ιδιότητες των βαθμίδων εξαρτώνται από την κατανομή του μοριακού βάρους και τη μετατροπή του μονομερούς σε πολυμερές, χαρακτηριστικά τα οποία καθορίζονται από τις συνθήκες/ρυθμίσεις λειτουργίας.

Κατά την αλλαγή των ρυθμίσεων του εξοπλισμού σε μία βιομηχανία πολυμερισμού απαιτούνται κάποιες διαδικασίες μετάβασης στη διάρκεια των οποίων παράγονται παραπροϊόντα με ιδιότητες εκτός των απαιτούμενων προδιαγραφών. Δεδομένου όμως ότι αυτοί οι χρόνοι μετάβασης μπορεί να είναι μεγάλης διάρκειας, προκύπτει ως πιθανό ενδεχόμενο η παραγωγή μεγάλων ποσοτήτων παραπροϊόντων εκτός των απαιτούμενων προδιαγραφών. Τα παραπροϊόντα θεωρείται ότι έχουν χαμηλότερα ποιοτικά χαρακτηριστικά από αυτά των τελικών προϊόντων και γι' αυτό είτε πωλούνται σε χαμηλότερη τιμή είτε αναμιγνύονται με ποιοτικά ανώτερα προϊόντα με συνέπεια όμως τη μείωση των ποιοτικών χαρακτηριστικών των προϊόντων. Επομένως ο αριθμός

των μεταβάσεων που πραγματοποιούνται κατά τη διάρκεια της παραγωγικής διαδικασίας μιας βιομηχανίας πολυμερισμού αποτελεί σημαντικό αντικείμενο μελέτης για τον καθορισμό του προγράμματος παραγωγής (Terrazas-Moreno et al., 2007).

Σύμφωνα με τους McGehee et al.(2004) οι παράγοντες που επιδρούν στην αύξηση του κόστους της παραγωγικής διαδικασίας σε μια βιομηχανία PET είναι: 1) η ανεπαρκής χρησιμοποίηση του εξοπλισμού, 2) η μη προγραμματισμένη διακοπή της παραγωγικής διαδικασίας, 3) οι διάφορες διαβαθμίσεις προϊόντων που παράγονται με χαμηλά ποιοτικά χαρακτηριστικά, και 4) ο αριθμός των μεταβάσεων κατά τη διάρκεια της παραγωγής. Οι τρεις πρώτοι παράγοντες μπορούν να αντιμετωπιστούν με τη χρήση αποδοτικών πρακτικών επιχειρησιακής έρευνας καθώς και με την ενσωμάτωση αλλαγών στις διαδικασίες, όπως είναι η εφαρμογή αποδοτικών τροποποιήσεων στον εξοπλισμό, η προσεκτική σχεδίαση της προληπτικής συντήρησης, η εγκαθίδρυση ορθολογιστικών προγραμμάτων διαχείρισης ποιότητας, και η ελαχιστοποίηση συστηματικών πηγών μεταβλητότητας του συστήματος. Ο τέταρτος παράγοντας μπορεί να εξεταστεί με την εφαρμογή ενός προσεκτικού και ευφυούς προγραμματισμού παραγωγής.

Η χημική βιομηχανία η οποία μελετάται παράγει μια θερμοπλαστική πολυμερή ρητίνη της οικογένειας των πολυεστέρων. Το συγκεκριμένο προϊόν είναι ένα αδρανές πλαστικό το οποίο δεν είναι επιβλαβές και για το λόγο αυτό χρησιμοποιείται στην τροφοδοσία της αγοράς φιαλών συσκευασίας νερού και αναψυκτικών περισσότερο από 20 χρόνια. Το συγκεκριμένο προϊόν είναι 100% ανακυκλώσιμο και εξαιρετικά ελαφρύ με αποτέλεσμα αφενός τη μείωση των απορριμμάτων και αφετέρου τη μείωση της περιβαλλοντικής επιβάρυνσης κατά τη διαδικασία μεταφοράς, καθώς λόγω μικρού βάρους απαιτείται λιγότερη ενέργεια. Ο Αμερικάνικος Οργανισμός Φαρμάκων και Τροφίμων έχει διασφαλίσει μετά από αυστηρή δοκιμή ότι η μεταφορά της ρητίνης PET είναι ασφαλή τόσο σε σιλοφόρα φορτηγά όσο και χύδην σε κοντέινερ, ενώ μία τρίτη μορφή υπό την οποία πωλείται από την εξεταζόμενη βιομηχανία είναι σε μεγάλους σάκους (σακευμένη μορφή).

2.2 Περιγραφή παραγωγικής διαδικασίας

Η παραγωγική διαδικασία της υπό εξέταση χημικής βιομηχανίας είναι συνεχής και αποτελείται από δύο διαδοχικά παραγωγικά στάδια συνεχούς ροής. Στο πρώτο στάδιο επεξεργασίας της πρώτης ύλης γίνεται ο πολυμερισμός υγρής φάσης (στάδιο POLY - *Liquid State or Melt Polymerization*) ενώ στο δεύτερο στάδιο επεξεργασίας γίνεται η πολυσυμπύκνωση στερεάς φάσης (στάδιο SSP - *Solid State Polymerization or Poly-condensation*) κατά την οποία αυξάνουν το μοριακό βάρος και οι ελαστικές ιδιότητες των προϊόντων. Μια συνηθισμένη πρακτική είναι ο συγχρονισμός των ρυθμών παραγωγής των δύο σταδίων POLY και SSP ώστε η ροή υλικών να είναι σταθερή, ενώ σε αντίθετη περίπτωση ο χώρος ενδιάμεσης αποθήκευσης καταλήγει είτε

γεμάτος (εάν το στάδιο POLY παράγει με μεγαλύτερο ρυθμό από αυτό του σταδίου SSP) είτε άδειος (εάν το στάδιο POLY παράγει με μικρότερο ρυθμό από αυτό του σταδίου SSP). Η ασύγχρονη ροή προϊόντος μεταξύ διαδοχικών παραγωγικών σταδίων, η οποία επιβάλλει τη δυναμική μεταβολή του επιπέδου των ενδιάμεσων αποθεμάτων, έχει μελετηθεί εκτενώς στο πλαίσιο παραγωγής των μη-αξιόπιστων διακριτών εξαρτημάτων όπως παρουσιάζουν οι Dallery και Gershwin (1992). Στα πλαίσια της συνεχούς ροής προϊόντων που πραγματεύεται η παρούσα εργασία, η ροή προϊόντων μεταξύ των δύο παραγωγικών σταδίων είναι συγχρονισμένη. Ωστόσο, ο κοινός ρυθμός παραγωγής μπορεί να μεταβάλλεται σε τακτά χρονικά διαστήματα ώστε να ισούται με τη μέση συνολική αναμενόμενη ζήτηση των προϊόντων, η οποία μπορεί να παρουσιάζει εποχικότητα ή άλλες μεταβολές. Για το σκοπό του βραχυπρόθεσμου προγραμματισμού που παρουσιάζεται στην παρούσα εργασία, ο κοινός ρυθμός παραγωγής θεωρείται σταθερός και ίσος ή σχεδόν ίσος με τη συνολική αναμενόμενη ζήτηση όλων των προϊόντων.

Το τελικό προϊόν που εξέρχεται από το στάδιο SSP χαρακτηρίζεται από δύο ιδιότητες, το χρώμα και το ιξώδες. Το χρώμα του τελικού προϊόντος καθορίζεται στο πρώτο στάδιο της παραγωγικής διαδικασίας (POLY), ενώ το ιξώδες καθορίζεται στο δεύτερο στάδιο της παραγωγικής διαδικασίας (SSP). Το ιξώδες σχετίζεται με το μήκος των πολυμερών αλυσίδων, καθώς μεγάλο μήκος αλυσίδας συνεπάγεται δύσκαμπτο υλικό. Στον Πίνακα 1 παρουσιάζονται οι τέσσερις αποδεκτοί συνδυασμοί χρώματος και ιξώδους οι οποίοι οδηγούν σε τέσσερα τελικά προϊόντα. Η εμφάνιση παύλας (-) στον πίνακα δείχνει συνδυασμό χρώματος και ιξώδους ο οποίος δεν παράγεται διότι δεν υπάρχει ζήτηση για αυτόν. Οι συμβολισμοί L, Gr και D αντιστοιχούν στο ανοιχτόχρωμο ημιέτοιμο προϊόν *Light*, στο γκρι ημιέτοιμο προϊόν *Gray* και στο σκούρο ημιέτοιμο προϊόν *Dark*, ενώ οι συμβολισμοί WG, SD, G και FH αντιστοιχούν στα τελικά προϊόντα *Water Grade*, *Soft Drink*, *Gray*, και *Fast Heat* αντίστοιχα.

Πίνακας 1. Είδη τελικών προϊόντων με βάση τους συνδυασμούς χρώματος και ιξώδους

Ιξώδες	Χρώμα		
	Ανοιχτό (L)	Γκρι (Gr)	Σκούρο (D)
Χαμηλό (< 0.8)	WG	-	-
Υψηλό (> 0.8)	SD	G	FH

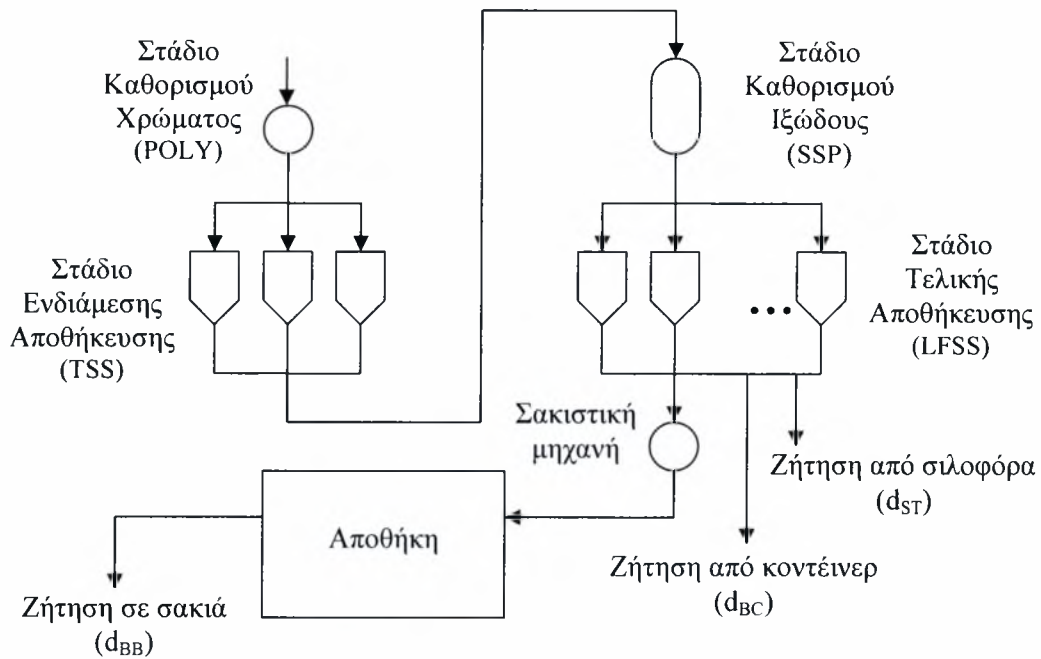
Ειδικότερα:

- Το προϊόν *Water Grade* (WG) χρησιμοποιείται κυρίως στην εμφιάλωση νερού διότι είναι ανοιχτόχρωμο και διαφανές (χαρακτηριστικά διαύγειας). Το χαμηλό ιξώδες είναι επιθυμητό καθώς στην εμφιάλωση νερού δεν απαιτείται υλικό υψηλής αντοχής.

- Το προϊόν *Soft Drink* (SD) χρησιμοποιείται κυρίως στην εμφιάλωση ανοιχτόχρωμων αναψυκτικών. Είναι ανοιχτού χρώματος και υψηλού ιξώδους αφού τα υγρά υψηλής πίεσης απαιτούν πιο ανθεκτικό υλικό.
- Το προϊόν *Grey* (G) προκύπτει ως παραπροϊόν και δεν έχει αυτοτελή ζήτηση με συγκεκριμένη τιμή πώλησης.
- Το προϊόν *Fast Heat* (FH) χρησιμοποιείται κυρίως στην εμφιάλωση σκούροχρωμων ποτών και για το λόγο αυτό είναι σκούρου χρώματος και υψηλού ιξώδους.

Κατά τη διάρκεια της παραγωγικής διαδικασίας είναι απαραίτητο να συμβούν μεταβάσεις από ένα προϊόν σε ένα άλλο με σκοπό την έγκαιρη ικανοποίηση της ζήτησης των πελατών. Οι μεταβάσεις αυτές δεν είναι επιθυμητές διότι απαιτούν σημαντικό χρόνο κατά τη διάρκεια του οποίου παράγονται προϊόντα μη αποδεκτών ποιοτικών προδιαγραφών. Επιπλέον, λόγω τεχνικών περιορισμών, οι μόνες επιτρεπτές μεταβάσεις της παραγωγικής διαδικασίας είναι από το προϊόν WG στο SD, από το SD στο G, και τέλος από το G στο FH και αντίστροφα. Το προϊόν *Gray* (G) είναι ένα ενδιάμεσο προϊόν εκτός προδιαγραφών το οποίο παράγεται αναπόφευκτα κατά τη διάρκεια μετάβασης από το προϊόν SD (ανοιχτού χρώματος) στο προϊόν FH (σκούρου χρώματος) και αντίστροφα. Το συγκεκριμένο προϊόν δεν προτιμάται από τους πελάτες της βιομηχανίας παρόλο που πωλείται σε χαμηλότερη τιμή. Ένα άλλο ενδιάμεσο προϊόν που παράγεται είναι κατά τη διάρκεια της αλλαγής του ιξώδους που συμβαίνει στη μετάβαση της παραγωγικής διαδικασίας από προϊόν WG σε προϊόν SD και αντίστροφα. Η πολιτική που ακολουθεί η βιομηχανία για το συγκεκριμένο ενδιάμεσο προϊόν είναι η διαίρεση της ποσότητας του σε δύο ισόποσα τμήματα, το πρώτο μισό εκ των οποίων αποδίδεται στο προηγούμενο παραγόμενο προϊόν ενώ το δεύτερο μισό αποδίδεται στο επόμενο παραγόμενο προϊόν. Ειδικότερα, το ενδιάμεσο προϊόν αναμιγνύεται με τις υπάρχουσες ποσότητες προϊόντων WG και SD με αποτέλεσμα την ανεπιθύμητη μείωση των ποιοτικών χαρακτηριστικών των προϊόντων.

Όπως φαίνεται στο Σχήμα 1 τα δύο παραγωγικά στάδια POLY και SSP χωρίζονται από ένα στάδιο ενδιάμεσης αποθήκευσης (TSS - *Temporary Storage State*) το οποίο αποτελείται από τρία σιλό. Ένας τρόπος χρησιμοποίησης των σιλό είναι να αφιερωθεί το καθένα από αυτά σε κάποιο συγκεκριμένο είδος ενδιάμεσου προϊόντος οπότε να υπάρχει ένα σιλό με L, ένα με G και ένα με D. Με αυτό τον τρόπο εάν η παραγωγή αλλάξει από L σε D το προϊόν το οποίο εξέρχεται από το στάδιο POLY θα κατευθυνθεί από το σιλό L στο σιλό G κατά τη διάρκεια της μεταβατικής περιόδου και έπειτα στο σιλό D. Ομοίως εάν το ημίτομο προϊόν που εισέρχεται στο στάδιο SSP προέρχεται από αλλαγή SD σε FH το σιλό που θα τροφοδοτούσε το SSP θα άλλαζε από L σε D.



Σχήμα 1. Απεικόνιση της διαδικασίας παραγωγής και αποθήκευσης

Στην πραγματικότητα η κατεύθυνση των προϊόντων που θα εισέλθουν ή εξέλθουν από το στάδιο TSS δεν γίνεται ακαριαία, αλλά απαιτεί κάποιο χρόνο για την αλλαγή των σωληνώσεων μεταφοράς. Για να αποφευχθούν τέτοια προβλήματα η υπό εξέταση βιομηχανία έχει καταλήξει να χρησιμοποιεί κατά τη διάρκεια της παραγωγικής διαδικασίας ένα μόνο από τα τρία σιλό του σταδίου ενδιάμεσης αποθήκευσης ενώ τα άλλα δύο σιλό χρησιμοποιούνται σε περιπτώσεις διακοπής του σταδίου επεξεργασίας SSP εξαιτίας κάποιας βλάβης ή διακοπής της λειτουργίας του για συντήρηση. Επομένως ο χώρος ενδιάμεσης αποθήκευσης TSS αποτελείται από στρώματα τριών (ή και λιγότερων) διαφορετικών προϊόντων L, G και D το ένα πάνω από το άλλο. Σημειώνεται ότι η ανάμιξη που παρουσιάζεται εξαιτίας της στρωματοποίησης αυτής θεωρείται αμελητέα. Η ύπαρξη ενός ενεργού σιλό στο στάδιο TSS απλοποιεί το πρόβλημα αφού το ίδιο σιλό τροφοδοτείται από το στάδιο POLY και το ίδιο τροφοδοτεί το στάδιο SSP. Συγχρόνως όμως περιπλέκεται το πρόβλημα καθώς θα πρέπει να παρακολουθούνται τα στρώματα των διαφορετικών ημιετοιμών προϊόντων μέσα στο ενεργό σιλό του TSS.

Το ημιετοιμο προϊόν που εισέρχεται στο στάδιο SSP απαιτεί ένα αξιόλογο χρονικό διάστημα επεξεργασίας μέχρι να εξέλθει, γεγονός το οποίο επίσης αυξάνει την πολυπλοκότητα του προβλήματος. Με την πάροδο αυτού του χρόνου εξέρχεται το τελικό προϊόν από το στάδιο SSP το οποίο στη συνέχεια οδηγείται σε ένα από τα οχτώ διαφορετικά σιλό τελικής αποθήκευσης (LFSS - *Loading or Final Storage Stage*). Σε αντίθεση με ό,τι συμβαίνει στην ενδιάμεση αποθήκευση, στα σιλό αποθήκευσης τελικών προϊόντων δεν επιτρέπεται η συνύπαρξη διαφορετικών τύπων προϊόντων σε καμία χρονική περίοδο και για να αλλάξει ο τύπος του περιεχόμενου τελικού προϊόντος ενός σιλό θα πρέπει πρώτα αυτό να αδειάσει πλήρως από το προηγούμενο προϊόν.

Το τελικό προϊόν που εξέρχεται από τα σιλό του σταδίου LFSS μπορεί είτε να σακιαστεί από μια σακιστική μηχανή οπότε και αποθηκεύεται σε μορφή σάκου στην αποθήκη, είτε να φορτωθεί απευθείας σε σιλοφόρα φορτηγά ή σε κοντέινερ που ανήκουν στους πελάτες της βιομηχανίας. Επιπρόσθετα, οι σάκοι τελικού προϊόντος που ζητούνται απευθείας από την αποθήκη μεταφέρονται από φορτηγά ειδικού τύπου. Για κάθε ένα από τους τρεις αυτούς τύπους ζήτησης τελικού προϊόντος υπάρχει ένας διαφορετικός ρυθμός τροφοδοσίας.

2.3 Ποιοτικά δεδομένα εισόδου κατά τη μοντελοποίηση

Όπως κάθε παραγωγικό σύστημα, έτσι και η υπό εξέταση χημική βιομηχανία παραγωγής ρητίνης PET υπόκειται σε συγκεκριμένους περιορισμούς οι οποίοι πηγάζουν από τα χαρακτηριστικά της γνωρίσματα. Οι περιορισμοί αυτοί αφορούν στην εξάρτηση που υπάρχει μεταξύ των ρυθμίσεων των διαφόρων σταδίων της παραγωγικής διαδικασίας ώστε να παράγονται τυποποιημένα προϊόντα, στους περιορισμούς δυναμικότητας της παραγωγής και αποθήκευσης ημιετοίμων και τελικών προϊόντων, και τέλος στους περιορισμούς ικανοποίησης της ζήτησης των προϊόντων οι οποίοι προκύπτουν με βάση τόσο την εφαρμοζόμενη πολιτική της βιομηχανίας όσο και από την εκάστοτε κατάσταση του συστήματος.

Ένα από τα χαρακτηριστικά ιδιαίτερης σημασίας για την υπό εξέταση χημική βιομηχανία είναι ότι η αλληλουχία των μεταβάσεων ενός σταδίου επεξεργασίας εξαρτάται από την κατάσταση μεταβάσεων του άλλου σταδίου. Πιο αναλυτικά και σύμφωνα με τον Πίνακα 1, μια μετάβαση των ρυθμίσεων του παραγωγικού σταδίου SSP από υψηλό σε χαμηλό ιξώδες δεν επιτρέπεται εάν το χρώμα της πρώτης ύλης που επεξεργάζεται στο στάδιο αυτό είναι το G ή το FH. Το γεγονός αυτό κάνει τη μοντελοποίηση πιο περίπλοκη διότι το χρώμα της διερχόμενης ποσότητας υλικών που επεξεργάζεται το στάδιο SSP σε οποιαδήποτε χρονική περίοδο έχει καθοριστεί αρκετές περιόδους ωρίτερα στο στάδιο POLY.

Σημαντικό βήμα αποτελεί και η αρχικοποίηση των καταστάσεων των παραγωγικών σταδίων POLY και SSP καθώς και ο καθορισμός των αρχικών ποσοτήτων των αποθεμάτων στους χώρους TSS, LFSS και στην αποθήκη (Warehouse). Αυτό συμβαίνει διότι τόσο οι αρχικές ρυθμίσεις της παραγωγικής διαδικασίας όσο και ο όγκος των αποθεμάτων ημιετοίμων και τελικών προϊόντων μπορούν να επηρεάσουν άμεσα τον απαιτούμενο αριθμό αλλαγών της παραγωγικής διαδικασίας κατά τη διάρκεια του χρονικού ορίζοντα εξέτασης με τον εξής τρόπο: αφενός, όσο οι αρχικές ρυθμίσεις της παραγωγικής διαδικασίας συμπίπτουν με αυτές που προκύπτουν με βάση τη ζήτηση και τα υπάρχοντα αποθέματα, τόσο μειώνονται οι αλλαγές που θα χρειαστεί να πραγματοποιηθούν κατά τη διάρκεια του ορίζοντα ελέγχου. Αφετέρου, όσο αυξάνονται οι ποσότητες των ημιετοίμων και τελικών προϊόντων, τόσο αυξάνεται η ευελιξία της βιομηχανίας απέναντι στη μεταβλητότητα της ζήτησης.

Ένα ακόμη δεδομένο που συνδέεται με τα προηγούμενα και που πρέπει να εισάγεται κατά τη μορφοποίηση μαθηματικών μοντέλων απεικόνισης πραγματικών παραγωγικών συστημάτων είναι ο ορισμός των επιθυμητών επιπέδων των αποθεμάτων κατά την ολοκλήρωση του χρονικού ορίζοντα ελέγχου. Τα διατηρούμενα αποθέματα διευκολύνουν την ικανοποίηση της ζήτησης σε περιόδους όπου αυτή ξεπερνά την υφιστάμενη εγκατεστημένη δυναμικότητα, από την άλλη όμως επιβαρύνουν την παραγωγική διαδικασία η οποία θα πρέπει να λαμβάνει υπόψη και την αναπλήρωση των αποθεμάτων αυτών. Η προσεκτική μελέτη των επιθυμητών επιπέδων διατηρούμενων αποθεμάτων μπορεί να οδηγήσει σε σημαντικά οφέλη.

Ίσως το πιο σημαντικό στοιχείο εισόδου για την ακρίβεια των αποτελεσμάτων οποιουδήποτε μοντέλου απεικόνισης πραγματικού παραγωγικού συστήματος είναι η ακρίβεια των προβλέψεων των μελλοντικών ζητήσεων για κάθε είδος και μορφή προϊόντος κατά τη διάρκεια του ορίζοντα προγραμματισμού. Σύμφωνα με τους Liberopoulos et al.^a (2009), η πρακτική που χρησιμοποιεί η συγκεκριμένη χημική βιομηχανία είναι να παραλαμβάνει τις παραγγελίες των πελατών της από την προηγούμενη εβδομάδα για την επόμενη, με αποτέλεσμα στο τέλος κάθε εβδομάδας να γνωρίζει με μεγάλη ακρίβεια τη ζήτηση για χρονικό διάστημα μίας εβδομάδας. Για το λόγο αυτό, έκριναν σκόπιμο να ορίσουν το μήκος του χρονικού ορίζοντα προγραμματισμού ίσο με μία εβδομάδα στη διάρκεια της οποίας η βιομηχανία παίρνει διάφορες αποφάσεις προγραμματισμού όπως το ποιο είδος προϊόντος θα παραχθεί και πότε, πότε θα αλλάξει το χρώμα ή θα γίνει η μετάβαση ιξώδους, σε ποιο σιλό του σταδίου LFSS θα αποθηκευτεί το τελικό προϊόν μετά την έξοδο του από το στάδιο SSP, από ποιο σιλό του LFSS θα σακιαστεί προϊόν και πόσο, και από ποιο σιλό του LFSS θα φορτωθούν τα σιλοφόρα φορτηγά ή τα κοντέινερ ικανοποιώντας την αντίστοιχη ζήτηση.

Επιπρόσθετα όπως αναφέρθηκε προηγουμένως παρόλο που ο χρόνος μετάβασης από χαμηλό σε υψηλό ιξώδες (και αντίστροφα) είναι σημαντικός οι Liberopoulos et al.^a (2009) θεωρούν ότι μία μετάβαση πραγματοποιείται ακαριαία στη μέση του χρόνου μετάβασης ώστε να επιτευχθεί η ταξινόμηση του είδους των προϊόντων που παράγονται από το στάδιο SSP. Ακόμη το γεγονός ότι ο αντιδραστήρας του σταδίου SSP έχει πεπερασμένη χωρητικότητα επιφέρει επιπλέον καθυστέρηση στην επεξεργασία του προϊόντος στο στάδιο SSP και κάνει το μοντέλο ακόμη πιο πολύπλοκο.

Ένα άλλο ιδιαίτερο χαρακτηριστικό του υπό εξέταση παραγωγικού συστήματος είναι η δυνατότητα αποθήκευσης ποσοτήτων ημετοίμων προϊόντων με διαφορετικά χρώματα σε επάλληλα στρώματα στο ενεργό σιλό του σταδίου προσωρινής αποθήκευσης TSS. Το γεγονός αυτό δημιουργεί την ανάγκη ιχνηλασιμότητας των ενδιάμεσων προϊόντων που περιέχονται στο ενεργό σιλό με αποτέλεσμα την περαιτέρω επιβάρυνση των ελαχίστων απαιτήσεων κατά τη μοντελοποίηση. Συνεπώς, οι απαιτήσεις αποθήκευσης για το συγκεκριμένο σιλό χαρακτηρίζονται ως μεικτού και ευέλικτου χώρου ενδιάμεσης αποθήκευσης με πεπερασμένη χωρητικότητα (Liberopoulos et al.^a, 2009).

Τα παραπάνω χαρακτηριστικά καθιστούν πολύπλοκη τη μαθηματική μορφοποίηση κατά τη διαδικασία μοντελοποίησης αλλά συγχρόνως παρέχουν ισχυρό κίνητρο για επιστημονική έρευνα. Το πραγματικό κίνητρο ανάπτυξης ενός τέτοιου μοντέλου πηγάζει από το γεγονός ότι πρόκειται για μία πραγματική εφαρμογή επιχειρησιακής έρευνας και ειδικότερα προγραμματισμού παραγωγής. Στη παρούσα μεταδιπλωματική εργασία γίνεται μία σύντομη αναφορά στο μοντέλο MILP των Liberopoulos et al.^a (2009) ενώ δημιουργείται και ένα προσομοιωτικό μοντέλο στο οποίο εμπεριέχονται όλοι οι παραπάνω παράγοντες και παρουσιάζεται η αποδοτικότητά του υπό την ίδια ζήτηση με αυτή του μοντέλου MILP. Η συγκριτική ανάλυση που θα ακολουθήσει στα επόμενα κεφάλαια παρέχει σημαντικές πληροφορίες για τις δύο εναλλακτικές μεθοδολογίες.

2.4 Ποσοτικά δεδομένα εισόδου κατά τη μοντελοποίηση

Ο ρυθμός παραγωγής τελικών προϊόντων της χημικής βιομηχανίας είναι ίσος με 200 τόνους/24hr. Η τιμή αυτή μπορεί και να μεταβληθεί μέσα στο εύρος από 180 έως 220 τόνους/24hr ωστόσο η παραγωγή 200 τόνων τελικού προϊόντος/24hr προτιμάται τις περισσότερες φορές από τη συγκεκριμένη βιομηχανία ως μέση τιμή ομαλής λειτουργίας. Η αύξηση ή μείωση του ημερήσιου ρυθμού παραγωγής γίνεται σε αραιά χρονικά διαστήματα και για σχετικά μεγάλες περιόδους, με γνώμονα την ικανοποίηση της ζήτησης, την αναπλήρωση των αποθεμάτων ημιετοιμών και τελικών προϊόντων όταν φτάσουν σε πολύ χαμηλά επίπεδα, τη δημιουργία αποθεμάτων ώστε να τεθούν οι μηχανές εκτός λειτουργίας για συντήρηση και διάφορους άλλους παραπλήσιους παράγοντες.

Η διάρκεια των απαιτούμενων ρυθμίσεων στο στάδιο POLY για την αλλαγή χρώματος από L σε D και αντίστροφα απαιτεί 4 ώρες, ενώ η διάρκεια των απαιτούμενων ρυθμίσεων στο στάδιο SSP για την αλλαγή ιξώδους από χαμηλό σε υψηλό και αντίστροφα διαρκεί 24 ώρες. Οι διάρκειες και των δύο μεταβάσεων είναι ακέραια πολλαπλάσια της 4hr περιόδου και για το λόγο αυτό ως χρονικό βήμα προγραμματισμού (εφεξής χρονική περίοδος) θεωρείται το 4hr το οποίο διαιρεί και τη διάρκεια των 8 ωρών μίας βάρδιας αλλά και τη διάρκεια των 24 ωρών μίας παραγωγικής ημέρας. Επομένως οι χρονικές περίοδοι που απαιτούνται για την αλλαγή του χρώματος (συμβολίζεται με B) είναι μία ($B=1$) ενώ για την αλλαγή του ιξώδους (συμβολίζεται με F) είναι έξι ($F=6$). Σημειώνεται ότι ο αντίστοιχος ρυθμός παραγωγής ανά χρονική περίοδο που προκύπτει από τη διαίρεση των 200 τόνων ανά παραγωγική ημέρα με τις 6 περιόδους μία παραγωγικής ημέρας, είναι ίσος με 33.33 τόνους/4hr.

Το κάθε σιλό αποθήκευσης ημιετοιμού προϊόντος του σταδίου TSS έχει χωρητικότητα 430 τόνους, συνήθως όμως χρησιμοποιείται μόνο ένα σιλό από το οποίο αξιοποιείται η μισή του χωρητικότητα. Επομένως ως ομαλή κατάσταση του σταδίου TSS ορίζεται η ύπαρξη σε αυτό 200 τόνων ημιετοιμών προϊόντων, κατάσταση η οποία μπορεί να χρησιμοποιηθεί στην αρχικοποίηση ενός χρονικού ορίζοντα προγραμματισμού που ξεκινάει με κανονικές συνθήκες ημιετοιμών

προϊόντων. Εξαιτίας του ότι ο ρυθμός παραγωγής είναι ίσος με 33.33 τόνους/4hr, ο χώρος του κάθε σιλό χωρίζεται σε 12 διακριτές θέσεις αποθήκευσης ($N=12$ /σιλό TSS) καθεμία εκ των οποίων έχει χωρητικότητα ίση με 33.33 τόνους. Ομοίως χωρίζεται και το παραγωγικό στάδιο SSP, το οποίο έχει χωρητικότητα διερχόμενου προϊόντος ίση με 200 τόνους, σε 6 ισοδύναμες θέσεις ($M=6$) καθεμία από τις οποίες έχει χωρητικότητα ίση με 33.33 τόνους.

Κατά τη λειτουργία της βιομηχανίας, η παραγωγή προγραμματίζεται σε εβδομαδιαία βάση, επειδή σύμφωνα με τους Liberopoulos et al.^a (2009) στο τέλος κάθε εβδομάδας είναι γνωστή με βεβαιότητα η ζήτηση της επόμενης. Επομένως ο ορίζοντας προγραμματισμού τίθεται ίσος με μία εβδομάδα και ισοδυναμεί με 42 χρονικές περιόδους (24hrs/day, 7days/week). Θεωρώντας ότι το παραγωγικό σύστημα εκκινεί από απόψεως ημιετοιμών προϊόντων σε ομαλή κατάσταση, τότε χρησιμοποιούνται μόνο 6 από τις διακριτές θέσεις αποθήκευσης του TSS. Δεδομένου ακόμη ότι και το $M=6$, προκύπτει ότι η πρώτη περίοδος του ορίζοντα προγραμματισμού είναι η περίοδος 13 ($=N+M+1$) καθώς το τι προϊόν θα αποδώσει στην έξοδο η βιομηχανία τις πρώτες 12 περιόδους εξαρτάται από τις προηγούμενες αποφάσεις που γέμισαν το σύστημα με τη διερχόμενη ποσότητα που ενυπάρχει και θα εξαχθεί μέχρι τη δωδέκατη περίοδο. Αντίστοιχα, η τελευταία περίοδος T του ορίζοντα προγραμματισμού είναι η περίοδος 54 ($=N+M+42$).

Η χωρητικότητα της αποθήκης όπου αποθηκεύονται τελικά προϊόντα σε σακευμένη μορφή είναι ίση με 3.500 τόνους ενώ η χωρητικότητα του καθενός από τα οκτώ σιλό αποθήκευσης τελικών προϊόντων στο στάδιο LFSS είναι 430 τόνοι. Όπως προαναφέρθηκε σε αντίθεση με το τι συμβαίνει στο στάδιο TSS, σε αυτό το στάδιο δεν επιτρέπεται η αποθήκευση δύο διαφορετικών τύπων προϊόντων στο ίδιο σιλό, επομένως κάθε σιλό περιέχει έναν τύπο προϊόντος ή κανένα (περιεχόμενη ποσότητα ίση με το μηδέν). Προκειμένου να αλλάξει ο τύπος του τελικού προϊόντος που περιέχει ένα σιλό LFSS θα πρέπει η περιεχόμενη ποσότητα του προηγούμενου προϊόντος στο σιλό να είναι μικρότερη ή ίση από ένα τόνο και αυτό διότι η αλλοίωση των ποιοτικών χαρακτηριστικών του νέου εισερχόμενου προϊόντος από τη μικρή ποσότητα του προηγούμενου προϊόντος είναι ασήμαντη.

Ο μέγιστος ρυθμός ικανοποίησης της εισερχόμενης ζήτησης σε μορφή σιλοφόρου φορτηγού για οποιοδήποτε προϊόν και οποιοδήποτε σιλό είναι ίσος με 224 τόνους/4hr ενώ ο αντίστοιχος μέγιστος ρυθμός για τα κοντέινερ είναι σαφώς μικρότερος και ίσος με 69,2 τόνους/4hr. Η διαδικασία σακιάσματος τελικών προϊόντων υπόκειται στον περιορισμό της δυναμικότητας της μηχανής σακιάσματος με μέγιστο ρυθμό σακιάσματος 40 τόνους/4hr. Σημειώνεται ότι η χημική βιομηχανία διαθέτει μόνο μία μηχανή σακιάσματος η οποία μπορεί να σακιάσει εξερχόμενο προϊόν από οποιοδήποτε σιλό LFSS, όμως μόνο από ένα κάθε χρονική στιγμή (δεν επιτρέπεται ταυτόχρονο σάκιασμα). Στη συνέχεια οι σάκοι τοποθετούνται στην αποθήκη από όπου και εξέρχονται χωρίς άνω όριο στο μέγιστο ρυθμό εξόδου.

Στα ποσοτικά δεδομένα εισόδου κατά τη μοντελοποίηση περιέχεται και η ζήτηση των τελικών προϊόντων σε οποιαδήποτε μορφή. Για λόγους εξοικονόμησης χώρου δεν παρουσιάζονται οι τιμές της ζήτησης ανά μορφή, περίοδο και τύπο προϊόντος παρά μόνο οι μέσες τιμές και οι τυπικές αποκλίσεις της ημερήσιας ζήτησης για κάθε προϊόν και μορφή ζήτησης οι οποίες προέκυψαν από την επεξεργασία αυτών των δεδομένων, όπως φαίνεται στον Πίνακα 2 (Liberopoulos et al.^a, 2009). Σημειώνεται ότι η συνολική μέση ημερήσια ζήτηση για όλα τα προϊόντα και τις μορφές ζήτησης είναι περίπου 195 τόνοι, ποσότητα η οποία είναι σχεδόν ίση με τον επιδιωκόμενο ρυθμό παραγωγής των 200 τόνων/day. Επίσης, σημειώνεται ότι το τελικό προϊόν SD είναι το μόνο που ζητείται σε σακιασμένη μορφή ενώ το τελικό προϊόν G επειδή είναι μη τυποποιημένων ιδιοτήτων και πωλείται σε χαμηλότερη τιμή, θεωρείται ότι δεν παρουσιάζει ζήτηση. Υπενθυμίζεται ότι στόχο της παρούσας μελέτης αποτελεί η μείωση ή/και ελαχιστοποίηση των απαιτούμενων αλλαγών χρώματος και ιξώδους κατά τη διάρκεια της παραγωγικής διαδικασίας.

Πίνακας 2. Μέση τιμή και τυπική απόκλιση της ημερήσιας ζήτησης για κάθε τύπο προϊόντος και κάθε μορφή ζήτησης

Προϊόν <i>j</i>	1		2		4		Άθροισμα μέσων όρων
	Μέση τιμή	Τυπική απόκλιση	Μέση τιμή	Τυπική απόκλιση	Μέση τιμή	Τυπική απόκλιση	
Ζήτηση σε σιλοφόρα	25.82	41.00	0	0	17.67	29.52	43.49
Ζήτηση σε κοντέινερ	29.96	41.52	0	0	16.13	19.63	46.09
Ζήτηση σε σακιά	25.27	32.15	45.43	58.45	34.37	46.77	105.07
Σύνολο	81.05		45.43		68.17		194.65

Σημαντικό δεδομένο εισόδου κατά τη διαδικασία της μοντελοποίησης αποτελεί και το ύψος των αρχικών αποθεμάτων ημιετοιμών και τελικών προϊόντων με τα οποία εκκινεί η χημική βιομηχανία στην αρχή του ορίζοντα προσομοίωσης. Στην αρχή κάθε εβδομάδας (ορίζοντας προγραμματισμού) εκτός της πρώτης, η κατάσταση του συστήματος τίθεται ίση με την κατάσταση του συστήματος στο τέλος της προηγούμενης εβδομάδας. Στην πρώτη εβδομάδα, οι αρχικές ποσότητες των αποθεμάτων ημιετοιμών και τελικών προϊόντων τίθενται ίσες με τις αντίστοιχες αυτών που ορίστηκαν από τους Liberopoulos et al.^a (2009) για το ίδιο πρόβλημα. Στους Πίνακες 3 και 4 που ακολουθούν παρουσιάζονται αυτές οι τιμές.

Πίνακας 3. Αρχικές ποσότητες στα σιλό του σταδίου LFSS

Σιλό q	1	2	3	4	5	6	7	8
Προϊόν j	WG	-	WG	FH	-	FH	FH	G
Αρχική ποσότητα	400	-	400	400	-	400	350	150

Πίνακας 4. Αρχικές ποσότητες στην αποθήκη

Προϊόν j	WG	SD	G	FH
Αρχική ποσότητα	300	1100	0	500

Τέλος, σημαντικά δεδομένα εισόδου κατά τη διαδικασία της μοντελοποίησης αποτελούν τα επιθυμητά αποθέματα ασφαλείας των ημιτοίμων και τελικών προϊόντων, η διαδικασία καθορισμού των οποίων περιγράφεται από τους Liberopoulos et al.^a (2009). Εν συντομία, προκειμένου να ληφθούν έγκυρες τιμές των επιθυμητών αποθεμάτων ασφαλείας μοντελοποιήθηκε αρχικά το πρόβλημα προγραμματισμού ως ένα Stochastic Economic Lot Sizing Problem (SELSP) το οποίο επιλύθηκε αριθμητικά. Έπειτα, χρησιμοποιώντας τα αποτελέσματα της λύσης του SELSP υπολογίστηκαν τα επιθυμητά επίπεδα των στοκ ασφαλείας. Τα αποτελέσματα αυτών των υπολογισμών παρουσιάζονται στον Πίνακα 5.

Πίνακας 5. Στοκ ασφαλείας στο στάδιο LFSS και στην αποθήκη

Προϊόν j	WG	SD	G	FH
Στοκ ασφαλείας σε σιλό	450	0	0	450
Στοκ ασφαλείας σε αποθήκη	250	880	0	450

3 ΓΡΑΜΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

3.1 Γενικά

Σε μία παραγωγική μονάδα λαμβάνονται συνεχώς σημαντικές αποφάσεις βραχυπρόθεσμου χαρακτήρα όπως ο προγραμματισμός της παραγωγής των προϊόντων, οι ποσότητες προϊόντων που θα παραχθούν και ο χρόνος παραγωγής τους (ταχύτητα λειτουργίας των μηχανημάτων), με στόχο τη βελτιστοποίηση της αντικειμενικής συνάρτησης στο τέλος του χρονικού ορίζοντα μελέτης. Σε κάθε περίοδο οι συνθήκες του προβλήματος οι οποίες επηρεάζουν τη βέλτιστη λύση μεταβάλλονται κυρίως λόγω της αυξημένης μεταβλητότητας της ζήτησης προϊόντων. Πιο συγκεκριμένα, οι απαιτήσεις για την παραγωγή διαφόρων προϊόντων εξαρτώνται από πολλούς εσωτερικούς και εξωτερικούς παράγοντες, οι κυριότεροι εκ των οποίων είναι:

Εσωτερικοί παράγοντες:

- Επίπεδα αποθεμάτων και επιθυμητά επίπεδα αποθεμάτων.
- Δυναμικότητα παραγωγής και επενδύσεις.
- Δυναμικότητα αποθήκευσης πρώτων υλών, ημιετοιμών και τελικών προϊόντων.
- Τεχνολογικοί περιορισμοί.
- Λειτουργικά κόστη παραγωγής.
- Διαχωρισμός των πελατών σε κλάσεις αναλόγως της προσφοράς τους στα έσοδα.
- Τιμολογιακή πολιτική και πολιτικές προώθησης και προσφορών.

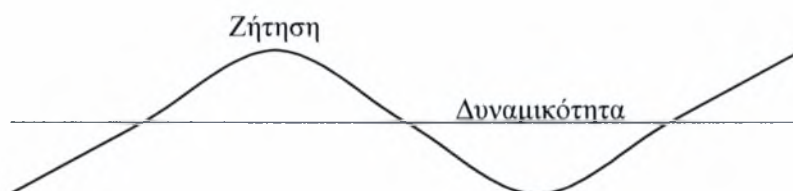
Εξωτερικοί παράγοντες:

- Σταθερότητα ή μεταβλητότητα της αγοράς και της ζήτησης.
- Έκτακτες παραγγελίες.
- Άμεση ικανοποίηση της ζήτησης ή δυνατότητα ικανοποίησής της με καθυστέρηση.
- Ανταγωνισμός των επιχειρήσεων.
- Απρόβλεπτοι παράγοντες και γενικότερη πορεία της οικονομίας.

3.2 Αναγκαιότητα ύπαρξης μεθόδων βελτιστοποίησης

Κατά την κατάστρωση ενός προγράμματος παραγωγής πρέπει να λαμβάνονται υπόψη όσο το δυνατόν περισσότεροι από τους παραπάνω παράγοντες ώστε οι αλληλεπιδράσεις των μεταβλητών και των διαδικασιών να μελετώνται με λεπτομέρεια και να προκύπτει ένα πρόγραμμα παραγωγής που να βελτιστοποιεί τη λειτουργία του συστήματος. Η αναγκαιότητα αυτή απαιτεί μία μέθοδο βελτιστοποίησης η οποία να τους λαμβάνει υπόψη, να τους συνδυάζει και να βελτιστοποιεί μία αντικειμενική συνάρτηση. Ακόμη από τη χρήση μεθόδων βελτιστοποίησης μπορούν να σχεδιαστούν πολιτικές διαχείρισης κρίσεων που ενδεχομένως να αντιμετωπίσει μία επιχείρηση ώστε η επάνοδός της σε ομαλή λειτουργία να είναι η ταχύτερη δυνατή και με το μικρότερο κόστος.

Η διαδικασία βελτιστοποίησης απαιτεί την ύπαρξη αλγορίθμων οι οποίοι μέσω μίας περίπλοκης μαθηματικής επαναλαμβανόμενης διαδικασίας πολλαπλών βημάτων να αποδίδουν μία λύση. Η ποιότητα της λύσης εξαρτάται από τον αλγόριθμο βελτιστοποίησης ο οποίος μπορεί να είναι είτε ευρετικός είτε αναλυτικός. Οι ευρετικοί αλγόριθμοι επίλυσης έχουν το πλεονέκτημα του σημαντικά μικρότερου απαιτούμενου υπολογιστικού χρόνου σε σχέση με τον αντίστοιχο των αναλυτικών μεθόδων, παρουσιάζουν όμως αμφίβολη ποιότητα του εξαγόμενου αποτελέσματος καθώς η συνολικά βέλτιστη λύση παραμένει άγνωστη. Για παράδειγμα, θεωρείται ότι η ζήτηση των προϊόντων μίας επιχείρησης ακολουθεί το Διάγραμμα 1.



Διάγραμμα 1. Εποχικό μοντέλο ζήτησης σε σχέση με τη δυναμικότητα (ποιοτική απεικόνιση)

Κατά τη διάρκεια των περιόδων χαμηλής ζήτησης η δυναμικότητα παραγωγής υπερβαίνει τη ζήτηση, ενώ κατά τις περιόδους υψηλής ζήτησης είναι λιγότερη από αυτή. Προκύπτουν ευλόγως δύο δυνατές εναλλακτικές στρατηγικές σχεδίασης δυναμικότητας: Σύμφωνα με την πρώτη, αυξάνεται η δυναμικότητα ώστε να διατηρούνται ελάχιστα αποθέματα λόγω πλεονάσματος δυναμικότητας. Σύμφωνα με τη δεύτερη, διατηρείται σταθερή δυναμικότητα και δημιουργούνται αποθέματα κατά τη διάρκεια των περιόδων χαμηλής ζήτησης τα οποία πωλούνται στις περιόδους υψηλής ζήτησης. Οι δύο αυτές στρατηγικές παρουσιάζουν συγκεκριμένα μειονεκτήματα και πλεονεκτήματα. Η πρώτη ενώ από τη μία ελαχιστοποιεί τα αποθέματα και μπορεί ευκολότερα να καλύψει έκτακτες ανάγκες, από την άλλη παρουσιάζει επιπρόσθετα κόστη τα οποία σχετίζονται είτε με τη δέσμευση κεφαλαίων σε δυναμικότητα και τη συντήρηση, είτε με τη συνεχή μεταβολή του προσωπικού με επιβαρυντικές προεκτάσεις που προκύπτουν από τα κόστη πρόσληψης και εκπαίδευσης νέου προσωπικού, τις πρόσθετες αμοιβές αποζημίωσης, και τη μείωση παραγωγικότητας λόγω μειωμένου ηθικού του προσωπικού. Η δεύτερη έχει υψηλό κόστος διατήρησης αποθεμάτων με αποφυγή των κοστοβόρων αλλαγών δυναμικότητας και προσωπικού.

Το συνολικό κόστος αυτών των πολιτικών αποτελείται κυρίως από δύο στοιχεία, το κόστος αποθήκευσης και το κόστος διατήρησης δυναμικότητας και μεταβολής του προσωπικού. Οι δύο αυτές στρατηγικές είναι δύο μονοσήμαντοι τρόποι βελτιστοποίησης οι οποίοι εστιάζουν στην βελτιστοποίηση ενός στοιχείου κόστους αγνοώντας την επίδραση του άλλου. Μια αποδοτικότερη βελτιστοποίηση θα πρέπει να βρει ισορροπία μεταξύ αυτών των δύο παραμέτρων κόστους. Μία τέτοια προσέγγιση δεν είναι δυνατό να επιλύσει το πρόβλημα με παραδοσιακές μεθόδους αλλά απαιτεί ένα σύγχρονο, ισχυρό λογισμικό περιβάλλον.

Τα περισσότερα προβλήματα σχεδίασης συνολικής παραγωγής ταιριάζουν πολύ καλά σε μία δομή μοντέλου ΓΠ. Η βελτιστοποίηση οδηγεί είτε σε ελαχιστοποίηση είτε στη μεγιστοποίηση της αντικειμενικής συνάρτησης σε ένα χρονικό ορίζοντα, υπό κάποιους περιορισμούς οι οποίοι είναι πρωτίστως περιορισμοί συνέχειας. Πλεονέκτημα του ΓΠ είναι ότι μπορεί να διαχειριστεί προβλήματα σχετικά υψηλού μεγέθους και πολυπλοκότητας που μπορεί να παρουσιαστούν στη σχεδίαση συνολικής παραγωγής. Ο χρόνος επίλυσης αποτελεί σημαντική παράμετρο της διαδικασίας επίλυσης και εξαρτάται πέραν των αλγορίθμων επίλυσης, των ικανοτήτων των Η/Υ, το πλήθος των μεταβλητών και την πολυπλοκότητα του μοντέλου, και από το είδος του μοντέλου ΓΠ (γραμμικός προγραμματισμός ή μεικτός ακέραιος γραμμικός προγραμματισμός).

3.3 Μαθηματικός προγραμματισμός

Τα προβλήματα βελτιστοποίησης αναφέρονται στη μεγιστοποίηση ή ελαχιστοποίηση μίας ή περισσότερων αντικειμενικών συναρτήσεων ενός αριθμού μεταβλητών που πρέπει να ικανοποιούν ένα σύνολο περιορισμών. Τα προβλήματα βελτιστοποίησης διακρίνονται σε προβλήματα κλασικής βελτιστοποίησης και προβλήματα μαθηματικού προγραμματισμού, η βασική διαφορά μεταξύ των οποίων είναι το είδος και η φύση των περιορισμών. Ειδικότερα, στα προβλήματα κλασικής βελτιστοποίησης το σύνολο των περιορισμών αποτελείται μόνο από εξισώσεις και οι μέθοδοι επίλυσής τους που ονομάζονται κλασικές μέθοδοι βελτιστοποίησης, χρησιμοποιούν εργαλεία κλασικής μαθηματικής ανάλυσης, όπως ο διαφορικός και ο ολοκληρωτικός λογισμός. Αντιθέτως, στα προβλήματα μαθηματικού προγραμματισμού περιλαμβάνονται περιορισμοί τόσο εξισώσεων όσο και ανισώσεων. Σημειώνεται ότι μεταξύ των περιορισμών των προβλημάτων μαθηματικού προγραμματισμού περιλαμβάνονται και οι περιορισμοί μη αρνητικότητας των μεταβλητών οι οποίοι αν και τόσο βασικοί σε προβλήματα οικονομικού προγραμματισμού, δεν εμπεριέχονται στα προβλήματα κλασικής βελτιστοποίησης.

Οι κλασικές μέθοδοι βελτιστοποίησης οι οποίες είναι γνωστές εδώ και 150 χρόνια αναπτύχθηκαν για τη βελτιστοποίηση πολλών προβλημάτων που παρουσιάστηκαν κυρίως στις φυσικές επιστήμες, στη μηχανολογία και τη γεωμετρία. Αργότερα χρησιμοποιήθηκαν ευρέως από τις οικονομικές επιστήμες για την εξαγωγή συμπερασμάτων ποιοτικής φύσεως της κλασικής θεωρίας παραγωγής και κατανάλωσης. Τα τελευταία σαράντα χρόνια παρουσιάστηκαν πολλά καινούρια και ενδιαφέροντα πρακτικά προβλήματα οικονομικής βελτιστοποίησης. Τα προβλήματα αυτά ονομάστηκαν προβλήματα μαθηματικού προγραμματισμού και η επίλυσή τους δε μπορούσε να γίνει με τις τεχνικές των κλασικών μεθόδων βελτιστοποίησης. Αναπτύχθηκαν έτσι καινούριες μέθοδοι βελτιστοποίησης για την επίλυση των προβλημάτων οι οποίες και ονομάστηκαν μέθοδοι μαθηματικού προγραμματισμού (methods of mathematical programming).

3.4 Βασικά χαρακτηριστικά των προβλημάτων μαθηματικού προγραμματισμού

Ο μαθηματικός προγραμματισμός αναφέρεται γενικά στην επίλυση προβλημάτων κατανομής περιορισμένων οικονομικών πόρων ή μέσων σε ένα σύνολο δραστηριοτήτων ώστε να βελτιστοποιείται (μεγιστοποίηση ή ελαχιστοποίηση) μία ή περισσότερες συναρτήσεις στόχου. Τα κύρια χαρακτηριστικά στοιχεία κάθε μοντέλου μαθηματικού προγραμματισμού είναι τα εξής:

- Υπάρχει ένα σύνολο δραστηριοτήτων έστω $J = \{1, 2, \dots, n\}$ οι οποίες μπορεί να αφορούν την παραγωγή προϊόντων, την παροχή διαφόρων υπηρεσιών ή οτιδήποτε άλλο. Σε κάθε δραστηριότητα j του συνόλου J αντιστοιχεί μία μεταβλητή έστω x_j η τιμή της οποίας προσδιορίζεται από την επίλυση του συγκεκριμένου προβλήματος του μαθηματικού προγραμματισμού και υποδεικνύει το ύψος εκτέλεσης ή παραγωγής της αντίστοιχης δραστηριότητας j . Οι μεταβλητές αυτές ονομάζονται και μεταβλητές απόφασης ή μεταβλητές πολιτικής.
- Υπάρχει ένα σύνολο οικονομικών πόρων ή μέσων $I = \{1, 2, \dots, m\}$ που διατίθενται σε περιορισμένες ποσότητες για την εκτέλεση των δραστηριοτήτων. Οι πόροι αυτοί μπορεί να είναι πρώτες ύλες, βοηθητικά υλικά, μηχανολογικός εξοπλισμός, αποθηκευτικοί χώροι, εργατοώρες και γενικά οποιοσδήποτε συντελεστής παραγωγής ο οποίος είναι απαραίτητος για την εκτέλεση των δραστηριοτήτων j . Σε κάθε συντελεστή παραγωγής i του I αντιστοιχεί μία περιοριστική σταθερά έστω b_i η τιμή της οποίας μας δείχνει το ύψος της διατιθέμενης ποσότητάς του. Ακόμη σε κάθε συντελεστή i αντιστοιχεί και μία περιοριστική συνάρτηση έστω $g_i(x_1, x_2, \dots, x_n)$ η οποία συνδέεται με τη b_i με σχέση ισότητας ή ανισότητας. Οι περιοριστικές αυτές σχέσεις ονομάζονται τεχνολογικοί περιορισμοί γιατί εκφράζουν τις διάφορες τεχνολογικές σχέσεις παραγωγής. Τους τεχνολογικούς περιορισμούς συμπληρώνουν οι περιορισμοί πολιτικής με τους οποίους εκφράζονται αποφάσεις σχετικές με τα επιθυμητά ανώτατα ή κατώτατα όρια στις τιμές των μεταβλητών ή περιορισμών, στην τιμή της αντικειμενικής συνάρτησης και αλλού.
- Υπάρχει τέλος ένα σύνολο αντικειμενικών συναρτήσεων, η βελτιστοποίηση των οποίων επιδιώκεται από την επίλυση του προβλήματος μαθηματικού προγραμματισμού. Ανάλογα με το εάν υπάρχουν μία ή περισσότερες αντικειμενικές συναρτήσεις περιγραφής του σκοπού βελτιστοποίησης, προκύπτουν οι περιπτώσεις του μονοκριτηριακού ή πολυκριτηριακού μαθηματικού προγραμματισμού.

Σημειώνεται τέλος ότι η έννοια του προγραμματισμού είναι συνυφασμένη με την έννοια της επιλογής. Αυτό σημαίνει ότι πρώτον, θα πρέπει να υπάρχουν τουλάχιστον δύο δραστηριότητες οι οποίες να λειτουργούν σε συνθήκες ανταγωνιστικής αλληλεξάρτησης και δεύτερον το σύνολο των περιορισμών δε θα πρέπει να περιέχει ασυμβίβαστους περιορισμούς. Με βάση την ανάλυση που περιγράφηκε, το γενικό πρόβλημα μαθηματικού προγραμματισμού διατυπώνεται ως εξής:

$$\text{Opt } \{z(x_1, \dots, x_n) : (x_1, \dots, x_n) \in X\}$$

όπου $X = \{(x_1, \dots, x_n) : g_i(x_1, \dots, x_n) \geq = b_i \text{ με } i=1, \dots, m, x_j \geq 0, j=1, \dots, n\}$

ή ως $\text{Opt } \{z(x_1, \dots, x_n)\}$

με περιορισμούς : $g_i(x_1, \dots, x_n) \geq = b_i, \text{ με } i=1, \dots, m, x_1, \dots, x_n \geq 0$

Το σύνολο X των διανυσμάτων x_1, \dots, x_n που ικανοποιούν τους περιορισμούς:

$$g_i(x) \geq = b_i, \quad i=1, \dots, m \text{ και } x_j \geq 0, \quad \forall i \in j,$$

ονομάζεται σύνολο των εφικτών λύσεων. Ανάλογα με το εάν $x=0$ ή $x \neq 0$ προκύπτουν οι περιπτώσεις του αδύνατου προγράμματος ή εφικτού αντίστοιχα. Ανάλογα με τη μαθηματική μορφή της αντικειμενικής συνάρτησης και των περιοριστικών συναρτήσεων, ο μαθηματικός προγραμματισμός ταξινομείται στις εξής βασικές κατηγορίες:

Γραμμικός Προγραμματισμός (ΓΠ), όπου η αντικειμενική συνάρτηση και οι περιοριστικές συναρτήσεις είναι γραμμικές συναρτήσεις (πρέπει όλες οι συναρτήσεις να είναι γραμμικές).

Μη Γραμμικός προγραμματισμός, όπου από τις περιοριστικές συναρτήσεις και την αντικειμενική συνάρτηση τουλάχιστον μία είναι μη γραμμική (η γραμμικότητα καταστρέφεται εάν έστω και μία συνάρτηση είναι μη γραμμική).

Αμιγής ακέραιος προγραμματισμός, όπου όλες οι μεταβλητές παίρνουν μόνον ακέραιες τιμές. Προφανώς ο αμιγής ακέραιος προγραμματισμός μπορεί να είναι γραμμικός ή μη γραμμικός.

Μικτός ακέραιος προγραμματισμός, όπου κάποιες μεταβλητές παίρνουν ακέραιες τιμές ενώ οι υπόλοιπες είναι συνεχείς. Όπως και προηγουμένως διακρίνονται περιπτώσεις μικτού ακέραιου γραμμικού προγραμματισμού και μικτού ακέραιου μη γραμμικού προγραμματισμού.

Για την επίλυση των προβλημάτων μαθηματικού προγραμματισμού όπως είναι το πρόβλημα του μικτού ακέραιου γραμμικού προγραμματισμού (MILP) των Liberopoulos et al.^a (2009), έχουν αναπτυχθεί αποδοτικοί αλγόριθμοι επίλυσης που μπορούν να αντεπεξέλθουν στις απαιτήσεις των πολύπλοκων μοντέλων μεγάλου πλήθους μεταβλητών. Η συμβολή του MILP στη βελτίωση της παραγωγικότητας έγκειται στο ότι πριν από την ανάπτυξη του κάθε σχεδόν βελτίωση της παραγωγής ήταν αποτέλεσμα τεχνολογικής προόδου ενώ με τις τεχνικές MILP επιτυγχάνεται αύξηση της παραγωγικότητας με τα υπάρχοντα μέσα. Τα τελευταία πενήντα χρόνια αναπτύχθηκε ένας μεγάλος αριθμός αλγορίθμων οι οποίοι εφαρμόζονται μέσω των H/Y για τη βελτιστοποίηση προβλημάτων εντυπωσιακού μεγέθους. Επομένως, η παράλληλη ανάπτυξη των H/Y και των αλγορίθμων επίλυσης δε μπορεί να χαρακτηριστεί ως τυχαίο γεγονός.

3.5 Μορφοποίηση προβλημάτων MILP

Σύμφωνα με την παραπάνω ανάλυση η γενική μορφή ενός προβλήματος MILP μπορεί να διατυπωθεί σε μαθηματική μορφή ως εξής:

$$\text{Opt } \{z(x_1, \dots, x_n) = \sum_{j=1}^n c_j \cdot x_j\}$$

με τους περιορισμούς:

$$\sum_{j=1}^n a_{ij} \cdot x_j \geq = \leq b_i, \quad i=1, \dots, m, \quad x_j \geq 0 \quad \forall j \in J$$

$$x_m \in Z^+, \quad m \in J$$

Σύμφωνα με τη δομή του γενικού αυτού μοντέλου MILP μορφοποιείται μεγάλο πλήθος τυπικών πρακτικών προβλημάτων, διαδικασία η οποία απαιτεί τόσο επιστημονικές γνώσεις όσο και εμπειρία. Το τελευταίο συμβαίνει διότι αν και έχουν αναπτυχθεί άκρως αποτελεσματικοί αλγόριθμοι για την επίλυση προβλημάτων MILP, ωστόσο δεν υπάρχει συγκροτημένη θεωρία που να βοηθά στη μορφοποίησή τους παρά μόνο προσεγγίσεις. Οι πιο συχνά χρησιμοποιούμενες προσεγγίσεις κατά τη μορφοποίηση είναι η «άμεση προσέγγιση» και η προσέγγιση μέσω των «εισροών-εκροών».

3.5.1 Μορφοποίηση με τη μέθοδο της άμεσης προσέγγισης

Στη μέθοδο αυτή διαμορφώνεται στο **πρώτο βήμα** το σύνολο των μεταβλητών του προβλήματος. Το σύνολο αυτό θα πρέπει να είναι πλήρες με την έννοια ότι οι τιμές των μεταβλητών μίας βέλτιστης λύσης θα δίνουν απάντηση στο ζητούμενο του προβλήματος έτσι ώστε να μπορούν να μεταφραστούν σε μία άριστη πολιτική που θα είναι άμεσα εφαρμόσιμη. Για τη μορφοποίηση ενός προβλήματος γίνονται οι εξής υποθέσεις: η υπόθεση της αναλογικότητας των πρώτων υλών ως προς τα προϊόντα, η υπόθεση της προσθετικότητας των πρώτων υλών ως προς το σύνολο των παραγόμενων προϊόντων (οι δύο πρώτες υποθέσεις μπορούν να περιγραφούν και ως υπόθεση της γραμμικότητας) και η υπόθεση της συνέχειας (για συνεχείς μεταβλητές) ή της ακεραιότητας (για ακέραιες μεταβλητές) στο πεδίο ορισμού των τιμών των μεταβλητών.

Στο **δεύτερο βήμα** κατασκευάζεται το σύνολο των περιορισμών του προβλήματος. Μεταξύ των βασικότερων (ή πιο συχνά χρησιμοποιούμενων) περιορισμών είναι και οι εξής:

Περιορισμοί μη αρνητικότητας: Σε προβλήματα τα οποία περιγράφονται από φυσικά μεγέθη, οι μεταβλητές πρέπει να λαμβάνουν θετικές τιμές (οι αρνητικές τιμές δεν έχουν φυσική υπόσταση).

Περιορισμοί παραγωγικής δυναμικότητας: Οι περιορισμοί αυτοί προέρχονται από τα ανώτατα όρια των πόρων που διατίθενται για την εκτέλεση των παραγωγικών διαδικασιών και είναι αδύνατο να ξεπεραστούν.

Περιορισμοί υλικών και πρώτων υλών παραγωγής: Οι περιορισμοί αυτοί προέρχονται από τα ανώτατα όρια στις διαθέσιμες ποσότητες πρώτων και βοηθητικών υλών παραγωγής.

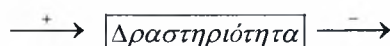
Περιορισμοί ακεραιότητας: Οι περιορισμοί αυτοί ορίζουν την ακεραιότητα των μεταβλητών που μπορούν να λάβουν στο πεδίο ορισμού τους μόνο ακέραιες τιμές.

Περιορισμοί πολιτικής: Οι περιορισμοί αυτοί εκφράζουν διάφορους επιθυμητούς στόχους οι οποίοι αναφέρονται συνήθως στις λειτουργίες της παραγωγής ή της αγοράς.

Στο **τρίτο βήμα** κατασκευάζεται η αντικειμενική συνάρτηση. Η εκτέλεση κάθε δραστηριότητας $j \in J$ σε επίπεδο μονάδας επιφέρει ένα αποτέλεσμα το οποίο συμβολίζουμε ως c_j . Το τι παριστάνει το c_j καθώς και η μονάδα μέτρησής του εξαρτώνται από το ίδιο το πρόβλημα (€, τμχ, hrs). Οι πιο συχνά χρησιμοποιούμενες αντικειμενικές συναρτήσεις σε προβλήματα βελτιστοποίησης είναι η μεγιστοποίηση του κέρδους, η ελαχιστοποίηση του κόστους, η μεγιστοποίηση του βαθμού χρησιμοποίησης και η μεγιστοποίηση της καθαρής παρούσας αξίας.

3.5.2 Μορφοποίηση με την προσέγγιση των εισροών-εκροών

Με την προσέγγιση αυτή κατασκευάζονται αρχικά το σύνολο των δραστηριοτήτων και των μέσων ή συντελεστών παραγωγής που δημιουργούν τους περιορισμούς. Μαζί με το σύνολο των δραστηριοτήτων προσδιορίζονται και οι μονάδες μέτρησης των επιπέδων στα οποία εκτελούνται και τα οποία αποτελούν τα πεδία ορισμού των μεταβλητών x_j του προβλήματος. Σε κάθε δραστηριότητα j αντιστοιχεί ένα διάνυσμα $a_j = \{a_{ij}\}$ τα στοιχεία του οποίου παριστάνουν τις εισροές ή τις εκροές του πόρου i από την εκτέλεση της δραστηριότητας j σε επίπεδο μονάδας. Οι τεχνολογικοί συντελεστές a_{ij} έχουν θετικό πρόσημο αν είναι εισροές και αρνητικό αν είναι εκροές, όπως παρουσιάζεται στο Σχήμα 2.



Σχήμα 2. Ποιοτική απεικόνιση της εισροής και της εκροής μίας δραστηριότητας

Τέλος, εισάγεται το διάνυσμα $b = \{b_i\}$ τα στοιχεία του οποίου παριστάνουν τα ανώτατα όρια των διατιθέμενων ποσοτήτων των πόρων i ή τα κατώτατα όρια των απαιτούμενων ποσοτήτων των συντελεστών i . Με την εκτέλεση κάθε δραστηριότητας j σε επίπεδο μονάδας παράγεται κάποιο αποτέλεσμα το οποίο μετράται με το συντελεστή c_j , οπότε κατασκευάζεται συνολικά ο Πίνακας 6.

Πίνακας 6. Παράγοντες που επιδρούν κατά την εκτέλεση μίας δραστηριότητας

Δραστηριότητες		
Μέσα ή πόροι	Τεχνολογικοί συντελεστές a_{ij}	Ανώτατα ή κατώτατα όρια διαθέσιμων ή απαιτούμενων πόρων b_i
Μέγιστο ή ελάχιστο	Συντελεστές αντικειμενικής συνάρτησης c_j	

4 ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΜΟΝΤΕΛΟΥ MILP

4.1 Μαθηματική μορφοποίηση

4.1.1 Ορισμός παραμέτρων και μεταβλητών

Για τις ανάγκες του μοντέλου MILP των Liberopoulos et al.^a (2009) διακριτοποιείται ο ορίζοντας προγραμματισμού της μίας εβδομάδας σε πεπερασμένο αριθμό χρονικών περιόδων ίσης διάρκειας και το μήκος κάθε περιόδου τίθεται ίσο με το μικρότερο χρόνο μετάβασης μιας διεργασίας από μια κατάσταση σε κάποια άλλη. Στην περίπτωση της υπό εξέταση χημικής βιομηχανίας η διάρκεια αυτή είναι ίση με ένα τετράωρο (4hr) το οποίο απαιτείται για την αλλαγή του χρώματος στο στάδιο POLY. Δεδομένου ότι η παραγωγή είναι 24ωρη, σε κάθε περίοδο τεσσάρων ωρών το στάδιο POLY παράγει σταθερή ποσότητα ημιέτοιμων προϊόντων ίση με 33,33 τόνους/περίοδο (παρτίδα) ενώ θεωρείται λόγω της φύσης του προβλήματος ότι στο στάδιο POLY δεν παρουσιάζεται ποτέ έλλειψη πρώτων υλών.

Στο επόμενο βήμα διακριτοποιείται η χωρητικότητα των σιλό TSS και της μηχανής πολυμερισμού του σταδίου SSP χωρίζοντάς τες σε ακέραιο αριθμό διακριτών θέσεων καθεμία εκ των οποίων έχει χωρητικότητα μίας παρτίδας. Στην αρχή του χρονικού ορίζοντα, το ενεργό σιλό (υπενθυμίζεται ότι χρησιμοποιείται ένα μόνο από τα τρία) του σταδίου TSS έχει κάποιο αρχικό απόθεμα πρώτης ύλης το οποίο καταλαμβάνει κάποιες διακριτές θέσεις (έστω N θέσεις) ενώ στο στάδιο SSP το αρχικό διερχόμενο προϊόν καταλαμβάνει όλη τη χωρητικότητα (έστω M θέσεων) της μηχανής. Ο ρυθμός επεξεργασίας του σταδίου SSP ανά 4hr είναι ίδιος με εκείνον του σταδίου POLY και ίσος με την ποσότητα προϊόντος μίας παρτίδας. Επομένως, αφού η υπό εξέταση χημική βιομηχανία λειτουργεί συνεχώς, ο αριθμός των παρτίδων που εμπεριέχουν τα στάδια TSS και SSP σε κάθε περίοδο του ορίζοντα προγραμματισμού είναι σταθερός και ίσος με N και M αντίστοιχα.

Το χρώμα (L , G ή D) της παρτίδας που βρίσκεται στη n -οστή θέση του συνολικού αριθμού των $N+M$ διακριτών θέσεων των σταδίων TSS και SSP στην αρχή του ορίζοντα προγραμματισμού, εξαρτάται από τις ρυθμίσεις του σταδίου POLY n περιόδους πριν από την έναρξη του χρονικού ορίζοντα προγραμματισμού. Επομένως, για να είναι γνωστό το χρώμα των $N+M$ θέσεων είναι απαραίτητο να είναι γνωστές οι ρυθμίσεις του σταδίου POLY κατά τη διάρκεια των προηγούμενων $N+M$ περιόδων. Κατά την έναρξη λοιπόν του χρονικού ορίζοντα προγραμματισμού μεταφέρεται η πρώτη περίοδος που μπορεί να προγραμματιστεί και αφορά στο παρόν από το 1 στο $N+M+1$ καθώς οι περίοδοι 1 έως $N+M$ αναφέρονται στο παρελθόν.

Στη συνέχεια παρουσιάζεται η μαθηματική διατύπωση των παραμέτρων και των μεταβλητών του μοντέλου βελτιστοποίησης καθώς και τα σετ που δείχνουν τις διαστάσεις τους.

Σετ:

- I : σετ χρωμάτων που παράγονται στο στάδιο POLY, $i \in I = \{1, 2, 3\} = \{L, G, D\}$.
 J : σετ τελικών προϊόντων, $j \in J = \{1, 2, 3, 4\} \equiv \{WG, SD, G, FH\}$.
 Q : σετ σιλό στο στάδιο (LFSS), $q \in Q = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

Παράμετροι:

- T : δείκτης της τελευταίας περιόδου του χρονικού ορίζοντα.
 P : ποσότητα που παράγεται κατά τη διάρκεια μίας περιόδου.
 N : αριθμός των διακριτών θέσεων αποθήκευσης του ενεργού σιλό στο στάδιο TSS.
 M : αριθμός των διακριτών θέσεων διερχόμενης ποσότητας του σταδίου SSP.
 d : κόστος αλλαγής χρώματος στο στάδιο POLY.
 c : κόστος αλλαγής ιξώδους στο στάδιο SSP.
 B : χρονική διάρκεια αλλαγής χρώματος (σε αριθμό περιόδων) στο στάδιο POLY .
 F : χρονική διάρκεια αλλαγής ιξώδους (σε αριθμό περιόδων) στο στάδιο SSP.
 XO_{1t} : δυαδική παράμετρος η οποία παίρνει την τιμή 1 αν η παρτίδα που παράγεται στο στάδιο POLY στο χρόνο t έχει χρώμα «L», και 0 αλλιώς, $t = 1, \dots, N + M$.
 XO_{2t} : δυαδική παράμετρος η οποία παίρνει την τιμή 1 αν η παρτίδα που παράγεται στο στάδιο POLY στο χρόνο t έχει χρώμα «G», και 0 αλλιώς, $t = 1, \dots, N + M$.
 XO_{3t} : δυαδική παράμετρος η οποία παίρνει την τιμή 1 αν η παρτίδα που παράγεται στο στάδιο POLY στο χρόνο t έχει χρώμα «D», και 0 αλλιώς, $t = 1, \dots, N + M$.
 AO_t : δυαδική παράμετρος που παίρνει την τιμή 1 αν μία αλλαγή ιξώδους έχει ενεργοποιηθεί στην αρχή της περιόδου t , και 0 αλλιώς, $t = N + M - F + 2, \dots, N + M$.
 ZO : δυαδική παράμετρος που παίρνει την τιμή 1 αν το ιξώδες της παρτίδας στην τελευταία διακριτή θέση του σταδίου SSP στην αρχή του χρονικού ορίζοντα είναι υψηλό, και 0 αλλιώς.
 WO_{qj} : δυαδική παράμετρος που παίρνει την τιμή 1 αν το προϊόν j αποθηκεύεται στο σιλό q του σταδίου LFSS στην αρχή του χρονικού ορίζοντα, και 0 αλλιώς.
 SO_{qj} : ποσότητα προϊόντος j που περιέχεται στο σιλό q του σταδίου LFSS στην αρχή του χρονικού ορίζοντα.
 S_{max} : μέγιστη χωρητικότητα του σιλό του σταδίου LFSS.
 S_{min} : ελάχιστη ποσότητα προϊόντος σε ένα σιλό του σταδίου LFSS ώστε να μη θεωρείται άδειο.
 SS_{minj} : επιθυμητό απόθεμα ασφαλείας του προϊόντος j στο στάδιο LFSS στο τέλος του ορίζοντα προγραμματισμού.
 u_{ST} : μέγιστη ποσότητα προϊόντος που μπορεί να φορτωθεί από ένα σιλό του σταδίου LFSS σε ένα σιλοφόρο φορτηγό κατά τη διάρκεια μιας περιόδου.
 u_{BC} : μέγιστη ποσότητα προϊόντος που μπορεί να φορτωθεί από ένα σιλό του σταδίου LFSS σε ένα κοντέινερ κατά τη διάρκεια μιας περιόδου.

u_{BB} : μέγιστη ποσότητα προϊόντος που μπορεί να σακιαστεί κατά τη διάρκεια μιας περιόδου από το στάδιο LFSS.

$R0_j$: απόθεμα προϊόντος j στην αποθήκη στην αρχή του χρονικού ορίζοντα προγραμματισμού.

R_{max} : μέγιστη χωρητικότητα αποθήκης.

R_{min_j} : επιθυμητό απόθεμα ασφαλείας του προϊόντος j στην αποθήκη στο τέλος του χρονικού ορίζοντα προγραμματισμού.

dST_{jt} : ζήτηση του προϊόντος j σε σιλοφόρα φορτηγά την περίοδο $t, t = N + M + 1, \dots, T$.

dBC_{jt} : ζήτηση του προϊόντος j σε κοντέινερ την περίοδο $t, t = N + M + 1, \dots, T$.

dBB_{jt} : ζήτηση του προϊόντος j σε σακιά την περίοδο $t, t = N + M + 1, \dots, T$.

Μεταβλητές:

x_{it} : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν η παρτίδα που παράγεται στο στάδιο POLY την περίοδο t έχει χρώμα i , και 0 αλλιώς, $t = 1, \dots, T$.

y_{jt} : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν το τελικό προϊόν που παράγεται στο στάδιο SSP την χρονική περίοδο t είναι το j , και 0 αλλιώς, $t = N + M + 1, \dots, T + (F/2)$.

a_t : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν ενεργοποιείται αλλαγή στο ιξώδες στην αρχή της περιόδου t , και 0 αλλιώς, $t = N + M - F + 2, \dots, T$.

z_t : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν το ιξώδες του προϊόντος που παράγεται την χρονική περίοδο t είναι υψηλό, και 0 αλλιώς, $t = N + M, \dots, T + (F/2)$.

S_{qjt} : ποσότητα του προϊόντος j στο σιλό q του σταδίου LFSS στο τέλος της περιόδου $t, t = N + M, \dots, T$.

W_{qjt} : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν στο σιλό q του σταδίου LFSS την περίοδο t περιέχεται προϊόν j , και 0 αλλιώς, $t = N + M, \dots, T$.

g_{qjt} : δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν το προϊόν j που εξέρχεται από το στάδιο SSP διοχετεύεται μέσα στο σιλό q του σταδίου LFSS στην περίοδο t , και 0 αλλιώς, $t = N + M + 1, \dots, T$.

G_{qjt} : ποσότητα προϊόντος j η οποία εξέρχεται από το σιλό q του σταδίου LFSS στην περίοδο $t, t = N + M + 1, \dots, T$.

b_{qjt} : ποσότητα του προϊόντος j η οποία φορτώνεται από το σιλό q του σταδίου LFSS σε σακιά την περίοδο $t, t = N + M + 1, \dots, T$.

f_{qjt} : ποσότητα του προϊόντος j η οποία φορτώνεται από το σιλό q του σταδίου LFSS σε σιλοφόρα φορτηγά την περίοδο $t, t = N + M + 1, \dots, T$.

h_{qjt} : ποσότητα του προϊόντος j η οποία φορτώνεται από το σιλό q του σταδίου LFSS σε κοντέινερ την περίοδο $t, t = N + M + 1, \dots, T$.

R_{jt} : ποσότητα του προϊόντος j στην αποθήκη στο τέλος της περιόδου $t, t = N + M, \dots, T$.

4.1.2 Μοντέλο MILP

Στην παράγραφο αυτή παρουσιάζεται συνοπτικά τα βασικά σημεία του μοντέλου MILP που ανέπτυξαν οι Liberopoulos et al.^a (2009) για την επίλυση του προβλήματος που περιγράφηκε. Σημειώνεται ότι η μορφοποίηση που αναπτύχθηκε περιλαμβάνει και τις εξής ανισότητες προσδιορισμού των χρονικών περιόδων:

$$F < N + M, \quad B < N + M, \quad N + M < T - (N + M).$$

Η αντικειμενική συνάρτηση του μοντέλου των Liberopoulos et al.^a (2009) είναι η εξής:

$$\text{Min } c \left(\sum_{t=N+M+1}^T a_t \right) + d \left(\frac{1}{B} \sum_{t=N+M+1}^T x_{2t} \right) \quad (0.0)$$

Η παραπάνω έκφραση της αντικειμενικής συνάρτησης (0.0) ελαχιστοποιεί το συνολικό άθροισμα του αριθμού των αλλαγών ιξώδους και χρώματος οι οποίες πραγματοποιούνται στον χρονικό ορίζοντα προγραμματισμού. Το πρώτο άθροισμα της αντικειμενικής συνάρτησης αναπαριστά το συνολικό αριθμό των αλλαγών ιξώδους ενώ το δεύτερο αναπαριστά το συνολικό αριθμό των αλλαγών χρώματος κατά τη διάρκεια του ορίζοντα προγραμματισμού. Ο δεύτερος όρος διαιρείται με την παράμετρο B η οποία αντιστοιχεί στη χρονική διάρκεια μιας αλλαγής χρώματος. Οι περιορισμοί του μοντέλου MILP παρουσιάζονται και περιγράφονται αναλυτικά στην εργασία των Liberopoulos et al. (2009).

Στη μορφοποίηση των Liberopoulos et al.^a (2009) γίνεται η υπόθεση ότι η παραγωγική διαδικασία βρίσκεται κατά την έναρξη του ορίζοντα προγραμματισμού στην κατάσταση στην οποία βρέθηκε κατά το τέλος του προηγούμενου ορίζοντα παραγωγής. Αντιθέτως, σε περίπτωση όπου η βιομηχανία ξεκινά να παράγει έπειτα από μία μακροχρόνια διακοπή της παραγωγής της, όλες οι θέσεις των σταδίων TSS και SSP είναι άδειες και είναι πιθανό τα σιλό και η αποθήκη να έχουν πολύ χαμηλά ή ακόμη και μηδενικά αποθέματα. Η κατάσταση αυτή παρατηρείται συνήθως έπειτα από μία σημαντική βλάβη ή έπειτα από την ετήσια συντήρηση του εργοστασίου. Στις περιπτώσεις αυτές όταν το εργοστάσιο ξεκινήσει να παράγει ο κύριος στόχος δεν είναι η ικανοποίηση της ζήτησης αλλά η σταθεροποίηση της παραγωγής και η πρόοδος της, προκειμένου το παραγωγικό σύστημα να ισορροπήσει όσο το δυνατόν πιο γρήγορα σε συνθήκες ομαλής λειτουργίας κατά τις οποίες μπορεί να επιλυθεί η μακροχρόνια σχεδίαση του προγραμματισμού παραγωγής. Στο μοντέλο των Liberopoulos et al.^a (2009) δε λαμβάνεται υπόψη μία τέτοια περίπτωση καθώς οι συγγραφείς ανέπτυξαν το μοντέλο για την περίπτωση ομαλής λειτουργίας όπου η παραγωγική διαδικασία θεωρείται συνεχόμενη και αδιάκοπη.

Επίσης σημειώνεται ότι απαιτείται η ζήτηση τελικών προϊόντων σε σιλοφόρα φορτηγά, σε κοντέινερ και σε σακιά να ικανοποιείται σε κάθε περίοδο του χρονικού ορίζοντα προγραμματισμού. Η αυστηρή απαίτηση της έγκαιρης ικανοποίησης της ζήτησης προέρχεται από

τον τρόπο λειτουργίας της υπό εξέταση βιομηχανίας καθώς οι πελάτες της προγραμματίζουν την αποστολή δικών τους φορτηγών ή κοντέινερ για να παραλάβουν τις παραγγελίες τους τις συγκεκριμένες περιόδους που είναι γνωστές στην αρχή του χρονικού ορίζοντα προγραμματισμού και δεν υπάρχει η δυνατότητα να περιμένουν. Αυτή η απαίτηση είναι δυνατό να οδηγήσει σε μη εφικτή λύση του προγράμματος βελτιστοποίησης, αν τα αποθέματα του σταδίου LFSS και της αποθήκης δεν είναι αρκετά μεγάλα ώστε να απορροφήσουν τις μεταβολές της ζήτησης. Η κατάλληλη επιλογή των επιθυμητών αποθεμάτων ασφαλείας για κάθε προϊόν μπορεί να βοηθήσει στη μείωση του αριθμού των μεταβάσεων κατά την διάρκεια της παραγωγικής διαδικασίας και να εξαλείψει τις ελλείψεις σε απόθεμα, όμως ταυτόχρονα θα πρέπει να μην επιβαρύνεται η παραγωγική διαδικασία λόγω μείωσης ή και πιθανής έλλειψης του διαθέσιμου χώρου αποθήκευσης.

Οι Liberopoulos et al.^a (2009) εφαρμόζουν το μοντέλο που περιγράφηκε θέτοντας τον χρονικό ορίζοντα προγραμματισμού ίσο με μία εβδομάδα ο οποίος και διακριτοποιείται σε 42 περιόδους των 4 ωρών. Στη συνέχεια επιλύουν το μοντέλο τους για ένα κυλιόμενο συνολικό χρονικό ορίζοντα 24 εβδομάδων ο οποίος αντιστοιχεί σε ημερολογιακή διάρκεια περίπου 6 μηνών, θεωρώντας ότι η παραγωγική διαδικασία της υπό εξέταση χημικής βιομηχανίας παραμένει αδιάκοπη σε ολόκληρο το χρονικό ορίζοντα των 6 μηνών. Επομένως, η κατάσταση του συστήματος στην έναρξη κάθε εβδομάδας τίθεται ίση με την κατάσταση στην οποία βρέθηκε το σύστημα κατά το τέλος της προηγούμενης εβδομάδας. Για κάθε εβδομάδα χρησιμοποιήθηκαν από τους συγγραφείς πραγματικά δεδομένα της ζήτησης και σημειώνουν ότι δεν παρατηρήθηκαν περιπτώσεις κατά τις οποίες η επίλυση του μοντέλου να είναι μη εφικτή λόγω αδυναμίας έγκαιρης ικανοποίησης της ζήτησης. Τα επίπεδα των επιθυμητών αποθεμάτων ασφαλείας που χρησιμοποίησαν οι συγγραφείς στη μοντελοποίηση που προτείνουν προέκυψαν από τη μοντελοποίηση και επίλυση του προβλήματος προγραμματισμού της συνεχόμενης παραγωγής ως ένα *στοχαστικό πρόβλημα βέλτιστου χρονικού προγραμματισμού παρτίδων παραγωγής (Stochastic Economic Lot Scheduling Problem ή SELSP)* όπως περιγράφεται από τους Liberopoulos et al.^b (2009).

Στην επόμενη παράγραφο αναλύονται οι υποθέσεις του γραμμικού μοντέλου μεικτού ακέрайου προγραμματισμού των Liberopoulos et al.^a (2009) οι οποίες αποτέλεσαν και το ερέθισμα για την ανάπτυξη του προσομοιωτικού μοντέλου που περιγράφεται στα επόμενα κεφάλαια. Με τη χρήση προσομοιωτικών μεθόδων επίλυσης γίνεται μία προσπάθεια αναίρεσης κάποιων από τις υποθέσεις του μοντέλου MILP με ταυτόχρονη όμως μείωση της ποιότητας της αποδιδόμενης λύσης.

4.2 Κυριότερες υποθέσεις του γραμμικού μοντέλου

Κατά τη μορφοποίηση που προτείνουν οι Liberopoulos et al.^a (2009) για την περιγραφή της λειτουργίας της υπό εξέταση χημικής βιομηχανίας γίνονται συγκεκριμένες παραδοχές και απλοποιήσεις προκειμένου να μειωθεί τόσο το πλήθος των μεταβλητών όσο και η πολυπλοκότητα του προβλήματος. Οι κυριότερες απλοποιήσεις που επιφέρουν και τις σημαντικότερες παραδοχές του μοντέλου MILP είναι οι εξής:

- Θεωρείται ότι στην αρχή κάθε εβδομάδας η ζήτηση ανά τύπο και είδος προϊόντος είναι γνωστή και δεν προκύπτουν έκτακτες παραγγελίες κατά τη διάρκειά της. Ωστόσο τα πραγματικά συστήματα αντιμετωπίζουν πολύ συχνά τόσο μεταβολές στην ήδη εκδηλωμένη ζήτηση όσο και έκτακτες παραγγελίες πελατών. Όταν οι πελάτες αυτοί δεν αποτελούν σημαντική πηγή εσόδων για μία βιομηχανία τότε αυτή μπορεί να αρνηθεί να διαταράξει το τρέχων πρόγραμμα παραγωγής και αν συμφωνήσει κι ο πελάτης να προγραμματίσει την ικανοποίηση της έκτακτης παραγγελίας ως κανονική την επόμενη εβδομάδα. Αν όμως ο πελάτης αποτελεί σημαντική πηγή εσόδων για μία βιομηχανία τότε αυτή συνήθως προχωρά στην ικανοποίηση της έκτακτης παραγγελίας ακόμη και σε βάρος πελατών που έχουν θέσει έγκαιρα τις παραγγελίες τους προκειμένου να αποφύγει ενδεχόμενες αρνητικές επιπτώσεις στις μελλοντικές παραγγελίες του πελάτη.
- Θεωρείται ότι πρέπει πρώτον, να ικανοποιηθεί έγκαιρα το σύνολο της ζήτησης και δεύτερον, να βρίσκεται η στάθμη των αποθεμάτων στο τέλος του ορίζοντα προγραμματισμού πάνω από την επιθυμητή της τιμή. Η παραδοχή αυτή μπορεί να επιφέρει σοβαρά προβλήματα κατά την επίλυση του μοντέλου γιατί εάν σε κάποια εβδομάδα η δυναμικότητα παραγωγής μαζί με τα αποθέματα δεν επαρκούν για να ικανοποιήσουν τη ζήτηση των προϊόντων τότε η επίλυση του προβλήματος θα είναι αδύνατη (infeasible problem). Το ίδιο πρόβλημα θα προκύψει ακόμη και στην περίπτωση όπου η ζήτηση μπορεί να ικανοποιηθεί αλλά δε μπορεί να αναπληρωθεί τμήμα των αποθεμάτων ασφαλείας (incomplete replenishment). Αυτές οι παραδοχές του MILP έχουν ως αποτέλεσμα την αδυναμία απόδοσης ενός προγράμματος παράγωγης σε μία περίοδο υψηλής ζήτησης.
- Θεωρείται ότι η χημική βιομηχανία λειτουργεί συνεχώς και χωρίς διακοπή σε ολόκληρη τη διάρκεια του χρονικού προγραμματισμού. Η παραδοχή αυτή θεωρείται ρεαλιστική για ένα σύντομο χρονικό ορίζοντα προγραμματισμού διότι ότι η πιθανότητα διακοπής της παραγωγικής διαδικασίας είναι σχετικά μικρή. Ωστόσο στη διάρκεια ενός μεγάλου χρονικού ορίζοντα υπάρχει πληθώρα τόσο ενδογενών όσο και εξωγενών παραγόντων των οποίων η επίδραση μπορεί να ακυρώσει αυτήν την παραδοχή: μία διακοπή στην παροχή ενέργειας, μία απεργία στους μεταφορικούς κόμβους παροχής πρώτων υλών, κάποια φυσική καταστροφή, ένα εργατικό ατύχημα, μία βλάβη στα μηχανήματα (ο παράγοντας αυτός λόγω της μεγάλης σημασίας του θα αναλυθεί και ξεχωριστά), το κλείσιμο των οδικών οδών από διαμαρτυρομένους είναι μερικά μόνο παραδείγματα που μπορούν

να επιφέρουν τέτοιου είδους επιδράσεις. Επειδή ο ΓΠ μπορεί να εφαρμοστεί ως μεθοδολογία επίλυσης μόνο σε καθοριστικό περιβάλλον δεν είναι δυνατό να συμπεριλάβει τη στοχαστικότητα που υπάρχει λόγω τέτοιων ζητημάτων. Επομένως είναι δόκιμο κατά την ολοκληρωμένη προσέγγιση πραγματικών προβλημάτων να γίνεται ταυτόχρονη χρήση πολλών μεθοδολογιών ώστε η μία να συμπληρώνει τις αδυναμίες της άλλης.

➤ Θεωρείται ότι κατά τη διάρκεια του χρονικού ορίζοντα προγραμματισμού η συνάρτηση αξιοπιστίας των παραγωγικών μηχανημάτων είναι ίση με τη μονάδα κι επομένως δεν υπάρχουν βλάβες. Παρόλο που είναι σύνηθες να γίνεται αυτή η παραδοχή, η ρεαλιστικότητά της εξαρτάται από πολλούς παράγοντες οι κυριότεροι εκ των οποίων είναι η διάρκεια του χρονικού ορίζοντα προγραμματισμού, η χρονική του απόσταση από την τελευταία συντήρηση του εξοπλισμού, οι δεξιότητες των χειριστών των μηχανημάτων ώστε να αποφεύγονται κακοί χειρισμοί, και τα τυχαία γεγονότα. Στην πράξη, όταν η μέση συνολική ζήτηση είναι πολύ κοντά στο μέσο ρυθμό παραγωγής τότε είναι πολύ πιθανό ακόμη και μία σχετικά μικρή βλάβη του εξοπλισμού να ακυρώσει ολόκληρο το πρόγραμμα παραγωγής. Για το λόγο αυτό προτείνεται η ταυτόχρονη χρήση εναλλακτικών μεθοδολογιών ώστε να μελετηθούν παράλληλα και τέτοιου είδους στοχαστικά φαινόμενα τα οποία επηρεάζουν άμεσα τη λειτουργία της χημικής βιομηχανίας.

➤ Θεωρείται ότι ο ρυθμός παραγωγής της χημικής βιομηχανίας είναι σταθερός και αμετάβλητος. Αυτό σημαίνει ότι ανεξαρτήτως των συνθηκών που αναπτύσσονται κατά τη διάρκεια του ορίζοντα προγραμματισμού, η παραγωγική διαδικασία θα εξακολουθεί να παράγει προϊόντα με τον ίδιο πάντοτε σταθερό ρυθμό. Ωστόσο, είναι γνωστό ότι οι παραγωγικές μονάδες έχουν μία σχετική δυνατότητα επιτάχυνσης ή επιβράδυνσης της παραγωγής τους γύρω από τη συνήθη μέση παραγωγικότητά τους ώστε να μπορούν να αντεπεξέλθουν στις εκάστοτε συνθήκες. Για παράδειγμα, αν η ζήτηση υπερβεί κατά λίγο τον υφιστάμενο ρυθμό παραγωγής τότε συνήθως επιταχύνεται κατά λίγο η παραγωγική διαδικασία προκειμένου να ικανοποιηθούν οι πελάτες. Ακόμη, σε μία ακραία περίπτωση όπου η ζήτηση για ένα χρονικό διάστημα είναι πολύ χαμηλή ή και μηδενική τότε μπορεί η παραγωγική διαδικασία να σταματήσει και να εκτελεστούν εργασίες όπως είναι η συντήρηση προκειμένου μεταγενέστερα να εξοικονομηθεί παραγωγικός χρόνος.

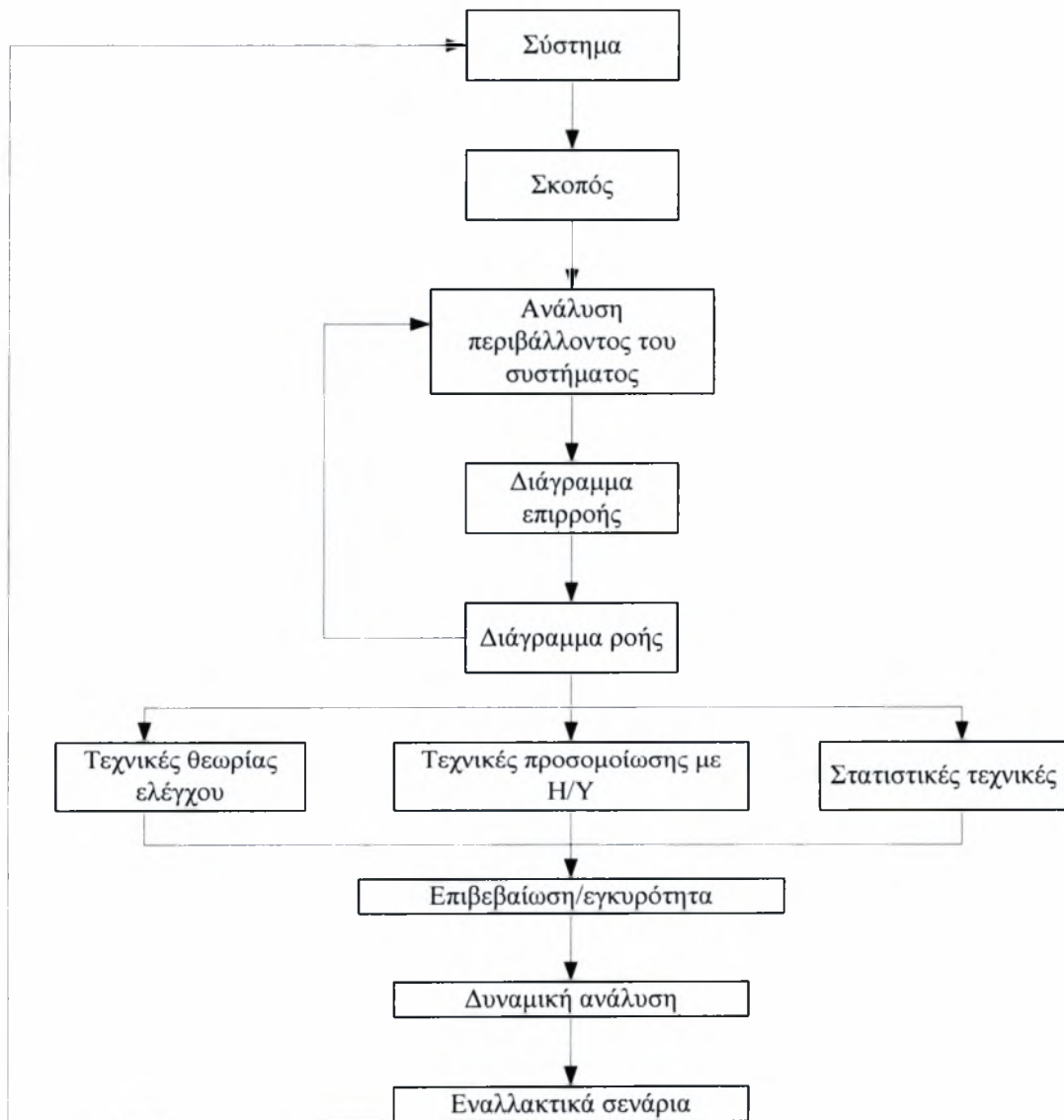
➤ Θεωρείται ότι στο στάδιο TSS χρησιμοποιείται μόνο ένα σιλό. Η απλοποίηση αυτή οφείλεται στην υπόθεση του καθοριστικού περιβάλλοντος στο οποίο δεν υπάρχει ανάγκη διατήρησης αποθεμάτων ημιετοιμών προϊόντων. Ωστόσο, η ευελιξία της βιομηχανίας για παραγωγή έκτακτων παραγγελιών μειώνεται καθώς το προϊόν που εισέρχεται στο στάδιο επεξεργασίας SSP δεν εξαρτάται μόνο από την ύπαρξη αποθεμάτων ημιετοιμών προϊόντων αλλά και από την αλληλουχία παραγωγής τους στο στάδιο POLY. Ακόμη, λόγω της έλλειψης δυνατότητας αποθήκευσης ημιετοιμών προϊόντων στα τρία σιλό του σταδίου TSS δε συμπεριλαμβάνεται η περίπτωση όπου το στάδιο POLY εξακολουθεί να παράγει ενώ το στάδιο SSP είναι σε διαδικασίες συντήρησης.

➤ Θεωρείται ότι το μεγαλύτερο χρονικό βήμα για το οποίο η αναπαράσταση του χρόνου δεν έχει επιπτώσεις στο πρόγραμμα παραγωγής είναι ίσο με τη διάρκεια της συντομότερης διαδικασίας μετάβασης από τις δύο που υπάρχουν, χρώματος και ιξώδους. Ωστόσο μία τέτοια προσέγγιση μειώνει την ακρίβεια και την αποτελεσματικότητα της αποδιδόμενης λύσης διότι αν υπήρχε η δυνατότητα ενεργοποίησης μεταβάσεων και σε ενδιάμεσες χρονικές στιγμές μικρότερες του τετραώρου, όπως άλλωστε συμβαίνει σε όλα τα πραγματικά συστήματα, τότε θα υπήρχε και αυξημένη ευελιξία κατά τη διαδικασία βελτιστοποίησης. Η απόφαση για ένα τόσο μεγάλο χρονικό βήμα μοντελοποίησης πηγάζει από την πρόθεση να μειωθεί ο συνολικός αριθμός των μεταβλητών του μοντέλου, καθώς η διακριτοποίηση του χρόνου σε μικρά τμήματα (πχ λεπτού, δεκάλεπτου, μισάωρου, ώρας) αυξάνει κατά πολύ το πλήθος τους και κατά συνέπεια των απαιτούμενο υπολογιστικό χρόνο επίλυσης.

5 ΔΥΝΑΜΙΚΗ ΣΥΣΤΗΜΑΤΩΝ

5.1 Μεθοδολογία Δυναμικής Συστημάτων (System Dynamics)

Η Θεωρία της Δυναμικής Συστημάτων (ΔΣ) αναπτύχθηκε από τον Forrester το 1961 και έγινε ευρέως γνωστή μέσα από τις διάφορες εφαρμογές της. Στόχο της ΔΣ αποτελεί η μελέτη των δυναμικών αλληλεπιδράσεων μεταξύ των διαφόρων μεταβλητών ενός συστήματος στη διαδικασία λήψης αποφάσεων μακροπρόθεσμου συνήθως ορίζοντα. Κατά τον Sterman (2000), η ΔΣ βασίζεται στις Θεωρίες : α) της Λήψης Αποφάσεων, β) της Ανάλυσης Αποφάσεων και γ) της Προσομοίωσης. Τα συστηματικά βήματα κατά τη διαμόρφωση ενός μοντέλου με τη μέθοδο της ΔΣ αναπαριστώνται γραφικά στο Σχήμα 4.



Σχήμα 4. Συστηματικά βήματα κατά τη διαμόρφωση μοντέλων Δυναμικής Συστημάτων

Η διαδικασία δημιουργίας μοντέλων σύμφωνα με τη Θεωρία της Δυναμικής Συστημάτων έχει ως σκοπό την όσο πιο πιστή (κατά το μέτρο του εφικτού) απεικόνιση του πραγματικού συστήματος. Τα βήματα αυτά μπορούν να συνοψιστούν στις παρακάτω γενικές ομάδες διαδικασιών:

1. Καθορισμός του υπό εξέταση συστήματος, των διαδικασιών που περιγράφουν τη λειτουργία του και των αντικειμενικών του στόχων.
2. Ανάλυση των ορίων επιρροής του συστήματος και προσδιορισμός του εσωτερικού περιβάλλοντος (σύνολο μεταβλητών που μπορούν να επηρεαστούν από τις εσωτερικές αποφάσεις του συστήματος) και του εξωτερικού περιβάλλοντος (σύνολο μεταβλητών που δε μπορούν να επηρεαστούν από τις εσωτερικές αποφάσεις του συστήματος αλλά προκύπτουν εξωγενώς).
3. Γενική περιγραφή του συστήματος μέσω ενός διαγράμματος επιρροής στο οποίο καταγράφονται ποιοτικά οι επιδράσεις μεταξύ των μεταβλητών του συστήματος.
4. Μαθηματική διατύπωση του μοντέλου η οποία περιλαμβάνει την κατάστρωση του διαγράμματος ροής και των μαθηματικών σχέσεων μεταξύ των μεταβλητών και των παραμέτρων που περιγράφουν το πραγματικό σύστημα.
5. Ανάπτυξη του δυναμικού μοντέλου προσομοίωσης με χρήση λογισμικού (όπως DYSMAP[®], DYNAMO[®], Vensim[®], Stella[®], i-think[®], Powersim[®]). Η αρχικοποίηση του μοντέλου προκύπτει με επεξεργασία των καταγεγραμμένων ιστορικών στοιχείων του συστήματος και από πληροφορίες οι οποίες παρέχονται από άτομα που έχουν εμπειρία στο υπό μελέτη πραγματικό σύστημα.
6. Επιβεβαίωση του μοντέλου ως ικανοποιητική απεικόνιση του πραγματικού συστήματος μέσω της εφαρμογής κατάλληλων κριτηρίων ώστε να καθιερωθεί στα αποτελέσματά του «ικανοποιητική εμπιστοσύνη» (model validation). Στο βήμα αυτό είναι πιθανή η επιστροφή και επιδιόρθωση του διαγράμματος επιρροής του μοντέλου με στόχο την ακριβέστερη απεικόνιση του πραγματικού συστήματος.
7. Δυναμική ανάλυση και εξέταση διαφόρων εναλλακτικών κανόνων λειτουργίας του πραγματικού συστήματος προς την κατεύθυνση επίτευξης των στόχων του ώστε να αξιολογηθεί η αποτελεσματικότητα εναλλακτικών κανόνων λειτουργίας και να εντοπιστούν εκείνες οι πολιτικές με τις οποίες επιτυγχάνονται αποδοτικότερα οι επιδιωκόμενοι στόχοι.

5.2 Δομή των διαγραμμάτων επιρροής

Η Δυναμική Συστημάτων επικεντρώνεται στην αναγνώριση της δομής και της συμπεριφοράς των πραγματικών συστημάτων ώστε να προσδιοριστούν οι σχέσεις αλληλεπίδρασης και οι αναδράσεις μεταξύ εκείνων των μεταβλητών που χαρακτηρίζουν τη συμπεριφορά του συστήματος. Οι επιρροές μεταξύ των μεταβλητών του συστήματος συμβολίζονται με τις γραμμές επιρροής (βέλη) οι οποίες είναι είτε συνεχείς είτε διακεκομμένες, ανάλογα με το είδος της επιρροής που αντιπροσωπεύουν. Μία συνεχής γραμμή δηλώνει φυσική ροή ενώ μία διακεκομμένη δηλώνει ροή πληροφορίας. Η φορά των γραμμών επιρροής υποδηλώνει την κατεύθυνση της επιρροής, δηλαδή από τον παράγοντα της επίδρασης (μεταβλητή βάσεως) προς τον αποδέκτη της (μεταβλητή κορυφής) ενώ τα σύμβολα «+» ή «-» πάνω σε κάθε γραμμή επιρροής δηλώνουν το είδος της επίδρασης. Συγκεκριμένα, εάν μια θετική/ αρνητική μεταβολή στη μεταβλητή βάσης παράγει μία θετική/ αρνητική μεταβολή αντίστοιχα στη μεταβλητή κορυφής τότε η επιρροή μεταξύ των δύο μεταβλητών αναφέρεται ως θετική και συμβολίζεται με «+». Αντιθέτως, εάν μια θετική/ αρνητική μεταβολή στη μεταβλητή βάσης παράγει μία αρνητική/ θετική μεταβολή αντίστοιχα στη μεταβλητή κορυφής τότε η επιρροή μεταξύ των δύο μεταβλητών αναφέρεται ως αρνητική και συμβολίζεται με «-». Τέλος η ύπαρξη του συμβόλου «||» στις γραμμές επιρροής δηλώνει ότι η μεταβλητή βάσεως επιδρά στη μεταβλητή κορυφής με κάποια χρονική καθυστέρηση.

Σε ένα διάγραμμα επιρροής προσδιορίζονται επίσης οι κύριοι βρόγχοι ανατροφοδότησης οι οποίοι είναι είτε θετικές αναδράσεις (reinforcing) είτε αρνητικές (balancing). Στις περιπτώσεις των θετικών αναδράσεων, μία αρχική διαταραχή σε κάποια από τις μεταβλητές του βρόγχου οδηγεί μέσω της ανάδρασης σε συνεχόμενη μεταβολή όλων των μεταβλητών του βρόγχου (συμπεριλαμβανομένης και της ίδιας) προς την ίδια κατεύθυνση, οδηγώντας το σύστημα σε ασταθή συμπεριφορά. Αντιθέτως, μία διαταραχή σε κάποια από τις μεταβλητές ενός βρόγχου αρνητικής ανάδρασης οδηγεί το σύστημα μέσω της ανάδρασης σε μία συμπεριφορά επιστροφής στην ίδια κατάσταση ισορροπίας. Η συνύπαρξη των επιδράσεων μίας θετικής και μίας αρνητικής ανάδρασης στην ίδια μεταβλητή έχει ως αποτέλεσμα όταν υπάρχει μία διαταραχή να οδηγείται το σύστημα σε μία συμπεριφορά αναζήτησης μίας νέας κατάστασης ισορροπίας.

Τα διαγράμματα επιρροής εξυπηρετούν δύο σημαντικούς σκοπούς κατά την ανάπτυξη μοντέλων Δυναμικής Συστημάτων: πρώτον, χρησιμεύουν ως προκαταρκτικά σχέδια των υποθέσεων επιρροής μεταξύ των μεταβλητών και των παραμέτρων που συμμετέχουν στην κατάστρωσή του. Δεύτερον, τα διαγράμματα επιρροής επιτρέπουν την οπτικοποίηση αυτών των επιδράσεων για την καλύτερη κατανόηση της λειτουργίας του πραγματικού συστήματος και διευκολύνουν τη δημιουργία του διαγράμματος ροής, δηλαδή του καθεαυτού μαθηματικού μοντέλου.

5.3 Δομή των διαγραμμάτων ροής

Τα διαγράμματα επιρροής αποτελούν ένα μέσο προκαταρκτικής κατανόησης των λειτουργιών του υπό μελέτη συστήματος και διευκολύνουν την επικοινωνία μεταξύ του προγραμματιστή και της Διοίκησης του συστήματος. Ωστόσο η μορφοποίηση ενός πλήρως λειτουργικού μοντέλου απεικόνισης του συστήματος απαιτεί την όσο το δυνατόν πιο πλήρη και λεπτομερή περιγραφή του συστήματος. Η περιγραφή αυτή επιτυγχάνεται μέσω των διαγραμμάτων ροής όπου απεικονίζονται η κατάσταση του συστήματος σε καταστατικές μεταβλητές επιπέδων, οι μεταβολές της κατάστασης του συστήματος σε ρυθμούς μεταβολής των καταστατικών μεταβλητών, ο καθορισμός πολιτικών σε βοηθητικές μεταβλητές αποφάσεων, οι παράμετροι του συστήματος σε σταθερές, η μεταφορά υλών σε ροές υλών, η μετάδοση πληροφοριών σε ροές πληροφοριών και τέλος οι υπαρκτές καθυστερήσεις σε συναρτήσεις καθυστέρησης.

Ειδικότερα, οι καταστατικές μεταβλητές επιπέδων περιγράφουν την κατάσταση του συστήματος συναρτήσει του χρόνου μέσω αθροιστικών διαφορικών εξισώσεων. Οι ρυθμοί μεταβολής περιγράφουν το αποτέλεσμα (αποφάσεις) συγκεκριμένων κανόνων λήψης αποφάσεων και μεταβάλλουν τις καταστατικές μεταβλητές του μοντέλου. Οι κανόνες λήψης αποφάσεων προκύπτουν από πολιτικές και πρωτόκολλα που είτε εφαρμόζονται είτε ελέγχεται αν μπορούν να εφαρμοστούν στο πραγματικό σύστημα, ενώ οι αποφάσεις είναι το αποτέλεσμα αυτών των διαδικασιών. Κατά τη διαμόρφωση των ρυθμών μεταβολής χρησιμοποιούνται διάφορες βοηθητικές μεταβλητές οι τιμές των οποίων προκύπτουν βάσει των μηχανισμών αναδράσεων του μοντέλου. Τόσο οι ρυθμοί μεταβολής όσο και οι βοηθητικές μεταβλητές εξαρτώνται από τις σταθερές του μοντέλου οι οποίες αναπαριστούν τις παραμέτρους του συστήματος που παραμένουν αμετάβλητες κατά τη διάρκεια του χρονικού ορίζοντα προσομοίωσης. Τέλος οι συναρτήσεις καθυστέρησης απεικονίζουν τις καθυστερήσεις που υπάρχουν στις πραγματικές φυσικές ροές (π.χ. χρόνοι μεταφοράς), στη μετάδοση των πληροφοριών (π.χ. χρόνος εφαρμογής αποφάσεων, χρόνος συλλογής και επεξεργασίας δεδομένων) και στη φυσική αδράνεια του συστήματος (π.χ. χρόνος απόκρισης σε ερεθίσματα). Σημειώνεται ότι καθυστερήσεις που είναι μικρότερες του χρονικού βήματος προσομοίωσης δεν είναι δυνατό να ληφθούν υπόψη.

Η δομή του μοντέλου αποτελείται από ένα σύστημα εξισώσεων, ανισώσεων και συναρτήσεων ικανό να αναπαράγει τη συμπεριφορά του συστήματος κατά τη διάρκεια του χρονικού ορίζοντα προσομοίωσης. Οι προκύπτουσες τιμές των επιπέδων συναρτήσει του χρόνου αποδίδουν την αναμενόμενη συμπεριφορά του συστήματος για τους εναλλακτικούς κανόνες λειτουργίας που εξετάζονται μέσω του μοντέλου προσομοίωσης. Συμπερασματικά, τα διαγράμματα ροής περιγράφουν μαθηματικά τις σχέσεις επιρροής μεταξύ του συνόλου των εμπλεκόμενων παραγόντων του πραγματικού συστήματος σε ένα μοντέλο προσομοίωσης και αποδίδουν την αναμενόμενη απόκριση του πραγματικού συστήματος.

5.4 Λογισμικά προγραμματισμού Δυναμικής Συστημάτων

Τα σημαντικότερα προγράμματα προσομοίωσης που έχουν αναπτυχθεί για την επίλυση μοντέλων διαμορφωμένα με τους κανόνες της Δυναμικής Συστημάτων είναι τα εξής:

DYNAMO[®]: Το πρώτο λογισμικό προγραμματισμού Δυναμικής Συστημάτων ήταν το DYNAMO[®] (Dynamic Modeling) το οποίο δημιουργήθηκε στις αρχές του 1960 στο MIT από τον Jack Pugh υπό την καθοδήγηση του Jay Forrester. Βασική λειτουργία του λογισμικού αποτελεί η ανάπτυξη διαφορικών εξισώσεων που απεικονίζουν μοντέλα Δυναμικής Συστημάτων. Το λογισμικό αυτό είναι πλέον ξεπερασμένο και είναι συμβατό μόνο με το λειτουργικό περιβάλλον DOS[®].

DYSMAP[®]: Το πανεπιστήμιο του Salford (UK) δημιούργησε το 1978 το λογισμικό προγραμματισμού DYSMAP[®] (Dynamic System Modeling and Analysis Package) το οποίο έχει παρόμοια σύνταξη με αυτή του DYNAMO[®] και περιλαμβάνει επιπλέον τη δυνατότητα βελτιστοποίησης με την τεχνική hill-climbing. Η συμβατότητά του περιορίζεται επίσης μόνο με το λειτουργικό περιβάλλον DOS[®].

Stella[®], i-think[®]: Η εταιρεία High Performance Systems δημιούργησε αρχικά το 1984 το πρόγραμμα προσομοίωσης Stella[®] για το λειτουργικό περιβάλλον Macintosh[®] και το 1985 το αντίστοιχο λογισμικό προσομοίωσης i-think[®] για το λειτουργικό περιβάλλον Windows[®]. Τα λογισμικά αυτά βοηθούν στην κατανόηση και βελτίωση του τρόπου λειτουργίας φυσικών, κοινωνικών και σύνθετων επιχειρηματικών συστημάτων μέσω της γραφικής απεικόνισής τους. Μάλιστα τα λογισμικά αυτά είναι τα πρώτα τα οποία εισήγαγαν τη γραφική απεικόνιση των συστημάτων που προσομοιώνονται.

Vensim[®]: Η εταιρεία Ventana Systems του πανεπιστημίου Harvard της Μασαχουσέτης δημιούργησε το λογισμικό Vensim[®] (Ventana Simulation) για την ανάπτυξη και ανάλυση επιχειρησιακών και τεχνολογικών μοντέλων προσομοίωσης. Τέτοια μοντέλα έχουν επιτυχώς εφαρμοστεί στους τομείς των προβλέψεων, της λειτουργίας μίας επιχείρησης και στη σχεδίαση στρατηγικών για την αγορά προϊόντων. Το λογισμικό Vensim[®] είναι συμβατό με το λειτουργικό περιβάλλον Windows[®] και Macintosh[®] και η πρώτη του έκδοση εμφανίστηκε το 1988 ενώ η διάθεσή του σήμερα είναι δωρεάν.

Powersim[®]: Η εταιρεία Micro Worlds είναι η δημιουργός του λογισμικού δυναμικών μοντέλων Powersim[®] (Power Simulation) το οποίο προσφέρει μία ολοκληρωμένη δυνατότητα απεικόνισης πραγματικών συστημάτων. Η γλώσσα Powersim[®] εμφανίστηκε για πρώτη φορά το 1988 και είναι συμβατή μόνο με το λειτουργικό περιβάλλον των Windows[®]. Βασικές λειτουργίες του λογισμικού αποτελούν ο στρατηγικός σχεδιασμός και η κατανόηση της λειτουργίας των διαφόρων συστημάτων με σκοπό την αποδοτικότερη διαχείρισή τους.

5.5 Η λειτουργία της γλώσσας προγραμματισμού Powersim[®] 2.5c

Για την προσομοίωση του μοντέλου που αναπτύχθηκε στην παρούσα μεταδιπλωματική εργασία χρησιμοποιείται η γλώσσα προγραμματισμού Powersim[®] 2.5c. Οι απαιτήσεις του λογισμικού αυτού, σύμφωνα με το Users Guide της Powersim[®] 2.5c (1996), είναι οι εξής:

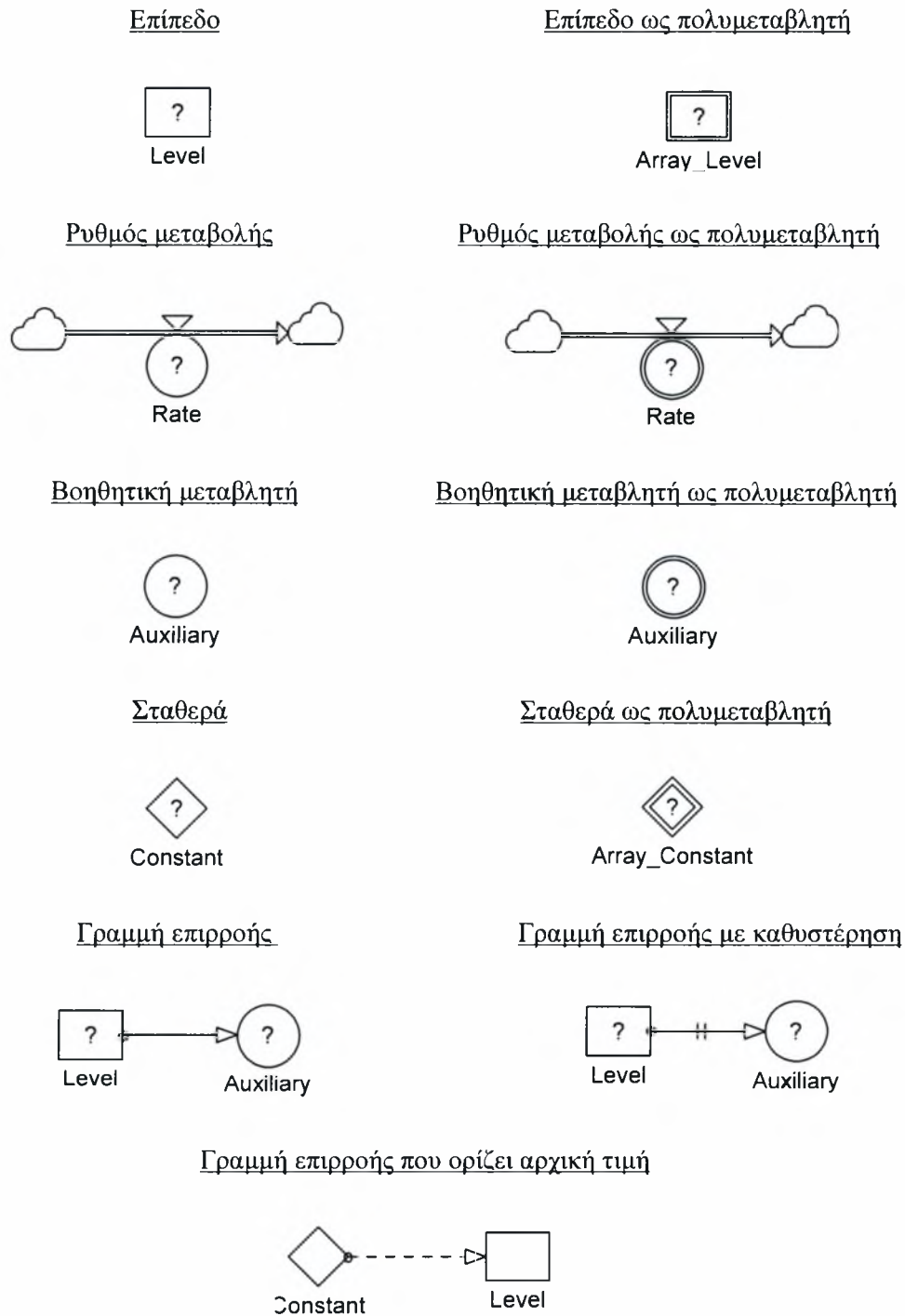
- Η/Υ (PC) συμβατός με IBM, με επεξεργαστή 80386 τουλάχιστον.
- Λειτουργικό σύστημα Windows[®] 3.1x, Nt ή '95 ή μεταγενέστερη έκδοση.
- Μνήμη άμεσης προσπέλασης (RAM) 4÷8 Mb
- Σκληρός δίσκος με ελάχιστο διαθέσιμο χώρο 10÷20 Mb.

Ένα από τα σημαντικότερα πλεονεκτήματα αυτής της γλώσσας είναι οι πολύ χαμηλές υπολογιστικές απαιτήσεις που έχει, λόγω των οποίων αυξάνεται σημαντικά η ταχύτητα προσομοίωσης σε ισχυρούς Η/Υ. Επιπλέον, λόγω της απεικόνισης του μοντέλου απλοποιείται η σύνταξη του κώδικα προγραμματισμού ο οποίος μπορεί και να αντλεί τυποποιημένες μαθηματικές συναρτήσεις από τη βιβλιοθήκη εντολών που είναι διαθέσιμη κατά τον ορισμό κάθε μεταβλητής (μαθηματικές, στατιστικές, τριγωνομετρικές, χρονικές, χρηματοοικονομικές, λογικές, υποθέσεων, καθυστερήσεων, ελέγχου, επεξεργασίας «ιστορικών» δεδομένων, πολυμεταβλητών (arrays), γραφικής απεικόνισης, τυχαιότητας και μετατροπής). Συνεπώς απλοποιείται η διαδικασία και μειώνεται ο χρόνος διαμόρφωσης του μοντέλου λόγω της ταυτόχρονης διαμόρφωσης του διαγράμματος ροής και του μοντέλου. Τα αποτελέσματα της προσομοίωσης μπορούν να αποδοθούν σε πίνακες, διαγράμματα και άλλες γραφικές απεικονίσεις ενώ υπάρχει και δυνατότητα εξαγωγής δεδομένων και σε άλλα προγράμματα όπως το Microsoft Excel[®] και το WinWord[®].

Σημαντικό πλεονέκτημα του λογισμικού αποτελεί η δυνατότητα χρησιμοποίησης πολυμεταβλητών (arrays) με την οποία μειώνεται δραστικά ο αριθμός των μεταβλητών ενός μοντέλου ενώ παράλληλα απλοποιείται και το διάγραμμα ροής. Οι πολυμεταβλητές μπορεί να είναι επίπεδα, ρυθμοί, βοηθητικές μεταβλητές ή και σταθερές. Παραδείγματος χάριν, το προσωπικό μίας επιχείρησης μπορεί να κατηγοριοποιηθεί σε τρεις βαθμίδες (κατώτερη, μεσαία και ανώτερη) οι οποίες να περιγράφονται με μία μόνο πολυμεταβλητή.

Ένα ακόμα βασικό πλεονέκτημα της Powersim[®] αποτελεί η δυνατότητά της να λειτουργεί σε δίκτυο Η/Υ. Με τον τρόπο αυτό επιτυγχάνεται αύξηση της υπολογιστική ισχύος με ταυτόχρονη αύξηση της ταχύτητας της προσομοιωτικής διαδικασίας ενώ παράλληλα δίνεται η δυνατότητα σε περισσότερα του ενός άτομα να παρακολουθούν τη διαδικασία και την εξέλιξη της προσομοίωσης και να επέμβουν σε αυτήν αν υπάρχει κάποιος λόγος.

Στο Σχήμα 5 παρουσιάζονται οι συμβολισμοί των κυριότερων μεταβλητών καθώς και τα είδη των γραμμών επιρροής που χρησιμοποιούνται κατά τη δημιουργία των διαγραμμάτων ροής στη γλώσσα προγραμματισμού Powersim® 2.5c.



Σχήμα 5. Συμβολισμός των κυριότερων μεταβλητών και είδη γραμμών επιρροής

6 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΜΟΝΤΕΛΟΥ ΔΥΝΑΜΙΚΗΣ ΣΥΣΤΗΜΑΤΩΝ

6.1 Εσωτερικό και εξωτερικό περιβάλλον του συστήματος

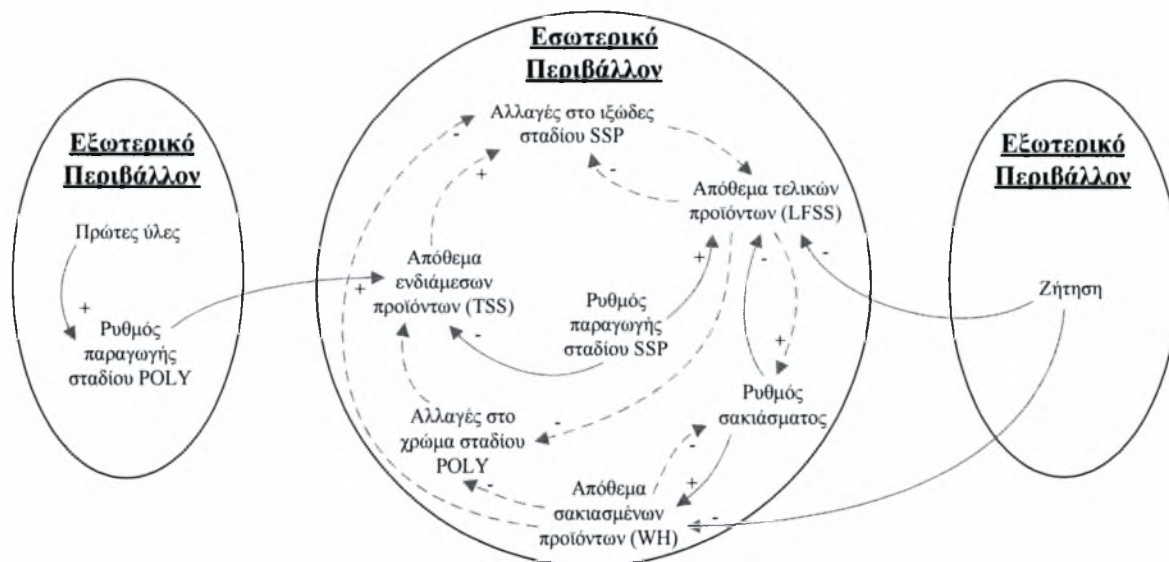
Σύμφωνα με τους Liberopoulos et al.^a (2009) το *εσωτερικό περιβάλλον* του συστήματος αποτελούν τα στοιχεία:

- Στάδιο καθορισμού χρώματος ημιτεϊμων προϊόντων (POLY)
- Στάδιο ενδιάμεσης αποθήκευσης ημιτεϊμων προϊόντων (TSS)
- Στάδιο καθορισμού ιξώδους τελικών προϊόντων (SSP)
- Στάδιο αποθήκευσης τελικών προϊόντων σε σιλό (LFSS) και στην αποθήκη (WH)
- Στάδιο σακιάσματος.

Το *εξωτερικό περιβάλλον* του συστήματος αποτελούν τα στοιχεία:

- Προμηθευτές πρώτων υλών
- Ζήτηση προϊόντων PET.

Στο Σχήμα 6 παρουσιάζεται η οριοθέτηση του συστήματος όπου φαίνονται οι εσωτερικές και οι εξωτερικές επιδράσεις και οι σχέσεις του συστήματος.



Σχήμα 6. Οριοθέτηση του συστήματος με τις εσωτερικές και τις εξωτερικές επιδράσεις

6.2 Υποθέσεις του δυναμικού μοντέλου

Όπως είναι φυσικό, κάθε πραγματικό σύστημα εμπεριέχει πληθώρα στοιχείων τα οποία είτε είναι αδύνατο είτε θεωρείται μη σκόπιμο (από απόψεως χρόνου και συνθετότητας) να απεικονιστούν στο μοντέλο. Η ορθολογικότερη ίσως προσέγγιση η οποία θεωρείται σκόπιμο να χρησιμοποιείται προτείνει η εμβάθυνση στις λεπτομέρειες περιγραφής ενός συστήματος να σταματάει εκεί όπου η περαιτέρω περιγραφή επιβαρύνει σημαντικά την πολυπλοκότητα του μοντέλου αλλά ταυτόχρονα η μη συμμετοχή αυτής της περιγραφής δεν αλλοιώνει την ποιότητα των αποτελεσμάτων (το σφάλμα το οποίο υπεισέρχεται να είναι αμελητέο).

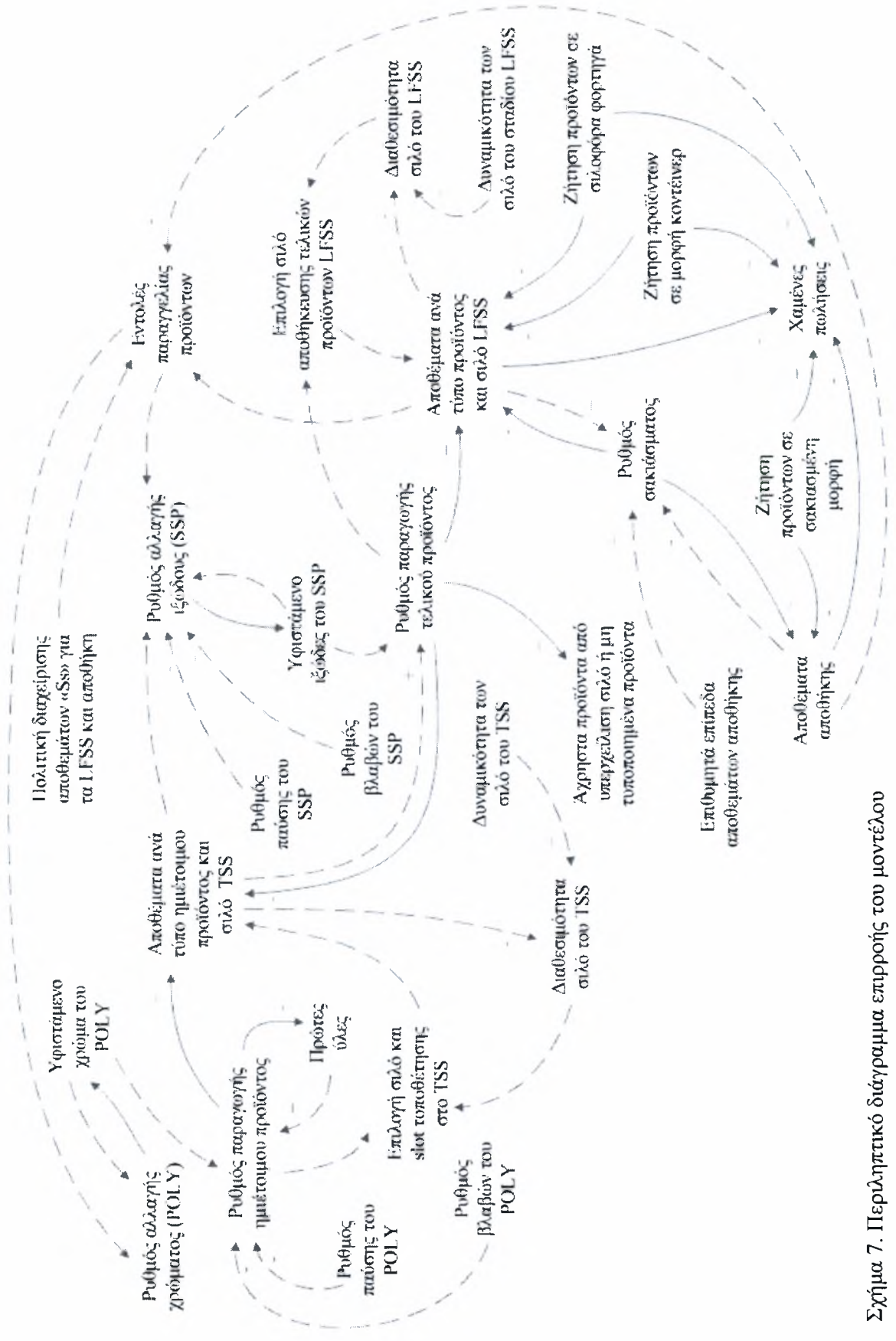
Κατά την κατάστροψη του μοντέλου ΔΣ λαμβάνεται υπόψη το σύνολο των υποθέσεων που σημειώθηκαν κατά την περιγραφή του γραμμικού μοντέλου, σημαντικότερη εκ των οποίων θεωρείται η λήψη αποφάσεων μόνο στην αρχή κάθε τετράωρου και όχι κατά τη διάρκειά του. Η ύπαρξη αυτής της υπόθεσης προκύπτει από την ανάγκη συμβατότητας του μοντέλου ΔΣ με το γραμμικό μοντέλο των Liberopoulos et al.³ (2009) προκειμένου τα προκύπτοντα αποτελέσματα να είναι άμεσα συγκρίσιμα. Ειδικότερα, κατά τη μείωση της χρονικής διάρκειας του βήματος ενός μοντέλου (μείωση του Δt) πολλαπλασιάζεται ο αριθμός των μεταβλητών του. Παρόλο που για τα προσομοιωτικά μοντέλα αυτό το γεγονός δε συνεπάγεται καμία σχεδόν επιβάρυνση ούτε στο χρόνο προσομοίωσης αλλά ούτε και στην πολυπλοκότητα του μοντέλου, δεν ισχύει το ίδιο για ένα αναλυτικό μοντέλο καθώς ο χρόνος επίλυσής εξαρτάται άμεσα από το πλήθος των μεταβλητών απόφασης που πρέπει να βελτιστοποιηθούν. Στη γενική περίπτωση, αύξηση του χρονικού βήματος συνεπάγεται μείωση της ευελιξίας της διαδικασίας προσομοίωσης για το μοντέλο ΔΣ και ταυτόχρονη μείωση της ποιότητας της βέλτιστης πολιτικής για το γραμμικό μοντέλο. Παρά λοιπόν το γεγονός ότι το μοντέλο ΔΣ θα μπορούσε να έχει πολύ μικρότερο χρονικό βήμα από το τετράωρο, ωστόσο σκοπός της δημιουργίας του παραμένει η σύγκριση των αποτελεσμάτων των δύο μεθοδολογιών στο ίδιο σύστημα, με τις ίδιες υποθέσεις και απλοποιήσεις ώστε τα αποτελέσματά τους να είναι άμεσα συγκρίσιμα.

Μία επιπλέον υπόθεση του μοντέλου ΔΣ είναι ότι μεταξύ των τελικών προϊόντων της βιομηχανίας υπάρχει διαφορετική προτεραιότητα και ειδικότερα θεωρείται ότι το προϊόν WG έχει προτεραιότητα 1, το SD έχει 2, το FH έχει 3 ενώ το G έχει 4. Η ανάγκη ύπαρξης προτεραιοτήτων προκύπτει από τη διαχείριση των αποθεμάτων από πολιτικές τύπου «Ss» στις οποίες πρέπει να οριστούν προτεραιότητες. Επιπλέον το προϊόν G αντιμετωπίζεται ως τελικό προϊόν μηδενικής ζήτησης και όχι ως παραπροϊόν διότι είναι επιθυμητό να ανευρεθεί πελάτης ο οποίος θα το αγοράζει σε τιμή τελικού προϊόντος. Τέλος το μοντέλο περιέχει το σύνολο των παραγόντων που μπορεί να επιδρούν στη λειτουργία του, όπως το σύνολο των σιλό του σταδίου TSS και ρυθμούς βλαβών του εξοπλισμού οι οποίοι όμως παραμένουν ανενεργοί λόγω της επιθυμίας συμβατότητας των εξαγόμενων αποτελεσμάτων.

6.3 Διάγραμμα επιρροής

Στο Σχήμα 7 παρουσιάζεται το περιληπτικό διάγραμμα επιρροής της παραγωγικής διαδικασίας της χημικής βιομηχανίας, τα διάφορα στάδια αποθήκευσης καθώς και οι διάφορες διαδικασίες λήψης αποφάσεων για την λειτουργία του, σύμφωνα με την ορολογία που περιγράφηκε στα προηγούμενα κεφάλαια. Στο σχήμα αυτό οι φυσικές ροές παρουσιάζονται με συνεχείς γραμμές ενώ οι ροές πληροφοριών παρουσιάζονται με διακεκομμένες. Σημειώνεται ότι στις παραγράφους που ακολουθούν παρουσιάζεται αναλυτικά το μοντέλο που αναπτύχθηκε ενώ στο Παράρτημα Α παρουσιάζεται και ο κώδικας στη γλώσσα προγραμματισμού Powersim[®] 2.5c.

Ειδικότερα, λαμβάνοντας υπόψη την «*Πολιτική διατήρησης αποθεμάτων Ss για τα LFSS και αποθήκη*», τα «*Αποθέματα αποθήκης*» και τα «*Αποθέματα ανά είδος προϊόντος και σιλό LFSS*» προκύπτουν οι «*Εντολές παραγγελίας προϊόντος*» ώστε να συμπληρωθούν τα αποθέματα τελικών προϊόντων. Με βάση το «*Ρυθμό βλαβών του POLY*», το «*Ρυθμό παύσης του POLY*» (π.χ. για λόγους προγραμματισμένης συντήρησης) και το «*Υφιστάμενο χρώμα του POLY*» προκύπτει ο «*Ρυθμός αλλαγής χρώματος (POLY)*» ο οποίος μεταβάλλει (ή όχι) το χρώμα του σταδίου POLY ώστε να καλύψει τις «*Εντολές παραγγελίας προϊόντος*». Οι «*Πρώτες ύλες*» χρησιμοποιούνται από το «*Ρυθμό παραγωγής ημιέτοιμου προϊόντος*» ο οποίος σε συνδυασμό με το «*Υφιστάμενο χρώμα του POLY*» παράγει ημιέτοιμα προϊόντα που έχουν χρωματιστεί. Προκειμένου να γίνει «*Επιλογή σιλό και slot τοποθέτησης στο TSS*» των ημιετοιμών προϊόντων, ελέγχεται η «*Διαθεσιμότητα σιλό του TSS*» η οποία προκύπτει από τη «*Δυναμικότητα των σιλό του TSS*» και τα «*Αποθέματα ανά είδος ημιέτοιμου προϊόντος και σιλό TSS*» τα οποία μειώνονται από το «*Ρυθμό παραγωγής τελικού προϊόντος*». Ο τελευταίος προκύπτει από το συνδυασμό του χρώματος που φέρει ήδη το προϊόν και το «*Υφιστάμενο ιξώδες του SSP*», το οποίο προκύπτει από το «*Ρυθμό αλλαγής ιξώδους (SSP)*» συναρτήσει του «*Ρυθμού παύσης του SSP*», του «*Ρυθμού βλαβών του SSP*», των αποθεμάτων του TSS και των «*Εντολών παραγγελίας προϊόντων*». Σε περίπτωση που είτε ο συνδυασμός χρώματος και ιξώδους του παραγόμενου προϊόντος δεν είναι κάποιος από τους αποδεκτούς είτε δεν υπάρχει διαθέσιμος χώρος αποθήκευσης του παραγόμενου προϊόντος, τότε το προϊόν απορρίπτεται μέσω της αντίστοιχης μεταβλητής. Προκειμένου να γίνει «*Επιλογή σιλό αποθήκευσης τελικών προϊόντων LFSS*» ελέγχεται η «*Διαθεσιμότητα σιλό του LFSS*» μέσω της «*Δυναμικότητας σιλό του LFSS*» και των «*Αποθεμάτων ανά είδος προϊόντος και σιλό LFSS*». Έχοντας επιλέξει το κατάλληλο σιλό αποθήκευσης, ο «*Ρυθμός παραγωγής τελικού προϊόντος*» αυξάνει τα «*Αποθέματα ανά είδος προϊόντος και σιλό LFSS*» τα οποία μειώνονται από τη «*Ζήτηση προϊόντων σε μορφή κοντέινερ*», τη «*Ζήτηση προϊόντων σε σιλοφόρα φορτηγά*» και το «*Ρυθμό σακιάσματος*» ο οποίος αυξάνει τα «*Αποθέματα αποθήκης*» τα οποία με τη σειρά τους μειώνονται από τη «*Ζήτηση προϊόντων σε σακιασμένη μορφή*». Σημειώνεται ότι όσες παραγγελίες δεν ικανοποιούνται άμεσα χάνονται («*Χαμένες πωλήσεις*») καθώς δεν υπάρχει δυνατότητα καθυστερημένης ικανοποίησης της ζήτησης.



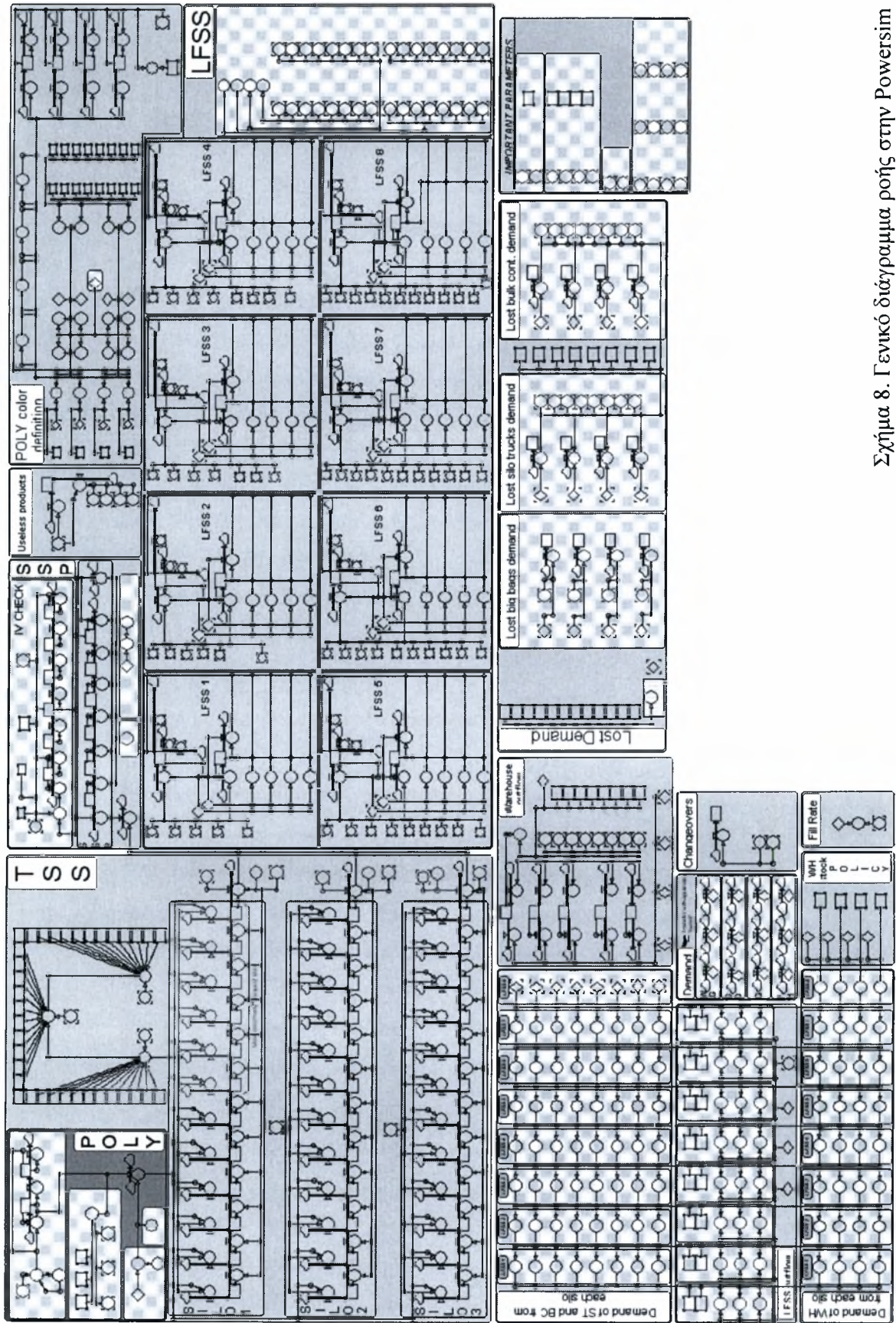
Σχήμα 7. Περιληπτικό διάγραμμα επιρροής του μοντέλου

6.4 Διάγραμμα ροής στην Powersim® 2.5c

6.4.1 Γενικά

Στο Σχήμα 8 παρουσιάζεται το διάγραμμα ροής του μοντέλου που δημιουργήθηκε στη γλώσσα προγραμματισμού Powersim® 2.5c και το οποίο περιλαμβάνει 89 μεταβλητές επιπέδου, 185 μεταβλητές ρυθμών, 337 βοηθητικές μεταβλητές, 30 σταθερές και 3.521 σχέσεις επιρροής. Το μοντέλο χωρίζεται στους τομείς παραγωγής που περιγράφηκαν στο Κεφάλαιο 2 καθώς και σε τομείς περιγραφής συγκεκριμένων ενδογενών διαδικασιών λήψης αποφάσεων. Κάθε ένας από αυτούς περιγράφεται μέσω ενός πλήθους μεταβλητών οι οποίες ορίζονται από συγκεκριμένες συναρτήσεις που παρουσιάζονται αναλυτικότερα στις παραγράφους που ακολουθούν, ενώ το σύνολο του προγράμματος παρουσιάζεται στο Παράρτημα Α.

Εν συντομία και ξεκινώντας από την επάνω αριστερή γωνία του Σχήματος 8, η πρώτη ύλη διέρχεται από το παραγωγικό στάδιο πολυμερισμού (POLY) όπου της προσδίδεται χρώμα με βάση τις ανάγκες αναπλήρωσης των αποθεμάτων στα σιλό του LFSS και στην αποθήκη. Στη συνέχεια το ημιέτοιμο προϊόν εξέρχεται του POLY και μέσω μίας ενδογενούς διαδικασίας επιλογής σιλό και θέσης αποθήκευσης (slot), που στο σχήμα βρίσκεται δίπλα και κάτω από το POLY, αποθηκεύεται σε ένα από τα τρία σιλό του σταδίου TSS. Από εκεί, το ημιέτοιμο προϊόν εισέρχεται στο στάδιο SSP, που στο σχήμα βρίσκεται δεξιά των POLY και TSS, όπου μέσω μίας ενδογενούς διαδικασίας που λαμβάνει υπόψη το χρώμα της διερχόμενης ύλης και τις ανάγκες αναπλήρωσης των αποθεμάτων, προσδιορίζεται το ιξώδες του σταδίου. Από το συνδυασμό χρώματος και ιξώδους προκύπτει το τελικό προϊόν το οποίο μέσω μίας ενδογενούς διαδικασίας ανεύρεσης του βέλτιστου σιλό αποθήκευσης, που στο σχήμα βρίσκεται κάτω από το SSP, τοποθετείται σε ένα από τα οκτώ σιλό του LFSS. Σε περίπτωση που είτε τα σιλό του LFSS γεμίσουν είτε ο συνδυασμός χρώματος και ιξώδους δεν είναι κάποιος από τους αποδεκτούς, η όποια παραγόμενη ποσότητα απορρίπτεται από το σύστημα μέσω των εντολών του τομέα Spill over που στο σχήμα βρίσκεται δεξιά του SSP. Η εξερχόμενη ποσότητα από τα σιλό του LFSS, που στο σχήμα βρίσκεται κάτω από το TSS, καλύπτει τη ζήτηση μορφής κοντέινερ και σιλοφόρων φορτηγών καθώς και τις ανάγκες αναπλήρωσης των αποθεμάτων της αποθήκης μέσω του τομέα σακιάσματος προϊόντων που βρίσκεται στο κάτω μέρος του σχήματος. Σημειώνεται ότι η ζήτηση που δεν ικανοποιείται άμεσα χάνεται και η τιμή της υπολογίζεται στον τομέα που βρίσκεται στο κάτω δεξί μέρος του σχήματος. Τέλος, στον τομέα που βρίσκεται στο πάνω δεξί μέρος του σχήματος παρουσιάζεται η ενδογενής διαδικασία επιλογής χρώματος του POLY για την αναπλήρωση των αποθεμάτων με βάση τις πολιτικές «Ss» που απεικονίζονται σε αυτόν τον τομέα για το LFSS και στον τομέα που βρίσκεται στο κάτω μέρος του σχήματος για την αποθήκη (Warehouse=WH). Υπενθυμίζεται ότι στόχο αποτελεί η μείωση των αλλαγών των POLY και SSP που υπολογίζονται στο κάτω μέρος του σχήματος, με ταυτόχρονη μείωση της ανικανοποίητης ζήτησης.

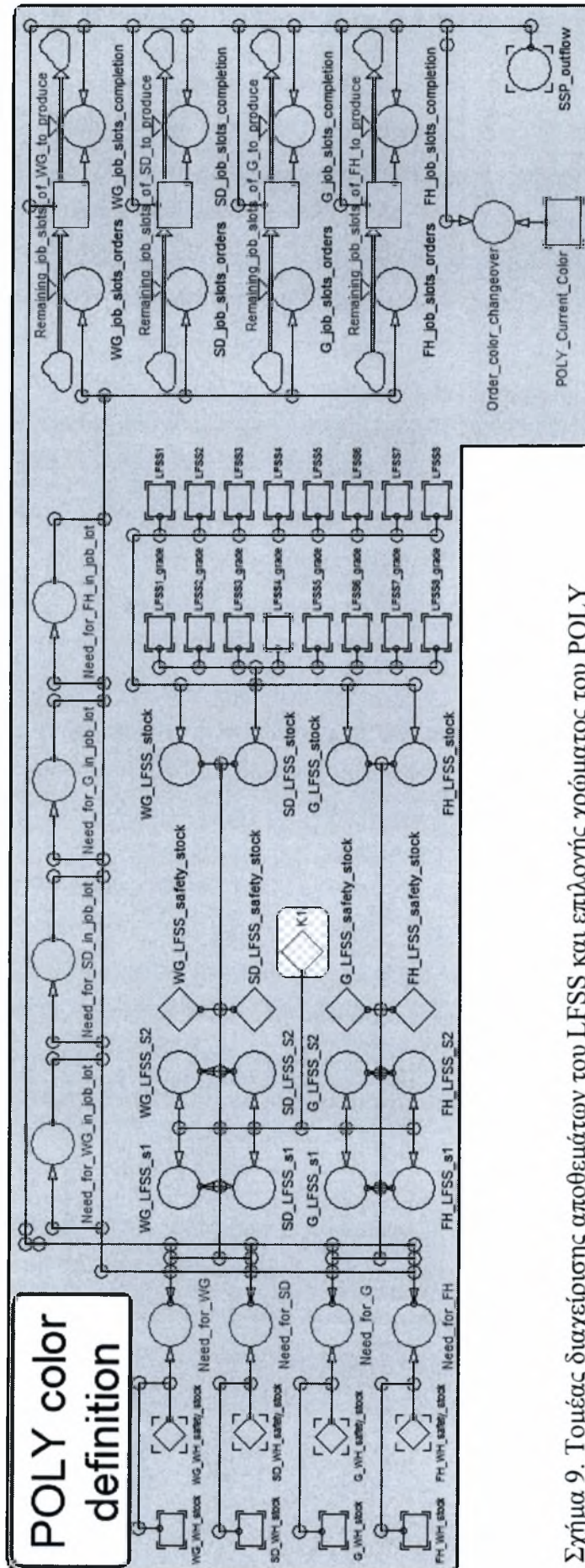


Σχήμα 8. Γενικό διάγραμμα ροής στην Powersim

6.4.2 Διαχείριση αποθεμάτων του LFSS και επιλογή χρώματος του POLY

Στο Σχήμα 9 παρουσιάζεται η διαχείριση αποθεμάτων του LFSS μέσω πολιτικών Ss. Ειδικότερα, αντί των S και s χρησιμοποιούνται οι στάθμες S2 και s1 αντίστοιχα οι οποίες προκύπτουν ως ποσοστά των επιθυμητών αποθεμάτων ανά προϊόν [π.χ. $S2=(1+K1)*WG_LFSS_safety_stock$, $s1=(1-K1)*WG_LFSS_safety_stock$]. Στη συνέχεια ελέγχεται αν τα συνολικά αποθέματα των σιλό του LFSS ανά προϊόν (π.χ. WG_LFSS_stock) τα οποία προκύπτουν λαμβάνοντας υπόψη ανά σιλό το είδος και την ποσότητα του περιεχόμενου προϊόντος (π.χ. $LFSS1_grade$, $LFSS1$), είναι πάνω ή κάτω από τη στάθμη παραγγελίας του (π.χ. WG_LFSS_s1): αν είναι πάνω από αυτή τότε δε δίνεται νέα παραγγελία ενώ αν είναι κάτω από αυτή τότε παραγγέλλεται ποσότητα ικανή να αναπληρώσει τα αποθέματα τόσο του LFSS όσο και της αποθήκης. Συγκεκριμένα, η ποσότητα παραγγελίας ανά προϊόν (π.χ. $Need_for_WG$) προκύπτει από το άθροισμα δύο διαφορών: πρώτον, της διαφοράς μεταξύ της άνω στάθμης της αντίστοιχης πολιτικής Ss (π.χ. WG_LFSS_S2) και των συνολικών αποθεμάτων του προϊόντος στο LFSS (π.χ. WG_LFSS_stock) και δεύτερον, της διαφοράς μεταξύ των επιθυμητών αποθεμάτων του προϊόντος στην αποθήκη (π.χ. $WG_WH_safety_stock$) και του υφιστάμενου αποθέματός του στην αποθήκη (π.χ. WG_WH_stock). Η ποσότητα παραγγελίας μετατρέπεται από τόνους σε παρτίδες παραγωγής των 33,33 τόνων (π.χ. $Need_for_WG_in_job_lot$) με αποτέλεσμα την αύξηση των παρτίδων παραγωγής που εκκρεμούν (π.χ. $Remaining_job_slots_of_WG_to_produce$), οι οποίες μειώνονται από την παραγωγή τελικών προϊόντων.

Οι εντολές αλλαγής χρώματος στο στάδιο POLY ($Order_color_changeover$) προκύπτουν από μία ενδογενή διαδικασία η οποία λαμβάνει υπόψη τις παραγγελίες που εκκρεμούν (π.χ. $Remaining_job_slots_of_WG_to_produce$) και το υφιστάμενο χρώμα (π.χ. $POLY_Current_Color$): όταν εκκρεμούν παραγγελίες προϊόντων που έχουν χρώμα ίδιο με το υφιστάμενο ή όταν δεν εκκρεμούν παραγγελίες προϊόντων με χρώμα διαφορετικό από το υφιστάμενο, τότε το χρώμα δεν αλλάζει. Η τακτική αυτή έχει ως στόχο τη μείωση των αλλαγών χρώματος καθώς προτού αλλάξει το υφιστάμενο χρώμα εξαντλείται η τρέχουσα απαίτηση για παραγωγή προϊόντων με το χρώμα αυτό μέσω της αναπλήρωσης του συνόλου των αποθεμάτων των τους. Όταν ολοκληρώνεται αυτή η διαδικασία αναπλήρωσης των αποθεμάτων και υπάρχει ανάγκη παραγωγής προϊόντων με διαφορετικό χρώμα τότε ενεργοποιείται η διαδικασία μετάβασης από το υφιστάμενο χρώμα στο νέο (περιγράφεται στη συνέχεια) μέσω του ενδιάμεσου χρώματος Gray το οποίο δημιουργεί παραπροϊόντα χαμηλών ποιοτικών χαρακτηριστικών που πωλούνται δυσκολότερα και σε χαμηλότερη τιμή.

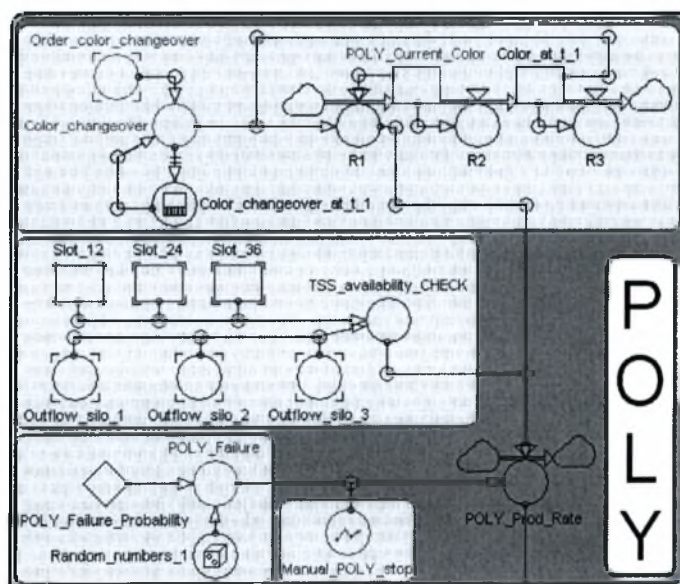


Σχήμα 9. Τομέας διαχείρισης αποθεμάτων του LFSS και επιλογής χρώματος του POLY

6.4.3 Παραγωγικό στάδιο πολυμερισμού POLY

Στο Σχήμα 10 παρουσιάζεται η λειτουργία του παραγωγικού σταδίου POLY. Συγκεκριμένα ο ρυθμός πολυμερισμού (POLY_Prod_Rate) ελέγχει πρώτον, αν υπάρχει βλάβη στον εξοπλισμό του POLY (POLY_Failure), δεύτερον αν υπάρχει εντολή παύσης της παραγωγής λόγω προγραμματισμένης συντήρησης (Manual_POLY_stor) και τρίτον αν υπάρχει διαθέσιμος χώρος στα σιλό του TSS για την αποθήκευση της παραγόμενης ποσότητας ημιτέτοιμου προϊόντος (TSS_availability_CHECK). Αν οποιοσδήποτε από τους παραπάνω ελέγχους είναι «αληθής» τότε η παραγωγή του POLY διακόπτεται ενώ σε αντίθετη περίπτωση παράγεται σταθερή ποσότητα ημιτέτοιμου προϊόντος ίση με 33,33 τόνων/4hr, με χρώμα που προκύπτει συναρτήσεως του υφιστάμενου χρώματος και της ενδεχόμενης εντολής αλλαγής του. Σημειώνεται πρώτον, ότι για να ενεργοποιηθεί μία εντολή αλλαγής χρώματος θα πρέπει πρώτα να έχει ολοκληρωθεί οποιαδήποτε τυχόν προηγούμενη εντολή αλλαγής χρώματος (δηλαδή θα πρέπει Color_changeover_at_t_1=0) και δεύτερον, ότι επιτρέπεται η αλλαγή χρώματος κατά τη διάρκεια μίας διακοπής λόγω βλάβης ή προγραμματισμένης συντήρησης λόγω των τεχνολογικών δυνατοτήτων του εξοπλισμού (διαδικασία κάθετη στη διαδικασία της παραγωγής).

Κατά την ενεργοποίηση μίας εντολής αλλαγής χρώματος ελέγχεται ποιο είναι το υφιστάμενο χρώμα και ποιο ήταν το προηγούμενό του (π.χ στην περίπτωση όπου το τρέχων χρώμα είναι το Gray πρέπει να είναι γνωστό ποιο ήταν το προηγούμενο χρώμα ώστε να προσδιοριστεί το νέο χρώμα). Η παρακολούθηση της χρονοσειράς χρώματος του POLY επιτυγχάνεται μέσω των μεταβλητών που σημειώνονται στο επάνω μέρος του Σχήματος 10 (R1, POLY_Current_Color, R2, Color_at_t_1, R3) όπου στην πραγματικότητα απλώς απομνημονεύεται για ένα μόνο dt το χρώμα που είχε το POLY την προηγούμενη χρονική στιγμή.

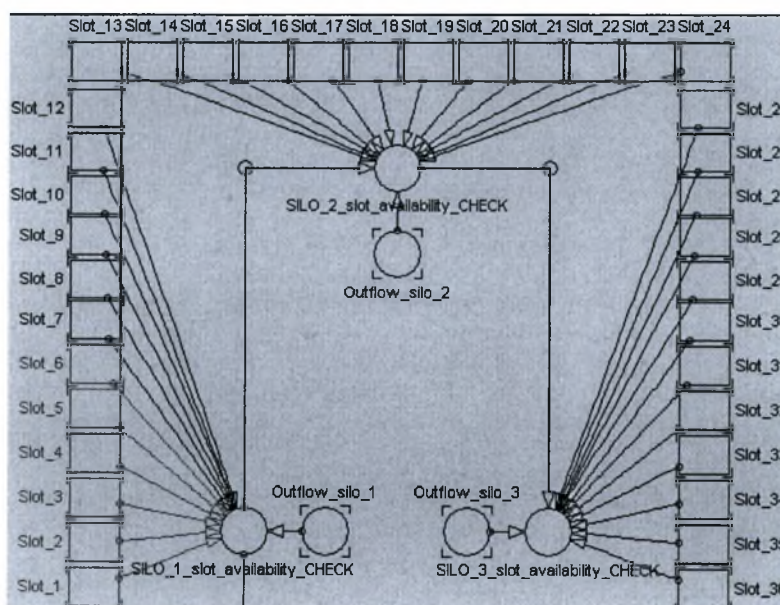


Σχήμα 10. Λειτουργία παραγωγικού σταδίου POLY

6.4.4 Επιλογή σιλό και slot αποθήκευσης ημιετοιμών προϊόντων στο TSS

Στο Σχήμα 11 παρουσιάζεται η διαδικασία επιλογής σιλό και slot αποθήκευσης ημιετοιμών προϊόντων στο στάδιο ενδιάμεσης αποθήκευσης TSS. Ειδικότερα ελέγχεται με σειρά προτεραιότητας αν η ποσότητα παραγωγής του POLY μπορεί να εισέλθει στο πρώτο, στο δεύτερο ή στο τρίτο σιλό συναρτήσει των άδειων slot και του σιλό από το οποίο υπάρχει εκροή προϊόντων προς το στάδιο πολυμερισμού SSP. Η εκροή των σιλό ελέγχεται διότι ο κανόνας εύρεσης του slot τοποθέτησης μέσα σε ένα σιλό εξαρτάται από το εάν υπάρχει ή όχι εκροή από αυτό. Για παράδειγμα εάν το σιλό τοποθέτησης έχει έξι slot κατειλημμένα από προϊόν και υπάρχει εκροή από αυτό τότε η εισερχόμενη ποσότητα θα αποθηκευτεί στο slot 6 που αδειάζει, ενώ σε αντίθετη περίπτωση η ποσότητα αυτή θα αποθηκευτεί στο slot 7. Όταν ένα σιλό γεμίσει και δεν υπάρχει εκροή από αυτό, τότε η διαδικασία ανεύρεσης διαθέσιμου slot ελέγχει το επόμενο σιλό ενώ αν γεμίσουν όλα τα σιλό και δεν υπάρχει εκροή από αυτά τότε διακόπτεται η παραγωγή του σταδίου POLY. Σύμφωνα με τις παραδοχές των Liberopoulos et al.^a (2009), το εξερχόμενο προϊόν του σταδίου POLY μπορεί να εισέλθει μόνο σε ένα σιλό ανά τετράωρο από τα τρία του σταδίου TSS ενώ το πολύ ένα από αυτά μπορεί να έχει θετική εκροή προς το παραγωγικό στάδιο SSP.

Κατά τις συνθήκες ομαλής λειτουργίας της βιομηχανίας χρησιμοποιείται μόνο το σιλό 1 και συγκεκριμένα μόνο τα έξι πρώτα του slot, καθώς η εισροή προϊόντων είναι ίση με την εκροή (εξισορροπημένη ροή). Στην περίπτωση αυτή το παραγόμενο προϊόν του σταδίου POLY εισέρχεται στο slot 6 του σιλό 1 από όπου προωθείται κατά ένα slot ανά τετράωρο προς το slot 1 και την έξοδο από το σιλό. Το σιλό 2 χρησιμοποιείται μόνο αν το σιλό 1 γεμίσει και η εκροή του είναι μηδενική οπότε ο έλεγχος οδηγεί τα προϊόντα στο σιλό 2 κι όταν γεμίσει κι αυτό, στο σιλό 3.



Σχήμα 11. Έλεγχος και επιλογή σιλό και slot αποθήκευσης ημιετοιμών προϊόντων στο TSS

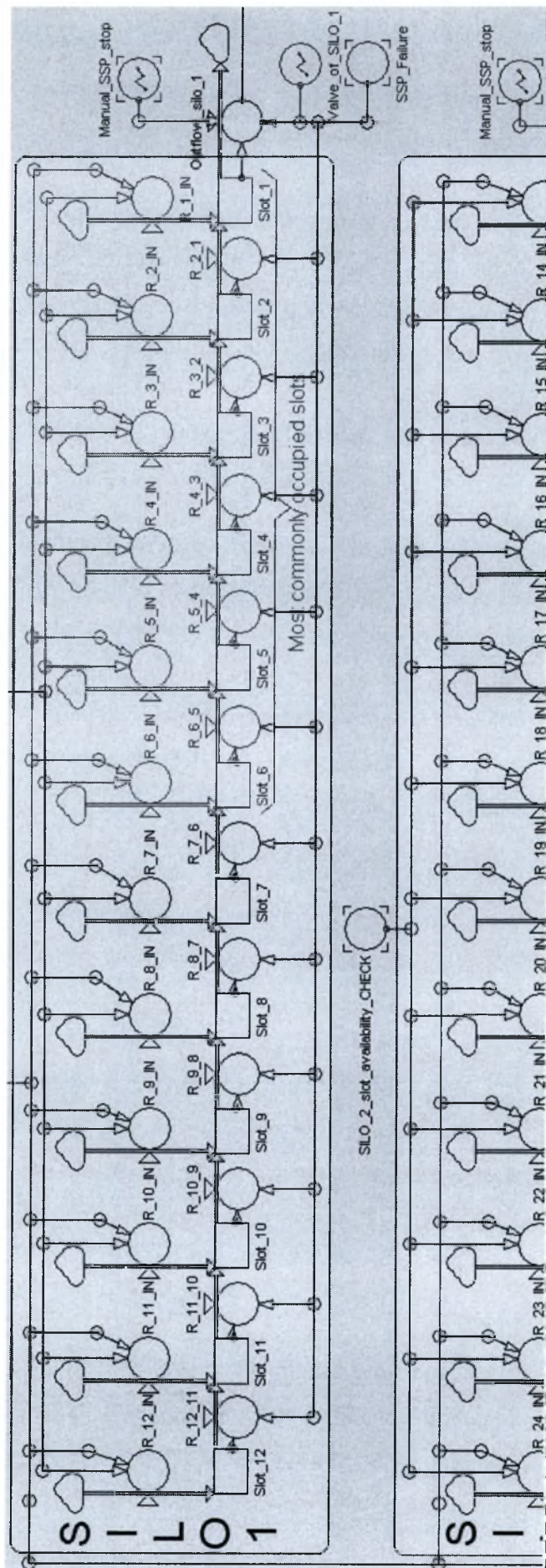
6.4.5 Στάδιο αποθήκευσης ημιετοιμών προϊόντων TSS

Στο Σχήμα 12 παρουσιάζεται η λειτουργία του σιλό 1 του σταδίου ενδιάμεσης αποθήκευσης TSS η οποία είναι αντίστοιχη με τις λειτουργίες των σιλό 2 και 3 τα οποία δεν παρουσιάζονται για λόγους συντομίας. Η χωρητικότητα του κάθε σιλό χωρίζεται σε δώδεκα ονομαστικές διακριτές θέσεις αποθήκευσης των 33,33 τόνων η κάθε μία διότι η ποσότητα αυτή αντιστοιχεί στους ρυθμούς εισόδου και εξόδου ανά τετράωρο ενός σιλό που χρησιμοποιείται: ο ρυθμός εισόδου αντιστοιχεί στην παραγωγή ημιετοιμου προϊόντος ανά τετράωρο του παραγωγικού σταδίου POLY ενώ ο ρυθμός εξόδου αντιστοιχεί στο ρυθμό απορρόφησης ημιετοιμών προϊόντων ανά τετράωρο από το παραγωγικό στάδιο SSP προς παραγωγή τελικών προϊόντων. Επομένως στο σιλό 1 ανήκουν τα slot 1 έως 12, στο σιλό 2 ανήκουν τα slot 13 έως 24 και στο σιλό 3 ανήκουν τα slot 25 έως 36.

Ο ρυθμός εισόδου ενός slot (π.χ. slot 5) προκύπτει είτε από τη μετακίνηση αντίστοιχης ποσότητας από το προηγούμενό του slot όταν αυτό περιέχει ημιέτοιμο προϊόν (π.χ. R_6_5) και ταυτόχρονα υπάρχει εκροή από το συγκεκριμένο σιλό (π.χ. Valve_of_SILO_1=1), είτε από την παραγωγή ημιετοιμών προϊόντων στο στάδιο POLY όταν υπάρχει ταυτοχρόνως εισροή προϊόντος στο σιλό, το επόμενο από αυτό slot είναι κατειλημμένο από προϊόν και τα προηγούμενά του slot είναι άδεια (π.χ. R_5_IN). Ο ρυθμός εξόδου ενός slot συμβολίζει την καθοδική κίνηση του προϊόντος μέσα σε ένα σιλό που χρησιμοποιείται (π.χ. R_5_4) ή την έξοδό του από το σιλό για την τροφοδοσία του σταδίου SSP (π.χ. Outflow_silo_1).

Η εκροή ενός σιλό (π.χ. Outflow_silo_1) εξαρτάται από τη βάνα εκροής (π.χ. Valve_of_SILO_1) και την κατάσταση λειτουργίας του παραγωγικού σταδίου SSP (SSP_Failure, Manual_SSP_stop). Ειδικότερα, η Διοίκηση μπορεί για οποιονδήποτε λόγο να αλλάξει το σιλό τροφοδοσίας του SSP (π.χ. σιλό 1) με κάποιο άλλο σιλό που περιέχει θετική ποσότητα ημιετοιμου προϊόντος (π.χ. χρήση των σιλό 2 ή 3). Στην περίπτωση αυτή κλείνει η βάνα εκροής του σιλό που χρησιμοποιούταν (π.χ. Valve_of_SILO_1=0) και ανοίγει η βάνα κάποιου άλλου. Επιπρόσθετα, όταν το παραγωγικό στάδιο SSP έχει κάποια βλάβη (SSP_Failure=0) τότε κλείνουν οι βάνες εκροής όλων των σιλό καθώς παύει η τροφοδοσία του SSP από το TSS μέχρι την αποκατάσταση της βλάβης (Valve_of_SILO_1/2/3=0). Το ίδιο συμβαίνει και στην περίπτωση διενέργειας προγραμματισμένης συντήρησης του εξοπλισμού του SSP καθώς η τροφοδοσία του από το TSS διακόπτεται (Manual_SSP_stop=0).

Τέλος, από τους τρεις ρυθμούς εκροής των σιλό μόνο ένας μπορεί να είναι μη μηδενικός. Αυτό συμβαίνει διότι η ποσότητα ημιετοιμου προϊόντος που επεξεργάζεται το στάδιο SSP κατά τη διάρκεια ενός τετραώρου είναι ίση με την ποσότητα εκροής ανά τετράωρο ενός σιλό και ίση με την περιεχόμενη ποσότητα ενός slot. Επομένως, όταν υπάρχει εκροή από το σιλό 1 το οποίο συνήθως χρησιμοποιείται τότε οι δύο ρυθμοί εκροής των σιλό 2 και 3 είναι υποχρεωτικά μηδενικοί.



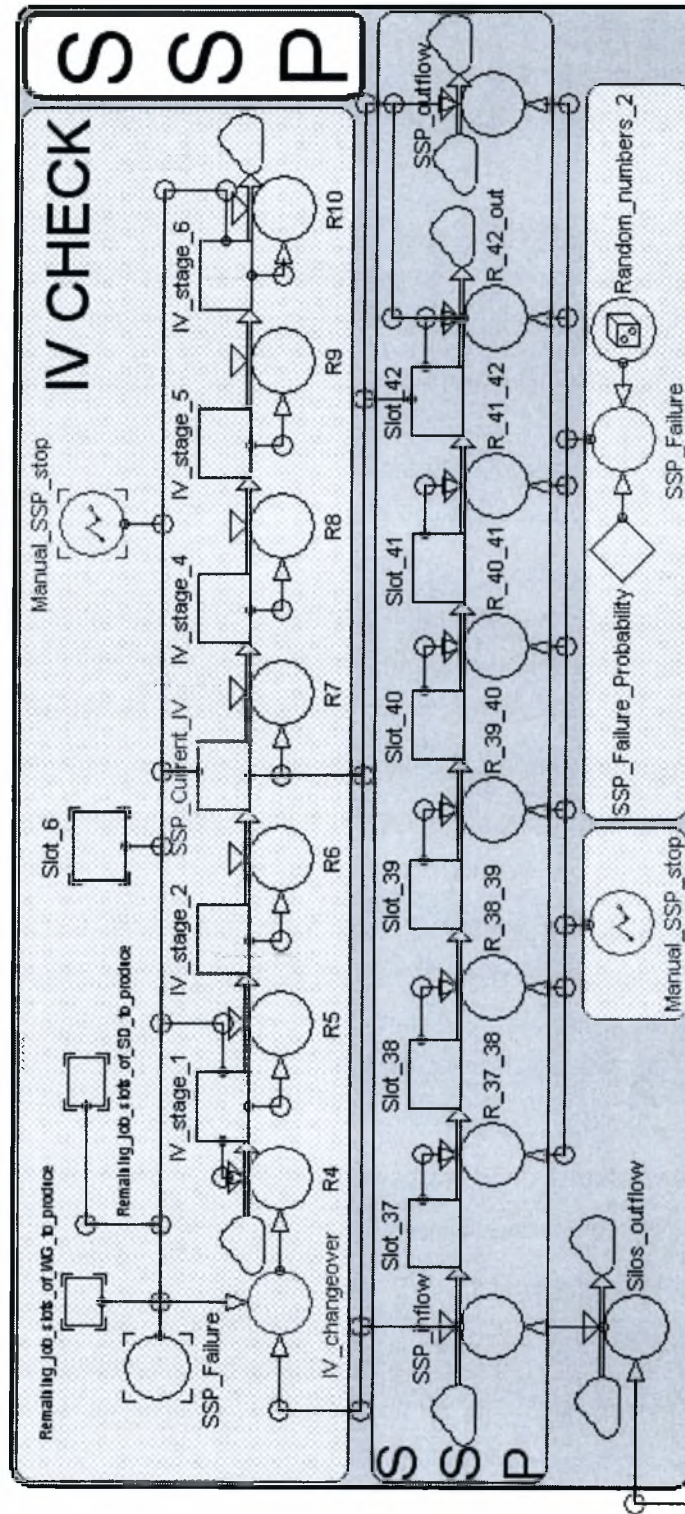
Σχήμα 12. Στάδιο αποθήκευσης ημιέτοιμων προϊόντων TSS

6.4.6 Παραγωγικό στάδιο απόδοσης ιξώδους SSP

Στο παραγωγικό στάδιο SSP αποδίδεται στο εισερχόμενο ημιέτοιμο προϊόν η τιμή του ιξώδους οπότε και προκύπτει το είδος του τελικού παραγόμενου προϊόντος από το συνδυασμό χρώματος και ιξώδους. Η εισροή του SSP (SSP_inflow) προκύπτει από το άθροισμα των ρυθμών εκροής του TSS από τους οποίους το πολύ ένας είναι μη μηδενικός (Silos_outflow). Υπό την προϋπόθεση ότι το SSP δεν έχει βλάβη (SSP_Failure=1) και ότι δε γίνονται εργασίες συντήρησης (Manual_SSP_stor=1), το ημιέτοιμο προϊόν προωθείται μέσα στα έξι slot του SSP κατά ένα slot ανά τετράωρο ενώ η τιμή του ιξώδους αποδίδεται στο προϊόν κατά την έξοδό του από το SSP (SSP_outflow) με βάση την τρέχουσα τιμή του ιξώδους (SSP_Current_IV) οπότε και προκύπτει το είδος του τελικού προϊόντος.

Η τιμή του ιξώδους προκύπτει από μία ενδογενή διαδικασία παρακολούθησης των εξής παραμέτρων: (1) της ολοκλήρωσης τυχόν προηγούμενης διαδικασίας αλλαγής ιξώδους (IV_CHECK), (2) του χρώματος του ημιέτοιμου προϊόντος που διέρχεται του SSP (SSP_inflow, Slot_42) καθώς και αυτού που έπεται (Slot_6), και (3) των παραγγελιών τελικών προϊόντων που εκκρεμούν (Remaining_job_slots_of_WG/SD/G/FH_to_produce). Ειδικότερα, η παρακολούθηση του εάν εκκρεμεί η ολοκλήρωση προηγούμενης εντολής αλλαγής ιξώδους γίνεται μέσω της παρακολούθησης της ομοιότητας του ιξώδους μεταξύ της πρώτης (IV_stage_1) και της τελευταίας περιόδου (IV_stage_6) από τις συνολικά έξι που απαιτούνται για την ολοκλήρωση μίας αλλαγής ιξώδους. Σε περίπτωση που εκκρεμεί η ολοκλήρωση μίας προηγούμενης αλλαγής τότε δε δίνεται νέα εντολή αλλαγής ιξώδους μέχρι να ολοκληρωθεί η προηγούμενη. Επιπλέον, σε περίπτωση που το ημιέτοιμο προϊόν είναι χρώματος 2 (≡Gray) ή 3 (≡Dark) τότε το ιξώδες θα πρέπει υποχρεωτικά να είναι ίσο με 2 (≡υψηλό). Αντιθέτως, εάν το ημιέτοιμο προϊόν είναι χρώματος 1 (≡Light) τότε η τιμή του ιξώδους αποφασίζεται βάσει των παραγγελιών που εκκρεμούν για τα προϊόντα WG και SD με ιξώδες 1 (≡χαμηλό) και 2 αντίστοιχα (Remaining_job_slots_of_WG/SD_to_produce), αλλά και βάσει του χρώματος των ημιέτοιμων προϊόντων που έπονται (Slot_6 του TSS) ώστε να μπορεί το ιξώδες να μεταβεί (αν είναι επιθυμητό) από 2 σε 1 και να προλάβει να επιστρέψει σε 2 αν αλλάξει το χρώμα από 1 σε 2.

Σημειώνεται ότι η τρέχουσα τιμή του ιξώδους (SSP_Current_IV) προκύπτει έπειτα από τρία τετράωρα μετάβασης (IV_stage_1/2/3) από τη στιγμή ενεργοποίησης μίας εντολής αλλαγής ιξώδους ενώ η διαδικασία συνεχίζεται και ολοκληρώνεται στα επόμενα τρία τετράωρα (IV_stage_4/5/6). Η παραδοχή αυτή είναι σύμφωνη τόσο με την πρακτική που ακολουθεί η χημική βιομηχανία όσο και με τις παραδοχές του γραμμικού μοντέλου. Τέλος σημειώνεται ότι σε περίπτωση βλάβης (SSP_Failure=0) ή προγραμματισμένης συντήρησης (Manual_SSP_stor=0) δε μπορεί να ενεργοποιηθεί ή να ολοκληρωθεί μία αλλαγή ιξώδους (αν έχει ξεκινήσει να ριζώνει) καθώς λόγω τεχνολογικών περιορισμών απαιτείται η λειτουργία του εξοπλισμού.



Σχήμα 13. Παραγωγικό στάδιο SSP

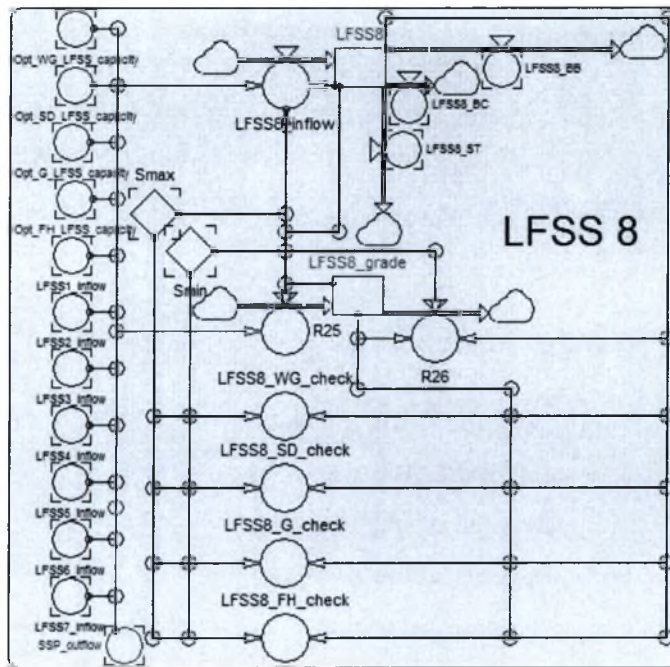
6.4.7 Υπολογισμός βέλτιστου σιλό αποθήκευσης στο LFSS

Στο Σχήμα 14 παρουσιάζεται για λόγους συντομίας η λειτουργία μόνο του σιλό 8 (LFSS 8) του σταδίου αποθήκευσης τελικών προϊόντων LFSS καθώς η λειτουργία των υπολοίπων 7 σιλό είναι αντίστοιχη με αυτήν του σιλό 8. Ειδικότερα, το περιεχόμενο του σιλό ορίζεται τόσο από την ποσότητά του (LFSS8) όσο και από το είδος του (LFSS_grade). Το είδος του περιεχόμενου προϊόντος σημειώνεται μέσω ενός ρυθμού καταγραφής (R25) κατά την είσοδο της πρώτης ποσότητας σε ένα σιλό που είτε είναι άδειο ($LFSS8=0$) είτε είναι σχεδόν άδειο ($LFSS8 < S_{min}$) και παραμένει αμετάβλητο μέχρι είτε αυτό να αδειάσει πλήρως είτε η εναπομείνασα ποσότητα προϊόντος να είναι πολύ μικρή ($LFSS8 < S_{min}$) οπότε και αποχαρακτηρίζεται μέσω ενός ρυθμού (R26). Το τελευταίο είναι σύμφωνο με τις παραδοχές τόσο της βιομηχανίας όσο και του γραμμικού μοντέλου προκειμένου να μη δεσμεύεται ένα σιλό από μία πολύ μικρή ποσότητα προϊόντος. Από τις ποσότητες που περιέχονται στα σιλό ικανοποιείται η ζήτηση προϊόντων σε κοντέινερ (LFSS_BC) και σιλοφόρα φορτηγά (LFSS_ST) καθώς και το σάκιασμα προϊόντων (LFSS_BB) προς αναπλήρωση των αποθεμάτων της αποθήκης, όπως περιγράφεται σε επόμενη παράγραφο.

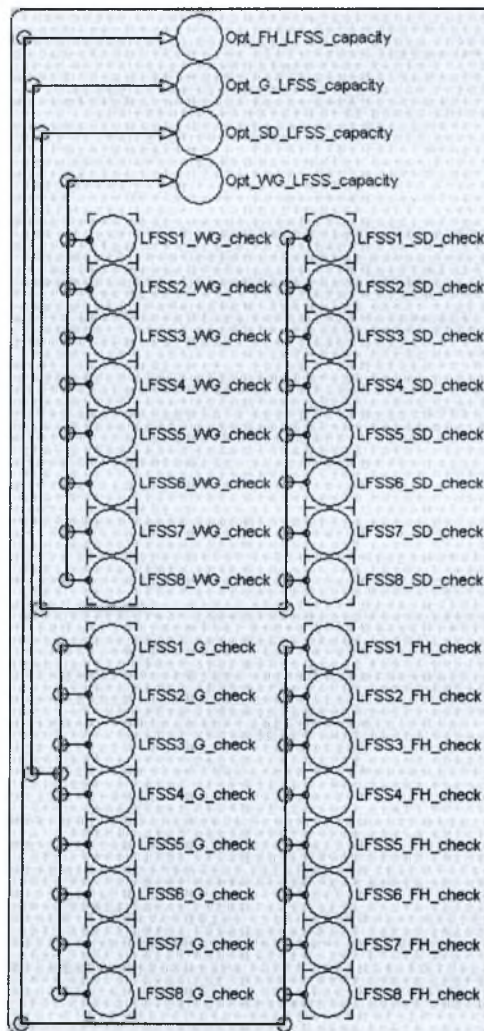
Προκειμένου να εντοπιστεί το βέλτιστο σιλό τοποθέτησης του προϊόντος που εξέρχεται του SSP (SSP_outflow) ακολουθείται μία διαδικασία δύο βημάτων:

(1) Αρχικά, υπολογίζεται η βέλτιστη διαθεσιμότητα των οκτώ σιλό για την αποθήκευση κάθε είδους προϊόντος (π.χ. LFSS8_WG/SD/G/FH_check). Ειδικότερα ελέγχεται πρώτον, αν το είδος προϊόντος που παράγεται είναι ίδιο με το περιεχόμενο ενός σιλό (εάν υπάρχει περιεχόμενο προϊόν) και δεύτερον, αν χωράει η παραγόμενη ποσότητα στο σιλό αυτό ($S_{max}-LFSS8 \geq 33,33$ τόνων). Έχοντας υπολογίσει τις διαθεσιμότητες όλων των σιλό, υπολογίζεται στη συνέχεια η βέλτιστη διαθεσιμότητα ανά είδος προϊόντος (Σχήμα 15) ως η ελάχιστη μεταξύ των διαθεσιμοτήτων κάθε σιλό η οποία όμως να είναι μεγαλύτερη από ή ίση με 33,33 τόνους (π.χ. Opt_WG_LFSS_capacity).

(2) Στη συνέχεια εντοπίζεται το βέλτιστο σιλό τοποθέτησης ως εκείνο που παρουσιάζει τη βέλτιστη διαθεσιμότητα. Ειδικότερα ελέγχεται σειριακά από το σιλό 1 προς το σιλό 8 πρώτον, αν οι εισροές των προηγούμενων σιλό ήταν μηδενικές (π.χ. $LFSS1_inflow=0, \dots, LFSS7_inflow=0$) και δεύτερον, ελέγχεται αν η βέλτιστη διαθεσιμότητα που υπολογίστηκε είναι ίση με αυτή του εξεταζόμενου σιλό. Στην περίπτωση όπου και τα δύο προαπαιτούμενα είναι αληθή τότε το βέλτιστο σιλό είναι το εξεταζόμενο ($LFSS8_inflow=33,33$ τόνους) και όλες οι εισροές των υπολοίπων σιλό είναι μηδενικές. Η διαδικασία αυτή βασίζεται στο γεγονός ότι μόνο μία εισροή μπορεί να είναι μη μηδενική ενώ η σειριακή εξέταση από το σιλό 1 προς το 8 σε συνδυασμό με τη σειριακή ικανοποίηση της ζήτησης από το σιλό 8 προς το 1 (όπως περιγράφεται στις παραγράφους που ακολουθούν) αποδίδει περισσότερα γεμάτα και περισσότερα άδεια σιλό αποφεύγοντας τα μισογεμάτα σιλό τα οποία μειώνουν την ευελιξία της αποθήκευσης τελικών προϊόντων.



Σχήμα 14. Σιλό 8 του σταδίου αποθήκευσης τελικών προϊόντων LFSS

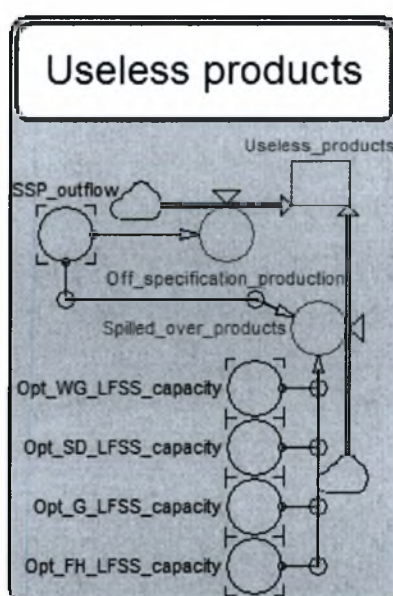


Σχήμα 15. Υπολογισμός της βέλτιστης διαθεσιμότητας των σιλό του LFSS

6.4.8 Υπολογισμός προϊόντων που απορρίπτονται

Στο Σχήμα 16 παρουσιάζεται ο υπολογισμός των προϊόντων που απορρίπτονται (Useless_products) είτε διότι ο συνδυασμός χρώματος και ιξώδους που έχουν είναι εκτός των απαιτούμενων προδιαγραφών (Off_specification_production) είτε λόγω έλλειψης διαθεσιμότητας στα οκτώ σιλό αποθήκευσης του LFSS (Spilled_over_products). Ειδικότερα, όταν μία ποσότητα παραγωγής ενός τετραώρου ίση με 33,33 τόνους δεν προέρχεται από κάποιον αποδεκτό συνδυασμό χρώματος και ιξώδους από αυτούς που έχουν περιγραφεί τότε η ποσότητα αυτή δε μπορεί να πουληθεί και πετιέται. Το ίδιο συμβαίνει ακόμη και με προϊόντα τα οποία είναι αποδεκτά ως προς τις προδιαγραφές τους αλλά δεν υπάρχει διαθέσιμος χώρος σε κανένα από τα οκτώ σιλό του LFSS για να αποθηκευτούν. Ο έλεγχος για τη διαθεσιμότητα των οκτώ σιλό του LFSS γίνεται μέσω της μεταβλητής Opt_WG/SD/G/FH_LFSS_capacity που παρουσιάστηκε στην προηγούμενη παράγραφο κατά την οποία όταν η διαθεσιμότητα των σιλό που περιέχουν ίδιο προϊόν με αυτό που εξέρχεται από το στάδιο SSP είναι μικρότερη από 33,33 τόνους και ταυτόχρονα δεν υπάρχει κάποιο άδειο σιλό τότε το παραγόμενο προϊόν πρέπει να πεταχτεί.

Μέσω της παρακολούθησης του ρυθμού απόρριψης μη τυποποιημένων προϊόντων μπορεί να ελεγχθεί η ορθότητα του μοντέλου (validation) διότι αν ο ρυθμός αυτός είναι θετικός τότε οι ρυθμίσεις του εξοπλισμού μεταξύ των δύο παραγωγικών σταδίων δεν είναι συγχρονισμένες. Επιπρόσθετα, ελέγχεται και η αποδοτικότητα του μοντέλου διότι αν απορρίπτονται προϊόντα που πληρούν τις προδιαγραφές τότε είτε δε γίνεται ορθολογική διαχείριση των σιλό του LFSS είτε έχουν εισαχθεί μη αποδοτικές πολιτικές διαχείρισης των αποθεμάτων οι οποίες οδηγούν στην παραγγελία υπερβολικών ποσοτήτων για κάποια προϊόντα ενώ κάποια άλλα ίσως να αντιμετωπίζουν χαμένες πωλήσεις λόγω ανεπάρκειας των αποθεμάτων τους.

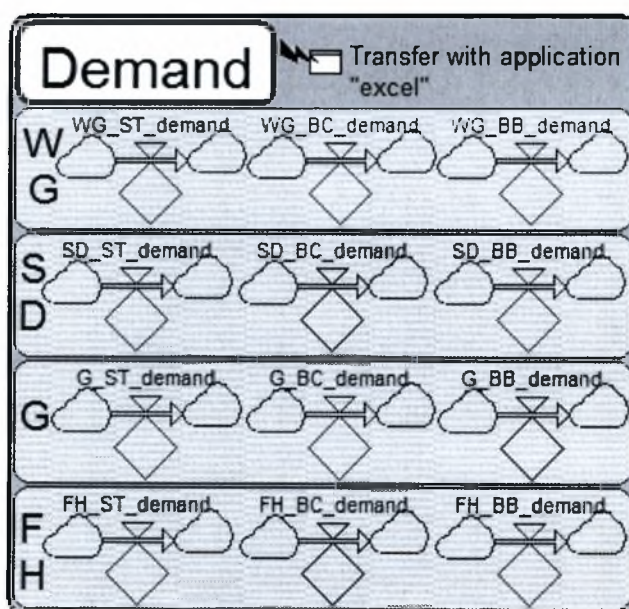


Σχήμα 16. Υπολογισμός ποσοτήτων των προϊόντων που απορρίπτονται

6.4.9 Ζήτηση τελικών προϊόντων

Στο Σχήμα 17 παρουσιάζεται η μακροεντολή (Transfer with application “excel”) μέσω της οποίας αντλούνται από φύλλο του Excel δεδομένα τα οποία στη συνέχεια αποδίδονται στις τιμές των ζητήσεων των προϊόντων μέσω των αντίστοιχων μεταβλητών ρυθμού (τόνου/τετράωρο) για κάθε είδος προϊόντος και για κάθε μορφή ζήτησης. Σημειώνεται ότι για τη ζήτηση του μοντέλου ΔΣ χρησιμοποιούνται τα ίδια δεδομένα με εκείνα του γραμμικού μοντέλου προκειμένου τα αποτελέσματα να είναι μεταξύ τους απολύτως συγκρίσιμα, με μία όμως σημαντική διαφορά: το μοντέλο ΔΣ δε γνωρίζει τη ζήτηση πριν από την εκδήλωσή της και προσομοιώνει τη λειτουργία του συστήματος βάσει των ενδογενών πολιτικών διαχείρισης αποθεμάτων. Ως αποτέλεσμα το μοντέλο ΔΣ να μπορεί να λειτουργεί σε περιβάλλον αβεβαιότητας και να αποδίδει μία λύση οποιαδήποτε και αν είναι η ζήτηση, η ποιότητα της οποίας εξαρτάται κυρίως από δύο παράγοντες: την ικανοποιητική προσέγγιση του πραγματικού συστήματος από το μοντέλο και την ποιότητα των ενδογενών διαδικασιών λήψεως αποφάσεων. Συμπερασματικά, όσο πιο περιεκτικό είναι το μοντέλο σε καίρια σημεία της παραγωγικής διαδικασίας και όσο πιο αποδοτικές είναι οι ενδογενείς διαδικασίες λήψεως αποφάσεων, τόσο ποιοτικά ανώτερη είναι η αποδιδόμενη λύση του μοντέλου.

Το ισχυρό πλεονέκτημα που διαθέτουν οι μέθοδοι προσομοίωσης έναντι των ποιοτικά ανώτερων αναλυτικών μεθόδων επίλυσης είναι η εξαιρετικά υψηλή ταχύτητα προσομοίωσης και η δυνατότητα λειτουργίας σε περιβάλλον αβεβαιότητας. Για το λόγο αυτό, με την εκπόνηση της παρούσας μεταδιπλωματικής εργασίας γίνεται μία προσπάθεια παρουσίασης των πλεονεκτημάτων τόσο του Γραμμικού Προγραμματισμού όσο και της Δυναμικής Συστημάτων ώστε οι δύο μέθοδοι να λειτουργούν συμπληρωματικά έναντι των σύνθετων πολλαπλών απαιτήσεων των πραγματικών συστημάτων και όχι ανταγωνιστικά όπως αποτυπώνεται στη βιβλιογραφία.

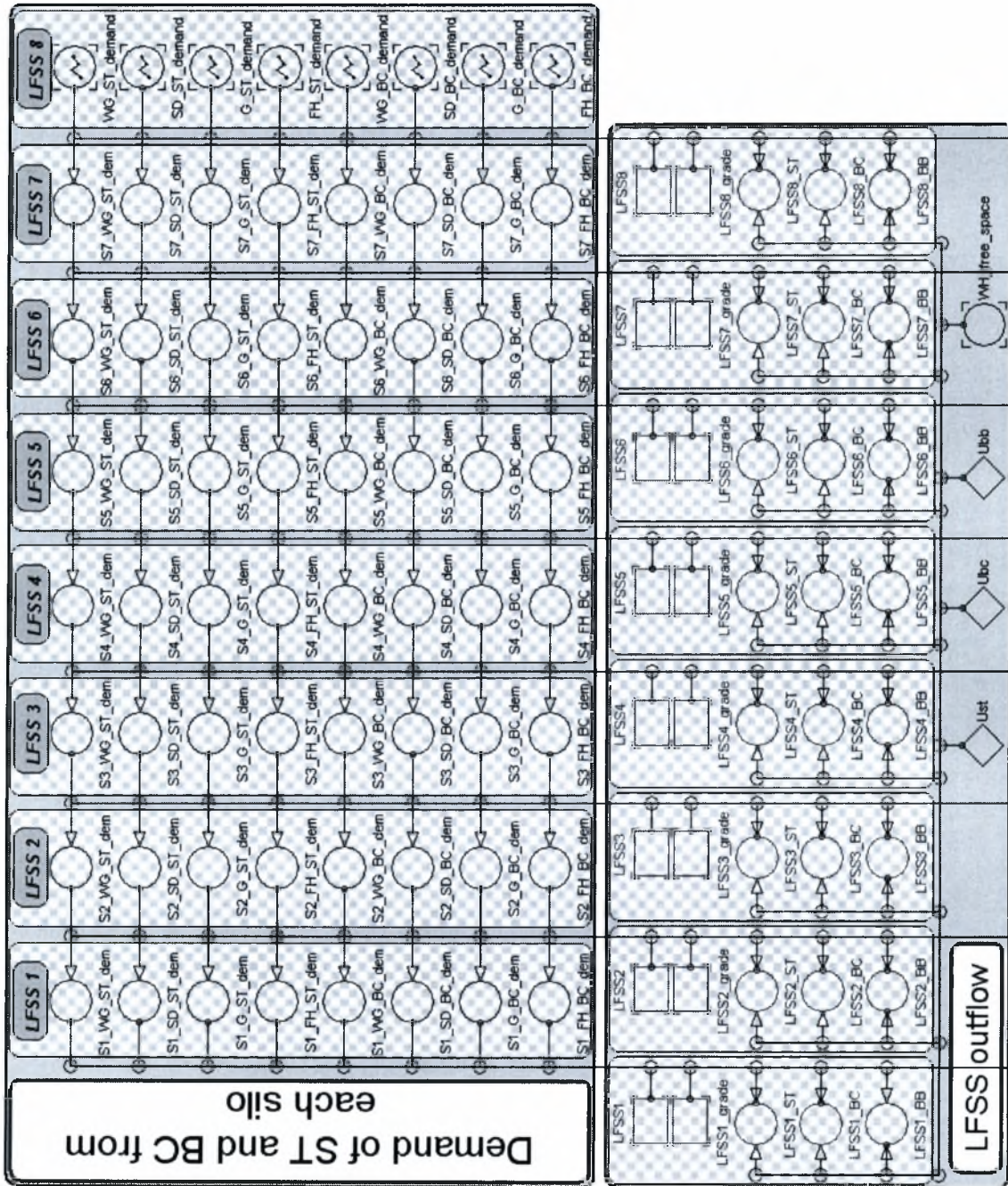


Σχήμα 17. Ζήτηση ανά είδος προϊόντος και μορφή μεταφοράς

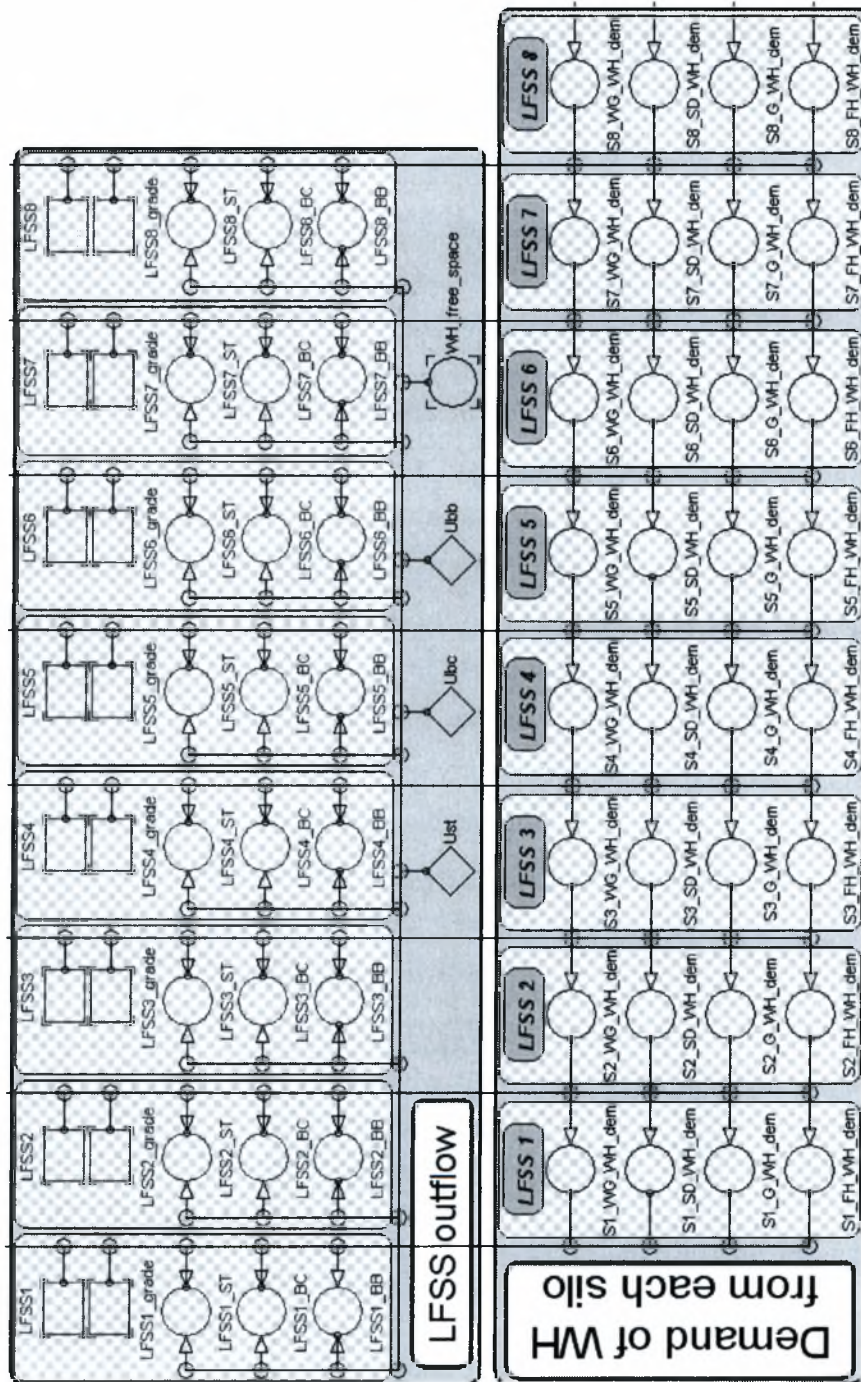
6.4.10 Εκροή των σιλό του LFSS

Στο Σχήμα 18 και στο Σχήμα 19 παρουσιάζεται ο υπολογισμός της εκροής του κάθε σιλό του LFSS προς ικανοποίηση της ζήτησης των διαφορετικών ειδών προϊόντων σε μορφή σιλοφόρων φορτηγών και κοντέινερ, καθώς και προς αναπλήρωση των αποθεμάτων της αποθήκης μέσω μίας μηχανής σακιάσματος. Σύμφωνα με τη λογική διαχείρισης των αποθεμάτων που περιγράφηκε ώστε να προκύπτουν περισσότερα άδεια και γεμάτα σιλό, ελέγχεται αρχικά αν η ζήτηση μπορεί να ικανοποιηθεί από το σιλό 8 και έπειτα σειριακά από τα σιλό 7, 6, 5, 4, 3, 2 και 1. Ειδικότερα ζητείται από το σιλό 8 (LFSS 8) η ικανοποίηση όλων των μορφών ζήτησεων για όλα τα είδη προϊόντος, καθώς και η αναπλήρωση των αποθεμάτων της αποθήκης. Για το σιλό 8 ελέγχεται πρώτον, το είδος του περιεχόμενου προϊόντος (LFSS8_grade) και δεύτερον η ποσότητά του (LFSS8). Εάν το είδος του προϊόντος που ζητείται είναι ίδιο με αυτό του περιεχόμενου προϊόντος στο σιλό 8 τότε εκρέει από αυτό ποσότητα (LFSS8_ST/BC/BB) ώστε να καλύψει τη συγκεκριμένη μορφή της ζήτησης. Σημειώνεται ότι η ικανοποίηση της ζήτησης σε μορφή σιλοφόρων φορτηγών έχει προτεραιότητα 1, σε μορφή κοντέινερ έχει προτεραιότητα 2 ενώ η αναπλήρωση των αποθεμάτων της αποθήκης σε μορφή σάκων έχει προτεραιότητα 3 εφόσον υπάρχει διαθέσιμος χώρος στην αποθήκη για τους σάκους ($WH_free_space > 0$). Οι προτεραιότητες τίθενται με αυτό τον τρόπο διότι ο ρυθμός εκροής προϊόντων προς τα σιλοφόρα φορτηγά είναι μεγαλύτερος από τον αντίστοιχο του προς τα κοντέινερ ($Ust > Ubc$) και αυτός με τη σειρά του είναι μεγαλύτερος από το ρυθμό σακιάσματος προϊόντων για την αποθήκη ($Ubc > Ubb$). Επομένως μέσω αυτών των προτεραιοτήτων ικανοποιείται όσο το δυνατό μεγαλύτερη ποσότητα της συνολικής ζήτησης.

Εάν η ποσότητα του σιλό επαρκεί για την ικανοποίηση της ζήτησης σε σιλοφόρα φορτηγά τότε αφού ικανοποιηθεί αυτή, ελέγχεται αν υπάρχει χρόνος κατά την υπόλοιπη διάρκεια του τετραώρου ώστε να ικανοποιηθεί η ενδεχόμενη ζήτηση του ίδιου προϊόντος στις υπόλοιπες μορφές της (με σειρά προτεραιότητας, για κοντέινερ και σάκιασμα). Όση ποσότητα από τη συνολική ζήτηση για οποιοδήποτε είδος προϊόντος και οποιαδήποτε μορφή δε μπορεί να ικανοποιηθεί από το σιλό 8 μεταφέρεται ως ζήτηση προς ικανοποίηση στο σιλό 7 (LFSS 7). Το σιλό 7 λειτουργεί με ανάλογο τρόπο και μεταφέρει την υπολειπόμενη ζήτηση που δε μπορεί ούτε αυτό να ικανοποιήσει προς το σιλό 6 (LFSS 6). Η διαδικασία μεταφοράς της υπολειπόμενης ζήτησης του κάθε σιλό προς το προηγούμενό του συνεχίζεται μέχρι και το σιλό 1 (LFSS 1) όπου όση ζήτηση δε μπορεί να ικανοποιηθεί χάνεται καθώς είτε τα αποθέματα του LFSS δεν επαρκούν για την ικανοποίησή της είτε οι ρυθμοί εκροής οι οποίοι περιορίζονται από τις μέγιστες τιμές τους (Ust, Ubc) δεν επαρκούν για να την ικανοποιήσουν εγκαίρως. Αντιθέτως, όση ποσότητα των αποθεμάτων της αποθήκης δε μπορεί να αναπληρωθεί μεταφέρεται προς ικανοποίηση στο επόμενο τετράωρο μέσω του επανυπολογισμού της διαφοράς μεταξύ των νέων υφιστάμενων αποθεμάτων της αποθήκης και των επιθυμητών τους τιμών.



Σχήμα 18. Εκροή του LFSS προς ικανοποίηση της ζήτησης τύπου ST και BC

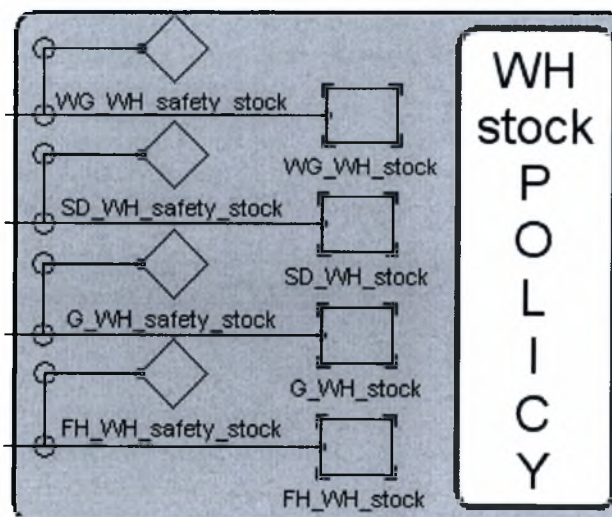


Σχήμα 19. Εκροή του LFSS προς αναπήρωση των αποθεμάτων της αποθήκης

6.4.11 Διαχείριση αποθεμάτων αποθήκης

Στο Σχήμα 20 παρουσιάζεται η πολιτική διαχείρισης των αποθεμάτων της αποθήκης τα οποία είναι σε σακιασμένη μορφή. Ειδικότερα, σύμφωνα με την πολιτική αυτή ελέγχεται αρχικά αν το υφιστάμενο απόθεμα στην αποθήκη ενός είδους προϊόντος (π.χ. WG_WH_stock) είναι πάνω ή κάτω από το επιθυμητό επίπεδο αποθεμάτων του (π.χ. WG_WH_safety_stock). Όταν το υφιστάμενο απόθεμα είναι ίσο ή πάνω από την επιθυμητή τιμή του τότε δε γίνεται καμία ενέργεια ενώ σε αντίθετη περίπτωση ζητείται από τα σιλό του LFSS και τη σακιστική μηχανή το σάκιασμα τόσης ποσότητας ώστε το απόθεμα του κάθε είδους προϊόντος σε σάκους να φτάσει στην επιθυμητή του τιμή. Η διαδικασία ελέγχου του σακιάσματος προϊόντων από τη σακιστική μηχανή παρουσιάστηκε στην προηγούμενη παράγραφο.

Ο λόγος για τον οποίο δε χρησιμοποιείται κάποια πολιτική τύπου «Ss» στη διαχείριση των αποθεμάτων της αποθήκης όπως συμβαίνει με τη διαχείριση των αποθεμάτων του LFSS είναι απλός: το σάκιασμα των προϊόντων δεν προαπαιτεί την παραγωγή προϊόντων αλλά μόνο τη ύπαρξη και διαθεσιμότητά τους για σάκιασμα στα σιλό του LFSS (υπενθυμίζεται ότι το σάκιασμα έχει προτεραιότητα 3 ως προς τις ζητήσεις σε μορφή σιλοφόρων φορτηγών και κοντέινερ). Επομένως αν υπάρχουν διαθέσιμα ικανά αποθέματα στα σιλό του LFSS τότε η αναπλήρωση των αποθεμάτων της αποθήκης γίνεται άμεσα (μέσω του ρυθμού σακιάσματος) χωρίς να προαπαιτούνται διαδικασίες ανάλογες της παραγωγής. Ωστόσο όπως σημειώθηκε στην παράγραφο «Διαχείριση αποθεμάτων του LFSS και επιλογή χρώματος του POLY», όταν τίθεται μία παραγγελία στο τμήμα παραγωγής της βιομηχανίας προς αναπλήρωση των αποθεμάτων του LFSS τότε το μέγεθος της παραγγελίας προσαυξάνεται κατά τις ανάγκες αναπλήρωσης των αποθεμάτων της αποθήκης ώστε να υπολογίζονται και να αναπληρώνονται συνολικά οι ελλείψεις προϊόντων όλων των αποθεμάτων.



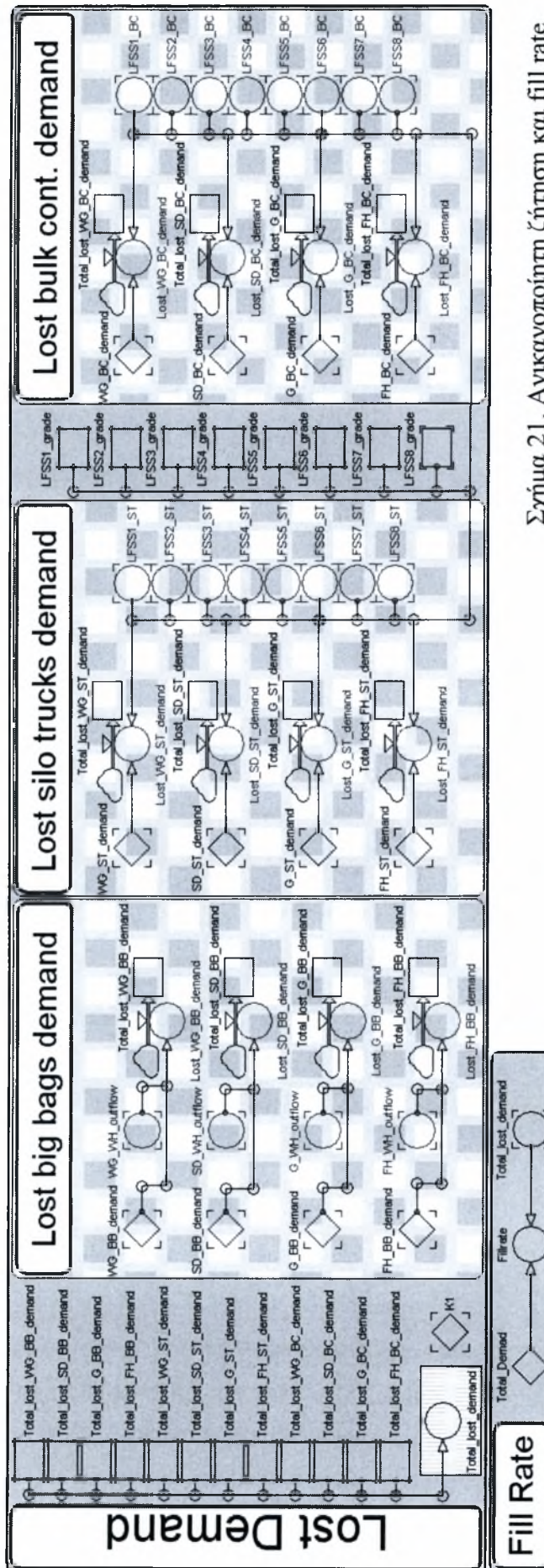
Σχήμα 20. Διαχείριση των αποθεμάτων της αποθήκης

6.4.12 Υπολογισμός της ανικανοποίητης ζήτησης και του fill rate

Σε αυτό το πεδίο υπολογίζεται η ανικανοποίητη ζήτηση ανά είδος προϊόντος και μορφή για κάθε τετράωρο καθώς και το ποσοστό της συνολικής ζήτησης που ικανοποιείται άμεσα (fill rate). Ξεκινώντας την περιγραφή του Σχήματος 21 από αριστερά προς τα δεξιά, υπολογίζονται οι χαμένες πωλήσεις προϊόντων σε μορφή σάκων, σιλοφόρων φορτηγών και κοντέινερ. Ειδικότερα για τα προϊόντα σε μορφή σάκων, συγκρίνεται σε κάθε τετράωρο η ζήτηση κάθε είδους προϊόντος (π.χ. WG_BB_demand) με την αντίστοιχη εκροή της αποθήκης (π.χ. WG_WH_outflow) οπότε η ύπαρξη θετικής διαφοράς μεταξύ αυτών των δύο είναι ίση με τη χαμένη ζήτηση του συγκεκριμένου προϊόντος για το συγκεκριμένο τετράωρο (π.χ. Lost_WG_BB_demand). Οι χαμένες ζητήσεις των διαφορετικών ειδών προϊόντων ανά τετράωρο σε μορφή σάκων αθροίζονται σε ξεχωριστά επίπεδα τα οποία υπολογίζουν τη συνολική χαμένη ζήτηση για κάθε είδος προϊόντος σε μορφή σάκων ξεχωριστά (π.χ. Total_lost_WG_BB_demand).

Η χαμένη ζήτηση προϊόντων σε μορφή σιλοφόρων φορτηγών υπολογίζεται με τρόπο ανάλογο με αυτόν που αναφέρθηκε για τον υπολογισμό της χαμένης ζήτησης σε μορφή σάκων. Συγκεκριμένα, ελέγχεται το είδος του προϊόντος που περιέχεται σε κάθε ένα από τα οκτώ σιλό του LFSS (π.χ. LFSS1_grade) καθώς και η εκροή του καθενός από αυτά σε μορφή σιλοφόρων φορτηγών (π.χ. LFSS1_ST). Γνωρίζοντας το περιεχόμενο είδος προϊόντος του κάθε σιλό και την εκροή του καθενός από αυτά προς σιλοφόρα φορτηγά, υπολογίζεται η συνολική εκροή όλων των σιλό για την ικανοποίηση της ζήτησης σε μορφή σιλοφόρων φορτηγών ανά είδος προϊόντος. Η θετική διαφορά μεταξύ της ζήτησης ενός είδους προϊόντος σε μορφή σιλοφόρων φορτηγών (π.χ. WG_ST_demand) και της συνολικής αντίστοιχης εκροής των σιλό είναι η χαμένη ζήτηση του συγκεκριμένου είδους προϊόντος σε μορφή σιλοφόρων φορτηγών για το συγκεκριμένο τετράωρο (π.χ. Lost_WG_ST_demand). Οι χαμένες ζητήσεις των διαφορετικών ειδών προϊόντων ανά τετράωρο σε μορφή σιλοφόρων φορτηγών αθροίζονται σε ξεχωριστά επίπεδα τα οποία υπολογίζουν τη συνολική χαμένη ζήτηση για κάθε είδος προϊόντος σε μορφή σιλοφόρων φορτηγών ξεχωριστά (π.χ. Total_lost_WG_ST_demand).

Με τρόπο απολύτως ανάλογο με αυτόν που αναφέρθηκε για τον υπολογισμό της χαμένης ζήτησης προϊόντων σε μορφή σιλοφόρων φορτηγών υπολογίζεται και η χαμένη ζήτηση σε μορφή κοντέινερ ανά είδος προϊόντος και τετράωρο (π.χ. Lost_WG_BC_demand) από τη σύγκριση μεταξύ της ζήτησης ενός είδους προϊόντος σε μορφή κοντέινερ (π.χ. WG_BC_demand) και της αντίστοιχης εκροής των σιλό του LFSS. Οι χαμένες ζητήσεις των διαφορετικών ειδών προϊόντων ανά τετράωρο σε μορφή κοντέινερ αθροίζονται σε ξεχωριστά επίπεδα τα οποία υπολογίζουν τη συνολική χαμένη ζήτηση για κάθε είδος προϊόντος σε μορφή κοντέινερ φορτηγών ξεχωριστά (π.χ. Total_lost_WG_BC_demand).

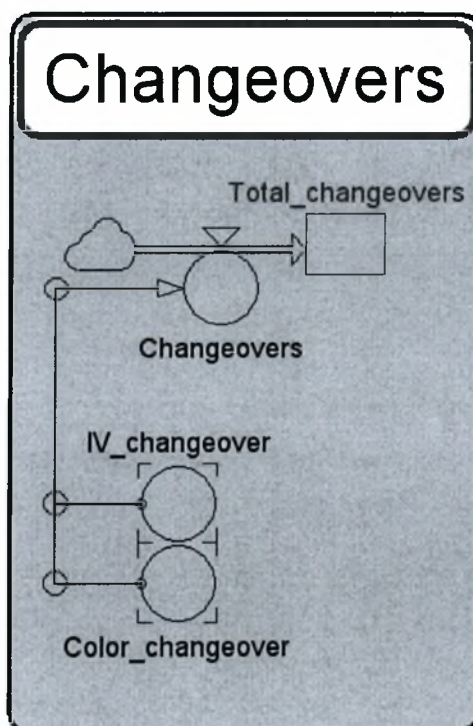


Σχήμα 21. Ανικανοποίητη ζήτηση και fill rate

6.4.13 Υπολογισμός του συνόλου των μεταβάσεων

Στο πεδίο αυτό υπολογίζεται ο συνολικός αριθμός των μεταβάσεων (Total_changeovers) κατά τη διάρκεια του χρονικού ορίζοντα προσομοίωσης, ο οποίος μαζί με το fill rate αποτελούν τα αντικειμενικά κριτήρια ελέγχου της αποδοτικότητας του μοντέλου και των ενδογενών πολιτικών ελέγχου του. Ειδικότερα, κατά τη διάρκεια ενός τετραώρου υπολογίζεται ο συνολικός αριθμός των μεταβάσεων (Changeovers) ο οποίος προκύπτει από το άθροισμα των μεταβάσεων χρώματος στο στάδιο POLY (Color_changeover) και των μεταβάσεων ιξώδους στο στάδιο SSP (IV_changeover). Ο συνολικός αριθμός των μεταβάσεων ανά τετράωρο αθροίζει σε ένα επίπεδο το οποίο υπολογίζει το συνολικό αριθμό μεταβάσεων κατά τη διάρκεια του χρονικού ορίζοντα προγραμματισμού (Total_changeovers).

Αντικειμενικός στόχος του μοντέλου ΔΣ είναι η βελτίωση (και όχι βελτιστοποίηση) της απόδοσης του συστήματος, δηλαδή ο περιορισμός των μεταβάσεων κατά τη διάρκεια των οποίων δημιουργούνται παραπροϊόντα χαμηλών ποιοτικών χαρακτηριστικών και η ικανοποίηση όσο το δυνατόν περισσότερης ζήτησης. Το μοντέλο ΔΣ δημιουργήθηκε με τέτοιο τρόπο ώστε ο αριθμός των παραμέτρων των διαδικασιών ελέγχου που απαιτείται να οριστούν εξωγενώς να είναι ο ελάχιστος δυνατός. Ο στόχος αυτός θεωρείται ότι επιτεύχθηκε καθώς για τη λειτουργία των διαδικασιών ελέγχου απαιτείται ο εξωγενής ορισμός μόνο των ορίων της κάτω στάθμης παραγγελίας και της άνω στάθμης πλήρωσης των αποθεμάτων της πολιτικής «Ss» που χρησιμοποιείται για την αναπλήρωση των αποθεμάτων τελικών προϊόντων των σιλό του LFSS.



Σχήμα 22. Υπολογισμός των συνολικών μεταβάσεων χρώματος και ιξώδους

6.4.14 Διαμόρφωση μαθηματικού μοντέλου.

Το επόμενο βήμα από τη διαμόρφωση του διαγράμματος ροής είναι η σύνδεση των μεταβλητών με μαθηματικές σχέσεις, το σύνολο των οποίων αποτελούν το μαθηματικό μοντέλο. Οι μαθηματικές εξισώσεις που χρησιμοποιούνται κατά τη διαμόρφωση των μαθηματικών μοντέλων ΔΣ χωρίζονται σε δύο κατηγορίες:

- α) στις καταστατικές εξισώσεις, και
- β) στις εξισώσεις ροών.

Στο Παράρτημα Α παρουσιάζεται το σύνολο των εξισώσεων του μοντέλου που δημιουργήθηκε στα πλαίσια της παρούσας μεταδιπλωματικής εργασίας προκειμένου να προκύψει ένα εφικτό πρόγραμμα παραγωγής για τη χημική βιομηχανία και να μειωθούν οι απαιτούμενες μεταβάσεις χρώματος και ιξώδους.

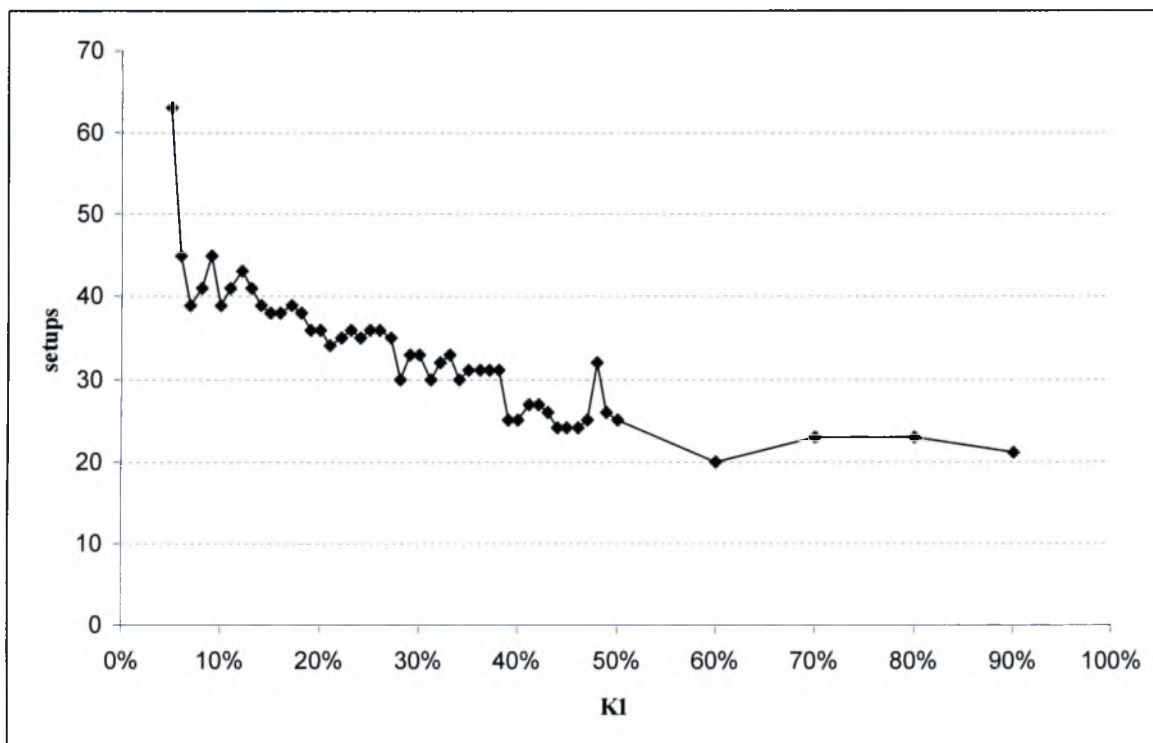
7 ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ - ΣΥΝΟΨΗ

7.1 Αποτελέσματα μοντέλου ΔΣ

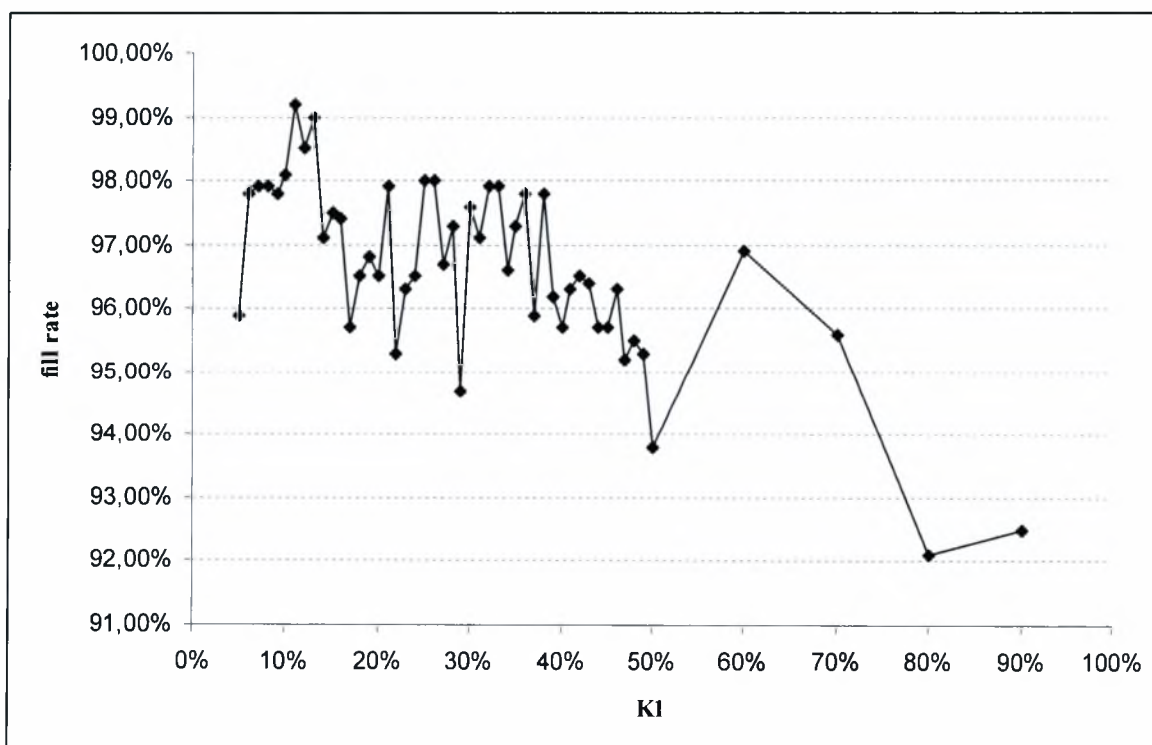
Λόγω της ευελιξίας που προσέφερε το μοντέλο ΔΣ κατά τη σχεδίαση των διαδικασιών ελέγχου της παραγωγικής διαδικασίας, το πλήθος των μεταβλητών οι οποίες πρέπει να βελτιωθούν εξωγενώς προκειμένου να βελτιωθεί η λειτουργία της βιομηχανίας ανέρχεται μόλις σε μία μεταβλητή: το ποσοστό K1 των πολιτικών Ss διαχείρισης των αποθεμάτων του LFSS (παράγραφος 6.4.2). Οι τιμές όλων των υπολοίπων μεταβλητών προκύπτουν από τις ενδογενείς διαδικασίες του μοντέλου οι οποίες περιγράφηκαν συνοπτικά στο έκτο κεφάλαιο και παρουσιάζονται αναλυτικά στο Παράρτημα Α. Η εξωγενής διαδικασία βελτίωσης περιλαμβάνει τη συστηματική δοκιμή όλων των τιμών του K1 σε ένα αποδεκτό διάστημα τιμών αυτού ενώ ως κριτήριο «βελτιστοποίησης» χρησιμοποιούνται δύο μεταβλητές εξόδου: (1) το ποσοστό της ζήτησης που ικανοποιείται άμεσα (μεταβλητή Fill_rate, παρ. 6.4.12) και (2) το πλήθος των συνολικών μεταβάσεων χρώματος και εξόδους του εξοπλισμού (μεταβλητή Total_changeovers, παρ. 6.4.13). Στον Πίνακα 7 και στα Σχήματα 23 και 24 παρουσιάζονται οι τιμές των δύο μεταβλητών εξόδου συναρτήσει της τιμής του K1 στο εύρος τιμών από 5% έως 50% με βήμα 1% και από 50% έως 90% με βήμα 10%.

Πίνακας 7. Total_changeovers και Fill_rate συναρτήσει του K1

K1	Total changeovers	Fill rate	K1	Total changeovers	Fill rate
5%	63	95,9%	30%	33	97,6%
6%	45	97,8%	31%	30	97,1%
7%	39	97,9%	32%	32	97,9%
8%	41	97,9%	33%	33	97,9%
9%	45	97,8%	34%	30	96,6%
10%	39	98,1%	35%	31	97,3%
11%	41	99,2%	36%	31	97,8%
12%	43	98,5%	37%	31	95,9%
13%	41	99,0%	38%	31	97,8%
14%	39	97,1%	39%	25	96,2%
15%	38	97,5%	40%	25	95,7%
16%	38	97,4%	41%	27	96,3%
17%	39	95,7%	42%	27	96,5%
18%	38	96,5%	43%	26	96,4%
19%	36	96,8%	44%	24	95,7%
20%	36	96,5%	45%	24	95,7%
21%	34	97,9%	46%	24	96,3%
22%	35	95,3%	47%	25	95,2%
23%	36	96,3%	48%	32	95,5%
24%	35	96,5%	49%	26	95,3%
25%	36	98,0%	50%	25	93,8%
26%	36	98,0%	60%	20	96,9%
27%	35	96,7%	70%	23	95,6%
28%	30	97,3%	80%	23	92,1%
29%	33	94,7%	90%	21	92,5%



Σχήμα 23. Γραφική απεικόνιση του πλήθους των setups συναρτήσει του KI



Σχήμα 24. Γραφική απεικόνιση του fill rate συναρτήσει του KI

7.2 Ανάλυση των αποτελεσμάτων του μοντέλου ΔΣ

Τα αποτελέσματα του Πίνακα 7 καθώς και η οπτικοποίησή τους μέσω των Σχημάτων 23 και 24 οδηγούν στα ακόλουθα συμπεράσματα:

➤ Όσο αυξάνεται η τιμή του $K1$ τόσο μειώνεται το fill rate. Το αποτέλεσμα αυτό προκύπτει διότι αύξηση της τιμής του $K1$ συνεπάγεται αύξηση του εύρους των αποθεμάτων προϊόντων στο οποίο η χημική βιομηχανία δε λαμβάνει καμία απόφαση αναπλήρωσής τους. Ειδικότερα και σύμφωνα με την πολιτική Ss , η Διοίκηση δε λαμβάνει καμία απόφαση όσο το απόθεμα ενός προϊόντος δεν πέσει κάτω από την τιμή $s1=(1-K1)*(Επιθυμητό\ απόθεμα)$. Αυτό σημαίνει ότι όσο το $K1$ αυξάνει τόσο περισσότερο αυτή η στάθμη πλησιάζει στο μηδέν με αποτέλεσμα εάν εκδηλωθεί μία υψηλή ζήτηση για ένα προϊόν του οποίου το απόθεμα είναι κοντά ή κάτω από τη στάθμη $s1$ τότε είναι πολύ πιθανό αυτή η ζήτηση να μη μπορεί να ικανοποιηθεί πλήρως. Από την άλλη, όταν το απόθεμα πέσει κάτω από τη στάθμη $S2$ τότε η βιομηχανία θα πρέπει να δεσμεύσει μεγάλο χρονικό διάστημα για την παραγωγή της μεγαλύτερης πλέον ποσότητας $S2$ - (Υφιστάμενο απόθεμα) γιατί η μεγάλη τιμή του $K1$ αφενός έχει επιτρέψει στο υφιστάμενο απόθεμα να λάβει χαμηλή τιμή και αφετέρου, έχει ανεβάσει τη μέγιστη στάθμη $S2$. Επομένως συμπερασματικά μπορεί να ειπωθεί ότι η αύξηση του $K1$ έχει ως αποτέλεσμα τη μείωση της ευελιξίας της παραγωγικής διαδικασίας προς αναπλήρωση των αποθεμάτων της, με αποτέλεσμα τη μείωση του fill rate.

➤ Όσο αυξάνεται η τιμή του $K1$ τόσο μειώνονται τα setups. Όπως περιγράφηκε συνοπτικά στην παράγραφο διαχείρισης των αποθεμάτων του LFSS και παρουσιάζεται αναλυτικά στο Παράρτημα Α, όταν αποφασιστεί η παραγωγή ενός προϊόντος για αναπλήρωση των αποθεμάτων του τότε δε λαμβάνεται καμία απόφαση μετάβασης για χρονικό διάστημα που αντιστοιχεί στην παραγωγή ποσότητας τουλάχιστον $S2$ - (Υφιστάμενο απόθεμα). Όμως, όσο αυξάνει το $K1$ τόσο μειώνεται το $s1$ και αυξάνεται το $S2$ κι επομένως αυξάνεται η ποσότητα του προϊόντος που πρέπει να αναπληρωθεί με αποτέλεσμα να δεσμεύεται μεγαλύτερος παραγωγικός χρόνος για αυτήν την αναπλήρωσή στον οποίο η χημική βιομηχανία δεν επιτρέπεται να λάβει κάποια απόφαση μετάβασης. Επομένως συμπερασματικά μπορεί να ειπωθεί ότι η αύξηση του $K1$ έχει ως αποτέλεσμα τη μείωση της ευελιξίας της παραγωγικής διαδικασίας προς αναπλήρωση των αποθεμάτων της, με αποτέλεσμα τη μείωση των setups.

➤ Η βέλτιστη τιμή του $K1$ είναι 11% για την οποία προκύπτουν 41 setups και fill rate 99,2%. Το αποτέλεσμα αυτό σημαίνει ότι για να μεγιστοποιείται η ευελιξία της παραγωγικής διαδικασίας τα όρια $s1$ και $S2$ δεν πρέπει να έχουν ούτε μεγάλη «απόσταση» αλλά ούτε και μικρή από τα επιθυμητά επίπεδα ασφαλείας. Επιπλέον, αν αναλογιστεί κανείς ότι στην περίπτωση του μοντέλου ΔΣ η ζήτηση πρέπει να ικανοποιείται άμεσα τη στιγμή που εκδηλώνεται και όχι να προγραμματίζεται η ικανοποίησή της για αργότερα, η τιμή του fill rate είναι πολύ υψηλή.

- Το fill rate είναι μεγαλύτερο από 92% για όλες τις τιμές του K1 που σημειώνονται στον Πίνακα 24 ενώ για τις περισσότερες από αυτές είναι μεγαλύτερο και από 95%. Το αποτέλεσμα αυτό σημαίνει ότι οι ενδογενείς διαδικασίες του μοντέλου για τον ορισμό των πολιτικών διαχείρισης των αποθεμάτων και για τον προγραμματισμό της παραγωγικής διαδικασίας, σε συνδυασμό με τις προτεραιότητες μεταξύ των προϊόντων ως προς την ικανοποίηση των παραγγελιών αναπλήρωσης των αποθεμάτων του που ορίστηκαν εξωγενώς, είναι πράγματι αποδοτικές και τα αποτελέσματα που προέκυψαν για το «βέλτιστο» K1 δεν οφείλονται σε τυχαίο γεγονός.
- Στο σύνολο σχεδόν των υπό μελέτη περιπτώσεων για τις εναλλακτικές τιμές του ποσοστού K1 προκύπτει ότι ο αριθμός των μεταβάσεων χρώματος και ιξώδους της παραγωγικής διαδικασίας είναι μεταξύ 20 και 45 μεταβάσεις. Η σχετική απόκλιση μεταξύ αυτών των δύο τιμών είναι αρκετά μεγάλη καθώς οι 45 μεταβάσεις είναι υπερδιπλάσιες των 20. Επομένως, το πλήθος των μεταβάσεων της παραγωγικής διαδικασίας δε βρίσκεται σε ένα συστηματικό αυστηρά περιορισμένο εύρος τιμών κι επομένως δε μπορούν να εξαχθούν ασφαλή συμπεράσματα για τη συμπεριφορά αυτής της μεταβλητής.
- Παρόλο που τόσο το πλήθος των μεταβάσεων όσο και η τιμή του fill rate επηρεάζονται από την τιμή του K1, ωστόσο από τη συγκριτική παράθεση των Σχημάτων 23 και 24 προκύπτει ότι η ισχύς των επιδράσεων αυτών διαφέρει. Ειδικότερα, ενώ η ισχύς της επίδρασης της τιμής του K1 στο πλήθος των μεταβάσεων είναι μεγάλη (εύρος τιμών από 20 έως 63 μεταβάσεις), δε συμβαίνει το ίδιο και με την τιμή του fill rate (εύρος τιμών από 92,1% έως 99,2%). Η χαμηλή ισχύς της επίδρασης του ποσοστού K1 στο fill rate είναι περισσότερο επιθυμητή από την αντίστοιχη της στο πλήθος των μεταβάσεων κι αυτό διότι ως αντικειμενικός στόχος, η ικανοποίηση της ζήτησης είναι πολύ πιο σημαντική από ότι η μείωση του αριθμού των μεταβάσεων. Για παράδειγμα εάν συνέβαινε το αντίθετο, δηλαδή το πλήθος των μεταβάσεων είχε πιο σταθερή συμπεριφορά και το fill rate παρουσίαζε τόσο μεγάλες αποκλίσεις, τότε το μοντέλο ΔΣ δε θα ήταν αποδοτικό καθώς ένα μεγάλο ποσοστό της ζήτησης θα έμενε ανικανοποίητο.
- Ποσοτικοποιώντας τα αποτελέσματα της ανικανοποίητης ζήτησης για K1=11%, προκύπτει ότι από συνολική ζήτηση του εξαμήνου των 32.683 τόνων η ποσότητα που δεν ικανοποιείται άμεσα από τα υφιστάμενα αποθέματα και επομένως χάνεται αντιστοιχεί σε 261,5 τόνους. Ωστόσο εάν θεωρηθεί ότι οι πελάτες της βιομηχανίας προκειμένου να εξασφαλίσουν σε ένα μακροπρόθεσμο ορίζοντα ότι οι παραγγελίες που θέτουν θα ικανοποιούνται άμεσα σε ποσοστό της τάξεως του 99% τότε θα είναι διατεθειμένοι να δεχτούν την πιθανότητα 1% η ικανοποίηση των παραγγελιών τους να γίνει με μικρή καθυστέρηση, τότε θα μπορούσε να επιτευχθεί fill rate ακόμη και 100%. Λαμβάνοντας υπόψη ότι το σύνολο των Διοικήσεων των βιομηχανιών επιθυμεί υψηλή ταχύτητα απόκρισης των προμηθευτών της, είναι πολύ πιθανό μία τέτοια πρόταση από την πλευρά της χημικής βιομηχανίας να γινόταν ακόμη και άμεσα αποδεκτή.

7.3 Σύγκριση των αποτελεσμάτων των μοντέλων ΔΣ και ΓΠ

Όπως προέκυψε στην εργασία των Liberopoulos et al.^a (2009), η επίλυση του μοντέλου MILP ελαχιστοποίησης των συνολικών μεταβάσεων χρώματος και ιξώδους αποδίδει σαν βέλτιστη λύση τις 45 μεταβάσεις ενώ παράλληλα ικανοποιείται το σύνολο της ζήτησης. Από την άλλη, η επίλυση του μοντέλου ΔΣ διαχείρισης των αποθεμάτων και της παραγωγικής διαδικασίας που αναπτύχθηκε στην παρούσα μεταδιπλωματική εργασία αποδίδει ως βελτιωμένη λύση τις 45 μεταβάσεις ενώ παράλληλα ικανοποιείται το 99,2% της συνολικής ζήτησης. Παρατηρώντας μακροσκοπικά αυτά τα αποτελέσματα προκύπτουν τα εξής συμπεράσματα:

- Το μοντέλο MILP βασίζεται σε αναλυτικές μεθοδολογίες επίλυσης οι οποίες αποδίδουν σε κάθε περίπτωση τη συνολικά βέλτιστη λύση του μαθηματικού μοντέλου υπό τους συγκεκριμένους περιορισμούς που περιέχει. Αυτό σημαίνει ότι ο ελάχιστος αριθμός μεταβάσεων χρώματος και ιξώδους της παραγωγικής διαδικασίας δε μπορεί να είναι μικρότερος από 45 (βέλτιστη λύση του MILP των Liberopoulos et al.^a, 2009) και ταυτόχρονα να ικανοποιείται το σύνολο της ζήτησης. Για το λόγο αυτό η λύση του μοντέλου MILP αποτελεί ικανό σημείο σύγκρισης για την αποδοτικότητα και του μοντέλου ΔΣ.
- Από την επίλυση του μοντέλου ΔΣ προκύπτει ότι εάν εισαχθούν αποδοτικές ενδογενείς πολιτικές ελέγχου της λειτουργίας του συστήματος και ταυτόχρονα ρυθμιστεί κατάλληλα η τιμή του ποσοστού K1 που ορίζει τις στάθμες S2 και s1 των πολιτικών διαχείρισης των αποθεμάτων τότε η αποδιδόμενη λύση μπορεί να προσεγγίζει ακόμη και τη βέλτιστη λύση προγραμματισμού της παραγωγής του μοντέλου MILP. Για το λόγο αυτό κατά τη σχεδίαση ενός μοντέλου ΔΣ απαιτείται ταυτόχρονα η άριστη γνώση τόσο της διαδικασίας μοντελοποίησης και του λογισμικού προσομοίωσης όσο και του πραγματικού συστήματος και των μεθόδων βελτίωσης και βελτιστοποίησης του προγραμματισμού παραγωγής.
- Ο χρόνος επίλυσης ενός μοντέλου αποτελεί στα περισσότερα πραγματικά συστήματα παραγωγής μία σημαντική παράμετρο κατά τη διαδικασία αξιολόγησης της μεθοδολογίας επίλυσης του προβλήματος. Το μοντέλο MILP απαιτεί για την επίλυσή του περίπου 7 λεπτά (420,4 second) ενώ το μοντέλο ΔΣ απαιτεί στον ίδιο Η/Υ χρόνο προσομοίωσης περίπου 2 δευτερολέπτων. Επιπλέον, ο χρόνος προσομοίωσης για το μοντέλο ΔΣ παραμένει ανεξάρτητος της πολυπλοκότητας του μοντέλου ενώ και η επιρροή σε αυτόν από τη διάρκεια του χρονικού ορίζοντα προγραμματισμού του προβλήματος είναι ανεπαίσθητη. Αντιθέτως, ο χρόνος επίλυσης ενός μοντέλου MILP εξαρτάται εμφανώς τόσο από την πολυπλοκότητα της δομής του μοντέλου όσο και από τη διάρκεια του χρονικού ορίζοντα προγραμματισμού του προβλήματος.

➤ Προκειμένου το μοντέλο MILP να μπορεί να αποδώσει μία βέλτιστη λύση θα πρέπει να ικανοποιούνται όλοι οι περιορισμοί του μοντέλου. Το γεγονός αυτό έχει ως αποτέλεσμα την αδυναμία απόδοσης μίας εφικτής λύσης εάν κάποιος ή κάποιοι από τους περιορισμούς παραβιάζονται. Για παράδειγμα, εάν κατά τη διάρκεια μίας εβδομάδας σημειωθεί μία απότομη αύξηση της ζήτησης η οποία δε θα μπορεί να ικανοποιηθεί από το συνδυασμό των αποθεμάτων και της παραγωγικής διαδικασίας, τότε δε θα μπορεί να αποδοθεί από το λογισμικό επίλυσης μία εφικτή λύση. Αντιθέτως, τα μοντέλα ΔΣ αποδίδουν πάντοτε μία εφικτή λύση ανεξαρτήτως των συνθηκών που αναπτύσσονται κατά τη διάρκεια της προσομοίωσης. Ωστόσο όπως προαναφέρθηκε, η ποιότητα της αποδιδόμενης λύσης εξαρτάται από την ποιότητα των ενδογενών διαδικασιών ελέγχου και κυρίως των μηχανισμών αναδράσεων (control loops). Για το λόγο αυτό, στη γενική περίπτωση είναι άγνωστο αν η λύση η οποία λαμβάνεται από ένα πρότυπο ΔΣ προσεγγίζει ή όχι τη βέλτιστη λύση και με τι ακρίβεια.

➤ Η αυξημένη ευελιξία της Θεωρίας ΔΣ επιτρέπει τη σχεδίαση ενός μοντέλου το οποίο απαιτεί τον εξωγενή ορισμό μίας μόνο μεταβλητής και ειδικότερα του ποσοστού K1. Η ευελιξία αυτή επιτρέπει το συνυπολογισμό στο μοντέλο ακόμη περισσότερων παραγόντων του πραγματικού συστήματος οι οποίοι καθιστούν τις διαδικασίες λήψης αποφάσεων ακόμη πιο αποδοτικές. Ωστόσο για να είναι τα αποτελέσματα των δύο μοντέλων άμεσα συγκρίσιμα μεταξύ τους, περιλαμβάνονται στο μοντέλο ΔΣ μόνο οι παράγοντες που συμπεριλαμβάνονται στο μοντέλο MILP.

➤ Το μοντέλο ΔΣ μπορεί να περιλάβει και άλλα φαινόμενα τα οποία επηρεάζουν τον προγραμματισμό παραγωγής, όπως είναι για παράδειγμα οι βλάβες και τα τυχαία γεγονότα. Για λόγους επίδειξης της λειτουργίας τέτοιων παραγόντων έχουν συμπεριληφθεί στο μοντέλο ΔΣ μεταβλητές οι οποίες προσομοιώνουν φαινόμενα βλαβών και επιδιόρθωσης του εξοπλισμού, προγραμματισμένης συντήρησης, καθώς και διακοπή της παραγωγής με απόφαση της Διοίκησης. Ωστόσο οι διαδικασίες αυτές παραμένουν ανενεργές (μηδενική πιθανότητα συμβάντος) ώστε τα αποτελέσματα των δύο μοντέλων να είναι μεταξύ τους άμεσα συγκρίσιμα.

➤ Το μοντέλο ΔΣ εφαρμόζει πολιτικές διαχείρισης των αποθεμάτων τύπου Ss οι οποίες είναι απλές και κατανοητές τόσο από τους εργαζομένους όσο και από τη Διοίκηση. Αυτό σημαίνει ότι η λειτουργία ενός παραγωγικού συστήματος μπορεί να βελτιωθεί από την εγκαθίδρυση απλοϊκών κανόνων οι οποίοι είναι εύκολοι στην εφαρμογή τους, και κατ' επέκταση γίνονται εύκολα αποδεκτοί από το προσωπικό.

➤ Συγκεντρωτικά προτείνεται η ανά περίπτωση χρήση εκείνου του μοντέλου για το οποίο μπορεί να λεχθεί ότι ισχύουν οι υποθέσεις και οι περιορισμοί του. Ειδικότερα όταν ισχύουν οι προϋποθέσεις του MILP τότε είναι προτιμότερο να χρησιμοποιείται αυτό, εναλλακτικά είναι προτιμότερο να χρησιμοποιείται το μοντέλο ΔΣ.

7.4 Συμπεράσματα από τη χρήση των δύο μεθοδολογιών προσέγγισης

Από τη χρήση των δύο μεθοδολογιών προσέγγισης, MILP και ΔΣ, προέκυψαν τα εξής συμπεράσματα:

- Οι αναλυτικές μεθοδολογίες επίλυσης όπως είναι ο MILP δημιουργούν αφαιρετικά μοντέλα τα οποία αποδίδουν τις συνολικά βέλτιστες λύσεις, δεδομένου ότι ισχύουν οι περιορισμοί τους. Η ενδελεχής μελέτη αυτών των μεθόδων στη διεθνή βιβλιογραφία επιβεβαιώνει τη σημαντικότητα τους καθώς εύρεση βέλτιστων λύσεων αποτελεί το σημαντικότερο στόχο όλων των συστημάτων.
- Η ΔΣ επιτρέπει τη δημιουργία μοντέλων να οποία περιλαμβάνουν πιο ρεαλιστικές παραδοχές λόγω της ευελιξίας που επιτρέπουν κατά τη διαδικασία μοντελοποίησης. Το γεγονός αυτό επιτρέπει την προσομοίωση σύνθετων συστημάτων που δε μπορούν να απεικονιστούν με τη χρήση των κλασικών μεθόδων απεικόνισης και βελτιστοποίησης.
- Ο χρόνος επίλυσης των μοντέλων προσομοίωσης είναι πολύ μικρός κι επομένως επιτρέπει τη διακριτοποίηση του χρόνου σε πολύ μικρότερα χρονικά βήματα (dt). Το γεγονός αυτό έχει ως αποτέλεσμα την αυξημένη ευελιξία και αποκρισιμότητα (responsiveness) των μοντέλων ΔΣ καθώς επιτρέπεται η λήψη αποφάσεων σε μικρότερα χρονικά διαστήματα.
- Η ποιότητα μίας αποδιδόμενης λύσης εξαρτάται από τη δομή του μοντέλου, τόσο για τις αναλυτικές μεθόδους απεικόνισης και επίλυσης όσο και για τις προσομοιωτικές. Όσον αφορά τα αναλυτικά μοντέλα, γίνονται κατά τη δημιουργία τους διάφορες παραδοχές οι οποίες είναι άλλοτε περισσότερο και άλλοτε λιγότερο ρεαλιστικές, προκειμένου το μοντέλο να επιδέχεται επίλυση σε ένα σχετικά αποδεκτό χρόνο. Αντιθέτως, ένα προσομοιωτικό μοντέλο μπορεί να είναι εξαιρετικά λεπτομερές με σχετικά μικρό χρόνο επίλυσης αλλά χωρίς καμία εγγύηση στην ποιότητα της αποδιδόμενης λύσης η οποία εξαρτάται από την ποιότητα των ενδογενών πολιτικών του μοντέλου.
- Τα μοντέλα ΔΣ αποδίδουν πάντοτε μία λύση η οποία άλλοτε μπορεί να είναι ποιοτικά ανώτερη και άλλοτε κατώτερη. Αντιθέτως τα αναλυτικά μοντέλα αποδίδουν λύσεις μόνο όταν δεν παραβιάζονται οι περιορισμοί τους, ωστόσο εγγυώνται πάντοτε τη βελτιστότητά τους μέσω των διαδικασιών βελτιστοποίησης που περιέχουν.
- Τα μοντέλα ΔΣ επιτρέπουν την οπτικοποίηση του υπό μελέτη συστήματος μέσω της δημιουργίας των διαγραμμάτων ροής και επιρροής. Με τον τρόπο αυτό είναι ευκολότερη τόσο η κατανόηση των μηχανισμών λειτουργίας του πραγματικού συστήματος, όσο και η διαδικασία μοντελοποίησης.

7.5 Προτάσεις για περαιτέρω διερεύνηση.

Προκειμένου να προκύψουν πιο ασφαλή συμπεράσματα για την αποτελεσματικότητα του μοντέλου ΔΣ ως προς τη συνολικά βέλτιστη λύση του μοντέλου MILP (υπό τις συγκεκριμένες προϋποθέσεις) παρατίθενται οι παρακάτω προτάσεις για περαιτέρω διερεύνηση:

- Θα πρέπει να ελεγχθεί εάν η τιμή του ποσοστού K1 που υπολογίστηκε από το μοντέλο ΔΣ παραμένει βέλτιστη για περισσότερα σετ ζητήσεων τα οποία ακολουθούν την ίδια κατανομή ζήτησης ή ακόμη και διαφορετική κατανομή (robustness).
- Έχοντας εξασφαλίσει την καταλληλότητα της Θεωρίας ΔΣ ως μεθοδολογίας προσέγγισης του προβλήματος, θα πρέπει να ελεγχθεί εάν μειώνοντας τη διάρκεια του χρονικού βήματος επιτυγχάνεται βελτιωμένη τιμή του fill rate λόγω της αυξημένης ευελιξίας που θα έχει στην περίπτωση αυτή το μοντέλο.
- Θα πρέπει να ελεγχθεί εάν μειώνοντας το χρονικό βήμα του MILP επιτυγχάνεται καλύτερη λύση. Ο συνδυασμός του χρόνου επίλυσης και της βελτίωσης της αποδιδόμενης βέλτιστης λύσης μπορεί να οδηγήσει σε ένα ποιοτικό κριτήριο (π.χ. ποιοτική κυρτή καμπύλη) το οποίο να εντοπίζει το βέλτιστο χρονικό βήμα συναρτήσει του διαθέσιμου χρόνου επίλυσης.
- Τέλος, τα μοντέλα δημιουργούνται προκειμένου να δωθούν απαντήσεις σε πραγματικά προβλήματα. Θα πρέπει λοιπόν να συζητηθούν οι προτάσεις και τα αποτελέσματα της παρούσας μεταδιπλωματικής εργασίας με τη Διοίκηση της χημικής βιομηχανίας προκειμένου να ελεγχθεί το πώς μπορεί να βελτιωθεί ο χρόνος ικανοποίησης της ζήτησης καθώς το fill rate.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Ελληνική

- Γεωργιάδης Πάτροκλος, 2006. Θεωρία Δυναμικής Συστημάτων με εφαρμογές στη σχεδίαση και λειτουργία συστημάτων παραγωγής. Εκδόσεις Σοφία, Θεσσαλονίκη.
- Μανώλης Λουκάκης, 1994. Γραμμικός προγραμματισμός - Αριστοποίηση σε δίκτυα - Τόμος Α. Εκδόσεις Σοφία, Θεσσαλονίκη.
- Ψωινός Π. Δημήτριος, 1999. Ποσοτική Ανάλυση - Πρώτος Τόμος, Δεύτερη Έκδοση. Εκδόσεις Ζήτη, Θεσσαλονίκη.

Διεθνής

- Dallery Y., Gershwin S.B. (1992). Manufacturing flow line systems: A review of models and analytical results. *Queuing Systems*, vol. 12 (1-2), pp. 3-94.
- Fourer, Gay D, Kernighan B, 2002. AMPL - A modeling language for mathematical programming, Second edition - Thomson, Brooks/Cole, Duxbury. (www.thomsonlearning.com)
- Liberopoulos G, Kozanidis G, Hatzikonstantinou O, 2009^a. Production scheduling of a multi-grade PET resin plant. *Computers and Chemical Engineering*, doi: 10.1016/j.compchemeng.2009.05.017
- Liberopoulos G, Pandelis D, Hatzikonstantinou O, 2009^b. The stochastic economic lot sizing problem for continuous multi-grade production. In: 7th conference on stochastic modeling of manufacturing and service operations, June 7-12, 2009, Ostuni, Italy.
- Martinich J, 1997. *Production and Operations Management - An Applied Modern Approach*. University of Missouri, St. Louis.
- McGehee J.F., Johnson J.A., Bertelli C. (2004). Maximizing PET SSP line profitability through world scale process design and operations. In: POLYESTER, 9th world congress, 7-9 December, 2004, Zurich, Switzerland.
- Sterman J, 2000. *Business Dynamics - Systems Thinking and Modeling for a Complex World*, McGraw - Hill Higher Education.
- Terrazas-Moreno S, Flores-Tlacuahuac A, Grossmann I.E., 2007. Simultaneous cyclic scheduling and optimal control of polymerizations reactors. *AIChE Journal*, vol. 53 (9), pp. 2301-2315.

ΠΑΡΑΡΤΗΜΑ Α

**Μοντέλο του υπό μελέτη συστήματος σε γλώσσα
προγραμματισμού Powersim 2.5c**

Συντομογραφίες:

init≡ initial value of a stock
 flow≡ inflow and outflow of a stock
 doc≡ documentation of the variable
 aux≡ auxiliary variable
 const≡ constant

```

init   Color_at_t_1 = 1
flow   Color_at_t_1 = -dt*R3+dt*R2
doc    Color_at_t_1 = By this variable we keep track of the color that POLY had at time
        step t-1.
        Color at t-1=1 is equal to Color at t-1=LIGHT
        Color at t-1=2 is equal to Color at t-1=GRAY
        Color at t-1=3 is equal to Color at t-1=DARK
-----
init   FH_WH_stock = 500
flow   FH_WH_stock = -dt*FH_WH_outflow+dt*FH_WH_inflow
doc    FH_WH_stock = Quantity of FH stored in the warehouse.
-----
init   G_WH_stock = 0
flow   G_WH_stock = +dt*G_WH_inflow-dt*G_WH_outflow
doc    G_WH_stock = Quantity of G stored in the warehouse.
-----
init   IV_stage_1 = 2
flow   IV_stage_1 = -dt*R5+dt*R4
doc    IV_stage_1 = This variable represents the first 4-hour period of an IV changeover
        (if any).
-----
init   IV_stage_2 = 2
flow   IV_stage_2 = -dt*R6+dt*R5
doc    IV_stage_2 = This variable represents the second 4-hour period of an IV changeover
        (if any).
-----
init   IV_stage_4 = 2
flow   IV_stage_4 = -dt*R8+dt*R7
doc    IV_stage_4 = This variable represents the fourth 4-hour period of an IV changeover
        (if any).
-----
init   IV_stage_5 = 2
flow   IV_stage_5 = -dt*R9+dt*R8
doc    IV_stage_5 = This variable represents the fifth 4-hour period of an IV changeover
        (if any).
-----
init   IV_stage_6 = 2
flow   IV_stage_6 = -dt*R10+dt*R9
doc    IV_stage_6 = This variable represents the sixth 4-hour period of an IV changeover
        (if any).
-----
init   LFSS1 = 400
flow   LFSS1 = -dt*LFSS1_ST-dt*LFSS1_BC-dt*LFSS1_BB+dt*LFSS1_inflow
doc    LFSS1 = Quantity of the grade stored in LFSS1.
-----
init   LFSS1_grade = 1
flow   LFSS1_grade = -dt*R12 +dt*R11
doc    LFSS1_grade = This variable represents the type of the grade contained in LFSS1.
-----
init   LFSS2 = 400
flow   LFSS2 = -dt*LFSS2_ST-dt*LFSS2_BC-dt*LFSS2_BB+dt*LFSS2_inflow
doc    LFSS2 = Quantity of the grade stored in LFSS2.
-----
init   LFSS2_grade = 1
flow   LFSS2_grade = -dt*R14+dt*R13
doc    LFSS2_grade = This variable represents the type of the grade contained in LFSS2.
-----
init   LFSS3 = 400
flow   LFSS3 = -dt*LFSS3_ST-dt*LFSS3_BC-dt*LFSS3_BB+dt*LFSS3_inflow
doc    LFSS3 = Quantity of the grade stored in LFSS3.
-----
init   LFSS3_grade = 4
flow   LFSS3_grade = -dt*R16+dt*R15
doc    LFSS3_grade = This variable represents the type of the grade contained in LFSS3.
-----
init   LFSS4 = 400
    
```

```

flow  LFSS4 = -dt*LFSS4_ST-dt*LFSS4_BC-dt*LFSS4_BB+dt*LFSS4_inflow
doc   LFSS4 = Quantity of the grade stored in LFSS4.
-----
init  LFSS4_grade = 4
flow  LFSS4_grade = +dt*R17-dt*R18
doc   LFSS4_grade = This variable represents the type of the grade contained in LFSS4.
-----
init  LFSS5 = 350
flow  LFSS5 = -dt*LFSS5_ST-dt*LFSS5_BC-dt*LFSS5_BB+dt*LFSS5_inflow
doc   LFSS5 = Quantity of the grade stored in LFSS5.
-----
init  LFSS5_grade = 4
flow  LFSS5_grade = +dt*R19-dt*R20
doc   LFSS5_grade = This variable represents the type of the grade contained in LFSS5.
-----
init  LFSS6 = 150
flow  LFSS6 = -dt*LFSS6_BC-dt*LFSS6_ST-dt*LFSS6_BB+dt*LFSS6_inflow
doc   LFSS6 = Quantity of the grade stored in LFSS6.
-----
init  LFSS6_grade = 2
flow  LFSS6_grade = +dt*R21-dt*R22
doc   LFSS6_grade = This variable represents the type of the grade contained in LFSS6.
-----
init  LFSS7 = 0
flow  LFSS7 = -dt*LFSS7_BC-dt*LFSS7_ST-dt*LFSS7_BB+dt*LFSS7_inflow
doc   LFSS7 = Quantity of the grade stored in LFSS7.
-----
init  LFSS7_grade = 0
flow  LFSS7_grade = +dt*R23-dt*R24
doc   LFSS7_grade = This variable represents the type of the grade contained in LFSS7.
-----
init  LFSS8 = 0
flow  LFSS8 = -dt*LFSS8_BC-dt*LFSS8_ST-dt*LFSS8_BB+dt*LFSS8_inflow
doc   LFSS8 = Quantity of the grade stored in LFSS8.
-----
init  LFSS8_grade = 0
flow  LFSS8_grade = +dt*R25-dt*R26
doc   LFSS8_grade = This variable represents the type of the grade contained in LFSS8.
-----
init  POLY_Current_Color = 1
flow  POLY_Current_Color = -dt*R2+dt*R1
doc   POLY_Current_Color = Current color=1 is equal to Current color=LIGHT
      Current color=2 is equal to Current color=GRAY
      Current color=3 is equal to Current color=DARK
-----
init  Remaining_job_slots_of_FH_to_produce = 0
flow  Remaining_job_slots_of_FH_to_produce = +dt*FH_job_slots_orders
      -dt*FH_job_slots_completion
doc   Remaining_job_slots_of_FH_to_produce = Remaining job slots of product WG to produce.
-----
init  Remaining_job_slots_of_G_to_produce = 0
flow  Remaining_job_slots_of_G_to_produce = +dt*G_job_slots_orders
      -dt*G_job_slots_completion
doc   Remaining_job_slots_of_G_to_produce = Remaining job slots of product WG to produce.
-----
init  Remaining_job_slots_of_SD_to_produce = 0
flow  Remaining_job_slots_of_SD_to_produce = -dt*SD_job_slots_completion
      +dt*SD_job_slots_orders
doc   Remaining_job_slots_of_SD_to_produce = Remaining job slots of product WG to produce.
-----
init  Remaining_job_slots_of_WG_to_produce = 0
flow  Remaining_job_slots_of_WG_to_produce = -dt*WG_job_slots_completion
      +dt*WG_job_slots_orders
doc   Remaining_job_slots_of_WG_to_produce = Remaining job slots of product WG to produce.
-----
init  SD_WH_stock = 1100
flow  SD_WH_stock = +dt*SD_WH_inflow-dt*SD_WH_outflow
doc   SD_WH_stock = Quantity of SD stored in the warehouse.
-----
init  Slot_1 = 2
flow  Slot_1 = -dt*Outflow_silo_1+dt*R_1_IN+dt*R_2_1
doc   Slot_1 = This variable shows the color type of the semi product contained (if any)
      in Slot 1.
-----
init  Slot_10 = 0
flow  Slot_10 = +dt*R_10_IN+dt*R_11_10-dt*R_10_9

```

```

doc    Slot_10 = This variable shows the color type of the semi product contained (if any)
        in Slot 10.
-----
init   Slot_11 = 0
flow   Slot_11 = +dt*R_11_IN+dt*R_12_11-dt*R_11_10
doc    Slot_11 = This variable shows the color type of the semi product contained (if any)
        in Slot 11.
-----
init   Slot_12 = 0
flow   Slot_12 = +dt*R_12_IN-dt*R_12_11
doc    Slot_12 = This variable shows the color type of the semi product contained (if any)
        in Slot 12.
-----
init   Slot_13 = 0
flow   Slot_13 = -dt*Outflow_silo_2+dt*R_14_13+dt*R_13_IN
doc    Slot_13 = This variable shows the color type of the semi product contained (if any)
        in Slot 13.
-----
init   Slot_14 = 0
flow   Slot_14 = +dt*R_15_14-dt*R_14_13+dt*R_14_IN
doc    Slot_14 = This variable shows the color type of the semi product contained (if any)
        in Slot 14.
-----
init   Slot_15 = 0
flow   Slot_15 = +dt*R_16_15-dt*R_15_14+dt*R_15_IN
doc    Slot_15 = This variable shows the color type of the semi product contained (if any)
        in Slot 15.
-----
init   Slot_16 = 0
flow   Slot_16 = +dt*R_17_16-dt*R_16_15+dt*R_16_IN
doc    Slot_16 = This variable shows the color type of the semi product contained (if any)
        in Slot 16.
-----
init   Slot_17 = 0
flow   Slot_17 = +dt*R_18_17-dt*R_17_16+dt*R_17_IN
doc    Slot_17 = This variable shows the color type of the semi product contained (if any)
        in Slot 17.
-----
init   Slot_18 = 0
flow   Slot_18 = -dt*R_18_17+dt*R_18_IN+dt*R_19_18
doc    Slot_18 = This variable shows the color type of the semi product contained (if any)
        in Slot 18.
-----
init   Slot_19 = 0
flow   Slot_19 = +dt*R_19_IN+dt*R_20_19-dt*R_19_18
doc    Slot_19 = This variable shows the color type of the semi product contained (if any)
        in Slot 19.
-----
init   Slot_2 = 1
flow   Slot_2 = +dt*R_2_IN+dt*R_3_2-dt*R_2_1
doc    Slot_2 = This variable shows the color type of the semi product contained (if any)
        in Slot 2.
-----
init   Slot_20 = 0
flow   Slot_20 = +dt*R_21_20+dt*R_20_IN-dt*R_20_19
doc    Slot_20 = This variable shows the color type of the semi product contained (if any)
        in Slot 20.
-----
init   Slot_21 = 0
flow   Slot_21 = +dt*R_22_21-dt*R_21_20+dt*R_21_IN
doc    Slot_21 = This variable shows the color type of the semi product contained (if any)
        in Slot 21.
-----
init   Slot_22 = 0
flow   Slot_22 = +dt*R_23_22-dt*R_22_21+dt*R_22_IN
doc    Slot_22 = This variable shows the color type of the semi product contained (if any)
        in Slot 22.
-----
init   Slot_23 = 0
flow   Slot_23 = +dt*R_24_23-dt*R_23_22+dt*R_23_IN
doc    Slot_23 = This variable shows the color type of the semi product contained (if any)
        in Slot 23.
-----
init   Slot_24 = 0
flow   Slot_24 = -dt*R_24_23+dt*R_24_IN
doc    Slot_24 = This variable shows the color type of the semi product contained (if any)
        in Slot 24.

```

```

-----
init   Slot_25 = 0
flow   Slot_25 = -dt*Outflow_silo_3+dt*R_26_25+dt*R_25_IN
doc    Slot_25 = This variable shows the color type of the semi product contained (if any)
        in Slot 25.
-----
init   Slot_26 = 0
flow   Slot_26 = +dt*R_26_IN+dt*R_27_26-dt*R_26_25
doc    Slot_26 = This variable shows the color type of the semi product contained (if any)
        in Slot 26.
-----
init   Slot_27 = 0
flow   Slot_27 = +dt*R_27_IN+dt*R_28_27-dt*R_27_26
doc    Slot_27 = This variable shows the color type of the semi product contained (if any)
        in Slot 27.
-----
init   Slot_28 = 0
flow   Slot_28 = +dt*R_28_IN+dt*R_29_28-dt*R_28_27
doc    Slot_28 = This variable shows the color type of the semi product contained (if any)
        in Slot 28.
-----
init   Slot_29 = 0
flow   Slot_29 = +dt*R_30_29-dt*R_29_28+dt*R_29_IN
doc    Slot_29 = This variable shows the color type of the semi product contained (if any)
        in Slot 29.
-----
init   Slot_3 = 1
flow   Slot_3 = +dt*R_3_IN+dt*R_4_3-dt*R_3_2
doc    Slot_3 = This variable shows the color type of the semi product contained (if any)
        in Slot 3.
-----
init   Slot_30 = 0
flow   Slot_30 = +dt*R_30_IN+dt*R_31_30-dt*R_30_29
doc    Slot_30 = This variable shows the color type of the semi product contained (if any)
        in Slot 30.
-----
init   Slot_31 = 0
flow   Slot_31 = +dt*R_31_IN+dt*R_32_31-dt*R_31_30
doc    Slot_31 = This variable shows the color type of the semi product contained (if any)
        in Slot 31.
-----
init   Slot_32 = 0
flow   Slot_32 = +dt*R_32_IN+dt*R_33_32-dt*R_32_31
doc    Slot_32 = This variable shows the color type of the semi product contained (if any)
        in Slot 32.
-----
init   Slot_33 = 0
flow   Slot_33 = +dt*R_33_IN+dt*R_34_33-dt*R_33_32
doc    Slot_33 = This variable shows the color type of the semi product contained (if any)
        in Slot 33.
-----
init   Slot_34 = 0
flow   Slot_34 = +dt*R_34_IN+dt*R_35_34-dt*R_34_33
doc    Slot_34 = This variable shows the color type of the semi product contained (if any)
        in Slot 34.
-----
init   Slot_35 = 0
flow   Slot_35 = +dt*R_35_IN+dt*R_36_35-dt*R_35_34
doc    Slot_35 = This variable shows the color type of the semi product contained (if any)
        in Slot 35.
-----
init   Slot_36 = 0
flow   Slot_36 = +dt*R_36_IN-dt*R_36_35
doc    Slot_36 = This variable shows the color type of the semi product contained (if any)
        in Slot 36.
-----
init   Slot_37 = 3
flow   Slot_37 = -dt*R_37_38+dt*SSP_inflow
doc    Slot_37 = This variable represents the first slot of the SSP and shows the color of
        the contained (if any) semi-product.
-----
init   Slot_38 = 3
flow   Slot_38 = -dt*R_38_39+dt*R_37_38
doc    Slot_38 = This variable represents the second slot of the SSP and shows the color of
        the contained (if any) semi-product.
-----
init   Slot_39 = 3

```

```

flow  Slot_39 = -dt*R_39_40+dt*R_38_39
doc   Slot_39 = This variable represents the third slot of the SSP and shows the color of
       the contained (if any) semi-product.

-----
init  Slot_4 = 1
flow  Slot_4 = +dt*R_4_IN-dt*R_4_3+dt*R_5_4
doc   Slot_4 = This variable shows the color type of the semi product contained (if any)
       in Slot 4.

-----
init  Slot_40 = 3
flow  Slot_40 = -dt*R_40_41+dt*R_39_40
doc   Slot_40 = This variable represents the fourth slot of the SSP and shows the color of
       the contained (if any) semi-product.

-----
init  Slot_41 = 3
flow  Slot_41 = -dt*R_41_42+dt*R_40_41
doc   Slot_41 = This variable represents the fifth slot of the SSP and shows the color of
       the contained (if any) semi-product.

-----
init  Slot_42 = 3
flow  Slot_42 = -dt*R_42_out+dt*R_41_42
doc   Slot_42 = This variable represents the sixth slot of the SSP and shows the color of
       the contained (if any) semi-product.

-----
init  Slot_5 = 1
flow  Slot_5 = +dt*R_5_IN-dt*R_5_4+dt*R_6_5
doc   Slot_5 = This variable shows the color type of the semi product contained (if any)
       in Slot 5.

-----
init  Slot_6 = 1
flow  Slot_6 = +dt*R_6_IN+dt*R_7_6-dt*R_6_5
doc   Slot_6 = This variable shows the color type of the semi product contained (if any)
       in Slot 6.

-----
init  Slot_7 = 0
flow  Slot_7 = +dt*R_7_IN-dt*R_7_6+dt*R_8_7
doc   Slot_7 = This variable shows the color type of the semi product contained (if any)
       in Slot 7.

-----
init  Slot_8 = 0
flow  Slot_8 = +dt*R_8_IN+dt*R_9_8-dt*R_8_7
doc   Slot_8 = This variable shows the color type of the semi product contained (if any)
       in Slot 8.

-----
init  Slot_9 = 0
flow  Slot_9 = +dt*R_9_IN+dt*R_10_9-dt*R_9_8
doc   Slot_9 = This variable shows the color type of the semi product contained (if any)
       in Slot 9.

-----
init  SSP_Current_IV = 2
flow  SSP_Current_IV = -dt*R7+dt*R6
doc   SSP_Current_IV = This variable represents the third 4-hour period of an IV
       changeover (if any), when according to our hypothesis the IV is instantly changed.
       SSP_Current_IV=1 equals to LOW VISCOSITY
       SSP_Current_IV=2 equals to HIGH VISCOSITY

-----
init  Total_changeovers = 0
flow  Total_changeovers = +dt*Changeovers

-----
init  Total_lost_FH_BB_demand = 0
flow  Total_lost_FH_BB_demand = +dt*Lost_FH_BB_demand
doc   Total_lost_FH_BB_demand = This variable represents the total lost demand of grade FH
       in big bags.

-----
init  Total_lost_FH_BC_demand = 0
flow  Total_lost_FH_BC_demand = +dt*Lost_FH_BC_demand
doc   Total_lost_FH_BC_demand = This variable represents the total lost demand of grade FH
       in bulk containers.

-----
init  Total_lost_FH_ST_demand = 0
flow  Total_lost_FH_ST_demand = +dt*Lost_FH_ST_demand
doc   Total_lost_FH_ST_demand = This variable represents the total lost demand of grade FH
       in silo trucks.

-----
init  Total_lost_G_BB_demand = 0
flow  Total_lost_G_BB_demand = +dt*Lost_G_BB_demand

```



```

doc    Total_lost_G_BB_demand = This variable represents the total lost demand of grade G
      in big bags.
-----
init   Total_lost_G_BC_demand = 0
flow   Total_lost_G_BC_demand = +dt*Lost_G_BC_demand
doc    Total_lost_G_BC_demand = This variable represents the total lost demand of grade G
      in bulk containers.
-----
init   Total_lost_G_ST_demand = 0
flow   Total_lost_G_ST_demand = +dt*Lost_G_ST_demand
doc    Total_lost_G_ST_demand = This variable represents the total lost demand of grade G
      in silo trucks.
-----
init   Total_lost_SD_BB_demand = 0
flow   Total_lost_SD_BB_demand = +dt*Lost_SD_BB_demand
doc    Total_lost_SD_BB_demand = This variable represents the total lost demand of grade SD
      in big bags.
-----
init   Total_lost_SD_BC_demand = 0
flow   Total_lost_SD_BC_demand = +dt*Lost_SD_BC_demand
doc    Total_lost_SD_BC_demand = This variable represents the total lost demand of grade SD
      in bulk containers.
-----
init   Total_lost_SD_ST_demand = 0
flow   Total_lost_SD_ST_demand = +dt*Lost_SD_ST_demand
doc    Total_lost_SD_ST_demand = This variable represents the total lost demand of grade SD
      in silo trucks.
-----
init   Total_lost_WG_BB_demand = 0
flow   Total_lost_WG_BB_demand = +dt*Lost_WG_BB_demand
doc    Total_lost_WG_BB_demand = This variable represents the total lost demand of grade WG
      in big bags.
-----
init   Total_lost_WG_BC_demand = 0
flow   Total_lost_WG_BC_demand = +dt*Lost_WG_BC_demand
doc    Total_lost_WG_BC_demand = This variable represents the total lost demand of grade WG
      in bulk containers.
-----
init   Total_lost_WG_ST_demand = 0
flow   Total_lost_WG_ST_demand = +dt*Lost_WG_ST_demand
doc    Total_lost_WG_ST_demand = This variable represents the total lost demand of grade WG
      in silo trucks.
-----
init   Useless_products = 0
flow   Useless_products = +dt*Spilled_over_products+dt*Off_specification_production
doc    Useless_products = Products that can not be used to satisfy the demand for the types
      of final grades.
-----
init   Warehouse = 0
flow   Warehouse = -dt*WH_outflow+dt*WH_inflow
doc    Warehouse = Total occupied space of the warehouse at time t.
-----
init   WG_WH_stock = 300
flow   WG_WH_stock = -dt*WG_WH_outflow+dt*WG_WH_inflow
doc    WG_WH_stock = Quantity of WG stored in the warehouse.
-----
aux    Changeovers = Color_changeover+IV_changeover
-----
aux    FH_job_slots_completion = IF(Remaining_job_slots_of_FH_to_produce>0,
      IF(SSP_outflow=4, 1, 0),
      0)
doc    FH_job_slots_completion = The production of product FH has priority 3 in comparison
      to the other products because its total safety stock (LFSS+WH) is third-less than
      those of the others, resulting in third-less production time (SPT logic).
-----
aux    FH_job_slots_orders = Need_for_FH_in_job_lot
-----
aux    FH_WH_inflow = IF(LFSS1_grade=4, LFSS1_BB, 0) + IF(LFSS2_grade=4, LFSS2_BB, 0) +
      IF(LFSS3_grade=4, LFSS3_BB, 0) + IF(LFSS4_grade=4, LFSS4_BB, 0) +
      IF(LFSS5_grade=4, LFSS5_BB, 0) + IF(LFSS6_grade=4, LFSS6_BB, 0) +
      IF(LFSS7_grade=4, LFSS7_BB, 0) + IF(LFSS8_grade=4, LFSS8_BB, 0)
doc    FH_WH_inflow = Quantity of FH coming into the warehouse at time t.
-----
aux    FH_WH_outflow = MIN(FH_WH_stock,FH_BB_demand)
doc    FH_WH_outflow = Quantity of FH coming out from the warehouse at time t.
-----
aux    G_job_slots_completion = IF(Remaining_job_slots_of_G_to_produce>0,

```

```

                                IF(SSP_outflow=3, 1, 0),
                                0)
doc    G_job_slots_completion = The production of product G has priority 4 in comparison to
the other products because it is an undesired by-product.
aux    G_job_slots_orders = Need_for_G_in_job_lot
aux    G_WH_inflow = IF(LFSS1_grade=3, LFSS1_BB, 0) + IF(LFSS2_grade=3, LFSS2_BB, 0) +
IF(LFSS3_grade=3, LFSS3_BB, 0) + IF(LFSS4_grade=3, LFSS4_BB, 0) +
IF(LFSS5_grade=3, LFSS5_BB, 0) + IF(LFSS6_grade=3, LFSS6_BB, 0) +
IF(LFSS7_grade=3, LFSS7_BB, 0) + IF(LFSS8_grade=3, LFSS8_BB, 0)
doc    G_WH_inflow = Quantity of G coming into the warehouse at time t.
aux    G_WH_outflow = MIN(G_WH_stock,G_BB_demand)
doc    G_WH_outflow = Quantity of G coming out from the warehouse at time t.
aux    LFSS1_BB = IF(LFSS1_grade=1,
MIN((LFSS1-LFSS1_ST-LFSS1_BC),S1_WG_WH_dem,Ubb*(1-(LFSS1_ST/Ust)-
(LFSS1_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB-LFSS3_BB-
LFSS2_BB)),
IF(LFSS1_grade=2,
MIN((LFSS1-LFSS1_ST-LFSS1_BC),S1_SD_WH_dem,Ubb*(1-(LFSS1_ST/Ust)-
(LFSS1_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB-LFSS3_BB-
LFSS2_BB)),
IF(LFSS1_grade=3,
MIN((LFSS1-LFSS1_ST-LFSS1_BC),S1_G_WH_dem,Ubb*(1-(LFSS1_ST/Ust)-
(LFSS1_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB-LFSS3_BB-
LFSS2_BB)),
IF(LFSS1_grade=4,
MIN((LFSS1-LFSS1_ST-LFSS1_BC),S1_FH_WH_dem,Ubb*(1-(LFSS1_ST/Ust)-
(LFSS1_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB-LFSS3_BB-
LFSS2_BB)),
0) ) ) )
doc    LFSS1_BB = Outflow of LFSS1 in big bags each 4-hour period.
The big bags demand-type has priority 3 and is satisfied last in regard to the other
two demand-types (bulk container and silo trucks).
aux    LFSS1_BC = IF(LFSS1_grade=1,
MIN((LFSS1-LFSS1_ST),S1_WG_BC_dem,Ubc*(1-(LFSS1_ST/Ust))),
IF(LFSS1_grade=2,
MIN((LFSS1-LFSS1_ST),S1_SD_BC_dem,Ubc*(1-(LFSS1_ST/Ust))),
IF(LFSS1_grade=3,
MIN((LFSS1-LFSS1_ST),S1_G_BC_dem,Ubc*(1-(LFSS1_ST/Ust))),
IF(LFSS1_grade=4,
MIN((LFSS1-LFSS1_ST),S1_FH_BC_dem,Ubc*(1-(LFSS1_ST/Ust))),0) ) ) )
doc    LFSS1_BC = Outflow of LFSS1 in bulk containers each 4-hour period.
The bulk containers demand-type has priority 2 and is satisfied second in regard to
the other two demand-types (big bags and silo trucks).
aux    LFSS1_inflow = IF(SSP_outflow=LFSS1_grade OR LFSS1<=Smin,
IF(SSP_outflow=1,
IF((Opt_WG_LFSS_capacity=500),
0,
IF(Opt_WG_LFSS_capacity=(Smax-LFSS1),
33.33,
0)),
0) +
IF(SSP_outflow=2,
IF((Opt_SD_LFSS_capacity=500),
0,
IF(Opt_SD_LFSS_capacity=(Smax-LFSS1),
33.33,
0)),
0) +
IF(SSP_outflow=3,
IF((Opt_G_LFSS_capacity=500),
0,
IF(Opt_G_LFSS_capacity=(Smax-LFSS1),
33.33,
0)),
0) +
IF(SSP_outflow=4,
IF((Opt_FH_LFSS_capacity=500),

```

```

0,
IF(Opt_FH_LFSS_capacity=(Smax-LFSS1),
33.33,
0)),
0),
0)
doc LFSS1_inflow = Inflow of LFSS1 in each 4-hour period. The model uses a priority rule
where an LFSS silo can receive a grade only if all the previous LFSS silos (counting
from 1 to 8) are unappropriate for receiving the type and quantity of grade coming
out from SSP.
-----
aux LFSS1_ST = IF(LFSS1_grade=1,
MIN(LFSS1,S1_WG_ST_dem,Ust),
IF(LFSS1_grade=2,
MIN(LFSS1,S1_SD_ST_dem,Ust),
IF(LFSS1_grade=3,
MIN(LFSS1,S1_G_ST_dem,Ust),
IF(LFSS1_grade=4,
MIN(LFSS1,S1_FH_ST_dem,Ust),0) ) ) ) )
doc LFSS1_ST = Outflow of LFSS1 in silo trucks each 4-hour period.
The silo trucks demand-type has priority 1 and is satisfied first in regard to the other
two demand-types (bulk container and big bags).
-----
aux LFSS2_BB = IF(LFSS2_grade=1,
MIN((LFSS2-LFSS2_ST-LFSS2_BC),S2_WG_WH_dem,Ubb*(1-(LFSS2_ST/Ust)-
(LFSS2_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB-
LFSS3_BB)),
IF(LFSS2_grade=2,
MIN((LFSS2-LFSS2_ST-LFSS2_BC),S2_SD_WH_dem,Ubb*(1-(LFSS2_ST/Ust)-
(LFSS2_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB-
LFSS3_BB)),
IF(LFSS2_grade=3,
MIN((LFSS2-LFSS2_ST-LFSS2_BC),S2_G_WH_dem,Ubb*(1-(LFSS2_ST/Ust)-
(LFSS2_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB-
LFSS3_BB)),
IF(LFSS2_grade=4,
MIN((LFSS2-LFSS2_ST-LFSS2_BC),S2_FH_WH_dem,Ubb*(1-(LFSS2_ST/Ust)-
(LFSS2_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB-
LFSS3_BB)),
0) ) ) )
doc LFSS2_BB = Outflow of LFSS2 in big bags each 4-hour period.
The big bags demand-type has priority 3 and is satisfied last in regard to the other
two demand-types (bulk container and silo trucks).
-----
aux LFSS2_BC = IF(LFSS2_grade=1,
MIN((LFSS2-LFSS2_ST),S2_WG_BC_dem,Ubc*(1-(LFSS2_ST/Ust))),
IF(LFSS2_grade=2,
MIN((LFSS2-LFSS2_ST),S2_SD_BC_dem,Ubc*(1-(LFSS2_ST/Ust))),
IF(LFSS2_grade=3,
MIN((LFSS2-LFSS2_ST),S2_G_BC_dem,Ubc*(1-(LFSS2_ST/Ust))),
IF(LFSS2_grade=4,
MIN((LFSS2-LFSS2_ST),S2_FH_BC_dem,Ubc*(1-(LFSS2_ST/Ust))),0) ) ) )
doc LFSS2_BC = Outflow of LFSS2 in bulk containers each 4-hour period.
The bulk containers demand-type has priority 2 and is satisfied second in
regard to the other two demand-types (big bags and silo trucks).
-----
aux LFSS2_inflow = IF(LFSS1_inflow=0 AND (SSP_outflow=LFSS2_grade OR LFSS2<=Smin),
IF(SSP_outflow=1, IF((Opt_WG_LFSS_capacity=500), 0,
IF(Opt_WG_LFSS_capacity=(Smax-LFSS2), 33.33, 0)), 0) +
IF(SSP_outflow=2, IF((Opt_SD_LFSS_capacity=500), 0,
IF(Opt_SD_LFSS_capacity=(Smax-LFSS2), 33.33, 0)), 0) +
IF(SSP_outflow=3, IF((Opt_G_LFSS_capacity=500), 0,
IF(Opt_G_LFSS_capacity=(Smax-LFSS2), 33.33, 0)), 0) +
IF(SSP_outflow=4, IF((Opt_FH_LFSS_capacity=500), 0,
IF(Opt_FH_LFSS_capacity=(Smax-LFSS2), 33.33, 0)), 0), 0)
doc LFSS2_inflow = Inflow of LFSS2 in each 4-hour period. The model uses a priority rule
where an LFSS silo can receive a grade only if all the previous LFSS silos (counting
from 1 to 8) are unappropriate for receiving the type and quantity of grade coming
out from SSP.
-----
aux LFSS2_ST = IF(LFSS2_grade=1,
MIN(LFSS2,S2_WG_ST_dem,Ust),
IF(LFSS2_grade=2,

```

```

        MIN(LFSS2,S2_SD_ST_dem,Ust),
        IF(LFSS2_grade=3,
            MIN(LFSS2,S2_G_ST_dem,Ust),
            IF(LFSS2_grade=4,
                MIN(LFSS2,S2_FH_ST_dem,Ust),0) ) ) )
doc    LFSS2_ST = Outflow of LFSS2 in silo trucks each 4-hour period.
The silo trucks demand-type has priority 1 and is satisfied first in regard to the other
two demand-types (bulk container and big bags).

-----
aux    LFSS3_BB = IF(LFSS3_grade=1,
                    MIN((LFSS3-LFSS3_ST-LFSS3_BC),S3_WG_WH_dem,Ubb*(1-(LFSS3_ST/Ust))-
                        (LFSS3_BC/Ubc)),
                    (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB)),
        IF(LFSS3_grade=2,
            MIN((LFSS3-LFSS3_ST-LFSS3_BC),S3_SD_WH_dem,Ubb*(1-(LFSS3_ST/Ust))-
                (LFSS3_BC/Ubc)),
            (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB)),
        IF(LFSS3_grade=3,
            MIN((LFSS3-LFSS3_ST-LFSS3_BC),S3_G_WH_dem,Ubb*(1-(LFSS3_ST/Ust))-
                (LFSS3_BC/Ubc)),
            (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB)),
        IF(LFSS3_grade=4,
            MIN((LFSS3-LFSS3_ST-LFSS3_BC),S3_FH_WH_dem,Ubb*(1-(LFSS3_ST/Ust))-
                (LFSS3_BC/Ubc)),
            (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB-LFSS4_BB)),
        0) ) ) )
doc    LFSS3_BB = Outflow of LFSS3 in big bags each 4-hour period.
The big bags demand-type has priority 3 and is satisfied last in regard to the other
two demand-types (bulk container and silo trucks).

-----
aux    LFSS3_BC = IF(LFSS3_grade=1,
                    MIN((LFSS3-LFSS3_ST),S3_WG_BC_dem,Ubc*(1-(LFSS3_ST/Ust))),
                    IF(LFSS3_grade=2,
                        MIN((LFSS3-LFSS3_ST),S3_SD_BC_dem,Ubc*(1-(LFSS3_ST/Ust))),
                        IF(LFSS3_grade=3,
                            MIN((LFSS3-LFSS3_ST),S3_G_BC_dem,Ubc*(1-(LFSS3_ST/Ust))),
                            IF(LFSS3_grade=4,
                                MIN((LFSS3-LFSS3_ST),S3_FH_BC_dem,Ubc*(1-(LFSS3_ST/Ust))),0) ) ) ) )
doc    LFSS3_BC = Outflow of LFSS3 in bulk containers each 4-hour period.
The bulk containers demand-type has priority 2 and is satisfied second in regard to
the other two demand-types (big bags and silo trucks).

-----
aux    LFSS3_inflow = IF((LFSS1_inflow=0) AND (LFSS2_inflow=0) AND (SSP_outflow=LFSS3_grade
OR LFSS3<=Smin),
                    IF(SSP_outflow=1, IF((Opt_WG_LFSS_capacity=500), 0,
                    IF(Opt_WG_LFSS_capacity=(Smax-LFSS3), 33.33, 0)), 0) +
                    IF(SSP_outflow=2, IF((Opt_SD_LFSS_capacity=500), 0,
                    IF(Opt_SD_LFSS_capacity=(Smax-LFSS3), 33.33, 0)) , 0) +
                    IF(SSP_outflow=3, IF((Opt_G_LFSS_capacity=500), 0,
                    IF(Opt_G_LFSS_capacity=(Smax-LFSS3), 33.33, 0)) , 0) +
                    IF(SSP_outflow=4, IF((Opt_FH_LFSS_capacity=500), 0,
                    IF(Opt_FH_LFSS_capacity=(Smax-LFSS3), 33.33, 0)) , 0), 0)
doc    LFSS3_inflow = Inflow of LFSS3 in each 4-hour period. The model uses a priority rule
where an LFSS silo can receive a grade only if all the previous LFSS silos (counting
from 1 to 8) are inappropriate for receiving the type and quantity of grade coming
out from SSP.

-----
aux    LFSS3_ST = IF(LFSS3_grade=1,
                    MIN(LFSS3,S3_WG_ST_dem,Ust),
                    IF(LFSS3_grade=2,
                        MIN(LFSS3,S3_SD_ST_dem,Ust),
                        IF(LFSS3_grade=3,
                            MIN(LFSS3,S3_G_ST_dem,Ust),
                            IF(LFSS3_grade=4,
                                MIN(LFSS3,S3_FH_ST_dem,Ust),0) ) ) ) )
doc    LFSS3_ST = Outflow of LFSS3 in silo trucks each 4-hour period.
The silo trucks demand-type has priority 1 and is satisfied first in regard to the other
two demand-types (bulk container and big bags).

-----
aux    LFSS4_BB = IF(LFSS4_grade=1,
                    MIN((LFSS4-LFSS4_ST-LFSS4_BC),S4_WG_WH_dem,Ubb*(1-(LFSS4_ST/Ust))-
                        (LFSS4_BC/Ubc)),
                    (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB)),
        IF(LFSS4_grade=2,
            MIN((LFSS4-LFSS4_ST-LFSS4_BC),S4_SD_WH_dem,Ubb*(1-(LFSS4_ST/Ust))-
                (LFSS4_BC/Ubc)),
            (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB)),

```

```

IF(LFSS4_grade=3,
  MIN((LFSS4-LFSS4_ST-LFSS4_BC),S4_G_WH_dem,Ubb*(1-(LFSS4_ST/Ust))-
(LFSS4_BC/Ubc)),
  (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB)),
IF(LFSS4_grade=4,
  MIN((LFSS4-LFSS4_ST-LFSS4_BC),S4_FH_WH_dem,Ubb*(1-(LFSS4_ST/Ust))-
(LFSS4_BC/Ubc)),
  (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB-LFSS5_BB)),
0) ) ) )
doc LFSS4_BB = Outflow of LFSS4 in big bags each 4-hour period.
The big bags demand-type has priority 3 and is satisfied last in regard to the other two
demand-types (bulk container and silo trucks).
-----
aux LFSS4_BC = IF(LFSS4_grade=1,
  MIN((LFSS4-LFSS4_ST),S4_WG_BC_dem,Ubc*(1-(LFSS4_ST/Ust))),
  IF(LFSS4_grade=2,
    MIN((LFSS4-LFSS4_ST),S4_SD_BC_dem,Ubc*(1-(LFSS4_ST/Ust))),
    IF(LFSS4_grade=3,
      MIN((LFSS4-LFSS4_ST),S4_G_BC_dem,Ubc*(1-(LFSS4_ST/Ust))),
      IF(LFSS4_grade=4,
        MIN((LFSS4-LFSS4_ST),S4_FH_BC_dem,Ubc*(1-(LFSS4_ST/Ust))),0) ) ) ) )
doc LFSS4_BC = Outflow of LFSS4 in bulk containers each 4-hour period.
The bulk containers demand-type has priority 2 and is satisfied second in regard to
the other two demand-types (big bags and silo trucks).
-----
aux LFSS4_inflow = IF((LFSS1_inflow=0) AND (LFSS2_inflow=0) AND (LFSS3_inflow=0) AND
(SSP_outflow=LFSS4_grade OR LFSS4<=Smin),
  IF(SSP_outflow=1, IF((Opt_WG_LFSS_capacity=500), 0,
  IF(Opt_WG_LFSS_capacity=(Smax-LFSS4), 33.33, 0)), 0) +
  IF(SSP_outflow=2, IF((Opt_SD_LFSS_capacity=500), 0,
  IF(Opt_SD_LFSS_capacity=(Smax-LFSS4), 33.33, 0)), 0) +
  IF(SSP_outflow=3, IF((Opt_G_LFSS_capacity=500), 0,
  IF(Opt_G_LFSS_capacity=(Smax-LFSS4), 33.33, 0)), 0) +
  IF(SSP_outflow=4, IF((Opt_FH_LFSS_capacity=500), 0,
  IF(Opt_FH_LFSS_capacity=(Smax-LFSS4), 33.33, 0)), 0), 0)
doc LFSS4_inflow = Inflow of LFSS4 in each 4-hour period. The model uses a priority rule
where an LFSS silo can receive a grade only if all the previous LFSS silos (counting
from 1 to 8) are unappropriate for receiving the type and quantity of grade coming
out from SSP.
-----
aux LFSS4_ST = IF(LFSS4_grade=1,
  MIN(LFSS4,S4_WG_ST_dem,Ust),
  IF(LFSS4_grade=2,
    MIN(LFSS4,S4_SD_ST_dem,Ust),
    IF(LFSS4_grade=3,
      MIN(LFSS4,S4_G_ST_dem,Ust),
      IF(LFSS4_grade=4,
        MIN(LFSS4,S4_FH_ST_dem,Ust),0) ) ) ) )
doc LFSS4_ST = Outflow of LFSS4 in silo trucks each 4-hour period.
The silo trucks demand-type has priority 1 and is satisfied first in regard to the
other two demand-types (bulk container and big bags).
-----
aux LFSS5_BB = IF(LFSS5_grade=1,
  MIN((LFSS5-LFSS5_ST-LFSS5_BC),S5_WG_WH_dem,Ubb*(1-(LFSS5_ST/Ust))-
(LFSS5_BC/Ubc)),
  (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB)),
IF(LFSS5_grade=2,
  MIN((LFSS5-LFSS5_ST-LFSS5_BC),S5_SD_WH_dem,Ubb*(1-(LFSS5_ST/Ust))-
(LFSS5_BC/Ubc)),
  (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB)),
IF(LFSS5_grade=3,
  MIN((LFSS5-LFSS5_ST-LFSS5_BC),S5_G_WH_dem,Ubb*(1-(LFSS5_ST/Ust))-
(LFSS5_BC/Ubc)),
  (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB)),
IF(LFSS5_grade=4,
  MIN((LFSS5-LFSS5_ST-LFSS5_BC),S5_FH_WH_dem,Ubb*(1-(LFSS5_ST/Ust))-
(LFSS5_BC/Ubc)),
  (WH_free_space-LFSS8_BB-LFSS7_BB-LFSS6_BB)),
0) ) ) )
doc LFSS5_BB = Outflow of LFSS5 in big bags each 4-hour period.
The big bags demand-type has priority 3 and is satisfied last in regard to the other
two demand-types (bulk container and silo trucks).
-----
aux LFSS5_BC = IF(LFSS5_grade=1,
  MIN((LFSS5-LFSS5_ST),S5_WG_BC_dem,Ubc*(1-(LFSS5_ST/Ust))),
  IF(LFSS5_grade=2,
    MIN((LFSS5-LFSS5_ST),S5_SD_BC_dem,Ubc*(1-(LFSS5_ST/Ust))),

```



```

IF(LFSS5_grade=3,
  MIN((LFSS5-LFSS5_ST),S5_G_BC_dem,Ubc*(1-(LFSS5_ST/Ust))),
IF(LFSS5_grade=4,
  MIN((LFSS5-LFSS5_ST),S5_FH_BC_dem,Ubc*(1-(LFSS5_ST/Ust))),0) ) )
)
doc LFSS5_BC = Outflow of LFSS5 in bulk containers each 4-hour period.
The bulk containers demand-type has priority 2 and is satisfied second in regard to
the other two demand-types (big bags and silo trucks).

aux LFSS5_inflow = IF((LFSS1_inflow=0) AND (LFSS2_inflow=0) AND (LFSS3_inflow=0) AND
(LFSS4_inflow=0) AND (SSP_outflow=LFSS5_grade OR LFSS5<=Smin),
IF(SSP_outflow=1, IF((Opt_WG_LFSS_capacity=500), 0, IF(Opt_WG_LFSS_capacity=(Smax-
LFSS5), 33.33, 0)), 0) +
IF(SSP_outflow=2, IF((Opt_SD_LFSS_capacity=500), 0,
IF(Opt_SD_LFSS_capacity=(Smax-LFSS5), 33.33, 0)), 0) +
IF(SSP_outflow=3, IF((Opt_G_LFSS_capacity=500), 0,
IF(Opt_G_LFSS_capacity=(Smax-LFSS5), 33.33, 0)), 0) +
IF(SSP_outflow=4, IF((Opt_FH_LFSS_capacity=500), 0,
IF(Opt_FH_LFSS_capacity=(Smax-LFSS5), 33.33, 0)), 0), 0)
doc LFSS5_inflow = Inflow of LFSS5 in each 4-hour period. The model uses a priority rule
where an LFSS silo can receive a grade only if all the previous LFSS silos (counting
from 1 to 8) are inappropriate for receiving the type and quantity of grade coming
out from SSP.

aux LFSS5_ST = IF(LFSS5_grade=1,
  MIN(LFSS5,S5_WG_ST_dem,Ust),
IF(LFSS5_grade=2,
  MIN(LFSS5,S5_SD_ST_dem,Ust),
IF(LFSS5_grade=3,
  MIN(LFSS5,S5_G_ST_dem,Ust),
IF(LFSS5_grade=4,
  MIN(LFSS5,S5_FH_ST_dem,Ust),0) ) ) )
doc LFSS5_ST = Outflow of LFSS5 in silo trucks each 4-hour period.
The silo trucks demand-type has priority 1 and is satisfied first in regard to the
other two demand-types (bulk container and big bags).

aux LFSS6_BB = IF(LFSS6_grade=1,
  MIN((LFSS6-LFSS6_ST-LFSS6_BC),S6_WG_WH_dem,Ubb*(1-(LFSS6_ST/Ust)-
(LFSS6_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB)),
IF(LFSS6_grade=2,
  MIN((LFSS6-LFSS6_ST-LFSS6_BC),S6_SD_WH_dem,Ubb*(1-(LFSS6_ST/Ust)-
(LFSS6_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB)),
IF(LFSS6_grade=3,
  MIN((LFSS6-LFSS6_ST-LFSS6_BC),S6_G_WH_dem,Ubb*(1-(LFSS6_ST/Ust)-
(LFSS6_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB)),
IF(LFSS6_grade=4,
  MIN((LFSS6-LFSS6_ST-LFSS6_BC),S6_FH_WH_dem,Ubb*(1-(LFSS6_ST/Ust)-
(LFSS6_BC/Ubc)),
(WH_free_space-LFSS8_BB-LFSS7_BB)),
0) ) ) )
doc LFSS6_BB = Outflow of LFSS6 in big bags each 4-hour period.
The big bags demand-type has priority 3 and is satisfied last in regard to the other
two demand-types (bulk container and silo trucks).

aux LFSS6_BC = IF(LFSS6_grade=1,
  MIN((LFSS6-LFSS6_ST),S6_WG_BC_dem,Ubc*(1-(LFSS6_ST/Ust))),
IF(LFSS6_grade=2,
  MIN((LFSS6-LFSS6_ST),S6_SD_BC_dem,Ubc*(1-(LFSS6_ST/Ust))),
IF(LFSS6_grade=3,
  MIN((LFSS6-LFSS6_ST),S6_G_BC_dem,Ubc*(1-(LFSS6_ST/Ust))),
IF(LFSS6_grade=4,
  MIN((LFSS6-LFSS6_ST),S6_FH_BC_dem,Ubc*(1-(LFSS6_ST/Ust))),0) ) ) )
doc LFSS6_BC = Outflow of LFSS6 in bulk containers each 4-hour period.
The bulk containers demand-type has priority 2 and is satisfied second in regard to
the other two demand-types (big bags and silo trucks).

aux LFSS6_inflow = IF((LFSS1_inflow=0) AND (LFSS2_inflow=0) AND (LFSS3_inflow=0) AND
(LFSS4_inflow=0) AND (LFSS5_inflow=0) AND (SSP_outflow=LFSS6_grade OR
LFSS6<=Smin),
IF(SSP_outflow=1, IF((Opt_WG_LFSS_capacity=500), 0, IF(Opt_WG_LFSS_capacity=(Smax-LFSS6),
33.33, 0)), 0) +
IF(SSP_outflow=2, IF((Opt_SD_LFSS_capacity=500), 0, IF(Opt_SD_LFSS_capacity=(Smax-
LFSS6), 33.33, 0)), 0) +

```

```

IF(SSP_outflow=3, IF((Opt_G_LFSS_capacity=500), 0, IF(Opt_G_LFSS_capacity=(Smax-
LFSS6), 33.33, 0)), 0) +
IF(SSP_outflow=4, IF((Opt_FH_LFSS_capacity=500), 0, IF(Opt_FH_LFSS_capacity=(Smax-
LFSS6), 33.33, 0)), 0), 0)
doc LFSS6_inflow = Inflow of LFSS6 in each 4-hour period. The model uses a priority rule
where an LFSS silo can receive a grade only if all the previous LFSS silos (counting
from 1 to 8) are unappropriate for receiving the type and quantity of grade coming
out from SSP.
-----
aux LFSS6_ST = IF(LFSS6_grade=1,
MIN(LFSS6,S6_WG_ST_dem,Ust),
IF(LFSS6_grade=2,
MIN(LFSS6,S6_SD_ST_dem,Ust),
IF(LFSS6_grade=3,
MIN(LFSS6,S6_G_ST_dem,Ust),
IF(LFSS6_grade=4,
MIN(LFSS6,S6_FH_ST_dem,Ust),0) ) ) ) )
doc LFSS6_ST = Outflow of LFSS6 in silo trucks each 4-hour period.
The silo trucks demand-type has priority 1 and is satisfied first in regard to the
other two demand-types (bulk container and big bags).
-----
aux LFSS7_BB = IF(LFSS7_grade=1,
MIN((LFSS7-LFSS7_ST-LFSS7_BC),S7_WG_WH_dem,Ubb*(1-(LFSS7_ST/Ust)-
(LFSS7_BC/Ubc)),
(WH_free_space-LFSS8_BB)),
IF(LFSS7_grade=2,
MIN((LFSS7-LFSS7_ST-LFSS7_BC),S7_SD_WH_dem,Ubb*(1-(LFSS7_ST/Ust)-
(LFSS7_BC/Ubc)),
(WH_free_space-LFSS8_BB)),
IF(LFSS7_grade=3,
MIN((LFSS7-LFSS7_ST-LFSS7_BC),S7_G_WH_dem,Ubb*(1-(LFSS7_ST/Ust)-
(LFSS7_BC/Ubc)),
(WH_free_space-LFSS8_BB)),
IF(LFSS7_grade=4,
MIN((LFSS7-LFSS7_ST-LFSS7_BC),S7_FH_WH_dem,Ubb*(1-(LFSS7_ST/Ust)-
(LFSS7_BC/Ubc)),
(WH_free_space-LFSS8_BB)),
0) ) ) ) )
doc LFSS7_BB = Outflow of LFSS7 in big bags each 4-hour period.
The big bags demand-type has priority 3 and is satisfied last in regard to the other
two demand-types (bulk container and silo trucks).
-----
aux LFSS7_BC = IF(LFSS7_grade=1,
MIN((LFSS7-LFSS7_ST),S7_WG_BC_dem,Ubc*(1-(LFSS7_ST/Ust))),
IF(LFSS7_grade=2,
MIN((LFSS7-LFSS7_ST),S7_SD_BC_dem,Ubc*(1-(LFSS7_ST/Ust))),
IF(LFSS7_grade=3,
MIN((LFSS7-LFSS7_ST),S7_G_BC_dem,Ubc*(1-(LFSS7_ST/Ust))),
IF(LFSS7_grade=4,
MIN((LFSS7-LFSS7_ST),S7_FH_BC_dem,Ubc*(1-(LFSS7_ST/Ust))),0) ) ) ) )
doc LFSS7_BC = Outflow of LFSS7 in bulk containers each 4-hour period.
The bulk containers demand-type has priority 2 and is satisfied second in regard to
the other two demand-types (big bags and silo trucks).
-----
aux LFSS7_inflow = IF((LFSS1_inflow=0) AND (LFSS2_inflow=0) AND (LFSS3_inflow=0) AND
(LFSS4_inflow=0) AND (LFSS5_inflow=0) AND (LFSS6_inflow=0) AND
(SSP_outflow=LFSS7_grade OR LFSS7<=Smin),
IF(SSP_outflow=1, IF((Opt_WG_LFSS_capacity=500), 0,
IF(Opt_WG_LFSS_capacity=(Smax-LFSS7), 33.33, 0)), 0) +
IF(SSP_outflow=2, IF((Opt_SD_LFSS_capacity=500), 0,
IF(Opt_SD_LFSS_capacity=(Smax-LFSS7), 33.33, 0)), 0) +
IF(SSP_outflow=3, IF((Opt_G_LFSS_capacity=500), 0,
IF(Opt_G_LFSS_capacity=(Smax-LFSS7), 33.33, 0)), 0) +
IF(SSP_outflow=4, IF((Opt_FH_LFSS_capacity=500), 0,
IF(Opt_FH_LFSS_capacity=(Smax-LFSS7), 33.33, 0)), 0), 0)
doc LFSS7_inflow = Inflow of LFSS7 in each 4-hour period. The model uses a priority rule
where an LFSS silo can receive a grade only if all the previous LFSS silos (counting
from 1 to 8) are unappropriate for receiving the type and quantity of grade coming
out from SSP.
-----
aux LFSS7_ST = IF(LFSS7_grade=1,
MIN(LFSS7,S7_WG_ST_dem,Ust),
IF(LFSS7_grade=2,
MIN(LFSS7,S7_SD_ST_dem,Ust),
IF(LFSS7_grade=3,
MIN(LFSS7,S7_G_ST_dem,Ust),
IF(LFSS7_grade=4,

```

```

MIN(LFSS7,S7_FH_ST_dem,Ust),0) ) ) )
doc LFSS7_ST = Outflow of LFSS7 in silo trucks each 4-hour period.
The silo trucks demand-type has priority 1 and is satisfied first in regard to the
other two demand-types (bulk container and big bags).
-----
aux LFSS8_BB = IF(LFSS8_grade=1,
MIN((LFSS8-LFSS8_ST-LFSS8_BC),S8_WG_WH_dem,Ubb*(1-(LFSS8_ST/Ust))-
(LFSS8_BC/Ubc)),
WH_free_space),
IF(LFSS8_grade=2,
MIN((LFSS8-LFSS8_ST-LFSS8_BC),S8_SD_WH_dem,Ubb*(1-(LFSS8_ST/Ust))-
(LFSS8_BC/Ubc)),
WH_free_space),
IF(LFSS8_grade=3,
MIN((LFSS8-LFSS8_ST-LFSS8_BC),S8_G_WH_dem,Ubb*(1-(LFSS8_ST/Ust))-
(LFSS8_BC/Ubc)),
WH_free_space),
IF(LFSS8_grade=4,
MIN((LFSS8-LFSS8_ST-LFSS8_BC),S8_FH_WH_dem,Ubb*(1-(LFSS8_ST/Ust))-
(LFSS8_BC/Ubc)),
WH_free_space),
0) ) ) )
doc LFSS8_BB = Outflow of LFSS8 in big bags each 4-hour period.
The big bags demand-type has priority 3 and is satisfied last in regard to the other
two demand-types (bulk container and silo trucks).
-----
aux LFSS8_BC = IF(LFSS8_grade=1,
MIN((LFSS8-LFSS8_ST),WG_BC_demand,Ubc*(1-(LFSS8_ST/Ust))),
IF(LFSS8_grade=2,
MIN((LFSS8-LFSS8_ST),SD_BC_demand,Ubc*(1-(LFSS8_ST/Ust))),
IF(LFSS8_grade=3,
MIN((LFSS8-LFSS8_ST),G_BC_demand,Ubc*(1-(LFSS8_ST/Ust))),
IF(LFSS8_grade=4,
MIN((LFSS8-LFSS8_ST),FH_BC_demand,Ubc*(1-(LFSS8_ST/Ust))),0) ) ) )
doc LFSS8_BC = Outflow of LFSS8 in bulk containers each 4-hour period.
The bulk containers demand-type has priority 2 and is satisfied second in regard to
the other two demand-types (big bags and silo trucks).
-----
aux LFSS8_inflow = IF((LFSS1_inflow=0) AND (LFSS2_inflow=0) AND (LFSS3_inflow=0) AND
(LFSS4_inflow=0) AND (LFSS5_inflow=0) AND (LFSS6_inflow=0) AND (LFSS7_inflow=0)
AND (SSP_outflow=LFSS8_grade OR LFSS8<=Smin),
IF(SSP_outflow=1, IF((Opt_WG_LFSS_capacity=500), 0,
IF(Opt_WG_LFSS_capacity=(Smax-LFSS8), 33.33, 0)), 0) +
IF(SSP_outflow=2, IF((Opt_SD_LFSS_capacity=500), 0,
IF(Opt_SD_LFSS_capacity=(Smax-LFSS8), 33.33, 0)), 0) +
IF(SSP_outflow=3, IF((Opt_G_LFSS_capacity=500), 0,
IF(Opt_G_LFSS_capacity=(Smax-LFSS8), 33.33, 0)), 0) +
IF(SSP_outflow=4, IF((Opt_FH_LFSS_capacity=500), 0,
IF(Opt_FH_LFSS_capacity=(Smax-LFSS8), 33.33, 0)), 0), 0)
doc LFSS8_inflow = Inflow of LFSS8 in each 4-hour period. The model uses a priority rule
where an LFSS silo can receive a grade only if all the previous LFSS silos (counting
from 1 to 8) are inappropriate for receiving the type and quantity of grade coming
out from SSP.
-----
aux LFSS8_ST = IF(LFSS8_grade=1,
MIN(LFSS8,WG_ST_demand,Ust),
IF(LFSS8_grade=2,
MIN(LFSS8,SD_ST_demand,Ust),
IF(LFSS8_grade=3,
MIN(LFSS8,G_ST_demand,Ust),
IF(LFSS8_grade=4,
MIN(LFSS8,FH_ST_demand,Ust),0) ) ) )
doc LFSS8_ST = Outflow of LFSS8 in silo trucks each 4-hour period.
The silo trucks demand-type has priority 1 and is satisfied first in regard to the
other two demand-types (bulk container and big bags).
-----
aux Lost_FH_BB_demand = MAX(FH_BB_demand-FH_WH_outflow, 0)
doc Lost_FH_BB_demand = This variable represents the lost demand of grade FH in big bags
at time t.
-----
aux Lost_FH_BC_demand = MAX(FH_BC_demand- IF(LFSS1_grade=4, LFSS1_BC, 0) -
IF(LFSS2_grade=4, LFSS2_BC, 0) - IF(LFSS3_grade=4, LFSS3_BC, 0) - IF(LFSS4_grade=4,
LFSS4_BC, 0) - IF(LFSS5_grade=4, LFSS5_BC, 0) - IF(LFSS6_grade=4, LFSS6_BC, 0) -
IF(LFSS7_grade=4, LFSS7_BC, 0) - IF(LFSS8_grade=4, LFSS8_BC, 0) , 0)
doc Lost_FH_BC_demand = This variable represents the lost demand of grade FH in bulk
containers at time t.
-----

```

```

aux   Lost_FH_ST_demand = MAX(FH_ST_demand- IF(LFSS1_grade=4, LFSS1_ST, 0) -
IF(LFSS2_grade=4, LFSS2_ST, 0)- IF(LFSS3_grade=4, LFSS3_ST, 0)- IF(LFSS4_grade=4,
LFSS4_ST, 0)- IF(LFSS5_grade=4, LFSS5_ST, 0)- IF(LFSS6_grade=4, LFSS6_ST, 0)-
IF(LFSS7_grade=4, LFSS7_ST, 0)- IF(LFSS8_grade=4, LFSS8_ST, 0) ,
0)
doc   Lost_FH_ST_demand = This variable represents the lost demand of grade FH in silo
trucks at time t.

-----
aux   Lost_G_BB_demand = MAX(G_BB_demand-G_WH_outflow, 0)
doc   Lost_G_BB_demand = This variable represents the lost demand of grade G in big bags
at time t.

-----
aux   Lost_G_BC_demand = MAX(G_BC_demand- IF(LFSS1_grade=3, LFSS1_BC, 0) -
IF(LFSS2_grade=3, LFSS2_BC, 0)- IF(LFSS3_grade=3, LFSS3_BC, 0)- IF(LFSS4_grade=3,
LFSS4_BC, 0)- IF(LFSS5_grade=3, LFSS5_BC, 0)- IF(LFSS6_grade=3, LFSS6_BC, 0)-
IF(LFSS7_grade=3, LFSS7_BC, 0)- IF(LFSS8_grade=3, LFSS8_BC, 0) ,
0)
doc   Lost_G_BC_demand = This variable represents the lost demand of grade G in bulk
containers at time t.

-----
aux   Lost_G_ST_demand = MAX(G_ST_demand- IF(LFSS1_grade=3, LFSS1_ST, 0) -
IF(LFSS2_grade=3, LFSS2_ST, 0)- IF(LFSS3_grade=3, LFSS3_ST, 0)- IF(LFSS4_grade=3,
LFSS4_ST, 0)- IF(LFSS5_grade=3, LFSS5_ST, 0)- IF(LFSS6_grade=3, LFSS6_ST, 0)-
IF(LFSS7_grade=3, LFSS7_ST, 0)- IF(LFSS8_grade=3, LFSS8_ST, 0) ,
0)
doc   Lost_G_ST_demand = This variable represents the lost demand of grade G in silo
trucks at time t.

-----
aux   Lost_SD_BB_demand = MAX(SD_BB_demand-SD_WH_outflow, 0)
doc   Lost_SD_BB_demand = This variable represents the lost demand of grade SD in big bags
at time t.

-----
aux   Lost_SD_BC_demand = MAX(SD_BC_demand- IF(LFSS1_grade=2, LFSS1_BC, 0) -
IF(LFSS2_grade=2, LFSS2_BC, 0)- IF(LFSS3_grade=2, LFSS3_BC, 0)- IF(LFSS4_grade=2,
LFSS4_BC, 0)- IF(LFSS5_grade=2, LFSS5_BC, 0)- IF(LFSS6_grade=2, LFSS6_BC, 0)-
IF(LFSS7_grade=2, LFSS7_BC, 0)- IF(LFSS8_grade=2, LFSS8_BC, 0) ,
0)
doc   Lost_SD_BC_demand = This variable represents the lost demand of grade SD in bulk
containers at time t.

-----
aux   Lost_SD_ST_demand = MAX(SD_ST_demand- IF(LFSS1_grade=2, LFSS1_ST, 0) -
IF(LFSS2_grade=2, LFSS2_ST, 0)- IF(LFSS3_grade=2, LFSS3_ST, 0)- IF(LFSS4_grade=2,
LFSS4_ST, 0)- IF(LFSS5_grade=2, LFSS5_ST, 0)- IF(LFSS6_grade=2, LFSS6_ST, 0)-
IF(LFSS7_grade=2, LFSS7_ST, 0)- IF(LFSS8_grade=2, LFSS8_ST, 0) ,
0)
doc   Lost_SD_ST_demand = This variable represents the lost demand of grade SD in silo
trucks at time t.

-----
aux   Lost_WG_BB_demand = MAX(WG_BB_demand-WG_WH_outflow, 0)
doc   Lost_WG_BB_demand = This variable represents the lost demand of grade WG in big bags
at time t.

-----
aux   Lost_WG_BC_demand = MAX(WG_BC_demand- IF(LFSS1_grade=1, LFSS1_BC, 0) -
IF(LFSS2_grade=1, LFSS2_BC, 0)- IF(LFSS3_grade=1, LFSS3_BC, 0)- IF(LFSS4_grade=1,
LFSS4_BC, 0)- IF(LFSS5_grade=1, LFSS5_BC, 0)- IF(LFSS6_grade=1, LFSS6_BC, 0)-
IF(LFSS7_grade=1, LFSS7_BC, 0)- IF(LFSS8_grade=1, LFSS8_BC, 0) ,
0)
doc   Lost_WG_BC_demand = This variable represents the lost demand of grade WG in bulk
containers at time t.

-----
aux   Lost_WG_ST_demand = MAX(WG_ST_demand- IF(LFSS1_grade=1, LFSS1_ST, 0) -
IF(LFSS2_grade=1, LFSS2_ST, 0)- IF(LFSS3_grade=1, LFSS3_ST, 0)- IF(LFSS4_grade=1,
LFSS4_ST, 0)- IF(LFSS5_grade=1, LFSS5_ST, 0)- IF(LFSS6_grade=1, LFSS6_ST, 0)-
IF(LFSS7_grade=1, LFSS7_ST, 0)- IF(LFSS8_grade=1, LFSS8_ST, 0) ,
0)
doc   Lost_WG_ST_demand = This variable represents the lost demand of grade WG in silo
trucks at time t.

-----
aux   Off_specification_production = IF(SSP_outflow=5, 33.33, 0)
doc   Off_specification_production = Grade 5 is of off specification type and can not be
sold.

-----
aux   Outflow_silo_1 = Slot_1*Valve_of_SILO_1*SSP_Failure*Manual_SSP_stop
doc   Outflow_silo_1 = This variable represents the color of the outflow coming out of
SILO 1 (if any).

-----
aux   Outflow_silo_2 = Slot_13*Valve_of_SILO_2*SSP_Failure*Manual_SSP_stop
doc   Outflow_silo_2 = This variable represents the color of the outflow coming out of
SILO 2 (if any).

-----
aux   Outflow_silo_3 = Slot_25*Valve_of_SILO_3*SSP_Failure*Manual_SSP_stop
doc   Outflow_silo_3 = This variable represents the color of the outflow coming out of
SILO 3 (if any).

```



```

aux   POLY_Prod_Rate = (POLY_Failure)*(Manual_POLY_stop)*(TSS_availability_CHECK)*
IF(R1=1, 1, IF(R1=2, 2, 3))
doc   POLY_Prod_Rate = This variable shows the color type of the semi product coming out
of the POLY (if any) in each 4-hour period (dt).

-----
aux   R_1_IN = IF(SILO_1_slot_availability_CHECK=1, POLY_Prod_Rate, 0)
doc   R_1_IN = This variable represents the color type inflow of the slot, coming directly
from POLY (if any).

-----
aux   R_10_9 = IF(Outflow_silo_1>0, Slot_10, 0)
doc   R_10_9 = If "Outflow_silo_1" is positive, then the contained semi product of the
slot is moved one slot forward.

-----
aux   R_10_IN = IF(SILO_1_slot_availability_CHECK=10, POLY_Prod_Rate, 0)
doc   R_10_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

-----
aux   R_11_10 = IF(Outflow_silo_1>0, Slot_11, 0)
doc   R_11_10 = If "Outflow_silo_1" is positive, then the contained semi product of the
slot is moved one slot forward.

-----
aux   R_11_IN = IF(SILO_1_slot_availability_CHECK=11, POLY_Prod_Rate, 0)
doc   R_11_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

-----
aux   R_12_11 = IF(Outflow_silo_1>0, Slot_12, 0)
doc   R_12_11 = If "Outflow_silo_1" is positive, then the contained semi product of the
slot is moved one slot forward.

-----
aux   R_12_IN = IF(SILO_1_slot_availability_CHECK=12, POLY_Prod_Rate, 0)
doc   R_12_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

-----
aux   R_13_IN = IF(SILO_2_slot_availability_CHECK=13, POLY_Prod_Rate, 0)
doc   R_13_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

-----
aux   R_14_13 = IF(Outflow_silo_2>0, Slot_14, 0)
doc   R_14_13 = If "Outflow_silo_2" is positive, then the contained semi product of the
slot is moved one slot forward.

-----
aux   R_14_IN = IF(SILO_2_slot_availability_CHECK=14, POLY_Prod_Rate, 0)
doc   R_14_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

-----
aux   R_15_14 = IF(Outflow_silo_2>0, Slot_15, 0)
doc   R_15_14 = If "Outflow_silo_2" is positive, then the contained semi product of the
slot is moved one slot forward.

-----
aux   R_15_IN = IF(SILO_2_slot_availability_CHECK=15, POLY_Prod_Rate, 0)
doc   R_15_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

-----
aux   R_16_15 = IF(Outflow_silo_2>0, Slot_16, 0)
doc   R_16_15 = If "Outflow_silo_2" is positive, then the contained semi product of the
slot is moved one slot forward.

-----
aux   R_16_IN = IF(SILO_2_slot_availability_CHECK=16, POLY_Prod_Rate, 0)
doc   R_16_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

-----
aux   R_17_16 = IF(Outflow_silo_2>0, Slot_17, 0)
doc   R_17_16 = If "Outflow_silo_2" is positive, then the contained semi product of the
slot is moved one slot forward.

-----
aux   R_17_IN = IF(SILO_2_slot_availability_CHECK=17, POLY_Prod_Rate, 0)
doc   R_17_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

-----
aux   R_18_17 = IF(Outflow_silo_2>0, Slot_18, 0)
doc   R_18_17 = If "Outflow_silo_2" is positive, then the contained semi product of the
slot is moved one slot forward.

-----
aux   R_18_IN = IF(SILO_2_slot_availability_CHECK=18, POLY_Prod_Rate, 0)
doc   R_18_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).

```



```

aux   R_19_18 = IF(Outflow_silo_2>0, Slot_19, 0)
doc   R_19_18 = If "Outflow_silo_2" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_19_IN = IF(SILO_2_slot_availability_CHECK=19, POLY_Prod_Rate, 0)
doc   R_19_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_2_1 = IF(Outflow_silo_1>0, Slot_2, 0)
doc   R_2_1 = If "Outflow_silo_1" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_2_IN = IF(SILO_1_slot_availability_CHECK=2, POLY_Prod_Rate, 0)
doc   R_2_IN = This variable represents the color type inflow of the slot, coming directly
      from POLY (if any).

-----
aux   R_20_19 = IF(Outflow_silo_2>0, Slot_20, 0)
doc   R_20_19 = If "Outflow_silo_2" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_20_IN = IF(SILO_2_slot_availability_CHECK=20, POLY_Prod_Rate, 0)
doc   R_20_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_21_20 = IF(Outflow_silo_2>0, Slot_21, 0)
doc   R_21_20 = If "Outflow_silo_2" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_21_IN = IF(SILO_2_slot_availability_CHECK=21, POLY_Prod_Rate, 0)
doc   R_21_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_22_21 = IF(Outflow_silo_2>0, Slot_22, 0)
doc   R_22_21 = If "Outflow_silo_2" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_22_IN = IF(SILO_2_slot_availability_CHECK=22, POLY_Prod_Rate, 0)
doc   R_22_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_23_22 = IF(Outflow_silo_2>0, Slot_23, 0)
doc   R_23_22 = If "Outflow_silo_2" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_23_IN = IF(SILO_2_slot_availability_CHECK=23, POLY_Prod_Rate, 0)
doc   R_23_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_24_23 = IF(Outflow_silo_2>0, Slot_24, 0)
doc   R_24_23 = If "Outflow_silo_2" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_24_IN = IF(SILO_2_slot_availability_CHECK=24, POLY_Prod_Rate, 0)
doc   R_24_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_25_IN = IF(SILO_3_slot_availability_CHECK=25, POLY_Prod_Rate, 0)
doc   R_25_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_26_25 = IF(Outflow_silo_3>0, Slot_26, 0)
doc   R_26_25 = If "Outflow_silo_3" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_26_IN = IF(SILO_3_slot_availability_CHECK=26, POLY_Prod_Rate, 0)
doc   R_26_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_27_26 = IF(Outflow_silo_3>0, Slot_27, 0)
doc   R_27_26 = If "Outflow_silo_3" is positive, then the contained semi product of the
      slot is moved one slot forward.

-----
aux   R_27_IN = IF(SILO_3_slot_availability_CHECK=27, POLY_Prod_Rate, 0)
doc   R_27_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).

-----
aux   R_28_27 = IF(Outflow_silo_3>0, Slot_28, 0)

```

```

doc R_28_27 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_28_IN = IF(SILO_3_slot_availability_CHECK=28, POLY_Prod_Rate, 0)
doc R_28_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).
-----
aux R_29_28 = IF(Outflow_silo_3>0, Slot_29, 0)
doc R_29_28 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_29_IN = IF(SILO_3_slot_availability_CHECK=29, POLY_Prod_Rate, 0)
doc R_29_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).
-----
aux R_3_2 = IF(Outflow_silo_1>0, Slot_3, 0)
doc R_3_2 = If "Outflow_silo_1" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_3_IN = IF(SILO_1_slot_availability_CHECK=3, POLY_Prod_Rate, 0)
doc R_3_IN = This variable represents the color type inflow of the slot, coming directly
from POLY (if any).
-----
aux R_30_29 = IF(Outflow_silo_3>0, Slot_30, 0)
doc R_30_29 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_30_IN = IF(SILO_3_slot_availability_CHECK=30, POLY_Prod_Rate, 0)
doc R_30_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).
-----
aux R_31_30 = IF(Outflow_silo_3>0, Slot_31, 0)
doc R_31_30 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_31_IN = IF(SILO_3_slot_availability_CHECK=31, POLY_Prod_Rate, 0)
doc R_31_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).
-----
aux R_32_31 = IF(Outflow_silo_3>0, Slot_32, 0)
doc R_32_31 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_32_IN = IF(SILO_3_slot_availability_CHECK=32, POLY_Prod_Rate, 0)
doc R_32_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).
-----
aux R_33_32 = IF(Outflow_silo_3>0, Slot_33, 0)
doc R_33_32 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_33_IN = IF(SILO_3_slot_availability_CHECK=33, POLY_Prod_Rate, 0)
doc R_33_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).
-----
aux R_34_33 = IF(Outflow_silo_3>0, Slot_34, 0)
doc R_34_33 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_34_IN = IF(SILO_3_slot_availability_CHECK=34, POLY_Prod_Rate, 0)
doc R_34_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).
-----
aux R_35_34 = IF(Outflow_silo_3>0, Slot_35, 0)
doc R_35_34 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_35_IN = IF(SILO_3_slot_availability_CHECK=35, POLY_Prod_Rate, 0)
doc R_35_IN = This variable represents the color type inflow of the slot, coming
directly from POLY (if any).
-----
aux R_36_35 = IF(Outflow_silo_3>0, Slot_36, 0)
doc R_36_35 = If "Outflow_silo_3" is positive, then the contained semi product of the
slot is moved one slot forward.
-----
aux R_36_IN = IF(SILO_3_slot_availability_CHECK=36, POLY_Prod_Rate, 0)

```

```

doc   R_36_IN = This variable represents the color type inflow of the slot, coming
      directly from POLY (if any).
-----
aux   R_37_38 = SSP_Failure*Manual_SSP_stop*Slot_37
doc   R_37_38 = The production is stopped whether a failure occurs or a Management order
      arrives.
      Every SSP operational dt the semi-product is pushed forward by this variable.
-----
aux   R_38_39 = SSP_Failure*Manual_SSP_stop*Slot_38
doc   R_38_39 = The production is stopped whether a failure occurs or a Management order
      arrives.
      Every SSP operational dt the semi-product is pushed forward by this variable.
-----
aux   R_39_40 = SSP_Failure*Manual_SSP_stop*Slot_39
doc   R_39_40 = The production is stopped whether a failure occurs or a Management order
      arrives.
      Every SSP operational dt the semi-product is pushed forward by this variable.
-----
aux   R_4_3 = IF(Outflow_silo_1>0, Slot_4, 0)
doc   R_4_3 = If "Outflow_silo_1" is positive, then the contained semi product of the
      slot is moved one slot forward.
-----
aux   R_4_IN = IF(SILO_1_slot_availability_CHECK=4, POLY_Prod_Rate, 0)
doc   R_4_IN = This variable represents the color type inflow of the slot, coming directly
      from POLY (if any).
-----
aux   R_40_41 = SSP_Failure*Manual_SSP_stop*Slot_40
doc   R_40_41 = The production is stopped whether a failure occurs or a Management order
      arrives.
      Every SSP operational dt the semi-product is pushed forward by this variable.
-----
aux   R_41_42 = SSP_Failure*Manual_SSP_stop*Slot_41
doc   R_41_42 = The production is stopped whether a failure occurs or a Management order
      arrives.
      Every SSP operational dt the semi-product is pushed forward by this variable.
-----
aux   R_42_out = SSP_Failure*Manual_SSP_stop*Slot_42
doc   R_42_out = The production is stopped whether a failure occurs or a Management order
      arrives.
      Every SSP operational dt the semi-product is pushed forward by this variable.
-----
aux   R_5_4 = IF(Outflow_silo_1>0, Slot_5, 0)
doc   R_5_4 = If "Outflow_silo_1" is positive, then the contained semi product of the
      slot is moved one slot forward.
-----
aux   R_5_IN = IF(SILO_1_slot_availability_CHECK=5, POLY_Prod_Rate, 0)
doc   R_5_IN = This variable represents the color type inflow of the slot, coming directly
      from POLY (if any).
-----
aux   R_6_5 = IF(Outflow_silo_1>0, Slot_6, 0)
doc   R_6_5 = If "Outflow_silo_1" is positive, then the contained semi product of the
      slot is moved one slot forward.
-----
aux   R_6_IN = IF(SILO_1_slot_availability_CHECK=6, POLY_Prod_Rate, 0)
doc   R_6_IN = This variable represents the color type inflow of the slot, coming directly
      from POLY (if any).
-----
aux   R_7_6 = IF(Outflow_silo_1>0, Slot_7, 0)
doc   R_7_6 = If "Outflow_silo_1" is positive, then the contained semi product of the
      slot is moved one slot forward.
-----
aux   R_7_IN = IF(SILO_1_slot_availability_CHECK=7, POLY_Prod_Rate, 0)
doc   R_7_IN = This variable represents the color type inflow of the slot, coming directly
      from POLY (if any).
-----
aux   R_8_7 = IF(Outflow_silo_1>0, Slot_8, 0)
doc   R_8_7 = If "Outflow_silo_1" is positive, then the contained semi product of the
      slot is moved one slot forward.
-----
aux   R_8_IN = IF(SILO_1_slot_availability_CHECK=8, POLY_Prod_Rate, 0)
doc   R_8_IN = This variable represents the color type inflow of the slot, coming directly
      from POLY (if any).
-----
aux   R_9_8 = IF(Outflow_silo_1>0, Slot_9, 0)
doc   R_9_8 = If "Outflow_silo_1" is positive, then the contained semi product of the
      slot is moved one slot forward.
-----

```

```

aux   R_9_IN = IF(SILO_1_slot_availability_CHECK=9, POLY_Prod_Rate, 0)
doc   R_9_IN = This variable represents the color type inflow of the slot, coming directly
       from POLY (if any).

-----
aux   R1 = IF(((Color_changeover=0) AND (POLY_Current_Color=1 OR POLY_Current_Color=3)),
             POLY_Current_Color,
             IF((Color_changeover=0) AND (POLY_Current_Color=2),
               IF(Color_at_t_1=1,3, 1),
               2))
doc   R1 = Every dt the "Current color" is revised by this variable, based on the color
       changeover.
       Current color=1 is equal to Current color=LIGHT
       Current color=2 is equal to Current color=GRAY
       Current color=3 is equal to Current color=DARK

-----
aux   R10 = IV_stage_6
doc   R10 = To keep track of the IV at the sixth stage we should get rid of the IV at the
       seventh stage.
       SSP_Current_IV=1 equals to LOW VISCOSITY
       SSP_Current_IV=2 equals to HIGH VISCOSITY

-----
aux   R11 = IF(LFSS1<=Smin,
             IF(LFSS1_inflow>0, SSP_outflow, 0),
             0)
doc   R11 = This variable represents the type of the grade coming into LFSS1 in each 4-
       hour period.

-----
aux   R12 = IF((LFSS1<=Smin) , LFSS1_grade, 0)
doc   R12 = This variable is used so as to reboot the type of the grade contained in LFSS1
       whenever its contained quantity is below Smin.

-----
aux   R13 = IF(LFSS2<=Smin,
             IF(LFSS2_inflow>0, SSP_outflow, 0),
             0)
doc   R13 = This variable represents the type of the grade coming into LFSS2 in each 4-
       hour period.

-----
aux   R14 = IF((LFSS2<=Smin) , LFSS2_grade, 0)
doc   R14 = This variable is used so as to reboot the type of the grade contained in LFSS2
       whenever its contained quantity is below Smin.

-----
aux   R15 = IF(LFSS3<=Smin,
             IF(LFSS3_inflow>0, SSP_outflow, 0),
             0)
doc   R15 = This variable represents the type of the grade coming into LFSS3 in each 4-
       hour period.

-----
aux   R16 = IF((LFSS3<=Smin) , LFSS3_grade, 0)
doc   R16 = This variable is used so as to reboot the type of the grade contained in LFSS3
       whenever its contained quantity is below Smin.

-----
aux   R17 = IF(LFSS4<=Smin,
             IF(LFSS4_inflow>0, SSP_outflow, 0),
             0)
doc   R17 = This variable represents the type of the grade coming into LFSS4 in each 4-
       hour period.

-----
aux   R18 = IF((LFSS4<=Smin) , LFSS4_grade, 0)
doc   R18 = This variable is used so as to reboot the type of the grade contained in LFSS4
       whenever its contained quantity is below Smin.

-----
aux   R19 = IF(LFSS5<=Smin,
             IF(LFSS5_inflow>0, SSP_outflow, 0),
             0)
doc   R19 = This variable represents the type of the grade coming into LFSS5 in each 4-
       hour period.

-----
aux   R2 = POLY_Current_Color
doc   R2 = Every dt the "Color at t-1" is revised by this variable, based on the color
       changeover.

-----
aux   R20 = IF((LFSS5<=Smin) , LFSS5_grade, 0)
doc   R20 = This variable is used so as to reboot the type of the grade contained in LFSS5
       whenever its contained quantity is below Smin.

-----
aux   R21 = IF(LFSS6<=Smin,
             IF(LFSS6_inflow>0, SSP_outflow, 0),
             0)
doc   R21 = This variable represents the type of the grade coming into LFSS6 in each 4-
       hour period.

-----
aux   R22 = IF((LFSS6<=Smin) , LFSS6_grade, 0)

```

```

doc    R22 = This variable is used so as to reboot the type of the grade contained in LFSS6
        whenever its contained quantity is below Smin.
-----
aux    R23 = IF(LFSS7_inflow>0, SSP_outflow, 0), IF(LFSS7<=Smin,
doc    R23 = This variable represents the type of the grade coming into LFSS7 in each 4-
        hour period.
-----
aux    R24 = IF((LFSS7<=Smin) , LFSS7_grade, 0)
doc    R24 = This variable is used so as to reboot the type of the grade contained in LFSS7
        whenever its contained quantity is below Smin.
-----
aux    R25 = IF(LFSS8_inflow>0, SSP_outflow, 0), IF(LFSS8<=Smin,
doc    R25 = This variable represents the type of the grade coming into LFSS8 in each 4-
        hour period.
-----
aux    R26 = IF((LFSS8<=Smin) , LFSS8_grade, 0)
doc    R26 = This variable is used so as to reboot the type of the grade contained in LFSS8
        whenever its contained quantity is below Smin.
-----
aux    R3 = Color_at_t_1
doc    R3 = To keep track of the color at time step t-1 we should get rid of the color at
        t-2.
-----
aux    R4 = IF(IV_changeover=0, IV_stage_1, IF((IV_stage_1=1), 2, 1))
doc    R4 = This block of orders represents the mechanism rulling an IV changeover.
        Although an IV changeover occurs gradually in a 24-hour period (six 4-hour periods),
        our model makes the following hypothesis; if an IV changeover is initiated at "t"
        then the IV will remain unchanged till time "t+3" when the IV changes instantly and
        remains unchanged till time and "t+6".
        Every dt the "IV_stage_1" is revised by this variable, based on the "IV_
        changeover".
        SSP_Current_IV=1 equals to LOW VISCOSITY
        SSP_Current_IV=2 equals to HIGH VISCOSITY
-----
aux    R5 = IV_stage_1
doc    R5 = Every dt the "IV_stage_2" is revised by this variable, based on the
        "IV_stage_1".
        SSP_Current_IV=1 equals to LOW VISCOSITY
        SSP_Current_IV=2 equals to HIGH VISCOSITY
-----
aux    R6 = IV_stage_2
doc    R6 = Every dt the "IV_stage_3" is revised by this variable, based on the
        "IV_stage_2".
        SSP_Current_IV=1 equals to LOW VISCOSITY
        SSP_Current_IV=2 equals to HIGH VISCOSITY
-----
aux    R7 = SSP_Current_IV
doc    R7 = Every dt the "IV_stage_4" is revised by this variable, based on the
        "IV_stage_3".
        SSP_Current_IV=1 equals to LOW VISCOSITY
        SSP_Current_IV=2 equals to HIGH VISCOSITY
-----
aux    R8 = IV_stage_4
doc    R8 = Every dt the "IV_stage_5" is revised by this variable, based on the
        "IV_stage_4".
        SSP_Current_IV=1 equals to LOW VISCOSITY
        SSP_Current_IV=2 equals to HIGH VISCOSITY
-----
aux    R9 = IV_stage_5
doc    R9 = Every dt the "IV_stage_6" is revised by this variable, based on the
        "IV_stage_5".
        SSP_Current_IV=1 equals to LOW VISCOSITY
        SSP_Current_IV=2 equals to HIGH VISCOSITY
-----
aux    SD_job_slots_completion = IF(Remaining_job_slots_of_SD_to_produce>0,
        IF(SSP_outflow=2, 1, 0),
        0)
doc    SD_job_slots_completion = The production of product SD has priority 2 in comparison
        to the other products because its total safety stock (LFSS+WH) is second-less than
        those of the others, resulting in second-less production time (SPT logic).
-----
aux    SD_job_slots_orders = Need_for_SD_in_job_lot
-----

```



```

aux   SD_WH_inflow = IF(LFSS1_grade=2, LFSS1_BB, 0) + IF(LFSS2_grade=2, LFSS2_BB, 0) +
        IF(LFSS3_grade=2, LFSS3_BB, 0) + IF(LFSS4_grade=2, LFSS4_BB, 0) +
        IF(LFSS5_grade=2, LFSS5_BB, 0) + IF(LFSS6_grade=2, LFSS6_BB, 0) +
        IF(LFSS7_grade=2, LFSS7_BB, 0) + IF(LFSS8_grade=2, LFSS8_BB, 0)
doc   SD_WH_inflow = Quantity of SD coming into the warehouse at time t.

-----
aux   SD_WH_outflow = MIN(SD_WH_stock,SD_BB_demand)
doc   SD_WH_outflow = Quantity of SD coming out from the warehouse at time t.

-----
aux   Silos_outflow = Outflow_silo_1+Outflow_silo_2+Outflow_silo_3
doc   Silos_outflow = This variable represents the color of the semi-product coming out
        from the silos. We would like to remind the reader that only one silo outflow can be
        positive.

-----
aux   Spilled_over_products = IF((Opt_WG_LFSS_capacity=500) AND (SSP_outflow=1), 33.33, 0)
        +
        IF((Opt_SD_LFSS_capacity=500) AND (SSP_outflow=2), 33.33, 0) +
        IF((Opt_G_LFSS_capacity=500) AND (SSP_outflow=3), 33.33, 0) +
        IF((Opt_FH_LFSS_capacity=500) AND (SSP_outflow=4), 33.33, 0)
doc   Spilled_over_products = If there is not any LFSS silo available for storing the
        final grades, then the grades coming out from LFSS are spilled over.

-----
aux   SSP_inflow = Silos_outflow
doc   SSP_inflow = This variable represents the color of the semi-product entering the
        SSP.

-----
aux   SSP_outflow = SSP_Failure*Manual_SSP_stop*
        IF(R_42_out=0, 0,
            IF((R_42_out=1) AND (SSP_Current_IV=1), 1,
                IF((R_42_out=1) AND (SSP_Current_IV=2), 2,
                    IF((R_42_out=2) AND (SSP_Current_IV=2), 3,
                        IF((R_42_out=3) AND (SSP_Current_IV=2), 4, 5))))))
doc   SSP_outflow = This variable converts the semi-product contained in Slot 42 into a
        final grade, according to the color of the semi product and the SSP current IV of
        the SSP stage.
        If the produced grade is of off-specification type, then we consider that Grade 5 is
        produced.
        SSP_Outflow=1 is equal to that the final grade WG is produced.
        SSP_Outflow=2 is equal to that the final grade SD is produced.
        SSP_Outflow=3 is equal to that the final grade G is produced.
        SSP_Outflow=4 is equal to that the final grade FH is produced.
        SSP_Outflow=5 is equal to that the final grade is of off specification type.

-----
aux   WG_job_slots_completion = IF(Remaining_job_slots_of_WG_to_produce>0,
        IF(SSP_outflow=1, 1, 0),
        0)
doc   WG_job_slots_completion = The production of product WG has priority 1 in comparison
        to the other products because its total safety stock (LFSS+WH) is less than those of
        the others, resulting in less production time (SPT logic).

-----
aux   WG_job_slots_orders = Need_for_WG_in_job_lot

-----
aux   WG_WH_inflow = IF(LFSS1_grade=1, LFSS1_BB, 0) + IF(LFSS2_grade=1, LFSS2_BB, 0) +
        IF(LFSS3_grade=1, LFSS3_BB, 0) + IF(LFSS4_grade=1, LFSS4_BB, 0) +
        IF(LFSS5_grade=1, LFSS5_BB, 0) + IF(LFSS6_grade=1, LFSS6_BB, 0) +
        IF(LFSS7_grade=1, LFSS7_BB, 0) + IF(LFSS8_grade=1, LFSS8_BB, 0)
doc   WG_WH_inflow = Quantity of WG coming into the warehouse at time t.

-----
aux   WG_WH_outflow = MIN(WG_WH_stock,WG_BB_demand)
doc   WG_WH_outflow = Quantity of WG coming out from the warehouse at time t.

-----
aux   WH_inflow = WG_WH_inflow+SD_WH_inflow+G_WH_inflow+FH_WH_inflow
doc   WH_inflow = Total quantity of grades coming into the warehouse at time t.

-----
aux   WH_outflow = WG_WH_outflow+SD_WH_outflow+G_WH_outflow+FH_WH_outflow
doc   WH_outflow = Total quantity of grades coming out from the warehouse at time t.

-----
aux   Color_changeover = IF(Color_changeover_at_t_1=0, Order_color_changeover, 0)
doc   Color_changeover = A color changeover can be initiated whether the POLY is
        operational or not.
        The event of 2 successive color changeovers is forbidden in practice and should not
        be encountered.
        If this variable is equal to 0 then no color changeover is initialized.
        If this variable is equal to 1, if the var. "color changeover at t-1" equals 0 then
        a color changeover is initiated, else no color changeover is initiated.

```

```

aux   Color_changeover_at_t_1 = DELAYPPL(Color_changeover, 1, 0)
-----
aux   FH_LFSS_s1 = (1-K1)*FH_LFSS_safety_stock
-----
aux   FH_LFSS_S2 = (1+K1)*FH_LFSS_safety_stock
-----
aux   FH_LFSS_stock = IF(LFSS1_grade=4, LFSS1, 0) + IF(LFSS2_grade=4, LFSS2, 0) +
                    IF(LFSS3_grade=4, LFSS3, 0) + IF(LFSS4_grade=4, LFSS4, 0) +
                    IF(LFSS5_grade=4, LFSS5, 0) + IF(LFSS6_grade=4, LFSS6, 0) +
                    IF(LFSS7_grade=4, LFSS7, 0) + IF(LFSS8_grade=4, LFSS8, 0)
doc   FH_LFSS_stock = Quantity of FH stored in the LFSS.
-----
aux   Fillrate = (Total_Demand-Total_lost_demand)/Total_Demand
-----
aux   G_LFSS_s1 = (1-K1)*G_LFSS_safety_stock
-----
aux   G_LFSS_S2 = (1+K1)*G_LFSS_safety_stock
-----
aux   G_LFSS_stock = IF(LFSS1_grade=3, LFSS1, 0) + IF(LFSS2_grade=3, LFSS2, 0) +
                    IF(LFSS3_grade=3, LFSS3, 0) + IF(LFSS4_grade=3, LFSS4, 0) +
                    IF(LFSS5_grade=3, LFSS5, 0) + IF(LFSS6_grade=3, LFSS6, 0) +
                    IF(LFSS7_grade=3, LFSS7, 0) + IF(LFSS8_grade=3, LFSS8, 0)
doc   G_LFSS_stock = Quantity of G stored in the LFSS.
-----
aux   IV_changeover = SSP_Failure*Manual_SSP_stop*
                    IF(IV_stage_1=IV_stage_6,
                      IF(Slot_42=1,
                        IF(SSP_inflow=2,
                          IF(SSP_Current_IV=1,
                            1,
                            0),
                          IF(SSP_inflow=3,
                            0,
                            IF(SSP_Current_IV=1,
                              IF(Remaining_job_slots_of_WG_to_produce>3,
                                0,
                                IF(Remaining_job_slots_of_SD_to_produce>0,
                                  1,
                                  0)),
                              IF(SSP_inflow=1 AND Slot_6=1,
                                IF(Remaining_job_slots_of_SD_to_produce>3,
                                  0,
                                  IF(Remaining_job_slots_of_WG_to_produce>0,
                                    1,
                                    0))),
                                0))),
                          0),
                      0)
doc   IV_changeover = An IV changeover can be initiated only if SSP is operational.
                    An IV changeover can be completed only if the SSP is operational and not during an
                    SSP failure.
                    A new IV changeover can be initiated only if the previous one is completed.
                    If this variable is equal to 0 then no IV changeover is initiated.
                    If this variable is equal to 1 then an IV changeover is initiated (the SSP is
                    operational and the previous IV changeover has been completed).
-----
aux   LFSS1_FH_check = IF(LFSS1_grade=4,
                        IF((Smax-LFSS1)>=33.33, (Smax-LFSS1), 500),
                        IF(LFSS1<=Smin, Smax, 500))
doc   LFSS1_FH_check = Quantity of FH coming out from SSP that can be stored in LFSS1.
                    Notice that we use a trick for the case where the grade already stored in LFSS1 is
                    of another type; we use the number 500 which is above Smax and with a later check
                    described by the equations of variable "Opt_WG_LFSS" we define that if
                    LFSS1_FH_check equals 500 then LFSS1 can not receive the outflow of the SSP stage.
-----
aux   LFSS1_G_check = IF(LFSS1_grade=3,
                        IF((Smax-LFSS1)>=33.33, (Smax-LFSS1), 500),
                        IF(LFSS1<=Smin, Smax, 500))
doc   LFSS1_G_check = Quantity of G coming out from SSP that can be stored in LFSS1.
                    Notice that we use a trick for the case where the grade already stored in LFSS1 is
                    of another type; we use the number 500 which is above Smax and with a later check
                    described by the equations of variable "Opt_WG_LFSS" we define that if LFSS1_G_check
                    equals 500 then LFSS1 can not receive the outflow of the SSP stage.
-----
aux   LFSS1_SD_check = IF(LFSS1_grade=2,
                        IF((Smax-LFSS1)>=33.33, (Smax-LFSS1), 500),

```

```

                                IF(LFSS1<=Smin, Smax, 500))
doc  LFSS1_SD_check = Quantity of SD coming out from SSP that can be stored in LFSS1.
      Notice that we use a trick for the case where the grade already stored in LFSS1 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS1_SD_check equals 500 then LFSS1 can not receive the outflow of the SSP stage.
-----
aux  LFSS1_WG_check = IF(LFSS1_grade=1,
                        IF((Smax-LFSS1)>=33.33,
                          (Smax-LFSS1),
                          500),
                        IF(LFSS1<=Smin,
                          Smax,
                          500))
doc  LFSS1_WG_check = Quantity of WG coming out from SSP that can be stored in LFSS1.
      Notice that we use a trick for the case where the grade already stored in LFSS1 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS1_WG_check equals 500 then LFSS1 can not receive the outflow of the SSP stage.
-----
aux  LFSS2_FH_check = IF(LFSS2_grade=4,
                        IF((Smax-LFSS2)>=33.33, (Smax-LFSS2), 500),
                        IF(LFSS2<=Smin, Smax, 500))
doc  LFSS2_FH_check = Quantity of FH coming out from SSP that can be stored in LFSS2.
      Notice that we use a trick for the case where the grade already stored in LFSS2 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS2_FH_check equals 500 then LFSS2 can not receive the outflow of the SSP stage.
-----
aux  LFSS2_G_check = IF(LFSS2_grade=3,
                        IF((Smax-LFSS2)>=33.33, (Smax-LFSS2), 500),
                        IF(LFSS2<=Smin, Smax, 500))
doc  LFSS2_G_check = Quantity of G coming out from SSP that can be stored in LFSS2.
      Notice that we use a trick for the case where the grade already stored in LFSS2 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if LFSS2_G_check
      equals 500 then LFSS2 can not receive the outflow of the SSP stage.
-----
aux  LFSS2_SD_check = IF(LFSS2_grade=2,
                        IF((Smax-LFSS2)>=33.33, (Smax-LFSS2), 500),
                        IF(LFSS2<=Smin, Smax, 500))
doc  LFSS2_SD_check = Quantity of SD coming out from SSP that can be stored in LFSS2.
      Notice that we use a trick for the case where the grade already stored in LFSS2 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS2_SD_check equals 500 then LFSS2 can not receive the outflow of the SSP stage.
-----
aux  LFSS2_WG_check = IF(LFSS2_grade=1,
                        IF((Smax-LFSS2)>=33.33, (Smax-LFSS2), 500),
                        IF(LFSS2<=Smin, Smax, 500))
doc  LFSS2_WG_check = Quantity of WG coming out from SSP that can be stored in LFSS2.
      Notice that we use a trick for the case where the grade already stored in LFSS2 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS2_WG_check equals 500 then LFSS2 can not receive the outflow of the SSP stage.
-----
aux  LFSS3_FH_check = IF(LFSS3_grade=4,
                        IF((Smax-LFSS3)>=33.33, (Smax-LFSS3), 500),
                        IF(LFSS3<=Smin, Smax, 500))
doc  LFSS3_FH_check = Quantity of FH coming out from SSP that can be stored in LFSS3.
      Notice that we use a trick for the case where the grade already stored in LFSS3 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS3_FH_check equals 500 then LFSS3 can not receive the outflow of the SSP stage.
-----
aux  LFSS3_G_check = IF(LFSS3_grade=3,
                        IF((Smax-LFSS3)>=33.33, (Smax-LFSS3), 500),
                        IF(LFSS3<=Smin, Smax, 500))
doc  LFSS3_G_check = Quantity of G coming out from SSP that can be stored in LFSS3.
      Notice that we use a trick for the case where the grade already stored in LFSS3 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if LFSS3_G_check
      equals 500 then LFSS3 can not receive the outflow of the SSP stage.
-----
aux  LFSS3_SD_check = IF(LFSS3_grade=2,
                        IF((Smax-LFSS3)>=33.33, (Smax-LFSS3), 500),
                        IF(LFSS3<=Smin, Smax, 500))

```

```

doc  LFSS3_SD_check = Quantity of SD coming out from SSP that can be stored in LFSS3.
      Notice that we use a trick for the case where the grade already stored in LFSS3 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS3_SD_check equals 500 then LFSS3 can not receive the outflow of the SSP stage.
-----
aux  LFSS3_WG_check = IF(LFSS3_grade=1,
                        IF((Smax-LFSS3)>=33.33, (Smax-LFSS3), 500),
                        IF(LFSS3<=Smin, Smax, 500))
doc  LFSS3_WG_check = Quantity of WG coming out from SSP that can be stored in LFSS3.
      Notice that we use a trick for the case where the grade already stored in LFSS3 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS3_WG_check equals 500 then LFSS3 can not receive the outflow of the SSP stage.
-----
aux  LFSS4_FH_check = IF(LFSS4_grade=4,
                        IF((Smax-LFSS4)>=33.33, (Smax-LFSS4), 500),
                        IF(LFSS4<=Smin, Smax, 500))
doc  LFSS4_FH_check = Quantity of FH coming out from SSP that can be stored in LFSS4.
      Notice that we use a trick for the case where the grade already stored in LFSS4 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS4_FH_check equals 500 then LFSS4 can not receive the outflow of the SSP stage.
-----
aux  LFSS4_G_check = IF(LFSS4_grade=3,
                       IF((Smax-LFSS4)>=33.33, (Smax-LFSS4), 500),
                       IF(LFSS4<=Smin, Smax, 500))
doc  LFSS4_G_check = Quantity of G coming out from SSP that can be stored in LFSS4.
      Notice that we use a trick for the case where the grade already stored in LFSS4 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if LFSS4_G_check
      equals 500 then LFSS4 can not receive the outflow of the SSP stage.
-----
aux  LFSS4_SD_check = IF(LFSS4_grade=2,
                        IF((Smax-LFSS4)>=33.33, (Smax-LFSS4), 500),
                        IF(LFSS4<=Smin, Smax, 500))
doc  LFSS4_SD_check = Quantity of SD coming out from SSP that can be stored in LFSS4.
      Notice that we use a trick for the case where the grade already stored in LFSS4 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS4_SD_check equals 500 then LFSS4 can not receive the outflow of the SSP stage.
-----
aux  LFSS4_WG_check = IF(LFSS4_grade=1,
                        IF((Smax-LFSS4)>=33.33, (Smax-LFSS4), 500),
                        IF(LFSS4<=Smin, Smax, 500))
doc  LFSS4_WG_check = Quantity of WG coming out from SSP that can be stored in LFSS4.
      Notice that we use a trick for the case where the grade already stored in LFSS4 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS4_WG_check equals 500 then LFSS4 can not receive the outflow of the SSP stage.
-----
aux  LFSS5_FH_check = IF(LFSS5_grade=4,
                        IF((Smax-LFSS5)>=33.33, (Smax-LFSS5), 500),
                        IF(LFSS5<=Smin, Smax, 500))
doc  LFSS5_FH_check = Quantity of FH coming out from SSP that can be stored in LFSS5.
      Notice that we use a trick for the case where the grade already stored in LFSS5 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS5_FH_check equals 500 then LFSS5 can not receive the outflow of the SSP stage.
-----
aux  LFSS5_G_check = IF(LFSS5_grade=3,
                       IF((Smax-LFSS5)>=33.33, (Smax-LFSS5), 500),
                       IF(LFSS5<=Smin, Smax, 500))
doc  LFSS5_G_check = Quantity of G coming out from SSP that can be stored in LFSS5.
      Notice that we use a trick for the case where the grade already stored in LFSS5 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if LFSS5_G_check
      equals 500 then LFSS5 can not receive the outflow of the SSP stage.
-----
aux  LFSS5_SD_check = IF(LFSS5_grade=2,
                        IF((Smax-LFSS5)>=33.33, (Smax-LFSS5), 500),
                        IF(LFSS5<=Smin, Smax, 500))
doc  LFSS5_SD_check = Quantity of SD coming out from SSP that can be stored in LFSS5.
      Notice that we use a trick for the case where the grade already stored in LFSS5 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS5_SD_check equals 500 then LFSS5 can not receive the outflow of the SSP stage.

```



```

aux  LFSS5_WG_check = IF(LFSS5_grade=1,
                        IF((Smax-LFSS5)>=33.33, (Smax-LFSS5), 500),
                        IF(LFSS5<=Smin, Smax, 500))
doc  LFSS5_WG_check = Quantity of WG coming out from SSP that can be stored in LFSS5.
      Notice that we use a trick for the case where the grade already stored in LFSS5 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS5_WG_check equals 500 then LFSS5 can not receive the outflow of the SSP stage.

aux  LFSS6_FH_check = IF(LFSS6_grade=4,
                        IF((Smax-LFSS6)>=33.33, (Smax-LFSS6), 500),
                        IF(LFSS6<=Smin, Smax, 500))
doc  LFSS6_FH_check = Quantity of FH coming out from SSP that can be stored in LFSS6.
      Notice that we use a trick for the case where the grade already stored in LFSS6 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS6_FH_check equals 500 then LFSS6 can not receive the outflow of the SSP stage.

aux  LFSS6_G_check = IF(LFSS6_grade=3,
                       IF((Smax-LFSS6)>=33.33, (Smax-LFSS6), 500),
                       IF(LFSS6<=Smin, Smax, 500))
doc  LFSS6_G_check = Quantity of G coming out from SSP that can be stored in LFSS6.
      Notice that we use a trick for the case where the grade already stored in LFSS6 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS6_G_check equals 500 then LFSS6 can not receive the outflow of the SSP stage.

aux  LFSS6_SD_check = IF(LFSS6_grade=2,
                        IF((Smax-LFSS6)>=33.33, (Smax-LFSS6), 500),
                        IF(LFSS6<=Smin, Smax, 500))
doc  LFSS6_SD_check = Quantity of SD coming out from SSP that can be stored in LFSS6.
      Notice that we use a trick for the case where the grade already stored in LFSS6 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS6_SD_check equals 500 then LFSS6 can not receive the outflow of the SSP stage.

aux  LFSS6_WG_check = IF(LFSS6_grade=1,
                        IF((Smax-LFSS6)>=33.33, (Smax-LFSS6), 500),
                        IF(LFSS6<=Smin, Smax, 500))
doc  LFSS6_WG_check = Quantity of WG coming out from SSP that can be stored in LFSS6.
      Notice that we use a trick for the case where the grade already stored in LFSS6 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS6_WG_check equals 500 then LFSS6 can not receive the outflow of the SSP stage.

aux  LFSS7_FH_check = IF(LFSS7_grade=4,
                        IF((Smax-LFSS7)>=33.33, (Smax-LFSS7), 500),
                        IF(LFSS7<=Smin, Smax, 500))
doc  LFSS7_FH_check = Quantity of FH coming out from SSP that can be stored in LFSS7.
      Notice that we use a trick for the case where the grade already stored in LFSS7 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS7_FH_check equals 500 then LFSS7 can not receive the outflow of the SSP stage.

aux  LFSS7_G_check = IF(LFSS7_grade=3,
                       IF((Smax-LFSS7)>=33.33, (Smax-LFSS7), 500),
                       IF(LFSS7<=Smin, Smax, 500))
doc  LFSS7_G_check = Quantity of G coming out from SSP that can be stored in LFSS7.
      Notice that we use a trick for the case where the grade already stored in LFSS7 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS7_G_check equals 500 then LFSS7 can not receive the outflow of the SSP stage.

aux  LFSS7_SD_check = IF(LFSS7_grade=2,
                        IF((Smax-LFSS7)>=33.33, (Smax-LFSS7), 500),
                        IF(LFSS7<=Smin, Smax, 500))
doc  LFSS7_SD_check = Quantity of SD coming out from SSP that can be stored in LFSS7.
      Notice that we use a trick for the case where the grade already stored in LFSS7 is
      of another type; we use the number 500 which is above Smax and with a later check
      described by the equations of variable "Opt_WG_LFSS" we define that if
      LFSS7_SD_check equals 500 then LFSS7 can not receive the outflow of the SSP stage.

aux  LFSS7_WG_check = IF(LFSS7_grade=1,
                        IF((Smax-LFSS7)>=33.33, (Smax-LFSS7), 500),
                        IF(LFSS7<=Smin, Smax, 500))

```



```

doc   Need_for_G = Quantity of product G that must be ordered so as to meet the "ss"
      policy.
-----
aux   Need_for_G_in_job_lot = CEIL(Need_for_G/33.33)
doc   Need_for_G_in_job_lot = Number of job lots of product G that must be produced to
      "reach" the upper bound of the "ss" policy, in the beginning of a new production
      order for G.
-----
aux   Need_for_SD = IF(SD_WH_stock<SD_WH_safety_stock*(1-K1),
      MAX( (SD_WH_safety_stock*(1+K1) - SD_WH_stock) + MAX(SD_LFSS_S2-SD_LFSS_stock,0)
      - Remaining_job_slots_of_SD_to_produce*33.33, 0),
      0)
doc   Need_for_SD = Quantity of product SD that must be ordered so as to meet the "ss"
      policy.
-----
aux   Need_for_SD_in_job_lot = CEIL(Need_for_SD/33.33)
doc   Need_for_SD_in_job_lot = Number of job lots of product SD that must be produced to
      "reach" the upper bound of the "ss" policy, in the beginning of a new production
      order for SD.
-----
aux   Need_for_WG = IF(WG_LFSS_stock<WG_LFSS_sl,
      MAX( (WG_LFSS_S2-WG_LFSS_stock)+MAX(WG_WH_safety_stock-WG_WH_stock,0)
      -Remaining_job_slots_of_WG_to_produce*33.33, 0),
      0)
doc   Need_for_WG = Quantity of product WG that must be ordered so as to meet the "ss"
      policy.
-----
aux   Need_for_WG_in_job_lot = CEIL(Need_for_WG/33.33)
doc   Need_for_WG_in_job_lot = Number of job lots of product WG that must be produced to
      "reach" the upper bound of the "ss" policy, in the beginning of a new production
      order for WG.
-----
aux   Opt_FH_LFSS_capacity = MIN(LFSS1_FH_check, LFSS2_FH_check, LFSS3_FH_check,
      LFSS4_FH_check, LFSS5_FH_check, LFSS6_FH_check, LFSS7_FH_check, LFSS8_FH_check)
doc   Opt_FH_LFSS_capacity = This variable finds the LFSS silo that first, can receive a
      FH quantity of 33.33 tons and second, has the most FH quantity among all so as
      finally, to receive full LFSS silos that create more LFSS that are empty and
      available for whatever grade. This is attributed to the fact that the semi-full
      silos or even worse, silos with a small quantity above Smin, are unavailable for
      storing other types of grades.
      If this variable is equal to 500 then this means that there is not any available
      slot in the LFSS silos for FH and the quantity coming out from SSP is spilled over.
-----
aux   Opt_G_LFSS_capacity = MIN(LFSS1_G_check, LFSS2_G_check, LFSS3_G_check,
      LFSS4_G_check, LFSS5_G_check, LFSS6_G_check, LFSS7_G_check, LFSS8_G_check)
doc   Opt_G_LFSS_capacity = This variable finds the LFSS silo that first, can receive a G
      quantity of 33.33 tons and second, has the most G quantity among all so as finally,
      to receive full LFSS silos that create more LFSS that are empty and available for
      whatever grade. This is attributed to the fact that the semi-full silos or even
      worse, silos with a small quantity above Smin, are unavailable for storing other
      types of grades.
      If this variable is equal to 500 then this means that there is not any available
      slot in the LFSS silos for G and the quantity coming out from SSP is spilled over.
-----
aux   Opt_SD_LFSS_capacity = MIN(LFSS1_SD_check, LFSS2_SD_check, LFSS3_SD_check,
      LFSS4_SD_check, LFSS5_SD_check, LFSS6_SD_check, LFSS7_SD_check, LFSS8_SD_check)
doc   Opt_SD_LFSS_capacity = This variable finds the LFSS silo that first, can receive a
      SD quantity of 33.33 tons and second, has the most SD quantity among all so as
      finally, to receive full LFSS silos that create more LFSS that are empty and
      available for whatever grade. This is attributed to the fact that the semi-full
      silos or even worse, silos with a small quantity above Smin, are unavailable for
      storing other types of grades.
      If this variable is equal to 500 then this means that there is not any available
      slot in the LFSS silos for SD and the quantity coming out from SSP is spilled over.
-----
aux   Opt_WG_LFSS_capacity = MIN(LFSS1_WG_check, LFSS2_WG_check, LFSS3_WG_check,
      LFSS4_WG_check, LFSS5_WG_check, LFSS6_WG_check, LFSS7_WG_check, LFSS8_WG_check)
doc   Opt_WG_LFSS_capacity = This variable finds the LFSS silo that first, can receive a
      WG quantity of 33.33 tons and second, has the most WG quantity among all so as
      finally, to receive full LFSS silos that create more LFSS that are empty and
      available for whatever grade. This is attributed to the fact that the semi-full
      silos or even worse, silos with a small quantity above Smin, are unavailable for
      storing other types of grades.
      If this variable is equal to 500 then this means that there is not any available
      slot in the LFSS silos for WG and the quantity coming out from SSP is spilled over.
-----

```

```

aux  Order_color_changeover = IF( POLY_Current_Color=1 AND
(Remaining_job_slots_of_WG_to_produce>0 OR
  Remaining_job_slots_of_SD_to_produce>0),
  0,
  IF(POLY_Current_Color=1 AND Remaining_job_slots_of_WG_to_produce=0 AND
    Remaining_job_slots_of_SD_to_produce=0,
    IF( Remaining_job_slots_of_FH_to_produce>0, 1, 0),
    IF( POLY_Current_Color=2,
      0,
      IF( POLY_Current_Color=3 AND Remaining_job_slots_of_FH_to_produce>0,
        0,
        IF( Remaining_job_slots_of_SD_to_produce>0 OR
Remaining_job_slots_of_WG_to_produce>0,
          1,
          0))))))
doc  Order_color_changeover = A color changeover can be initiated whether the POLY is
operational or not.
The event of 2 successive color changeovers is forbidden in practice and should not
be encountered.
If this variable is equal to 0 then no color changeover is initialized.
If this variable is equal to 1, if the var. "color changeover at t-1" equals 0 then
a color changeover is initiated, else no color changeover is initiated.
-----
aux  POLY_Failure = IF(Random_numbers_1<=POLY_Failure_Probability, 0, 1)
doc  POLY_Failure = POLY_Failure=1 is equal to the fact where no failure occurs at the
current 4-hour period (dt).
POLY_Failure=0 is equal to the fact where a failure occurs at the current 4-hour
period (dt).
-----
aux  Random_numbers_1 = RANDOM(0, 1, 0.3)
doc  Random_numbers_1 = Random numbers coming out of a uniform distribution between 0 and
1.
-----
aux  Random_numbers_2 = RANDOM(0, 1, 0.2)
doc  Random_numbers_2 = Random numbers coming out of a uniform distribution between 0 and
1.
-----
aux  S1_FH_BC_dem = IF(LFSS2_grade=4,
  MAX(S2_FH_BC_dem-LFSS2_BC, 0),
  S2_FH_BC_dem)
-----
aux  S1_FH_ST_dem = IF(LFSS2_grade=4,
  MAX(S2_FH_ST_dem-LFSS2_ST, 0),
  S2_FH_ST_dem)
-----
aux  S1_FH_WH_dem = IF(LFSS2_grade=4,
  MAX(S2_FH_WH_dem-LFSS2_BB, 0),
  S2_FH_WH_dem)
-----
aux  S1_G_BC_dem = IF(LFSS2_grade=3,
  MAX(S2_G_BC_dem-LFSS2_BC, 0),
  S2_G_BC_dem)
-----
aux  S1_G_ST_dem = IF(LFSS2_grade=3,
  MAX(S2_G_ST_dem-LFSS2_ST, 0),
  S2_G_ST_dem)
-----
aux  S1_G_WH_dem = IF(LFSS2_grade=3,
  MAX(S2_G_WH_dem-LFSS2_BB, 0),
  S2_G_WH_dem)
-----
aux  S1_SD_BC_dem = IF(LFSS2_grade=2,
  MAX(S2_SD_BC_dem-LFSS2_BC, 0),
  S2_SD_BC_dem)
-----
aux  S1_SD_ST_dem = IF(LFSS2_grade=2,
  MAX(S2_SD_ST_dem-LFSS2_ST, 0),
  S2_SD_ST_dem)
-----
aux  S1_SD_WH_dem = IF(LFSS2_grade=2,
  MAX(S2_SD_WH_dem-LFSS2_BB, 0),
  S2_SD_WH_dem)
-----
aux  S1_WG_BC_dem = IF(LFSS2_grade=1,
  MAX(S2_WG_BC_dem-LFSS2_BC, 0),
  S2_WG_BC_dem)
-----

```

```

aux   S1_WG_ST_dem = IF(LFSS2_grade=1,
      MAX(S2_WG_ST_dem-LFSS2_ST, 0),
      S2_WG_ST_dem)
-----
aux   S1_WG_WH_dem = IF(LFSS2_grade=1,
      MAX(S2_WG_WH_dem-LFSS2_BB, 0),
      S2_WG_WH_dem)
-----
aux   S2_FH_BC_dem = IF(LFSS3_grade=4,
      MAX(S3_FH_BC_dem-LFSS3_BC, 0),
      S3_FH_BC_dem)
-----
aux   S2_FH_ST_dem = IF(LFSS3_grade=4,
      MAX(S3_FH_ST_dem-LFSS3_ST, 0),
      S3_FH_ST_dem)
-----
aux   S2_FH_WH_dem = IF(LFSS3_grade=4,
      MAX(S3_FH_WH_dem-LFSS3_BB, 0),
      S3_FH_WH_dem)
-----
aux   S2_G_BC_dem = IF(LFSS3_grade=3,
      MAX(S3_G_BC_dem-LFSS3_BC, 0),
      S3_G_BC_dem)
-----
aux   S2_G_ST_dem = IF(LFSS3_grade=3,
      MAX(S3_G_ST_dem-LFSS3_ST, 0),
      S3_G_ST_dem)
-----
aux   S2_G_WH_dem = IF(LFSS3_grade=3,
      MAX(S3_G_WH_dem-LFSS3_BB, 0),
      S3_G_WH_dem)
-----
aux   S2_SD_BC_dem = IF(LFSS3_grade=2,
      MAX(S3_SD_BC_dem-LFSS3_BC, 0),
      S3_SD_BC_dem)
-----
aux   S2_SD_ST_dem = IF(LFSS3_grade=2,
      MAX(S3_SD_ST_dem-LFSS3_ST, 0),
      S3_SD_ST_dem)
-----
aux   S2_SD_WH_dem = IF(LFSS3_grade=2,
      MAX(S3_SD_WH_dem-LFSS3_BB, 0),
      S3_SD_WH_dem)
-----
aux   S2_WG_BC_dem = IF(LFSS3_grade=1,
      MAX(S3_WG_BC_dem-LFSS3_BC, 0),
      S3_WG_BC_dem)
-----
aux   S2_WG_ST_dem = IF(LFSS3_grade=1,
      MAX(S3_WG_ST_dem-LFSS3_ST, 0),
      S3_WG_ST_dem)
-----
aux   S2_WG_WH_dem = IF(LFSS3_grade=1,
      MAX(S3_WG_WH_dem-LFSS3_BB, 0),
      S3_WG_WH_dem)
-----
aux   S3_FH_BC_dem = IF(LFSS4_grade=4,
      MAX(S4_FH_BC_dem-LFSS4_BC, 0),
      S4_FH_BC_dem)
-----
aux   S3_FH_ST_dem = IF(LFSS4_grade=4,
      MAX(S4_FH_ST_dem-LFSS4_ST, 0),
      S4_FH_ST_dem)
-----
aux   S3_FH_WH_dem = IF(LFSS4_grade=4,
      MAX(S4_FH_WH_dem-LFSS4_BB, 0),
      S4_FH_WH_dem)
-----
aux   S3_G_BC_dem = IF(LFSS4_grade=3,
      MAX(S4_G_BC_dem-LFSS4_BC, 0),
      S4_G_BC_dem)
-----
aux   S3_G_ST_dem = IF(LFSS4_grade=3,
      MAX(S4_G_ST_dem-LFSS4_ST, 0),
      S4_G_ST_dem)
-----
aux   S3_G_WH_dem = IF(LFSS4_grade=3,

```

```

MAX(S4_G_WH_dem-LFSS4_BB, 0),
S4_G_WH_dem)
aux S3_SD_BC_dem = IF(LFSS4_grade=2,
MAX(S4_SD_BC_dem-LFSS4_BC, 0),
S4_SD_BC_dem)
aux S3_SD_ST_dem = IF(LFSS4_grade=2,
MAX(S4_SD_ST_dem-LFSS4_ST, 0),
S4_SD_ST_dem)
aux S3_SD_WH_dem = IF(LFSS4_grade=2,
MAX(S4_SD_WH_dem-LFSS4_BB, 0),
S4_SD_WH_dem)
aux S3_WG_BC_dem = IF(LFSS4_grade=1,
MAX(S4_WG_BC_dem-LFSS4_BC, 0),
S4_WG_BC_dem)
aux S3_WG_ST_dem = IF(LFSS4_grade=1,
MAX(S4_WG_ST_dem-LFSS4_ST, 0),
S4_WG_ST_dem)
aux S3_WG_WH_dem = IF(LFSS4_grade=1,
MAX(S4_WG_WH_dem-LFSS4_BB, 0),
S4_WG_WH_dem)
aux S4_FH_BC_dem = IF(LFSS5_grade=4,
MAX(S5_FH_BC_dem-LFSS5_BC, 0),
S5_FH_BC_dem)
aux S4_FH_ST_dem = IF(LFSS5_grade=4,
MAX(S5_FH_ST_dem-LFSS5_ST, 0),
S5_FH_ST_dem)
aux S4_FH_WH_dem = IF(LFSS5_grade=4,
MAX(S5_FH_WH_dem-LFSS5_BB, 0),
S5_FH_WH_dem)
aux S4_G_BC_dem = IF(LFSS5_grade=3,
MAX(S5_G_BC_dem-LFSS5_BC, 0),
S5_G_BC_dem)
aux S4_G_ST_dem = IF(LFSS5_grade=3,
MAX(S5_G_ST_dem-LFSS5_ST, 0),
S5_G_ST_dem)
aux S4_G_WH_dem = IF(LFSS5_grade=3,
MAX(S5_G_WH_dem-LFSS5_BB, 0),
S5_G_WH_dem)
aux S4_SD_BC_dem = IF(LFSS5_grade=2,
MAX(S5_SD_BC_dem-LFSS5_BC, 0),
S5_SD_BC_dem)
aux S4_SD_ST_dem = IF(LFSS5_grade=2,
MAX(S5_SD_ST_dem-LFSS5_ST, 0),
S5_SD_ST_dem)
aux S4_SD_WH_dem = IF(LFSS5_grade=2,
MAX(S5_SD_WH_dem-LFSS5_BB, 0),
S5_SD_WH_dem)
aux S4_WG_BC_dem = IF(LFSS5_grade=1,
MAX(S5_WG_BC_dem-LFSS5_BC, 0),
S5_WG_BC_dem)
aux S4_WG_ST_dem = IF(LFSS5_grade=1,
MAX(S5_WG_ST_dem-LFSS5_ST, 0),
S5_WG_ST_dem)
aux S4_WG_WH_dem = IF(LFSS5_grade=1,
MAX(S5_WG_WH_dem-LFSS5_BB, 0),
S5_WG_WH_dem)
aux S5_FH_BC_dem = IF(LFSS6_grade=4,
MAX(S6_FH_BC_dem-LFSS6_BC, 0),

```



```

S6_FH_BC_dem)
aux  S5_FH_ST_dem = IF(LFSS6_grade=4,
    MAX(S6_FH_ST_dem-LFSS6_ST, 0),
    S6_FH_ST_dem)
aux  S5_FH_WH_dem = IF(LFSS6_grade=4,
    MAX(S6_FH_WH_dem-LFSS6_BB, 0),
    S6_FH_WH_dem)
aux  S5_G_BC_dem = IF(LFSS6_grade=3,
    MAX(S6_G_BC_dem-LFSS6_BC, 0),
    S6_G_BC_dem)
aux  S5_G_ST_dem = IF(LFSS6_grade=3,
    MAX(S6_G_ST_dem-LFSS6_ST, 0),
    S6_G_ST_dem)
aux  S5_G_WH_dem = IF(LFSS6_grade=3,
    MAX(S6_G_WH_dem-LFSS6_BB, 0),
    S6_G_WH_dem)
aux  S5_SD_BC_dem = IF(LFSS6_grade=2,
    MAX(S6_SD_BC_dem-LFSS6_BC, 0),
    S6_SD_BC_dem)
aux  S5_SD_ST_dem = IF(LFSS6_grade=2,
    MAX(S6_SD_ST_dem-LFSS6_ST, 0),
    S6_SD_ST_dem)
aux  S5_SD_WH_dem = IF(LFSS6_grade=2,
    MAX(S6_SD_WH_dem-LFSS6_BB, 0),
    S6_SD_WH_dem)
aux  S5_WG_BC_dem = IF(LFSS6_grade=1,
    MAX(S6_WG_BC_dem-LFSS6_BC, 0),
    S6_WG_BC_dem)
aux  S5_WG_ST_dem = IF(LFSS6_grade=1,
    MAX(S6_WG_ST_dem-LFSS6_ST, 0),
    S6_WG_ST_dem)
aux  S5_WG_WH_dem = IF(LFSS6_grade=1,
    MAX(S6_WG_WH_dem-LFSS6_BB, 0),
    S6_WG_WH_dem)
aux  S6_FH_BC_dem = IF(LFSS7_grade=4,
    MAX(S7_FH_BC_dem-LFSS7_BC, 0),
    S7_FH_BC_dem)
aux  S6_FH_ST_dem = IF(LFSS7_grade=4,
    MAX(S7_FH_ST_dem-LFSS7_ST, 0),
    S7_FH_ST_dem)
aux  S6_FH_WH_dem = IF(LFSS7_grade=4,
    MAX(S7_FH_WH_dem-LFSS7_BB, 0),
    S7_FH_WH_dem)
aux  S6_G_BC_dem = IF(LFSS7_grade=3,
    MAX(S7_G_BC_dem-LFSS7_BC, 0),
    S7_G_BC_dem)
aux  S6_G_ST_dem = IF(LFSS7_grade=3,
    MAX(S7_G_ST_dem-LFSS7_ST, 0),
    S7_G_ST_dem)
aux  S6_G_WH_dem = IF(LFSS7_grade=3,
    MAX(S7_G_WH_dem-LFSS7_BB, 0),
    S7_G_WH_dem)
aux  S6_SD_BC_dem = IF(LFSS7_grade=2,
    MAX(S7_SD_BC_dem-LFSS7_BC, 0),
    S7_SD_BC_dem)
aux  S6_SD_ST_dem = IF(LFSS7_grade=2,
    MAX(S7_SD_ST_dem-LFSS7_ST, 0),
    S7_SD_ST_dem)

```

```

aux   S6_SD_WH_dem = IF(LFSS7_grade=2,
      MAX(S7_SD_WH_dem-LFSS7_BB, 0),
      S7_SD_WH_dem)

aux   S6_WG_BC_dem = IF(LFSS7_grade=1,
      MAX(S7_WG_BC_dem-LFSS7_BC, 0),
      S7_WG_BC_dem)

aux   S6_WG_ST_dem = IF(LFSS7_grade=1,
      MAX(S7_WG_ST_dem-LFSS7_ST, 0),
      S7_WG_ST_dem)

aux   S6_WG_WH_dem = IF(LFSS7_grade=1,
      MAX(S7_WG_WH_dem-LFSS7_BB, 0),
      S7_WG_WH_dem)

aux   S7_FH_BC_dem = IF(LFSS8_grade=4,
      MAX(FH_BC_demand-LFSS8_BC, 0),
      FH_BC_demand)

aux   S7_FH_ST_dem = IF(LFSS8_grade=4,
      MAX(FH_ST_demand-LFSS8_ST, 0),
      FH_ST_demand)

aux   S7_FH_WH_dem = IF(LFSS8_grade=4,
      MAX(S8_FH_WH_dem-LFSS8_BB, 0),
      S8_FH_WH_dem)

aux   S7_G_BC_dem = IF(LFSS8_grade=3,
      MAX(G_BC_demand-LFSS8_BC, 0),
      G_BC_demand)

aux   S7_G_ST_dem = IF(LFSS8_grade=3,
      MAX(G_ST_demand-LFSS8_ST, 0),
      G_ST_demand)

aux   S7_G_WH_dem = IF(LFSS8_grade=3,
      MAX(S8_G_WH_dem-LFSS8_BB, 0),
      S8_G_WH_dem)

aux   S7_SD_BC_dem = IF(LFSS8_grade=2,
      MAX(SD_BC_demand-LFSS8_BC, 0),
      SD_BC_demand)

aux   S7_SD_ST_dem = IF(LFSS8_grade=2,
      MAX(SD_ST_demand-LFSS8_ST, 0),
      SD_ST_demand)

aux   S7_SD_WH_dem = IF(LFSS8_grade=2,
      MAX(S8_SD_WH_dem-LFSS8_BB, 0),
      S8_SD_WH_dem)

aux   S7_WG_BC_dem = IF(LFSS8_grade=1,
      MAX(WG_BC_demand-LFSS8_BC, 0),
      WG_BC_demand)

aux   S7_WG_ST_dem = IF(LFSS8_grade=1,
      MAX(WG_ST_demand-LFSS8_ST, 0),
      WG_ST_demand)

aux   S7_WG_WH_dem = IF(LFSS8_grade=1,
      MAX(S8_WG_WH_dem-LFSS8_BB, 0),
      S8_WG_WH_dem)

aux   S8_FH_WH_dem = IF(FH_WH_stock<FH_WH_safety_stock,
      FH_WH_safety_stock-FH_WH_stock,
      0)

aux   S8_G_WH_dem = IF(G_WH_stock<G_WH_safety_stock,
      G_WH_safety_stock-G_WH_stock,
      0)

aux   S8_SD_WH_dem = IF(SD_WH_stock<SD_WH_safety_stock,
      SD_WH_safety_stock-SD_WH_stock,
      0)

```

```

aux   S8_WG_WH_dem = IF(WG_WH_stock<WG_WH_safety_stock,
                    WG_WH_safety_stock-WG_WH_stock,
                    0)

aux   SD_LFSS_s1 = (1-K1)*SD_LFSS_safety_stock
aux   SD_LFSS_s2 = (1+K1)*SD_LFSS_safety_stock
aux   SD_LFSS_stock = IF(LFSS1_grade=2, LFSS1, 0) + IF(LFSS2_grade=2, LFSS2, 0) +
                    IF(LFSS3_grade=2, LFSS3, 0) + IF(LFSS4_grade=2, LFSS4, 0) +
                    IF(LFSS5_grade=2, LFSS5, 0) + IF(LFSS6_grade=2, LFSS6, 0) +
                    IF(LFSS7_grade=2, LFSS7, 0) + IF(LFSS8_grade=2, LFSS8, 0)
doc   SD_LFSS_stock = Quantity of SD stored in the LFSS.

aux   SILO_1_slot_availability_CHECK = IF(Outflow_silo_1>0,
    IF(Slot_1=0, 1, IF(Slot_1>0 AND Slot_2=0, 1, IF(Slot_2>0 AND Slot_3=0, 2,
    IF(Slot_3>0 AND Slot_4=0, 3, IF(Slot_4>0 AND Slot_5=0, 4, IF(Slot_5>0 AND Slot_6=0,
    5, IF(Slot_6>0 AND Slot_7=0, 6, IF(Slot_7>0 AND Slot_8=0, 7, IF(Slot_8>0 AND
    Slot_9=0, 8, IF(Slot_9>0 AND Slot_10=0, 9, IF(Slot_10>0 AND Slot_11=0, 10,
    IF(Slot_11>0 AND Slot_12=0, 11, 12))))))))),
    IF(Slot_1=0, 1, IF(Slot_1>0 AND Slot_2=0, 2, IF(Slot_2>0 AND Slot_3=0, 3,
    IF(Slot_3>0 AND Slot_4=0, 4, IF(Slot_4>0 AND Slot_5=0, 5, IF(Slot_5>0 AND Slot_6=0,
    6, IF(Slot_6>0 AND Slot_7=0, 7, IF(Slot_7>0 AND Slot_8=0, 8, IF(Slot_8>0 AND
    Slot_9=0, 9, IF(Slot_9>0 AND Slot_10=0, 10, IF(Slot_10>0 AND Slot_11=0, 11,
    IF(Slot_11>0 AND Slot_12=0, 12, 100))))))))),
doc   SILO_1_slot_availability_CHECK = By this variable we check whether SILO 1 can
    receive semi products coming out of POLY and at which slots, or not. The check-rule
    is diversified by whether SILO's 1 outflow is positive or 0.

aux   SILO_2_slot_availability_CHECK = IF(SILO_1_slot_availability_CHECK=100,
    IF(Outflow_silo_2>0,
    IF(Slot_13=0, 13, IF(Slot_13>0 AND Slot_14=0, 13, IF(Slot_14>0 AND Slot_15=0, 14,
    IF(Slot_15>0 AND Slot_16=0, 15, IF(Slot_16>0 AND Slot_17=0, 16, IF(Slot_17>0 AND
    Slot_18=0, 17, IF(Slot_18>0 AND Slot_19=0, 18, IF(Slot_19>0 AND Slot_20=0, 19,
    IF(Slot_20>0 AND Slot_21=0, 20, IF(Slot_21>0 AND Slot_22=0, 21, IF(Slot_22>0 AND
    Slot_23=0, 22, IF(Slot_23>0 AND Slot_24=0, 23, 24))))))))),
    IF(Slot_13=0, 13, IF(Slot_13>0 AND Slot_14=0, 14, IF(Slot_14>0 AND Slot_15=0, 15,
    IF(Slot_15>0 AND Slot_16=0, 16, IF(Slot_16>0 AND Slot_17=0, 17, IF(Slot_17>0 AND
    Slot_18=0, 18, IF(Slot_18>0 AND Slot_19=0, 19, IF(Slot_19>0 AND Slot_20=0, 20,
    IF(Slot_20>0 AND Slot_21=0, 21, IF(Slot_21>0 AND Slot_22=0, 22, IF(Slot_22>0 AND
    Slot_23=0, 23, IF(Slot_23>0 AND Slot_24=0, 24, 100))))))))),
doc   SILO_2_slot_availability_CHECK = By this variable we check whether SILO 2 can
    receive semi products coming out of POLY and at which slots, or not. The check-rule
    is diversified by whether SILO's 2 outflow is positive or 0.

aux   SILO_3_slot_availability_CHECK = IF(SILO_2_slot_availability_CHECK=100,
    IF(Outflow_silo_3>0,
    IF(Slot_25=0, 25, IF(Slot_25>0 AND Slot_26=0, 25, IF(Slot_26>0 AND Slot_27=0, 26,
    IF(Slot_27>0 AND Slot_28=0, 27, IF(Slot_28>0 AND Slot_29=0, 28, IF(Slot_29>0 AND
    Slot_30=0, 29, IF(Slot_30>0 AND Slot_31=0, 30, IF(Slot_31>0 AND Slot_32=0, 31,
    IF(Slot_32>0 AND Slot_33=0, 32, IF(Slot_33>0 AND Slot_34=0, 33, IF(Slot_34>0 AND
    Slot_35=0, 34, IF(Slot_35>0 AND Slot_36=0, 35, 36))))))))),
    IF(Slot_25<1, 25, IF(Slot_25>0 AND Slot_26=0, 26, IF(Slot_26>0 AND Slot_27=0, 27,
    IF(Slot_27>0 AND Slot_28=0, 28, IF(Slot_28>0 AND Slot_29=0, 29, IF(Slot_29>0 AND
    Slot_30=0, 30, IF(Slot_30>0 AND Slot_31=0, 31, IF(Slot_31>0 AND Slot_32=0, 32,
    IF(Slot_32>0 AND Slot_33=0, 33, IF(Slot_33>0 AND Slot_34=0, 34, IF(Slot_34>0 AND
    Slot_35=0, 35, IF(Slot_35>0 AND Slot_36=0, 36, 100))))))))),
doc   SILO_3_slot_availability_CHECK = By this variable we check whether SILO 3 can
    receive semi products coming out of POLY and at which slots, or not. The check-rule
    is diversified by whether SILO's 3 outflow is positive or 0.

aux   SSP_Failure = IF(Random_numbers_2<SSP_Failure_Probability, 0, 1)
doc   SSP_Failure = This variable represents the randomness of an SSP failure.
    SSP_Failure=0 is equal to the fact where a failure occurs at the current 4-hour
    period (dt).
    SSP_Failure=1 is equal to the fact where no failure occurs at the current 4-hour
    period (dt).

aux   Total_lost_demand
    Total_lost_FH_BB_demand+Total_lost_FH_BC_demand+Total_lost_FH_ST_demand+Total_lost_G
    _BB_demand+Total_lost_G_BC_demand+Total_lost_G_ST_demand+Total_lost_SD_BB_demand+Tot
    al_lost_SD_BC_demand+Total_lost_SD_ST_demand+Total_lost_WG_BB_demand+Total_lost_WG_B
    C_demand+Total_lost_WG_ST_demand

aux   TSS_availability_CHECK = IF( (Slot_12>0 AND Slot_24>0 AND Slot_36>0) AND
    (Outflow_silo_1=0 AND Outflow_silo_2=0 AND Outflow_silo_3=0), 0, 1)

```



```

-----
const WG_BB_demand = 0
doc WG_BB_demand = This variable represents the big bags demand of WG in the t(th) 4-
hour period.

-----
const WG_BC_demand = 0
doc WG_BC_demand = This variable represents the bulk containers demand of WG in the
t(th) 4-hour period.

-----
const WG_ST_demand = 0
doc WG_ST_demand = This variable represents the silo trucks demand of WG in the t(th) 4-
hour period.

-----
const FH_LFSS_safety_stock = 450
doc FH_LFSS_safety_stock = Minimum quantity of FH stored in the LFSS so as not to hit
the alarm for low FH LFSS stock level. This safety stock quantity stems from the
solution of the SLSP problem, presented in Liberopoulos et al. 2009.

-----
const FH_WH_safety_stock = 450
doc FH_WH_safety_stock = Minimum quantity of FH stored in the warehouse so as not to hit
the alarm for low FH warehouse stock level. This safety stock quantity stems from
the solution of the SLSP problem, presented in Liberopoulos et al. 2009.

-----
const G_LFSS_safety_stock = 0
doc G_LFSS_safety_stock = Minimum quantity of WG stored in the LFSS so as not to hit the
alarm for low WG LFSS stock level. This safety stock quantity stems from the
solution of the SLSP problem, presented in Liberopoulos et al. 2009.
G is an off specification by-product sold at a lower price

-----
const G_WH_safety_stock = 0
doc G_WH_safety_stock = Minimum quantity of G stored in the warehouse so as not to hit
the alarm for low G warehouse stock level. This safety stock quantity stems from the
solution of the SLSP problem, presented in Liberopoulos et al. 2009.

-----
const K1 = 11%

-----
const POLY_Failure_Probability = 0
doc POLY_Failure_Probability = This constant shows the mean value of the long run
probability for having a POLY failure in each individual 4-hour period

-----
const SD_LFSS_safety_stock = 0
doc SD_LFSS_safety_stock = Minimum quantity of WG stored in the LFSS so as not to hit
the alarm for low WG LFSS stock level. This safety stock quantity stems from the
solution of the SLSP problem, presented in Liberopoulos et al. 2009.
SD is currently sold only in big-bags due to the clients' requirements.

-----
const SD_WH_safety_stock = 880
doc SD_WH_safety_stock = Minimum quantity of SD stored in the warehouse so as not to hit
the alarm for low SD warehouse stock level. This safety stock quantity stems from
the solution of the SLSP problem, presented in Liberopoulos et al. 2009.

-----
const Smax = 430
doc Smax = Capacity of a LFSS silo.

-----
const Smin = 1
doc Smin = Minimum quantity contained in an LFSS silo so as not to be considered empty
for putting in a new (or the same) grade.

-----
const SSP_Failure_Probability = 0
doc SSP_Failure_Probability = This constant shows the mean value of the long run
probability for having an SSP failure in each individual 4-hour period

-----
const Total_Demand = 32683.21

-----
const Ubb = 40
doc Ubb = Maximum quantity of material that can be sacked from LFSS into big bags in one
period

-----
const Ubc = 69.2
doc Ubc = Maximum quantity of material that can be loaded from LFSS into a bulk
container in one period

-----
const Ust = 224
doc Ust = Maximum quantity of material that can be loaded from LFSS into a silo truck in
one period.

-----
const Warehouse_capacity = 3500

```



```
doc    Warehouse_capacity = Maximum warehouse capacity in tons of final grades.
-----
const  WG_LFSS_safety_stock = 450
doc    WG_LFSS_safety_stock = Minimum quantity of WG stored in the LFSS so as not to hit
       the alarm for low WG LFSS stock level. This safety stock quantity stems from the
       solution of the SLSP problem, presented in Liberopoulos et al. 2009.
-----
const  WG_WH_safety_stock = 250
doc    WG_WH_safety_stock = Minimum quantity of WG stored in the warehouse so as not to hit
       the alarm for low WG warehouse stock level. This safety stock quantity stems from
       the solution of the SLSP problem, presented in Liberopoulos et al. 2009.
```



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ



004000073724

