



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

Υβριδικοί αλγόριθμοι διαφορικής εξέλιξης στο πρόβλημα  
προγραμματισμός συστημάτων παραγωγής συνεχούς ροής

Εφαρμογή με πραγματικά δεδομένα εταιρίας με ευέλικτο  
σύστημα παραγωγής συνεχούς ροής

---

Διπλωματική Εργασία

ΩΡΑΙΟΖΗΛΗ ΔΕΠΟΥΝΤΗ

ΕΠΙΒΛΕΠΩΝ: ΙΩΑΝΝΗΣ ΜΑΡΙΝΑΚΗΣ

ΟΚΤΩΒΡΙΟΣ 2012

ΧΑΝΙΑ

## **ΠΕΡΙΛΗΨΗ**

Η παρούσα διπλωματική εργασία πραγματεύεται τον προγραμματισμό συστημάτων και ευέλικτων συστημάτων παραγωγής συνεχούς ροής με χρήση υβριδικών αλγορίθμων Διαφορικής Εξέλιξης (ΔΕ). Αρχικά αποσαφηνίζονται βασικές έννοιες που αφορούν τους Μεθευρετικούς Αλγορίθμους καθώς και τον προγραμματισμό συστημάτων παραγωγής συνεχούς ροής. Το θεωρητικό υπόβαθρο ακολουθεί μία σύντομη βιβλιογραφική ανασκόπηση αφού πρώτα έχουν αναλυθεί οι αλγόριθμοι που χρησιμοποιούνται στην εργασία. Έπειτα, εφαρμόζονται ο αλγόριθμος της ΔΕ με διάφορες παραλλαγές και υβριδικοί αλγόριθμοι ΔΕ σε δεδομένα τα οποία έχουν αντληθεί από τη βιβλιογραφία και συγκρίνονται τα αποτελέσματά τους. Στη συνέχεια περιγράφεται και μοντελοποιείται το σύστημα παραγωγής μίας εταιρίας καλλυντικών ως ευέλικτο σύστημα παραγωγής συνεχούς ροής με διαφορετικούς περιορισμούς σε κάθε στάδιο και ακολουθεί η πρακτική εφαρμογή των αλγορίθμων σε πραγματικά δεδομένα τα οποία έχουν συλλεχθεί από την εταιρία. Τέλος παρουσιάζονται τα συμπεράσματα όπως προέκυψαν από την εργασία και ακολουθεί ο σχολιασμός αυτών.

## **ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**

Μεθευρετικοί Αλγόριθμοι, Υβριδικοί Αλγόριθμοι, Αλγόριθμος Διαφορικής Εξέλιξης, Χρονικός Προγραμματισμός Εργασιών, Βελτιστοποίηση Παραγωγής, Σύστημα Παραγωγής Συνεχούς Ροής



TECHNICAL UNIVERSITY OF CRETE  
DEPARTMENT OF PRODUCTION ENGINEERING AND MANAGEMENT

# Hybrid differential evolution algorithms for the flow-shop scheduling problem

## A real life application for a flexible flow-shop scheduling problem

---

Diploma Thesis

ORAIOSILI DEPOUNTI

SUPERVISOR: IOANNIS MARINAKIS

OCTOBER 2012

CHANIA

## **ABSTRACT**

The present diploma thesis deals with the scheduling of flow-shop and flexible flow-shop using hybrid differential evolution (DE) algorithms. First some basic concepts about metaheuristics and flow-shops are clarified and then the algorithms which are implemented in the current work are explained. The theoretical background follows a brief literature review on flow-shop and flexible flow-shop scheduling using metaheuristics. Furthermore different versions of DE algorithm and hybrid DE algorithms are applied for literature data and the comparison of their results is presented. Thereafter the production system of a cosmetics production company is described as flexible flow-shop with different constraints in each stage and an application of collected data is carried out. Finally the conclusions from the whole work are presented followed by an overall discussion.

## **KEY WORDS**

Metaheuristics, Hybrid Algorithms, Differential Evolution Algorithm, Production Optimization, Flowshop Scheduling, Flexible Flowshop Scheduling

## ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ .....	11
1. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ .....	13
1.1. ΠΡΟΒΛΗΜΑΤΑ ΧΡΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ .....	13
1.1.1. ΣΥΣΤΗΜΑ ΠΑΡΑΓΩΓΗΣ ΣΥΝΕΧΟΥΣ ΡΟΗΣ .....	15
1.1.2. ΕΥΕΛΙΚΤΟ ΣΥΣΤΗΜΑ ΠΑΡΑΓΩΓΗΣ ΣΥΝΕΧΟΥΣ ΡΟΗΣ .....	20
1.2. ΕΥΡΕΤΙΚΟΙ - ΜΕΘΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ .....	23
1.2.1. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ .....	27
1.2.2. ΑΛΓΟΡΙΘΜΟΙ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ .....	32
1.2.3. ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ .....	33
1.2.4. ΑΛΓΟΡΙΘΜΟΣ ΕΠΑΝΑΛΗΠΤΙΚΗΣ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ .....	35
1.2.5 ΑΛΓΟΡΙΘΜΟΣ ΜΕΤΑΒΛΗΤΗΣ ΓΕΙΤΟΝΙΑΣ ΑΝΑΖΗΤΗΣΗΣ .....	36
2. ΣΥΝΤΟΜΗ ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ .....	37
3. ΕΦΑΡΜΟΓΗ ΣΕ ΒΙΒΛΙΟΓΡΑΦΙΚΑ ΔΕΔΟΜΕΝΑ .....	41
3.1. ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΟΥ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ .....	41
3.1.1. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΛΗΘΥΣΜΟΥ .....	41
3.1.2. ΑΞΙΟΛΟΓΗΣΗ ΑΝΤΙΚΕΙΜΕΝΙΚΗΣ ΣΥΝΑΡΤΗΣΗΣ .....	43
3.1.3. ΜΕΤΑΛΛΑΞΗ, ΔΙΑΣΤΑΥΡΩΣΗ, ΕΠΙΛΟΓΗ .....	44
3.1.4. ΜΕΤΑΤΡΟΠΗ ΣΥΝΕΧΩΝ ΤΙΜΩΝ ΣΕ ΔΙΑΚΡΙΤΕΣ ΚΑΙ ΤΟ ΑΝΤΙΣΤΡΟΦΟ .....	47
3.1.5. ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ ΚΑΙ ΣΤΡΑΤΗΓΙΚΗΣ .....	49
3.1.6. ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ .....	60
3.2. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ .....	61
3.2.1. ΣΥΓΚΡΙΣΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΩΝ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ .....	63

3.2.2. ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ .....	65
3.2.3 ΠΕΡΑΙΤΕΡΩ ΔΙΕΡΕΥΝΗΣΗ .....	67
3.3. ΥΒΡΙΔΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ .....	68
3.3.1. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ .....	69
3.3.2. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΕΠΑΝΑΛΗΠΤΙΚΗ ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ .....	73
3.3.3. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΑΛΛΑΓΗ ΓΕΙΤΟΝΙΑΣ ΑΝΑΖΗΤΗΣΗΣ .....	76
3.3.4. ΣΥΓΚΡΙΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΩΝ .....	78
4. ΕΦΑΡΜΟΓΗ ΣΕ ΠΡΑΓΜΑΤΙΚΑ ΔΕΔΟΜΕΝΑ .....	82
4.1. ΥΠΟΔΟΜΗ ΤΗΣ ΠΡΟΣ ΜΕΛΕΤΗ ΕΤΑΙΡΙΑΣ .....	83
4.2. ΔΙΑΔΙΚΑΣΙΑ ΠΑΡΑΓΩΓΗΣ .....	84
4.3. ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ .....	89
4.3.1. ΣΤΑΔΙΟ ΠΑΡΑΓΩΓΗΣ BULK .....	90
4.3.2. ΣΤΑΔΙΟ ΜΙΚΡΟΒΙΟΛΟΓΙΚΟΥ ΕΛΕΓΧΟΥ .....	91
4.3.3. ΣΤΑΔΙΟ ΣΥΣΚΕΥΑΣΙΑΣ .....	92
4.3.4. ΔΕΔΟΜΕΝΑ .....	93
4.3.5. ΖΗΤΟΥΜΕΝΟ ΚΑΙ ΚΡΙΤΗΡΙΟ ΠΡΟΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ .....	95
4.4. ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΩΝ – ΣΥΓΚΡΙΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ .....	100
5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ .....	102
ΑΝΑΦΟΡΕΣ .....	105
ΠΑΡΑΡΤΗΜΑ Ι .....	117
ΠΑΡΑΡΤΗΜΑ ΙΙ .....	123
ΠΑΡΑΡΤΗΜΑ ΙΙΙ .....	127

ΠΑΡΑΡΤΗΜΑ IV .....	131
--------------------	-----

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ 1. ΚΡΙΤΗΡΙΑ ΧΡΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΕ ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΓΩΓΗΣ .....	14
ΠΙΝΑΚΑΣ 2. ΧΡΟΝΟΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΙΑ FSSP ΤΕΣΣΑΡΩΝ ΜΗΧΑΝΩΝ ΚΑΙ ΤΕΣΣΑΡΩΝ ΕΡΓΑΣΙΩΝ .....	17
ΠΙΝΑΚΑΣ 3. ΣΤΡΑΤΗΓΙΚΕΣ ΑΛΓΟΡΙΘΜΟΥ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ .....	31
ΠΙΝΑΚΑΣ 4. ΕΠΙΛΟΓΗ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DE ΓΙΑ ΤΟ FSSP ΣΤΗ ΒΙΒΛΙΟΓΡΑΦΙΑ .....	51
ΠΙΝΑΚΑΣ 5. ΣΥΓΚΡΙΣΗ ΤΩΝ ΔΕΚΑ ΣΤΡΑΤΗΓΙΚΩΝ ΤΟΥ DE ΓΙΑ ΤΑ ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ TAILLARD .....	52
ΠΙΝΑΚΑΣ 6. ΣΥΓΚΡΙΣΗ ΤΩΝ ΔΕΚΑ ΣΤΡΑΤΗΓΙΚΩΝ ΤΟΥ DE ΓΙΑ ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ TAILLARD ΜΕ ΠΑΡΑΜΕΤΡΟΥΣ $F = 0.3$ , $CR = 0.9$ , $NP = 150$ , $G_{MAX} = 50$ .....	54
ΠΙΝΑΚΑΣ 7. ΣΥΝΔΥΑΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ ΓΙΑ ΚΑΘΕ ΣΤΡΑΤΗΓΙΚΗ ΣΕ ΠΡΟΒΛΗΜΑΤΑ 20x5 .....	56
ΠΙΝΑΚΑΣ 8. ΣΥΝΔΥΑΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ ΓΙΑ ΚΑΘΕ ΣΤΡΑΤΗΓΙΚΗ ΣΕ ΠΡΟΒΛΗΜΑΤΑ 50x5 .....	57
ΠΙΝΑΚΑΣ 9. ΣΥΝΔΥΑΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ ΓΙΑ ΚΑΘΕ ΣΤΡΑΤΗΓΙΚΗ ΣΕ ΠΡΟΒΛΗΜΑΤΑ 100x5 .....	57
ΠΙΝΑΚΑΣ 10. ΑΠΟΚΛΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ DE ΜΕ ΤΑ ΑΝΩΤΕΡΑ ΟΡΙΑ ΤΟΥ TAILLARD .....	60
ΠΙΝΑΚΑΣ 11. ΜΕΣΗ ΤΙΜΗ MAKESPAN ΓΙΑ ΤΡΕΙΣ ΔΙΑΦΟΡΕΤΙΚΟΥΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥΣ LS ΣΕ ΜΙΚΡΑ ΚΑΙ ΜΕΣΑΙΑ ΠΡΟΒΛΗΜΑΤΑ ΤΟΥ TAILLARD .....	64
ΠΙΝΑΚΑΣ 12. ΑΠΟΚΛΙΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ DELS ΜΕ ΤΑ ΑΝΩΤΕΡΑ ΟΡΙΑ ΤΟΥ TAILLARD .....	65
ΠΙΝΑΚΑΣ 13. ΚΑΛΑ ΑΠΟΤΕΛΕΣΜΑΤΑ 5 ΜΗΧΑΝΩΝ .....	66
ΠΙΝΑΚΑΣ 14. ΒΕΛΤΙΩΣΗ DE ΜΕ ΕΦΑΡΜΟΓΗ LS ΤΕΧΝΙΚΗΣ .....	67
ΠΙΝΑΚΑΣ 15. ΣΥΓΚΡΙΣΗ ΑΠΟΚΛΙΣΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΩΝ DE, DELS, DESA ΜΕ ΤΑ ΑΝΩΤΕΡΑ ΟΡΙΑ ΤΟΥ TAILLARD .....	78
ΠΙΝΑΚΑΣ 16. ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ DESA ΠΟΥ ΠΡΟΣΕΓΓΙΖΟΥΝ ΤΑ ΑΝΩΤΕΡΑ ΟΡΙΑ ΤΟΥ TAILLARD .....	80
ΠΙΝΑΚΑΣ 17. ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ DEVNS ΠΟΥ ΠΡΟΣΕΓΓΙΖΟΥΝ ΤΑ ΑΝΩΤΕΡΑ ΟΡΙΑ ΤΟΥ TAILLARD .....	80
ΠΙΝΑΚΑΣ 18. ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ DEILS ΠΟΥ ΠΡΟΣΕΓΓΙΖΟΥΝ ΤΑ ΑΝΩΤΕΡΑ ΟΡΙΑ ΤΟΥ TAILLARD .....	81
ΠΙΝΑΚΑΣ 19. ΜΙΚΡΟΒΙΟΛΟΓΙΚΟΣ ΕΛΕΓΧΟΣ ΓΙΑ ΣΥΓΚΕΚΡΙΜΕΝΟ ΚΡΕΜΩΔΕΣ ΠΡΟΪΟΝ .....	91
ΠΙΝΑΚΑΣ 20. ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ ΜΕΤΑΣΧΗΜΑΤΙΣΜΩΝ LS ΤΟΥ DELS .....	100
ΠΙΝΑΚΑΣ 21. ΣΥΓΚΡΙΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΥΒΡΙΔΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ .....	100
ΠΙΝΑΚΑΣ 22. ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ ΤΩΝ ΑΛΓΟΡΙΘΜΩΝ DELS ΚΑΙ DEVNS .....	101

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 1. ΣΥΝΔΕΣΗ ΜΗΧΑΝΩΝ ΣΕ FLOW-SHOP .....	15
ΕΙΚΟΝΑ 2. ΔΙΑΓΡΑΜΜΑ GANTT ΓΙΑ FLOW-SHOP - ΣΕΙΡΑ ΕΚΤΕΛΕΣΗΣ ΕΡΓΑΣΙΩΝ: E1, E2, E3, E4 .....	17
ΕΙΚΟΝΑ 3. ΔΙΑΓΡΑΜΜΑ GANTT ΓΙΑ FLOW-SHOP - ΣΕΙΡΑ ΕΚΤΕΛΕΣΗΣ ΕΡΓΑΣΙΩΝ: E3, E4, E1, E2 .....	18

ΕΙΚΟΝΑ 4. FLEXIBLE FLOW-SHOP - ΠΙΘΑΝΕΣ ΜΕΤΑΒΑΣΕΙΣ .....	21
ΕΙΚΟΝΑ 5. ΔΙΆΓΡΑΜΜΑ GANTT ΓΙΑ FLEXIBLE FLOW-SHOP .....	22
ΕΙΚΟΝΑ 6. ΧΡΟΝΟΔΙΆΓΡΑΜΜΑ ΕΜΦΆΝΙΣΗΣ ΜΕΘΕΥΡΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ .....	25
ΕΙΚΟΝΑ 7. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ.....	32
ΕΙΚΟΝΑ 8. ΕΦΑΡΜΟΓΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ MIRROR .....	33
ΕΙΚΟΝΑ 9. ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ .....	34
ΕΙΚΟΝΑ 10. ΑΠΕΙΚΟΝΙΣΗ ΕΠΑΝΑΛΗΠΤΙΚΗΣ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ .....	35
ΕΙΚΟΝΑ 11. ΑΛΓΟΡΙΘΜΟΣ ΜΕΤΑΒΛΗΤΗΣ ΓΕΙΤΟΝΙΑΣ ΑΝΑΖΗΤΗΣΗΣ .....	36
ΕΙΚΟΝΑ 12. ΣΥΝΑΡΤΗΣΗ ΑΡΧΙΚΟΠΟΙΗΣΗΣ ΠΛΗΘΥΣΜΟΥ .....	42
ΕΙΚΟΝΑ 13. ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ ΚΟΣΤΟΥΣ ΓΙΑ ΤΟΝ ΥΠΟΛΟΓΙΣΜΟ ΤΟΥ MAKESPAN.....	44
ΕΙΚΟΝΑ 14. ΥΠΟΛΟΓΙΣΜΟΣ ΔΟΚΙΜΑΣΤΙΚΟΥ ΔΙΑΝΥΣΜΑΤΟΣ ΓΙΑ ΕΝΑ ΜΕΛΟΣ ΤΟΥ ΠΛΗΘΥΣΜΟΥ ΑΝΑΛΟΓΑ ΜΕ ΤΗ ΣΤΡΑΤΗΓΙΚΗ ΠΟΥ ΘΑ ΕΠΙΛΕΧΘΕΙ .....	46
ΕΙΚΟΝΑ 15. ΕΠΙΛΟΓΗ ΜΕΛΟΥΣ ΝΕΟΥ ΠΛΗΘΥΣΜΟΥ .....	47
ΕΙΚΟΝΑ 16. ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΤΡΟΠΗΣ ΣΥΝΕΧΩΝ ΣΕ ΔΙΑΚΡΙΤΕΣ ΤΙΜΕΣ .....	48
ΕΙΚΟΝΑ 17. ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΤΡΟΠΗΣ ΔΙΑΚΡΙΤΩΝ ΤΙΜΩΝ ΣΕ ΣΥΝΕΧΕΙΣ .....	49
ΕΙΚΟΝΑ 18. ΜΕΣΗ ΤΙΜΗ MAKESPAN ΓΙΑ ΤΙΣ 10 ΣΤΡΑΤΗΓΙΚΕΣ ΤΩΝ PRICE & STORN ΓΙΑ ΠΑΡΑΜΕΤΡΟΥΣ $F = 0.2$ , $CR = 0.7$ , $NP = 2D$ , $GMAX = 500$ .....	53
ΕΙΚΟΝΑ 19. ΜΕΣΗ ΤΙΜΗ MAKESPAN ΓΙΑ ΤΙΣ 10 ΣΤΡΑΤΗΓΙΚΕΣ ΤΟΥ DE ΜΕ ΠΑΡΑΜΕΤΡΟΥΣ: $F = 0.2$ , $CR = 0.7$ , $NP = 2D$ , $GMAX = 500$ ΣΕ ΠΡΟΒΛΗΜΑΤΑ 20 ΕΡΓΑΣΙΩΝ ΜΕΤΑ ΑΠΟ 10 ΕΠΑΝΑΛΗΨΕΙΣ .....	55
ΕΙΚΟΝΑ 20. ΜΕΣΗ ΤΙΜΗ MAKESPAN ΓΙΑ ΤΙΣ 10 ΣΤΡΑΤΗΓΙΚΕΣ ΤΟΥ DE ΜΕ ΠΑΡΑΜΕΤΡΟΥΣ: $F = 0.3$ , $CR = 0.9$ , $NP = 150$ , $GMAX = 50$ ΣΕ ΠΡΟΒΛΗΜΑΤΑ 20 ΕΡΓΑΣΙΩΝ ΜΕΤΑ ΑΠΟ 10 ΕΠΑΝΑΛΗΨΕΙΣ .....	56
ΕΙΚΟΝΑ 21. DE ΜΕ ΠΑΡΑΜΕΤΡΟ STOP.....	59
ΕΙΚΟΝΑ 22. ΣΧΕΣΗ MAKESPAN ΜΕ ΜΕΓΙΣΤΟ ΑΡΙΘΜΟ ΕΠΑΝΑΛΗΨΕΩΝ ΣΤΟ ΠΡΟΒΛΗΜΑ TAI31.....	59
ΕΙΚΟΝΑ 23. DE ΜΕ LS.....	63
ΕΙΚΟΝΑ 24. ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ 1-0 RELOCATE .....	63
ΕΙΚΟΝΑ 25. ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ 1-1 EXCHANGE .....	63
ΕΙΚΟΝΑ 26. ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ MIRROR .....	64
ΕΙΚΟΝΑ 27. ΥΒΡΙΔΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΚΡΙΤΗΡΙΟ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ ΣΤΗΝ ΕΠΙΛΟΓΗ ΤΟΥ ΝΕΟΥ ΜΕΛΟΥΣ ΤΟΥ ΠΛΗΘΥΣΜΟΥ .....	70
ΕΙΚΟΝΑ 28. ΥΒΡΙΔΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ ΚΑΙ ΚΡΙΤΗΡΙΟ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ ΣΤΗΝ ΕΠΙΛΟΓΗ ΤΟΥ ΝΕΟΥ ΜΕΛΟΥΣ ΤΟΥ ΠΛΗΘΥΣΜΟΥ ΤΗΣ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ .....	72
ΕΙΚΟΝΑ 29. DEILS .....	75
ΕΙΚΟΝΑ 30. ΔΙΑΤΑΡΑΧΗ ΣΤΟΝ ISL .....	75
ΕΙΚΟΝΑ 31. DEVNS.....	78
ΕΙΚΟΝΑ 32. ΟΙ ΤΕΣΣΕΡΕΙΣ ΦΑΣΕΙΣ ΤΗΣ ΠΑΡΑΓΩΓΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ.....	84



ΕΙΚΟΝΑ 33. Η ΦΑΣΗ ΤΩΝ Α΄ ΥΛΩΝ ΚΑΙ ΤΗΣ ΠΡΟΕΤΟΙΜΑΣΙΑΣ ΠΑΡΑΓΩΓΗΣ ΗΜΙΕΤΟΙΜΟΥ ΠΡΟΪΟΝΤΟΣ .....	85
ΕΙΚΟΝΑ 34. Η ΦΑΣΗ ΤΗΣ ΠΑΡΑΓΩΓΗΣ ΗΜΙΕΤΟΙΜΟΥ ΚΡΕΜΩΔΟΥΣ ΠΡΟΪΟΝΤΟΣ .....	86
ΕΙΚΟΝΑ 35. Η ΦΑΣΗ ΤΗΣ ΠΑΡΑΓΩΓΗΣ ΗΜΙΕΤΟΙΜΟΥ ΑΦΡΩΔΟΥΣ ΠΡΟΪΟΝΤΟΣ .....	87
ΕΙΚΟΝΑ 36. Η ΦΑΣΗ ΤΩΝ ΥΛΙΚΩΝ ΣΥΣΚΕΥΑΣΙΑΣ ΚΑΙ ΤΗΣ ΠΡΟΕΤΟΙΜΑΣΙΑΣ ΤΗΣ ΓΡΑΜΜΗΣ ΣΥΣΚΕΥΑΣΙΑΣ .....	87
ΕΙΚΟΝΑ 37. Η ΦΑΣΗ ΤΗΣ ΣΥΣΚΕΥΑΣΙΑΣ .....	88
ΕΙΚΟΝΑ 38. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ .....	90
ΕΙΚΟΝΑ 39. ΣΥΝΑΡΤΗΣΗ ΤΟΥ ΣΤΑΔΙΟΥ ΤΗΣ ΠΑΡΑΓΩΓΗΣ BULK .....	97
ΕΙΚΟΝΑ 40. ΣΥΝΑΡΤΗΣΗ ΤΟΥ ΣΤΑΔΙΟΥ ΤΟΥ ΜΙΚΡΟΒΙΟΛΟΓΙΚΟΥ ΕΛΕΓΧΟΥ .....	98
ΕΙΚΟΝΑ 41. ΣΥΝΑΡΤΗΣΗ ΤΟΥ ΣΤΑΔΙΟΥ ΤΗΣ ΣΥΣΚΕΥΑΣΙΑΣ .....	99

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

FSSP: Flow-Shop Scheduling Problem – Πρόβλημα χρονικού προγραμματισμού συστήματος παραγωγής συνεχούς ροής

PFSSP: Permutation Flow-Shop Scheduling Problem – Μεταθετικό πρόβλημα χρονικού προγραμματισμού συστήματος παραγωγής συνεχούς ροής.

FFSSP: Flexible Flow-Shop Scheduling Problem – Πρόβλημα χρονικού προγραμματισμού ευέλικτου συστήματος παραγωγής συνεχούς ροής

DE: Differential Evolution Algorithm – Αλγόριθμος Διαφορικής Εξέλιξης

LS: Local Search – Τοπική Αναζήτηση

SA: Simulated Annealing Algorithm – Αλγόριθμος Προσομοιωμένης Ανόπτησης

ILS: Iterated Local Search Algorithm – Αλγόριθμος Επαναληπτικής Τοπικής Αναζήτησης

VNS: Variable Neighborhood Search Algorithm – Αλγόριθμος Μεταβλητής Γειτονιάς Αναζήτησης

Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης

IGA: Iterated Greedy Algorithm – Επαναληπτικός Αλγόριθμος Απληστίας

HDE: Hybrid Differential Evolution Algorithm – Υβριδικός Αλγόριθμος Διαφορικής Εξέλιξης

DESA: Υβριδικός Αλγόριθμος Διαφορικής Εξέλιξης με Προσομοιωμένη Ανόπτηση

DEILS: Υβριδικός Αλγόριθμος Διαφορικής Εξέλιξης με Επαναληπτική Τοπική Αναζήτηση

DEVNS: Υβριδικός Αλγόριθμος Διαφορικής Εξέλιξης με Μεταβλητή Γειτονιά Αναζήτησης

## ΕΙΣΑΓΩΓΗ

Ο χρονικός προγραμματισμός συστημάτων παραγωγής επιδιώκει να μειώσει το χρόνο και τα κόστη παραγωγής βελτιστοποιώντας την αποδοτικότητα των διάφορων λειτουργιών στις διεργασίες παραγωγής. Έχει δηλαδή τεράστια επιρροή στην αποτελεσματικότητα ενός συστήματος παραγωγής. Για αυτόν τον λόγο αποτελεί αντικείμενο μελέτης για πολλούς ερευνητές και βρίσκεται συνεχώς στην επικαιρότητα των δημοσιεύσεων για αρκετές δεκαετίες.

Η παρούσα εργασία επικεντρώνεται στον χρονικό προγραμματισμό παραγωγής συστημάτων και ευέλικτων συστημάτων συνεχούς ροής. Προβλήματα τέτοιου τύπου παρουσιάζουν πληθώρα εφικτών λύσεων με τεράστιο υπολογιστικό κόστος για την εύρεση της βέλτιστης λύσης. Έτσι λοιπόν χρησιμοποιούνται τεχνικές μεθευρετικών αλγορίθμων με σκοπό την εύρεση μιας καλής λύσης. Στη συγκεκριμένη εργασία αναπτύσσονται τεχνικές υβριδικών αλγορίθμων διαφορικής εξέλιξης, οι οποίες δοκιμάζονται σε δεδομένα παρμένα από τη βιβλιογραφία ώστε να μετρηθεί η αποδοτικότητά τους.

Στην πράξη όμως τα συστήματα παραγωγής παρουσιάζουν διάφορους περιορισμούς τους οποίους τα θεωρητικά μοντέλα δε λαμβάνουν υπόψη. Αρκετοί μελετητές έχουν επισημάνει τις διαφορές και έχουν επιχειρήσει να γεφυρώσουν το χάσμα μεταξύ θεωρίας και πράξης (Urlings, 2011; Ruiz et al, 2006; Kulksar & Erdelyi, 2005). Στην πραγματικότητα λοιπόν, υπάρχει μία μεγάλη ποικιλία περιορισμών γεγονός που καθιστά πολύ δύσκολη την προσέγγιση των ρεαλιστικών συστημάτων παραγωγής με θεωρητικά μοντέλα.

Σκοπός της εργασίας είναι τόσο η μελέτη των συστημάτων παραγωγής και η δημιουργία νέων προτάσεων για την επίλυση προβλημάτων χρονικού προγραμματισμού, όσο η μοντελοποίηση ενός συστήματος παραγωγής της ελληνικής πραγματικότητας και η προσπάθεια προγραμματισμού του με την εφαρμογή μεθευρετικών αλγορίθμων.

Πιο συγκεκριμένα στην πρώτη ενότητα παρουσιάζεται το θεωρητικό υπόβαθρο που απαιτείται για την κατανόηση των τεχνικών που εφαρμόζονται στην παρούσα εργασία. Αρχικά αποσαφηνίζονται οι βασικές έννοιες και έπειτα παρουσιάζονται εν συντομία οι αλγόριθμοι που χρησιμοποιήθηκαν για να κατασκευαστούν οι υβριδικοί αλγόριθμοι διαφορικής εξέλιξης. Το θεωρητικό υπόβαθρο ακολουθεί μία σύντομη βιβλιογραφική ανασκόπηση, ώστε να παρουσιαστούν προγενέστερες έρευνες στο πεδίο του προγραμματισμού συστημάτων και ευέλικτων συστημάτων παραγωγής συνεχούς ροής με μεθευρετικούς αλγορίθμους.

Στην τρίτη ενότητα εμφανίζονται διάφορες εκδοχές του αλγορίθμου της διαφορικής εξέλιξης και κατασκευάζονται υβριδικοί αλγόριθμοι με συνδυασμούς άλλων, ήδη διατυπωμένων μεθευρετικών αλγορίθμων. Οι αλγόριθμοι έχουν προγραμματιστεί σε γλώσσα προγραμματισμού C και εφαρμόζονται σε προβλήματα ποικίλων διαστάσεων παρμένων από τη βιβλιογραφία. Αφού παρουσιαστούν συγκεντρωτικά τα αποτελέσματα, πραγματοποιείται η σύγκριση αυτών και διαπιστώνεται η αποδοτικότητά τους.

Η τέταρτη ενότητα αφορά την πρακτική εφαρμογή σε μια ελληνική εταιρία παραγωγής και εμπορίας καλλυντικών με παγκόσμια παρουσία. Αφού περιγραφεί η υποδομή και ο τρόπος λειτουργίας της εταιρίας ακολουθεί η μοντελοποίηση του συστήματος παραγωγής της. Στην συγκεκριμένη περίπτωση το σύστημα προσομοιώνεται ως ένα ευέλικτο συνεχούς ροής με διαφορετικούς περιορισμούς σε κάθε στάδιο και προγραμματίζεται με ιδιαίτερο τρόπο. Πραγματικά δεδομένα τα οποία έχουν δοθεί από την εταιρία χρησιμοποιούνται στην εφαρμογή των υβριδικών αλγορίθμων διαφορικής εξέλιξης.

Τέλος, παρουσιάζονται τα συμπεράσματα της εργασίας και πραγματοποιείται σχολιασμός εφ' όλης της ύλης.

## 1. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

### 1.1. ΠΡΟΒΛΗΜΑΤΑ ΧΡΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ

Ο χρονικός προγραμματισμός της παραγωγής επιδιώκει την αποτελεσματική χρησιμοποίηση μηχανών και προσωπικού, έτσι ώστε να επιτευχθούν στόχοι όπως η ελαχιστοποίηση του χρόνου αναμονής πελατών, του χρόνου αποθήκευσης, του χρόνου πραγματοποίησης της παραγωγικής διαδικασίας κλπ. Στοχεύει λοιπόν στον προγραμματισμό ενός συνόλου εργασιών έτσι ώστε να βελτιστοποιούνται ορισμένα κριτήρια και να ικανοποιούνται συγκεκριμένοι περιορισμοί.

Η πλειονότητα των προβλημάτων χρονικού προγραμματισμού παραγωγής παρουσιάζει ιδιαίτερη πολυπλοκότητα στην επίλυσή τους για την εύρεση της ολικής βέλτιστης λύσης. Στις περισσότερες περιπτώσεις, όταν το μέγεθος του προβλήματος αυξάνεται απαιτείται μεγάλο χρονικό διάστημα για την επίλυσή του και μερικές φορές η επίλυση είναι πρακτικά αδύνατη. Προβλήματα για τα οποία δεν υπάρχει γνωστός αποδοτικός αλγόριθμος που να δίνει βέλτιστη λύση και είναι σχεδόν αδύνατο να βρεθεί, ανήκουν στην κατηγορία των NP-hard προβλημάτων σύμφωνα με την ορολογία που διατύπωσε ο Knuth το 1974 (Garey & Johnson, 1990).

Υπάρχουν διάφορα κριτήρια για να κατηγοριοποιηθούν τα προβλήματα συστημάτων παραγωγής. Τα πιο διαδεδομένα είναι: ο τρόπος άφιξης των παραγγελιών, η πολιτική αποθεμάτων, οι ιδιότητες των εργασιών και η δομή των συστημάτων (Zobolas et al, 2008). Τα βασικότερα συστήματα παραγωγής σύμφωνα με τα οποία κατηγοριοποιούνται τα λεγόμενα shop scheduling problems είναι τα εξής:

- σύστημα παραγωγής κατά παραγγελία (job-shop): Κάθε εργασία αποτελείται από επιμέρους διεργασίες καθεμία από τις οποίες εκτελείται σε μία ορισμένη μηχανή και η σειρά εκτέλεσης των διεργασιών είναι διαφορετική για κάθε εργασία.

- σύστημα παραγωγής συνεχούς ροής (flow-shop): Κάθε εργασία αποτελείται από επιμέρους διεργασίες καθεμία από τις οποίες εκτελείται σε ορισμένη μηχανή και η σειρά εκτέλεσης των διεργασιών για όλες τις εργασίες είναι η ίδια. Τα προβλήματα που αφορούν τα συστήματα αυτά θα παρουσιαστούν αναλυτικότερα στις επόμενες παραγράφους.
- σύστημα ανοιχτής παραγωγής (open-shop): Κάθε εργασία αποτελείται από επιμέρους διεργασίες καθεμία από τις οποίες εκτελείται σε ορισμένη μηχανή. Η σειρά εκτέλεσης των διεργασιών είναι διαφορετική για κάθε εργασία και μια εργασία μπορεί να έχει παραπάνω από μία εναλλακτικές σειρές εκτέλεσης διεργασιών.

Πέρα από τη δομή του συστήματος παραγωγής ένα άλλο σημαντικό χαρακτηριστικό για τα προβλήματα χρονικού προγραμματισμού είναι το **κριτήριο προς βελτιστοποίηση**. Ενδεικτικά αναφέρονται μερικά από αυτά στον παρακάτω πίνακα:

Πίνακας 1. κριτήρια χρονικού προγραμματισμού σε συστήματα παραγωγής

Όνομα κριτηρίου	Επεξήγηση	Παραδείγματα μελετών
Makespan	χρόνος περάτωσης τελευταίας διεργασίας	Tasgetiren et al (2004) Hejazi & Saghafian (2005) Pan et al (2007)
total flow time	συνολικός χρόνος ροής	Pan & Ruiz (2011) Costa et al (2011)
total completion time	συνολικός χρόνος ολοκλήρωσης	Aydilek & Allahverdi (2006)
maximum tardiness	μέγιστη βραδύτερη περάτωση	Vallada & Ruiz (2010)
earliness - tardiness	χρόνοι νωρίτερης – βραδύτερης περάτωσης	Zhonghua et al (2009)
multi-criteria	συνδυασμός κριτηρίων	Eren (2007)

Επίσης, σημαντικό ρόλο στα προβλήματα χρονικού προγραμματισμού παίζουν οι **περιορισμοί** που προκύπτουν από τις διάφορες απαιτήσεις του συστήματος. Οι απαιτήσεις αυτές μπορεί να είναι η διαθεσιμότητα του παραγωγικού εξοπλισμού, η διαθεσιμότητα του

ανθρώπινου δυναμικού, η διαθεσιμότητα οικονομικών πόρων, ο χρόνος, η συντήρηση των μηχανών κλπ.

Στη συνέχεια θα αναλυθούν λεπτομερέστερα τα συστήματα και τα ευέλικτα συστήματα παραγωγής συνεχούς ροής τα οποία πραγματεύεται η παρούσα εργασία.

### 1.1.1. ΣΥΣΤΗΜΑ ΠΑΡΑΓΩΓΗΣ ΣΥΝΕΧΟΥΣ ΡΟΗΣ

Τα συστήματα παραγωγής συνεχούς ροής (flow-shop) χρησιμοποιούνται για μαζική παραγωγή περιορισμένης ποικιλίας τυποποιημένων προϊόντων, όπως για παράδειγμα τα προϊόντα διατροφής. Ο μηχανικός εξοπλισμός είναι τοποθετημένος σειριακά (Εικόνα 1) και τα προϊόντα που παράγονται συνήθως ακολουθούν την ίδια διαδρομή μέσα στο σύστημα.



Εικόνα 1. σύνδεση μηχανών σε flow-shop

Δηλαδή όλες οι εργασίες που πρέπει να πραγματοποιηθούν περνούν από όλες τις μηχανές του συστήματος οι οποίες βρίσκονται σε συγκεκριμένη σειρά. Η κάθε εργασία λοιπόν αποτελείται από επιμέρους διεργασίες οι οποίες ουσιαστικά αντιστοιχίζονται με τις μηχανές. Άρα αν το σύστημα περιλαμβάνει  $M$  διαφορετικές μηχανές και  $N$  διαφορετικές εργασίες, η κάθε μία από τις οποίες περιέχει  $M$  διεργασίες, τότε συμβολίζουμε τις μηχανές με  $1, 2, \dots, M$  και τις διεργασίες μιας εργασίας  $i$  με  $(1, i), (2, i) \dots (M, i)$ , όπου  $i = 1, 2, \dots, N$ . Στην περίπτωση που κάποιες εργασίες δεν απαιτείται να περάσουν από όλες τις μηχανές πάλι υπάρχει συνεχής ροή εφόσον η σειρά με την οποία επισκέπτονται τις υπόλοιπες μηχανές είναι η καθορισμένη.

Οι υποθέσεις που χαρακτηρίζουν το σύστημα έτσι όπως έχουν διατυπωθεί από τους Baker και Trietsch (2009) είναι οι εξής:

- Μια ομάδα  $N$  ασυσχέτιστων εργασιών είναι διαθέσιμη για επεξεργασία τη χρονική στιγμή 0.
- Η κάθε εργασία απαιτεί  $M$  διεργασίες και η κάθε διεργασία απαιτεί διαφορετική μηχανή.
- Οι χρόνοι εξάρμωσης (setup time) είναι «εξαρτημένοι ακολουθίας» (sequencing dependent)<sup>1</sup> των μηχανών και συμπεριλαμβάνονται στους χρόνους επεξεργασίας.
- Οι χρόνοι επεξεργασίας των διεργασιών από τις μηχανές είναι γνωστοί από την αρχή.
- Όλες οι μηχανές είναι πάντα διαθέσιμες.
- Όταν ξεκινήσει η διεργασία συνεχίζει χωρίς καμία διακοπή.

Στο flow-shop η χρονική διάρκεια ενός προγράμματος με  $N$  εργασίες και  $M$  μηχανές, ισούται με τον μέγιστο χρόνο περάτωσης των διεργασιών στην τελευταία μηχανή, δηλαδή:

$$C_{\max} = \max (C_{iM}), \quad i = 1, 2, \dots, n \quad (1)$$

όπου  $C_{\max}$  ο χρόνος λήξης του προγράμματος και  $C_{iM}$  ο χρόνος περάτωσης κάθε εργασίας  $i$  στην τελευταία μηχανή  $M$ .

Οι χρόνοι επεξεργασίας  $p_{ij}$  αφορούν το χρόνο που απαιτείται για την κάθε εργασία  $i = 1, 2, \dots, N$  να περάσει στην κάθε μηχανή  $j = 1, 2, \dots, M$ .

Όταν μία εργασία  $i$  φτάνει στη μηχανή  $j$  τότε είτε η μηχανή  $j$  είναι απασχολημένη με την επεξεργασία της προηγούμενης εργασίας  $i-1$  είτε η μηχανή  $j$  είναι διαθέσιμη και «περιμένει» την εργασία  $i$  για επεξεργασία. Έτσι λοιπόν ισχύει:

---

<sup>1</sup> Χρόνος εξάρμωσης μιας μηχανής (setup time) είναι ο χρόνος προετοιμασίας που χρειάζεται η μηχανή. Αν εξαρτάται μόνο από το είδος της εργασίας που πρόκειται να εκτελεσθεί στη μηχανή λέγεται «ανεξάρτητος ακολουθίας» (sequence independent), ενώ αν εξαρτάται και από το είδος της εργασίας της οποίας η επεξεργασία προηγήθηκε στη μηχανή, λέγεται «εξαρτημένος ακολουθίας» (sequence dependent). Στην παρούσα εργασία υποθέτουμε ότι οι χρόνοι εξάρμωσης περιλαμβάνονται στον χρόνο επεξεργασίας και δεν λαμβάνονται υπόψη.



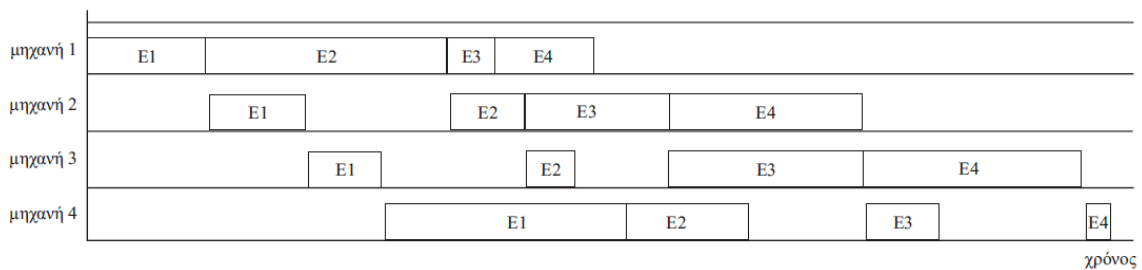
$$C_{i,j} = \max(C_{i-1,j}, C_{i,j-1}) + p_{i,j} \quad (2)$$

Στον παρακάτω πίνακα για παράδειγμα εμπεριέχονται οι χρόνοι επεξεργασίας για ένα πρόβλημα με 4 μηχανές και 4 εργασίες:

Πίνακας 2. χρόνοι επεξεργασίας για FSSP τεσσάρων μηχανών και τεσσάρων εργασιών

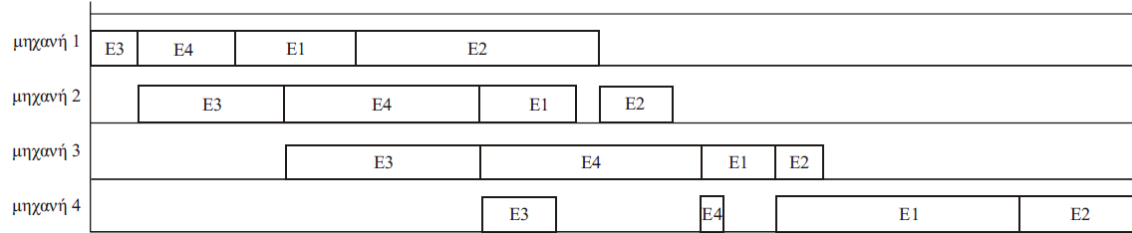
εργασίες μηχανές	E1	E2	E3	E4
M1	5	10	2	4
M2	4	3	6	8
M3	3	2	8	9
M4	10	5	3	1

Στο παρακάτω διάγραμμα απεικονίζεται η χρονική διάρκεια των διεργασιών στην περίπτωση που οι εργασίες πραγματοποιούνται με τη σειρά E1, E2, E3 και E4. Οι μηχανές έχουν μία καθορισμένη σειρά: M1, M2, M3 και M4. Όταν εκτελεσθεί και η τελευταία διεργασία τότε ολοκληρώνεται η εκάστοτε εργασία. Δηλαδή όταν περάσουν όλες οι εργασίες και από την τελευταία μηχανή, την M4, θα έχουν ολοκληρωθεί όλες οι εργασίες.



Εικόνα 2. διάγραμμα Gantt για flow-shop - σειρά εκτέλεσης εργασιών: E1, E2, E3, E4

Στο επόμενο διάγραμμα απεικονίζονται πάλι οι χρονικές διάρκειες των διεργασιών, όμως αυτή τη φορά η σειρά πραγματοποίησης των εργασιών είναι διαφορετική: E3, E4, E1, E2.



Εικόνα 3. διάγραμμα Gantt για flow-shop - σειρά εκτέλεσης εργασιών: E3, E4, E1, E2

Παρατηρείται ότι ανάλογα με τη σειρά εκτέλεσης των εργασιών τα χρονικά αποτελέσματα διαφέρουν. Στα προβλήματα χρονικού προγραμματισμού flow-shop (Flow Shop Scheduling Problems – FSSP) το ζητούμενο είναι ακριβώς αυτό. Να βρεθεί μία αλληλουχία εργασιών, η οποία θα βελτιστοποιεί κάποιο συγκεκριμένο κριτήριο. Το κριτήριο που χρησιμοποιείται ευρέως στη βιβλιογραφία και το οποίο θα χρησιμοποιηθεί σε αυτή την εργασία είναι ο χρόνος περάτωσης της τελευταίας εργασίας (makespan).

Στο παραπάνω παράδειγμα χρησιμοποιούμε μια συγκεκριμένη αλληλουχία εργασιών για όλες τις μηχανές. Σε αυτή την περίπτωση έχουμε **μεταθετικό πρόβλημα χρονικού προγραμματισμού συστήματος συνεχούς ροής** (Permutation Flow-Shop Scheduling Problem - PFSSP). Στην περίπτωση που η σειρά των εργασιών αλλάζει για κάθε μηχανή και ο αριθμός των εργασιών είναι  $N$ , τότε υπάρχουν  $N!$  πιθανές αλληλουχίες εργασιών για κάθε μηχανή και άρα υπάρχουν  $(N!)^M$  διαφορετικά προγράμματα. Στην περίπτωση όμως του μεταθετικού flow-shop υπάρχουν μόνο  $N!$  διαφορετικές διατάξεις των  $N$  εργασιών.

Στην πραγματικότητα στα μεταθετικά προγράμματα εμφανίζονται συχνότερα νεκροί χρόνοι (idle times)<sup>2</sup> μεταξύ των διεργασιών συγκριτικά με τα μη μεταθετικά. Το γεγονός όμως ότι η καλύτερη μεταθετική λύση δεν είναι πολύ χειρότερη από μία μη μεταθετική και δεδομένου του υπολογιστικού κόστους των δύο περιπτώσεων, οι περισσότερες έρευνες στη βιβλιογραφία αφορούν μεταθετικά προβλήματα. Στις επόμενες ενότητες της παρούσας εργασίας όταν χρησιμοποιείται ο όρος flow-shop θα εννοείται permutation flow-shop.

<sup>2</sup> Νεκρός χρόνος (idle time) ονομάζεται το χρονικό διάστημα για το οποίο μια μηχανή του συστήματος δε χρησιμοποιείται.

Εκτός από τα μεταθετικά και τα μη μεταθετικά προβλήματα υπάρχουν διάφορες άλλες παραλλαγές του κλασικού FSSP, οι οποίες μπορούν να βασίζονται σε μια ποικιλία παραγόντων όπως για παράδειγμα τους χρόνους εξάρμωσης, τους νεκρούς χρόνους, την ύπαρξη αποθηκών με περιορισμένο χώρο μεταξύ των μηχανών κλπ. Η παρούσα εργασία αναφέρεται στο κλασικό FSSP το οποίο περιγράφηκε παραπάνω.

Το FSSP έχει απασχολήσει επί δεκαετίες τους ερευνητές. Πρώτος ο Johnson (1954) απέδειξε ότι είναι δυνατό να βρεθεί βέλτιστη λύση στο πρόβλημα προγραμματισμού flow-shop με δύο μηχανές. Το όνομά του έχει ταυτιστεί με τα προβλήματα δύο μηχανών καθώς έχει επηρεάσει αισθητά τη μετέπειτα θεωρία προγραμματισμού σε προβλήματα συνεχούς ροής (Baker & Trietsch, 2009). Σύμφωνα με τον κανόνα του Johnson στην περίπτωση μεταθετικού προβλήματος τριών μηχανών είναι δυνατό να βρεθεί βέλτιστη λύση, ενώ στην περίπτωση του μη μεταθετικού συστήματος το πρόβλημα γίνεται NP-hard (Baker & Trietsch, 2009) καθώς όλα τα προβλήματα των μη μεταθετικών συστημάτων που έχουν πάνω από τρεις μηχανές κατατάσσονται στην κατηγορία των NP-hard προβλημάτων (Garey et al, 1976).

Βέλτιστη λύση για μεταθετικά προβλήματα τριών μηχανών μπορεί να αποδώσει και η μέθοδος Branch and Bound με την οποία από τους πρώτους που ασχολήθηκαν ήταν οι Ignall και Schrage το 1965. Έκτοτε πολλοί άλλοι ερευνητές χρησιμοποίησαν διάφορες τεχνικές στα πλαίσια αυτής της μεθόδου. Η δυσκολία όμως έγκειται στο ότι η μέθοδος Branch and Bound όπως και άλλες μέθοδοι δυναμικού προγραμματισμού χάνουν την ισχύ τους στα μεγάλα προβλήματα λόγω του υπολογιστικού κόστους. Για αυτό το λόγο το ενδιαφέρον των ερευνητών κατευθύνεται σε άλλες αποτελεσματικές τεχνικές, τους μεθευρετικούς αλγορίθμους.

### 1.1.2. ΕΥΕΛΙΚΤΟ ΣΥΣΤΗΜΑ ΠΑΡΑΓΩΓΗΣ ΣΥΝΕΧΟΥΣ ΡΟΗΣ

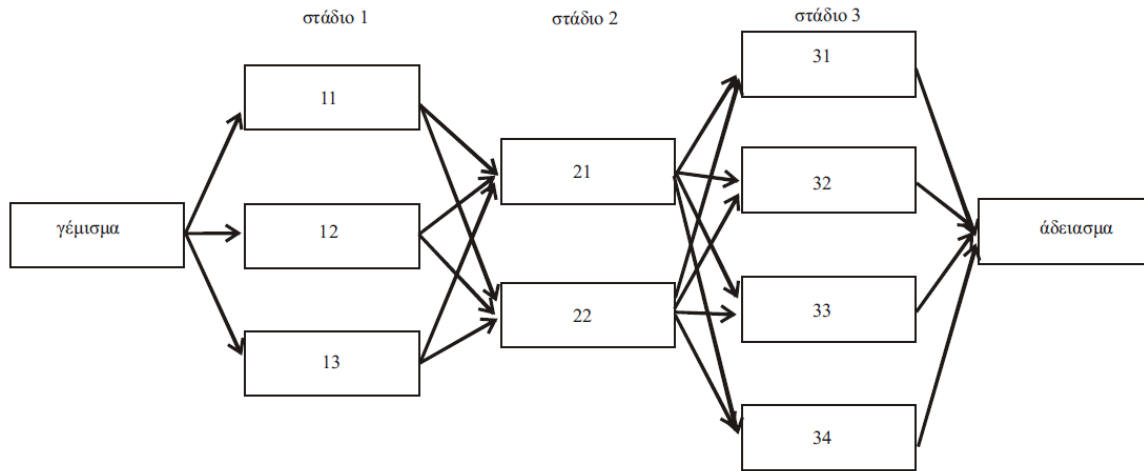
Το πρόβλημα του χρονικού προγραμματισμού του ευέλικτου συστήματος συνεχούς ροής (Flexible Flow-Shop Scheduling Problem - FFSSP) γνωστό και ως πρόβλημα υβριδικού συστήματος συνεχούς ροής (hybrid flow-shop ή multi-processor flow-shop problem) παρουσιάζει ένα συνεχώς αυξανόμενο ενδιαφέρον στη βιβλιογραφία (Ruiz & Vázquez-Rodríguez, 2010). Το flexible flow-shop είναι ουσιαστικά μία γενίκευση του flow-shop και των συστημάτων με παράλληλες μηχανές (Sriskandarajah & Sethi, 1989; Pinedo, 2008). Αντί για  $M$  μηχανές σε σειρά υπάρχουν  $c$  στάδια σε σειρά από τα οποία το κάθε ένα περιλαμβάνει έναν αριθμό από μηχανές σε παράλληλη γεωμετρία. Κάθε εργασία πρέπει να περάσει από όλα τα στάδια σειριακά, χρησιμοποιώντας μόνο μία μηχανή στο καθένα.

Τα παραδοσιακά flexible flow-shop είναι αυτοματοποιημένα σε μεγάλο βαθμό. Συνήθως, υπάρχει μία πρώτη μηχανή η οποία «γεμίζει» το σύστημα και μία τελευταία που το «αδειάζει», ενώ οι μεταφορές προϊόντων μεταξύ των μηχανών πραγματοποιούνται από ένα συγκεκριμένο και αυτοματοποιημένο σύστημα μεταφοράς, το οποίο έχει τη δυνατότητα να προσπερνάει μηχανές (Hitz, 1979).

Σύμφωνα με τους Ruiz και Vázquez-Rodríguez (2010) στο κλασικό μοντέλο flexible flow-shop υπάρχουν οι εξής παραδοχές:

- Σε κάθε στάδιο υπάρχουν σε παράλληλη σύνδεση πανομοιότυπες μηχανές.
- Όλες οι μηχανές και οι εργασίες είναι διαθέσιμες στη χρονική στιγμή 0.
- Η μηχανή έχει τη δυνατότητα να επεξεργάζεται μόνο μία εργασία τη φορά.
- Οι χρόνοι εξάρμωσης είναι αμελητέοι.
- Η χωρητικότητα των αποθηκών μεταξύ των μηχανών είναι απεριόριστη.
- Τα δεδομένα του προβλήματος είναι καθορισμένα και γνωστά εξ αρχής.

Παρακάτω παρουσιάζονται διαγραμματικά οι πιθανές μεταβάσεις σε ένα παράδειγμα flexible flow-shop με  $c = 3$ ,  $M_1 = 3$ ,  $M_2 = 2$  και  $M_3 = 4$ :



Εικόνα 4. flexible flow-shop - πιθανές μεταβάσεις

Έστω λοιπόν ότι το σύστημα έχει  $c \geq 2$  στάδια μηχανών και στο στάδιο 1 υπάρχουν  $M_1 \geq 1$  πανομοιότυπες παράλληλες μηχανές όπου  $l = 1, 2, \dots, c$  και  $m_{li}$  οι μηχανές με  $i = 1, 2, \dots, M_l$ . Οι εργασίες  $j = 1, 2, \dots, N$  πρέπει να περάσουν από κάθε στάδιο και οι χρόνοι επεξεργασίας είναι  $p_{1j}, p_{2j}, \dots, p_{cj}$  (όταν οι μηχανές είναι πανομοιότυπες σε κάθε στάδιο, δηλαδή στην περίπτωση του κλασικού flexible flow-shop).

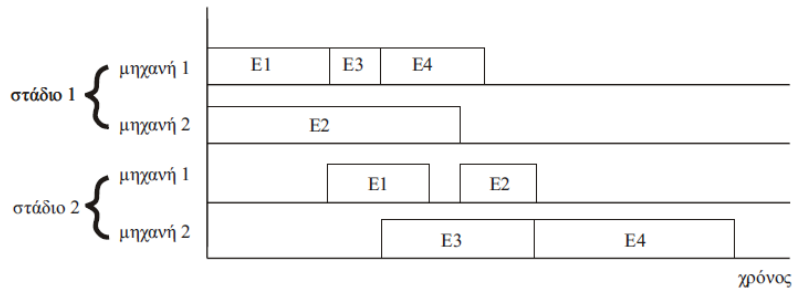
Οι Brockmann & Dangelmaier (1998) συμβολίζουν με  $C_{jl}$  το χρόνο ολοκλήρωσης κάθε εργασίας  $j$  σε κάθε στάδιο  $l$  και με  $a_{li}$  το διαθέσιμο χρονικό διάστημα που έχει για επεξεργασία η μηχανή  $i$  στο στάδιο  $l$ . Αν η εργασία  $j$  τοποθετηθεί στην πρώτη ελεύθερη μηχανή του σταδίου  $l$  τότε ο χρόνος επεξεργασίας της  $s_{j,l}$  ξεκινάει:

$$s_{j,l} = \max \left( \min_{1 \leq i \leq M_l} \{a_{li}\}, c_{j,l-1} \right)$$

Άρα ο χρόνος ολοκλήρωσης της κάθε εργασίας  $j$  στο κάθε στάδιο  $l$  θα είναι:  $C_{j,l} = s_{j,l} + p_{j,l}$

και ο υπολογισμός του makespan προκύπτει από:  $C_{\max} = \max (C_{j,c})$

Στην παρούσα εργασία το ζητούμενο είναι η εύρεση της σειράς εκτέλεσης των εργασιών στο πρώτο στάδιο με σκοπό την ελαχιστοποίηση του makespan. Η σταθερή σειρά εκτέλεσης των εργασιών σε όλα τα στάδια δεν έχει νόημα όπως στο flow-shop αφού σε αυτή την περίπτωση οι εργασίες εκτελούνται και παράλληλα. Για να γίνει πιο κατανοητό παρουσιάζεται το διάγραμμα Gantt για ένα flexible flow-shop με 2 στάδια και 4 μηχανές.



Εικόνα 5. διάγραμμα Gantt για flexible flow-shop

Το μοντέλο flexible flow-shop έχει προσδιοριστεί με διάφορες παραλλαγές. Για παράδειγμα οι Verma και Dessouky (1999) καθώς και ο Sethanan (2001) μελέτησαν μοντέλα με μη πανομοιότυπες μηχανές σε κάθε στάδιο, ενώ οι Kulksar και Erdelyi (2005) πρότειναν επεκτάσεις του προβλήματος, ούτως ώστε το θεωρητικό μοντέλο να έρθει πιο κοντά στα ρεαλιστικά συστήματα παραγωγής. Όπως αναφέρθηκε και στην εισαγωγή, υπάρχει ένα κενό μεταξύ θεωρίας και πράξης μιας και στην πραγματικότητα τα συστήματα παραγωγής παρουσιάζουν πολλούς περιορισμούς που δε συμπεριλαμβάνονται στα θεωρητικά μοντέλα. Στην τέταρτη ενότητα της παρούσας εργασίας θα περιγραφεί ένα flexible flow-shop όπου οι περιορισμοί είναι διαφορετικοί για κάθε στάδιο.

Έχει αποδειχτεί ότι τα FFSSP ανήκουν αυστηρώς στην κατηγορία των NP-hard προβλημάτων (Gupta, 1988), επομένως όπως παρατηρείται και στη βιβλιογραφία η επίλυσή τους προσεγγίζεται κυρίως από Branch and Bound και ευρετικές τεχνικές με έντονο ενδιαφέρον τα τελευταία χρόνια στους μεθευρετικούς αλγορίθμους (Ruiz & Vázquez-Rodríguez, 2010; Quadt & Kuhn, 2007). Στη συνέχεια αυτού του κεφαλαίου θα παρουσιαστούν οι μεθευρετικές τεχνικές.

## 1.2. ΕΥΡΕΤΙΚΟΙ - ΜΕΘΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Έχουν διατυπωθεί διάφορες τεχνικές που οδηγούν, σε σχετικά σύντομο χρονικό διάστημα, σε μια σχεδόν βέλτιστη - ικανοποιητική λύση των NP-hard προβλημάτων. Οι τεχνικές αυτές είναι οι **ευρετικοί αλγόριθμοι** οι οποίοι καταλήγουν σε μια λύση όταν αυτή ικανοποιεί συγκεκριμένα κριτήρια με λογικό υπολογιστικό κόστος (Vob, 2009). Τα κριτήρια διαφέρουν σε κάθε αλγόριθμο, αλλά διαφορετικοί αλγόριθμοι μπορούν να εφαρμοστούν στο ίδιο πρόβλημα. Για παράδειγμα, σε ένα συγκεκριμένο πρόβλημα μπορεί να είναι πιο αποτελεσματικός διαφορετικός αλγόριθμος για μικρό όγκο δεδομένων και διαφορετικός για μεγάλο.

Ένα άλλο θέμα που οφείλεται να εξεταστεί είναι η ποιότητα της λύσης των ευρετικών αλγορίθμων. Έχουν διατυπωθεί διάφορες μέθοδοι για τον σκοπό αυτό (Αναγνωστόπουλος & Κώτσικας, 2004) καθώς και τεχνικές για την επιλογή αλγορίθμου (Allen & Minton, 1996). Ένας κλασικός τρόπος για τη σύγκριση της αποδοτικότητας διαφορετικών αλγορίθμων είναι μέσω δοκιμαστικών «τρεξιμάτων» του ίδιου προβλήματος και η εν τέλει η σύγκριση των αποτελεσμάτων τους. Το μεγαλύτερο πρόβλημα του συγκεκριμένου τρόπου είναι ότι το δείγμα που δοκιμάζεται δε μπορεί να είναι αντιπροσωπευτικό για όλες τις περιπτώσεις. Δεν σημαίνει δηλαδή ότι αν ένας αλγόριθμος βελτιστοποιεί ένα σετ δεδομένων ενός συγκεκριμένου προβλήματος θα συμβαίνει το ίδιο και με όλα τα πιθανά σετ δεδομένων.

Στη συνέχεια παρουσιάζονται τρεις γενικές κατηγορίες ευρετικών αλγορίθμων (Μαρινάκης et al, 2011):

- Αλγόριθμοι απληστίας (greedy algorithms): Κατασκευάζουν σταδιακά τη βέλτιστη λύση και σε κάθε στάδιο επιλέγεται η τοπικά βέλτιστη λύση με βάση κάποιο «κριτήριο απληστίας». Η επιλογή που ήδη έχει γίνει σε κάποιο στάδιο δε μπορεί να αλλάξει μεταγενέστερα.
- Προσεγγιστικοί αλγόριθμοι (approximation algorithms): Είναι πολυωνυμικοί αλγόριθμοι με εγγυημένη ποιότητα για τη λύση του προβλήματος. Εδώ αξίζει να

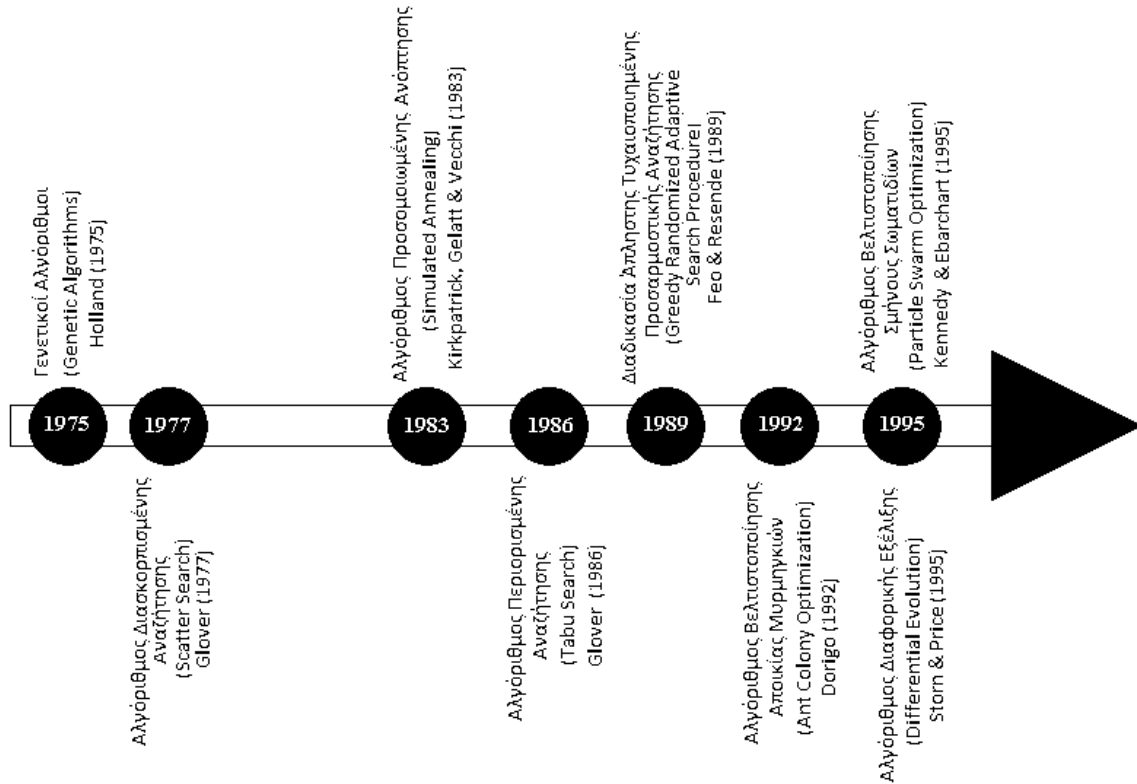
σημειωθεί ότι κάποιοι ερευνητές θεωρούν τους προσεγγιστικούς αλγορίθμους κάτι διαφορετικό από τους ευρετικούς διότι οι πρώτοι εγγυώνται για την ποιότητα της λύσης (Talbi, 2009).

- Αλγόριθμοι τοπικής αναζήτησης (local search algorithms): Επιχειρούν να βελτιώσουν μια αρχική λύση με κάποια μέθοδο αναζήτησης στη γειτονιά της λύσης, δηλαδή οι αλγόριθμοι τοπικής αναζήτησης προαπαιτούν την ύπαρξη μιας εφικτής λύσης.

Ο Glover (1986) χρησιμοποίησε για πρώτη φορά τον όρο μεθευρετικός αλγόριθμος (meta-heuristic) στην προσπάθειά του να περιγράψει τον αλγόριθμο Περιορισμένης Αναζήτησης (Tabu Search). Οι **μεθευρετικοί αλγόριθμοι** μπορούν να οριστούν ως **«μέθοδοι επίλυσης που συνδυάζουν διαδικασίες τοπικής αναζήτησης και υψηλότερου επιπέδου στρατηγικές για να δημιουργήσουν μια διαδικασία που είναι ικανή να ξεφύγει από κάποιο τοπικό ελάχιστο»** (Μαρινάκης & Μυγδαλάς, 2008). Με απλά λόγια θα μπορούσε να ειπωθεί ότι οι μεθευρετικοί είναι αναβαθμισμένοι ευρετικοί αλγόριθμοι.

Οι μεθευρετικοί αλγόριθμοι επιλύουν περιπτώσεις NP-hard προβλημάτων εξερευνώντας συνήθως ένα μεγάλο πεδίο λύσεων, το οποίο σταδιακά μειώνεται με σκοπό να βρεθεί μια καλύτερη λύση. Επιπλέον, είναι εύκολοι στο σχεδιασμό και πολύ ευέλικτοι. Έτσι, τα τελευταία χρόνια οι περισσότεροι αλγόριθμοι που έχουν αναπτυχθεί για την επίλυση των προβλημάτων συνδυαστικής βελτιστοποίησης ανήκουν σε αυτή την κατηγορία (Μαρινάκης & Μυγδαλάς, 2008). Μερικοί από τους πιο διαδεδομένους μεθευρετικούς αλγορίθμους παρουσιάζονται στο παρακάτω χρονοδιάγραμμα:





Εικόνα 6. χρονοδιάγραμμα εμφάνισης μεθευρετικών αλγορίθμων

Υπάρχει πληθώρα μεθευρετικών αλγορίθμων γεγονός που καθιστά την προσέγγιση όλων δύσκολη. Για αυτόν τον λόγο θα παρουσιαστούν σε τέσσερεις γενικές κατηγορίες:

- **Μεθευρετικοί αλγόριθμοι βασισμένοι στη γειτονιά αναζήτησης:** Αλγόριθμοι οι οποίοι χρησιμοποιώντας τη μέθοδο της τοπικής αναζήτησης διερευνούν την περιοχή γύρω από ένα σημείο που έχει βρεθεί ως τοπικό ελάχιστο, δηλαδή πραγματοποιούν μικρές αλλαγές σε μία μόνο λύση. Οι αλγόριθμοι αυτοί λειτουργούν με διάφορες στρατηγικές και έχουν κατηγοριοποιηθεί σύμφωνα με τους Μαρινάκη et al (2011) ως εξής:

1. Επαναληπτικές διαδικασίες που αρχίζουν από διαφορετικές αρχικές λύσεις π.χ. αλγόριθμος επαναληπτικής τοπικής αναζήτησης (iterated local search).

2. Αλγόριθμοι που δέχονται γειτονικές κινήσεις που δεν βελτιώνουν τη λύση π.χ. αλγόριθμος προσομοιωμένης απόπτησης (simulated annealing).
  3. Αλγόριθμοι που αλλάζουν τη γειτονιά αναζήτησης π.χ. αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (variable neighborhood search).
  4. Αλγόριθμοι που αλλάζουν την αντικειμενική συνάρτηση ή κάποια από τα δεδομένα του προβλήματος π.χ. αλγόριθμος καθοδηγούμενης τοπικής αναζήτησης (guided local search)
- **Κατασκευαστικοί μεθευρετικοί αλγόριθμοι:** Κατασκευάζουν λύσεις από δικά τους συστατικά στοιχεία αντί να βελτιώνουν υπάρχουσες λύσεις π.χ. η διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (greedy randomized adaptive search procedure - GRASP).
  - **Μεθευρετικοί αλγόριθμοι βασισμένοι στον πληθυσμό:** Αλγόριθμοι οι οποίοι βρίσκουν καλές λύσεις επιλέγοντας και συνδυάζοντας κατ' επανάληψη ήδη υπάρχουσες λύσεις από μια ομάδα που ονομάζεται πληθυσμός. Ξεκινώντας από έναν αρχικό πληθυσμό επανειλημμένα αντικαθιστούν την τρέχουσα με μία νέα γενιά πληθυσμού. Τα πιο σημαντικά μέλη αυτής της κατηγορίας είναι οι **εξελικτικοί αλγόριθμοι**, οι οποίοι μέσω μιας επαναληπτικής διαδικασίας εξελίσσουν νέους πληθυσμούς ατόμων από τους παλιούς π.χ. γενετικοί αλγόριθμοι (genetic algorithms). Σε αυτήν την κατηγορία ανήκει και ο αλγόριθμος της διαφορικής εξέλιξης που εξετάζεται εκτενέστερα στην παρούσα εργασία.
  - **Υβριδικοί μεθευρετικοί αλγόριθμοι:** Αλγόριθμοι οι οποίοι ενσωματώνουν σε μεθευρετικούς αλγόριθμους κάποιες άλλες τεχνικές. Οι Blum και Roli (2008) διαχωρίζουν του υβριδικούς μεθευρετικούς αλγόριθμους σε δύο κατηγορίες:
    1. Αλγόριθμοι οι οποίοι συνδυάζουν στοιχεία διαφορετικών μεθευρετικών τεχνικών.

2. Αλγόριθμοι οι οποίοι συνδυάζουν μεθευρετικές τεχνικές με τεχνικές άλλων πεδίων όπως της επιχειρησιακής έρευνας και της τεχνητής νοημοσύνης.

Στη συνέχεια θα αναλυθούν οι αλγόριθμοι που θα χρησιμοποιηθούν για τη δημιουργία των υβριδικών αλγορίθμων διαφορικής εξέλιξης.

### 1.2.1. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ

Ο αλγόριθμος της διαφορικής εξέλιξης (Differential Evolution Algorithm - DE) για πρώτη φορά δημοσιεύτηκε το 1995 σε μία τεχνική έκθεση των Storn και Price. Αμέσως απέδειξε την αξία του σε διαγωνισμούς όπως το 1996 στον International Contest on Evolutionary Optimization που διοργανώθηκε από την IEEE (Price et al, 2005) και έκτοτε απασχολεί πολλούς ερευνητές αποτελώντας πλέον έναν από τους πιο διαδεδομένους μεθευρετικούς αλγόριθμους.

Όπως όλοι οι εξελικτικοί αλγόριθμοι έτσι και ο DE είναι βασισμένος στον πληθυσμό. Τα μέλη του πληθυσμού είναι διανύσματα τα οποία αποτελούνται από μεταβλητές, οι οποίες ονομάζονται γονίδια. Η βασική ιδέα βρίσκεται στη χρήση διαφορικών διανυσμάτων για την διατάραξη (perturbation) των διανυσμάτων του πληθυσμού με σκοπό τη δημιουργία νέων, δηλαδή νέας γενιάς πληθυσμού. Σημαντικό ρόλο στη διατάραξη παίζει ένα διάνυσμα από τον παλαιότερο πληθυσμό το οποίο ονομάζεται γονέας.

Η συνάρτηση προς βελτιστοποίηση,  $C$ , είναι της μορφής  $C(X): R^D \rightarrow R$  και όπου  $D$  είναι ο αριθμός των μεταβλητών απόφασης, οι οποίες συμβολίζονται με  $X = x_1, x_2, \dots, x_D$  (γονίδια). Ο πληθυσμός αποτελείται από  $NP$  διανύσματα και το κάθε διάνυσμα έχει ένα δείκτη από το 0 μέχρι το  $NP-1$ . Συνεπώς ο πληθυσμός  $P$  της γενιάς  $G$  είναι:

$$P^{(G)} = X_i^{(G)} = x_{i,j}^{(G)} \quad (3)$$

με  $i = 1, 2, \dots, NP$ ;  $j = 1, 2, \dots, D$  και  $G = 1, 2, \dots, G_{\max}$

Στις κλασικές περιπτώσεις προβλημάτων που επιλύονται με τον DE δίδονται κάποια όρια στα οποία μπορεί να κινηθεί η τιμή των μεταβλητών απόφασης:

$$x_j^{(L)} \geq x_j \geq x_j^{(U)} \quad \forall j \in [1, D]$$

Έτσι λοιπόν ένας τρόπος αρχικοποίησης του πληθυσμού είναι (Price et al, 2005):

$$P^{(0)} = x_{i,j}^{(0)} = x_j^{(L)} + \text{rand}[0,1](x_j^{(U)} - x_j^{(L)}) \quad \forall i \in [1, NP] \text{ και } \forall j \in [1, D] \quad (4)$$

όπου  $\text{rand}[0,1]$  μία τυχαία τιμή που κυμαίνεται από το 1 μέχρι το 0.

Στη φάση της **μετάλλαξης (mutation)**, από ένα διάνυσμα του πληθυσμού (γονέας) και από δύο άλλα τυχαία διανύσματα παράγεται το διάνυσμα μετάλλαξης ( $v_i$ ) με τον εξής τρόπο:

$$v_i^{G+1} = X_{r1}^G + F(X_{r2}^G - X_{r3}^G) \quad (5)$$

όπου

$r1, r2, r3 \in [1, NP]$ , τυχαίες τιμές με την προϋπόθεση ότι  $r1 \neq r2 \neq r3 \neq i$

$F$ , ο τελεστής κανονικότητας ο οποίος παίρνει συνήθως τιμές από 0 μέχρι 1. Όσο μικρότερος είναι τόσο περισσότερο χρόνο θα χρειαστεί για να πραγματοποιηθεί η σύγκλιση ενώ το αντίθετο συμβαίνει για μεγάλες τιμές του  $F$ .

Αφού υπολογιστεί το διάνυσμα μετάλλαξης στη συνέχεια δημιουργείται το δοκιμαστικό διάνυσμα ( $u_i$ ) σύμφωνα με το **κριτήριο διασταύρωσης (crossover)**:

$$u_{i,j}^{G+1} = \begin{cases} v_{i,j}^{G+1}, & \text{εάν } \text{rand}(0,1) \leq CR \text{ ή } j = j_{\text{rand}} \\ x_{i,j}^G, & \text{αλλιώς} \end{cases} \quad \forall i \in [1, NP] \text{ και } \forall j \in [1, D] \quad (6)$$

όπου

$CR$ , ο τελεστής διασταύρωσης ο οποίος παίρνει τιμές από 0 μέχρι 1

$j_{\text{rand}}$ , ένας ακέραιος αριθμός από 1 μέχρι D

Δηλαδή αν ένας τυχαίος αριθμός στο διάστημα (0,1) είναι μικρότερος από τον τελεστή διασταύρωσης, τότε το δοκιμαστικό διάνυσμα λαμβάνει το γονίδιο του διανύσματος μετάλλαξης, διαφορετικά, το γονίδιο του γονέα. Γίνεται αντιληπτό ότι όσο η τιμή του τελεστή διασταύρωσης είναι πιο κοντά (ή ίση) με 1 τόσο πιο πολλά γονίδια (ή όλα τα γονίδια) του δοκιμαστικού διανύσματος κληρονομούνται από το διάνυσμα μετάλλαξης αλλιώς αν είναι κοντά (ή ίσο) με το 0 τόσο πιο πολλά γονίδια (ή όλα τα γονίδια) κληρονομούνται από τον γονέα (Μαρινάκης et al, 2011). Ο αριθμός  $j_{\text{rand}}$  ουσιαστικά εξασφαλίζει ότι θα περάσει στο δοκιμαστικό διάνυσμα τουλάχιστον ένα γονίδιο από το διάνυσμα μετάλλαξης (Storn & Price, 1997).

Μετά τη διασταύρωση πραγματοποιείται το **στάδιο της επιλογής (selection)** όπου καθορίζεται ποιο διάνυσμα θα περάσει στην επόμενη γενιά. Η σύγκριση γίνεται μεταξύ του δοκιμαστικού διανύσματος και του γονέα σύμφωνα με την ποιότητα των αντικειμενικών τους συναρτήσεων ( $f$ ):

$$X_i^{G+1} = \begin{cases} u_i^{G+1}, & \text{εάν } f(u_i^{G+1}) \leq f(X_i^G) \\ X_i^G, & \text{αλλιώς} \end{cases} \quad (7)$$

Συνοψίζοντας, τα βήματα του DE είναι τα εξής:

1. αρχικοποίηση πληθυσμού
2. αξιολόγηση των αντικειμενικών συναρτήσεων όλων των ατόμων του πληθυσμού
3. για κάθε μέλος του πληθυσμού: μετάλλαξη, διασταύρωση, επιλογή, ενημέρωση νέας γενιάς
4. αν το κριτήριο τερματισμού ικανοποιείται τότε τερματισμός, διαφορετικά επιστροφή στο βήμα 3

Παραπάνω παρουσιάστηκε η βασική στρατηγική για τον υπολογισμό του δοκιμαστικού διανύσματος. Οι Price και Storn (2001) στην επίσημη ιστοσελίδα του DE<sup>3</sup> προτείνουν συνολικά δέκα διαφορετικές στρατηγικές (Πίνακας 3). Οι στρατηγικές έχουν τη μορφή DE/α/β/γ όπου, το DE σημαίνει ότι χρησιμοποιείται η μέθοδος της διαφορικής εξέλιξης, το α αναπαριστά το είδος του διανύσματος που προκαλεί τη διαταραχή, το β αναφέρεται στο πλήθος των εξισώσεων διαφορών και το γ στον τύπο της διασταύρωσης που χρησιμοποιείται.

Δύο είναι οι βασικοί τύποι της διασταύρωσης και συμβολίζονται με exp (exponential recombination) και bin (binomial recombination) (Storn & Price, 1997). Η διαφορά τους είναι ότι στην πρώτη περίπτωση τα γονίδια του δοκιμαστικού διανύσματος λαμβάνονται από τον γονέα μέχρι μια τυχαία τιμή από 0 μέχρι 1 να ξεπεράσει την τιμή του τελεστή διασταύρωσης και έπειτα λαμβάνονται από το διάνυσμα μετάλλαξης, ενώ στη δεύτερη περίπτωση κάθε γονίδιο του δοκιμαστικού διανύσματος λαμβάνεται είτε από τον γονέα είτε από το διάνυσμα μετάλλαξης σύμφωνα με την κλασική περίπτωση διασταύρωσης (De La Fraga & Coello Coello, 2011).

Επίσης, μερικές στρατηγικές χρησιμοποιούν το διάνυσμα  $X_{best}$  το οποίο συμβολίζει το καλύτερο ποιοτικά διάνυσμα του πληθυσμού. Επιπλέον στις στρατηγικές 3 και 8 (Πίνακας 3) χρησιμοποιείται η τιμή  $c$  η οποία υποδηλώνει το βαθμό επιρροής του καλύτερου και του τυχαίου διανύσματος. Ο αλγόριθμος που έχει αναπτυχθεί από τους Price & Storn (2001) θέτει  $c = F$  κάτι που θα χρησιμοποιηθεί στην παρούσα εργασία, άρα οι παράμετροι του αλγόριθμου που θα πρέπει να προσδιοριστούν είναι οι:  $F$ ,  $CR$  και  $NP$  καθώς και ο αριθμός επαναλήψεων  $GMAX$  (όπου  $GMAX$  ο αριθμός της μεγαλύτερης γενιάς).

---

<sup>3</sup> <http://www1.icsi.berkeley.edu/~storn/code.html>

Πίνακας 3. στρατηγικές αλγορίθμου διαφορικής εξέλιξης

στρατηγική		μοντελοποίηση
1. DE/best/1/exp	6. DE/best/1/bin	$u_i = X_{best} + F(X_{r1} - X_{r2})$
2. DE/rand/1/exp	7. DE/rand/1/bin	$u_i = X_{r1} + F(X_{r2} - X_{r3})$
3. DE/rand-to-best/1/exp	8. DE/rand-to best/1/bin	$u_i = X_i + c(X_{best} - X_i) + F(X_{r1} - X_{r2})$
4. DE/best/2/exp	9. DE/best/2/bin	$u_i = X_{best} + F(X_{r1} + X_{r2} - X_{r3} - X_{r4})$
5. DE/rand/2/exp	10. DE/rand/2/bin	$u_i = X_{r5} + F(X_{r1} + X_{r2} - X_{r3} - X_{r4})$

Θα πρέπει να τονιστεί επιπλέον ότι ο DE λειτουργεί στο πεδίο των συνεχών τιμών, άρα η εφαρμογή του σε προβλήματα όπως το FSSP απαιτεί την κατασκευή συναρτήσεων για μετατροπή των αριθμών από συνεχείς σε ακέραιους και το αντίθετο.

Στη συνέχεια παρουσιάζεται ο DE:

#### ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ

προσδιορισμός παραμέτρων DE: F, CR, NP, D, GMAX, στρατηγική μετάλλαξης - διασταύρωσης  
είσοδος δεδομένων

αρχικοποίηση πληθυσμού

υπολογισμός ποιότητας του κάθε μέλους του πληθυσμού

εύρεση βέλτιστου μέλους

γενιά = 0

**ΟΣΟ** γενιά < GMAX **ΕΠΑΝΑΛΑΒΕ**

**ΓΙΑ** κάθε μέλος του πληθυσμού **ΕΠΑΝΑΛΑΒΕ**

μετατροπή διακριτών σε συνεχείς τιμές /\*αν χρειάζεται\*/

δημιουργία δοκιμαστικού διανύσματος

μετατροπή συνεχών σε διακριτές τιμές /\*αν χρειάζεται\*/

σύγκριση δοκιμαστικού διανύσματος με παλιό πληθυσμό

ενημέρωση νέας γενιάς

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

υπολογισμός ποιότητας κάθε μέλους

εύρεση βέλτιστου μέλους

γενιά = γενιά + 1

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

εξαγωγή καλύτερου μέλους του πληθυσμού

<b>ΤΕΡΜΑΤΙΣΜΟΣ ΑΛΓΟΡΙΘΜΟΥ</b>
-------------------------------

Εικόνα 7. αλγόριθμος διαφορικής εξέλιξης

### 1.2.2. ΑΛΓΟΡΙΘΜΟΙ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

Οι τεχνικές τοπικής αναζήτησης (Local Search - LS) προϋποθέτουν την ύπαρξη μιας αρχικής λύσης στην οποία εφαρμόζονται διάφορες αλλαγές προκειμένου να βρεθεί μία καλύτερη λύση. Στην πλειονότητά τους είναι απλοί αλγόριθμοι, οι οποίοι χρησιμοποιούνται σε συνδυασμό με άλλους αλγορίθμους για να αποφευχθούν τοπικά βέλτιστα.

Η λογική των αλγορίθμων LS είναι αρκετά απλή. Αρχικά επιλέγεται μία λύση από το χώρο αναζήτησης (γειτονιά λύσεων, N) και ονομάζεται τρέχουσα. Στη συνέχεια εφαρμόζεται ένας μετασχηματισμός στην τρέχουσα λύση με αποτέλεσμα τη δημιουργία μιας νέας λύσης. Έπειτα γίνεται η σύγκριση της νέας με την τρέχουσα λύση και στην περίπτωση που η πρώτη είναι καλύτερη αντικαθιστά την τρέχουσα λύση, διαφορετικά απορρίπτεται. Η διαδικασία επαναλαμβάνεται συνήθως μέχρι να μην υπάρχει δυνατότητα περαιτέρω βελτίωσης της λύσης ή μετά από κάποιο συγκεκριμένο αριθμό επαναλήψεων.

Υπάρχει πληθώρα αλγορίθμων LS που μπορούν να εφαρμοστούν σε προβλήματα τύπου flow shop scheduling. Στη συγκεκριμένη εργασία θα αναπτυχθούν οι μετασχηματισμοί 1-1 exchange, 1-0 relocate και mirror.

Στην περίπτωση του 1-1 exchange δύο κόμβοι αλλάζουν θέση μεταξύ τους, ενώ στην περίπτωση του 1-0 relocate ένας κόμβος εισάγεται σε μία άλλη θέση «παραμερίζοντας» τους άλλους κόμβους. Για παράδειγμα το διάνυσμα 3 8 5 10 2 7 4 9 1 6 με την τεχνική 1-1 exchange στους κόμβους των θέσεων 6 και 8 θα γίνει 3 8 5 10 2 9 4 7 1 6, ενώ με την τεχνική 1-0 relocate της επανατοποθέτησης του κόμβου της θέσης 8 στην 6 θα γίνει 3 8 5 10 2 9 7 4 1 6.



Ο μετασχηματισμός mirror αντιστρέφει αντικατοπτρικά τις τιμές μεταξύ δύο τυχαίων θέσεων του διανύσματος λύσης. Για παράδειγμα αν οι τυχαίες θέσεις είναι η 3 και η 8 ενός διανύσματος 3 8 5 10 2 7 4 9 1 6 τότε το αποτέλεσμα μετά από την εφαρμογή της mirror θα είναι 3 8 9 4 7 2 10 5 1 6:

λύση πριν:	<div><div>3</div><div>8</div></div>	5	10	2		7	4	9	<div><div>1</div><div>6</div></div>
λύση μετά:	<div><div>3</div><div>8</div></div>	9	4	7		2	10	5	<div><div>1</div><div>6</div></div>

Εικόνα 8. εφαρμογή μετασχηματισμού mirror

Στη συνέχεια θα παρουσιαστούν μεθευρετικοί αλγόριθμοι βασισμένοι στη γειτονιά αναζήτησης.

### 1.2.3. ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ

Ο αλγόριθμος προσομοιωμένης ανόπτησης (Simulated Annealing Algorithm – SA) αποτελεί έναν από τους γνωστότερους μεθευρετικούς αλγορίθμους και πρωτοεμφανίστηκε το 1983 από δημοσίευση των Kirkpatrick et al. Ο αλγόριθμος είναι εμπνευσμένος από τη φυσική κατεργασία της ανόπτησης, η οποία είναι μία τεχνική που εφαρμόζεται στα μέταλλα και τα κράματα. Το σώμα θερμαίνεται μέχρι το σημείο τήξης και στη συνέχεια ψύχεται αργά με σκοπό τη βελτίωση κάποιων χαρακτηριστικών του.

Στη συνδυαστική βελτιστοποίηση ο αλγόριθμος της προσομοιωμένης ανόπτησης λειτουργεί ως εξής: Σε κάθε επανάληψη επιλέγεται μια τυχαία κίνηση και αν βελτιώνει την προϋπάρχουσα κατάσταση τότε η κίνηση γίνεται αποδεκτή. Στην περίπτωση που η κίνηση αδυνατεί να βελτιώσει την κατάσταση, δε σημαίνει ότι σίγουρα απορρίπτεται, διότι μπορεί να γίνει αποδεκτή με βάση κάποια πιθανότητα. Η πιθανότητα αυτή μειώνεται εκθετικά σύμφωνα με κάποια παράμετρο  $\alpha$  που σχετίζεται με τη θερμοκρασία  $T$ .

Πιο συγκεκριμένα, γίνεται η επιλογή μιας αρχικής λύσης  $s$  από την οποία με τυχαία διαταραχή γεννάται η λύση  $s'$ . Στη συνέχεια υπολογίζονται οι αντικειμενικές συναρτήσεις  $f(s)$  και  $f(s')$  των  $s$  και  $s'$  αντίστοιχα. Αν  $f(s) > f(s')$  τότε η  $s'$  παίρνει τη θέση της  $s$  ενώ αν η  $s'$  δεν έχει καλύτερη αντικειμενική συνάρτηση υπολογίζεται η τιμή  $e^{\frac{f(s)-f(s')}{T}}$ . Η τιμή αυτή αν είναι μεγαλύτερη από μια τυχαία τιμή μεταξύ 0 και 1 τότε η  $s'$  παίρνει τη θέση της  $s$ . Ο αλγόριθμος επαναλαμβάνει τη διαδικασία για τη λύση  $s$  και αφού τελειώνει η κάθε επανάληψη επαναυπολογίζεται η θερμοκρασία  $T = \alpha(t)$ . Συνήθως ο αλγόριθμος συγκλίνει όταν η θερμοκρασία έχει φτάσει σε ένα προκαθορισμένο χαμηλό στάδιο.

#### ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ

προσδιορισμός ελάχιστης θερμοκρασίας

προσδιορισμός προγράμματος μείωσης θερμοκρασίας

προσδιορισμός LS τεχνικής /\*ή κάποιας άλλης κίνησης\*/

είσοδος αρχικής λύσης

αρχικοποίηση θερμοκρασίας

**ΟΣΟ** θερμοκρασία είναι μεγαλύτερη από την ελάχιστη θερμοκρασία **ΕΠΑΝΑΛΑΒΕ**

**ΟΣΟ** ο αριθμός των μέγιστων επαναλήψεων δεν έχει ολοκληρωθεί **ΕΠΑΝΑΛΑΒΕ**

δημιουργία νέας λύσης με εφαρμογή LS τεχνικής

υπολογισμός ποιότητας νέας λύσης

**ΑΝ** η ποιότητα της νέας λύσης είναι καλύτερη από της παλιάς **TOTE**

στη θέση της λύσης περνάει η νέα λύση

**ΑΛΛΙΩΣ TOTE**

**ΑΝ**  $e^{\frac{\text{ποιότητα μέλους} - \text{ποιότητα νέας λύσης}}{\text{θερμοκρασία}}} > \text{random}(0,1)$  **TOTE**

στη θέση της λύσης περνάει η νέα λύση

**ΤΕΛΟΣ ΑΝ**

**ΤΕΛΟΣ ΑΝ**

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

επαναυπολογισμός θερμοκρασίας

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

εξαγωγή λύσης

**ΤΕΡΜΑΤΙΣΜΟΣ ΑΛΓΟΡΙΘΜΟΥ**

Εικόνα 9. αλγόριθμος προσομοιωμένης απόπτωσης

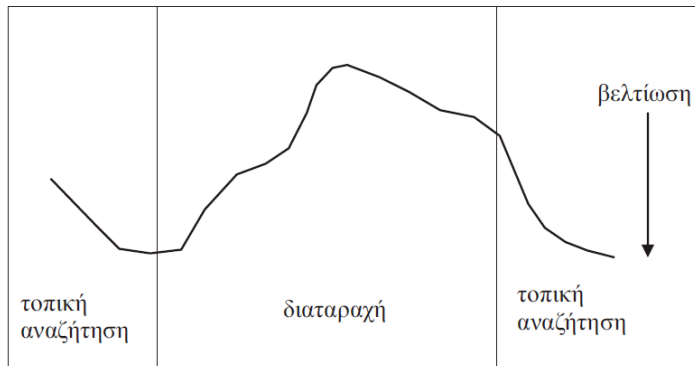
#### 1.2.4. ΑΛΓΟΡΙΘΜΟΣ ΕΠΑΝΑΛΗΠΤΙΚΗΣ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

Ο αλγόριθμος επαναληπτικής τοπικής αναζήτησης (Iterated Local Search Algorithm - ILS) ουσιαστικά έχει σχεδιαστεί για να αποφεύγεται ο εγκλωβισμός σε τοπικά ακρότατα. Σε μερικές περιπτώσεις η τοπική αναζήτηση δεν αρκεί για να ξεφύγει ο αλγόριθμος από μία τοπικά βέλτιστη λύση και απαιτείται να συμβεί κάτι παραπάνω.

Έτσι λοιπόν ο ILS βελτιώνει μια αρχική λύση με έναν οποιοδήποτε αλγόριθμο LS και όταν αυτός φτάνει στο σημείο όπου αδυνατεί να βελτιώσει τη λύση παραπάνω προκαλεί μια διαταραχή (perturbation) τροποποιώντας την ως ένα βαθμό. Στη συνέχεια εφαρμόζεται πάλι κάποιος αλγόριθμος LS ώστε να βελτιωθεί η νέα λύση. Είναι σημαντικό να κρατείται στη μνήμη η λύση πριν τη διαταραχή διότι υπάρχει πιθανότητα ο ILS να αποτύχει.

Ο Talbi (2009) διαχωρίζει τα τρία βασικά συστατικά του αλγορίθμου ως εξής:

- Τοπική αναζήτηση: Μπορεί να επιλεγεί οποιαδήποτε απλή μέθοδος LS ακόμα και πιο σύνθετη όπως η προσομοιωμένη ανόπτηση.
- Μέθοδος διαταραχής: Με τον όρο διαταραχή ουσιαστικά εννοείται μια μεγάλη τυχαία αλλαγή. Στην προκειμένη περίπτωση πρέπει να είναι αισθητή χωρίς όμως να αλλάζει εξ' ολοκλήρου η λύση. Θα πρέπει δηλαδή ένα κομμάτι της λύσης να παραμένει ως έχει.
- Κριτήριο αποδοχής: Το κριτήριο αποδοχής ορίζει τις συνθήκες τις οποίες το νέο τοπικό βέλτιστο πρέπει να ικανοποιεί για να αντικαταστήσει την τρέχουσα λύση.



Εικόνα 10. απεικόνιση επαναληπτικής τοπικής αναζήτησης

### 1.2.5 ΑΛΓΟΡΙΘΜΟΣ ΜΕΤΑΒΛΗΤΗΣ ΓΕΙΤΟΝΙΑΣ ΑΝΑΖΗΤΗΣΗΣ

Ο αλγόριθμος της μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search Algorithm - VNS) ο οποίος προτάθηκε για πρώτη φορά το 2001 από τους Hansen και Mladenovic, ανήκει και αυτός στην ευρύτερη κατηγορία των μεθευρετικών αλγορίθμων που είναι βασισμένοι στη γειτονιά αναζήτησης. Η διαφορά του όμως είναι ότι αλλάζει συστηματικά τη γειτονιά αναζήτησης μέσα από τυχαίες μεταβολές αλγορίθμων LS.

Αρχικά επιλέγεται ένα σύνολο από γειτονιές λύσεων,  $N_i$  όπου  $i = 1, \dots, i_{\max}$  και στη συνέχεια για την κάθε γειτονιά  $N_i$  δημιουργείται από την αρχική λύση  $s$  η λύση  $s'$ . Έπειτα για τη συγκεκριμένη γειτονιά λύσεων εφαρμόζεται κάποιος αλγόριθμος LS και παράγεται η λύση  $s''$ . Αν η λύση  $s''$  είναι καλύτερη από την  $s$  τότε επιλέγεται η  $s''$  ως αρχική λύση και η διαδικασία ξεκινάει πάλι από την αρχή για την ίδια γειτονιά. Στην περίπτωση που η λύση δε βελτιώνεται σε μια γειτονιά ο αλγόριθμος εξετάζει την επόμενη  $N_{i+1}$ .

#### ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ

επιλογή συνόλου γειτονιών  $N_i$  με  $i = 1, \dots, i_{\max}$

είσοδος αρχικής λύσης  $s$

$i = 1$

#### ΕΠΑΝΑΛΑΒΕ

δημιουργία λύσης  $s'$  από την αρχική λύση  $s$

δημιουργία νέας λύσης  $s''$  με εφαρμογή μετασχηματισμού LS της  $N_i$  στη λύση  $s'$

**ΑΝ** η ποιότητα της νέας λύσης είναι καλύτερη από της παλιάς **TOTE**

στη θέση της λύσης  $s$  περνάει η νέα λύση  $s''$

$i = 1$

#### ΑΛΛΙΩΣ

$i = i + 1$

#### ΤΕΛΟΣ ΑΝ

**ΜΕΧΡΙΣ ΟΤΟΥ**  $i \leq i_{\max}$

εξαγωγή λύσης

#### ΤΕΡΜΑΤΙΣΜΟΣ ΑΛΓΟΡΙΘΜΟΥ

Εικόνα 11. αλγόριθμος μεταβλητής γειτονιάς αναζήτησης

## 2. ΣΥΝΤΟΜΗ ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Στη συγκεκριμένη ενότητα θα εξεταστούν προγενέστερες μελέτες στο πεδίο του χρονικού προγραμματισμού των FSSP και FFSSP με χρήση μεθευρετικών αλγορίθμων. Υπάρχει ένας μεγάλος αριθμός μελετών πάνω στο συγκεκριμένο αντικείμενο και ήδη υπάρχουν μερικές σχετικά πρόσφατες ανασκοπήσεις (Yagmahan & Yenisey, 2009; Ruiz & Vázquez-Rodríguez, 2010; Pan & Ruiz, 2011; Hooda & Dhingra 2011) και παλαιότερες (Hejazi & Saghafeian, 2005). Η συγκεκριμένη βιβλιογραφική ανασκόπηση θα εστιάσει κυρίως στους αλγόριθμους που απασχολούν την παρούσα εργασία ώστε να εντοπιστούν σημεία τα οποία θα φανούν χρήσιμα για την περαιτέρω εξέλιξη και την πρόταση νέων αλγορίθμων.

Μία από τις πρώτες εφαρμογές του DE στο FSSP έγινε από τους Tasgetiren et al (2004), οι οποίοι διαμόρφωσαν έναν DE με LS εντάσσοντας τη μέθοδο smallest position value για την μετατροπή των συνεχών τιμών σε διακριτές. Τα αποτελέσματά τους ήταν ενθαρρυντικά για τη συνέχιση της έρευνας στην εφαρμογή του DE στο FSSP. Αργότερα μία «καθαρή εφαρμογή» του DE δημοσιεύτηκε το 2006 από τους Onwubolu και Davendra, όπου δοκιμάστηκαν όλες οι στρατηγικές του αλγόριθμου με αποτέλεσμα η έβδομη να αναδειχτεί ως η πιο αποτελεσματική για το πρόβλημα. Επίσης ο αλγόριθμος στη σύγκρισή του με τον γενετικό φάνηκε να είναι πιο αποδοτικός για τα μικρά προβλήματα. Αποτελεσματικότεροι για το κριτήριο makespan ήταν ο αλγόριθμος διακριτής διαφορικής εξέλιξης (discrete differential evolution) (Pan et al 2007) και ο επαναληπτικός αλγόριθμος απληστίας (Ruiz & Strutzle, 2007) τους οποίους οι Pan et al (2008) εφάρμοσαν για το κριτήριο total flowtime και έπειτα τους συνδύασαν μεταξύ τους με αρκετά θετικά αποτελέσματα. Άλλη παραλλαγή είναι αυτή των Qian et al (2008) οι οποίοι πρότειναν έναν υβριδικό DE που εμπεριέχει μία αναζήτηση βασισμένη στη διαφορική εξέλιξη και μία στην τοπική αναζήτηση, ενώ οι Davendra και Onwubolu (2007) πρότειναν μια ενισχυμένη έκδοση του DE (enhanced differential evolution) στον οποίο δεν πρόσθεσαν απλά μετασχηματισμό LS, αλλά και δύο ακόμα απλές τεχνικές βελτίωσης. Υποστήριζαν επίσης ότι ο ενισχυμένος DE είναι πιο αποτελεσματικός από τον DE με LS. Επίσης, οι Davendra et al (2009) πρότειναν μια τροποποιημένη μορφή του DE όπου τα μέλη του πληθυσμού ομαδοποιούνται με βάση τις

αποκλίσεις που έχουν μεταξύ τους. Ενδιαφέρον είναι επίσης ότι η αρχικοποίηση του πληθυσμού δεν έγινε μόνο με τυχαίο τρόπο αλλά και με κατασκευαστικό. Οι ίδιοι υποστηρίζουν ότι η μέθοδος αυτή είναι μία από τις πιο αποτελεσματικές για το FSSP. Οι Xu, Xiang et al (2010) χρησιμοποίησαν μία αυτοπροσαρμοστική τεχνική για τον προσδιορισμό των παραμέτρων του αλγορίθμου DE και κατάληξαν ότι μπορεί να εξισορροπήσει τη LS για το FSSP. Αλλά και για την επίλυση του FFSSP έχει προταθεί εφαρμογή του DE για τη βελτιστοποίηση του κριτηρίου earliness/ tardiness (Zhonghua et al, 2009) και για το κριτήριο makespan (Xu & Wang, 2011).

Στη βιβλιογραφική ανασκόπηση των Hooda και Dhingra (2011) είναι φανερό ότι πολύ μελετητές έχουν προσπαθήσει να προσεγγίσουν το FSSP με τη μέθοδο της προσομοιωμένης ανόπτησης. Από την έρευνά τους προκύπτει ότι για το συγκεκριμένο πρόβλημα δεν αρκεί ο απλός SA, αλλά ένας υβριδικός ή ένας ουσιαστικά τροποποιημένος. Για παράδειγμα ο Nearchou (2004) κατασκεύασε έναν αλγόριθμο βασισμένο στον SA σε συνδυασμό με στοιχεία από γενετικούς αλγορίθμους (Genetic Algorithm – GA) και τεχνικών LS. Ο αλγόριθμος απέδωσε καλά αποτελέσματα για μεγάλα προβλήματα και φάνηκε να είναι πιο αποδοτικός από τους κλασικούς SA και GA. Υβριδικό SA με στοιχεία GA προτείνουν και οι Azizi et al (2009). Οι Low, Yeh και Huang (2004) πρότειναν μια παραλλαγή του SA (robust simulated annealing algorithm) στην οποία όταν για κάποιο συνεχόμενο αριθμό επαναλήψεων η λύση δε βελτιώνεται τότε αυτή αρχικοποιείται με τον αλγόριθμο NEH<sup>4</sup> εκ νέου. Και οι Rajkumar και Shahabudeen (2007) χρησιμοποιούν τον αλγόριθμο NEH για την παραγωγή μιας λύσης στην οποία δοκίμασαν διάφορες τεχνικές LS ούτως ώστε να δημιουργηθεί η αρχική λύση για τον SA. Επιπλέον, ο SA έχει χρησιμοποιηθεί και για το FFSSP (Younes et al, 1998; Low, 2005; Jungwattanakit et al, 2007) καθώς οι Zhang et al (2003) υποστήριξαν τη χρησιμότητα του SA στο FFSP με στοχαστικούς χρόνους επεξεργασίας.

---

<sup>4</sup> Ο αλγόριθμος NEH προτάθηκε από τους Nawaz, Enscoe και Ham (1983) και θεωρείται μία από τις καλύτερες ευρετικές μεθόδους για το FSSP. Βασίζεται στη λογική ότι οι εργασίες με μεγαλύτερη χρονική διάρκεια επεξεργασίας θα πρέπει να γίνουν όσο το δυνατό γρηγορότερα. Συχνά χρησιμοποιείται ως αρχική λύση για τους μεθευρετικούς αλγόριθμους.

Όσον αφορά τον ISL για την επίλυση του FSSP ο Stutzle (1998) τον εφάρμοσε σε αρχικές λύσεις, οι οποίες είχαν προκύψει από τον αλγόριθμο NEH, βρίσκοντας αρκετά καλά αποτελέσματα σε σύγκριση με άλλους μεθευρετικούς αλγορίθμους. Στη μελέτη του υποστηρίζει ότι το κριτήριο αποδοχής θα πρέπει να προκύπτει από αλγόριθμο τύπου προσομοιωμένης απόδοσης ο οποίος δε δέχεται μόνο τις καλές λύσεις. Αργότερα, οι Burke et al. (2010) ανέπτυξαν διάφορες παραλλαγές του ILS, παρ' όλα αυτά συμπέραναν ότι η απλή εκδοχή του αλγορίθμου είναι η πιο αποδοτική για τρία διαφορετικά προβλήματα ένα εκ των οποίων το FSSP. Οι Juan et al (2012) όμως, αναπτύξαν έναν αποτελεσματικό, απλό χωρίς παραμέτρους (effective, simple and parameter-free) αλγόριθμο βασισμένο στον ILS. Ενδιαφέρον έχει και ο άπληστος αλγόριθμος επαναληπτικής αναζήτησης (IGA) τον οποίο οι Ruiz και Strutzle (2007) ενδυνάμωσαν ακόμα περισσότερο.

Στη μελέτη του ο Duda (2006) επιχείρησε να διαπιστώσει αν οι συνδυασμοί μεθευρετικών αλγορίθμων με τεχνικές LS είναι πιο αποτελεσματικοί από μεθευρετικούς αλγορίθμους που είναι βασισμένοι στη γειτονιά αναζήτησης. Συγκρίνοντας έναν υβριδικό GA με τον ILS και έναν υβριδικό αλγόριθμο σμήνους σωματιδίων (Kennedy & Eberhart, 1995) με τον VNS κατέληξε ότι οι μεθευρετικοί αλγόριθμοι που είναι βασισμένοι στη γειτονιά αναζήτησης είναι πιο αποτελεσματικοί. Στην ίδια λογική κυμαίνεται και η δημοσίευση των Pan και Ruiz (2011) στην οποία υποστήριξαν ότι απλές μεθευρετικές μέθοδοι όπως αυτές των ILS και IGA είναι προτιμότερες από τους πολύ σύνθετους και περίπλοκους μεθευρετικούς αλγορίθμους που έχουν προταθεί ανά καιρούς.

Τέλος, οι Costa et al (2011) εξέτασαν διάφορες τεχνικές LS οι οποίες εντάχθηκαν στον αλγόριθμο VNS στην βιβλιογραφία και κατέληξαν ότι ο συνδυασμός των τεχνικών insert και interchange είναι ο καλύτερος. Στην πρώτη τεχνική επιλέγεται τυχαία μία εργασία η οποία εισάγεται σε μια τυχαία θέση χωρίς να αλλάξει τη σειρά των άλλων εργασιών και η δεύτερη ουσιαστικά είναι η 1-1 exchange τεχνική που περιγράφηκε στην ενότητα 1.

Τα γενικά συμπεράσματα που προκύπτουν από τη βιβλιογραφική ανασκόπηση είναι ότι κλασικοί μεθευρετικοί αλγόριθμοι τύπου DE και GA δεν είναι οι καλύτερες επιλογές για τα προβλήματα που μελετούνται. Είναι απαραίτητο να συνδυαστούν με άλλους αλγορίθμους,

τόσο με απλές τεχνικές LS, όσο με μεθευρετικούς αλγορίθμους βασισμένους στη γειτονιά αναζήτησης. Στη συνέχεια λοιπόν θα αναπτυχθούν υβριδικοί αλγόριθμοι διαφορετικής εξέλιξης, άλλοι σύνθετοι και άλλοι πιο απλοί, για να διασταυρωθούν οι απόψεις που επικρατούν στη βιβλιογραφία και να προταθούν νέοι τρόποι επίλυσης του FSSP.



### 3. ΕΦΑΡΜΟΓΗ ΣΕ ΒΙΒΛΙΟΓΡΑΦΙΚΑ ΔΕΔΟΜΕΝΑ

Οι αλγόριθμοι που περιγράφηκαν στην ενότητα 1 κωδικοποιήθηκαν σε γλώσσα προγραμματισμού C και εφαρμόστηκαν σε FSSP ποικίλων μεγεθών όπως δημοσιεύτηκαν από τον Taillard (1993)<sup>5</sup>. Ο Taillard έδωσε τα ανώτερα και τα κατώτερα όρια της τιμής του makespan σε προβλήματα: 20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20, 200x10 και 200x20 (εργασίες x μηχανές). Η κάθε ομάδα προβλημάτων περιέχει δέκα προβλήματα. Το ανώτερο όριο προσεγγίζεται από ευρετικούς αλγορίθμους. Όσο πιο κοντά στο ανώτερο όριο είναι η λύση που βρίσκει ο αλγόριθμος τόσο καλύτερη απόδοση έχει. Με αυτόν τον τρόπο θα γίνει η αξιολόγηση των αλγορίθμων που αναπτύσσονται στην παρούσα εργασία ως προς την ποιότητα της λύσης τους. Ανά καιρούς έχουν δημοσιευτεί έρευνες οι οποίες αλλάζουν το ανώτερο όριο του Taillard, δηλαδή βρίσκουν καλύτερες λύσεις για κάποια προβλήματα (Pan et al, 2008).

#### 3.1. ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΟΥ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ

Ο σκοπός της συγκεκριμένης ενότητας είναι να διερευνήσει τον DE μιας και θα αποτελέσει την βάση των υβριδικών αλγορίθμων και να συγκρίνει την απόδοση των δέκα στρατηγικών των Price και Storn. Αφού λοιπόν παρουσιαστούν τα βασικά σημεία του αλγορίθμου και οι συναρτήσεις που τον απαρτίζουν, θα γίνει η προσπάθεια προσδιορισμού των παραμέτρων και της κατάλληλης στρατηγικής του αλγορίθμου για το FSSP μέσα από στατιστική διαδικασία.

##### 3.1.1. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΛΗΘΥΣΜΟΥ

Ο DE είναι αλγόριθμος βασισμένος στον πληθυσμό. Στην περίπτωση του FSSP το κάθε μέλος του πληθυσμού είναι μια εφικτή λύση και ουσιαστικά μία πιθανή αλληλουχία

---

<sup>5</sup> Τα δεδομένα μπορούν να βρεθούν στην ηλεκτρονική σελίδα:  
<http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>

εργασιών. Για παράδειγμα ο πληθυσμός με τρία μέλη σε ένα πρόβλημα 5 μηχανών και 5 εργασιών θα μπορούσε να είναι ο εξής:

1 5 3 4 2

2 3 4 5 1

4 2 3 1 5

Άρα ο αριθμός των μεταβλητών απόφασης (D), ο οποίος στην προκειμένη περίπτωση είναι ο αριθμός των εργασιών και ο αριθμός των μελών του πληθυσμού (NP), θα είναι  $D = 5$  και  $NP = 3$ .

Ο πληθυσμός θα μπορούσε να αναπαρασταθεί ως πίνακας όπου η κάθε γραμμή είναι ένα διαφορετικό μέλος του, δηλαδή μια διαφορετική εφικτή λύση:

$$P_{i*} = \begin{bmatrix} P_{1,1} & \cdots & P_{1,D} \\ \vdots & \ddots & \vdots \\ P_{NP,1} & \cdots & P_{NP,D} \end{bmatrix} = \begin{bmatrix} P_{1*} \\ \vdots \\ P_{NP*} \end{bmatrix}$$

όπου  $P_{i*}$  τα μέλη του πληθυσμού με  $i \in [1, NP]$

Υπάρχουν διάφορες τεχνικές για την αρχικοποίηση του πληθυσμού και η πιο συνηθισμένη είναι αυτή της σχέσης (4). Αν το ανώτερο και το κατώτερο όριο ληφθούν ως 1 και -1 αντίστοιχα τότε η συνάρτηση αρχικοποίησης θα είναι η παρακάτω:

**ΓΙΑ** κάθε μέλος του πληθυσμού i **ΕΠΑΝΑΛΑΒΕ**

**ΓΙΑ** κάθε στοιχείο j **ΕΠΑΝΑΛΑΒΕ**

$P[i][j] = -1 + 2 * \text{genrand}() / * \text{genrand}()$  γεννήτρια τυχαίων αριθμών  $[0,1]*/$

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

Εικόνα 12. συνάρτηση αρχικοποίησης πληθυσμού

Με αυτόν τον τρόπο αρχικοποιούνται οι τιμές στο συνεχές πεδίο αριθμών.

### 3.1.2. ΑΞΙΟΛΟΓΗΣΗ ΑΝΤΙΚΕΙΜΕΝΙΚΗΣ ΣΥΝΑΡΤΗΣΗΣ

Το κριτήριο προς ελαχιστοποίηση είναι το makespan, δηλαδή ο χρόνος περάτωσης της τελευταίας εργασίας. Σύμφωνα με τη σχέση (1) η αντικειμενική συνάρτηση του προβλήματος είναι:

$$\min (C_{\max}) = \min (\max (C_{iM})), \quad i = 1, 2, \dots, n$$

όπου  $C_{\max}$  ο χρόνος λήξης του προγράμματος (makespan) και  $C_{iM}$  ο χρόνος περάτωσης κάθε εργασίας  $i$  στην τελευταία μηχανή  $M$ .

Λαμβάνοντας υπόψη τη σχέση (2) για τον υπολογισμό περάτωσης της κάθε εργασίας στην κάθε μηχανή, δημιουργείται ο πίνακας κόστους  $C$  του οποίου οι σειρές συμβολίζουν τις μηχανές και οι στήλες τις εργασίες. Στην πρώτη μηχανή οι εργασίες εκτελούνται σειριακά χωρίς να αναγκαστεί η μηχανή να «περιμένει» για την επόμενη εργασία. Επίσης, η πρώτη εργασία εκτελείται σειριακά χωρίς να «περιμένει» τις μηχανές. Για όλες τις υπόλοιπες περιπτώσεις όμως θα πρέπει να υπολογιστεί αν όταν μία εργασία φτάνει στη μηχανή είναι αναγκασμένη να «περιμένει» τη μηχανή ή αν η μηχανή είναι διαθέσιμη και «περιμένει» την εργασία:  $C_{i,j} = \max(C_{i-1,j}, C_{i,j-1}) + p_{i,j}$  (όπου  $p_{i,j}$  οι χρόνοι επεξεργασίας της εργασίας  $j$  στη μηχανή  $i$ ).

Ο πίνακας κόστους των δεδομένων του παραδείγματος της ενότητας 1.1.1 (Πίνακας 2) με

$$\text{σειρά εργασιών } E1, E2, E3, E4 \text{ θα είναι: } C = \begin{vmatrix} 5 & 15 & 17 & 21 \\ 9 & 18 & 24 & 32 \\ 12 & 20 & 32 & 41 \\ 22 & 27 & 35 & 42 \end{vmatrix}$$

Το στοιχείο προς ελαχιστοποίηση, δηλαδή το makespan, είναι αυτό της τελευταίας σειράς και της τελευταίας γραμμής. Άρα το makespan του παραδείγματος είναι  $C_{\max} = 42$ .

Έτσι λοιπόν η αξιολόγηση της αντικειμενικής συνάρτησης γίνεται με βάση τον πίνακα C. Θα πρέπει λοιπόν για όλα τα μέλη του πληθυσμού να δημιουργείται ο πίνακας κόστους και με βάση αυτόν να γίνεται η επιλογή του καλύτερου μέλους.

```
/* όπου M ο αριθμός των μηχανών, D ο αριθμός των εργασιών, array το διάνυσμα που ισούται με κάποιο μέλος του πληθυσμού, C ο πίνακας κόστους και p ο πίνακας δεδομένων (η χρονική διάρκεια που απαιτείται για κάθε εργασία σε κάθε μηχανή) */
```

**ΓΙΑ** κάθε μηχανή i **ΕΠΕΝΑΛΑΒΕ**

**ΓΙΑ** κάθε εργασία j **ΕΠΑΝΑΛΑΒΕ**

**ΑΝ** i = 1 **ΚΑΙ** j = 1 **ΤΟΤΕ**

$C[1][1] = p[1][array[1]]$

**ΑΛΛΙΩΣ** **ΑΝ** i = 1 **ΚΑΙ** j ≠ 1 **ΤΟΤΕ**

$C[1][j] = C[1][j-1] + p[1][array[j]]$

**ΑΛΛΙΩΣ** **ΑΝ** i ≠ 1 **ΚΑΙ** j = 1 **ΤΟΤΕ**

$C[i][1] = C[i-1][1] + p[i][array[1]]$

**ΑΛΛΙΩΣ** **ΑΝ** i ≠ 1 **ΚΑΙ** j ≠ 1 **ΚΑΙ**  $C[i-1][j] \geq C[i][j-1]$  **ΤΟΤΕ**

$C[i][j] = C[i-1][j] + p[i][array[j]]$

**ΑΛΛΙΩΣ**

$C[i][j] = C[i][j-1] + p[i][array[j]]$

**ΤΕΛΟΣ** **ΑΝ**

**ΤΕΛΟΣ** **ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ** **ΕΠΑΝΑΛΗΨΗΣ**

makespan=C[M][D]

Εικόνα 13. δημιουργία πίνακα κόστους για τον υπολογισμό του makespan

### 3.1.3. ΜΕΤΑΛΛΑΞΗ, ΔΙΑΣΤΑΥΡΩΣΗ, ΕΠΙΛΟΓΗ

Κάθε μέλος του πληθυσμού περνάει από τη φάση της μετάλλαξης και της διασταύρωσης σύμφωνα με τις σχέσεις (5) και (6). Οι Price και Storn έχουν κωδικοποιήσει σε διάφορες

γλώσσες προγραμματισμού και τις δέκα στρατηγικές που έχουν προτείνει<sup>6</sup>. Στην Εικόνα 14. παρουσιάζεται ο αλγόριθμος γραμμένος σε στιλ γλώσσας προγραμματισμού C για τη διαδικασία μετάλλαξης και διασταύρωσης του DE για κάποιο μέλος του πληθυσμού σύμφωνα με τον κώδικα που έχουν προτείνει οι Price και Storn.

/\*όπου trial το δοκιμαστικό διάνυσμα, P ο πίνακας πληθυσμού, CR ο τελεστής διασταύρωσης, F ο τελεστής κανονικότητας, D ο αριθμός των εργασιών και r1, r2, r3, r4, r5 τυχαία μέλη του πληθυσμού διαφορετικά μεταξύ τους \*/

επιλογή στρατηγικής

**AN** η στρατηγική είναι η DE/best/1/exp **TOTE**

L = 0

αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D

**ΕΠΑΝΑΛΑΒΕ**

trial[n] = P[best][n] + F (P[r2][n] - P[r3][n])

n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)

L = L + 1

} **ΜΕΧΡΙΣ ΟΤΟΥ** random(0,1) < CR **ΚΑΙ** L ≤ D

**ΑΛΛΙΩΣ** **AN** η στρατηγική είναι η DE/rand/1/exp **TOTE**

L = 0

αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D

**ΕΠΑΝΑΛΑΒΕ**

trial[n] = P[r1][n] + F (P[r2][n] - P[r3][n])

n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)

L = L + 1

} **ΜΕΧΡΙΣ ΟΤΟΥ** random(0,1) < CR **ΚΑΙ** L ≤ D

**ΑΛΛΙΩΣ** **AN** η στρατηγική είναι η DE/rand-to-best/1/exp **TOTE**

L = 0

αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D

**ΕΠΑΝΑΛΑΒΕ**

trial[n] = trial[n] + F (P[best][n] - trial[n]) + F (P[r1][n] - P[r2][n])

n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)

L = L + 1

} **ΜΕΧΡΙΣ ΟΤΟΥ** random(0,1) < CR **ΚΑΙ** L ≤ D

**ΑΛΛΙΩΣ** **AN** η στρατηγική είναι η DE/best/2/exp **TOTE**

L = 0

αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D

**ΕΠΑΝΑΛΑΒΕ**

trial[n] = P[best][n] + F (P[r1][n] + P[r2][n] - P[r3][n] - P[r4][n])

n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)

L = L + 1

} **ΜΕΧΡΙΣ ΟΤΟΥ** random(0,1) < CR **ΚΑΙ** L ≤ D

**ΑΛΛΙΩΣ** **AN** η στρατηγική είναι η DE/rand/2/exp **TOTE**

L = 0

αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D

**ΕΠΑΝΑΛΑΒΕ**

trial[n] = P[r5][n] + F (P[r1][n] + P[r2][n] - P[r3][n] - P[r4][n])

n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)

L = L + 1

} **ΜΕΧΡΙΣ ΟΤΟΥ** random(0,1) < CR **ΚΑΙ** L ≤ D

**ΑΛΛΙΩΣ** **AN** η στρατηγική είναι η DE/best/1/bin **TOTE**

αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D

<sup>6</sup> <http://www1.icsi.berkeley.edu/~storn/code.html>

```

ΓΙΑ L = 1 μέχρι L = D με βήμα 1 ΕΠΑΝΑΛΑΒΕ
  ΑΝ random(0,1) < CR Ή L = D ΤΟΤΕ
    trial[n] = P[best][n] + F (P[r2][n] - P[r3][n])
  ΤΕΛΟΣ ΑΝ
  n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ ΑΝ η στρατηγική είναι η DE/rand/1/bin ΤΟΤΕ
  αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D
  ΓΙΑ L = 1 μέχρι L = D με βήμα 1 ΕΠΑΝΑΛΑΒΕ
    ΑΝ random(0,1) < CR Ή L = D ΤΟΤΕ
      trial[n] = P[r1][n] + F (P[r2][n] - P[r3][n])
    ΤΕΛΟΣ ΑΝ
    n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)
  ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ ΑΝ η στρατηγική είναι η DE/rand-to-best/1/bin ΤΟΤΕ
  αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D
  ΓΙΑ L = 1 μέχρι L = D με βήμα 1 ΕΠΑΝΑΛΑΒΕ
    ΑΝ random(0,1) < CR Ή L = D ΤΟΤΕ
      trial[n] = trial[n] + F (P[best][n] - trial[n]) + F (P[r1][n] - P[r2][n])
    ΤΕΛΟΣ ΑΝ
    n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)
  ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ ΑΝ η στρατηγική είναι η DE/best/2/bin ΤΟΤΕ
  αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D
  ΓΙΑ L = 1 μέχρι L = D με βήμα 1 ΕΠΑΝΑΛΑΒΕ
    ΑΝ random(0,1) < CR Ή L = D ΤΟΤΕ
      trial[n] = P[best][n] + F (P[r1][n] + P[r2][n] - P[r3][n] - P[r4][n])
    ΤΕΛΟΣ ΑΝ
    n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)
  ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ ΑΝ η στρατηγική είναι η DE/rand/2/bin ΤΟΤΕ
  αρχικοποίηση του n με κάποια τυχαία τιμή από 1 μέχρι D
  ΓΙΑ L = 1 μέχρι L = D με βήμα 1 ΕΠΑΝΑΛΑΒΕ
    ΑΝ random(0,1) < CR Ή L = D ΤΟΤΕ
      trial[n] = P[r5][n] + F (P[r1][n] + P[r2][n] - P[r3][n] - P[r4][n])
    ΤΕΛΟΣ ΑΝ
    n = n + 1 (αν το n υπερβεί την τιμή του D τότε n=1)
  ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ ΑΝ

```

Εικόνα 14. υπολογισμός δοκιμαστικού διανύσματος για ένα μέλος του πληθυσμού ανάλογα με τη στρατηγική που θα επιλεγεί

Αφού ολοκληρωθούν η μετάλλαξη και η διασταύρωση, έχουν δημιουργηθεί τα δοκιμαστικά διανύσματα όπως περιγράφηκαν στην ενότητα 1.2.1. Αυτό που απομένει είναι να πραγματοποιηθεί η επιλογή των διανυσμάτων που θα αποτελέσουν μέλος του πληθυσμού της νέας γενιάς. Έτσι λοιπόν γίνεται η απλή σύγκριση της ποιότητας του δοκιμαστικού με του παλαιού μέλους του πληθυσμού, δηλαδή η σύγκριση των makespan:

/\* όπου makespan(P[i]) το makespan του μέλους i του πληθυσμού P και makespan(trial) το makespan του δοκιμαστικού διανύσματος του τρέχοντος μέλους i\*/

AN makespan(P[i]) > makespan(trial) TOTE

k = 1

ΟΣΟ  $k \leq D$  ΕΠΑΝΑΛΑΒΕ

P[i][k] = trial[k]

k = k + 1

ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ AN

Εικόνα 15. επιλογή μέλους νέου πληθυσμού

### 3.1.4. ΜΕΤΑΤΡΟΠΗ ΣΥΝΕΧΩΝ ΤΙΜΩΝ ΣΕ ΔΙΑΚΡΙΤΕΣ ΚΑΙ ΤΟ ΑΝΤΙΣΤΡΟΦΟ

Όπως έχει ήδη ειπωθεί ο DE χρησιμοποιεί συνεχείς τιμές ενώ το FSSP είναι πρόβλημα διακριτού προγραμματισμού. Πιο συγκεκριμένα η δημιουργία του δοκιμαστικού διανύσματος απαιτεί συνεχείς τιμές, ενώ ο υπολογισμός του makespan και οι μεταβλητές απόφασης διακριτές. Θα πρέπει δηλαδή να δημιουργηθούν τεχνικές μετατροπής συνεχών τιμών σε διακριτές και το αντίστροφο.

Μετά τη διαδικασία της αρχικοποίησης του πληθυσμού, της μετάλλαξης και της διασταύρωσης οι τιμές των δοκιμαστικών διανυσμάτων είναι συνεχείς, άρα σε αυτά τα σημεία θα πρέπει να πραγματοποιηθεί η μετατροπή των συνεχών τιμών σε διακριτές. Δεδομένου ότι οι διακριτές τιμές που χρησιμοποιούνται είναι συγκεκριμένου εύρους, η μετατροπή γίνεται ως εξής: Η μικρότερη συνεχής τιμή αντιστοιχίζεται με τη μικρότερη ακέραια, η αμέσως μεγαλύτερη συνεχής με την αμέσως μεγαλύτερη ακέραια και πάει λέγοντα μέχρις ότου η μεγαλύτερη συνεχής αντιστοιχίζεται με τη μεγαλύτερη ακέραια τιμή (Tasgetiren et al, 2004). Στον παρακάτω πίνακα παρουσιάζεται η συνάρτηση μετατροπής συνεχών σε διακριτές τιμές.

εισαγωγή διανύσματος συνεχών τιμών D στοιχείων  
ορισμός διανύσματος διακριτών τιμών D στοιχείων

```
ορισμός μίας πολύ μεγάλης τιμής BIG
i = 1
ΟΣΟ i ≤ D ΕΠΑΝΑΛΑΒΕ
    small = BIG
    j = 1
    ΟΣΟ j ≤ D ΕΠΑΝΑΛΑΒΕ
        ΑΝ το στοιχείο j του διανύσματος συνεχών τιμών < small ΤΟΤΕ
            small = στοιχείο j του διανύσματος συνεχών τιμών
            k = j
        ΤΕΛΟΣ ΑΝ
        j = j + 1
    ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
    το στοιχείο k του διανύσματος συνεχών τιμών παίρνει την τιμή BIG
    το στοιχείο k του διανύσματος διακριτών τιμών παίρνει την τιμή i
    i = i + 1
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
εξαγωγή του διανύσματος διακριτών τιμών
```

Εικόνα 16. συνάρτηση μετατροπής συνεχών σε διακριτές τιμές

Η μετατροπή των ακέραιων τιμών ενός διανύσματος σε συνεχείς, πραγματοποιείται με τον εξής απλό τρόπο: Εντοπίζεται η μεγαλύτερη τιμή των στοιχείων του διανύσματος και έπειτα κάθε στοιχείο διαιρείται με αυτήν.

```
εισαγωγή διανύσματος διακριτών τιμών
ορισμός διανύσματος συνεχών τιμών
big = 0
i = 1
ΟΣΟ i ≤ D ΕΠΑΝΑΛΑΒΕ
    ΑΝ το στοιχείο i διανύσματος διακριτών τιμών > big ΤΟΤΕ
        big = με στοιχείο i του διανύσματος διακριτών τιμών
    ΤΕΛΟΣ ΑΝ
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
i = 1
ΟΣΟ i ≤ D ΕΠΑΝΑΛΑΒΕ
```



<p>στοιχείο <math>i</math> διανύσματος συνεχών τιμών = στοιχείο <math>i</math> διανύσματος διακριτών τιμών / big</p> <p><math>i = i + 1</math></p> <p><b>ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ</b></p> <p>εξαγωγή διανύσματος συνεχών τιμών</p>
--

Εικόνα 17. συνάρτηση μετατροπής διακριτών τιμών σε συνεχείς

### 3.1.5. ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ ΚΑΙ ΣΤΡΑΤΗΓΙΚΗΣ

Ένα από τα θέματα που θα πρέπει να διερευνηθούν είναι ο προσδιορισμός των παραμέτρων του αλγορίθμου (NP, CR και F) για το συγκεκριμένο πρόβλημα. Η επιλογή των κατάλληλων τιμών για τις παραμέτρους ελέγχου του DE συνήθως εξαρτάται από το πρόβλημα (Brest, 2006), έτσι λοιπόν η επιλογή των παραμέτρων δεν είναι και τόσο απλή διαδικασία.

Το συγκεκριμένο θέμα έχει απασχολήσει αρκετούς ερευνητές, τα συμπεράσματα μερικών από των οποίων θα παρουσιαστούν στη συνέχεια. Έπειτα μέσα από τη στατιστική θα γίνει η προσπάθεια προσδιορισμού της καλύτερης στρατηγικής με συγκεκριμένες παραμέτρους αλλά και με διάφορους συνδυασμούς παραμέτρων. Τέλος, θα προσδιοριστεί ο αριθμός των επαναλήψεων του αλγορίθμου GMAX κρατώντας σταθερό το πλήθος των μελών του πληθυσμού σε  $NP = 2D$ . Ο σκοπός της συγκεκριμένης παραγράφου είναι να προσδιοριστεί ένας καλός συνδυασμός παραμέτρων για τα προβλήματα του Taillard.

Σύμφωνα με τους Price και Storn (1997) ένα λογικό μέγεθος για τον πληθυσμό (NP) κυμαίνεται μεταξύ  $NP = 5D$  και  $NP = 10D$  ενώ οι Gamperle et al (2002) εμπειρικά κατέληξαν ότι μια λογική τιμή για τον πληθυσμό είναι μεταξύ  $NP = 3D$  και  $NP = 8D$  (όπου  $D$  οι μεταβλητές της αντικειμενικής συνάρτησης).

Όσον αφορά το εύρος του δείκτη κανονικότητας (F), το  $F = 1$  είναι εμπειρικά το ανώτατο όριο διότι καμία λειτουργία που έχει βελτιστοποιηθεί με επιτυχία δεν απαίτησε  $F > 1$  (Price et al, 2005). Επίσης, στην περίπτωση που  $F = 1$  οι διαφορετικοί συνδυασμοί διανυσμάτων γίνονται δυσδιάκριτοι (Price et al, 2005):

$$x_{r0,g} + x_{r1,g} - x_{r2,g} = \begin{cases} x_{r0,g} + F(x_{r1,g} - x_{r2,g}) \\ x_{r1,g} + F(x_{r0,g} - x_{r2,g}) \end{cases} \text{ όταν } F = 1$$

και η ταχύτητα σύγκλισης μειώνεται (Gamperle et al, 2002).

Επίσης, η Zaharie (2002) απέδειξε ότι για να βρεθούν οι κατάλληλες παράμετροι ελέγχου μπορεί να χρησιμοποιηθεί η απλή σχέση:

$$2F^2 - \frac{2}{m} + \frac{p}{m} = 0, \text{ δηλαδή θα πρέπει } F > F_{\text{crit}} \text{ και όπου } F_{\text{crit}} = \sqrt{\frac{1 - \frac{CR}{2}}{NP}} \text{ (Price et al, 2005).}$$

Οι Gamperle et al (2002) κατέληξαν στο ότι η τιμή του  $F$  δε θα πρέπει να είναι μικρότερη από μία συγκεκριμένη τιμή ώστε να αποφευχθεί η πρόωρη σύγκλιση. Επίσης η τιμή του  $F$  εξαρτάται από την αντικειμενική συνάρτηση και από τις άλλες παραμέτρους της στρατηγικής. Σύμφωνα με τους ίδιους μια μεγάλη τιμή του  $F$  (με  $F < 1$ ) αυξάνει την πιθανότητα να αποφευχθεί τοπικό ακρότατο και προτείνουν ότι μια καλή αρχική τιμή για να γίνει η δοκιμή είναι  $F = 0.6$ . Αν η τιμή αυτή «πέσει» σε τοπικό ακρότατο τότε θα πρέπει να αυξηθεί. Οι Price και Storn (1997) αναφέρουν ότι μια καλή αρχική τιμή για δοκιμή είναι η  $F = 0.5$ .

Όσον αφορά τον τελεστή διασταύρωσης ( $CR$ ), όσο αυξάνεται η τιμή του αυξάνεται και η ταχύτητα σύγκλισης, αλλά μετά από κάποια συγκεκριμένη τιμή ( $CR = 0.9 \dots 1.0$ ) η ταχύτητα σύγκλισης μειώνεται ή γίνεται πρόωρη σύγκλιση (Gamperle et al., 2002). Οι Price και Storn (1997) προτείνουν ως αρχικές τιμές δοκιμής τα 0.1, 0.5, 0.7 και 0.9, ενώ οι Gamperle et al (2002) θεωρούν ότι ιδανικά η τιμή του  $CR$  θα πρέπει να κυμαίνεται μεταξύ  $CR = 0.3$  και  $CR = 0.9$ . Επίσης, οι Ali και Torn (2004), Brest (2006), Xu et al (2010) πρότειναν υπολογισμό εκ νέου των  $CR$  και  $F$  σε κάθε γενιά με διάφορες τεχνικές.

Σε αυτό το σημείο όμως γεννάται ο προβληματισμός για το ποια στρατηγική είναι η καταλληλότερη για το πρόβλημα. Στον παρακάτω πίνακα φαίνονται οι παράμετροι και οι στρατηγικές που επιλέχτηκαν από δημοσιεύσεις που χρησιμοποίησαν τον DE για την επίλυση του FSSP με κριτήριο το makespan.

Πίνακας 4. επιλογή παραμέτρων του DE για το FSSP στη βιβλιογραφία

δημοσίευση	τύπος αλγορίθμου	στρατηγική DE	F	CR	NP	GMAX
Tasgetiren et al (2004)	DE με LS	DE/best/1/bin	0.4	0.5	2D	500
Onwubolu & Davendra (2006)	DE	DE/rand/1/bin	0.3	0.9	150	50
Davendra & Onwubolu (2007)	Enhanced DE	DE/rand-to-best/1/bin	0.1	0.3	100	100
Qian et al (2008)	DE με LS	DE/rand-to-best/1/exp	0.7	0.1	2D	100

Ένας τρόπος για να δημιουργηθούν συμπεράσματα και υποθέσεις είναι μέσα από τη στατιστική διαδικασία. Ο DE λοιπόν, αρχικά εφαρμόστηκε στα προβλήματα 20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20 (Taillard, 1993) και για τις δέκα στρατηγικές των Price και Storn (1997) με σταθερές παραμέτρους: CR = 0.7, F = 0.2, NP = 2D και GMAX = 500. Για το κάθε πρόβλημα και την κάθε στρατηγική πραγματοποιήθηκαν δέκα επαναλήψεις. Στο Παράρτημα I έχουν συγκεντρωθεί τα καλύτερα αποτελέσματα και οι μέσες τιμές του makespan για το κάθε πρόβλημα.

Από τη διαδικασία αυτή προέκυψε ότι είναι πιθανόν η αποδοτικότερη στρατηγική να διαφέρει ανάλογα με το μέγεθος του προβλήματος. Δηλαδή φαίνεται πως η απόδοση της κάθε στρατηγικής (συγκριτικά με τις άλλες) εξαρτάται από τις διαστάσεις του προβλήματος και κυρίως τις εργασίες. Για παράδειγμα στα προβλήματα 20x5 φαίνεται να είναι πιο αποδοτικές οι στρατηγικές με exponential crossover, αντίθετα, για τα μεγαλύτερα προβλήματα κάποιες από τις binomial crossover στρατηγικές φαίνεται να είναι αρκετά αποδοτικές για τις συγκεκριμένες πάντα παραμέτρους.

Στον παρακάτω πίνακα παρουσιάζονται οι μέσες τιμές (μτ) του makespan της κάθε στρατηγικής για κάθε μία ομάδα μεγέθους προβλημάτων. Επίσης παρουσιάζεται ο αριθμός εμφάνισης καλύτερου αποτελέσματος (συχν) για την κάθε στρατηγική στην εκάστοτε ομάδα 10 προβλημάτων. Υπενθυμίζεται ότι τα προβλήματα εκτελέστηκαν 10 φορές το καθένα.

Πίνακας 5. σύγκριση των δέκα στρατηγικών του DE για τα προβλήματα του Taillard

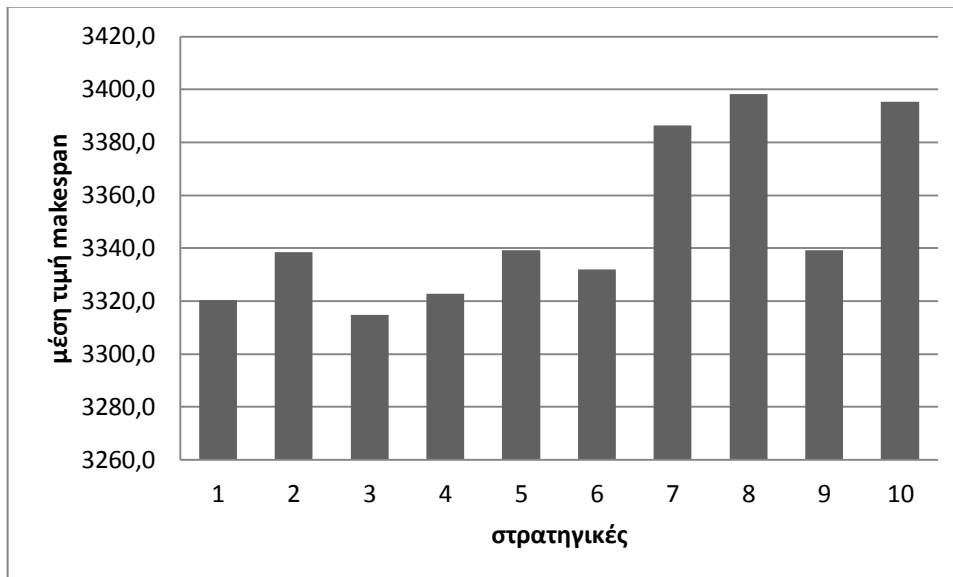
<div> <div>στρατηγική</div> <div>προβλήματα</div> </div>		1	2	3	4	5	6	7	8	9	10
20x5	μτ	<b>1263.6</b>	1269.4	1267.2	1263.7	1268.7	1300.6	1273.9	1276.9	1296.7	1279.5
	συχν	4	4	4	<b>5</b>	3	1	2	1	2	0
20x10	μτ	<b>1599.7</b>	1610.6	1610.7	1600.7	1612.2	1654.2	1618.2	1636.2	1640.2	1638.0
	συχν	1	0	2	<b>6</b>	0	0	1	0	0	0
20x20	μτ	<b>2331.3</b>	2357.0	2354.5	2337.2	2356.1	2406.0	2358.0	2384.5	2397.1	2380.1
	συχν	<b>7</b>	0	0	2	0	0	1	0	0	0
50x5	μτ	2802.7	2811.3	<b>2796.1</b>	2802.7	2811.0	2815.1	2842.5	2852.0	2807.8	2847.3
	συχν	2	0	2	1	1	1	0	0	2	0
50x10	μτ	3274.8	3297.1	3267.0	3278.6	3300.1	3275.1	3368.1	3377.7	<b>3248.3</b>	3372.0
	συχν	1	0	1	1	0	1	0	0	<b>6</b>	0
50x20	μτ	4170.9	4203.4	4147.1	4173.8	4203.6	4158.2	4288.0	4302.4	<b>4129.4</b>	4298.9
	συχν	1	0	0	0	0	0	0	0	<b>9</b>	0
100x5	μτ	5337.4	5342.9	<b>5319.7</b>	5337.5	5347.6	5328.5	5416.6	5424.2	5352.8	5420.0
	συχν	1	1	3	0	0	<b>6</b>	0	0	2	0
100x10	μτ	5782.1	5817.1	5756.1	5756.1	5815.0	<b>5718.6</b>	5925.3	5933.1	5842.4	5927.9
	συχν	0	0	0	0	0	<b>7</b>	0	0	3	0
100x20	μτ	7123.2	7168.5	7087.3	7139.6	7171.3	<b>7009.3</b>	7290.8	7303.1	7140.0	7297.3
	συχν	0	0	0	0	0	<b>6</b>	0	0	4	0

Σύμφωνα με τον παραπάνω πίνακα για τις παραμέτρους  $CR = 0.7$ ,  $F = 0.2$ ,  $NP = 2D$  και  $GMAX = 500$ :

- Οι στρατηγικές 1 και 4 παρουσιάζουν πολύ καλά αποτελέσματα για προβλήματα λίγων εργασιών ενώ για προβλήματα 50 και 100 εργασιών παρουσιάζουν μέτρια αποτελέσματα.
- Η στρατηγική 9 παρουσιάζει πολύ καλά αποτελέσματα για προβλήματα 50 εργασιών.

- Η στρατηγική 6 παρουσιάζει πολύ καλά αποτελέσματα για προβλήματα 100 εργασιών, αρκετά καλά για προβλήματα 50 εργασιών, ενώ καθόλου καλά για προβλήματα 20 εργασιών.
- Η στρατηγική 3 παρουσιάζει αρκετά καλούς μέσους χρόνους για όλα τα προβλήματα.
- Η στρατηγικές 5 και 2 παρουσιάζουν μέτρια αποτελέσματα.
- Η στρατηγική 7 δεν παρουσιάζει καλά αποτελέσματα ενώ οι στρατηγικές 8 και 10 δεν παρουσιάζουν καθόλου καλά αποτελέσματα για καμία ομάδα προβλημάτων συγκριτικά με τις υπόλοιπες στρατηγικές.

Στο παρακάτω διάγραμμα φαίνεται η συνολική απόδοση των στρατηγικών για όλα τα προβλήματα στα οποία εφαρμόστηκαν οι παράμετροι:  $F = 0.2$ ,  $CR = 0.7$ ,  $NP = 2D$  και  $GMAX = 500$ .



Εικόνα 18. Μέση τιμή makespan για τις 10 στρατηγικές των Price & Storn για παραμέτρους  $F = 0.2$ ,  $CR = 0.7$ ,  $NP = 2D$ ,  $GMAX = 500$

Σε αυτό το σημείο θα πρέπει να διερευνηθεί, γιατί άλλες μελέτες καταλήγουν σε διαφορετικά αποτελέσματα όπως αυτή των Onwubolu και Davendra (2006), σύμφωνα με την οποία η έβδομη στρατηγική προκύπτει να είναι καλύτερη μετά από δοκιμές για το FSSP σε

προβλήματα 25x10. Αυτό πιθανόν συμβαίνει για δύο κύριους λόγους: **διαφορετικά δεδομένα** και **διαφορετικές παράμετροι**.

Έτσι λοιπόν εφαρμόστηκε πάλι ο αλγόριθμος και με τις 10 στρατηγικές σε μερικά από τα προβλήματα του Taillard (20x5, 20x10, 20x20, 50x5 και 50x10) χρησιμοποιώντας αυτή τη φορά τις παραμέτρους των Onwubolu και Davendra:  $F = 0.3$ ,  $CR = 0.9$ ,  $NP = 150$ ,  $GMAX = 50$  (αναλυτικά τα αποτελέσματα παρατίθενται στο ΠΑΡΑΡΤΗΜΑ II). Με τον συγκεκριμένο τρόπο διαπιστώθηκε αν το συμπέρασμά τους αφορά και άλλα δεδομένα πέρα από αυτά που δοκίμασαν (25x10). Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα:

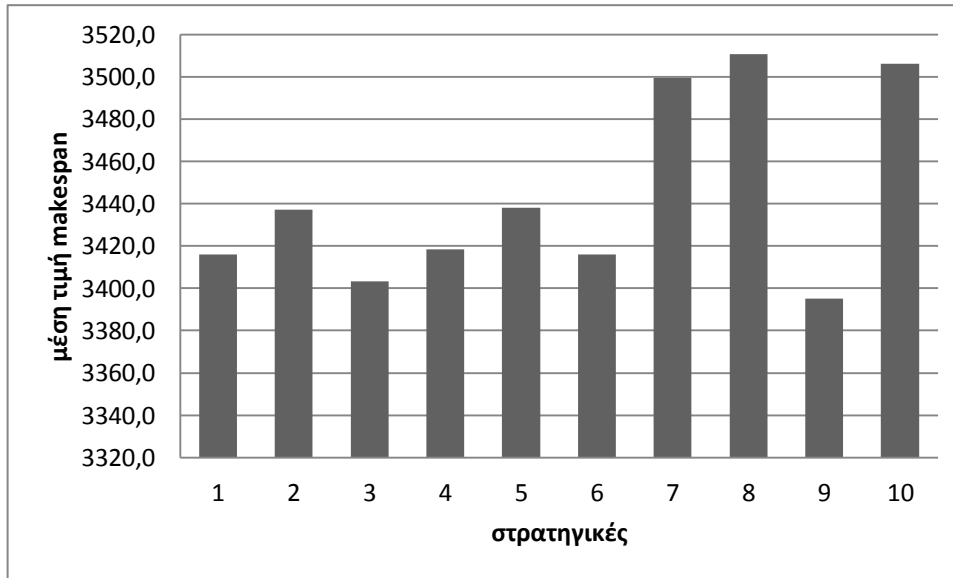
Πίνακας 6. σύγκριση των δέκα στρατηγικών του DE για προβλήματα του Taillard με παραμέτρους  $F = 0.3$ ,  $CR = 0.9$ ,  $NP = 150$ ,  $GMAX = 50$

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
20x5	min	<b>1276.8</b>	1294.5	1290.1	1283.7	1296.0	1299.6	1299.6	1300.8	1285.2	1301.0
	συχν	<b>6</b>	2	2	2	1	1	0	1	2	0
20x10	Min	<b>1617.9</b>	1655.6	1650.9	1636.3	1644.8	1647.2	1662.3	1666.5	1632.1	1670.0
	συχν	<b>6</b>	0	0	1	1	2	0	0	1	0
20x20	Min	<b>2361.1</b>	2408.8	2405.0	2388.8	2412.2	2399.8	2415.4	2421.7	2374.5	2426.2
	συχν	<b>7</b>	0	0	0	0	0	0	0	3	0
50x5	min	2847,7	2864,8	2852,7	2856,7	2862,2	<b>2834,1</b>	2882,2	2880,8	2865,0	2881,2
	συχν	1	0	0	0	0	<b>9</b>	0	0	0	0
50x10	min	3375,6	3385,2	3370,2	3382,3	3388,1	<b>3307,5</b>	3405,8	3406,3	3392,4	3410,5
	συχν	0	0	0	0	0	<b>8</b>	1	0	1	0
50x20	min	4305,4	4323,8	4301,6	4311,3	4322,8	<b>4203,0</b>	4343,4	4352,8	4330,3	4354,8
	συχν	0	0	0	0	0	<b>9</b>	0	0	1	0

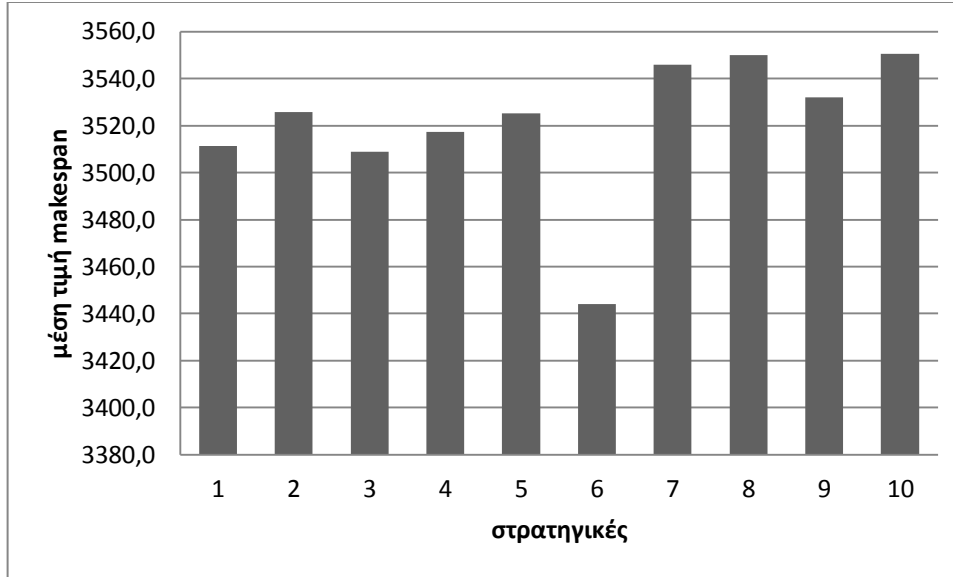
Διαπιστώνεται λοιπόν ότι για τις ίδιες παραμέτρους σε διαφορετικά μεγέθη προβλημάτων η έβδομη στρατηγική δεν είναι η πιο αποτελεσματική. Αντίθετα για τα προβλήματα 20 εργασιών πιο αποτελεσματική είναι η πρώτη και για τα προβλήματα 50 εργασιών η έκτη.

Συγκρίνοντας αυτόν τον πίνακα με τον προηγούμενο παρατηρείται ότι με την αλλαγή των παραμέτρων άλλαξε και η αποδοτικότητα των στρατηγικών για τα ίδια προβλήματα. Δηλαδή

ενώ για τις παραμέτρους  $CR = 0.7$ ,  $F = 0.2$ ,  $NP = 2D$ ,  $GMAX = 500$  πιο αποτελεσματική ήταν η ένατη στρατηγική για προβλήματα 50 εργασιών, με παραμέτρους  $F = 0.3$ ,  $CR = 0.9$ ,  $NP = 150$ ,  $GMAX = 50$  πιο αποτελεσματική ήταν η έκτη. Παρακάτω παρουσιάζονται τα διαγράμματα για την απόδοση των παραμέτρων στις δύο περιπτώσεις για τα προβλήματα 50 εργασιών:



Εικόνα 19. μέση τιμή makespan για τις 10 στρατηγικές του DE με παραμέτρους:  $F = 0.2$ ,  $CR = 0.7$ ,  $NP = 2D$ ,  $GMAX = 500$  σε προβλήματα 20 εργασιών μετά από 10 επαναλήψεις



Εικόνα 20. μέση τιμή makespan για τις 10 στρατηγικές του DE με παραμέτρους:  $F = 0.3$ ,  $CR = 0.9$ ,  $NP = 150$ ,  $GMAX = 50$  σε προβλήματα 20 εργασιών μετά από 10 επαναλήψεις

Λαμβάνοντας υπόψη τα παραπάνω και μετά από επιπλέον δοκιμές που πραγματοποιήθηκαν το συμπέρασμα είναι ότι η κάθε στρατηγική είναι πιο αποδοτική για διαφορετικές παραμέτρους. Στον πίνακα που ακολουθεί φαίνονται τα αποτελέσματα για προβλήματα διαστάσεων 20x5, 50x5 και 100x5 (Taillard, 1993) μετά από 10 δοκιμές για κάθε συνδυασμό στρατηγικής με  $CR = \{0.3, 0.5, 0.6, 0.7, 0.8, 0.9\}$  και  $F = \{0.1, 0.2, 0.3, 0.5, 0.6, 0.7\}$  δεδομένου  $NP = 20$  και  $GMAX = 200$ . Παρατηρείται ότι τα καλύτερα αποτελέσματα για τα προβλήματα 20x5 δίνονται από τον συνδυασμό της έβδομης στρατηγικής με  $F = 0.1$  και  $CR = 0.8$ , για προβλήματα 50x5 από την έκτη στρατηγική με  $F = 0.1$  και  $CR = 0.5$  και για προβλήματα 100x5 πάλι από την έκτη στρατηγική με  $F = 0.1$  και  $CR = 0.6$ .

Πίνακας 7. συνδυασμός παραμέτρων για κάθε στρατηγική σε προβλήματα 20x5

στρατηγική	1	2	3	4	5	6	7	8	9	10
CR, F	0.7, 0.1	0.9, 0.1	0.5, 0.3	0.7, 0.1	0.9, 0.1	0.3, 0.2	0.8, 0.1	0.3, 0.2	0.3, 0.1	0.9, 0.1
μ.τ. makespan	1274,8	1270,9	1275,6	1270,2	1272,7	1271,6	<b>1269,3</b>	1277,6	1271,2	1271,9



Πίνακας 8. συνδυασμός παραμέτρων για κάθε στρατηγική σε προβλήματα 50x5

στρατηγική	1	2	3	4	5	6	7	8	9	10
<b>CR, F</b>	0.9, 0.1	0.3, 0.6	0.6, 0.1	0.6, 0.1	0.3, 0.5	0.5, 0.1	0.9, 0.1	0.9, 0.2	0.5, 0.1	0.9, 0.1
<b>μ.τ. makespan</b>	2810,6	2826,0	2809,1	2815,4	2827,2	<b>2802,3</b>	2839,7	2837,0	2804,8	2850,1

Πίνακας 9. συνδυασμός παραμέτρων για κάθε στρατηγική σε προβλήματα 100x5

στρατηγική	1	2	3	4	5	6	7	8	9	10
<b>CR, F</b>	0.3, 0.1	0.3, 0.2	0.5, 0.1	0.5, 0.1	0.3, 0.3	0.6, 0.1	0.9, 0.1	0.3, 0.1	0.7, 0.1	0.3, 0.2
<b>μ.τ. makespan</b>	6057.1	6089.5	6013.2	6060.1	6092.9	<b>5949.2</b>	6188.2	6176.6	5959.3	6188.9

Συνοπτικά τα συμπεράσματα και οι υποθέσεις έτσι όπως προέκυψαν από την στατιστική διαδικασία είναι τα εξής:

- **Η κάθε στρατηγική αποδίδει καλύτερα με διαφορετικές παραμέτρους από τις υπόλοιπες στρατηγικές για συγκεκριμένα προβλήματα.**
- **Το κάθε σετ δεδομένων απαιτεί διαφορετικές παραμέτρους και στρατηγική.**
- **Πιθανότατα ο αριθμός των εργασιών επηρεάζει την αποδοτικότητα της κάθε στρατηγικής συγκριτικά με τις υπόλοιπες, άρα υπάρχει κάποια σχέση εξάρτισης: αριθμός εργασιών - στρατηγική που αποδίδει καλύτερα.**

Το πρόβλημα που τίθεται στο συγκεκριμένο σημείο είναι, πώς θα προγραμματιστεί ένα μοντέλο παραγωγής συνεχούς ροής με χρήση του DE αν πρέπει κάθε φορά που αλλάζουν τα δεδομένα να επαναπροσδιορίζονται οι παράμετροι. Το συγκεκριμένο ερώτημα θα μπορούσε να είναι αντικείμενο μελλοντικής μελέτης, δηλαδή να προσδιοριστούν μαθηματικές σχέσεις οι οποίες θα προσεγγίζουν τις καλύτερες παραμέτρους ανάλογα με τα δεδομένα του εκάστοτε προβλήματος. Μία τέτοια στατιστική διαδικασία είναι προφανώς χρονοβόρα και μη πρακτική. Στην παρούσα ενότητα, επειδή χρησιμοποιούνται τα προβλήματα του Taillard,

θα επιλεχτεί ένας συνδυασμός παραμέτρων και στρατηγικής που φαίνεται να αποδίδει σε ικανοποιητικό επίπεδο στα προβλήματα μεγέθους 50x5 και 100x5: στρατηγική 6 (DE/best/1/bin) με παραμέτρους  $CR = 0.5$  και  $F = 0.1$ .

Όσον αφορά τον αριθμό των γενεών GMAX το κάθε πρόβλημα παρουσιάζει σχετική σύγκλιση σε διαφορετικό αριθμό επαναλήψεων. Έτσι θα εισαχθεί η παράμετρος stop με την οποία ο αλγόριθμος θα ελέγχει αν έχει περάσει ένας συγκεκριμένος αριθμός γενεών χωρίς να υπάρχει βελτίωση και αν ναι τότε οι επαναλήψεις θα σταματάνε ανεξάρτητα από την τιμή του GMAX. Με αυτό τον τρόπο βελτιώνεται η ταχύτητα του αλγορίθμου και ο αριθμός των επαναλήψεων προσαρμόζεται ανάλογα με τις ανάγκες του κάθε προβλήματος. Παρακάτω παρουσιάζεται ο DE με την παράμετρο stop.

#### ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ

προσδιορισμός παραμέτρων DE: F, CR, NP, D, GMAX, στρατηγική μετάλλαξης – διασταύρωσης

προσδιορισμός παραμέτρου stop

είσοδος δεδομένων

αρχικοποίηση πληθυσμού

υπολογισμός ποιότητας του κάθε μέλους του πληθυσμού

εύρεση βέλτιστου μέλους

καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους

γενιά = 0

counter = 0

**ΟΣΟ** γενιά < GMAX **ΕΠΑΝΑΛΑΒΕ**

**ΓΙΑ** κάθε μέλος του πληθυσμού **ΕΠΑΝΑΛΑΒΕ**

        δημιουργία δοκιμαστικού διανύσματος

        υπολογισμός ποιότητας δοκιμαστικού διανύσματος

**ΑΝ** ποιότητα δοκιμαστικού διανύσματος καλύτερη από του μέλους **TOTE**

            μέλος = δοκιμαστικό διάνυσμα

            ποιότητα μέλους = ποιότητα δοκιμαστικού διανύσματος

**ΤΕΛΟΣ ΑΝ**

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

    εύρεση βέλτιστου μέλους

**ΑΝ** ποιότητα βέλτιστου μέλους = καλύτερη ποιότητα **TOTE**

        counter = counter + 1

**ΑΛΛΙΩΣ**

        καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους

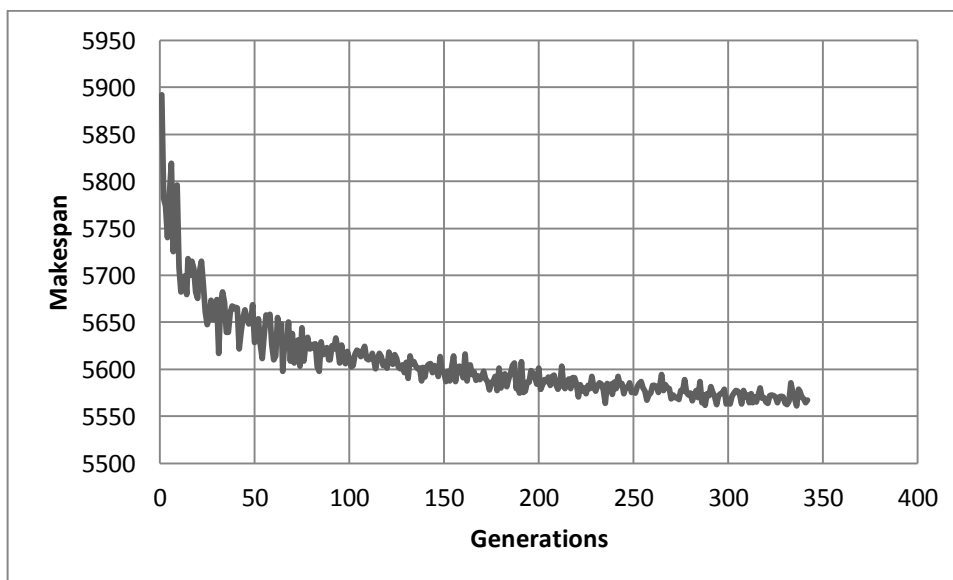
```

counter = 0

ΤΕΛΟΣ ΑΝ
ΑΝ counter < stop ΤΟΤΕ
    γενιά = γενιά + 1
ΑΛΛΙΩΣ
    γενιά = GMAX
ΤΕΛΟΣ ΑΝ
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
εκτύπωση βέλτιστου μέλους
ΤΕΡΜΑΤΙΣΜΟΣ ΑΛΓΟΡΙΘΜΟΥ
    
```

Εικόνα 21. DE με παράμετρο stop

Ένας ασφαλής αριθμός μέγιστων επαναλήψεων για όλα τα προβλήματα είναι  $GMAX = 500$  όταν  $NP = 2D$ . Στο παρακάτω διάγραμμα φαίνεται πώς βελτιώνεται η τιμή του makespan σε σχέση με τον αριθμό των επαναλήψεων για πρόβλημα 50x5 (tai31). Παρατηρείται ότι μετά τις 300 επαναλήψεις υπάρχει σχετική σύγκλιση.



Εικόνα 22. σχέση makespan με μέγιστο αριθμό επαναλήψεων στο πρόβλημα tai31

### 3.1.6. ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ

Σε όλα τα προβλήματα του Taillard εφαρμόστηκε η έκτη στρατηγική (DE/best/1/bin) με παραμέτρους  $F = 0.5$ ,  $CR = 0.1$ ,  $NP = 2D$ ,  $GMAX = 500$  σε συνδυασμό με την παράμετρο  $stop = 200$  η οποία σταματάει τις επαναλήψεις στην περίπτωση που η καλύτερη λύση δε βελτιώνεται για 200 συνεχόμενες γενιές. Η εφαρμογή πραγματοποιήθηκε κυρίως για να γίνεται αντιληπτή η βελτίωση που παρουσιάζει ο κάθε υβριδικός αλγόριθμος σχετικά με τον βασικό DE.

Στον πίνακα που ακολουθεί παρουσιάζεται η ποσοστιαία απόκλιση που έχει ο μέσος όρος των ελάχιστων τιμών, όπως προκύπτουν μετά από δέκα επαναλήψεις του DE, από τον μέσο όρο των ανώτερων ορίων του Taillard (2001). Υπολογίζεται δηλαδή η τιμή  $\Delta_{DE} = \frac{\text{μέση τιμή ελάχιστων τιμών} - \text{μέση τιμή ανώτερων ορίων}}{\text{μέση τιμή ανώτερων ορίων}} \times 100$ . Προφανώς όσο μικρότερη είναι η τιμή  $\Delta_{DE}$  τόσο το καλύτερο. Οι ελάχιστες τιμές για το κάθε πρόβλημα βρίσκονται στον συγκεντρωτικό πίνακα στο ΠΑΡΑΡΤΗΜΑ ΙΙΙ.

Πίνακας 10. απόκλιση αποτελεσμάτων της εφαρμογής του DE με τα ανώτερα όρια του Taillard

ομάδα προβλημάτων	DE
tai 20x5	3,11
tai 20x10	6,06
tai 20x20	4,69
tai 50x5	2,01
tai 50x10	5,83
tai 50x20	7,51
tai 100x5	0,68
tai 100x10	3,80
tai 100x20	7,16
tai 200x10	4,10
tai 200x20	8,52

Παρατηρείται ότι τα αποτελέσματα είναι ικανοποιητικά για τις ομάδες προβλημάτων 50x5 και 100x5 γεγονός αναμενόμενο μιας και επιλέχτηκε συνδυασμός παραμέτρων και στρατηγικής που αποδίδει αρκετά καλά για αυτές τις ομάδες προβλημάτων. Αντίθετα, χειρότερα αποτελέσματα παρουσιάζονται στις ομάδες προβλημάτων πολλών μηχανών 50x20, 100x20 και 200x20.

Στις επόμενες ενότητες του κεφαλαίου αυτού θα πραγματοποιηθεί η απόπειρα βελτίωσης των αποτελεσμάτων του Πίνακα 10 ενδυναμώνοντας τον DE με άλλες τεχνικές. Ο Πίνακας 10 θα χρησιμοποιείται ως μέτρο σύγκρισης ούτως ώστε να διαπιστώνεται ο βαθμός βελτίωσης του DE για τον εκάστοτε υβριδικό αλγόριθμο. Όλα τα αποτελέσματα των αποκλίσεων θα παρουσιαστούν συγκεντρωτικά στην ενότητα 3.3.4 όπου θα γίνει και η τελική σύγκριση των υβριδικών αλγορίθμων.

## **3.2. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ**

Ο μετασχηματισμός LS πραγματοποιείται κάθε φορά σε κάθε μέλος του πληθυσμού μετά την επιλογή της νέας γενιάς του DE. Χρησιμοποιείται επίσης η παράμετρος lsstop η οποία υποδηλώνει τον αριθμό επαναλήψεων της LS τεχνικής. Ο αλγόριθμος LS τερματίζεται όταν έχουν πραγματοποιηθεί lsstop επαναλήψεις χωρίς να βελτιώνεται η λύση. Επίσης εισάγεται ο μετρητής lscounter ο οποίος μετράει τις συνεχόμενες επαναλήψεις για τις οποίες η λύση δε βελτιώνεται. Ο DE με LS παρουσιάζεται παρακάτω:

### **ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ**

προσδιορισμός παραμέτρων DE: F, CR, NP, D, GMAX, στρατηγική μετάλλαξης – διασταύρωσης

προσδιορισμός μετασχηματισμού LS

προσδιορισμός παραμέτρων stop και lsstop

είσοδος δεδομένων

αρχικοποίηση πληθυσμού

υπολογισμός ποιότητας του κάθε μέλους του πληθυσμού

εύρεση βέλτιστου μέλους

καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους

γενιά = 0

counter = 0

**ΟΣΟ** γενιά < GMAX **ΕΠΑΝΑΛΑΒΕ**

**ΓΙΑ** κάθε μέλος του πληθυσμού **ΕΠΑΝΑΛΑΒΕ**

        δημιουργία δοκιμαστικού διανύσματος

        σύγκριση δοκιμαστικού διανύσματος με παλιό πληθυσμό

        ενημέρωση νέας γενιάς

**ΓΙΑ** κάθε μέλος της νέας γενιάς **ΕΠΑΝΑΛΑΒΕ**

        lscounter = 0

**ΟΣΟ** lscounter < lsstop **ΕΠΑΝΑΛΑΒΕ**

            δημιουργία νέας λύση με εφαρμογή μετασχηματισμού LS στο μέλος

            υπολογισμός ποιότητας νέας λύσης

**ΑΝ** νέα λύση καλύτερη από το μέλος **TOTE**

                η νέα λύση παίρνει τη θέση του μέλους

                η ποιότητα του μέλους είναι ίση με την ποιότητα της νέας λύσης

            lscounter = 0

**ΑΛΛΙΩΣ**

            lscounter = lscounter + 1

**ΤΕΛΟΣ ΑΝ**

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

εύρεση βέλτιστου μέλους

**ΑΝ** ποιότητα βέλτιστου μέλους = καλύτερη ποιότητα **TOTE**

    counter = counter + 1

**ΑΛΛΙΩΣ**

        καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους

        counter = 0

**ΤΕΛΟΣ ΑΝ**

**ΑΝ** counter < stop **TOTE**

    γενιά = γενιά + 1

**ΑΛΛΙΩΣ**

        γενιά = GMAX

**ΤΕΛΟΣ ΑΝ**

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

Εικόνα 23. DE με LS

### 3.2.1. ΣΥΓΚΡΙΣΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΩΝ ΤΟΠΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ

Για την επιλογή του μετασχηματισμού LS που θα εισαχθεί στον DE πραγματοποιήθηκε η σύγκριση των τριών μετασχηματισμών που περιγράφηκαν στην ενότητα 1.2.2.: 1-0 relocate (Εικόνα 24), 1-1 exchange (Εικόνα 25), mirror (Εικόνα 26).

```
εισαγωγή διανύσματος ls
γέννηση τυχαίων τιμών r1 και r2 με  $r2 \leq r1$ ,  $r1 \in [1, D - 1]$  και  $r2 \in [1, D]$ 
temp = ls[r1]
k = r1 + 1
ΟΣΟ k ≤ r2 ΕΠΑΝΑΛΑΒΕ
    ls[k-1] = ls[k]
    k = k + 1
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
ls[r2] = temp
εξαγωγή διανύσματος ls
```

Εικόνα 24. συνάρτηση μετασχηματισμού 1-0 relocate

```
εισαγωγή διανύσματος ls
γέννηση τυχαίων τιμών r1 και r2 με  $r2 \neq r1$ ,  $r1 \in [1, D]$  και  $r2 \in [1, D]$ 
temp = ls[r2]
ls[r2] = ls[r1]
ls[r1] = temp
εξαγωγή διανύσματος ls
```

Εικόνα 25. συνάρτηση μετασχηματισμού 1-1 exchange

```

εισαγωγή διανύσματος ls
γέννηση τυχαίων τιμών r1 και r2 με  $r2 \leq r1$ ,  $r1 \in [1, D - 1]$  και  $r2 \in [1, D]$ 
k = r1
ΟΣΟ k ≤ r2 ΕΠΑΝΑΛΑΒΕ
    temp = ls[k]
    ls[k] = ls[r2+r1-k]
    ls[r2+r1-k] = temp
    k = k + 1
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
εξαγωγή διανύσματος ls
    
```

Εικόνα 26. συνάρτηση μετασχηματισμού mirror

Οι μετασχηματισμοί LS εφαρμόστηκαν στον DE με τις εξής παραμέτρους:  $F = 0.1$ ,  $CR = 0.5$ ,  $NP = 2D$ ,  $GMAX = 500$  με  $stop = 200$  και  $lsstop = 15$  σε μικρά (20 και 50 εργασιών) και μεσαία (100 εργασιών) προβλήματα του Taillard. Τα αποτελέσματα παρουσιάζονται συγκεντρωτικά στον παρακάτω πίνακα.

Πίνακας 11. μέση τιμή makespan για τρεις διαφορετικούς μετασχηματισμούς LS σε μικρά και μεσαία προβλήματα του Taillard

αλγόριθμοι ομάδες προβλημάτων	$DE_{1-0 \text{ rel}}$	$DE_{1-1 \text{ exch}}$	$DE_{\text{mirror}}$
<b>20x5</b>	1238,26	<b>1236,74</b>	1241,14
<b>20x10</b>	1553,84	<b>1549,88</b>	1561,03
<b>20x20</b>	2284,5	<b>2280,77</b>	2297,39
<b>50x5</b>	2759,51	<b>2752,54</b>	2762,31
<b>50x10</b>	3123,85	<b>3092,48</b>	3154,69
<b>50x20</b>	3937,77	<b>3905,54</b>	4021,82
<b>100x5</b>	5272,67	<b>5264,23</b>	5279,15
<b>100x10</b>	5780,76	<b>5740,23</b>	5843,83

Από τη μελέτη του παραπάνω πίνακα φαίνεται ότι ο μετασχηματισμός 1-1 exchange είναι πιο αποδοτικός για όλες τις ομάδες προβλημάτων που δοκιμάστηκαν, άρα θα εφαρμοστεί και



στα υπόλοιπα προβλήματα του Taillard ούτως ώστε να πραγματοποιηθεί η σύγκριση του DE με τον DELS σε όλες τις ομάδες προβλημάτων.

### 3.2.2. ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ

Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα (οι αποκλίσεις του μέσου όρου των ελάχιστων τιμών με τον μέσο όρο των ανώτατων ορίων του Taillard) έτσι όπως προέκυψαν από την εφαρμογή του DELS στα προβλήματα του Taillard.

Πίνακας 12. απόκλιση των αποτελεσμάτων της εφαρμογής του DELS με τα ανώτερα όρια του Taillard

ομάδα προβλημάτων	$\Delta_{DE}$	$\Delta_{DELS}$
<b>tai 20x5</b>	3,11	0,37
<b>tai 20x10</b>	6,06	1,23
<b>tai 20x20</b>	4,69	1,23
<b>tai 50x5</b>	2,01	0,22
<b>tai 50x10</b>	5,83	1,45
<b>tai 50x20</b>	7,51	3,33
<b>tai 100x5</b>	0,68	1,09
<b>tai 100x10</b>	3,80	3,34
<b>tai 100x20</b>	7,16	0,22
<b>tai 200x10</b>	4,10	1,45
<b>tai 200x20</b>	8,52	3,33

Παρατηρείται ότι ο DE βελτιώνεται πάρα πολύ με την εισαγωγή της LS τεχνικής. Καθώς μερικές φορές αγγίζει και το ανώτερο όριο του Taillard στα προβλήματα 5 μηχανών. Ενδεικτικά αναφέρονται τα παρακάτω αποτελέσματα της εφαρμογής του DELS:

Πίνακας 13. καλά αποτελέσματα 5 μηχανών

πρόβλημα	ανώτερο όριο Taillard (2001)	αποτέλεσμα εφαρμογής DELS	
		makespan	σειρά εργασιών
tai01	1278	1278	9 15 6 17 4 8 19 5 13 7 11 1 3 2 14 18 16 10 20 12
		1278	17 3 15 8 1 11 13 19 14 4 2 6 5 9 7 18 16 10 20 12
tai02	1359	1360	6 19 10 7 15 3 12 18 11 17 9 5 13 16 20 2 1 4 8 14
tai06	1195	1195	11 14 20 13 16 17 7 12 6 18 1 9 19 4 15 10 8 2 5 3
tai08	1206	1206	17 2 16 9 14 5 1 12 10 3 18 19 13 7 8 20 4 15 6 11
		1206	17 2 16 12 6 9 5 19 8 20 4 7 15 14 13 10 1 3 18 11
tai10	1108	1109	11 7 1 16 6 19 3 12 18 13 15 10 2 8 5 14 20 17 4 9
tai31	2724	2724	31 50 41 8 30 5 10 40 42 6 3 32 23 24 47 1 13 38 18 25 27 4 11 45 22 43 28 20 15 2 7 34 49 17 14 12 39 21 46 33 26 44 35 29 9 19 16 48 37 36
		2724	31 41 39 21 40 45 3 44 25 32 27 50 46 29 4 26 13 9 28 34 47 7 37 10 22 11 14 8 2 17 23 49 1 24 6 18 15 38 42 48 43 33 12 30 5 35 16 20 19 36
tai35	2863	2864	46 48 30 1 40 34 28 20 7 3 21 2 37 31 14 19 5 13 36 35 23 44 29 4 50 33 15 16 24 26 43 17 12 11 8 32 38 9 22 10 18 39 27 6 41 49 47 45 25 42
		2864	46 48 7 38 36 27 14 40 23 16 1 17 12 34 47 8 20 26 5 3 21 4 24 32 19 9 35 44 10 30 31 13 43 45 15 29 2 28 39 50 22 37 41 11 25 6 49 33 18 42
tai40	2782	2782	6 34 48 8 11 12 10 3 49 15 47 40 14 43 38 18 41 39 17 50 44 4 46 19 30 22 36 5 31 33 35 2 24 20 16 23 21 42 28 9 7 13 37 29 45 27 25 1 26 32
		2782	6 50 8 33 11 44 17 24 2 40 28 38 35 5 47 13 49 12 43 15 3 14 19 1 16 23 27 9 36 48 18 34 21 10 45 41 4 22 39 30 7 31 46 42 29 37 20 25 26 32
tai61	5493	5493	74 14 60 49 90 71 26 72 41 31 48 63 19 13 27 40 66 64 29 80 94 11 93 18 86 23 35 10 30 57 24 3 50 77 70 54 16 22 89 45 98 96 33 34 83 42 100 44 62 61 12 39 28 68 67 21 85 73 58 25 95 79 87 4 75 38 6 17 51 65 46 36 99 43 9 47 20 81 55 15 92 2 56 37 53 88 52 5 78 7 69 76 91 97 8 59 32 82 1 84
tai66	5137	5137	83 20 73 7 88 93 6 98 3 81 70 5 58 1 71 53 94 22 45 47 60 37 86 17 90 96 41 50 61 13 91 19 55 21 33 95 44 82 62 2 72 97 65 11

			87 56 15 14 18 92 63 38 49 34 80 68 64 28 48 30 76 59 46 78 52
			10 23 57 35 24 74 69 43 25 12 99 85 9 16 31 89 40 36 27 67 77
			75 66 84 29 42 26 100 79 54 32 8 39 51 4

### 3.2.3 ΠΕΡΑΙΤΕΡΩ ΔΙΕΡΕΥΝΗΣΗ

Παρατηρήθηκε επίσης ότι ακόμα και με μία μόνο επανάληψη της εφαρμογής μετασχηματισμού LS το αποτέλεσμα βελτιώνεται σε ικανοποιητικό βαθμό. Σε αυτή την περίπτωση η σύγκλιση πραγματοποιείται σε μεγαλύτερο βαθμό γενεών ώστε να προσεγγίζονται οι επαναλήψεις του αλγορίθμου που περιγράφηκε παραπάνω. Ενδεικτικά στον παρακάτω πίνακα παρουσιάζεται το ελάχιστο makespan (min) και η μέση τιμή (μ. τ.) 10 επαναλήψεων για τους DE, DE<sub>rel</sub>, DE<sub>rel</sub> με μονό μετασχηματισμό LS (DE<sub>rel</sub> με SLS) στα προβλήματα 20 εργασιών. Φαίνεται λοιπόν ότι ο DE βελτιώνεται αρκετά ακόμα και με μία μόνο φορά εφαρμογής μετασχηματισμού LS. Στον Πίνακα 12. τα αποτελέσματα του DE<sub>rel</sub> με SLS που παρατίθενται προέκυψαν με GMAX = 300 χωρίς βέβαια να σημαίνει ότι αυτός είναι ο αριθμός σύγκλισης και χωρίς να ληφθούν υπόψη οι παράμετροι stop και lsstop.

Πίνακας 14. βελτίωση DE με εφαρμογή LS τεχνικής

αλγόριθμος πρόβλημα	DE		DE με LS		DE με SLS	
	min	μ. τ.	min	μ. τ.	min	μ. τ.
tai01	1297	1302.9	1278	1294.1	1288	1294
tai02	1368	1372	1361	1366.4	1366	1366.8
tai03	1132	1162.4	1098	1103.5	1124	1127.9
tai04	1360	1376	1301	1315.8	1302	1318.2
tai05	1277	1279.5	1244	1249.9	1239	1252.5
tai06	1233	1261.4	1207	1212	1210	1215.5
tai07	1251	1263.3	1249	1254.7	1256	1258.1
tai08	1260	1273.6	1222	1233.3	1226	1227.6
tai09	1273	1290.8	1255	1258.2	1271	1271
tai10	1136	1155.5	1127	1125.4	1127	1127.8
tai11	1673	1687.9	1620	1633.7	1629	1643
tai12	1746	1768.7	1692	1716.3	1712	1724

<b>tai13</b>	1579	1611.4	1523	1553.2	1562	1568
<b>tai14</b>	1456	1485.5	1396	1411.8	1391	1429
<b>tai15</b>	1506	1546.5	1459	1480.9	1467	1488
<b>tai16</b>	1479	1493	1425	1437.2	1430	1442
<b>tai17</b>	1548	1572.4	1509	1517.9	1515	1533
<b>tai18</b>	1631	1658.2	1564	1593.9	1556	1602
<b>tai19</b>	1663	1676.6	1630	1639.2	1625	1646
<b>tai20</b>	1683	1695.1	1616	1642.3	1643	1655
<b>tai21</b>	2401	2429.7	2328	2344.9	2339	2358.7
<b>tai22</b>	2180	2223.2	2132	2149.9	2128	2156.1
<b>tai23</b>	2413	2449.4	2375	2395.1	2385	2403
<b>tai24</b>	2326	2359.7	2240	2272.5	2265	2283.7
<b>tai25</b>	2324	2419.3	2299	2344	2337	2357.7
<b>tai26</b>	2358	2344.2	2257	2278.7	2277	2294.8
<b>tai27</b>	2314	2393.9	2322	2335.8	2338	2349.8
<b>tai28</b>	2331	2331.4	2237	2246.6	2242	2258.5
<b>tai29</b>	2331	2373	2291	2313.3	2301	2324.8
<b>tai30</b>	2299	2336.9	2209	2245.2	2251	2268.2

### 3.3. ΥΒΡΙΔΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ

Τόσο από τη βιβλιογραφική ανασκόπηση όσο και από την εφαρμογή των προβλημάτων του Taillard πρόέκυψε ότι ο DEA παρουσιάζει βελτίωση με απλές τεχνικές LS ενώ από μόνος του δεν είναι τόσο ισχυρός συγκριτικά με άλλους αλγορίθμους για το FSSP. Στην ενότητα αυτή θα διερευνηθεί αν η αποδοτικότητα του DE μπορεί να βελτιωθεί ακόμα πιο πολύ στην περίπτωση που συνδυαστεί με μεθευρετικούς αλγορίθμους βασισμένους στην τοπική αναζήτηση. Για αυτόν τον λόγο στη συνέχεια θα αναπτυχθούν τρεις διαφορετικοί υβριδικοί αλγόριθμοι βασισμένοι σε αυτόν της διαφορικής εξέλιξης (Hybrid Differential Evolution Algorithm – HDE).

### 3.3.1. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ

Ο ΗΔΕ με στοιχεία του αλγορίθμου SA (DESA) εφαρμόστηκε με διάφορες παραλλαγές, μερικές φορές βελτιώνοντας τον DE χωρίς όμως να αποδίδει καλύτερα αποτελέσματα από τον DELS. Ουσιαστικά αυτό που δοκιμάζεται στη συγκεκριμένη παράγραφο είναι κατά πόσο το **κριτήριο της προσομοιωμένης ανόπτησης**, το οποίο δε δέχεται μόνο «καλές» λύσεις, βελτιώνει την απόδοση του DE στο FSSP. Για αυτό το λόγο παρουσιάζονται δύο διαφορετικοί αλγόριθμοι DESA. Ο ένας εισάγει το κριτήριο της προσομοιωμένης ανόπτησης στη φάση της επιλογής νέου μέλους πληθυσμού του DE (DESA<sub>1</sub> - Εικόνα 27), ενώ ο άλλος στη φάση της επιλογής του αλγορίθμου LS (DESA<sub>2</sub> - Εικόνα 28).

Στον DESA<sub>1</sub> εισάγεται επίσης το διάνυσμα bestsofar, στο οποίο αποθηκεύεται η καλύτερη λύση που έχει βρεθεί στον αλγόριθμο για να μη χαθεί μελλοντικά και είναι αυτή που επιστρέφεται στο τέλος του αλγορίθμου. Για τον ίδιο λόγο στον DESA<sub>2</sub> εισάγεται το διάνυσμα sa στο οποίο αποθηκεύεται η λύση που προκύπτει από τη διαδικασία της προσομοιωμένης ανόπτησης και μετά εξετάζεται αν είναι καλύτερη από το τρέχον μέλος του πληθυσμού για να το αντικαταστήσει.

#### ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ

προσδιορισμός παραμέτρων DE: F, CR, NP, D, GMAX, στρατηγική μετάλλαξης – διασταύρωσης

προσδιορισμός ελάχιστης θερμοκρασίας

προσδιορισμός προγράμματος μείωσης θερμοκρασίας

είσοδος δεδομένων

αρχικοποίηση πληθυσμού

υπολογισμός ποιότητας του κάθε μέλους του πληθυσμού

εύρεση βέλτιστου μέλους

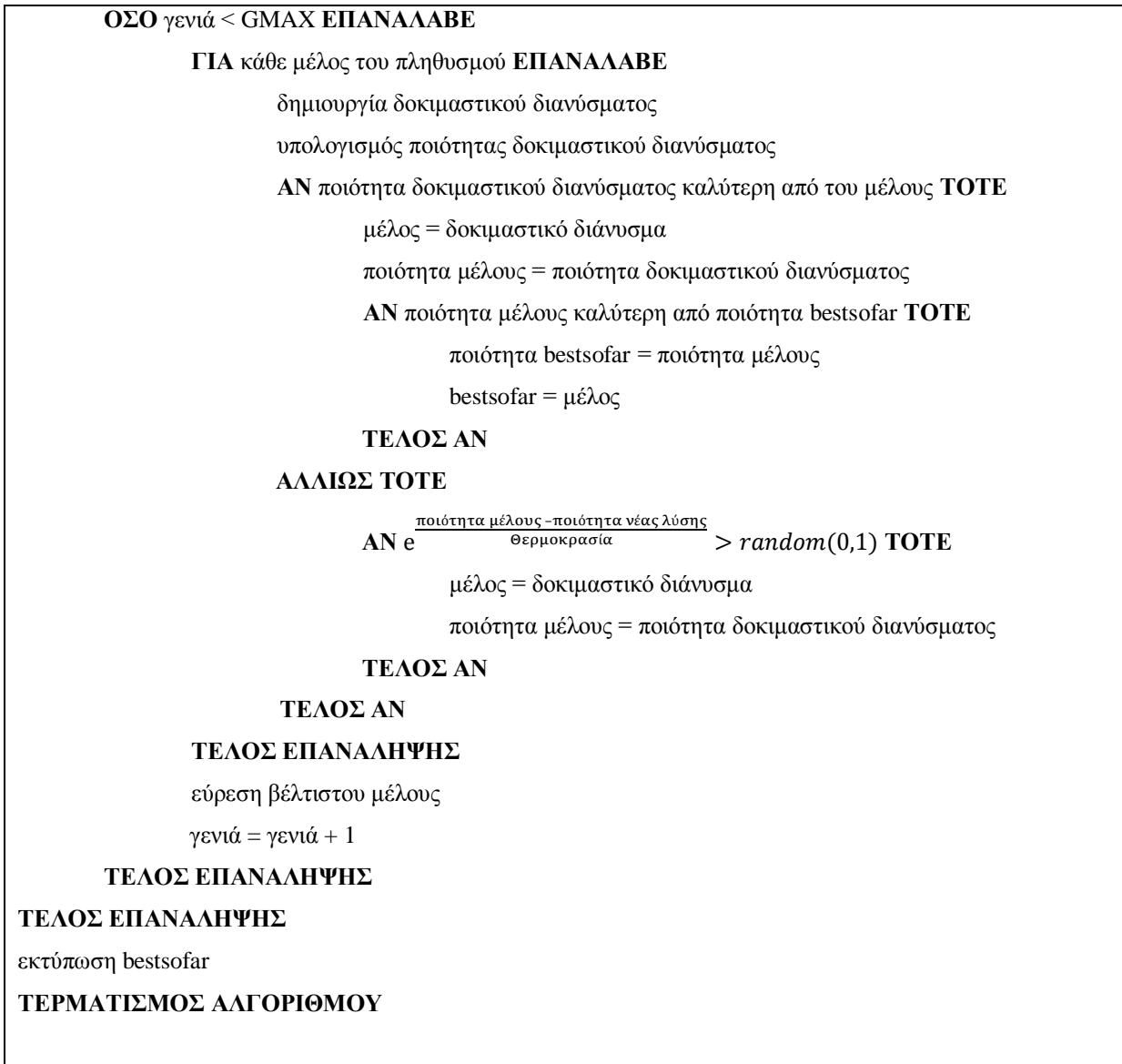
bestsofar = βέλτιστο μέλος

ποιότητα bestsofar = ποιότητα βέλτιστου μέλους

γενιά = 0

αρχικοποίηση θερμοκρασίας

**ΟΣΟ** θερμοκρασία > ελάχιστη θερμοκρασία **ΕΠΑΝΑΛΑΒΕ**



Εικόνα 27. υβριδικός αλγόριθμος διαφορικής εξέλιξης με κριτήριο προσομοιωμένης ανόπτησης στην επιλογή του νέου μέλους του πληθυσμού

#### **ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ**

προσδιορισμός παραμέτρων DE: F, CR, NP, D, GMAX, στρατηγική μετάλλαξης – διασταύρωσης

προσδιορισμός μετασχηματισμού LS

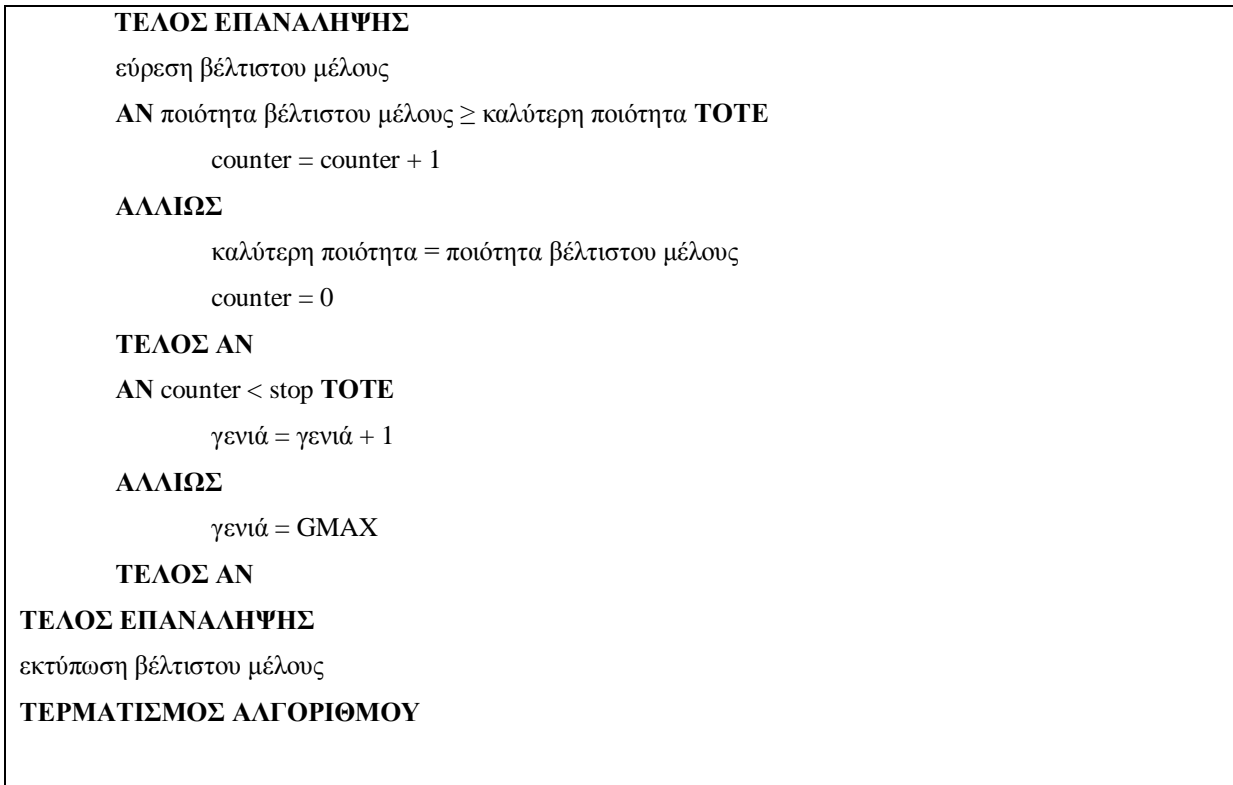
προσδιορισμός παραμέτρων stop και lsstop

προσδιορισμός ελάχιστης θερμοκρασίας

προσδιορισμός προγράμματος μείωσης θερμοκρασίας

είσοδος δεδομένων

```
αρχικοποίηση πληθυσμού
υπολογισμός ποιότητας του κάθε μέλους του πληθυσμού
εύρεση βέλτιστου μέλους
καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους
bestsofar = βέλτιστο μέλος
ποιότητα bestsofar = ποιότητα βέλτιστου μέλους
γενιά=0
counter = 0
ΟΣΟ γενιά < GMAX ΕΠΑΝΑΛΑΒΕ
    ΓΙΑ κάθε μέλος του πληθυσμού ΕΠΑΝΑΛΑΒΕ
        δημιουργία δοκιμαστικού διανύσματος
        σύγκριση δοκιμαστικού διανύσματος με παλιό πληθυσμό
        ενημέρωση νέας γενιάς και διανύσματος bestsofar
        Iscounter = 0
        αρχικοποίηση θερμοκρασίας
        διάνυσμα sa = τρέχον μέλος του πληθυσμού
        ΟΣΟ θερμοκρασία > ελάχιστη θερμοκρασία ΕΠΑΝΑΛΑΒΕ
            ΟΣΟ Iscounter < Isstop ΕΠΑΝΑΛΑΒΕ
                δημιουργία νέας λύσης με εφαρμογή LS τεχνικής στο διάνυσμα sa
                υπολογισμός ποιότητας νέας λύσης
                ΑΝ ποιότητα νέας λύσης καλύτερη από διανύσματος sa TOTE
                    διάνυσμα sa = νέα λύση
                    Iscounter = 0
                ΑΛΛΙΩΣ TOTE
                    Iscounter = counter+1
                ΑΝ  $e^{\frac{\text{ποιότητα μέλους} - \text{ποιότητα νέας λύσης}}{\text{θερμοκρασία}}} > \text{random}(0,1)$  TOTE
                    διάνυσμα sa = νέα λύση
                ΤΕΛΟΣ ΑΝ
            ΤΕΛΟΣ ΑΝ
        ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
        επαναυπολογισμός θερμοκρασίας
    ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ ποιότητα διανύσματος sa καλύτερη από ποιότητα μέλους TOTE
        ποιότητα μέλους = ποιότητα διανύσματος sa
        μέλος = διάνυσμα sa
    ΤΕΛΟΣ ΑΝ
```



Εικόνα 28. υβριδικός αλγόριθμος διαφορικής εξέλιξης με τοπική αναζήτηση και κριτήριο προσομοιωμένης ανόπτησης στην επιλογή του νέου μέλους του πληθυσμού της τοπικής αναζήτησης

Ο  $DESA_1$  ουσιαστικά είναι ο αλγόριθμος της προσομοιωμένης ανόπτησης αλλά αντί για απλό μετασχηματισμό τοπικής αναζήτησης εφαρμόζεται ο αλγόριθμος της διαφορικής εξέλιξης σε GMAX επαναλήψεις για κάθε τιμή της θερμοκρασίας. Από μια άλλη σκοπιά θα μπορούσε να προσδιοριστεί ως DE με το κριτήριο της προσομοιωμένης ανόπτησης.

Παρατηρήθηκε όμως ότι ο  $DESA_1$  είναι ακατάλληλος για το συγκεκριμένο πρόβλημα. Αυτό που συμβαίνει είναι ότι με το να γίνονται αποδεχτές μη καλές λύσεις χάνεται η ικανότητα του DE να αναβαθμίζει τη συνολική ποιότητα του νέου πληθυσμού, άρα χάνει το νόημά του το γεγονός ότι ο DE είναι αλγόριθμος βασισμένος στον πληθυσμό. Δηλαδή πρέπει να μικρύνει η τιμή του GMAX αλλά και αυτή της αρχικής θερμοκρασίας ούτως ώστε να περιοριστεί ο αριθμός των «κακών» λύσεων που περνούν στην επόμενη γενιά, όμως με αυτόν τον τρόπο χάνει το νόημα του και ο SA. Επομένως ουσιαστικά ο  $DESA_1$  δεν έχει νόημα εφαρμογής στο FSSP.



Ο DESA<sub>2</sub> εφαρμόστηκε με τις εξής παραμέτρους:  $F = 0.1$ ,  $CR = 0.5$ ,  $NP = 2D$ ,  $GMAX = 500$ ,  $stop = 200$  και  $lsstop = 1$ . Όσον αφορά το πρόγραμμα μείωσης της θερμοκρασίας επιλέχτηκε το  $T = \alpha T$  με  $T_{αρχική} = 100$  και  $T_{min} = 0.1$ . Η επιλογή των παραμέτρων της θερμοκρασίας έγιναν με το σκεπτικό να προσεγγίζεται ο αριθμός των επαναλήψεων του μετασχηματισμού LS που εφαρμόστηκαν στον DELS έτσι ώστε να γίνει αντικειμενικότερα η σύγκριση των αλγορίθμων με παρόμοιο υπολογιστικό κόστος.

### 3.3.2. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΕΠΑΝΑΛΗΠΤΙΚΗ ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ

Ο DE συχνά εγκλωβίζεται σε τοπικό ελάχιστο χωρίς να μπορεί να βελτιώσει την καλύτερη ποιοτικά λύση περισσότερο. Στον HDE με ILS (DEILS) όταν ο DE δε βελτιώνει τη λύση για  $startils$  συνεχόμενες γενιές τότε εφαρμόζεται ο ILS σε όλα τα μέλη της τρέχουσας γενιάς πληθυσμού με σκοπό να απεγκλωβιστεί η βέλτιστη λύση. Πριν εφαρμοστεί η διαταραχή του ILS το διάνυσμα του μέλους αποθηκεύεται και αντικαθιστάται με τη λύση που προκύπτει από τον ILS μόνο στην περίπτωση που είναι καλύτερη.

#### ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ

προσδιορισμός παραμέτρων DE:  $F$ ,  $CR$ ,  $NP$ ,  $D$ ,  $GMAX$ , στρατηγική μετάλλαξης – διασταύρωσης

προσδιορισμός διαταραχής

προσδιορισμός κριτηρίου επανάληψης

προσδιορισμός μετασχηματισμού LS πριν και μετά τη διαταραχή

προσδιορισμός παραμέτρων  $startils$ ,  $lsstop1$  και  $lsstop2$

είσοδος δεδομένων

αρχικοποίηση πληθυσμού

υπολογισμός ποιότητας του κάθε μέλους του πληθυσμού

εύρεση βέλτιστου μέλους

γενιά=0

counter = 0

**ΟΣΟ** γενιά <  $GMAX$  **ΕΠΑΝΑΛΑΒΕ**

παλιά καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους

**ΓΙΑ** κάθε μέλος του πληθυσμού **ΕΠΑΝΑΛΑΒΕ**

δημιουργία δοκιμαστικού διανύσματος  
σύγκριση δοκιμαστικού διανύσματος με παλιό πληθυσμό  
ενημέρωση νέας γενιάς

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

εύρεση βέλτιστου μέλους

**ΑΝ** ποιότητα βέλτιστου μέλους = καλύτερη ποιότητα **TOTE**

counter = counter + 1

**ΑΛΛΙΩΣ**

καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους

counter = 0

**ΤΕΛΟΣ ΑΝ**

**ΑΝ** counter > startils **TOTE** /\*έναρξη ILS\*/

**ΓΙΑ** κάθε μέλος της νέας γενιάς **ΕΠΑΝΑΛΑΒΕ**

lscounter = 0

**ΟΣΟ** lscounter < lsstop1 **ΕΠΑΝΑΛΑΒΕ**

δημιουργία νέας λύσης με εφαρμογή μετασχηματισμού LS στο μέλος

υπολογισμός ποιότητας νέας λύσης

**ΑΝ** νέα λύση καλύτερη από το μέλος **TOTE**

η νέα λύση παίρνει τη θέση του μέλους

η ποιότητα του μέλους είναι ίση με την ποιότητα της νέας λύσης

lscounter = 0

**ΑΛΛΙΩΣ**

lscounter = lscounter + 1

**ΤΕΛΟΣ ΑΝ**

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ** ilsit φορές **ΕΠΑΝΑΛΑΒΕ**

διάνυσμα ils = νέα λύση

ποιότητα διανύσματος ils = ποιότητα νέας λύσης

διαταραχή στο διάνυσμα ils

lscounter = 0

**ΟΣΟ** lscounter < lsstop2 **ΕΠΑΝΑΛΑΒΕ**

δημιουργία νέας λύσης με εφαρμογή μετασχ. LS στο διάνυσμα ils

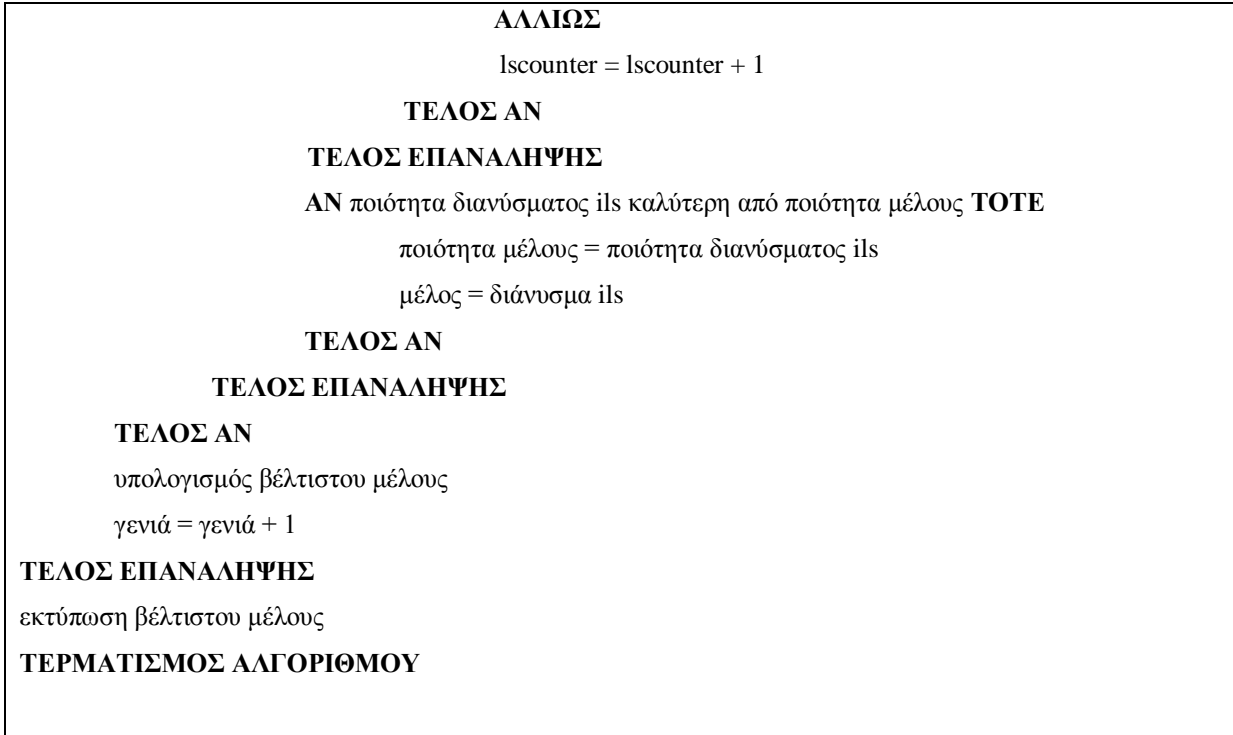
υπολογισμός ποιότητας νέας λύσης

**ΑΝ** ποιότητα νέας λύσης καλύτερη από διανύσματος ils **TOTE**

η νέα λύση παίρνει τη θέση του διανύσματος ils

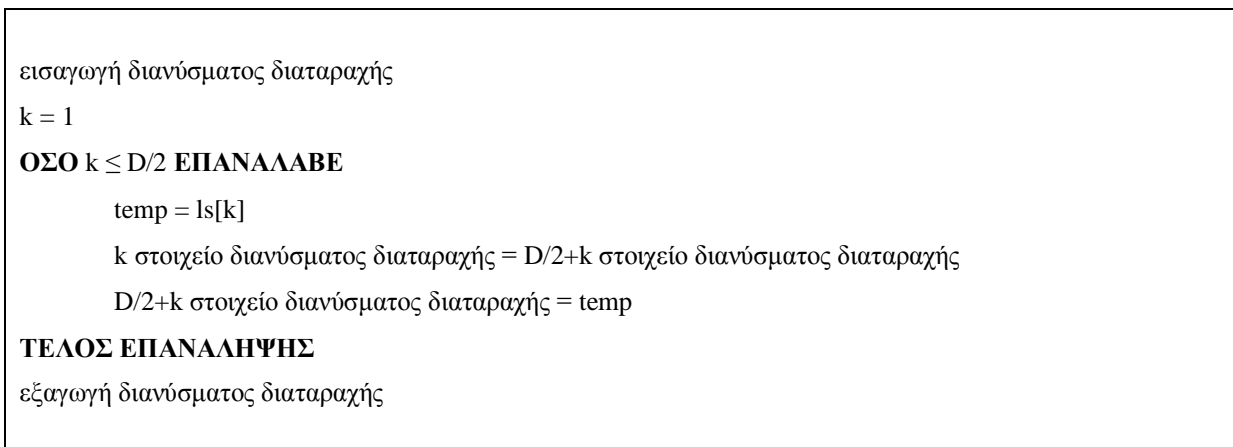
ποιότητα διανύσματος ils ίση με ποιότητα νέας λύσης

lscounter = 0



Εικόνα 29. DEILS

Η διαταραχή προκαλείται σε μία λύση χωρίζοντάς τη σε δύο μέρη και μεταφέροντας το ένα μέρος στη θέση του άλλου. Για παράδειγμα το διάνυσμα: 10 8 5 3 6 1 2 4 9 7 μετά τη διαταραχή θα μετατραπεί σε: 1 2 4 9 7 10 8 5 3 6.



Εικόνα 30. διαταραχή στον ISL

Η τεχνική LS που χρησιμοποιείται από τον ILS πριν τη διαταραχή και μετά είναι η 1-1 exchange όπως περιγράφηκε στις προηγούμενες ενότητες με τη διαφορά ότι πριν τη

διαταραχή πρέπει να πραγματοποιηθούν  $lsstop1 = 15$  συνεχόμενες επαναλήψεις χωρίς βελτίωση ενώ μετά  $lsstop = 20$ , ούτως ώστε να είναι σίγουρο ότι μετά τη διαταραχή η ποιότητα του νέου διανύσματος θα έχει φτάσει σε ένα ικανοποιητικό επίπεδο. Επίσης, το κριτήριο αποδοχής της λύσης που χρησιμοποιείται από τον ILS είναι το κριτήριο της απληστίας, ενώ η διαταραχή με την τοπική αναζήτηση που ακολουθεί επαναλαμβάνεται 3 φορές ( $ILSIT = 3$ ) για κάθε φορά που καλείται ο αλγόριθμος ILS μέσα από τον DE.

Χρησιμοποιείται και εδώ η έκτη στρατηγική των Price και Storn ενώ οι παράμετροι του DE είναι και αυτή τη φορά οι:  $F = 0.1$ ,  $CR = 0.5$ ,  $NP = 2D$  με τη μόνη διαφορά στην τιμή του GMAX. Πραγματοποιούνται  $GMAX = 200$  επαναλήψεις του αλγορίθμου χωρίς να λαμβάνεται υπόψη η παράμετρος stop και στην περίπτωση που για  $startils = 10$  συνεχόμενες γενιές η ποιότητα του βέλτιστου μέλους του αλγορίθμου δε βελτιώνεται, τότε πραγματοποιείται ο αλγόριθμος ILS.

### **3.3.3. ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΦΟΡΙΚΗΣ ΕΞΕΛΙΞΗΣ ΜΕ ΑΛΛΑΓΗ ΓΕΙΤΟΝΙΑΣ ΑΝΑΖΗΤΗΣΗΣ**

Ο VNS θα ενταχτεί στον DE (DEVNS) αφού πρώτα αυτός εγκλωβιστεί σε κάποιο τοπικό ελάχιστο όπως και στην περίπτωση του DEISL. Αυτό διαπιστώνεται με την παράμετρο  $startvns$  με τον ίδιο τρόπο που εφαρμόζεται η  $startils$  στον DEILS. Μετά τον εγκλωβισμό αρχίζει η εφαρμογή του VNS σε όλα τα μέλη της τρέχουσας γενιάς.

Στη συγκεκριμένη περίπτωση στα πλαίσια του VNS εφαρμόζονται τρεις διαφορετικές τεχνικές LS. Αρχικά πραγματοποιούνται οι επαναλήψεις της πρώτης LS τεχνικής και όταν πραγματοποιηθούν  $lsstop$  συνεχόμενες επαναλήψεις χωρίς βελτίωση της λύσης, ο αλγόριθμος συνεχίζει με τις επαναλήψεις της επόμενης τεχνικής και έπειτα προχωράει κυκλικά μέχρι να ικανοποιηθεί το κριτήριο σύγκλισης.

Ουσιαστικά ο αλγόριθμος τερματίζεται όταν δεν έχει πραγματοποιηθεί καμία βελτίωση της λύσης περνώντας και από τις τρεις τεχνικές LS. Στον DEVNS εισάγεται ο μετρητής

vnscounter ο οποίος χρησιμοποιείται για να ελέγχει αν έχει γίνει κάποια βελτίωση κατά τη διάρκεια των επαναλήψεων της εκάστοτε LS τεχνικής. Με αυτόν τον τρόπο διαπιστώνεται αν έχουν πραγματοποιηθεί και οι τρεις επιμέρους αλγόριθμοι LS χωρίς να βελτιωθεί η λύση. Αν συμβεί αυτό δεν πραγματοποιείται άλλη επανάληψη του VNS και ο αλγόριθμος προχωράει στο επόμενο μέλος του πληθυσμού. Επίσης επειδή ο DEVNS εφαρμόζεται σε κάθε μέλος του πληθυσμού δεν πραγματοποιείται κάποια επιπλέον διαταραχή προτού ξεκινήσει η πρώτη τεχνική LS.

#### ΕΝΑΡΞΗ ΑΛΓΟΡΙΘΜΟΥ

προσδιορισμός παραμέτρων DE: F, CR, NP, D, GMAX, στρατηγική μετάλλαξης – διασταύρωσης

προσδιορισμός παραμέτρου startvns

προσδιορισμός αλγορίθμου LS

προσδιορισμός μετασχηματισμών LS1, LS2 και LS3

είσοδος δεδομένων

αρχικοποίηση πληθυσμού

υπολογισμός ποιότητας του κάθε μέλους του πληθυσμού

εύρεση βέλτιστου μέλους

γενιά = 0

counter = 0

**ΟΣΟ** γενιά < GMAX **ΕΠΑΝΑΛΑΒΕ**

καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους

**ΓΙΑ** κάθε μέλος του πληθυσμού **ΕΠΑΝΑΛΑΒΕ**

δημιουργία δοκιμαστικού διανύσματος

σύγκριση δοκιμαστικού διανύσματος με παλιό πληθυσμό

ενημέρωση νέας γενιάς

**ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ**

εύρεση βέλτιστου μέλους

**ΑΝ** ποιότητα βέλτιστου μέλους = καλύτερη ποιότητα **TOTE**

counter = counter+1

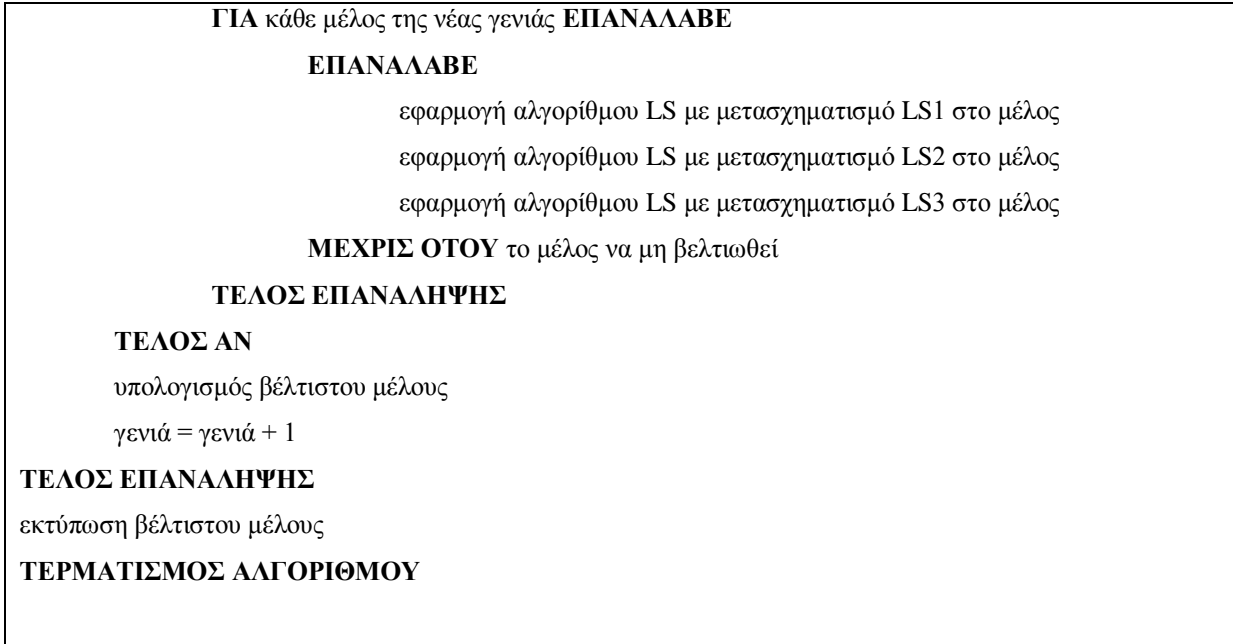
**ΑΛΛΙΩΣ**

καλύτερη ποιότητα = ποιότητα βέλτιστου μέλους

counter = 0

**ΤΕΛΟΣ ΑΝ**

**ΑΝ** counter > startvns **TOTE** /\*έναρξη VNS\*/



Εικόνα 31. DEVNS

Οι LS τεχνικές που χρησιμοποιήθηκαν είναι οι mirror, 1-0 relocate και 1-1 exchange στην αντίστοιχη σειρά με lsstop = 20. Ο VNS ξεκινάει όταν το βέλτιστο μέλος που προκύπτει από τον DE δεν έχει αλλάξει για 5 συνεχόμενες γενιές, δηλαδή όταν startvns = 5. Οι παράμετροι της DE παραμένουν οι ίδιες με GMAX = 200 όπως στην περίπτωση του DEILS.

### 3.3.4. ΣΥΓΚΡΙΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΩΝ

Παρακάτω παρουσιάζονται οι αποκλίσεις των καλύτερων αποτελεσμάτων (μετά από 10 επαναλήψεις για το κάθε πρόβλημα) από το ανώτερα όρια του Taillard (2001) έτσι όπως προέκυψαν από την εφαρμογή των αλγορίθμων DE, DELS, DESA, DEILS και DEVNS (αναλυτικότερα τα αποτελέσματα για κάθε πρόβλημα ξεχωριστά παρατίθενται στο ΠΑΡΑΡΤΗΜΑ ΙΙΙ).

Πίνακας 15. σύγκριση απόκλισης αποτελεσμάτων των DE, DELS, DESA με τα ανώτερα όρια του Taillard

ομάδα προβλημάτων	$\Delta_{DE}$	$\Delta_{DELS}$	$\Delta_{DESA2}$	$\Delta_{DEILS}$	$\Delta_{DEVNS}$
<b>tai 20x5</b>	3,11	0,37	0,51	0,40	0,38
<b>tai 20x10</b>	6,06	1,23	1,70	1,41	0,97

<b>tai 20x20</b>	4,69	1,23	1,71	1,17	0,88
<b>tai 50x5</b>	2,01	0,26	0,92	0,27	0,23
<b>tai 50x10</b>	5,83	1,88	1,70	2,93	1,96
<b>tai 50x20</b>	7,51	3,59	6,18	4,84	3,28
<b>tai 100x5</b>	0,68	0,22	0,41	0,29	0,23
<b>tai 100x10</b>	3,80	1,45	3,22	2,28	1,43
<b>tai 100x20</b>	7,16	3,33	6,61	5,15	3,41
<b>tai 200x10</b>	4,10	1,09	2,57	2,00	1,10
<b>tai 200x20</b>	8,52	3,34	6,74	5,28	3,65

Τα συμπεράσματα που βγαίνουν από τον παραπάνω πίνακα είναι τα εξής:

- Όλοι οι υβριδικοί αλγόριθμοι παρουσιάζουν καλύτερα αποτελέσματα από τον απλό DE. Άρα όντως ο DE δεν αρκεί για το FSSP και πρέπει απαραίτητα να συνδυαστεί με κάποιον άλλο αλγόριθμο για να επιλύσει το συγκεκριμένο πρόβλημα.
- Η απλή τεχνική LS είναι αρκετά αποτελεσματική βελτιώνοντας αρκετά την τιμή του DE. Ίσως δεν είναι απαραίτητο να εφαρμοστεί κάποιος περίπλοκος μεθευρετικός αλγόριθμος παρά ο βασικός DE με μια απλή τεχνική LS.
- Μόνο ο DEVNS φαίνεται να είναι πιο αποτελεσματικός ή να ανταγωνίζεται τον DELS.
- Ο DEILS παρουσιάζει καλά αποτελέσματα χωρίς όμως να υπερσχύει του DELS και του DEVNS.
- DESA<sub>2</sub> αποδίδει καλύτερα αποτελέσματα από τον DE αλλά όχι από τον DELS. Αυτό συμβαίνει επειδή ναι μεν εμπεριέχει μετασχηματισμό LS αλλά από την άλλη πλευρά αποδέχεται και τις κακές λύσεις της LS με αποτέλεσμα η νέα γενιά προς βελτιστοποίηση να είναι χαμηλότερου ποιοτικού επιπέδου από αυτή που δίνει ο DELS.
- Επιπλέον με την εφαρμογή των δύο DESA συμπεραίνεται ότι το κριτήριο της προσομοιωμένης απόπτωσης δεν ενδείκνυται για τον DE συγκριτικά με το κριτήριο της απληστίας στο συγκεκριμένο πάντα πρόβλημα.

- Στα προβλήματα 20 μηχανών οι αλγόριθμοι που δοκιμάστηκαν στην παρούσα εργασία είναι λιγότερο αποδοτικοί σε σχέση με αυτά των 10 και 5 μηχανών.

Όλοι οι HDE προσέγγισαν στα προβλήματα μικρών διαστάσεων τα ανώτερα όρια του Taillard. Ενδεικτικά στους παρακάτω πίνακες παρατίθενται μερικά από τα καλά αποτελέσματα για κάθε αλγόριθμο ξεχωριστά.

Πίνακας 16. αποτελέσματα της εφαρμογής του DESA που προσεγγίζουν τα ανώτερα όρια του Taillard

πρόβλημα	ανώτερο όριο Taillard (2001)	αποτέλεσμα εφαρμογής DESA	
		makespan	σειρά εργασιών
tai01	1278	1278	17 3 15 6 4 19 13 1 9 5 8 7 11 18 14 16 2 10 20 12
tai02	1359	1360	6 10 17 7 15 13 3 16 9 11 18 20 19 12 14 1 5 4 8 2
tai06	1195	1195	11 14 13 6 7 16 12 18 1 17 9 19 4 15 20 2 10 8 5 3
tai07	1239	1239	10 13 1 20 16 9 17 15 2 8 11 4 19 6 12 3 14 5 7 18
tai08	1206	1206	17 2 16 14 9 12 5 19 7 18 3 10 1 13 8 20 4 15 6 11
tai10	1108	1109	11 7 1 16 6 10 15 13 8 19 3 5 20 12 2 18 14 17 4 9
tai31	2724	2724	31 41 18 34 39 42 26 50 10 46 38 17 29 30 6 24 16 3 1 44 28 49 20 14 25 32 19 5 23 22 2 40 4 12 47 8 7 11 13 33 43 27 45 9 21 15 35 48 37 36

Πίνακας 17. αποτελέσματα της εφαρμογής του DEVNS που προσεγγίζουν τα ανώτερα όρια του Taillard

πρόβλημα	ανώτερο όριο Taillard (2001)	αποτέλεσμα εφαρμογής DEVNS	
		makespan	σειρά εργασιών
tai01	1278	1278	3 17 15 6 9 8 4 5 18 14 16 7 11 10 19 1 2 13 20 12
tai02	1359	1359	6 10 17 7 19 20 11 12 1 18 15 9 2 3 8 14 5 13 4 16
tai04	1293	1293	13 17 19 9 16 11 15 7 12 10 1 4 20 2 5 8 6 14 3 18
tai07	1239	1239	10 13 1 20 15 3 2 7 19 4 11 17 9 6 5 8 16 12 14 18
tai08	1206	1206	17 2 16 9 19 4 12 10 6 14 8 1 20 15 13 7 5 3 18 11
tai09	1232	1232	4 12 10 2 8 1 18 16 17 3 14 13 20 6 11 15 7 19 9 5
tai31	2724	2724	31 3 18 33 32 43 6 44 5 39 10 22 34 28 8 27 47 17 45 4 35 40 2 25 1 49 29 12 50 37 24 38 26 9 14 11 30 13 16 46 41 7 21 42 48 15 20 23 19 36
tai35	2863	2864	46 48 5 3 39 18 12 21 34 2 27 15 41 36 26 44 8 47 16 22 17 32 1



			31 28 30 29 14 50 43 33 20 11 35 4 25 19 9 23 40 38 7 24 6 37 10 45 13 49 42
tai36	2869	2869	41 30 39 1 50 33 8 44 10 47 15 11 9 22 19 29 43 23 20 45 37 26 40 28 35 21 3 46 32 7 6 2 4 17 24 25 36 48 18 34 5 14 49 31 27 38 13 12 16 42
tai40	2782	2782	6 2 24 34 10 27 20 31 44 8 30 39 37 29 9 49 25 45 28 17 36 3 35 13 48 15 5 16 18 21 12 33 22 11 42 14 40 1 38 41 47 7 4 46 23 19 50 43 26 32
tai61	5493	5493	90 25 39 49 38 73 4 85 74 98 43 21 75 27 86 100 71 44 69 33 46 23 78 24 13 70 60 81 52 29 7 37 28 40 62 87 94 32 11 31 59 95 89 64 58 54 92 9 55 72 65 79 16 80 83 26 42 2 47 48 19 34 3 30 53 93 15 66 22 61 5 56 96 76 41 63 17 36 68 45 91 6 51 18 67 50 99 10 57 77 97 8 20 14 82 12 88 35 1 84

Πίνακας 18. αποτελέσματα της εφαρμογής του DEILS που προσεγγίζουν τα ανώτερα όρια του Taillard

πρόβλημα	ανώτερο όριο Taillard (2001)	αποτέλεσμα εφαρμογής DEILS	
		makespan	σειρά εργασιών
tai02	1359	1359	6 10 17 7 18 15 19 4 12 3 20 13 9 1 8 2 16 5 11 14
tai04	1293	1293	13 17 19 9 16 11 15 7 12 20 2 1 10 4 5 8 6 14 3 18
tai06	1195	1195	11 14 13 17 6 16 18 2 12 9 1 7 19 4 15 20 10 8 5 3
tai07	1239	1239	10 13 1 20 15 2 11 3 8 6 17 19 12 16 4 5 7 9 14 18
tai08	1206	1207	17 16 9 12 13 2 19 5 20 10 18 8 14 3 4 7 1 15 6 11
tai31	2724	2724	31 41 18 40 32 28 27 48 39 10 29 6 30 14 33 4 3 42 9 7 46 35 8 34 25 45 47 15 26 2 22 1 43 12 11 24 49 13 17 50 20 21 38 5 23 16 44 37 19 36
tai35	2864	2864	46 48 7 3 39 36 21 31 11 10 40 44 32 28 8 9 41 38 12 18 47 35 4 43 22 1 24 2 14 6 37 25 15 16 45 30 34 29 23 20 19 50 33 5 13 27 26 17 49 42
tai40	2782	2782	6 8 2 24 27 31 42 1 44 47 5 14 18 33 49 11 16 30 29 20 13 35 12 10 15 23 50 9 4 22 48 45 38 36 28 37 46 17 40 34 43 25 21 41 39 7 19 3 26 32

## 4. ΕΦΑΡΜΟΓΗ ΣΕ ΠΡΑΓΜΑΤΙΚΑ ΔΕΔΟΜΕΝΑ

Στην παρούσα ενότητα μοντελοποιείται ένα πραγματικό σύστημα παραγωγής ως ευέλικτο σύστημα παραγωγής συνεχούς ροής. Σε αντίθεση με το παραδοσιακό μοντέλο, στη συγκεκριμένη περίπτωση υπάρχουν σε κάθε στάδιο διαφορετικοί περιορισμοί και οι μηχανές δεν είναι απαραίτητως πανομοιότυπες. Επίσης, το κριτήριο προς βελτιστοποίηση, το οποίο είναι πάλι η ελαχιστοποίηση του χρόνου περάτωσης της τελευταίας διεργασίας στο τελευταίο στάδιο, μετριέται με ώρες και λεπτά.

Η εταιρία που εξετάζεται είναι μια ελληνική εταιρία η οποία εμπορεύεται και παράγει καλλυντικά και ομοιοπαθητικά προϊόντα διαθέτοντας παγκόσμια παρουσία. Η παραγωγή των προϊόντων γίνεται σε σύγχρονη παραγωγική μονάδα πιστοποιημένη με ενοποιημένο σύστημα διαχείρισης ποιότητας και περιβάλλοντος και περιλαμβάνει το τμήμα υπερκάθαρου νερού, το τμήμα παραγωγής και το τμήμα συσκευασίας.

Λόγω της φύσης των προϊόντων που παράγει η εταιρία, η πολιτική παραγωγής που ακολουθείται είναι η δημιουργία stock στις αποθήκες ούτως ώστε όταν πραγματοποιηθεί η παραγγελία να υπάρχουν ήδη έτοιμα προϊόντα προς πώληση. Από την άλλη πλευρά βέβαια, τα προϊόντα έχουν ημερομηνία λήξης, άρα θα πρέπει να υπάρχει και κάποια σχετική πρόβλεψη.

Ενδιαφέρον παρουσιάζει επίσης η ένταξη του ποιοτικού ελέγχου ως στάδιο της παραγωγικής διαδικασίας. Βιομηχανίες όπως είναι αυτές των καλλυντικών, των τροφίμων και των φαρμάκων απαιτούν ποιοτικό – μικροβιολογικό έλεγχο με χρονική διάρκεια που πρέπει απαραίτητως να ληφθεί υπόψη στον χρονικό προγραμματισμό της παραγωγής. Στην περίπτωση όμως που τα στάδια παραγωγής εκτελούνται σειριακά και με συγκεκριμένη ροή λαμβάνοντας υπόψη και τον ποιοτικό έλεγχο ως στάδιο παραγωγής, το σύστημα λειτουργεί ως ευέλικτο ή μη σύστημα παραγωγής συνεχούς ροής.

#### 4.1. ΥΠΟΔΟΜΗ ΤΗΣ ΠΡΟΣ ΜΕΛΕΤΗ ΕΤΑΙΡΙΑΣ

Τα προϊόντα που παράγει η εταιρία μπορούν να χωριστούν σε τρεις γενικές κατηγορίες:

- α) αφρώδη (σαμπουάν, σαπούνια και αφρόλουτρα),
- β) κρεμώδη (κρέμες, αντηλιακά και γαλακτώματα) και
- γ) αλκοολούχα (αρώματα).

Η εταιρία διαθέτει πέντε καζάνια παραγωγής στα οποία παράγονται διαφορετικά προϊόντα:

- 6TN: Καζάνι χωρητικότητας έξι τόνων στο οποίο πραγματοποιείται η παραγωγή προϊόντων μεγαλύτερης συσκευασίας, κυρίως των αφρωδών.
- LACOUR: Καζάνι χωρητικότητας ενός τόνου στο οποίο πραγματοποιείται η παραγωγή συγκεκριμένων κρεμωδών προϊόντων.
- OSLA: Όπως στο καζάνι LACOUR, έχει χωρητικότητα ενός τόνου και πραγματοποιείται η παραγωγή συγκεκριμένων κρεμωδών προϊόντων διαφορετικών από αυτών του LACOUR.
- 200KG: Καζάνι διακοσίων κιλών στο οποίο πραγματοποιείται η παραγωγή κρεμωδών προϊόντων των οποίων η συσκευασία είναι μικρή, κυρίως των κρεμών για την περιοχή των ματιών.
- ALCOHOL: Καζάνι χωρητικότητας ενός τόνου στο οποίο παράγονται τα αλκοολούχα προϊόντα και βρίσκεται σε ειδικά διαμορφωμένη αίθουσα για τη μείωση του κινδύνου πυρκαγιάς.

Η εταιρία επίσης διαθέτει επτά αυτοματοποιημένες γραμμές συσκευασίας προϊόντων: STEV, SABRA, HISPAMEC, FARMAMAC, COMADIS, CAM2 και CAM1. Ανάλογα με τις ανάγκες του κάθε προϊόντος η εκάστοτε γραμμή συσκευασίας τροποποιείται.

Σημαντικό ρόλο στην παραγωγική διαδικασία παίζει ο ποιοτικός έλεγχος ο οποίος πραγματοποιείται τόσο στις πρώτες ύλες όσο στα ημιέτοιμα και στα τελικά προϊόντα. Στα

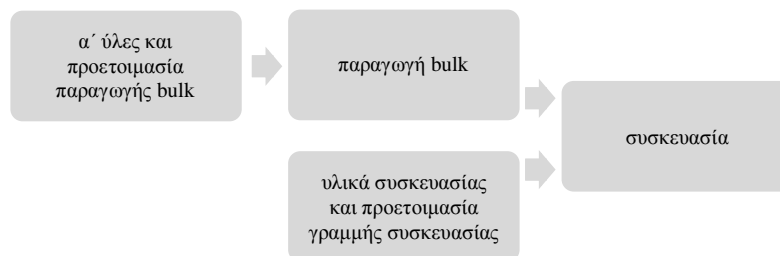
κρεμώδη και αφρώδη προϊόντα εκτός από τους απαιτούμενους χημικούς ελέγχους και ελέγχους κατάστασης πραγματοποιείται και μικροβιολογικός έλεγχος ενώ στα αλκοολούχα απαιτείται μόνο χρόνος ωρίμανσης.

## 4.2. ΔΙΑΔΙΚΑΣΙΑ ΠΑΡΑΓΩΓΗΣ

Η διαδικασία παραγωγής περιλαμβάνει όλες τις επιμέρους διεργασίες που πρέπει να πραγματοποιηθούν για να δημιουργηθούν από τις πρώτες ύλες τα τελικά προϊόντα. Διακρίνονται λοιπόν κάποιες γενικές φάσεις που περιλαμβάνουν τα επιμέρους στάδια της παραγωγικής διαδικασίας:

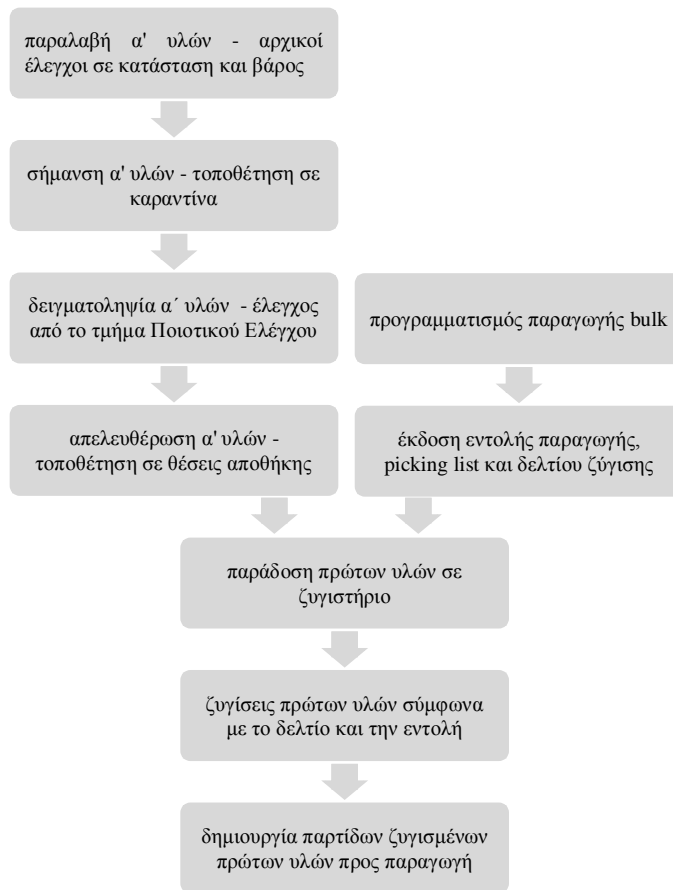
- πρώτες ύλες και προετοιμασία παραγωγής ημιέτοιμου προϊόντος
- παραγωγή ημιέτοιμου προϊόντος
- υλικά συσκευασίας και προετοιμασία γραμμής συσκευασίας
- συσκευασία

Στο παρακάτω διάγραμμα παρουσιάζονται οι φάσεις της παραγωγικής διαδικασίας με τη σειρά που πραγματοποιούνται στο εργοστάσιο.



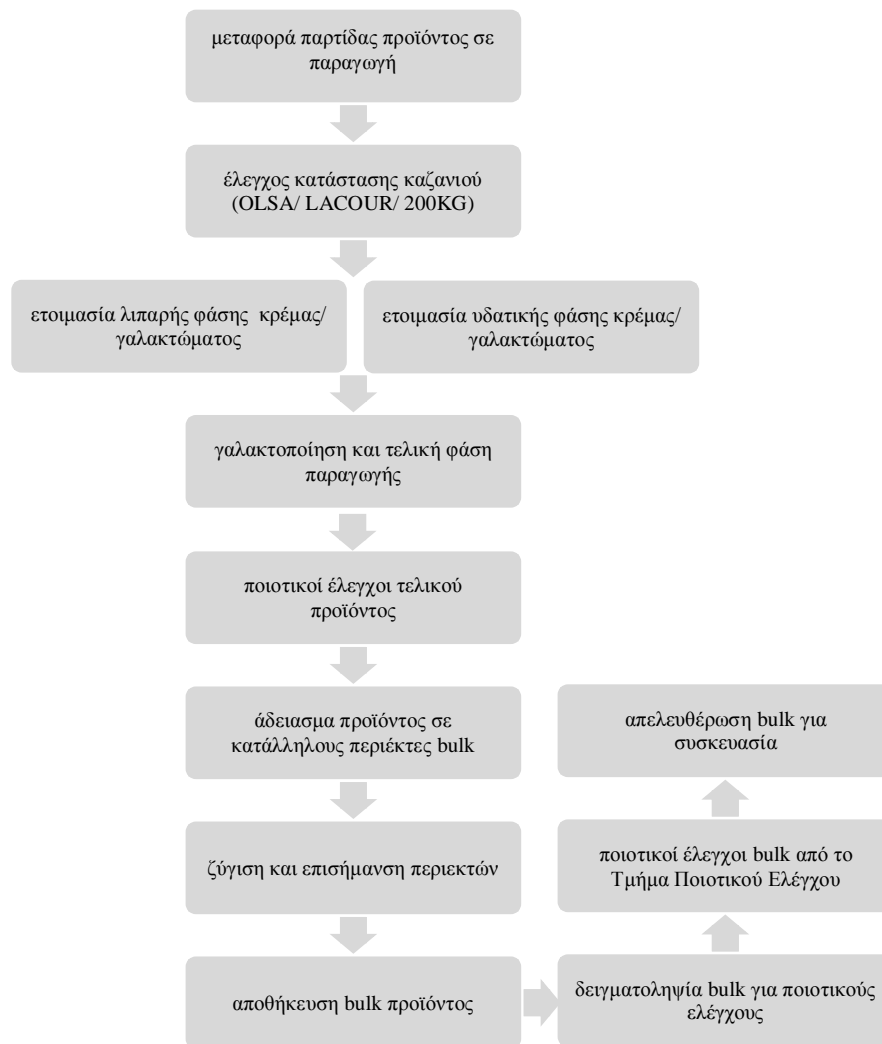
Εικόνα 32. οι τέσσερις φάσεις της παραγωγικής διαδικασίας

Προτού γίνει οποιαδήποτε διαδικασία παραγωγής ημιέτοιμου προϊόντος (bulk) θα πρέπει να έχουν καταφθάσει και να έχουν ελεχθεί οι πρώτες ύλες, καθώς επίσης θα πρέπει να έχει γίνει και η προετοιμασία παραγωγής. Στην Εικόνα 33. φαίνονται τα επιμέρους στάδια που εκτελούνται στη φάση των πρώτων υλών και της προετοιμασίας παραγωγής bulk.

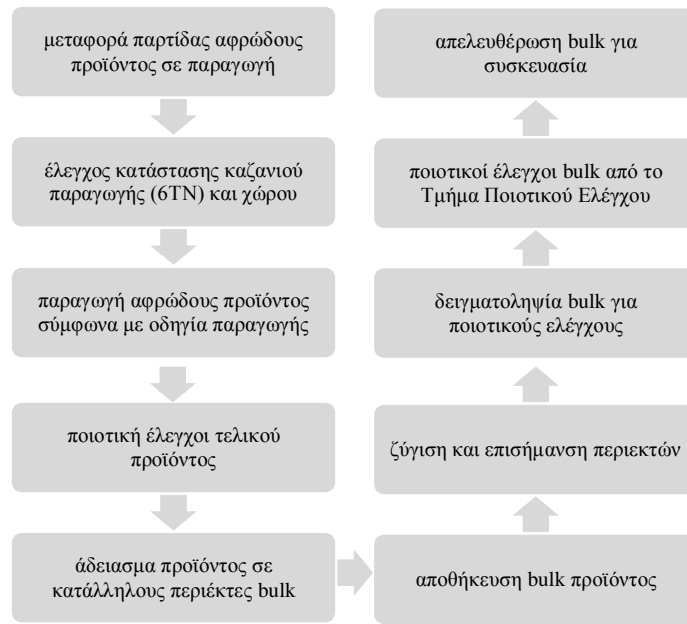


Εικόνα 33. η φάση των α' υλών και της προετοιμασίας παραγωγής ημιέτοιμου προϊόντος

Ανάλογα με την κατηγορία στην οποία ανήκει το προϊόν, η φάση της παραγωγής διαφέρει. Στα παρακάτω δύο διαγράμματα παρουσιάζονται η φάση παραγωγής bulk για την περίπτωση της παραγωγής κρεμώδους προϊόντος ή γαλακτώματος και για την περίπτωση αφρώδους προϊόντος.

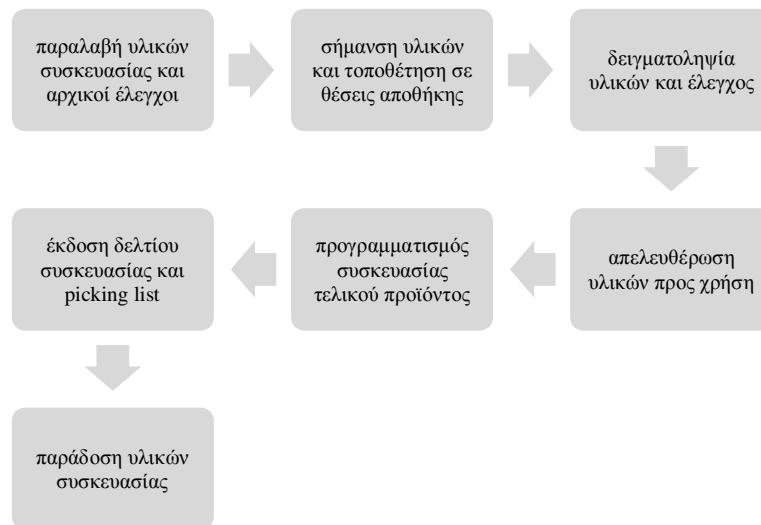


Εικόνα 34. η φάση της παραγωγής ημιέτοιμου κρεμώδους προϊόντος



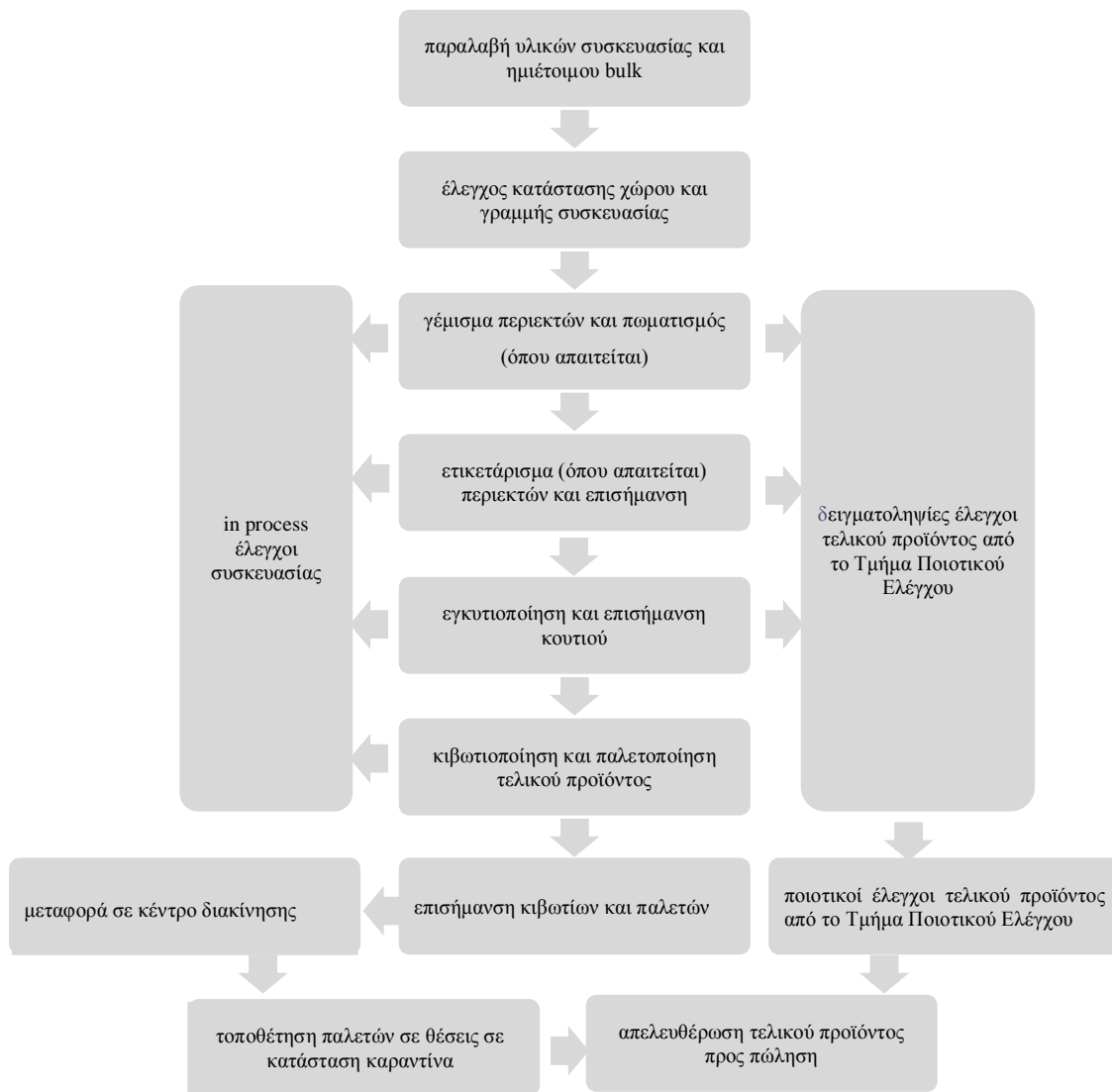
Εικόνα 35. η φάση της παραγωγής ημιτέτοιμου αφρώδους προϊόντος

Προτού ξεκινήσει το στάδιο της συσκευασίας θα πρέπει να έχουν προηγηθεί οι διαδικασίες που παρουσιάζονται στο παρακάτω διάγραμμα:



Εικόνα 36. η φάση των υλικών συσκευασίας και της προετοιμασίας της γραμμής συσκευασίας

Η τελευταία φάση της παραγωγικής διαδικασίας είναι αυτή της συσκευασίας της οποίας τα στάδια παρουσιάζονται στην Εικόνα 37. Αφού ολοκληρωθεί και η φάση αυτή, τα προϊόντα είναι έτοιμα προς διάθεση και πώληση στους καταναλωτές.



Εικόνα 37. η φάση της συσκευασίας



### 4.3. ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ

Η διαδικασία παραγωγής όπως περιγράφηκε στην προηγούμενη ενότητα αποτελείται από αρκετές διεργασίες, μερικές από τις οποίες μπορούν να πραγματοποιηθούν και παράλληλα. Άλλες επίσης μπορούν να ενσωματωθούν σε γενικότερα στάδια και άλλες απαιτούν μικρό χρονικό διάστημα για την πραγματοποίησή τους.

Στη συγκεκριμένη ενότητα θα θεωρηθεί η παραγωγή  $x$  τεμαχίων του  $y$  προϊόντος ως εργασία. Για παράδειγμα μια εργασία είναι η παραγωγή 100 τεμαχίων της κρέμας Wild Rose ή η παραγωγή 500 τεμαχίων του αντηλιακού YOGHART SPF30. Επιπλέον η παραγωγική διαδικασία θα χωριστεί σε συγκεκριμένα στάδια στα πλαίσια ενός ευέλικτου μοντέλου παραγωγής συνεχούς ροής.

Γίνεται η υπόθεση ότι τη στιγμή που πραγματοποιείται η παραλαβή των πρώτων υλών ολοκληρώνονται όλες οι αντίστοιχες διεργασίες της παραγωγικής διαδικασίας και οι πρώτες ύλες τοποθετούνται στην αποθήκη όντας έτοιμες προς χρήση. Θεωρείται δηλαδή ότι πριν ξεκινήσει η παραγωγή του bulk υπάρχουν οι απαραίτητες πρώτες ύλες στην αποθήκη. Η υπόθεση αυτή πραγματοποιείται διότι η παραγγελία των πρώτων υλών από την εταιρία γίνεται ανεξάρτητα και προηγείται της παραγωγής του bulk. Σίγουρα όμως βασίζεται στην πρόβλεψη πωλήσεων όπως και η παραγωγή του bulk άλλωστε. Με το ίδιο σκεπτικό όλα τα υλικά συσκευασίας είναι έτοιμα προς χρήση πριν την έναρξη της φάσης της συσκευασίας.

Ο προγραμματισμός παραγωγής και συσκευασίας, η πρόβλεψη πωλήσεων, η έκδοση εντολής παραγωγής και δελτίου συσκευασίας καθώς και όλες οι άλλες διεργασίες που έχουν πραγματοποιηθεί πριν την παραγωγή του bulk και της συσκευασίας δεν απασχολούν το συγκεκριμένο πρόβλημα. Οι συγκεκριμένες διεργασίες πραγματοποιούνται παράλληλα με άλλες διεργασίες και είναι έτοιμες πριν από την παραγωγή του bulk.

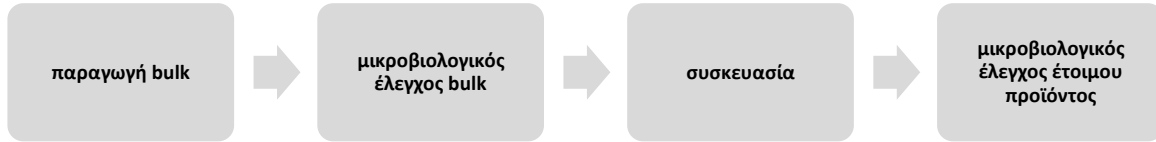
Στη φάση παραγωγής του bulk υπάρχουν αρκετές διεργασίες οι οποίες μπορούν όλες μαζί να θεωρηθούν ως μία. Αυτό δεν ισχύει για τον μικροβιολογικό έλεγχο ο οποίος πραγματοποιείται από το τμήμα Ποιοτικού Ελέγχου. Όπως και για τις διεργασίες της φάσης της συσκευασίας, οι επιμέρους διεργασίες μπορούν να θεωρηθούν ως μία εφόσον οι γραμμές συσκευασίας είναι πλήρως αυτοματοποιημένες. Μικροβιολογικός έλεγχος πραγματοποιείται και στη φάση της συσκευασίας και θεωρείται διαφορετική διεργασία εφόσον απαιτεί αρκετά μεγάλο χρονικό διάστημα για να πραγματοποιηθεί.

Λαμβάνοντας υπόψη τα παραπάνω η παραγωγική διαδικασία απλοποιείται και μοντελοποιείται στα εξής πλέον τέσσερα στάδια:

1. παραγωγή bulk
2. μικροβιολογικός έλεγχος bulk

3. συσκευασία
4. μικροβιολογικός έλεγχος έτοιμου προϊόντος

και διαγραμματικά:



Εικόνα 38. μοντελοποίηση προβλήματος

Στη συνέχεια θα αναλυθεί περαιτέρω το κάθε στάδιο ξεχωριστά.

#### 4.3.1. ΣΤΑΔΙΟ ΠΑΡΑΓΩΓΗΣ BULK

Είναι σχετικά δύσκολο να προσδιοριστεί η χρονική διάρκεια της παραγωγής του bulk, διότι μπορούν να συμβούν διάφορα απρόοπτα συμβάντα και ενέργειες που καθυστερούν τη διαδικασία όπως: ρύθμιση pH και ιζώδους, παράλληλη παραγωγή, παρακολούθηση R&D, βλάβη ατμογεννήτριας, αλλαγή βάρδιας, διόρθωση υφής, καθυστέρηση ψύξης, καθυστέρηση θέρμανσης κλπ. Παρόλα αυτά πραγματοποιήθηκε η προσπάθεια προσέγγισης της χρονικής διάρκειας της παραγωγής του bulk μετά από συλλογή στατιστικών δεδομένων.

Επίσης, στη χρονική διάρκεια της παραγωγής του bulk δεν παίζει ρόλο η ποσότητα όπως στην περίπτωση της συσκευασίας. Η παραγωγή γίνεται σε καζάνια συγκεκριμένου όγκου όπως έχει περιγραφεί παραπάνω και ανεξάρτητα με το πόσο γεμάτα είναι απαιτούν κάποιο συγκεκριμένο χρονικό διάστημα για να πραγματοποιηθεί η παραγωγή του bulk.

Το ανθρώπινο δυναμικό που ασχολείται με την παραγωγή του bulk είναι περιορισμένο διότι απαιτούνται ειδικευμένοι εργάτες. Για αυτόν τον λόγο μπορεί να πραγματοποιηθεί παράλληλη παραγωγή το πολύ σε δύο καζάνια. Υπάρχει δηλαδή ο περιορισμός που απαγορεύει και στις πέντε μηχανές-καζάνια να λειτουργούν παράλληλα. Επιπλέον η κάθε εργασία απαιτεί συγκεκριμένο καζάνι. Για παράδειγμα δεν είναι δυνατό ένα αλκοολούχο προϊόν να παραχθεί στο 6TN καζάνι που προορίζεται για τα αφρώδη προϊόντα.

Συνεπώς, το στάδιο παραγωγής του bulk έχει τους εξής περιορισμούς - διαφοροποιήσεις από το κλασικό FFSSP:

- **μόνο δύο παραγωγές μπορούν να εκτελούνται παράλληλα**
- **οι μηχανές δεν είναι πανομοιότυπες**
- **η κάθε εργασία εκτελείται σε μία συγκεκριμένη μηχανή και όχι σε οποιαδήποτε από τις πέντε**

#### 4.3.2. ΣΤΑΔΙΟ ΜΙΚΡΟΒΙΟΛΟΓΙΚΟΥ ΕΛΕΓΧΟΥ

Στο τμήμα Ποιοτικού Ελέγχου πραγματοποιείται μια σειρά από διαδικασίες που αφορούν τόσο το έτοιμο όσο και το ημιέτοιμο προϊόν. Οι οργανοληπτικοί, οι φυσικοχημικοί και οι έλεγχοι σταθερότητας απαιτούν μικρό χρονικό διάστημα σε σχέση με τον μικροβιολογικό έλεγχο, ο οποίος απαιτεί τουλάχιστον πέντε μέρες.

Πίνακας 19. Μικροβιολογικός έλεγχος για συγκεκριμένο κρεμώδες προϊόν

<b>ΜΙΚΡΟΒΙΟΛΟΓΙΚΟΣ ΕΛΕΓΧΟΣ</b> <b>MICROBIOLOGICAL TESTS</b>	<b>Όρια</b> <b>Limits</b>	<b>Μέθοδοι ελέγχου</b> <b>Test Method</b>
Aerobic microbial (TAMC)	Max.100cfu/ml	MFC-PC05-MM01
Yeast/ Moulds (TCYM)	Max. 10cfu/ml	MFC-PC05-MM01
Absence of Staphylococcus Aureus, Gram Negative rods and Candida Albicans (10g Enrichment)	Απουσία / Absence	MFC-PC05-MM01

Συνήθως στα πλαίσια του μικροβιολογικού ελέγχου πραγματοποιούνται οι διαδικασίες που περιγράφονται στον παραπάνω πίνακα, ενώ τα όρια διαφέρουν από προϊόν σε προϊόν. Για τον έλεγχο TAMC απαιτείται χρονική διάρκεια από 48 μέχρι 72 ώρες και για τον έλεγχο TCYM από 5 μέχρι 7 ημέρες. Στη συγκεκριμένη περίπτωση όμως τα προϊόντα δε δεσμεύονται καθ' όλη τη διάρκεια του μικροβιολογικού ελέγχου. Στον έλεγχο της παραγωγής του bulk δεσμεύονται από 0 μέχρι 3 μέρες και στην περίπτωση του έτοιμου

προϊόντος για 5 μέρες. Ο μικροβιολογικός έλεγχος εξακολουθεί **αυστηρά** να πραγματοποιείται ανεξάρτητα με την αποδέσμευση του προϊόντος και στην περίπτωση που βρεθεί κάποιο δείγμα το οποίο δεν πληροί τις προϋποθέσεις το προϊόν αυτομάτως αποσύρεται από όπου και αν βρίσκεται (στάδιο συσκευασίας, αποθήκη).

Επομένως,

- μετά το στάδιο της παραγωγής του bulk ακολουθεί το στάδιο του μικροβιολογικού ελέγχου του bulk το οποίο διαρκεί από 0 μέχρι 3 μέρες και
- μετά το στάδιο της συσκευασίας ακολουθεί το στάδιο του μικροβιολογικού ελέγχου του έτοιμου προϊόντος το οποίο διαρκεί 5 μέρες για όλα τα προϊόντα εκτός των αλκοολούχων (τα αλκοολούχα δεν απαιτείται να περάσουν από μικροβιολογικό έλεγχο).

Στο συγκεκριμένο στάδιο με τον όρο μηχανές εννοούνται οι συγκεκριμένοι ποιοτικοί έλεγχοι που πρέπει να πραγματοποιηθούν με συγκεκριμένη χρονική διάρκεια για την κάθε εργασία. Θεωρείται ότι υπάρχουν τέσσερις μηχανές, λαμβάνεται δηλαδή ο περιορισμός ότι μόνο τέσσερα δείγματα μπορούν να εξετάζονται παράλληλα στο εργαστήριο του ποιοτικού ελέγχου.

Το συγκεκριμένο στάδιο προσεγγίζει το θεωρητικό μοντέλο FFSSP που παρουσιάστηκε στην πρώτη ενότητα. Υπάρχουν δηλαδή **τέσσερις πανομοιότυπες μηχανές συνδεδεμένες παράλληλα και μπορούν να δεχτούν οποιαδήποτε εργασία σύμφωνα με τους περιορισμούς και τις υποθέσεις που παρουσιάστηκαν στην πρώτη ενότητα.**

#### 4.3.3. ΣΤΑΔΙΟ ΣΥΣΚΕΥΑΣΙΑΣ

Στο στάδιο της συσκευασίας υπάρχουν επτά γραμμές, οι οποίες μπορούν να λειτουργούν παράλληλα. Στο στάδιο της συσκευασίας δεν απαιτείται ειδικευμένο εργατικό προσωπικό,

έτσι σε αντίθεση με το στάδιο της παραγωγής του bulk, όλες οι γραμμές μπορούν να λειτουργούν παράλληλα.

Η κάθε εργασία όμως αντιστοιχεί σε συγκεκριμένη γραμμή συσκευασίας, άρα ενώ μπορούν να λειτουργούν παράλληλα όλες οι γραμμές συσκευασίας, απαγορεύεται να πραγματοποιούνται παράλληλα εργασίες που αντιστοιχούν στην ίδια γραμμή συσκευασίας.

Επομένως στο στάδιο της συσκευασίας:

- **οι μηχανές δεν είναι πανομοιότυπες**
- **η κάθε εργασία εκτελείται σε μία συγκεκριμένη μηχανή και όχι σε οποιαδήποτε από τις επτά.**

#### **4.3.4. ΔΕΔΟΜΕΝΑ**

Τα δεδομένα του προβλήματος βασίζονται σε μια υποθετική παραγγελία διαφορετικών προϊόντων. Το πλήθος των τεμαχίων που θα παραχθούν για το κάθε προϊόν θα είναι αυτό που απαιτείται για να αξιοποιηθεί η χωρητικότητα των καζανιών στο έπακρο. Δηλαδή, αν το προϊόν παράγεται στο καζάνι LACOUR τότε θα παραχθεί ποσότητα bulk σχεδόν ίση με 1 kg και θα συσκευαστεί ο αντίστοιχος αριθμός τεμαχίων.

Τα δεδομένα που έχουν δοθεί από την εταιρία αφορούν 55 διαφορετικά προϊόντα και είναι:

- Ο όγκος του bulk που μπορεί να παραχθεί στο κάθε καζάνι.
- Η χρονική διάρκεια παραγωγής του bulk για το κάθε προϊόν – εργασία  $j$ :  $bulk_dure_j$ .
- Το καζάνι το οποίο αντιστοιχεί σε κάθε προϊόν – εργασία  $j$ :  $boiler_j$ .
- Η γραμμή παραγωγής η οποία αντιστοιχεί σε κάθε προϊόν – εργασία  $j$ :  $packline_j$ .
- Ο όγκος που περιέχει η συσκευασία ενός τεμαχίου για κάθε προϊόν  $j$ :  $vproduct_j$ .

- Ο αριθμός των τεμαχίων που συσκευάζονται σε μία ώρα για κάθε προϊόν - εργασία  $j$ :  $packspeed_j$ .
- Η χρονική διάρκεια δέσμευσης του κάθε προϊόντος – εργασίας  $j$  από το τμήμα Ποιοτικού Ελέγχου μετά την παραγωγή του bulk σε ημέρες:  $qcbulkduree_j$ .
- Η χρονική διάρκεια δέσμευσης του κάθε προϊόντος – εργασίας  $j$  από το τμήμα Ποιοτικού Ελέγχου μετά τη συσκευασία σε ημέρες:  $qcpackdure_j$ .

Αφού είναι γνωστή η ποσότητα του bulk για κάθε εργασία και η ποσότητα που περιέχει η συσκευασία ενός τεμαχίου του κάθε προϊόντος, εύκολα υπολογίζεται ο αριθμός των τεμαχίων που θα πρέπει να παραχθούν ( $pieces_j$ ) για την ολοκλήρωση της κάθε εργασίας:

$$pieces_j = \frac{\text{ποσότητα bulk για την εργασία } j}{vproduct_j}.$$

Στη συνέχεια, μιας και είναι γνωστά τα τεμάχια που θα παραχθούν σε κάθε εργασία, μπορεί να υπολογιστεί η χρονική διάρκεια συσκευασίας ( $packdure_j$ ) για την κάθε εργασία ξεχωριστά σε ώρες:

$$packdure_j = 1.35 \frac{pieces_j}{packspeed_j}.$$

Παρατηρείται ότι το γινόμενο  $\frac{pieces_j}{packspeed_j}$  πολλαπλασιάζεται με τον συντελεστή 1.35. Ο συντελεστής αυτός σχετίζεται με το χρόνο εξάρμωσης και πλυσίματος της γραμμής συσκευασίας, ο οποίος στη συγκεκριμένη περίπτωση είναι σταθερός και ίσος με 1.35 για κάθε μία ώρα λειτουργίας της εκάστοτε γραμμής. Στο στάδιο της παραγωγής του bulk ο χρόνος εξάρμωσης και πλυσίματος εμπεριέχεται στη χρονική διάρκεια της παραγωγής και δε λαμβάνεται υπόψη.

Επομένως, τα δεδομένα τα οποία απαιτούνται για τον χρονικό προγραμματισμό και σχετίζονται με την υποδομή του εργοστασίου και την πρόβλεψη της παραγγελίας είναι:

- ο αριθμός εργασιών:  $D = 55$
- ο αριθμός σταδίων:  $stages = 4$
- ο αριθμός μηχανών στο κάθε στάδιο:  $M_1 = 5, M_2 = 4, M_3 = 7, M_4 = 4$
- οι χρονικές διάρκειες της κάθε εργασίας  $j$  για το κάθε στάδιο:  $bulkduration_j, packduration_j, qcbulkduration_j, qcpackduration_j$  (ΠΑΡΑΡΤΗΜΑ IV)
- οι μηχανές στα στάδια 1 και 2 που απαιτούνται για κάθε εργασία  $j$ :  $boiler_j, packline_j$  (ΠΑΡΑΡΤΗΜΑ IV)

και οι περιορισμοί:

- στο στάδιο 1 μόνο δύο παραγωγές μπορούν να εκτελούνται παράλληλα
- στα στάδια 1 και 3 οι μηχανές δεν είναι πανομοιότυπες
- στο στάδια 1 και 3 η κάθε εργασία εκτελείται σε μία συγκεκριμένη μηχανή και όχι σε οποιαδήποτε.

Επίσης, θα πρέπει να πραγματοποιηθεί η μετατροπή των μονάδων μέτρησης του χρόνου ούτως ώστε οι χρονικές διάρκειες σε όλα τα στάδια να μετρούνται με την ίδια μονάδα μέτρησης. Οι χρονικές διάρκειες στα στάδια του μικροβιολογικού ελέγχου μετρούνται σε ημέρες, στο στάδιο της συσκευασίας σε ώρες και στο στάδιο παραγωγής του bulk σε λεπτά. Ο χρονικός προγραμματισμός θα πραγματοποιηθεί σε λεπτά, άρα όλες οι μονάδες θα πρέπει να μετατραπούν σε λεπτά.

#### 4.3.5. ΖΗΤΟΥΜΕΝΟ ΚΑΙ ΚΡΙΤΗΡΙΟ ΠΡΟΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Το ζητούμενο στο συγκεκριμένο πρόβλημα είναι η σειρά με την οποία θα πραγματοποιηθούν οι εργασίες και το κριτήριο προς βελτιστοποίηση είναι η χρονική διάρκεια ολοκλήρωσης όλων των εργασιών. Για τις διάφορες αλληλουχίες εργασιών όπως προκύπτουν από τους μεθευρετικούς αλγορίθμους, οι οποίοι παρουσιάστηκαν στην προηγούμενη ενότητα, υπολογίζεται η χρονική στιγμή ολοκλήρωσης της τελευταίας εργασίας ξεκινώντας από τη

χρονική στιγμή 0. Η σειρά εργασιών η οποία θα υπερτερήσει χρονικά των άλλων θα είναι η λύση του προβλήματος.

Τα μέλη του πληθυσμού των μεθευρετικών αλγορίθμων υποδεικνύουν τη σειρά με την οποία θα εισαχθούν οι εργασίες στο σύστημα παραγωγής. Επειδή το συγκεκριμένο σύστημα είναι ευέλικτο και οι εργασίες εκτελούνται παράλληλα, η σειρά εκτέλεσης των εργασιών αλλάζει στα διαφορετικά στάδια του συστήματος.

Στο συγκεκριμένο πρόβλημα τα στάδια 1 και 3 έχουν διαφορετικούς περιορισμούς από τα στάδια 2 και 4. Επίσης πρέπει να δημιουργούνται διαφορετικά διανύσματα τα οποία αντιστοιχούν σε κάθε στάδιο και περιέχουν τις χρονικές στιγμές ολοκλήρωσης της κάθε εργασίας στο εκάστοτε στάδιο:  $bulktime_j$ ,  $packtime_j$ ,  $qcbulktime_j$ ,  $qcpacktime_j$  για τα στάδια 1, 3, 2 και 4 αντίστοιχα με  $j = 1, 2, \dots, D$ . Τα διαφορετικά διανύσματα εξυπηρετούν και την ενημέρωση του κάθε τμήματος ξεχωριστά. Δηλαδή, είναι σημαντικό να γνωρίζει το τμήμα Παραγωγής πότε θα πρέπει να εκτελέσει την εκάστοτε εργασία, το ίδιο φυσικά ισχύει τόσο για το τμήμα Συσκευασίας όσο και για το τμήμα Ποιοτικού Ελέγχου. Άρα θα πρέπει να φτιαχτούν διαφορετικές συναρτήσεις για το κάθε στάδιο ξεχωριστά και οι χρονικές διάρκειες του κάθε διανύσματος της καλύτερης λύσης να αποθηκεύονται και να μοιράζονται στα αρμόδια τμήματα της εταιρίας.

Στο στάδιο παραγωγής του bulk μόνο δύο εργασίες μπορούν να εκτελούνται παράλληλα. Επίσης, οι εργασίες που θα πραγματοποιούνται παράλληλα δεν πρέπει να αντιστοιχούν στο ίδιο καζάνι. Για αυτούς του λόγους στο πρόγραμμα εισάγονται τέσσερις μεταβλητές:  $boiler1$ ,  $boiler2$ ,  $dure1$ ,  $dure2$  από τις οποίες, οι δύο πρώτες δείχνουν τα καζάνια τα οποία είναι δεσμευμένα σε παράλληλη παραγωγή και οι δύο τελευταίες τις χρονικές διάρκειες που δεσμεύονται τα συγκεκριμένα καζάνια αντίστοιχα. Επίσης, ανεξάρτητα με την αρχική αλληλουχία των εργασιών, οι πρώτες δύο εργασίες πρέπει να αντιστοιχούν σε διαφορετικά καζάνια ούτως ώστε να ξεκινήσει η παραγωγική διαδικασία με παράλληλη παραγωγή. Παρακάτω παρουσιάζεται ο αλγόριθμος για την παραγωγή του bulk.



```

εισαγωγή αριθμού εργασιών D
εισαγωγή διανύσματος arrayj όπου j οι εργασίες με j = 1, 2, ..., D
εισαγωγή δεδομένων boilerj, bulkdurej
δήλωση μεταβλητών boiler1, dure1, boiler2, dure2, του βοηθητικού διανύσματος auxj και του διανύσματος
bulktimej το οποίο είναι και το ζητούμενο της συνάρτησης (j = 1, 2, ..., D)

boiler1 = boiler[array[1]]
dure1 = bulkdure[array[1]]
bulktime[array[1]] = dure1
AN boiler[array[2]] = boiler1 TOTE
    εύρεση του πρώτου στοιχείου n του διανύσματος arrayj για το οποίο ισχύει boiler[array[n]] ≠ boiler1
    και εισαγωγή του στη θέση 2 του διανύσματος arrayj
ΤΕΛΟΣ AN
boiler2 = boiler[array[2]]
dure2 = bulkdure[array[2]]
bulktime[array[2]] = dure2
ΓΙΑ i = 2 μέχρι i < D με βήμα 1 ΕΠΑΝΑΛΑΒΕ
    AN (boiler1 = boiler[array[i]]) Ή (boiler1 ≠ boiler[array[i]] ΚΑΙ boiler2 ≠ boiler[array[i]-1] ΚΑΙ
    dure1 ≤ dure2) TOTE
        dure1 = dure1 + bulkdure[array[i]]
        bulktime[array[i]] = dure1;
    ΑΛΛΙΩΣ
        dure2 = dure2 + bulkdure[array[i]]
        bulktime[array[i]] = dure2
    ΤΕΛΟΣ AN
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
εξαγωγή διανύσματος bulktimej
    
```

Εικόνα 39. συνάρτηση του σταδίου της παραγωγής bulk

Μετά την παραγωγή του bulk το επόμενο στάδιο της παραγωγικής διαδικασίας είναι ο μικροβιολογικός έλεγχος του bulk. Στο στάδιο αυτό αρχικά ελέγχεται ποια μηχανή θα απελευθερωθεί συντομότερα και στη συνέχεια εκτελείται στη μηχανή αυτή η τρέχουσα εργασία. Θα πρέπει δηλαδή να υπάρχει ένα διάνυσμα με στοιχεία όσα ο αριθμός των μηχανών το οποίο θα υποδεικνύει τη χρονική διάρκεια για την οποία είναι δεσμευμένη η

κάθε μηχανή ( $qc_m$  όπου  $m = 1, 2, \dots, M$ ). Παρακάτω παρουσιάζεται η συνάρτηση του μικροβιολογικού ελέγχου:

```

εισαγωγή αριθμού εργασιών D και αριθμού μηχανών M
εισαγωγή του διανύσματος bulktimej με j = 1, 2, ... D
δήλωση των διανυσμάτων qcm με m = 1, 2, ... M και qcbulktimej με j = 1, 2, ... D
μηδενισμός όλων των στοιχείων του διανύσματος qcm
ΓΙΑ i = 0 μέχρι i < D με βήμα 1 ΕΠΑΝΑΛΑΒΕ
    εντοπισμός του στοιχείου k του διανύσματος qcm με τη χαμηλότερη τιμή
    ΑΝ bulktime[array[i]] > qc[k] ΤΟΤΕ
        qcbulktime[array[i]] = bulktime[array[i]] + qcbulkdure[array[i]]
    ΑΛΛΙΩΣ
        qcbulktime [array[i]]=qc[k]+ qcbulkdure [array[i]-1]
    ΤΕΛΟΣ ΑΝ
        qc[k]=qc1time[array[i]-1];
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
εξαγωγή διανύσματος qcbulktimej
    
```

Εικόνα 40. συνάρτηση του σταδίου του μικροβιολογικού ελέγχου

Το επόμενο στάδιο είναι αυτό της συσκευασίας, όπου υπάρχουν επτά μηχανές οι οποίες μπορούν να λειτουργούν παράλληλα, αλλά η κάθε εργασία δε μπορεί να εκτελεστεί σε οποιαδήποτε μηχανή. Η σειρά με την οποία οι εργασίες έχουν ολοκληρωθεί από το στάδιο του μικροβιολογικού ελέγχου του bulk είναι γνωστή. Έτσι λοιπόν για την κάθε μηχανή ελέγχονται με τη σειρά όλες οι εργασίες και αν αντιστοιχούν στην τρέχουσα μηχανή τότε υπολογίζεται η χρονική διάρκεια εκτέλεσής τους στο συγκεκριμένο στάδιο. Αν τη στιγμή που η τρέχουσα εργασία έχει απελευθερωθεί από τον μικροβιολογικό έλεγχο η μηχανή στην οποία αντιστοιχεί είναι διαθέσιμη, τότε θα ξεκινήσει η εκτέλεσή της αμέσως, διαφορετικά θα «περιμένει» να αποδεσμευτεί η μηχανή. Εισάγεται λοιπόν μία μεταβλητή (dure) η οποία υπολογίζει τη χρονική διάρκεια που είναι δεσμευμένη η κάθε μηχανή. Η συνάρτηση του σταδίου της συσκευασίας παρουσιάζεται παρακάτω:

```

εισαγωγή αριθμού εργασιών D και αριθμού μηχανών M
εισαγωγή του διανύσματος qcbulktimej με j = 1, 2, ... D
εισαγωγή του διανύσματος packlinej με j = 1, 2, ... D
δήλωση της μεταβλητής dure και του διανύσματος packtimej με j = 1, 2, ... D
ΓΙΑ i = 1 μέχρι i < M με βήμα 1 ΕΠΑΝΑΛΑΒΕ
    dure = 0
    ΓΙΑ j = 0 μέχρι j < D με βήμα 1 ΕΠΑΝΑΛΑΒΕ
        ΑΝ packline[array[j]] = i ΤΟΤΕ
            dure = packdure[array[j]] + dure
            packtime[array[j]] = dure + qcbulktime[array[j]-1];
        ΤΕΛΟΣ ΑΝ
    ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
εξαγωγή διανύσματος packtimej
    
```

Εικόνα 41. συνάρτηση του σταδίου της συσκευασίας

Μετά από το στάδιο της συσκευασίας ακολουθεί το στάδιο του μικροβιολογικού ελέγχου του έτοιμου προϊόντος. Η συνάρτηση του σταδίου αυτού είναι η ίδια με τη συνάρτηση του μικροβιολογικού ελέγχου του bulk με τη διαφορά ότι τη θέση των διανυσμάτων bulktime<sub>j</sub>, qcbulktime<sub>j</sub> και qcbuldure<sub>j</sub> παίρνουν τα διανύσματα bulktime<sub>j</sub>, qcpacktime<sub>j</sub> και qcpackdure<sub>j</sub>.

Έτσι όπως έχουν διαμορφωθεί οι συναρτήσεις εισάγουν αρχικά το διάνυσμα array το οποίο εκφράζει τη σειρά των εργασιών με την οποία οι εργασίες είναι χρονικά έτοιμες να εισέλθουν στο εκάστοτε στάδιο. Μετά από κάθε στάδιο δηλαδή, θα πρέπει να επαναυπολογίζεται η σειρά των εργασιών (διάνυσμα array) μιας και αυτή θα έχει αλλάξει αφού οι εργασίες εκτελούνται παράλληλα.

Το κριτήριο προς βελτιστοποίηση λοιπόν είναι το qcpacktime<sub>array[D]</sub>, δηλαδή η χρονική στιγμή την οποία θα ολοκληρωθεί ο μικροβιολογικός έλεγχος έτοιμου προϊόντος της τελευταίας εργασίας του διανύσματος array.

#### 4.4. ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΩΝ – ΣΥΓΚΡΙΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Οι αλγόριθμοι DE, DELS και DEVNS εφαρμόστηκαν όπως ακριβώς στην ενότητα 3 με τη διαφορά ότι άλλαξε η διαδικασία υπολογισμού του κριτηρίου προς βελτιστοποίηση. Στο ΠΑΡΑΡΤΗΜΑ IV παρατίθενται κωδικοποιημένα τα δεδομένα έτσι όπως εισήχθησαν στο πρόγραμμα.

Αρχικά εφαρμόστηκε ο DELS και με τους τρεις μετασχηματισμούς LS για να εντοπιστεί ποιος είναι ο πιο αποδοτικός. Στον παρακάτω πίνακα παρουσιάζονται οι μέσες τιμές και τα καλύτερα αποτελέσματα των δέκα επαναλήψεων για τους τρεις διαφορετικούς μετασχηματισμούς LS. Παρατηρείται ότι ο μετασχηματισμός 1-0 relocate είναι ο πιο αποδοτικός για το συγκεκριμένο πρόβλημα και όχι ο 1-1 exchange που αναδείχθηκε στο θεωρητικό μοντέλο του FSSP της προηγούμενης ενότητας.

Πίνακας 20. αποτελέσματα των μετασχηματισμών LS του DELS

	<b>DELS<sub>1-1 exchange</sub></b>	<b>DELS<sub>1-0 relocate</sub></b>	<b>DELS<sub>mirror</sub></b>
<b>μέση τιμή (ώρες)</b>	224	<b>203</b>	228
<b>καλύτερο αποτέλεσμα (ώρες)</b>	219	<b>199</b>	219

Παρακάτω παρουσιάζονται τα καλύτερα αποτελέσματα και η μέση τιμή έτσι όπως προέκυψαν μετά από δέκα επαναλήψεις των αλγορίθμων DE, DELS, DEVNS και DEILS.

Πίνακας 21. συγκριτικά αποτελέσματα υβριδικών αλγορίθμων διαφορετικής εξέλιξης

	<b>DE</b>	<b>DELS<sub>1-0 relocate</sub></b>	<b>DEVNS</b>	<b>DEILS</b>
<b>μέση τιμή (ώρες)</b>	287	<b>203</b>	211	253
<b>καλύτερο αποτέλεσμα (ώρες)</b>	268	<b>199</b>	<b>199</b>	<b>199</b>

Εκ πρώτης όψεως φαίνεται ότι από τον αλγόριθμο DELS με μετασχηματισμό 1-0 relocate προκύπτουν τα καλύτερα αποτελέσματα ενώ τον ανταγωνίζεται ο DEVNS. Στην πραγματικότητα όμως ο DEVNS εμφανίζει μεγαλύτερη συχνότητα εμφάνισης καλών

αποτελεσμάτων, αλλά έχει μεγαλύτερη τυπική απόκλιση από τον  $DELS_{1-0\text{ relocate}}$ . Στον παρακάτω πίνακα φαίνονται αναλυτικά τα αποτελέσματα όπως προέκυψαν από τις δέκα επαναλήψεις των δύο αλγορίθμων.

Πίνακας 22. τυπική απόκλιση των αλγορίθμων DELS και DEVNS

<b>a/a</b>	<b><math>DELS_{1-0\text{ relocate}}</math></b>	<b>DEVNS</b>
<b>1</b>	200	<b>199</b>
<b>2</b>	204	<b>199</b>
<b>3</b>	200	<b>199</b>
<b>4</b>	<b>199</b>	267
<b>5</b>	205	<b>199</b>
<b>6</b>	200	203
<b>7</b>	219	223
<b>8</b>	199	<b>199</b>
<b>9</b>	200	203
<b>10</b>	200	219
<b>τυπική απόκλιση</b>	6.196	21.610

Το βέβαιο είναι πως και σε αυτό το πρόβλημα ο DE δεν είναι τόσο αποδοτικός και απαιτείται να ενισχυθεί είτε με απλές τεχνικές τοπικής αναζήτησης ( $DELS_{1-0\text{ relocate}}$ ) είτε με μεθευρετικούς αλγορίθμους βασισμένους στη γειτονιά αναζήτησης (DEVNS).

## 5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΣΧΟΛΙΑΣΜΟΣ

Στην παρούσα εργασία ο DE αποτέλεσε τη βάση όλων των αλγορίθμων που εφαρμόστηκαν. Για το λόγο αυτό, αρχικά πραγματοποιήθηκε η διερεύνησή του στο FSSP με σκοπό τον προσδιορισμό των παραμέτρων και της στρατηγικής που αποδίδουν περισσότερο.

Παρατηρήθηκε ότι η κάθε στρατηγική του DE αποδίδει καλύτερα με διαφορετικές παραμέτρους και ότι η απόδοση τόσο των παραμέτρων όσο και της στρατηγικής εξαρτάται από τα δεδομένα του εκάστοτε προβλήματος. Από τη στατιστική διαδικασία που πραγματοποιήθηκε στα προβλήματα του Taillard φάνηκε ότι η στρατηγική με την καλύτερη απόδοση εξαρτάται από τον αριθμό εργασιών του προβλήματος για συγκεκριμένες παραμέτρους. Τα συγκεκριμένα συμπεράσματα θα μπορούσαν να γίνουν αντικείμενο μελλοντικής μελέτης, ούτως ώστε να δημιουργηθεί ένας μηχανισμός που θα εντοπίζει τον καλύτερο συνδυασμό παραμέτρων και στρατηγικής για τα διαφορετικά δεδομένα του FSSP.

Ο DE εφαρμόστηκε στο FSSP σε συνδυασμό με μία τεχνική σύγκλισης και παρατηρήθηκε ότι τα αποτελέσματά του ήταν ικανοποιητικά για τις ομάδες προβλημάτων του Taillard 50x5 και 100x5. Γεγονός αναμενόμενο μιας και επιλέχτηκε συνδυασμός παραμέτρων και στρατηγικής που αποδίδει καλύτερα για αυτές τις ομάδες προβλημάτων. Αντίθετα, χειρότερα αποτελέσματα παρουσίασε στις ομάδες προβλημάτων 20 μηχανών.

Έπειτα πραγματοποιήθηκε η προσπάθεια εντοπισμού καλύτερου μετασχηματισμού LS μεταξύ των: 1-1 exchange, 1-0 relocate και mirror. Ο πρώτος προέκυψε να είναι πιο αποδοτικός για όλες τις ομάδες προβλημάτων που δοκιμάστηκαν, οπότε και εφαρμόστηκε στα υπόλοιπα προβλήματα του Taillard προκειμένου να πραγματοποιηθεί η σύγκριση του DE με τον DELS.

Παρατηρήθηκε ότι ο DE βελτιώθηκε πάρα πολύ με την εισαγωγή της LS τεχνικής. Μερικές φορές άγγιξε και το ανώτερο όριο του Taillard στα προβλήματα 5 μηχανών. Διαπιστώθηκε

ότι ο DE είναι πάρα πολύ ευαίσθητος στην LS εφόσον ακόμα και με μία μόνο επανάληψη μετασχηματισμού LS υπήρχε σημαντική βελτίωση των αποτελεσμάτων.

Την εφαρμογή του DELS στο FSSP ακολούθησαν οι εφαρμογές των αλγορίθμων DESA, DEISL και DEVNS στο ίδιο πρόβλημα. Όλοι οι υβριδικοί αλγόριθμοι παρουσίασαν καλύτερα αποτελέσματα από τον απλό DE. Επομένως, ο DE δεν αρκεί για το FSSP και πρέπει να συνδυαστεί με κάποιον άλλο αλγόριθμο ή τεχνική για να επιλύσει το συγκεκριμένο πρόβλημα.

Με την εφαρμογή στα προβλήματα του Taillard, διαπιστώθηκε ότι η απλή τεχνική LS είναι αρκετά αποτελεσματική βελτιώνοντας αρκετά την τιμή του DE και δεν είναι απαραίτητο να εφαρμοστεί κάποιος περίπλοκος μεθευρετικός αλγόριθμος. Μόνο ο DEVNS ανταγωνίστηκε τον DELS, αλλά και ο DEISL παρουσίασε καλά αποτελέσματα χωρίς όμως να υπερσχύει του DELS και του DEVNS.

Επιπλέον προέκυψε ότι το κριτήριο της προσομοιωμένης ανόπτησης δεν ενδείκνυται για τον DE συγκριτικά με το κριτήριο της απληστίας (στο συγκεκριμένο πάντα πρόβλημα). Παρόλα αυτά ο DESA<sub>2</sub> απέδωσε καλύτερα αποτελέσματα από τον DE αλλά όχι από τον DELS. Αυτό συμβαίνει επειδή, ναι μεν εμπεριέχει μετασχηματισμό LS, αλλά από την άλλη πλευρά αποδέχεται και τις κακές λύσεις της LS με αποτέλεσμα η νέα γενιά προς βελτιστοποίηση να είναι χαμηλότερου ποιοτικού επιπέδου από αυτή που δίνει ο DELS.

Ένα γενικό συμπέρασμα που προέκυψε από την εφαρμογή των αλγορίθμων στα προβλήματα του Taillard για το FSSP είναι ότι στα προβλήματα μεγάλων διαστάσεων οι αλγόριθμοι που δοκιμάστηκαν είναι λιγότερο αποδοτικοί σε σχέση με αυτά των μικρών και των μεσαίων.

Εκτός από το FSSP εξετάστηκε και ένα ευέλικτο σύστημα παραγωγής συνεχούς ροής ελληνικής εταιρίας. Διαπιστώθηκε για ακόμα μία φορά ότι τα θεωρητικά μοντέλα διαφέρουν από τα ρεαλιστικά, αλλά η προσέγγισή τους είναι εφικτή αν ληφθούν υπόψη η περιορισμοί που προκύπτουν από τις πραγματικές συνθήκες. Αφού μοντελοποιήθηκε το σύστημα

παραγωγής της εταιρίας σύμφωνα πάντα με το θεωρητικό FFSSP εφαρμόστηκε στο πρόβλημα τόσο ο DE όσο και οι DELS, DEVNS και DEILS.

Προσεγγίστηκαν πάλι και οι τρεις μετασχηματισμοί LS (1-1 exchange, 1-0 relocate, mirror) και παρατηρήθηκε ότι ο DELS με μετασχηματισμό 1-0 relocate είναι ο πιο αποδοτικός για το συγκεκριμένο πρόβλημα και όχι ο 1-1 exchange που αναδείχθηκε στο θεωρητικό μοντέλο του FSSP. Το συμπέρασμα αυτό προέκυψε από όλες τις επαναλήψεις των αλγορίθμων DELS.

Εκ πρώτης όψεως φάνηκε ότι από τον αλγόριθμο DELS με μετασχηματισμό 1-0 relocate προέκυψαν καλύτερα αποτελέσματα, αλλά στην πραγματικότητα ο DEVNS παρουσίασε μεγαλύτερη συχνότητα εμφάνισης καλών αποτελεσμάτων. Αυτό συνέβη διότι ο DEVNS είχε αρκετά μεγαλύτερη τυπική απόκλιση από τον  $DELS_{1-0\text{ relocate}}$ . Το σίγουρο είναι πως και σε αυτό το πρόβλημα ο DE δε φάνηκε να είναι τόσο αποδοτικός και χρειάστηκε να ενισχυθεί είτε με απλές τεχνικές τοπικής αναζήτησης ( $DELS_{1-0\text{ relocate}}$ ) είτε με μεθευρετικούς αλγορίθμους βασισμένους στη γειτονιά αναζήτησης (DEVNS).

Επομένως, το γενικό συμπέρασμα το οποίο προέκυψε τόσο από τη βιβλιογραφική ανασκόπηση όσο και από την πρακτική εφαρμογή της συγκεκριμένης εργασίας, είναι ότι ο κλασικός DE δεν αποτελεί την καλύτερη επιλογή για τα FSSP και FFSSP. Είναι απαραίτητο να συνδυαστεί με άλλους αλγορίθμους, τόσο με απλές τεχνικές LS, όσο με μεθευρετικούς αλγορίθμους βασισμένους στη γειτονιά αναζήτησης.



## ΑΝΑΦΟΡΕΣ

Ali M., & Torn A. (2004). Population Set-Based Global Optimization Algorithms: Some Modifications and Numerical Studies. *Computers & Operations Research* , 31 (10), 1703-1725.

Allen, J., & Minton, S. (1996). Selecting the right heuristic algorithm: runtime performance predictors. *11th Biennial Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence* (σσ. 41-53). London: Springer-Verlag.

Aydilek, H., & Allahverdi, A. (2006). Two-machine flowshop scheduling problem with bounded processing times to minimize total completion time. *International Journal of Production Economics* , 103 (1), 286-400.

Azizi, N., Liang, M., & Zolfaghari, S. (2009). Hybrid simulated annealing in flow-shop scheduling: a diversification and intensification approach. *International Journal of Industrial and Systems Engineering* , 4 (3), 326-348.

Baker, K., & Trietsch, D. (2009). *Principles of sequencing and scheduling*. Hoboken - New Jersey: John Wiley & Sons.

Blum, C., & Roli, A. (2008). Hybrid metaheuristics: An introduction. Στο C. Blum, A. Roli, & M. Sampels, *Hybrid metaheuristics: An emerging approach to optimization* (σσ. 1-30). Berlin - Heidelberg: Springer.

Blum, C., & Roli, A. (2008). Hybrid metaheuristics: An introduction. Στο C. Blum, A. Roli, & M. Sampels, *Hybrid metaheuristics: An emerging approach to optimization* (σσ. 1-30). Berlin - Heidelberg: Springer.

Brest J. (2006). Self-Adapting Control Parameters in Differential Evolution: A comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation* , 10 (6), 646 - 657.

Brockmann, K., & Dangelmaier, W. (1998). A parallel branch & bound algorithm for makespan optimal sequencing in flow shops with parallel machines. Στο P. Borne, M. Ksouri, & A. El Kamel (Επιμ.), *IMACS Multiconference on Computational Engineering in Systems Applications*, 3, σσ. 431 - 436. Hammamet, Tynisia.

Brosh, E., Levin, A., & Shavitt, Y. (2007). Approximation and heuristic algorithms for minimum-delay application-layer multicast trees. *IEEE/ACM Transactions on Networking* , 15 (2), 473-484.

Burke, E. K., Curtois, T., Hyde, M., Kendall, G., Ochoa, G., Petrovic, S., και συν. (2010). Iterated local search vs. hyper-heuristics: towards general-purpose search algorithms. *IEEE Congress on Evolutionary Computation*, (σσ. 3073 - 3080). Barcelona, Spain.

Costa, W. E., Goldbarg, M. C., & Goldbarg, E. G. (2011). *GRASP with path-relinking adapted to total flowtime permutational flowshop scheduling problem*. Technical Report, Universidade Federal do Rio Grande do Norte, Departamento de Informática e Matemática Aplicada.

Costa, W. E., Goldbarg, M. C., & Goldbarg, E. G. (2011). *New VNS heuristic for Total Flowtime Flowshop Scheduling Problem*. Technical Report, Universidade Federal do Rio Grande do Norte, Departamento de Informática e Matemática Aplicada.

Davendra, D., & Onwubolu, G. (2007). Flow shop scheduling using enhanced differential evolution algorithm. *21st European Conference on Modelling and Simulation*, (σσ. 259 - 264). Prague.

Davendra, D., Zelinka, I., & Onwubolu, G. (2009). Flow shop scheduling using clustered differential evolution. *33rd European Conference on Modelling and Simulation*, (σσ. 70-76). Madrid.

De La Fraga, L. G., & Coello Coello, C. A. (2011). A review of applications of evolutionary algorithms in pattern recognition. Στο P. S. Wang, *Pattern recognition, machine intelligence and biometrics* (σ. 866). Higher Education Press & Springer.

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*. Ph. D. Thesis, Politecnico di Milano.

Duda, J. (2006). Local search and nature based metaheuristics: a case of flowshop scheduling problem. *International Multiconference on Computer Science and Information Technology*, (σσ. 17 - 24). Wisła, Poland.

Eren, T. (2007). A multicriteria flowshop scheduling problem with setup times. *Journal of Materials Processing Technology* , 186, 60-65.

Gamperle G., Mueller S., & Koumoutsakos P. (2002). A Parameter Study for Differential Evolution. Στο Grmela A., & Mastorakis N., *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation* (σσ. 293-298). Athens: WSEAS Press.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flosshop and jobshop scheduling. *Mathematics of Operations Research* , 1 (2), 117 - 129.

Garey, M., & Johnson, D. (1990). *Computers and Intractability - A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman & Co. New York.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* , 13 (5), 533 - 549.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences* , 8 (1), 156 - 166.

Glover, F., & Laguna, M. (1989). Tabu search - Part 1. *ORSA Journal of Computing* , 1 (2), 109 - 206.

Gupta, J. N. (1988). Two-Stage, Hybrid Flowshop Scheduling Problem. *The Journal of the Operational Research Society* , 39 (4), 359 - 364.

Hansen, P., & Mladenovic, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research* , 130 (3), 449 - 467.

Hejazi, S., & Saghaian, S. (2005). Flowshop scheduling problems with makespan criterion: a review. *International Journal of Production Research* , 43 (14), 2895-2929.

Hitz, K. (1979). *Scheduling of flexible flowshops*. Cambridge: MIT Lab for Information and Decision Systems.

Holland, J. H. (1975). *Adaption in natural and artificial systems*. Ann Arbor, Michigan: University of Michigan Press.

Hooda, N., & Dhingra, A. K. (2011). Flow shop scheduling using simulated annealing: A review. *International Journal of Applied Engineering Research* , 2 (1), 234 - 249.

Ignall, E., & Schrage, L. (1965). Application of the branch and bound technique to some flow-shop scheduling problems. *Operations Research* , 13 (3), 400-412.

Johnson, S. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly* , 1 (1), 61-68.

Juan, A., Lourenco, H., Mateo, M., Castella, Q., & Barrios, B. (2012). ILS-ESP: an efficient, simple, and parameter-free algorithm for solving the permutation flow-shop problem. *6th Workshop on Computation*, (σ. 21). Covilha, Portugal.

Jungwattanakit, J., Reodecha, M., Chaovaitwongse, P., & Werner, F. (2007). Constructive and simulated annealing algorithms for hybrid flow shop problems with unrelated parallel machines. *Journal of Science and Technology*, , 12 (1), 31 - 41.

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *International Conference on Neural Networks* (σσ. 1942 - 1948). Washington: IEEE.

Khorshid-Doust, R. (1991). *A multiattribute approach to general flowshop problems*. Case Western Reserve University / OhioLINK.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science* , 220 (4598), 671 - 680.

Knuth, D. (1974). A terminological proposal. *SIGACT News* , 6 (1), 12-18.

Kulksar, G., & Erdelyi, F. (2005). Modeling and solving of the extended flexible flow shop scheduling problem. *Production Systems and Information Engineering* , 3, 121 - 139.

Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: A survey. *Computers & Industrial Engineering* , 37, 57 - 61.

Low, C. (2005). Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. *Computers & Operations Research* , 32 (8), 2013 - 2025.

Low, C., Yeh, J., & Huang, K. (2004). A robust simulated annealing heuristic for flow shop scheduling problems. *The International Journal of Advanced Manufacturing Technology* , 23 (9 - 10), 762 – 767.

Nawaz, M., Ensore, E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *OMEGA* , 11 (1), 91-95.

Nearchou, A. C. (2004). A novel metaheuristic approach for the flow shop scheduling problem. *Engineering Applications of Artificial Intelligence* , 17 (3), 289 - 300.

Onwubolu, G., & Davendra, D. (2006). Scheduling flow shops using differential evolution algorithm. *European Journal of Operational Research* , 171 (2), 674 - 692.

Pan, Q. K., Tasgetiren, M. F., & Liang, Y. C. (2007). A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Genetic and Evolutionary Computation Conference 2007*, (σσ. 126 - 133). London, England.

Pan, Q. K., Tasgetiren, M. F., & Liang, Y. C. (2008). A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Computers & Industrial Engineering* , 55 (4), 795 - 816.

Pan, Q., & Ruiz, R. (2011). *A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime*. Operativa Aplicadas y Calidad, Departamento de Estadística e Investigación.

Pan, Q., & Ruiz, R. (2011). Local search methods for the flowshop scheduling problem with flowtime minimization. *European Journal of Operational Research* , 222 (1), 31 - 43.

Pinedo, M. (2008). *Scheduling: Theory, algorithms, and systems*. New York: Springer Publishing Company, Incorporated.

Price K., & Storn R. (2001). *Differential Evolution Homepage*. Ανάκτηση από <http://www.icsi.berkeley.edu/~storn/code.html>

Price K., Storn R., & Lampinen J. (2005). *Differential Evolution - A practical Approach to Global Optimization*. Berlin, Heidelberg: Springer - Verlag.

Price, K. V., Storn, R., & Lampinen, J. A. (2005). *A practical approach to global optimization*. Germany: Springer.

Price, K., & Storn, R. (2001). *Differential Evolution Homepage*. Ανάκτηση από <http://www.icsi.berkeley.edu/~storn/code.html>

Qian, B., Wang, L., Hu, R., Wang, W., Huang, D., & Wang, X. (2008). A hybrid differential evolution method for permutation flow-shop scheduling. *The International Journal of Advanced Manufacturing Technology* , 38 (7-8), 757 - 777.

Quadt, D., & Kuhn, H. (2007). A taxonomy of flexible flow line scheduling procedures. *European Journal of Operational Research* , 178 (3), 686 – 698.

Rajkumar, R., & Shahabudeen, P. (2007). Performance evaluation of simulated annealing algorithm for flowshop scheduling problems. *The International Journal of Applied Management and Technology* , 5 (3), 172 - 189.

Ruiz, R., & Strutzle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research* , 177 (3), 2033-2049.

Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research* , 205 (1), 1-18.

Ruiz, R., Serifoglu, F. S., & Urlings, T. (2006). Modeling realistic hybrid flexible flowshop scheduling problems. *Computers & Operations Research* , 35 (4), 1151 - 1175.

Sbihi, M., & Varnier, C. (2008). Single-machine scheduling with periodic and flexible periodic maintenance to minimize maximum tardiness. *Computers & Industrial Engineering* , 55 (4), 830-840.

Sethanan, K. (2001). *Scheduling flexible flowshops with sequence dependent setup times*. Ph. D. Thesis, West Virginia University, Department of Industrial and Management Systems Engineering, Morgantown, West Virginia.

Sorn R., & Price K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* , 11 (4), 341-359.

Sriskandarajah, C., & Sethi, S. (1989). Scheduling algorithms for flexible flowshops: Worst and average case performance. *European Journal of Operational Research* , 43 (2), 143-160.

Storn, R. (1996). On the usage of Differential Evolution for function. *Biennial Conference of the North American Fuzzy Information Processing Society*, (σσ. 519-523). Berkeley, California.

Storn, R. (1996). On the usage of Differential Evolution for function. *Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society*, (σσ. 519-523).

Storn, R. P. (1997). A Simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* , 11 (4), 341-359.

Storn, R., & Price, K. (1995). *Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical Report: TR-95-012, International Computer Science Institute.



Storn, R., & Price, K. (1997). Differential Evolution - A Simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* , 11 (4), 341-359.

Stutzle, T. (1998). *Appllyng iterated local search to the permutation flow shop problem*. Technical Report, Darmstadt University of Technology, Intellectics Group, Computer Science Department, Darmstadt.

Taillard E. (1993). Benchmarks for Basic Scheduling Problems. *European Journal of Operational Research* , 64 (2), 278-285.

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*. Hoboken: John Wiley & Sons.

Tasgetiren, M. F., Liang, Y. C., Sevki, M., & Gencyilmaz, G. (2004). Differential evolution algorithm for permutation flowshop sequencing problem with makespan criterion. *4th International Symposium on Intelligent Manufacturing Systems*, (σσ. 442 - 452). Sakarya, Turkey.

Urlings, T. (2011). *Heuristics and metaheuristics for heavily constrained hybrid flowshop problems* (Τόμ. 331, Dissertations in Artificial Intelligence). IOS Presss.

Vallada, E., & Ruiz, R. (2010). Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem. *Omega* , 38, 57 - 67.

Varadharajan, T. K., & Rajendran, C. (2005). A multi-objective simulated-annealing algorithm for scheduling in flowshps to minimize the makespan and total flowtime of jobs. *European Journal of Operational Research* , 167 (3), 772 - 795.

Verma, S., & Dessouky, M. (1999). Multistage hybrid flowshop scheduling with identical jobs and uniform parallel machines. *Journal of Scheduling* , 2 (3), 135 - 150.

Vob, S. (2009). Metaheuristics. Στο C. Floudas, & P. Pardalos, *Encyclopedia of Optimization* (Τόμ. 4, σσ. 2061-2075). Verlag: Springer.

Xu, X., Xiang, Z., & Wang, W. (2010). A self-adaptive differential evolution for the permutation flow shop scheduling. *49th IEEE Control and Decision Conference* (σσ. 155 - 160). Atlanta, Georgia: IEEE Industrial Electronics.

Xu, Y., & Wang, L. (2011). Differential evolution algorithm for hybrid flow-shop scheduling problems. *Journal of Systems Engineering and Electronics* , 22 (5), 794 – 798.

Yagmahan, B., & Yenisey, M. M. (2009). Scheduling practice and recent developments in flow shop and job shop scheduling. Στο U. K. Chakraborty, *Computational intelligence in flow shop and job shop scheduling* (Τόμ. 230, σσ. 261 - 300). Springer.

Younes, N., Santo, D. L., & Maria, A. (1998). A Simulated Annealing approach to scheduling in a flow shop with multiple processors. *Industrial Engineering Research Conference*. Banff, Canada.

Zaharie D. (2002). Critical Values for the Control Parameters of Differential Evolution Algorithms. *8th International Conference on Soft Computing* (σσ. 62-67). Brno: Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automation and Computer Science.

Zhang, L., Wang, L., & Tang, F. (2003). Hypothesis - test based simulated annealing for stochastic flow shop scheduling. *The Second International Conference on Machine Learning and Cybernetics*, (σσ. 1607-1612). Xi'an.

Zhonghua, H., Haibo, S., & Chang, L. (2009). Differential evolution algorithm for the earliness/tardiness hybrid flow-shop. *Third International Symposium on Intelligent*

*Information Technology Application*. 2, σσ. 188 - 193. Piscataway, New Jersey: IEEE Conference Publications .

Zobolas, G., Tarantilis, C., & Ioannou, G. (2008). Exact, Heuristic and meta-heuristic algorithms for solvings shop scheduling problems. Στο F. Xhafa, & A. Abraham, *Metaheuristics for Scheduling in Industrial and Manufacturing Applications* (σ. 346). Springer.

Zobolas, G., Tarantilis, C., & Ioannou, G. (2008). Exact, Heuristic and meta-heuristic algorithms for solvings shop scheduling problems. Στο F. Xhafa, & A. Abraham, *Metaheuristics for Scheduling in Industrial and Manufacturing Applications* (σ. 346). Verlag Berlin Heidelberg: Springer.

Αναγνωστόπουλος, Κ., & Κώτσικας, Α. (2004). Στατιστικές τεχνικές για την αξιολόγηση ευρετικών αλγορίθμων. *17ο Πανελλήνιο Συνέδριο Στατιστικής* (σσ. 67 -74). Λευκάδα: Ελληνικό Στατιστικό Ινστιτούτο.

Μαρινάκης, Ι., & Μυγδαλάς, Α. (2008). *Σχεδιασμός και βελτιστοποίηση της εφοδιαστικής αλυσίδας*. Θεσσαλονίκη: Σοφία.

Μαρινάκης, Ι., Μαρινάκη, Μ., Ματσατσίνης, Ν., & Ζοπουνίδης, Κ. (2011). *Μεθευρετικοί και εξελικτικοί αλγόριθμοι σε προβλήματα διοικητικής επιστήμης*. Αθήνα: Κλειδάριθμος.



## ΠΑΡΑΡΤΗΜΑ Ι

Μικρότερη τιμή του makespan και μέση τιμή δέκα επαναλήψεων για τις δέκα στρατηγικές του DE στα δεδομένα του Taillard για το FSSP με  $F=0.2$ ,  $CR=0.7$ ,  $NP=2D$  και  $GMAX=500$ .

min=μικρότερη τιμή

μ.τ.=μέση τιμή makespan

προβλήματα 20x5

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai001	min	<b>1297</b>	<b>1297</b>	<b>1297</b>	<b>1297</b>	<b>1297</b>	<b>1297</b>	<b>1297</b>	<b>1297</b>	<b>1297</b>	1303
	μ.τ.	<b>1297,9</b>	1299,3	1301,8	1301,6	1300,3	1333,0	1309,7	1303,4	1330,5	1313,4
tai002	min	<b>1366</b>	<b>1366</b>	<b>1366</b>	<b>1366</b>	<b>1366</b>	1368	1366	1367	1366	1369
	μ.τ.	1369,4	1372,5	1368,9	<b>1368,4</b>	1370,5	1385,3	1372,5	1372,4	1376,6	1376,7
tai003	min	1116	1134	1144	1122	1125	1142	1132	1147	<b>1115</b>	1148
	μ.τ.	<b>1137,0</b>	1147,4	1150,0	1141,7	1142,7	1185,8	1157,7	1159,2	1180,9	1160,6
tai004	min	1340	1351	<b>1336</b>	1341	1346	1357	1343	1362	1349	1357
	μ.τ.	1361,9	1365,1	1364,3	<b>1353,3</b>	1365,0	1398,7	1363,5	1379,5	1397,2	1373,0
tai005	min	<b>1250</b>	1254	1254	1250	1273	1287	1265	1261	1277	1261
	μ.τ.	<b>1272,0</b>	1276,8	1272,4	1272,0	1280,7	1316,3	1285,9	1284,8	1312,3	1290,2
tai006	min	<b>1214</b>	1226	1229	<b>1214</b>	1217	1234	1224	1224	1244	1224
	μ.τ.	1244,7	1245,0	1242,2	<b>1229,1</b>	1241,2	1272,3	1246,1	1249,1	1275,4	1254,7
tai007	min	<b>1251</b>	<b>1251</b>	<b>1251</b>	<b>1251</b>	<b>1251</b>	1263	<b>1251</b>	1258	1259	1254
	μ.τ.	<b>1254,8</b>	1261,7	1260,7	1262,0	1260,9	1285,7	1261,4	1269,3	1284,5	1265,2
tai008	min	1249	<b>1241</b>	1250	1242	1255	1277	1273	1268	1255	1275
	μ.τ.	<b>1263,4</b>	1270,8	1268,4	1266,1	1274,9	1301,5	1283,6	1286,4	1294,0	1285,6
tai009	min	<b>1263</b>	1264	1271	1272	1265	1283	1265	1265	1287	1272
	μ.τ.	<b>1277,8</b>	1288,0	1289,8	1286,0	1289,6	1336,6	1284,3	1293,0	1328,0	1293,9
tai010	min	1135	1148	1137	<b>1127</b>	1150	1154	1161	1160	1174	1170
	μ.τ.	1156,9	1167,0	<b>1153,4</b>	1156,7	1161,7	1190,9	1174,0	1172,2	1187,7	1181,3

προβλήματα 20x10

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai011	min	1651	1679	1669	<b>1635</b>	1654	1691	1665	1663	1684	1663
	μ.τ.	<b>1664,1</b>	1692,0	1694,0	1671,4	1683,3	1735,3	1687,6	1699,3	1714,5	1707,3

tai012	min	1738	1758	1751	<b>1737</b>	1745	1768	1758	1789	1770	1783
	μ.τ.	<b>1761,2</b>	1774,0	1769,8	1768,9	1780,8	1810,9	1784,0	1801,0	1796,1	1802,7
tai013	min	1566	1580	1570	<b>1546</b>	1548	1610	1554	1599	1581	1600
	μ.τ.	<b>1583,3</b>	1601,5	1589,2	1588,7	1595,5	1655,6	1598,9	1628,1	1626,9	1626,3
tai014	min	1457	1449	1457	<b>1432</b>	1456	1494	1470	1472	1481	1491
	μ.τ.	1476,1	1481,0	1473,9	<b>1467,2</b>	1477,3	1536,6	1498,6	1500,8	1516,5	1521,8
tai015	min	<b>1482</b>	1533	1505	1492	1510	1509	1530	1541	1495	1513
	μ.τ.	1521,7	1545,2	1538,0	<b>1519,0</b>	1536,9	1583,9	1545,9	1569,4	1551,5	1568,1
tai016	min	1440	1440	1471	<b>1438</b>	1471	1480	1447	1481	1490	1449
	μ.τ.	<b>1456,8</b>	1466,2	1485,8	1458,1	1478,9	1522,2	1462,0	1505,6	1515,1	1482,5
tai017	min	1539	1537	1553	<b>1533</b>	1558	1559	1567	1562	1562	1559
	μ.τ.	<b>1554,2</b>	1562,7	1569,9	1555,2	1565,7	1605,5	1578,4	1584,3	1600,1	1585,4
tai018	min	1595	1601	<b>1585</b>	1618	1621	1632	1621	1664	1645	1658
	μ.τ.	<b>1622,7</b>	1630,9	1635,6	1638,6	1641,8	1681,3	1650,6	1674,8	1679,8	1675,7
tai019	min	1657	1653	<b>1646</b>	1648	1669	1689	1684	1662	1685	1678
	μ.τ.	1677,0	1669,4	1670,6	<b>1668,5</b>	1676,7	1705,8	1698,2	1692,9	1701,1	1709,3
tai020	min	1668	1658	1665	1656	1667	1671	<b>1643</b>	1691	1677	1684
	μ.τ.	1680,0	1682,7	1680,2	<b>1670,9</b>	1685,0	1704,6	1678,1	1705,6	1699,9	1700,7

προβλήματα 20x20

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai021	min	<b>2374</b>	2398	2391	2376	2403	2397	2386	2427	2426	2417
	μ.τ.	2396,8	2416,1	2410,6	<b>2396,4</b>	2420,6	2454,9	2426,0	2447,7	2449,8	2442,4
tai022	min	<b>2145</b>	2197	2186	2165	2191	2239	2165	2201	2172	2228
	μ.τ.	<b>2188,8</b>	2220,0	2222,5	2190,1	2213,7	2278,3	2208,8	2247,5	2260,2	2245,6
tai023	min	2405	2413	2422	2410	<b>2397</b>	2455	2400	2445	2429	2436
	μ.τ.	2425,8	2441,5	2439,0	<b>2425,3</b>	2439,0	2477,8	2434,6	2471,2	2471,9	2457,1
tai024	min	<b>2295</b>	2296	2320	2301	2326	2334	2345	2336	2351	2364
	μ.τ.	<b>2327,0</b>	2342,4	2335,7	2328,4	2352,9	2388,6	2376,8	2378,3	2392,5	2388,0
tai025	min	<b>2344</b>	2363	2382	2351	2374	2393	2351	2396	2387	2378
	μ.τ.	<b>2369,0</b>	2397,0	2405,6	2377,9	2396,4	2466,0	2381,3	2425,2	2456,9	2397,8
tai026	min	<b>2271</b>	2327	2324	2313	2296	2360	2287	2338	2314	2332
	μ.τ.	<b>2304,1</b>	2339,3	2342,1	2328,9	2326,9	2385,0	2328,0	2357,6	2369,5	2361,2
tai027	min	2357	2371	2357	<b>2332</b>	2370	2391	2373	2396	2412	2393
	μ.τ.	<b>2372,3</b>	2395,6	2381,5	2375,7	2391,5	2427,4	2399,2	2412,4	2433,6	2418,3
tai028	min	<b>2265</b>	2279	2275	2278	2288	2315	2291	2323	2311	2319
	μ.τ.	<b>2289,2</b>	2315,2	2303,2	2297,3	2310,7	2365,5	2321,8	2345,9	2350,4	2343,1
tai029	min	2308	2348	2345	2308	2333	2399	<b>2295</b>	2352	2357	2345
	μ.τ.	<b>2341,9</b>	2377,1	2377,2	2345,5	2371,1	2444,0	2348,9	2398,8	2423,9	2372,8
tai030	min	<b>2271</b>	2276	2299	2278	2321	2300	2305	2326	2312	2347

	μ.τ.	<b>2298,4</b>	2326,2	2327,7	2306,3	2337,7	2372,9	2354,3	2360,1	2362,2	2374,7
--	------	---------------	--------	--------	--------	--------	--------	--------	--------	--------	--------

προβλήματα 50x5

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai031	min	2740	2747	<b>2729</b>	2730	2731	2744	2762	2774	2730	2776
	μ.τ.	2751,7	2755,5	<b>2745,0</b>	2752,0	2751,6	2758,8	2785,7	2790,9	2753,6	2790,2
tai032	min	2890	2896	2895	2916	2912	<b>2884</b>	2922	2943	2897	2939
	μ.τ.	2913,7	2932,1	<b>2910,5</b>	2923,2	2934,0	2940,3	2958,8	2971,2	2938,3	2962,1
tai033	min	2671	2670	2671	2677	2672	2666	2707	2718	<b>2662</b>	2717
	μ.τ.	2684,2	2692,3	<b>2682,9</b>	2689,6	2691,4	2700,6	2719,5	2735,7	2688,7	2725,8
tai034	min	2807	2832	2814	<b>2794</b>	2827	2806	2860	2856	2797	2834
	μ.τ.	2838,7	2847,0	<b>2831,6</b>	2836,6	2846,3	2846,2	2879,0	2886,0	2842,5	2873,3
tai035	min	2891	2908	<b>2890</b>	2907	2907	2905	2931	2905	2900	2922
	μ.τ.	2913,4	2915,9	<b>2905,0</b>	2915,3	2919,7	2924,8	2940,1	2936,9	2918,5	2947,6
tai036	min	2878	2872	2873	2871	<b>2868</b>	2873	2908	2932	2873	2892
	μ.τ.	2895,1	2902,2	<b>2887,3</b>	2897,4	2903,6	2904,9	2935	2954,7	2907,2	2939,3
tai037	min	<b>2786</b>	2802	2796	2788	2809	2807	2809	2846	2794	2840
	μ.τ.	2811,3	2822,8	<b>2806,3</b>	2806,9	2821,9	2832,2	2844,1	2859,4	2809,0	2858,4
tai038	min	2741	2749	2730	2733	2741	<b>2729</b>	2805	2810	2741	2807
	μ.τ.	2765,6	2771,8	<b>2753,0</b>	2757,4	2766,9	2768,0	2824,4	2830,5	2768,8	2824,7
tai039	min	2606	2615	2616	2606	2628	2600	2666	2662	<b>2598</b>	2651
	μ.τ.	<b>2632,6</b>	2638,4	2623,6	2627,8	2636,6	2635,2	2679,5	2681,9	2626,5	2678,9
tai040	min	<b>2784</b>	2810	2793	2786	2825	2810	2839	2838	2799	2847
	μ.τ.	2821,1	2834,8	<b>2815,6</b>	2820,3	2837,5	2840,0	2858,5	2872,3	2824,9	2872,4

προβλήματα 50x10

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai041	min	3265	3306	3283	3247	3303	3226	3389	3372	<b>3206</b>	3379
	μ.τ.	3301,2	3333,6	3308,7	3318	3333,7	3291,1	3407,2	3413,4	<b>3287,8</b>	3408,1
tai042	min	3135	3186	3150	<b>3103</b>	3189	3125	3271	3267	3109	3220
	μ.τ.	3183,1	3213,5	3179,1	3184,2	3212,5	3186,5	3278,7	3294,9	<b>3142,4</b>	3276
tai043	min	3151	3159	3155	3143	3172	3145	3242	3263	<b>3097</b>	3284
	μ.τ.	3191,3	3207,6	3173	3198,1	3207,1	3199,3	3279,2	3286,2	<b>3172,1</b>	3308
tai044	min	<b>3222</b>	3280	3310	3274	3329	3263	3386	3411	3256	3364
	μ.τ.	3313,7	3329,6	3324,1	3311	3355,9	3300,7	3402,4	3422,3	<b>3297,5</b>	3408,9
tai045	min	3247	3260	3241	3256	3283	<b>3232</b>	3341	3323	3236	3332
	μ.τ.	3294,1	3316,5	<b>3276,2</b>	3296,5	3313,7	3290,4	3366,1	3376,9	3277,2	3367,9
tai046	min	3228	3271	3238	3264	3288	3249	3364	3377	<b>3213</b>	3351

	μ.τ.	3286,7	3316,7	3280,6	3291,5	3308,5	3307,8	3390,5	3402,9	<b>3255,9</b>	3379,1
<b>tai047</b>	min	3309	3316	3301	3328	3330	3309	3404	3439	<b>3289</b>	3419
	μ.τ.	3344,2	3370,9	3338,1	3353,9	3366,2	3346	3434,6	3455,8	<b>3315,7</b>	3437,9
<b>tai048</b>	min	3274	3258	3253	3267	3284	3237	3310	3318	<b>3216</b>	3348
	μ.τ.	3291	3299,8	3271	3286,6	3302	3295,4	3365	3374,7	<b>3252</b>	3380
<b>tai049</b>	min	3150	3177	3146	3144	3191	<b>3101</b>	3305	3283	3135	3283
	μ.τ.	3192,6	3214	3178,4	3200,3	3220,4	3188,3	3317,1	3311,4	<b>3174,2</b>	3308,9
<b>tai050</b>	min	3319	3326	3320	3310	3329	3310	3422	3406	<b>3279</b>	3420
	μ.τ.	3265	3306	3283	3247	3303	3226	3389	3372	<b>3206</b>	3379

προβλήματα 50x20

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
<b>tai051</b>	min	<b>4204</b>	4300	4207	4248	4306	4275	4366	4392	4213	4370
	μ.τ.	4285,1	4324,6	4268,4	4285,5	4336,7	4309,8	4414,9	4428,4	<b>4254,9</b>	4428
<b>tai052</b>	min	4116	4136	4123	4123	4148	4116	4224	4224	<b>4038</b>	4169
	μ.τ.	4145,7	4180,9	<b>4136,1</b>	4155,3	4188,5	4143,7	4250,5	4274,6	4141,8	4247,6
<b>tai053</b>	min	4073	4055	4061	4043	4118	4039	4196	4208	<b>4008</b>	4200
	μ.τ.	4101,9	4126,3	4084,8	4106,4	4142,4	4094,3	4220,4	4235,4	<b>4059,9</b>	4230,3
<b>tai054</b>	min	4115	4131	4122	4080	4129	4086	4245	4246	<b>4045</b>	4266
	μ.τ.	4155,4	4195,1	4145,4	4153,5	4189,5	4131,3	4279,9	4277,9	<b>4110,1</b>	4290,7
<b>tai055</b>	min	4053	4106	4047	4045	4125	4047	4202	4228	<b>4040</b>	4199
	μ.τ.	4103	4162,1	4095,8	4118,2	4149,7	4090,9	4236,2	4255,6	<b>4075,8</b>	4250,3
<b>tai056</b>	min	4088	4114	4105	4123	4118	4061	4211	4240	<b>4038</b>	4240
	μ.τ.	4137,8	4163,9	4124,5	4143,1	4171,7	4123,5	4257,6	4276,5	<b>4087,1</b>	4267,9
<b>tai057</b>	min	4136	4164	4121	4133	4162	4101	4270	4245	<b>4055</b>	4271
	μ.τ.	4167,2	4197	<b>4134,4</b>	4173,7	4194,1	4153,3	4296,1	4305,6	4138,6	4305,2
<b>tai058</b>	min	4157	4126	4115	4131	4151	4088	4255	4208	<b>4072</b>	4262
	μ.τ.	4184,9	4205,1	4143,3	4179,1	4185,2	4135,6	4282,1	4284,7	<b>4103,3</b>	4295,5
<b>tai059</b>	min	4185	4202	4133	4137	4197	4133	4274	4289	<b>4125</b>	4271
	μ.τ.	4195,7	4222,9	4165,9	4200,5	4222,4	4183,8	4310,5	4329,5	<b>4159,2</b>	4314,6
<b>tai060</b>	min	4182	4231	4112	4177	4208	4128	4290	4324	<b>4107</b>	4331
	μ.τ.	4231,9	4255,8	4171,9	4222,8	4255,6	4215,3	4331,4	4355,8	<b>4163</b>	4359,2

προβλήματα 100x5

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
<b>tai061</b>	min	5524	5544	5514	5540	5543	<b>5505</b>	5574	5598	5525	5574
	μ.τ.	5540,5	5555,4	<b>5533,5</b>	5555,9	5558,1	5545,5	5618	5630,7	5554,9	5618,2
<b>tai062</b>	min	5316	5316	5316	5304	5307	<b>5302</b>	5368	5400	5318	5401



	μ.τ.	5354,4	5352,9	<b>5328,7</b>	5348,7	5360	5344	5416,7	5442,8	5397,9	5430,8
<b>tai063</b>	min	5272	5289	5243	5252	5271	<b>5231</b>	5339	5352	5241	5313
	μ.τ.	5290,3	5303,4	<b>5265,5</b>	5287,8	5293,6	5278,9	5369,9	5372,6	5298,9	5364,8
<b>tai064</b>	min	5079	5063	5065	5079	5085	<b>5035</b>	5159	5170	5035	5123
	μ.τ.	5092,4	5094,8	5079,9	5093	5106,2	5078,2	5185,6	5191,2	<b>5072,9</b>	5171,5
<b>tai065</b>	min	<b>5305</b>	5331	5309	5312	5325	5311	5406	5411	5433	5439
	μ.τ.	5351,4	5367,1	5337,4	5354,9	5369,1	<b>5332,5</b>	5440,4	5446,2	5433	5453,8
<b>tai066</b>	min	5184	5183	5174	5176	5190	<b>5160</b>	5258	5248	5174	5270
	μ.τ.	5201,7	5207,1	5188,4	5200,4	5212,1	<b>5188,3</b>	5281,7	5283,9	5219,7	5291,8
<b>tai067</b>	min	5326	<b>5307</b>	5308	5324	5321	5308	5390	5371	5310	5379
	μ.τ.	5347,9	5340,5	<b>5322,5</b>	5343,9	5346,6	5345,7	5420	5419,1	5349,9	5410
<b>tai068</b>	min	5163	5200	5164	5173	5192	5171	5254	5270	<b>5141</b>	5273
	μ.τ.	5205,1	5217	5194	5206,6	5224,5	<b>5192,3</b>	5290,3	5303,1	5214,4	5296,7
<b>tai069</b>	min	5541	5520	5514	5525	5544	5514	5599	5599	<b>5508</b>	5619
	μ.τ.	5559,5	5553,3	<b>5536,1</b>	5557,2	5568,4	5549,5	5621,8	5623,5	5559,1	5631,3
<b>tai070</b>	min	5408	5407	5389	5391	5396	<b>5372</b>	5490	5505	5392	5488
	μ.τ.	5430,4	5437,8	<b>5410,7</b>	5426,8	5437	5430	5521,6	5528,6	5427,1	5531,2

προβλήματα 100x10

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
<b>tai071</b>	min	6138	6220	6116	6126	6169	<b>5991</b>	6314	6311	6075	6281
	μ.τ.	6195,7	6245,5	6162,4	6198,3	6223	<b>6101,8</b>	6350,3	6366,1	6215,3	6341,4
<b>tai072</b>	min	5741	5776	5730	5744	5815	<b>5646</b>	5918	5914	5829	5942
	μ.τ.	5794,5	5828,6	5769,3	5797,4	5832,7	<b>5714,7</b>	5958	5973,6	5894,9	5969,7
<b>tai073</b>	min	5997	6020	5964	5977	5973	<b>5893</b>	6156	6134	6001	6111
	μ.τ.	6032,9	6051,9	5990,9	6020,8	6056,6	<b>5944,1</b>	6182,4	6178,2	6091,2	6165,3
<b>tai074</b>	min	6186	6268	6191	6200	6271	<b>6093</b>	6389	6308	6148	6398
	μ.τ.	6252,2	6304,5	6235,5	6261,9	6299,4	<b>6168,4</b>	6420,9	6416,2	6329,6	6428,3
<b>tai075</b>	min	5864	5934	5886	5894	5929	<b>5828</b>	6078	6065	5863	6070
	μ.τ.	5947,9	5973,6	5918,3	5965,8	5991,8	<b>5873,5</b>	6102,6	6112,2	6037,4	6105,4
<b>tai076</b>	min	3223	3240	3229	3213	3256	3247	3282	3297	<b>3197</b>	3342
	μ.τ.	3252,8	3265,1	3250,5	3251,6	3279,8	3291,6	3340,9	3354,9	<b>3256,6</b>	3363,5
<b>tai077</b>	min	5891	5941	5879	5882	5923	5819	6045	6020	<b>5813</b>	6015
	μ.τ.	5932,4	5985,3	5908,3	5940,8	5969,6	5860,2	6080,5	6078,7	<b>5989,2</b>	6062,7
<b>tai078</b>	min	5984	6044	5932	5962	6020	5918	6100	6159	<b>5891</b>	6132
	μ.τ.	6026,4	6055,4	5978,8	6016	6041,5	<b>5955,7</b>	6162,3	6173,6	6077,1	6162,4
<b>tai079</b>	min	6180	6213	6169	6203	6213	<b>6084</b>	6315	6272	6182	6344
	μ.τ.	6214,6	6251	6202,7	6235,2	6238,9	<b>6142,9</b>	6350,7	6367,7	6275,2	6370,7
<b>tai080</b>	min	6120	6177	6101	6121	6180	<b>6101</b>	6266	6290	6120	6271
	μ.τ.	6171,9	6209,8	6144,1	6184,4	6216,2	<b>6133,4</b>	6304,1	6310,1	6257,3	6309,7

προβλήματα 100x20

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai081	min	6963	7055	6995	7017	7086	<b>6860</b>	7191	7213	6902	7170
	μ.τ.	7041,8	7103,4	7028,6	7073,5	7113,7	<b>6941,1</b>	7233,8	7243,7	7041,9	7234,6
tai082	min	6916	7072	6989	7035	7069	6855	7174	7171	<b>6844</b>	7182
	μ.τ.	7024,9	7091,7	7021	7081,2	7100,9	<b>6909,6</b>	7209	7250,3	7058,2	7233,2
tai083	min	6998	7085	7017	7079	7112	6886	7187	7189	<b>6880</b>	7247
	μ.τ.	7068,7	7129,1	7042,3	7109,6	7135,3	<b>6960,7</b>	7250,2	7264	7054,9	7270,2
tai084	min	7016	7078	6992	6980	7060	6874	7113	7193	<b>6777</b>	7161
	μ.τ.	7049,5	7109,8	7026,4	7051,8	7097,6	<b>6934,2</b>	7206	7248,5	6981,9	7236
tai085	min	7092	7119	7025	7089	7097	6933	7194	7262	<b>6856</b>	7247
	μ.τ.	7130,8	7167,9	7061,4	7138,9	7146,7	<b>6990,9</b>	7280	7288,6	7182,7	7276,7
tai086	min	7106	7168	7086	7056	7189	<b>6936</b>	7324	7238	6963	7307
	μ.τ.	7179,5	7208,4	7126,8	7151,5	7213,5	<b>7031,4</b>	7344,1	7334,6	7214,7	7341,7
tai087	min	7134	7122	7095	7097	7177	<b>6951</b>	7254	7243	7143	7285
	μ.τ.	7167,9	7195,8	7139,6	7164,5	7207,4	<b>7049,7</b>	7321,3	7326	7256,5	7324,2
tai088	min	7256	7255	7133	7219	7269	<b>7084</b>	7312	7368	7212	7367
	μ.τ.	7275,2	7301,6	7205,9	7263,9	7302,7	<b>7158,3</b>	7399,2	7411,2	7336,1	7402,1
tai089	min	7057	7065	7016	7089	7133	<b>6889</b>	7225	7259	6911	7226
	μ.τ.	7120	7143,9	7062,1	7126,2	7178,6	<b>7011,1</b>	7281,9	7296,8	7099,8	7290
tai090	min	7109	7172	7125	7127	7164	7003	7348	7326	<b>6953</b>	7308
	μ.τ.	7173,2	7232	7149,6	7201	7220,2	<b>7108,1</b>	7385,1	7377,8	7226,5	7367,7

## ΠΑΡΑΡΤΗΜΑ II

Μικρότερη τιμή του makespan και η μέση τιμή δέκα επαναλήψεων για τις δέκα στρατηγικές του DE στα δεδομένα του Taillard για το FSSP με παραμέτρους  $F=0.3$ ,  $CR=0.9$ ,  $NP=150$  και  $GMAX=50$

min=μικρότερη τιμή

μ.τ.=μέση τιμή makespan

προβλήματα 20x5

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai001	min	<b>1297</b>	<b>1297</b>	<b>1297</b>	<b>1297</b>	1316	1320	1306	<b>1297</b>	<b>1297</b>	1303
	μ.τ.	1312,7	1319,8	<b>1312,6</b>	1314,4	1322,6	1331,3	1322,7	1324,3	1316,5	1319,7
tai002	min	<b>1368</b>	<b>1368</b>	<b>1368</b>	<b>1368</b>	1372	1373	1377	1381	1373	1371
	μ.τ.	<b>1377,2</b>	1378,8	1377,3	1378,3	1380,4	1385,2	1383,4	1384,8	1379,4	1381,0
tai003	min	<b>1140</b>	1158	1162	1149	1154	1156	1154	1159	1140	1162
	μ.τ.	1153,1	1178,2	1177,4	1167,9	1179,9	1182,1	1185,1	1182,3	<b>1152,6</b>	1182,8
tai004	min	1351	1372	1387	1356	1387	1370	1384	1388	<b>1344</b>	1385
	μ.τ.	<b>1374,7</b>	1398,4	1400,8	1380,9	1401,0	1397,0	1399,9	1406,2	1378,7	1400,6
tai005	min	1252	1289	1284	1289	1284	1291	1301	1317	<b>1250</b>	1303
	μ.τ.	<b>1286,7</b>	1308,0	1298,0	1304,4	1305,9	1321,2	1317,2	1320,3	1297,9	1313,7
tai006	min	1225	1244	1234	1224	<b>1254</b>	1243	1241	1266	1242	1271
	μ.τ.	1249,4	1269,4	<b>1262,7</b>	1255,9	1273,4	1274,9	1273,9	1280,8	1268,4	1279,0
tai007	min	1251	1264	1265	1263	1257	1258	1269	1275	<b>1254</b>	1265
	μ.τ.	1266,3	1283,7	1278,3	1277,5	1283,5	1288,9	1289,4	1291,5	<b>1274,4</b>	1291,0
tai008	min	<b>1258</b>	1281	1263	1262	1296	1266	1283	1283	1266	1288
	μ.τ.	<b>1281,4</b>	1296,9	1296,7	1285,2	1308,0	1298,1	1316,2	1307,5	1286,4	1314,9
tai009	min	<b>1274</b>	1304	1278	1276	1298	1274	1299	1277	1282	1305
	μ.τ.	<b>1292,3</b>	1314,7	1307,7	1293,8	1314,0	1324,0	1311,6	1312,2	1317,0	1321,7
tai010	min	<b>1137</b>	1179	1177	1158	1177	1165	1188	1177	1165	1185
	μ.τ.	<b>1173,9</b>	1196,7	1189,1	1179,1	1191,0	1193,3	1196,4	1197,8	1180,7	1205,9

προβλήματα 20x10

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai011	min	<b>1652</b>	1704	1681	1669	1713	1687	1718	1732	1659	1722
	μ.τ.	<b>1678,2</b>	1731,1	1733,0	1701,5	1736,2	1727,9	1741,1	1744,9	1694,1	1743,6

tai012	min	1758	1798	1779	1764	1791	<b>1755</b>	1776	1795	1757	1808
	μ.τ.	1786,6	1815,1	1803,1	1795,0	1817,9	1810,4	1816,5	1827,2	<b>1783,5</b>	1828,0
tai013	min	1548	1607	1623	1571	<b>1493</b>	1582	1607	1615	1575	1632
	μ.τ.	1597,8	1638,0	1644,9	1625,4	<b>1524,3</b>	1638,6	1653,5	1660,7	1623,5	1657,9
tai014	min	1456	1505	1490	1499	1493	<b>1456</b>	1478	1522	1471	1518
	μ.τ.	<b>1494,0</b>	1527,5	1517,2	1521,4	1524,3	1504,2	1522,9	1543,1	1505,2	1544,9
tai015	min	<b>1506</b>	1527	1561	1538	1569	1550	1584	1550	1526	1559
	μ.τ.	<b>1536,2</b>	1577,7	1582,4	1566,9	1588,8	1584,8	1598,9	1590,0	1558,8	1595,4
tai016	min	<b>1466</b>	1502	1481	1479	1510	1485	1492	1501	1477	1528
	μ.τ.	<b>1483,5</b>	1517,7	1518,5	1499,3	1524,9	1515,8	1515,6	1525,1	1499,8	1541,0
tai017	min	<b>1530</b>	1586	1585	1542	1551	1570	1600	1598	1557	1606
	μ.τ.	<b>1575,8</b>	1609,0	1602,9	1584,1	1601,7	1590,4	1616,6	1620,0	1590,5	1621,7
tai018	min	<b>1620</b>	1678	1653	1628	1660	1638	1700	1669	1637	1682
	μ.τ.	<b>1645,6</b>	1694,9	1691,7	1668,9	1682,5	1677,3	1708,1	1696,4	1667,6	1709,6
tai019	min	<b>1655</b>	1704	1673	1669	1694	1679	1707	1697	1661	1705
	μ.τ.	<b>1682,4</b>	1718,2	1701,6	1698,1	1710,5	1714,5	1720,0	1715,6	1700,1	1723,9
tai020	min	1663	1712	1700	<b>1653</b>	1712	1687	1698	1713	1661	1710
	μ.τ.	1698,5	1727,1	1713,3	1702,0	1737,0	1708,0	1729,3	1741,9	<b>1698,0</b>	1733,8

προβλήματα 20x20

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai021	min	<b>2357</b>	2456	2446	2394	2426	2409	2431	2451	2406	2463
	μ.τ.	<b>2407,0</b>	2473,8	2462,4	2450,7	2465,8	2447,9	2468,8	2476,6	2441,4	2486,9
tai022	min	<b>2192</b>	2259	2252	2228	2241	2214	2247	2256	2208	2249
	μ.τ.	<b>2227,9</b>	2276,7	2269,3	2256,8	2277,8	2252,1	2283,8	2287,3	2244,8	2299,3
tai023	min	<b>2405</b>	2463	2467	2465	2462	2425	2434	2495	2405	2491
	μ.τ.	<b>2441,7</b>	2476,3	2490,7	2483,1	2487,7	2484,0	2488,3	2507,6	2451,9	2506,7
tai024	min	<b>2296</b>	2367	2394	2348	2346	2335	2392	2403	2351	2379
	μ.τ.	<b>2345,7</b>	2407,4	2406,2	2374,2	2401,5	2390,5	2419,4	2425,8	2368,8	2413,6
tai025	min	<b>2374</b>	2424	2420	2399	2416	2394	2430	2440	2378	2435
	μ.τ.	<b>2408,7</b>	2459,2	2458,8	2424,5	2457,7	2438,8	2462,1	2462,2	2415,8	2469,2
tai026	min	2314	2351	2362	2341	2360	2352	2378	2374	<b>2293</b>	2382
	μ.τ.	<b>2341,9</b>	2378,3	2378,2	2373,9	2396,8	2380,9	2394,0	2390,2	2342,2	2404,9
tai027	min	<b>2343</b>	2412	2405	2406	2425	2411	2418	2416	2374	2430
	μ.τ.	<b>2397,5</b>	2436,2	2426,8	2420,4	2443,7	2438,9	2443,3	2444,5	2400,7	2449,7
tai028	min	2278	2354	2321	2318	2317	2315	2306	2360	<b>2289</b>	2358
	μ.τ.	<b>2311,3</b>	2373,1	2362,6	2343,3	2373,3	2364,5	2366,8	2385,7	2323,5	2386,0
tai029	min	<b>2296</b>	2397	2375	2370	2397	2355	2395	2409	2355	2421
	μ.τ.	<b>2378,2</b>	2425,3	2413,3	2406,1	2431,5	2415,5	2427,1	2434,3	2411,8	2447,0
tai030	min	2330	2352	2334	2330	2371	2291	2376	2356	<b>2277</b>	2346

	μ.τ.	2351,4	2381,6	2381,3	2354,5	2386,6	2384,8	2400,3	2402,4	<b>2344,1</b>	2398,4
--	------	--------	--------	--------	--------	--------	--------	--------	--------	---------------	--------

προβλήματα 50x5

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai031	min	2752	2752	2758	2765	2787	<b>2752</b>	2799	2815	2772	2804
	μ.τ.	2791,6	2792,7	2796,7	2791,3	2812,1	<b>2785,2</b>	2824,1	2837,9	2793,3	2834,2
tai032	min	2949	2952	2975	2951	2976	<b>2903</b>	2940	2963	2957	2958
	μ.τ.	2974,1	2986,0	2984,3	2981,4	2991,6	<b>2940,2</b>	2994,5	3002,2	2993,0	3002,7
tai033	min	2672	2722	2701	2725	2720	<b>2660</b>	2725	2759	2699	2736
	μ.τ.	2735,1	2748,4	2731	2741,4	2742,4	<b>2702,4</b>	2753,9	2772,4	2747	2765,3
tai034	min	2863	2868	2864	2884	2855	<b>2843</b>	2875	2866	2859	2898
	μ.τ.	2890,3	2899,5	2890,9	2901,9	2905,2	<b>2864,7</b>	2916,0	2925,3	2890,3	2916,8
tai035	min	2922	2939	2924	2919	2926	<b>2903</b>	2937	2944	2942	2947
	μ.τ.	2951,4	2954,3	2948,9	2952,0	2953,3	<b>2929,1</b>	2972,8	2970,1	2967,1	2981,2
tai036	min	2921	2939	2943	2935	2940	<b>2888</b>	2960	2948	2931	2941
	μ.τ.	2948,7	2965,9	2962	2959,2	2963,3	<b>2913,1</b>	2979,2	2980,5	2970,3	2978,8
tai037	min	2837	2843	2839	2844	2851	<b>2807</b>	2834	2847	2857	2851
	μ.τ.	2861,1	2867,0	2858,9	2871,4	<b>2829,6</b>	2876,1	2883,7	2876,9	2879,3	2879,3
tai038	min	2799	2786	2809	2779	2772	<b>2727</b>	2823	2814	2812	2855
	μ.τ.	2830,1	2834,7	2838,3	2826,7	2835,8	<b>2788,8</b>	2865,7	2857,3	2848,7	2867,5
tai039	min	2644	2692	2638	2647	2684	<b>2618</b>	2696	2683	2664	2684
	μ.τ.	2688,0	2704,3	2679,5	2684,1	2701,6	<b>2654,4</b>	2711,7	2713,4	2702,0	2718,1
tai040	min	2854	2866	2848	2854	2882	<b>2829</b>	2887	2870	2852	2850
	μ.τ.	2876,2	2897,1	2885,0	2876,9	2891,5	<b>2864,3</b>	2907,9	2905,6	2894,2	2893,8

προβλήματα 50x10

στρατηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
tai041	min	3361	3347	3386	3400	3382	<b>3265</b>	3407	3443	3366	3428
	μ.τ.	3407,4	3429	3418,1	3435,9	3433,5	<b>3345,2</b>	3453,7	3470,2	3438,3	3460,5
tai042	min	3258	3267	3260	3272	3278	<b>3160</b>	3281	3293	3289	3283
	μ.τ.	3290,6	3305,2	3276,2	3295	3311,8	<b>3207,5</b>	3323	3317,3	3328,5	3337
tai043	min	3275	3263	3230	3288	3285	<b>3190</b>	3297	3271	3290	3290
	μ.τ.	3312,9	3321,4	3296,7	3316,8	3318,6	<b>3227,6</b>	3349,4	3333	3320,2	3341,6
tai044	min	<b>3415</b>	3400	3370	3391	3380	<b>3262</b>	3423	3430	3362	3437
	μ.τ.	3431,8	3429,4	3416,1	3430,1	3430	<b>3338,8</b>	3467,6	3472,8	3445,3	3452
tai045	min	3354	3344	3347	3372	3400	<b>3282</b>	3383	3400	3386	3377
	μ.τ.	3384,7	3409	3399,8	3401,5	3416,3	<b>3318,9</b>	3430,3	3434,2	3422,7	3419,9
tai046	min	3336	3343	3360	3337	3376	<b>3254</b>	3415	3382	3322	3402

	μ.τ.	3386	3400,3	3382,5	3410,2	3405,8	<b>3307,5</b>	3441,4	3422,8	3405,3	3438,9
<b>tai047</b>	min	3429	3442	3403	3412	3413	<b>3344</b>	3448	3438	3444	3459
	μ.τ.	3443,8	3457,4	3441,4	3443,1	3454	<b>3378,8</b>	3479,6	3478,6	3472,9	3484,5
<b>tai048</b>	min	3365	3367	3337	3375	3373	<b>3259</b>	3372	3365	3354	3347
	μ.τ.	3383,9	3403,1	3374,2	3392,9	3407,3	<b>3313,2</b>	3415,8	3419,7	3413,7	3416,8
<b>tai049</b>	min	3271	3321	3244	3280	3310	<b>3194</b>	3325	3317	3265	3335
	μ.τ.	3325,2	3343,5	3301,5	3312,6	3336,6	<b>3236,1</b>	3359,1	3372,2	3328,7	3359,1
<b>tai050</b>	min	3378	3374	3389	3387	3439	<b>3318</b>	3403	3449	3430	3435
	μ.τ.	3439	3451,1	3437,3	3439,8	3466,3	<b>3366,7</b>	3472,1	3483,5	3459,5	3476,7

προβλήματα 50x20

στратηγική πρόβλημα		1	2	3	4	5	6	7	8	9	10
<b>tai051</b>	min	3361	3347	3386	3400	3382	<b>3265</b>	3407	3443	3366	3428
	μ.τ.	3407,4	3429	3418,1	3435,9	3433,5	<b>3345,2</b>	3453,7	3470,2	3438,3	3460,5
<b>tai052</b>	min	3258	3267	3260	3272	3278	<b>3160</b>	3281	3293	3289	3283
	μ.τ.	3290,6	3305,2	3276,2	3295	3311,8	<b>3207,5</b>	3323	3317,3	3328,5	3337
<b>tai053</b>	min	3275	3263	3230	3288	3285	<b>3190</b>	3297	3271	3290	3290
	μ.τ.	3312,9	3321,4	3296,7	3316,8	3318,6	<b>3227,6</b>	3349,4	3333	3320,2	3341,6
<b>tai054</b>	min	<b>3415</b>	3400	3370	3391	3380	<b>3262</b>	3423	3430	3362	3437
	μ.τ.	3431,8	3429,4	3416,1	3430,1	3430	<b>3338,8</b>	3467,6	3472,8	3445,3	3452
<b>tai055</b>	min	3354	3344	3347	3372	3400	<b>3282</b>	3383	3400	3386	3377
	μ.τ.	3384,7	3409	3399,8	3401,5	3416,3	<b>3318,9</b>	3430,3	3434,2	3422,7	3419,9
<b>tai056</b>	min	3336	3343	3360	3337	3376	<b>3254</b>	3415	3382	3322	3402
	μ.τ.	3386	3400,3	3382,5	3410,2	3405,8	<b>3307,5</b>	3441,4	3422,8	3405,3	3438,9
<b>tai057</b>	min	3429	3442	3403	3412	3413	<b>3344</b>	3448	3438	3444	3459
	μ.τ.	3443,8	3457,4	3441,4	3443,1	3454	<b>3378,8</b>	3479,6	3478,6	3472,9	3484,5
<b>tai058</b>	min	3365	3367	3337	3375	3373	<b>3259</b>	3372	3365	3354	3347
	μ.τ.	3383,9	3403,1	3374,2	3392,9	3407,3	<b>3313,2</b>	3415,8	3419,7	3413,7	3416,8
<b>tai059</b>	min	3271	3321	3244	3280	3310	<b>3194</b>	3325	3317	3265	3335
	μ.τ.	3325,2	3343,5	3301,5	3312,6	3336,6	<b>3236,1</b>	3359,1	3372,2	3328,7	3359,1
<b>tai060</b>	min	3378	3374	3389	3387	3439	<b>3318</b>	3403	3449	3430	3435
	μ.τ.	3439	3451,1	3437,3	3439,8	3466,3	<b>3366,7</b>	3472,1	3483,5	3459,5	3476,7

## ΠΑΡΑΡΤΗΜΑ ΙΙΙ

ελάχιστες τιμές makespan μετά από 10 επαναλήψεις του κάθε αλγορίθμου για το κάθε πρόβλημα

πρόβλημα	upper bound	DE	DELS	DEILS	DEVNS	DESA
tai001	1278	1297	1278	1282	1278	1278
tai002	1359	1366	1360	1359	1359	1360
tai003	1081	1130	1089	1098	1089	1098
tai004	1293	1351	1302	1293	1293	1305
tai005	1235	1276	1243	1244	1244	1244
tai006	1195	1226	1195	1195	1210	1195
tai007	1239	1259	1251	1239	1239	1239
tai008	1206	1255	1206	1207	1206	1206
tai009	1230	1285	1237	1236	1232	1252
tai010	1108	1172	1109	1120	1120	1109
tai011	1582	1681	1591	1607	1599	1593
tai012	1659	1751	1681	1685	1678	1692
tai013	1496	1573	1523	1524	1512	1530
tai014	1377	1482	1398	1395	1386	1417
tai015	1419	1540	1438	1439	1439	1425
tai016	1397	1487	1412	1408	1401	1422
tai017	1484	1555	1493	1500	1486	1504
tai018	1538	1633	1569	1554	1557	1569
tai019	1593	1664	1611	1620	1614	1622
tai020	1591	1687	1606	1618	1611	1620
tai021	2297	2412	2330	2332	2305	2335
tai022	2099	2209	2118	2109	2123	2139
tai023	2326	2426	2358	2354	2342	2364
tai024	2223	2347	2249	2249	2234	2264
tai025	2291	2317	2310	2314	2313	2321
tai026	2226	2384	2253	2257	2247	2265
tai027	2273	2279	2311	2297	2298	2322
tai028	2200	2364	2231	2241	2217	2242
tai029	2237	2364	2247	2257	2253	2249
tai030	2178	2297	2217	2202	2214	2232

<b>tai031</b>	<b>2724</b>	2740	2724	2724	2724	2724
<b>tai032</b>	<b>2836</b>	2868	2849	2848	2848	2882
<b>tai033</b>	<b>2621</b>	2644	2625	2624	2624	2645
<b>tai034</b>	<b>2751</b>	2821	2759	2765	2762	2786
<b>tai035</b>	<b>2863</b>	2891	2864	2864	2864	2887
<b>tai036</b>	<b>2829</b>	2867	2831	2835	2832	2844
<b>tai037</b>	<b>2725</b>	2771	2736	2736	2735	2771
<b>tai038</b>	<b>2683</b>	2733	2704	2700	2699	2706
<b>tai039</b>	<b>2554</b>	2595	2564	2564	2561	2591
<b>tai040</b>	<b>2782</b>	2792	2782	2782	2782	2784
<b>tai041</b>	<b>3037</b>	3229	3114	3122	3086	3199
<b>tai042</b>	<b>2911</b>	3113	2964	3013	2970	3088
<b>tai043</b>	<b>2871</b>	3098	2958	2992	2955	3071
<b>tai044</b>	<b>3067</b>	3256	3101	3141	3124	3202
<b>tai045</b>	<b>3011</b>	3194	3087	3108	3100	3193
<b>tai046</b>	<b>3021</b>	3155	3079	3117	3086	3155
<b>tai047</b>	<b>3124</b>	3278	3156	3197	3156	3231
<b>tai048</b>	<b>3048</b>	3159	3084	3106	3082	3171
<b>tai049</b>	<b>2910</b>	3120	2976	3005	2980	3045
<b>tai050</b>	<b>3100</b>	3254	3159	3180	3152	3241
<b>tai051</b>	<b>3886</b>	4112	3993	4038	3968	4097
<b>tai052</b>	<b>3733</b>	4081	3881	3924	3853	3986
<b>tai053</b>	<b>3672</b>	4013	3803	3872	3825	3915
<b>tai054</b>	<b>3755</b>	4034	3887	3924	3878	3973
<b>tai055</b>	<b>3648</b>	3982	3813	3864	3779	3907
<b>tai056</b>	<b>3719</b>	4028	3864	3891	3835	3960
<b>tai057</b>	<b>3730</b>	4066	3860	3921	3872	3971
<b>tai058</b>	<b>3737</b>	4040	3872	3925	3839	3968
<b>tai059</b>	<b>3772</b>	4090	3905	3962	3894	3984
<b>tai060</b>	<b>3791</b>	4040	3909	3936	3931	3997
<b>tai061</b>	<b>5493</b>	5495	5493	5493	5493	5495
<b>tai062</b>	<b>5274</b>	5290	5284	5284	5284	5284
<b>tai063</b>	<b>5175</b>	5225	5205	5213	5207	5217
<b>tai064</b>	<b>5018</b>	5044	5021	5018	5021	5025
<b>tai065</b>	<b>5250</b>	5311	5255	5255	5255	5258
<b>tai066</b>	<b>5135</b>	5161	5137	5473	5144	5145
<b>tai067</b>	<b>5247</b>	5281	5259	5125	5260	5288



<b>tai068</b>	<b>5094</b>	5137	5124	5264	5115	5130
<b>tai069</b>	<b>5448</b>	5504	5464	5146	5467	5492
<b>tai070</b>	<b>5328</b>	5370	5334	5342	5337	5342
<b>tai071</b>	<b>5776</b>	6015	5857	5925	5863	5968
<b>tai072</b>	<b>5362</b>	5614	5441	5482	5436	5546
<b>tai073</b>	<b>5679</b>	5868	5757	5788	5737	5833
<b>tai074</b>	<b>5820</b>	6052	5914	5982	5926	6049
<b>tai075</b>	<b>5491</b>	5783	5579	5686	5610	5734
<b>tai076</b>	<b>5308</b>	5498	5370	5419	5369	5478
<b>tai077</b>	<b>5600</b>	5784	5690	5722	5664	5754
<b>tai078</b>	<b>5640</b>	5838	5721	5748	5725	5824
<b>tai079</b>	<b>5891</b>	6054	5972	5981	5962	6022
<b>tai080</b>	<b>5856</b>	6027	5903	5943	5903	5997
<b>tai081</b>	<b>6330</b>	6747	6533	6663	6555	6741
<b>tai082</b>	<b>6320</b>	6768	6530	6632	6536	6734
<b>tai083</b>	<b>6364</b>	6850	6610	6700	6596	6807
<b>tai084</b>	<b>6331</b>	6782	6567	6653	6512	6780
<b>tai085</b>	<b>6405</b>	6906	6619	6701	6641	6842
<b>tai086</b>	<b>6487</b>	6910	6674	6743	6697	6887
<b>tai087</b>	<b>6379</b>	6930	6586	6859	6612	6849
<b>tai088</b>	<b>6514</b>	6979	6757	6728	6781	6955
<b>tai089</b>	<b>6386</b>	6816	6598	6836	6609	6815
<b>tai090</b>	<b>6534</b>	6946	6707	6836	6692	6876
<b>tai091</b>	<b>10872</b>	11139	10980	11014	10954	11057
<b>tai092</b>	<b>10500</b>	11179	10654	10807	10678	10901
<b>tai093</b>	<b>10956</b>	11173	11087	11154	11088	11214
<b>tai094</b>	<b>10893</b>	11129	10955	10995	10955	11024
<b>tai095</b>	<b>10537</b>	10883	10692	10829	10694	10910
<b>tai096</b>	<b>10347</b>	10767	10485	10598	10455	10659
<b>tai097</b>	<b>10882</b>	11218	10979	11081	11005	11130
<b>tai098</b>	<b>10754</b>	11499	10890	10976	10875	11014
<b>tai099</b>	<b>10465</b>	11251	10548	10692	10574	10769
<b>tai100</b>	<b>10727</b>	11077	10825	10927	10835	10999
<b>tai101</b>	<b>11393</b>	12426	11722	11923	11786	12154
<b>tai102</b>	<b>11445</b>	12328	11885	12123	11914	12259
<b>tai103</b>	<b>11522</b>	12354	11881	12153	11942	12304
<b>tai104</b>	<b>11461</b>	12377	11855	12022	11859	12239

<b>tai105</b>	<b>11427</b>	12240	11789	12024	11830	12191
<b>tai106</b>	<b>11368</b>	12509	11778	12023	11824	12174
<b>tai107</b>	<b>11536</b>	12529	11953	12164	12003	12361
<b>tai108</b>	<b>11544</b>	12418	11904	12083	11948	12252
<b>tai109</b>	<b>11424</b>	12388	11815	12066	11799	12183
<b>tai110</b>	<b>11548</b>	12863	11919	12138	11945	12278

## ΠΑΡΑΡΤΗΜΑ IV

Πραγματικά δεδομένα της προς μελέτη εταιρίας.

εργασία	καζάνι	γραμμή συσκευασίας	διάρκεια παραγωγής bulk (λεπτά)	διάρκεια συσκευασίας (λεπτά)	διάρκεια βασικού μικροβιολογικού ελέγχου bulk	διάρκεια βασικού μικροβιολογικού ελέγχου έτοιμου
1	2	1	285	2915	2880	7200
2	2	1	285	2915	2880	7200
3	4	1	323	947	2880	7200
4	4	1	323	947	2880	7200
5	2	2	240	3950	1440	7200
6	2	2	291	3950	1440	7200
7	2	2	426	3950	1440	7200
8	5	2	79	2226	0	0
9	5	2	90	2226	0	0
10	5	2	90	2226	0	0
11	5	2	79	2226	0	0
12	5	2	90	2226	0	0
13	1	3	338	2099	1440	7200
14	1	3	330	3674	1440	7200
15	1	4	252	3060	1440	7200
16	1	4	240	3060	2880	7200
17	3	5	278	233	2880	7200
18	3	5	278	233	2880	7200
19	3	5	209	1749	2880	7200
20	3	5	216	245	2880	7200
21	3	5	216	245	2880	7200
22	2	5	325	441	2880	7200
23	2	5	325	441	2880	7200
24	4	5	312	720	2880	7200
25	4	5	272	720	2880	7200

26	2	5	254	1289	2880	7200
27	2	5	339	1289	2880	7200
28	3	5	334	516	2880	7200
29	4	5	312	720	2880	7200
30	4	5	272	720	2880	7200
31	4	5	272	720	2880	7200
32	1	6	566	1113	1440	7200
33	1	6	378	1113	1440	7200
34	1	6	240	525	1440	7200
35	1	6	491	525	1440	7200
36	1	6	491	820	1440	7200
37	1	6	378	820	1440	7200
38	1	6	491	820	1440	7200
39	1	6	378	820	1440	7200
40	1	7	240	2688	1440	7200
41	1	7	419	2296	1440	7200
42	1	7	240	2688	1440	7200
43	1	7	419	2296	1440	7200
44	1	7	240	2688	1440	7200
45	3	7	339	383	1440	7200
46	3	7	247	383	2880	7200
47	3	7	334	383	2880	7200
48	2	7	306	1065	2880	7200
49	2	7	291	1065	2880	7200
50	2	7	291	1065	2880	7200
51	2	7	306	1065	2880	7200
52	2	7	294	1391	2880	7200
53	2	7	267	1391	2880	7200
54	2	7	285	1391	2880	7200
55	2	7	267	1391	2880	7200