

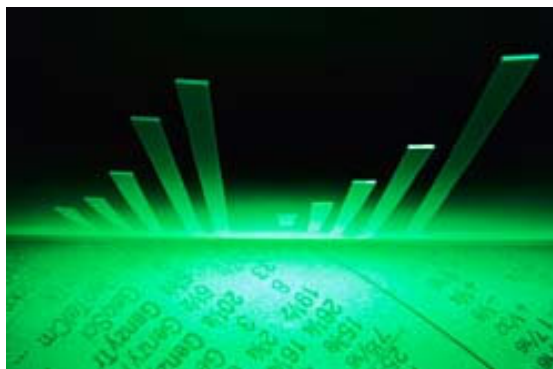


ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
Technical University of Crete

Τμήμα Μηχανικών Παραγωγής και Διοίκησης
Department of Production and Management Engineering

Διπλωματική Εργασία

Διοίκηση - Διαχείριση Έργου
Βελτιστοποίηση Κόστους με Περιορισμό
Αριθμού Εργατών



Ευάγγελος Ε. Παπαδάκης
(Α.Μ. 9811124)

Χανιά, Σεπτέμβριος 2006

Διοίκηση - Διαχείριση Έργου Βελτιστοποίηση Κόστους με Περιορισμό Αριθμού Εργατών

Ευάγγελος Ε. Παπαδάκης

Τμήμα Μηχανικών Παραγωγής και Διοίκησης,
Πολυτεχνείο Κρήτης
(Α.Μ. 9811124)

Επιβλέπων Καθηγητής

Δρ. Ηλίας Κοσματοπούλος,

Τμήμα Μηχανικών Παραγωγής & Διοίκησης
Εργαστήριο Δυναμικών Συστημάτων και Προσομοίωσης

*Αφιερώνεται στους γονείς μου,
Μανώλη και Ιωάννα ...*

Ολοκληρώνοντας αυτό τον κύκλο σπουδών νιώθω την ανάγκη να αναφερθώ σε κάποιους ανθρώπους που έπαιξαν σημαντικό ρόλο σε όλα αυτά τα χρόνια της διαρκούς προσπάθειας για την απόκτηση της γνώσης. Το επιστέγασμα των προσπαθειών αυτών, είναι η διπλωματική αυτή εργασία η οποία απαίτησε αρκετή σκέψη, χρόνο και πειραματισμό και η οποία εκπονήθηκε υπό την επίβλεψη και καθοδήγηση του Δρ.Ηλία Κοσματοπούλου. Είναι πραγματικά σπάνιο να συνδυάζεται ο ανθρώπινος χαρακτήρας με την επιστημονική γνώση και είμαι χαρούμενος που αυτό συνάντησα στο πρόσωπο του καθηγητή μου. Τον ευχαριστώ για την καθ' όλα άψογη συνεργασία μας, τις παραινέσεις και τις συμβουλές του, τη πίστη που έδειξε στο πρόσωπό μου καθώς και την παρουσία του σε κάθε φάση της εργασίας αυτής.

Θα ήθελα να ευχαριστήσω τους συμφοιτητές και συμφοιτήτριές μου που υπήρξαν φίλοι και που αυτά τα καλά χρόνια μοιραστήκαμε καλές και κακές στιγμές, αγωνίες και αναζητήσεις. Και τους ευχαριστώ γιατί είχα τη συμπαράστασή τους και ηθική αλλά και έμπρακτη όταν τη χρειάστηκα. Εύχομαι να ήμουν κι εγώ τόσο σημαντικός γι αυτούς όσο και αυτοί για μένα... Ιδιαίτερα, θα ήθελα να ευχαριστήσω το Θανάση Καλπάκα για την πολύτιμη βοήθειά του στη συγγραφή του τρίτου κεφαλαίου.

Και πάνω απ' όλα ευχαριστώ την οικογένειά μου, από εκεί που ξεκίνησαν όλα. Τον πατέρα μου, Μανώλη τη μητέρα μου Ιωάννα και τον αδερφό μου Μιχάλη. Τους ευχαριστώ για τις αρχές, τις κατευθυντήριες γραμμές και το ήθος που μου έδωσαν και βέβαια για την επικοινωνία, την υποστήριξη και τη συμπαράστασή τους.

Βαγγέλης Παπαδάκης

Περίληψη

Στον επιχειρηματικό κόσμο δίδεται ολοένα και περισσότερη σημασία στη Διοίκηση και Διαχείριση έργου και αυτό γιατί αφ' ενός οι αστάθμητοι και τυχαίοι παράγοντες για την έκβαση ενός έργου πληθαίνουν και περιπλέκονται ακολουθώντας τους ρυθμούς του σύγχρονου τρόπου ζωής και αφετέρου οι συνέπειες για λανθασμένες επιλογές τείνουν να είναι ασυγχώρητες απ' όσο ποτέ άλλοτε. Στο δρόμο της αναζήτησης αυτής για καλύτερη οργάνωση και σχεδιασμό ενός έργου εστιάζουμε την προσοχή μας σε ένα συγκεκριμένο πρόβλημα που ανακύπτει από κακές εκτιμήσεις στη χρηματοροή, στον χρονικό προγραμματισμό και στο προγραμματισμό διάθεσης πόρων. Αναλυτικότερα, εξετάζουμε εκείνες τις περιπτώσεις κατά τις οποίες παρατηρείται μεγάλη συμπίεση χρόνου, πολλές φορές αλόγιστη για την επίτευξη καλύτερων χρόνων ολοκλήρωσης του έργου, που έχουν σαν αποτέλεσμα την συσσώρευση πολλών εργασιών που πρέπει να εκτελεσθούν σε παράλληλο χρόνο και την εκτόξευση του αριθμού των απασχολούμενων εργατικών μονάδων σε αυτό το χρονικό διάστημα. Αυτό παρατηρείται συνήθως στη μέση του κύκλου ζωής ενός έργου και η επιδίωξη μας ήταν να ελέγξουμε τον τρόπο που θα κατανέμονται οι απασχολούμενες εργατικές μονάδες, αποσυμπιέζοντας χρονικά το έργο, για να επιτύχουμε μια πιο ασφαλής από άποψη κόστους και χρηματοροής εναλλακτική λύση. Δημιουργήσαμε λοιπόν, ένα πρόγραμμα το οποίο δέχεται τα δεδομένα ενός οποιοδήποτε έργου που έχει επιδεχθεί μεθόδους βελτιστοποίησης χρονικού προγραμματισμού, αλλά επεξεργάζεται και τις απαιτήσεις του χρήστη για ένα διαφορετικό τρόπο ολοκλήρωσης του έργου και κατόπιν καταλήγει σε προτάσεις. Μέσω ενός αλγορίθμου, ο κώδικας που κατασκευάσαμε τηρεί όλες τις χρονικές προϋποθέσεις και προσπαθεί βήμα με βήμα να εναρμονιστεί με τις απαιτήσεις του χρήστη προσφέροντας διαφορετικές λύσεις χρονικά σε κάθε εφαρμογή του. Χρησιμοποιώντας, πραγματικά δεδομένα από ένα αρκετά δύσκολο κατασκευαστικό έργο δοκιμάσαμε το πρόγραμμά μας με πολλούς ποικίλους και χρονοβόρους πειραματισμούς. Ορισμένες μεταβλητές ρυθμίστηκαν έτσι ώστε το πρόγραμμα να αποδώσει καλύτερα. Τα αποτελέσματα ήταν ενθαρρυντικά από την αρχή και η αποτελεσματικότητα των λύσεων κρίθηκε με βάση τη ποιότητα τους και κατά πόσο μας ικανοποίησαν στις χρονικές και γραφικές απαιτήσεις μας.

Περιεχόμενα

Περιεχόμενα	6
--------------------------	----------

1. Εισαγωγή στη Διοίκηση Διαχείριση Έργου	9
--	----------

1.1 Εισαγωγή	9
---------------------------	----------

1.2 Ο Κύκλος Ζωής του Έργου	10
-----------------------------------	----

1.2.1 Η σύλληψη του Έργου	10
---------------------------------	----

1.2.2 Η Έναρξη του Έργου	11
--------------------------------	----

1.2.3 Η Υλοποίηση και Κατασκευή του Έργου	12
---	----

1.2.4 Ολοκλήρωση και Αποδοχή του Έργου	12
--	----

1.2.5 Λειτουργία και Συντήρηση	13
--------------------------------------	----

1.3 Ο Σχεδιασμός Προγραμματισμός Έργου	13
--	----

1.4 Ο Χρονικός Προγραμματισμός	14
--------------------------------------	----

1.5 Μελέτη Χρηματοροής	14
------------------------------	----

1.6 Το ανθρώπινο Δυναμικό.....	15
--------------------------------	----

1.7 Το φαινόμενο της άκριτης συμπίεσης χρόνου	15
---	----

1.8 Το προς εξέταση πρόβλημα.....	16
-----------------------------------	----

2. Μοντελοποίηση και Επίλυση.....	17
--	-----------

2.1 Εισαγωγή στον Χρονικό Προγραμματισμό.....	17
---	----

2.2 Δομική Ανάλυση Έργου	18
--------------------------------	----

2.3 Δίκτυο Έργου	18
------------------------	----

2.4 Τοξωτό Δίκτυο.....	29
------------------------	----

2.5 Κομβικό Δίκτυο	21
2.6 Χρονικός Προγραμματισμός του Δικτύου	22
2.7 Χρονική Ανάλυση Τοξωτού Δικτύου	22
2.8 Τεχνική Pert	23
2.9 Άμεσο – Έμμεσο Κόστος.....	25
2.10 Βελτιστοποίηση Κόστους του Έργου	26
2.11 Σύνθετα Προβλήματα Προγραμματισμού Έργων.....	27
2.12 Προγραμματισμός Έργου με Περιορισμούς Διαχείρισης Πόρων	28
2.13 Μέθοδος Διακλάδωσης και Φράγματος	32
3. Προγραμματισμός	36
3.1 Ο Αλγόριθμος	36
3.2 Ο Κώδικας.....	37
3.2.1 Η Συνάρτηση Arrange.....	41
3.3 Αναλυτική επεξήγηση του κώδικα	42
3.3.1 Αναλυτική Επεξήγηση της Συνάρτησης Arrange	46
4. Το έργο της Δοκιμής.....	52
4.1 Περιγραφή του έργου δοκιμής.....	52
5. Παρουσίαση των Αποτελεσμάτων	60
5.1 Ενδεικτικά αποτελέσματα και γραφήματα	60
6. Συμπεράσματα	121
6.1 Συμπεράσματα.....	121
6.2 Συμβουλές προς χρήστες.....	122
6.3 Παρεμβάσεις- Βελτιώσεις.....	122

Βιβλιογραφία	124
---------------------------	------------

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ ΣΤΗ ΔΙΟΙΚΗΣΗ – ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ

1.1 Εισαγωγή

Ο σχεδιασμός και η υλοποίηση ενός έργου είναι μια διαδικασία περίπλοκη με οικονομικές συνέπειες, θετικές ή αρνητικές, τόσο για την εργολήπτρια εταιρεία όσο και για τον ιδιοκτήτη του έργου. Η Διοίκηση - Διαχείριση έργου καλείται να συνδυάσει με τον καλύτερο δυνατό τρόπο κεφάλαια, πόρους και χρόνο έτσι ώστε το έργο να αποβεί προσοδοφόρο για τους άμεσα ενδιαφερόμενους. Συχνό όμως, είναι το φαινόμενο να παρατηρούνται χρονικές και κοστολογικές υπερβάσεις, οι οποίες στην καλύτερη περίπτωση μειώνουν το κέρδος ενώ στην χειρότερη δυσφημούν την εταιρεία ή θέτουν σε κίνδυνο ακόμη και την ολοκλήρωση του έργου.

Τα κυριότερα προβλήματα τα οποία εμφανίζονται κατά την πορεία ενός έργου μπορούν να συνοψιστούν στα εξής :

- 1) Υπέρβαση κόστους.
- 2) Υπέρβαση χρόνου.
- 3) Εργασιακά προβλήματα.

Τα προβλήματα αυτά έχουν συνήθως κοινές και γνωστές ρίζες οι σημαντικότερες των οποίων είναι :

- 1) Η ασάφεια στόχων και αντικειμενικών σκοπών.
- 2) Οι ανεπαρκείς οικονομικές προβλέψεις.
- 3) Η ανεπαρκής πληροφόρηση.
- 4) Η ελλιπής οργανωτική δομή .
- 5) Η άκριτη συμπίεση του χρόνου του έργου.
- 6) Η έλλειψη τυποποίησης.
- 7) Η διοικητική ανεπάρκεια.

Όλα τα έργα ωστόσο παρουσιάζουν κοινά γενικά χαρακτηριστικά. Οι διαφορές τους μπορεί να είναι ποσοτικές ή να ανάγονται στην έμφαση που δίνεται σε κάθε συντελεστή τους. Έτσι, για παράδειγμα, σε έργα πληροφορικής το κύριο βάρος πέφτει στις μελέτες και την ανθρώπινη εργασία, ενώ στα κατασκευαστικά έργα θα πρέπει να εξετασθεί σοβαρά και το θέμα του εξοπλισμού και των υλικών. Σ' ένα μικρό έργο η διοίκηση μπορεί να ασκείται από ένα μόνο άτομο, ενώ σ' ένα μεγάλο έργο θα χρειαστούν επιτροπές και επιτελεία στο πλάι του υπεύθυνου του έργου. Επίσης, η λήψη αποφάσεων σε σχέση με το μέγεθος των συνεπειών έχει μικρότερη βαρύτητα στα μικρά έργα απ' ό,τι στα μεγάλα.

Οι βασικότερες λειτουργίες της διοικητικής Διαχείρισης ενός έργου είναι:

- 1) Ο Σχεδιασμός –Προγραμματισμός.
- 2) Η Οργάνωση.
- 3) Ο έλεγχος.

1.2 Ο Κύκλος Ζωής του Έργου

Η ζωή του κάθε έργου, ανεξάρτητα από το μέγεθος και τη φύση του, ακολουθεί κάποιες καθορισμένες φάσεις μέσα στο χρόνο. Ανάλογα με το έργο η διάρκεια των φάσεων αυτών ποικίλει. Σε γενικές γραμμές όμως ένα έργο μπορεί να καταμεριστεί στις εξής φάσεις :

1.2.1) Η σύλληψη του έργου

Η σύλληψη του έργου(Project Conception) αποτελεί την έναρξη της ζωής του έργου. Μια αρχική ιδέα εξετάζεται από διαφορετικές σκοπιές (οικονομική, τεχνική, λειτουργική, συμπεριφοράς προσωπικού) σε σχέση με τη σκοπιμότητα και εφικτότητα υλοποίησης της.

Εργαλεία της φάσης αυτής αποτελούν:

- α) ο καθορισμός του έργου(Project Definition) και
- β) η εκτίμηση της εφικτότητας(Feasibility Assessment) ή Μελέτη σκοπιμότητας (Feasibility Study). Η διαδικασία αυτή έχει σαν στόχο τη λήψη αποφάσεων υψηλού επιπέδου που αφορούν τις γενικές προδιαγραφές και την οργανωτική υποδομή ενός έργου, την εξεύρεση κεφαλαίων.

Ο καθορισμός του έργου περιλαμβάνει όλες εκείνες τις πληροφορίες που αφορούν τις βασικές προδιαγραφές του, δηλαδή :

- 1) τους αντικειμενικούς στόχους.
- 2) μεθοδολογία.
- 3) βασικές προδιαγραφές.
- 4) χρησιμοποιούμενη τεχνολογία.
- 5) απαιτούμενους πόρους .

και αποτελείται από δύο σκέλη:

- α) την περιγραφή του έργου .
- β) τους όρους αναφοράς δηλαδή:
 - τους αντικειμενικούς στόχους του έργου(Objectives).
 - τα όρια του πλαισίου μέσα στα οποία θα κινηθεί η μελέτη και η υλοποίηση του έργου(Scope).
 - τους περιορισμούς που θα ισχύσουν σε όλες τις φάσεις του έργου αναφορικά με την εργασία, το χρόνο, την οργάνωση, τους οικονομικούς πόρους και τα υλικά(Constraints) .
 - τους διαθέσιμους για το έργο πόρους, όπως ανθρωπόωρες, μηχανήματα, υλικά, χώροι εργασίας κ.λ.π.

Η εκτίμηση της εφικτότητας ασχολείται με

- τον όγκο της απαιτούμενης εργασίας,
- το κόστος ανάπτυξης και λειτουργίας,
- τη σκοπιμότητα του έργου από την αποτίμηση κόστους-ωφέλειας,
- το χρονοδιάγραμμα υλοποίησης,
- την αρχική περιγραφή του απαιτούμενου προγραμματισμού και
- την αρχική περιγραφή της απαιτούμενης οργανωτικής υποδομής.

Αποτελείται από τα ακόλουθα μέρη:

A) Τεχνική εφικτότητα(Technical Feasibility):

Αναφέρεται στη σύγκριση των τεχνικών δυνατοτήτων(τεχνολογία, τεχνογνωσία) οι οποίες είναι διαθέσιμες για την υλοποίηση του έργου και στην επιλογή της τεχνολογίας η οποία τελικά προκρίνεται για να χρησιμοποιηθεί.

B) Λειτουργική εφικτότητα(Operational Feasibility).

Αναφέρεται στην εκτίμηση ανταπόκρισης του έργου στους επιδιωκόμενους σκοπούς της υλοποίησης του.

Γ) Οικονομική εφικτότητα(Economic Feasibility).

Αναφέρεται στο κατά πόσο το έργο έχει παρούσα αξία θετικού καθαρού κέρδους.

Δ) Εφικτότητα συμπεριφοράς προσωπικού(Behavioral Feasibility).

Αναφέρεται στις επιδράσεις και επιπτώσεις του έργου στην ποιότητα εργασιακής ζωής των χρηστών του.

Η ολοκλήρωση της Εκτίμησης Εφικτότητας αποτελεί ένα σημαντικό σημείο ελέγχου. Η Διοίκηση πρέπει να αποφανθεί για τη συνέχιση ή όχι του έργου. Αλλά και στη περίπτωση συνέχισης του έργου, η εφικτότητα του θα πρέπει να εκτιμάται περιοδικά καθ' όλη τη διάρκεια του κύκλου ζωής του. Και τούτο διότι με την πρόοδο του έργου συγκεντρώνονται περισσότερες πληροφορίες καθιστώντας την αξιολόγηση πιο ολοκληρωμένη με πιθανό αποτέλεσμα την επαναθεώρηση ορισμένων αρχικών αποφάσεων από τη διοίκηση.

1.2.2) Η έναρξη του έργου

Η φάση αυτή, η οποία ακολουθεί εφ' όσον η προηγούμενη κατέληξε στη συνέχιση του έργου, χαρακτηρίζεται από μελέτες και προκαταρκτικές εργασίες. Σε επίπεδο μελετών , σε πρώτο χρόνο γίνεται η γενική μελέτη του έργου η οποία σε γενικές γραμμές καλύπτει την τεχνική και οικονομική πλευρά του(χονδρικός καταμερισμός εργασιών, χονδρική αποτίμηση απαιτήσεων σε ανθρώπους, εξοπλισμό, υλικά και ένας πρώτος απολογισμός του κόστους).Η μελέτη αυτή χρησιμεύει σαν πυξίδα για την έναρξη ορισμένων χρονοβόρων διαδικασιών και συντελεί στην εξοικονόμηση χρόνου του έργου. Με την πάροδο του χρόνου η γενική αυτή μελέτη εξειδικεύεται και καταλήγει σε μια αναλυτική μελέτη η οποία περιγράφει λεπτομερώς τα τεχνικά, οικονομικά και χρονικά χαρακτηριστικά του έργου όπως :

- 1) Καθορισμός των επιμέρους εργασιών.
- 2) Δημιουργία χρονοπρογραμμάτων.
- 3) Καταμερισμός των πόρων.
- 4) Κατάρτιση γενικού προϋπολογισμού.
- 5) Κατάρτιση επί μέρους προϋπολογισμών.
- 6) Κατάρτιση ημερολογίων.

Δημιουργείται έτσι το Σχέδιο Δράσης(Project Plan) το οποίο περιλαμβάνει το Σχεδιασμό-Προγραμματισμό και τον Προϋπολογισμό του έργου.

Οι εργασίες οι οποίες μπορούν να διεξάγονται παράλληλα με τις μελέτες, αφού βέβαια ολοκληρωθεί η πρώτη γενική μελέτη σχετίζονται με :

- Την έναρξη των διαδικασιών δημοπράτησης και την προεπιλογή υποψηφίων εργολάβων (εφ' όσον υπάρχει πρόθεση ανάθεσης σε τρίτους)
- Την ανίχνευση της αγοράς για το απαραίτητο ανθρώπινο δυναμικό, τον αναγκαίο εξοπλισμό και τα υλικά.
- Τις παραγγελίες υλικών με μεγάλο χρόνο παράδοσης
- Τη δημιουργία οργανωτικής υποδομής για τη συνέχιση του έργου κ.λ.π.

1.2.3) Υλοποίηση και Κατασκευή του έργου

Αυτή η φάση περιλαμβάνει τα ακόλουθα :

- Γίνεται η εγκατάσταση των εργολάβων, συνεργείων, εξοπλισμού, αγοράζονται υλικά

- Ξεκινούν οι εργασίες υλοποίησης του έργου.
- Αρχίζει η παρακολούθηση της εξέλιξης των εργασιών σε σχέση με τις μελέτες και

- Αναπροσαρμόζονται οι μελέτες και ο προϋπολογισμός ανάλογα με τα προβλήματα που παρουσιάζονται κατά την εξέλιξη του έργου.

Η περίοδος αυτή έχει τα εξής χαρακτηριστικά:

- Ο ιδιοκτήτης χάνει ένα μέρος ελέγχου του έργου. Οι εκτιμήσεις αντικαθίστανται από πραγματικά δεδομένα όπως προκύπτουν από τη παρακολούθηση του έργου ή προσαρμόζονται στις πραγματικές του απαιτήσεις όπως αυτές εκδηλώνονται σε καθημερινή βάση(απρόβλεπτες καθυστερήσεις , βλάβες απεργίες ,ελλείψεις υλικών, ανατιμήσεις)

- Δοκιμάζονται η ποιότητα των μελετών , η οργανωτική υποδομή του έργου, οι επιλογές ανθρώπων, εξοπλισμού, υλικών.

- Δοκιμάζονται η οξυδέρκεια , η ευλυγισία και η προσαρμοστικότητα της διοίκησης του έργου.

1.2.4) Ολοκλήρωση και Αποδοχή του έργου

Ολοκλήρωση του έργου σημαίνει ότι το έργο:

- Υλοποιήθηκε στη βάση των προδιαγραφών των τελευταίων αναθεωρημένων και αποδεκτών μελετών και προϋπολογισμού.
- Έγινε σύμφωνα με τις υπογραφείσες συμβάσεις.
- Είναι άμεσα λειτουργικό και αξιόπιστο .
- Τελείωσε χωρίς να υπάρχουν κανενός είδους εκκρεμότητες.

Η αποδοχή του έργου γίνεται με την υπογραφή πρωτοκόλλου παράδοσης-παραλαβής μεταξύ ιδιοκτήτη και εργολήπτη μετά από λεπτομερή έλεγχο.

1.2.5) Λειτουργία και Συντήρηση

Όσο καλή και αν ήταν η μελέτη και υλοποίηση του έργου, με τον καιρό παρουσιάζονται ανάγκες για βελτιωτικές μετατροπές. Οι λόγοι που υπαγορεύουν αυτές τις μετατροπές μπορεί να είναι:

- Ανάγκες οι οποίες δεν καταγράφηκαν κατά την περίοδο των μελετών
- Ελλείψεις οι οποίες διέφυγαν κατά τον έλεγχο παράδοσης του έργου και οι οποίες εμφανίστηκαν κατά τη λειτουργία του.
- Οργανωτικές, λειτουργικές και τεχνικές μεταβολές στο περιβάλλον του έργου
- Μεταβολές των θεσμών και νόμων
- Επέκταση, συμπλήρωση ή βελτίωση των λειτουργιών του έργου.

Σε ορισμένα έργα επιβάλλεται συντήρηση σε τακτά χρονικά διαστήματα η οποία μπορεί να γίνεται είτε από τον ιδιοκτήτη εάν διαθέτει τα απαραίτητα μέσα, είτε από τον κατασκευαστή του έργου εάν αυτό αναγράφεται στη σύμβαση του έργου, είτε από τρίτους με ανάθεση.

Γενικές παρατηρήσεις:

Ο χρονικός καταμερισμός του έργου του έργου σε φάσεις είναι αρκετά σχηματικός. Στην πράξη κατά την εξέλιξη μιας φάσης απαιτείται η αναδρομή σε κάποια προηγούμενη ή αναπροσαρμογή κάποιας προηγούμενης. Οι διάφορες φάσεις του κύκλου ζωής ενός έργου είναι συχνά δυσδιάκριτες μεταξύ τους, είτε γιατί το ξεκίνημα της μιας φάσης γίνεται πριν από το τέλος της προηγούμενης, είτε γιατί μια φάση έχει στοιχεία αναφοράς στην προηγούμενη, είτε γιατί η φύση και το μέγεθος του έργου προκαλούν την υποβάθμιση ορισμένων και τον υπερτονισμό άλλων. Κάθε φάση απαιτεί διαφορετικό τύπο οργάνωσης, διοίκησης και ελέγχου. Ο τύπος αυτός επιλέγεται ανάλογα με το μέγεθος και τη σημασία της φάσης καθώς και το μέγεθος και τη φύση του έργου. Ο υπολογισμός του κόστους, ο οποίος γίνεται στο τέλος κάθε φάσης, παρουσιάζει, μεγαλύτερη ακρίβεια όσο οι φάσεις προχωρούν.

1.3 Ο Σχεδιασμός - Προγραμματισμός Έργου

Ο Σχεδιασμός – Προγραμματισμός ενός έργου γίνεται κάτω από συνθήκες αβεβαιότητας. Αυτό σημαίνει ότι ο διοικητικός φορέας του έργου δεν έχει καθόλου ή έχει ελάχιστες πληροφορίες για ενδεχόμενα συμβάντα κατά την υλοποίηση αυτού του έργου (ασθένειες προσωπικού, βλάβες μηχανημάτων, καιρικές συνθήκες κ.λ.π.). Κατά συνέπεια οι εκτιμήσεις και η λήψη αποφάσεων αντανakλούν τις ικανότητες και δυνατότητες του διοικητικού φορέα να κατανοεί και να αξιολογεί τις συνθήκες του περιβάλλοντος της υλοποίησης του έργου(προβλέψεις χρόνων, δαπανών , εργασίας)

Στόχοι του Σχεδιασμού – Προγραμματισμού είναι :

- Η έγκαιρη παροχή των χρονοδιαγραμμάτων του όλου έργου με έμφαση στις αρχικές ενέργειες.
- Η κατανομή των πόρων του έργου.
- Η κατάρτιση προϋπολογισμού.

- Η μελέτη της χρηματοροής.
- Η προετοιμασία του σχεδίου Δράσης.
- Η έγκαιρη επισήμανση πιθανών καθυστερήσεων και η εξεύρεση εναλλακτικών λύσεων στην περίπτωση εμφάνισής τους.
- Η επισήμανση και ο προγραμματισμός των στρατηγικών δραστηριοτήτων.

Σε γενικές γραμμές στόχος του σχεδιασμού και προγραμματισμού έργου είναι η βελτιστοποίηση

- της χρονικής διάρκειας του έργου.
- του κόστους του έργου.
- της κατανομής των διαθέσιμων παραγωγικών πόρων για την υλοποίηση του έργου.

1.4 Ο Χρονικός Προγραμματισμός

Ο στόχος του χρονικού προγραμματισμού του έργου είναι ο προσεγγιστικός προσδιορισμός της χρονικής διάρκειας έναρξης και λήξης των δραστηριοτήτων, φάσεων και ολόκληρου του έργου.

Η προσοχή επικεντρώνεται σε ζητήματα όπως :

- Ποια θα είναι η αναμενόμενη διάρκεια του έργου.
- Ποια η επιτρεπόμενη καθυστέρηση στην έναρξη και λήξη κάθε δραστηριότητας και φάσης ώστε να μην υπάρξει ανατροπή του χρονοδιαγράμματος των εργασιών.
- Ποιες δραστηριότητες παρουσιάζουν κρισιμότητα για την τήρηση του χρονικού προγράμματος των εργασιών.
- Ποιες οι ημερομηνίες έναρξης και λήξης των εργασιών των δραστηριοτήτων και του έργου.

1.5 Μελέτη χρηματοροής

Συχνά οι κατασκευαστές του έργου βρίσκονται σε δύσκολη θέση εξαιτίας της χρηματοροής του έργου. Καλούνται να πληρώσουν για έξοδα ενώ τα διαθέσιμά τους βρίσκονται στο μηδέν. Στην καλύτερη περίπτωση απαιτείται δανεισμός με αποτέλεσμα την αύξηση του κόστους του έργου και τον περιορισμό του κέρδους, ενώ στη χειρότερη διακυβεύεται η ύπαρξή τους.

Για το λόγο αυτό η πρόβλεψη της χρηματοροής και η μελέτη των επιπτώσεων της στην εξέλιξη του έργου αποτελεί αναπόσπαστο μέρος του σχεδιασμού – προγραμματισμού του έργου. Είναι πιθανό μια τέτοια μελέτη να επιβάλει μεταβολές στο σχεδιασμό του δικτύου του έργου.

Η εκτίμηση και μελέτη της χρηματοροής παρουσιάζει τα ακόλουθα πλεονεκτήματα :

- Επιτρέπει την πρόβλεψη των απαιτήσεων σε πόρους σε σχέση με τις δυνατότητες τους.
- Προειδοποιεί έγκαιρα για τις στιγμιαίες αρνητικές ροές και επιτρέπει την εξεύρεση εναλλακτικών τρόπων αντιμετώπισής τους.
- Προβλέπει τις απαιτούμενες ελάχιστες καταβολές (χρονικά και ποσοτικά) από τον πελάτη.
- Δημιουργεί μέτρα σύγκρισης αξιοποίησης της επένδυσης (επιστροφές, αποδόσεις κεφαλαίων).
- Επιτρέπει τον καθορισμό πολιτικής δανεισμού (ποσά, επιτόκια, ημερομηνίες καταβολής τόκων).
- Επιτρέπει τον καθορισμό πολιτικής εξόφλησης των προμηθευτών
- Αποτελεί βάση για τον έλεγχο του έργου.
- Αποτελεί πηγή πληροφοριών για την οικονομική πολιτική της εταιρείας.

1.6 Το Ανθρώπινο Δυναμικό

Η παρακολούθηση και ο έλεγχος του ανθρώπινου δυναμικού επιτρέπει σε μεγάλο βαθμό την παρακολούθηση της εξέλιξης του χρόνου και του κόστους του έργου, δεδομένου ότι η ανθρώπινη εργασία αποτελεί το βασικότερο παράγοντα της υλοποίησης ενός έργου. Η αλληλεξάρτηση ανθρώπινου δυναμικού, χρόνου και κόστους είναι ιδιαίτερα σημαντική και η λεπτομερής μελέτη της συμβάλλει στην επιτυχή αποπεράτωση ενός έργου.

1.7 Το φαινόμενο της άκριτης συμπίεσης χρόνου

Σε αρκετά έργα παρατηρείται το συγκεκριμένο φαινόμενο και συνήθως προκαλείται είτε από τους χρονικούς περιορισμούς που είναι δεδομένοι για το έργο, είτε από την ολοένα αυξανόμενη ανταγωνιστικότητα, είτε και από υπέρμετρη αισιοδοξία του Υπευθύνου Διαχείρισης του έργου. Η άκριτη συμπίεση του χρόνου εκτός από το υψηλό ρίσκο για έγκαιρη παράδοση του έργου (ειδικά στα μεγάλα και πολυσύνθετα έργα), επηρεάζει αρνητικά τη χρηματοροή του έργου, καθώς συμπιέζει μέσα σε συγκεκριμένα χρονικά διαστήματα αρκετές εργασίες ή φάσεις του έργου, οι οποίες γίνονται παράλληλα ή με το μικρότερο δυνατό χρονικό περιθώριο εκτινάσσοντας τον συνολικό αριθμό των εργαζομένων σε αυτά τα διαστήματα. Η εκτίναξη των απασχολούμενων μονάδων του ανθρώπινου δυναμικού έχει ως αποτέλεσμα την ραγδαία αύξηση των εξόδων στις συγκεκριμένες χρονικές περιόδους. Καλούνται λοιπόν οι υπεύθυνοι-κατασκευαστές του έργου να πληρώσουν έξοδα πολύ περισσότερα από άλλες φάσεις του έργου και αυτό οφείλεται στη συσσώρευση των εργασιών. Στην καλύτερη περίπτωση όταν υπάρχουν διαθέσιμα κεφάλαια αυτά εξαντλούνται, ενώ όταν δεν υπάρχουν απαιτείται δανεισμός με αποτέλεσμα την αύξηση του κόστους και τον περιορισμό των κερδών ενώ μειώνεται και η αξιοπιστία της εργολήπτριας εταιρείας. Έτσι κρίνεται απαραίτητη η λελογισμένη συμπίεση του έργου καθώς το αντίθετο μπορεί να δυσβάσταχτες συνέπειες για την εργολήπτρια εταιρεία κυρίως οικονομικές.

1.8 Το προς εξέταση πρόβλημα

Ένα από τα σημαντικότερα προβλήματα που μπορεί να προκύψει από το κακό σχεδιασμό-προγραμματισμό ενός έργου όπως είδαμε είναι η ασύνετη διαχείριση του εργατικού δυναμικού από την στρέψη του ενδιαφέροντός μας περισσότερο στην χρονική βελτιστοποίηση. Στην ουσία, σε συγκεκριμένες φάσεις του έργου στις οποίες παρατηρείται υπεραπασχόληση εργατών εκτινάσσονται τα έξοδα της εργολήπτριας εταιρείας ή του δικαιούχου του έργου, θέτοντας έτσι σε δύσκολη κατάσταση την οικονομική θέση της εταιρείας ή του δικαιούχου. Και όχι μόνον από οικονομικής αλλά και από πλευράς διαχείρισης ανθρωπίνων πόρων δεν συμφέρει μια εταιρεία να διανύει περιόδους κατά τις οποίες απασχολεί τα συνεργεία της ταυτόχρονα και με υπερβολική συμπίεση χρόνου, γνωρίζοντας ότι στο εγγύς μέλλον κάποια από αυτά – λόγω αλληλουχίας του αντικειμένου εργασίας τους – θα μείνουν χωρίς εργασία ή δεν θα λειτουργούν πλήρως. Συνετότερο κρίνεται λοιπόν, να συνδυάσουμε χρόνο και εργάτες έτσι ώστε να καταλήξουμε σε μια πιο ομαλή κατανομή εργατικού δυναμικού. Αυτό βέβαια θα γίνει αφιερώνοντας παραπάνω χρονικό διάστημα για την πραγματοποίηση του έργου.

ΚΕΦΑΛΑΙΟ 2

ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΕΠΙΛΥΣΗ – ΧΡΟΝΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΧΡΗΣΗΣ ΠΟΡΩΝ

2.1 Εισαγωγή στον Χρονικό Προγραμματισμό

Καθοριστικό παράγοντα της επιτυχίας ενός έργου αποτελεί αδιαμφισβήτητα ο σχεδιασμός και προγραμματισμός του. Η αποτελεσματική Διοίκηση και Διαχείριση Έργου οδηγεί στην οικονομικότερη χρήση των απαιτούμενων μηχανημάτων, στην επίτευξη χαμηλών τιμών κόστους και διάρκειας της κατασκευής του έργου, στην πλέον αποδοτική και οικονομική χρήση των επενδυόμενων πόρων, σε συντονισμό του προσωπικού και των εργασιών ώστε να παρατηρούνται ελάχιστες απώλειες ανθρώπινης εργασίας αλλά και στον βέλτιστο χειρισμό και συνδυασμό κεφαλαίων και χρόνου. Βασικό συντελεστή της ποιότητας διοίκησης ενός έργου αποτελεί η ποιότητα του συστήματος πληροφόρησης και του μηχανισμού λήψης αποφάσεων. Το πρόβλημα της οργάνωσης του έργου απαιτεί τη δημιουργία μηχανισμού, του οποίου το μέγεθος και η πολυπλοκότητα ποικίλουν ανάλογα με το μέγεθος και τη φύση του έργου αλλά και την οργανωτική αντίληψη της διοίκησης. Ο προγραμματισμός και σχεδιασμός του έργου γίνεται κάτω από συνθήκες αβεβαιότητας καθώς ο διοικητικός φορέας του έργου δεν έχει πληροφορίες για ενδεχόμενα συμβάντα κατά την υλοποίηση του, όπως οι φυσικές καταστροφές, οι βλάβες των μηχανημάτων, οι καιρικές συνθήκες κτλ. Οι εκτιμήσεις λοιπόν και η λήψη αποφάσεων αντανakλούν εκτός των άλλων και την ικανότητα του διοικητικού φορέα να αξιολογεί κατάλληλα τις περιβαλλοντολογικές συνθήκες και τις συνθήκες υλοποίησης του έργου. Όπως λοιπόν προκύπτει από όλα τα παραπάνω, στόχο της Διοίκησης και Διαχείρισης Έργου αποτελούν :

- Η έγκαιρη παροχή των χρονοδιαγραμμάτων όλου του έργου με έμφαση στις αρχικές ενέργειες.
- Η δημιουργία χρονικού και οικονομικού προγράμματος του έργου.
- Η δημιουργία καναλιών επικοινωνίας και πληροφόρησης.
- Ο εντοπισμός εκείνων των κρίσιμων επιμέρους εργασιών του έργου, από την έγκαιρη εκτέλεση των οποίων εξαρτάται η συνολική διάρκεια του έργου.
- Η κατάρτιση του προϋπολογισμού και η επαρκής κατανομή των πόρων του έργου.
- Η επίτευξη ενός καλύτερου συντονισμού της εκτελέσεως των εργασιών που συνθέτουν το έργο, μέσω της αποτελεσματικής προετοιμασίας του σχεδίου δράσης.
- Η μελέτη της χρηματοροής.
- Η εύρεση της σχέσης ανάμεσα στο συνολικό κόστος του έργου και στη διάρκεια κατασκευής, καθώς και η ελαχιστοποίηση αυτής.
- Η έγκαιρη επισήμανση και ο προγραμματισμός των στρατηγικών

δραστηριοτήτων.

- Ο συντονισμός των εντός εργοταξίου δραστηριοτήτων με τις εκτός.

Σημειώνεται, τέλος, ότι η διοίκηση-διαχείριση έργου βασίζεται στην *Δικτυωτή Ανάλυση*, η οποία χρησιμοποιείται σαν μέσο προγραμματισμού και ελέγχου του έργου. Στόχος της δικτυωτής ανάλυσης είναι η βελτιστοποίηση:

- Της χρονικής διάρκειας του έργου
- Του συνολικού κόστους του έργου
- Της κατανομής των διαθέσιμων παραγωγικών πόρων για την υλοποίηση του έργου.

2.2 Δομική Ανάλυση Έργου

Το πρώτο βήμα στον χρονικό προγραμματισμό έργων αποτελεί η δομική ανάλυση του έργου, η οποία περιλαμβάνει την δομική ανάλυση σε φάσεις και τις αντίστοιχες δραστηριότητες αυτών. Σαν *Φάση* ορίζεται ένα τμήμα του έργου που καθορίζει μία προσπάθεια με συγκεκριμένο αποτέλεσμα ή μία ενότητα με συγκεκριμένο οικονομοτεχνικό και λειτουργικό περιεχόμενο. Κάθε φάση απαιτεί συγκεκριμένο χρονικό διάστημα και συγκεκριμένη ποσότητα πόρων για την παραγωγή του επιθυμητού αποτελέσματος. Σαν Δραστηριότητα (ή Επιμέρους Εργασία ή απλά Εργασία) ορίζεται κάθε επιμέρους εργασία του έργου ή μίας φάσης, η οποία απαιτεί για την υλοποίησή της χρόνο ή πόρους ή και τα δύο μαζί και αποτελεί το στοιχειώδες δομικό στοιχείο αναφοράς στην ανάλυση του δικτύου. Για κάθε φάση πρέπει να καθοριστούν οι δραστηριότητες που την αποτελούν και για κάθε δραστηριότητα η χρονική της διάρκεια και η σχέση της με τις προηγούμενες και τις επόμενες. Ο καθορισμός των δραστηριοτήτων δεν είναι απλή υπόθεση, μίας και πολλές φορές δύο ή και περισσότερες δραστηριότητες μπορεί να χρησιμοποιούν κοινούς πόρους (π.χ., μηχανήματα) την ίδια χρονική περίοδο. Σε γενικές γραμμές, ο καθορισμός της κάθε δραστηριότητας πρέπει να γίνεται κατά τέτοιο τρόπο ώστε αυτή να είναι «ανεξάρτητη» από τις άλλες, π.χ., να είναι δυνατή η χρονική συμπίεση αυτής της δραστηριότητας (με τη χρήση είτε επιπλέον προσωπικού ή υπερωριακής απασχόλησης) χωρίς συνέπειες/παρενέργειες στις υπόλοιπες δραστηριότητες που εκτελούνται παράλληλα με τη δραστηριότητα αυτή.

2.3 Δίκτυο Έργου

Μετά την δομική ανάλυση του έργου σε εργασίες (φάσεις και επιμέρους εργασίες) και τον προσδιορισμό των προηγούμενων εργασιών πραγματοποιείται ο σχεδιασμός του δικτύου. Υπάρχουν δύο τύποι δικτύων:

- Τα *Τοξωτά Δίκτυα*, στα οποία οι δραστηριότητες παριστάνονται με βέλη. Στα δίκτυα αυτά οι σχέσεις των δραστηριοτήτων είναι πολύ απλές γι'αυτό και είναι πιο κατανοητά, ευκρινή αλλά και πιο εκτεταμένα.

- Τα *Κομβικά Δίκτυα*, στα οποία κάθε δραστηριότητα παριστάνεται με ένα κόμβο του δικτύου. Τα δίκτυα αυτά επιτρέπουν την απεικόνιση πιο σύνθετων σχέσεων μεταξύ των δραστηριοτήτων και παρέχουν μεγαλύτερες δυνατότητες σε εναλλακτικές λύσεις στον σχεδιασμό, αλλά είναι περισσότερο δυσνόητα.

Εκτός από τα δύο παραπάνω δίκτυα υπάρχει και το **διάγραμμα GANTT**, το οποίο αποτελεί μία μορφή γραφικής αναπαράστασης ενός χρονοδιαγράμματος και θα διαδραματίσει σημαντικό ρόλο στην προσπάθεια μας να επιτύχουμε διαφορετικές και καλύτερες λύσεις στο έργο που θα εξετάσουμε. Στον οριζόντιο άξονα του διαγράμματος GANTT, σχεδιάζεται η χρονική κλίμακα του έργου και στον κάθετο άξονα τοποθετούνται ιεραρχημένες οι επιμέρους εργασίες του έργου, από κάτω προς τα πάνω και από την πρώτη εργασία προς την τελευταία. Για κάθε φάση σχεδιάζεται οριζόντια μία συμπαγής ράβδος με μήκος ίσο με την προβλεπόμενη χρονική διάρκεια της φάσης και από κάτω ακριβώς μία κενή ράβδος η οποία συμπληρώνεται κατά την εξέλιξη του έργου.

2.4 Τοξωτό δίκτυο

Τα δομικά στοιχεία ενός τοξωτού δικτύου είναι τα γεγονότα και οι δραστηριότητες. Τα γεγονότα είναι σημεία στο χρόνο τα οποία ορίζουν την αρχή και το τέλος μίας δραστηριότητας, παριστάνονται με τετράγωνο και αριθμούνται μονοσήμαντα. Η κάθε δραστηριότητα του δικτύου παριστάνεται με ένα βέλος (άνυσμα) που συνδέει δύο γεγονότα.

Σε ορισμένες περιπτώσεις, απαιτείται η λειτουργική σύνδεση στο δίκτυο, δύο γεγονότων η οποία δεν αποτελεί δραστηριότητα (δεν απαιτεί ούτε χρόνο ούτε πόρους). Η σύνδεση αυτή ονομάζεται *Πλασματική Δραστηριότητα* και συμβολίζεται με διακεκομμένο βέλος.

Η κάθε δραστηριότητα του δικτύου για να καθορίζεται πλήρως απαιτεί:

- Τον καθορισμό ενός μοναδικού κωδικού, ο οποίος μπορεί να είναι κεφαλαίο γράμμα, μονοσήμαντα ορισμένος αριθμός ή ο Δ_{ij} , όπου i, j οι αριθμοί των γεγονότων της αρχής και του τέλους της δραστηριότητας, αντίστοιχα.
- Την περιγραφή της
- Τη χρονική διάρκεια της ή διαθέσιμο χρόνο
- Τη λογική σύνδεση της με τις προηγούμενες και επόμενες

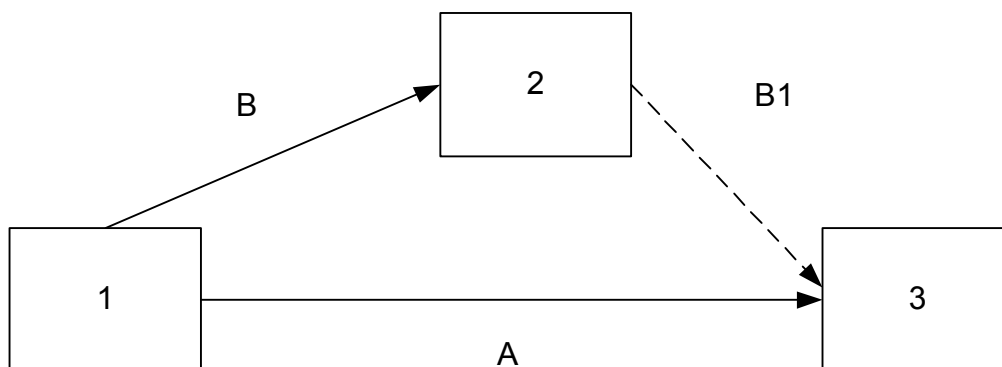
Ο κωδικός και η χρονική διάρκεια της δραστηριότητας σημειώνονται πάνω στο βέλος το οποίο την απεικονίζει στο δίκτυο.

Οι δραστηριότητες πρέπει να ακολουθούν τις παρακάτω σχέσεις αλληλουχίας:

- Μία δραστηριότητα δεν μπορεί να ξεκινήσει αν δεν έχει ολοκληρωθεί η προηγούμενη της. Αν συμβαίνει μία δραστηριότητα να μπορεί να ξεκινήσει πριν ολοκληρωθεί η προηγούμενη της, τότε θα πρέπει να ανακαθοριστούν οι δραστηριότητες αυτές.
- Δεν επιτρέπεται η διακοπή μίας δραστηριότητας. Αν για κάποιο λόγο

συμβεί αυτό, η δραστηριότητα αυτή θα πρέπει να διασπαστεί σε άλλες πιο μικρές έτσι ώστε η διακοπή να συμπίπτει με την ολοκλήρωση κάποιας από αυτές.

- Δύο (ή και παραπάνω) παράλληλες δραστηριότητες δεν μπορούν να έχουν τα ίδια γεγονότα σαν αρχή και τέλος. Στην περίπτωση αυτή χρησιμοποιούμε πλασματικές δραστηριότητες με μηδενικές χρονικές διάρκειες, όπως φαίνεται στο παρακάτω σχήμα :



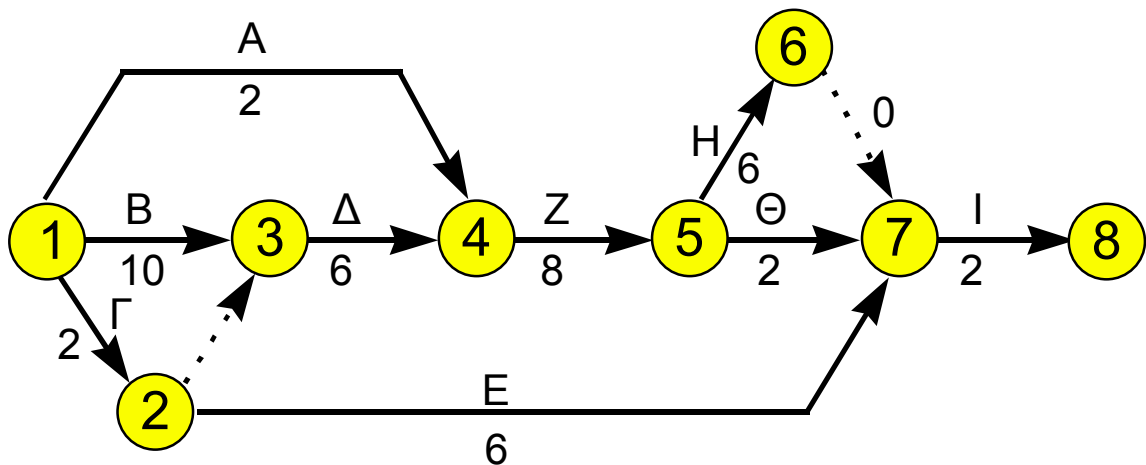
Σχήμα 2.1 Οι δραστηριότητες A και B εκτελούνται παράλληλα. Η δραστηριότητα B1 αποτελεί πλασματική δραστηριότητα με μηδενική χρονική διάρκεια

- Απαγορεύεται η ύπαρξη βρόχων (π.χ. A -> B -> Γ-> A).

Παρακάτω παραθέτουμε ένα παράδειγμα τοξωτού δικτύου που αντιστοιχεί στο έργο που περιγράφεται στον Πίνακα 2.1.

Κωδικός Εργασίας	Προηγούμενες Εργασίες	Διάρκεια
A	Αρχή Έργου	2
B	Αρχή Έργου	10
Γ	Αρχή Έργου	2
Δ	B, Γ	6
Ε	Γ	6
Z	A, Δ	8
Η	Z	6
Θ	Z	2
I	E, Θ,, Η	2

Πίνακας 2.1



Σχήμα 2.2 Τοξωτό Δίκτυο για το Έργο του Πίνακα 2.1 (σσ: το παράδειγμα αυτό είναι από τις σημειώσεις μαθήματος Διοίκηση & Χρονικός Προγραμματισμός Έργων, Εμίρης Δ., Πολυτεχνείο Κρήτης, 2000)

2.5 Κομβικό δίκτυο

Στο δίκτυο αυτό οι δραστηριότητες παριστάνονται με τετράγωνα τα οποία συνδέονται μεταξύ τους με βέλη που απεικονίζουν τις σχέσεις αλληλουχίας μεταξύ των δραστηριοτήτων. Η κάθε δραστηριότητα προσδιορίζεται όπως και στο τοξωτό δίκτυο, όπου ο κωδικός και η χρονική διάρκεια της κάθε μίας σημειώνονται στο τετράγωνο που την απεικονίζει.

Οι σχέσεις που μπορούν να υπάρξουν μεταξύ των δραστηριοτήτων είναι οι εξής:

- Η έναρξη μίας δραστηριότητας εξαρτάται από τη λήξη της προηγούμενης της (Finish-to-Start constraint, FS). Π.χ., όταν ορίζουμε ότι $FS(A,B)=5$, εννοούμε ότι η έναρξη της δραστηριότητας B πρέπει να γίνει 5 χρονικές μονάδες μετά την λήξη της δραστηριότητας A, ενώ όταν ορίζουμε ότι $FS(A,B)=-5$, εννοούμε ότι η έναρξη της δραστηριότητας B πρέπει να γίνει 5 χρονικές μονάδες πριν την λήξη της δραστηριότητας A.
- Η έναρξη μίας δραστηριότητας εξαρτάται από την έναρξη της προηγούμενης της (Start-to-Start constraint, SS)
- Η λήξη μίας δραστηριότητας εξαρτάται από την λήξη της προηγούμενης της (Finish-to-Finish constraint, FF)
- Η λήξη μίας δραστηριότητας εξαρτάται από την έναρξη της προηγούμενης της (Start-to-Finish constraint, SF)

Πάνω στα βέλη που συνδέουν τις δραστηριότητες σημειώνεται το είδος της σχέσης αλληλουχίας για κάθε ζεύγος δραστηριοτήτων (SS, SF, FS, FF) καθώς και οι επιβαλλόμενοι από αυτές τις σχέσεις χρόνοι.

Σημείωση: Από εδώ και στο εξής θα επικεντρωθούμε μόνο στα τοξωτά δίκτυα. Σε περίπτωση δραστηριοτήτων που εξαρτώνται από άλλες μέσω μίας

από τις παραπάνω σχέσεις (SS, SF, FS, FF), επανακαθορίζουμε τις δραστηριότητες, «σπάζοντας» τις δραστηριότητες αυτές σε επιμέρους δραστηριότητες, έτσι ώστε όλες οι να επανακαθορισμένες δραστηριότητες εξαρτώνται μεταξύ τους με σχέσεις FS=0.

2.6 Χρονικός Προγραμματισμός του δικτύου

Ο στόχος του χρονικού προγραμματισμού είναι ο προσεγγιστικός προσδιορισμός της χρονικής διάρκειας έναρξης και λήξης των δραστηριοτήτων του έργου.

Έτσι ζητείται να προσδιοριστεί:

- Η συνολική διάρκεια του έργου με βάση τη χρονική διάρκεια της κάθε εργασίας του έργου
- Το συνολικό κόστος του έργου
- Οι ημερομηνίες έναρξης και λήξης των εργασιών του έργου έτσι ώστε να τηρηθεί η συνολική διάρκεια του έργου
- Η επιτρεπόμενη καθυστέρηση στην έναρξη και τη λήξη κάθε εργασίας χωρίς να ανατραπεί το χρονοδιάγραμμα των εργασιών

Για να γίνει η παραπάνω ανάλυση του δικτύου θα πρέπει να υπολογιστούν κάποια χρονικά στοιχεία που αναφέρονται στο τοξωτό και το κομβικό δίκτυο.

2.7 Χρονική ανάλυση τοξωτού δικτύου

Τα χρονικά στοιχεία του τοξωτού δικτύου που πρέπει να υπολογιστούν είναι:

- *Ο Ενωρίτερος χρόνος (EX)* του κάθε γεγονότος που είναι ο μικρότερος δυνατός χρόνος πραγματοποίησής του. Ο χρόνος αυτός είναι ίσος με την μεγαλύτερη διαδρομή από το αρχικό γεγονός μέχρι το συγκεκριμένο.
- *Ο Βραδύτερος χρόνος (BX)* του κάθε γεγονότος που είναι ο μέγιστος διαθέσιμος χρόνος πραγματοποίησής του χωρίς να επηρεαστεί η συνολική διάρκεια του έργου. Ο χρόνος αυτός υπολογίζεται εάν από τον βραδύτερο χρόνο του επόμενου γεγονότος αφαιρεθεί η μικρότερη διαδρομή από το μικρότερο γεγονός.
- *Το Συνολικό περιθώριο χρόνου (ΣΠΧ)* που είναι το μεγαλύτερο χρονικό διάστημα κατά το οποίο μπορεί να επιβραδυνθεί μία δραστηριότητα χωρίς να μεγαλώσει η συνολική διάρκεια του έργου. Υπολογίζεται από την σχέση:

$$\Sigma \Pi X_{ij} = BX_j - EX_i - X\Delta_{ij}$$

όπου τα i, j αντιστοιχούν στους αριθμούς (κωδικούς) των γεγονότων έναρξης και λήξης της αντίστοιχης δραστηριότητας.

- *Το Ελεύθερο περιθώριο χρόνου (ΕΠΧ)* που είναι το χρονικό διάστημα κατά το οποίο μπορεί να επιβραδυνθεί μία δραστηριότητα χωρίς να

επηρεαστεί ο ενωρίτερος χρόνος έναρξης των επόμενων δραστηριοτήτων. Υπολογίζεται από την σχέση:

$$ΕΠΧ_{ij} = EX_j - EX_i - XΔ_{ij}$$

- *Το Ανεξάρτητο περιθώριο χρόνου (ΑΠΧ)* που είναι το χρονικό διάστημα κατά το οποίο μία δραστηριότητα μπορεί να επιβραδυνθεί, χωρίς να επηρεαστεί ούτε το βραδύτερο τέλος των προηγούμενων, ούτε η ενωρίτερη έναρξη των επόμενων δραστηριοτήτων. Υπολογίζεται από την σχέση:

$$ΑΠΧ_{ij} = EX_j - BX_i - XΔ_{ij}$$

Οι EX και BX υπολογίζονται με βάση τους παρακάτω τύπους:

$$EX_j = \max\{EX_i + XΔ_{ij}\}$$

$$BX_i = \min\{BX_j - XΔ_{ij}\}$$

όπου τα i, j αντιστοιχούν στους αριθμούς (κωδικούς) των γεγονότων για τα οποία υπάρχει τόξο το οποίο συνδέει τα γεγονότα αυτά (με κατεύθυνση από το i στο j).

Όσες εργασίες έχουν *Συνολικό Περιθώριο Χρόνου ίσο με το μηδέν* ονομάζονται **κρίσιμες**. Η έννοια της κρισιμότητας προέρχεται από το γεγονός ότι οι δραστηριότητες αυτές δεν επιτρέπουν μεταβολή στην προγραμματισμένη του διάρκειας, χωρίς να υπάρξει συνέπεια στη συνολική διάρκεια του έργου. Η επιβράδυνσή του συνεπάγεται την επιβράδυνση του έργου. Οι εργασίες που έχουν ΣΠΧ διάφορο του μηδενός σημαίνει ότι έχουν περιθώριο να καθυστερήσουν ίσο με το ΣΠΧ της καθεμίας χωρίς συνέπειες για τις άλλες δραστηριότητες του έργου ή την συνολική διάρκεια του έργου.

Μία σειρά δραστηριοτήτων (τόξων) που ξεκινά από την αρχή του δικτύου και τελειώνει στο τέλος του, ονομάζεται **κρίσιμη διαδρομή**. Υπάρχει περίπτωση να υπάρχουν περισσότερες της μίας, κρίσιμες διαδρομές.

Ο υπολογισμός των EX, BX και ΣΠΧ από εδώ και στο εξής θα ονομάζεται **χρονική ανάλυση του έργου (ή του δικτύου)**.

2.8 Τεχνική PERT

Η τεχνική PERT (Project Evaluation and Review Technique) χρησιμοποιείται για την αντιμετώπιση της αβεβαιότητας του χρόνου των δραστηριοτήτων ενός έργου. Η μέθοδος αυτή εφαρμόζεται στις περιπτώσεις έργων στα οποία μερικές ή όλες οι επιμέρους εργασίες εκτελούνται κάτω από συνθήκες που δεν ελέγχονται πλήρως, με αποτέλεσμα την αδυναμία προβλέψεως του τελικού χρόνου εκτελέσεως αυτών των εργασιών. Σε αυτές τις περιπτώσεις η χρονική διάρκεια της εργασίας θεωρείται μία στοχαστική μεταβλητή, κυμαινόμενη εντός ενός διαστήματος μεταβολής, με αποτέλεσμα την αδυναμία καθορισμού μίας και μόνο τιμής της χρονικής διάρκειας της εργασίας. Οι δυνατότητες της τεχνικής αυτής είναι:

- Ο υπολογισμός της πιθανότητας να κατασκευαστεί το όλο έργο σε ένα προκαθορισμένο χρονικό διάστημα

- Ο υπολογισμός του χρόνου εκτέλεσης ενός έργου με συγκεκριμένη πιθανότητα

Για να εφαρμοστεί η μέθοδος PERT για κάθε δραστηριότητα ορίζονται οι τρεις παρακάτω χρόνοι:

- Ο αισιόδοξος a
- Ο απαισιόδοξος b
- Ο πιο πιθανός m

όπου $b < m < a$. Οι παραπάνω χρόνοι καθορίζονται βασιζόμενοι στην εμπειρία υλοποίησης παρόμοιων έργων. Στη συνέχεια υπολογίζεται ο αναμενόμενος χρόνος T_e κάθε δραστηριότητας, ο οποίος είναι ο απαραίτητος μέσος χρόνος για την υλοποίησή της και δίνεται από την σχέση:

$$T_e = \frac{a + 4m + b}{6}$$

Η διακύμανση του χρόνου κάθε δραστηριότητας, η οποία εκφράζει την απόκλιση από την μέση τιμή δίνεται από την σχέση:

$$\sigma^2 = \left(\frac{b - a}{6} \right)^2$$

Η τεχνική PERT περιλαμβάνει τα παρακάτω βήματα:

Βήμα 1°

Για κάθε δραστηριότητα υπολογίζονται ο αναμενόμενος χρόνος T_e και η διακύμανση σ^2 .

Βήμα 2°

Για κάθε δραστηριότητα υπολογίζεται ο ενωρίτερος και βραδύτερος χρόνος έναρξης και λήξης με βάση τον αναμενόμενο χρόνο T_e (και όχι την κανονική χρονική διάρκεια m), ο αναμενόμενος χρόνος υλοποίησης του έργου T_n , το συνολικό περιθώριο χρόνου καθώς και οι κρίσιμες διαδρομές του δικτύου, δηλαδή με άλλα λόγια, γίνεται η χρονική ανάλυση του έργου υποθέτοντας ότι η χρονική διάρκεια της κάθε δραστηριότητας ισούται με T_e .

Βήμα 3°

Υπολογίζεται η διακύμανση του όλου έργου $\sigma_{ολ}^2$, η οποία ισούται με το άθροισμα των διακυμάνσεων των φάσεων της κρίσιμης διαδρομής του. Αν υπάρχουν περισσότερες της μίας κρίσιμες διαδρομές χρησιμοποιείται εκείνη που έχει μεγαλύτερη διακύμανση.

Βήμα 4°

Υπολογίζεται η σταθερά Z της κανονικής κατανομής μέσω της σχέσης:

$$Z = \frac{T_x - T_n}{\sigma_{ολ}}$$

Με βάση την τιμή της σταθεράς Z , υπολογίζουμε την πιθανότητα εκτέλεσης του έργου στο συγκεκριμένο χρονικό διάστημα T_x , με βάση τον

Πίνακα κανονικής κατανομής που βρίσκεται στο Παράρτημα.

2.9 Άμεσο -Έμμεσο κόστος

Το συνολικό κόστος ενός έργου αναλύεται σε δύο συνιστώσες:

- Έμμεσο κόστος, το οποίο αποτελεί το κόστος το οποίο εξαρτάται αποκλειστικά από την χρονική διάρκεια του έργου (π.χ., ημερήσια ενοικίαση οχημάτων/κτιρίων, ποινικές ρήτρες σε περίπτωση καθυστέρησης του έργου). Με άλλα λόγια, το έμμεσο κόστος (ΕΚ) δίνεται από την σχέση

$$EK = f(\Sigma\Delta K)$$

όπου $\Sigma\Delta K$ είναι η συνολική διάρκεια του έργου και f είναι μία αύξουσα - άλλα όχι απαραίτητα γνησίως αύξουσα - συνάρτηση.

- Άμεσο κόστος, το οποίο αποτελεί το κόστος που εξαρτάται από τους πόρους (π.χ., εργατικό δυναμικό) και τον τρόπο χρήσης τους (π.χ., υπερωρίες), και συνεπώς μπορεί να μεταβληθεί ανάλογα με τις ανάγκες του έργου. Θα χρησιμοποιήσουμε την υπόθεση, ότι το άμεσο κόστος εξαρτάται από την χρονική διάρκεια των επιμέρους δραστηριοτήτων, και πιο συγκεκριμένα ότι το άμεσο κόστος είναι το άθροισμα του άμεσου κόστους AE_{ik} της κάθε δραστηριότητας που δίνεται από την σχέση

$$AE_{ik} = g_{ik}(p_{ik})$$

όπου AE_{ik} είναι το άμεσο κόστος της δραστηριότητας που αρχίζει στον κόμβο (γεγονός) i και τελειώνει στον γεγονός j , p_{ik} είναι η χρονική διάρκεια της δραστηριότητας, και g_{ik} είναι μία γνωστή συνάρτηση.

- Παρακάτω, παρουσιάζουμε ένα απλό παράδειγμα υπολογισμού του άμεσου κόστους στην περίπτωση δραστηριοτήτων όπου επιτρέπεται η υπερωριακή απασχόληση (μέσα στις παρενθέσεις παραθέτουμε αριθμητικό παράδειγμα).

Παράδειγμα Υπολογισμού Άμεσου Κόστους και Κόστους Συμπίεσης Δραστηριότητας :

Έστω:

ΑΕ= Αριθμός Εργατών της συγκεκριμένης Εργασίας (3)

ΔΕ= Κανονική Διάρκεια Εργασίας (σε ημέρες) (4)

ΚΑΩ= Κανονικός Αριθμός Ωρών Εργασίας την Ημέρα = (8 ώρες)

Ω=ωρομίσθιο (π.χ., 2000 δρχ την ώρα)

Υ=κόστος υπερωρίας (π.χ., 3000 δρχ την ώρα ή 150% του ημερομισθίου)

ΣΑΕΩ= Συνολικός Αριθμός Εργατοωρών

ΑΚ=Άμεσο Κόστος Εργασίας στην περίπτωση που η χρονική διάρκεια της εργασίας είναι ίση με ΔΕ.

Έχουμε ότι :
 $\Sigma \text{ΑΕΩ} = \text{ΑΕ} * \Delta \text{Ε} * \text{ΚΑΩ} = \text{ΑΕ} * \Delta \text{Ε} * 8 \quad (= 3 * 4 * 8 = 96)$
 $\text{ΑΚ} = \Sigma \text{ΑΕΩ} * \Omega \quad (= 96 * 2000 = 192000)$

Υπολογισμός Κόστους Συμπίεσης κατά 1 ημέρα

Έστω:
 $\text{ΑΩΗ} =$ Αριθμός Ωρών Εργασίας ανά εργάτη την κάθε ημέρα
 (συμπεριλαμβανομένων των υπερωριών)
 $\text{ΑΥ} =$ Αριθμός Υπερωριών ανά εργάτη
 $\text{ΑΚ1} =$ Άμεσο Κόστος σε περίπτωση συμπίεσης κατά μία ημέρα
 $\text{Κ1} =$ Κόστος Συμπίεσης κατά μία ημέρα

Έχουμε ότι :
 $\text{ΑΩΗ} = \Sigma \text{ΑΕΩ} / ((\Delta \text{Ε} - 1) * \text{ΑΕ}) \quad (96 / ((4 - 1) * 3) = 10.667)$
 $\text{ΑΥ} = \text{ΑΩΗ} - \text{ΚΑΩ} \quad (10.667 - 8 = 4.667)$
 $\text{ΑΚ1} = (\Delta \text{Ε} - 1) * \text{ΑΕ} * (\Omega * \text{ΚΑΩ} + \text{Υ} * \text{ΑΩΗ}) \quad (= 3 * 3 * (8 * 2000 + 4.667 * 3000) = 270000)$
 $\text{Κ1} = \text{ΑΚ1} - \text{ΑΚ} \quad (= 78000)$

Παρομοίως υπολογίζουμε τα κόστη συμπίεσης για 2, 3, κλπ, ημέρα

2.10 Βελτιστοποίηση κόστους του έργου

Το άθροισμα του συνολικού κόστους κάθε δραστηριότητας, όπως αναφέρθηκε παραπάνω, αποτελεί το συνολικό κόστος του έργου. Το άθροισμα όμως του βέλτιστου κόστους κάθε δραστηριότητας δεν αποτελεί το βέλτιστο κόστος του έργου, αλλά απλά το προσεγγίζει. Αυτό οφείλεται στο ότι για τον υπολογισμό του κόστους κάθε δραστηριότητας δεν λαμβάνονται υπόψη οι σχέσεις αλληλουχίας των δραστηριοτήτων του έργου. Το βέλτιστο κόστος μπορεί να υπολογιστεί με τη βοήθεια της τεχνικής CPM (Critical Path Method), ενώ ταυτόχρονα θα υπολογιστεί και η ελάχιστη διάρκεια του έργου. Η τεχνική αυτή απαιτεί μεγαλύτερη βεβαιότητα στους χρόνους από την PERT, καθώς και καλή οικονομική αποτίμηση των πόρων του έργου. Για την εφαρμογή την μεθόδου CPM απαιτείται για κάθε δραστηριότητα η γνώση:

- Της κανονικής της διάρκειας T , η οποία εκφράζει τον προβλεπόμενο χρόνο για την, κάτω από κανονικές συνθήκες, υλοποίησή της.
- Της ελάχιστης δυνατής της διάρκειας T_e , η οποία εκφράζει τον ελάχιστο δυνατό χρόνο υλοποίησης της.
- Το περιθώριο συμπίεσης το οποίο από την διαφορά $T - T_e$
- Το συνολικό άμεσο κόστος, το οποίο αναφέρεται στην κανονική της διάρκεια.
- Το άμεσο κόστος συμπίεσης, για κάθε χρονική μονάδα, δηλαδή το πόσο κοστίζει η μείωση της διάρκειας της δραστηριότητας κατά μία μονάδα χρόνου. Το κόστος αυτό μπορεί να είναι θετικό, αρνητικό ή μηδενικό.

Σημειώνουμε ότι το πρόβλημα της βελτιστοποίησης του κόστους του έργου, μπορεί να μοντελοποιηθεί σαν ένα πρόβλημα μη γραμμικού προγραμματισμού με περιορισμούς:

ΠΡΟΒΛΗΜΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΟΣΤΟΥΣ ΤΟΥ ΕΡΓΟΥ

$$z = \min f(S_N) - \sum g_{ik}(p_{ik})$$

υπό τους περιορισμούς (α-γ)

ΠΕΡΙΟΡΙΣΜΟΙ:

(α) προηγούμενων εργασιών

$$S_k - p_{ik} - S_i \geq 0 \quad \text{για κάθε } i, k$$

(β) ελάχιστης και μέγιστης διάρκειας

$$Tm_{ik} \geq p_{ik} \geq Te_{ik} \quad \text{για κάθε } i, k$$

(γ) μη αρνητικών μεταβλητών

$$p_{ik}, S_i > 0 \quad \text{για κάθε } i > 0, k$$

όπου, N είναι ο αριθμός που αντιστοιχεί στο τελευταίο γεγονός του έργου, $i=0$ δηλώνει το αρχικό γεγονός (έναρξη έργου), S_i δηλώνει την χρονική στιγμή που λαμβάνει χώρα το γεγονός i , και Tm_{ik} δηλώνει τη μέγιστη επιτρεπτή χρονική διάρκεια της αντίστοιχης δραστηριότητας. Παρατηρήστε, ότι στην περίπτωση που οι συναρτήσεις f , g_{ik} είναι γραμμικές συναρτήσεις το παραπάνω πρόβλημα γίνεται πρόβλημα γραμμικού προγραμματισμού.

2.11 Σύνθετα Προβλήματα Προγραμματισμού Έργων

Στις προηγούμενες παραγράφους αντιμετωπίσαμε προβλήματα χρονικού προγραμματισμού και βελτιστοποίησης κόστους έργων και παρουσιάσαμε απλές υπολογιστικά μεθόδους για την αντιμετώπιση των προβλημάτων αυτών. Στην συνέχεια θα αντιμετωπίσουμε πιο σύνθετα προβλήματα οργάνωσης παραγωγής και προγραμματισμού έργων, και θα παρουσιάσουμε μεθόδους για την είτε προσεγγιστική είτε υπό-βέλτιστη επίλυση των προβλημάτων αυτών. Ο βασικός στόχος του κεφαλαίου αυτού είναι να γίνει κατανοητό ότι προβλήματα οργάνωσης παραγωγής και προγραμματισμού έργων μπορούν να μοντελοποιηθούν σαν προβλήματα μαθηματικού προγραμματισμού (δηλαδή προβλήματα βελτιστοποίησης) και ότι στην πλειοψηφία αυτών των προβλημάτων, η μοντελοποίησή τους οδηγεί σε δύσκολα προς επίλυση προβλήματα μαθηματικού προγραμματισμού (π.χ., προβλήματα μικτού ακέραιου γραμμικού προγραμματισμού). Παρόμοιες καταστάσεις (δηλαδή μοντελοποίηση του προβλήματος σαν πρόβλημα μαθηματικού

προγραμματισμού) θα αντιμετωπίσουμε όταν θα ασχοληθούμε συγκεκριμένα με προβλήματα σχεδίασης παραγωγής, μοντέλα χωροταξικής διάταξης, προγραμματισμό παραγωγής, και προγραμματισμό απαιτήσεων υλικών. Δεδομένου του γεγονότος ότι στην πλειοψηφία των περιπτώσεων, τα μοντέλα μαθηματικού προγραμματισμού που προκύπτουν είναι δύσκολο να επιλυθούν στην γενική τους μορφή, θα παρουσιάσουμε υπό-βέλτιστους ή ευρεστικούς αλγόριθμους (κατάλληλα σχεδιασμένους για το συγκεκριμένο πρόβλημα με το οποίο ασχολούμαστε) οι οποίοι οδηγούν σε ικανοποιητικές λύσεις.

2.12 Προγραμματισμός Έργου με Περιορισμούς Διαχείρισης Πόρων

Το πρόβλημα που αντιμετωπίζουμε σε αυτήν την παράγραφο, είναι παρόμοιο με το πρόβλημα χρονικού προγραμματισμού ενός έργου του προηγούμενου κεφαλαίου, με τη διαφορά ότι εδώ εισάγουμε επιπλέον περιορισμούς διαχείρισης πόρων. Με άλλα λόγια, το πρόβλημα γίνεται τώρα πιο σύνθετο, αφού θα πρέπει να λάβουμε υπόψη και το γεγονός ότι κάθε δραστηριότητα απαιτεί μία συγκεκριμένη ποσότητα πόρων (π.χ., αριθμός εργατών, κλπ) σε κάθε χρονική στιγμή, δεδομένου ότι η συνολική διαθέσιμη ποσότητα πόρων είναι περιορισμένη.

Πιο συγκεκριμένα, έστω ότι έχουμε N δραστηριότητες (αριθμημένες από $1...j...N$) και K πόρους (αριθμημένους από $1...k...K$). Όπως και στην περίπτωση του Κεφαλαίου 2, κάποιες από τις δραστηριότητες απαιτούν το πέρας άλλων δραστηριοτήτων για να αρχίσει η εκτέλεση τους. Το σύνολο προηγούμενων δραστηριοτήτων μίας δραστηριότητας j ορίζει το σύνολο $P(j)$. Δηλαδή το σύνολο $P(j)$ είναι ένα σύνολο ακεραίων που αντιστοιχούν στους δείκτες των προηγούμενων δραστηριοτήτων της δραστηριότητας j . Αν το $P(j)$ είναι κενό (άδειο), η δραστηριότητα j είναι «αρχική δραστηριότητα». Στη συνέχεια, με S_j συμβολίζουμε τον χρόνο έναρξης και με C_j το χρόνο περάτωσης της δραστηριότητας j . Στόχος μας είναι να ελαχιστοποιήσουμε τον χρόνο περάτωσης C_N , της μοναδικής τελικής δραστηριότητας [το N δεν θα ανήκει σε κανένα από τα σύνολα $P(j)$]. Έστω p_j η χρονική διάρκεια της δραστηριότητας j . Σ' αυτό το σημείο να αναφέρουμε ότι στην περίπτωση που δεν γνωρίζουμε ποια είναι η μοναδική τελική δραστηριότητα του προβλήματος, εισάγουμε μία πλασματική (την δραστηριότητα $N+1$), της οποίας το σύνολο προηγούμενων δραστηριοτήτων $P(N+1)$ περιέχει όλες τις άλλες (N το πλήθος) και έχει μηδενική διάρκεια. Η δραστηριότητα j καλείται ενεργή (active) τη χρονική στιγμή t αν και μόνο αν ισχύει: $S_j \leq t \leq C_j$ και συμβολίζουμε $j \in A_t$ αν η δραστηριότητα j είναι ενεργή τη χρονική στιγμή t . Η αμετάβλητη και σταθερή διαθέσιμη ποσότητα του πόρου k συμβολίζεται ως RA_k . Η δραστηριότητα j χρησιμοποιεί τον πόρο k σε βαθμό (ποσοστό χρήσης) r_{jk} , για όσο χρόνο αυτή είναι ενεργή. Η συνολική χρήση (εκμετάλλευση) κάθε ενός εκ των πόρων δεν πρέπει να υπερβεί την

διαθεσιμότητα του σε οποιαδήποτε χρονική στιγμή.

Πρόταση: Μια βέλτιστη λύση του προβλήματος, που παρουσιάστηκε παραπάνω, μπορεί να βρεθεί λύνοντας το παρακάτω πρόβλημα μαθηματικού προγραμματισμού:

$$\begin{aligned}
 &\min C_N \\
 &\text{u.π. } S_j \geq 0 && \text{αν } P(j) \text{ είναι κενό} \\
 &S_j \geq C_i && i \in P(j), \text{ για όλα τα } j \\
 &C_j = S_j + p_j && \text{για όλα τα } j \\
 &j \in A_i \leftrightarrow S_j \leq t \leq C_j && \text{για όλα τα } j, t \\
 &\sum_{j \in A_i} r_{jk} \leq RA_k && \text{για όλα τα } k
 \end{aligned}$$

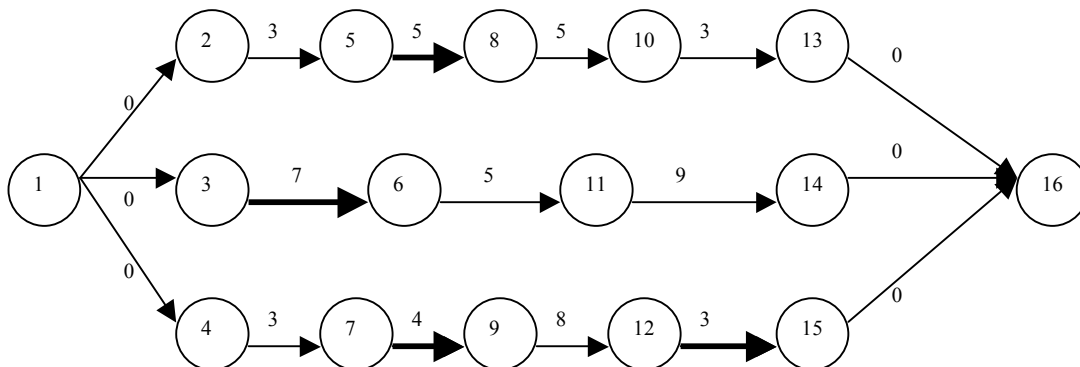
Αυτή η ειδική διατύπωση είναι εύκολο να κατανοηθεί διαισθητικά σαν μια απλή επέκταση του αντίστοιχου προβλήματος χωρίς περιορισμούς. Σ' αυτό το σημείο θα ασχοληθούμε με μερικές διαφοροποιήσεις. Μπορεί να δειχθεί ότι το παραπάνω πρόβλημα μπορεί να μετασχηματιστεί σε πρόβλημα μικτού ακέραιου γραμμικού προγραμματισμού.

Αν το r_{jk} είναι 1 για έναν μόνο πόρο, έχουμε ένα πρόβλημα διαχείρισης έργου με μηχανές σαν πόρους. Αν κάποιες εργασίες έχουν ξεχωριστούς αντικειμενικούς στόχους, όπως μεγάλη σπουδαιότητα (σημαντικότητα) ροής ή βραδύτητας, είναι απαραίτητο να μετατρέψουμε τους αντικειμενικούς στόχους, σαν άθροισμα των συναρτήσεων καθυστέρησης της τελικής δραστηριότητας για κάθε εργασία.

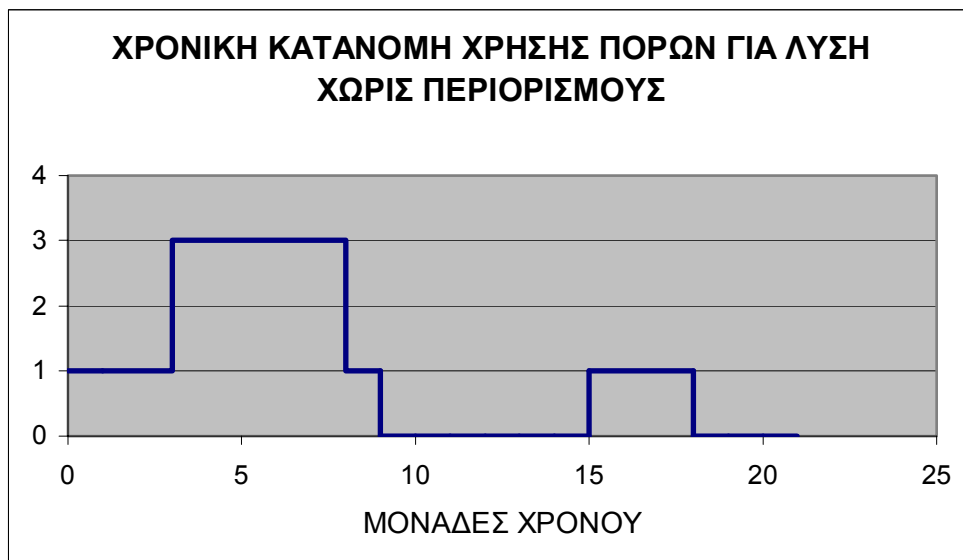
Ας προσπαθήσουμε τώρα να λύσουμε, προσεγγιστικά, το παραπάνω πρόβλημα. Αρχικά αγνοούμε τους περιορισμούς πόρων και λύνουμε το απλό πλέον πρόβλημα χωρίς περιορισμούς. Γι' αυτή τη λύση μπορούμε για κάθε πόρο να πάρουμε ένα διάγραμμα χρόνου-πόρου, δείχνοντας έτσι πότε χρησιμοποιούμε περισσότερο από το διαθέσιμο και πότε λιγότερο.

Για παράδειγμα, θεωρούμε το πρόβλημα ενός πόρου του σχήματος 2.3. Έχουμε έναν μόνο πόρο με διαθεσιμότητα 1.0. Όλες οι δραστηριότητες είτε χρησιμοποιούν τον πόρο σε ποσό 1.0 είτε δεν τον χρησιμοποιούν. Το παράδειγμα παρουσιάζεται σαν δίκτυο όπως και οι αντίστοιχοι χρόνοι. Με πιο έντονες γραμμές είναι οι δραστηριότητες που ανταγωνίζονται για τον πόρο.

Σχήμα 2.3 Λύση του προβλήματος χωρίς περιορισμούς. Εντός των κύκλων παρουσιάζεται ο αριθμός των δραστηριοτήτων



Η λύση χωρίς τους περιορισμούς έχει διάρκεια 21 και δίνει το διάγραμμα χρήσης (εκμετάλλευσης) του πόρου (Σχήμα 2.4). Όπως είναι φανερό έχουμε υπερεκμετάλλευση του πόρου από τη χρονική στιγμή 3 μέχρι τη χρονική στιγμή 7 από τις δραστηριότητες (5,8), (3,6) και (7,9). Είναι φανερό ότι προσθέτοντας τους περιορισμούς, υποχρεώνοντας έτσι τις δραστηριότητες να μην λαμβάνουν χώρα ταυτόχρονα, θα παίρναμε τη βέλτιστη λύση αν δοκιμάζονταν και οι έξι συνδυασμοί.



Σχήμα 2.4 Διάγραμμα χρήσης του πόρου.

Μία πιθανή ευρετική μέθοδος για την λύση αυτή αποτελεί η παρακάτω:

Βήμα 1: Καθόρισε πρώτα χρονικά την δραστηριότητα (3,6) ώστε να αποφευχθεί ο μεγαλύτερος άεργος χρόνος

Βήμα 2: Καθόρισε χρονικά τη δραστηριότητα (7,9), για την οποία ο υπολειπόμενος χρόνος, αυτής και των επόμενων της, είναι μεγαλύτερος από τον αντίστοιχο χρόνο της (5,8).

Η λύση μ' αυτόν τον τρόπο αφορά χρονική διάρκεια 24^{ω} χρονικών μονάδων. Είναι εύκολο να επαληθεύσουμε ότι αυτή είναι η πραγματικά βέλτιστη. Γενικά η ιδέα είναι να χρησιμοποιήσουμε την κατανομή των πόρων προκειμένου να εντοπίσουμε το «δύσκολο» χρονικό όριο της αντίστοιχης χρονικής περιόδου. Στη συνέχεια ευρετικά μεταφέρουμε (κινούμε) τις δραστηριότητες νωρίτερα ή αργότερα προκειμένου να «εξαπλώσουμε» το όριο σε ένα μεγαλύτερο χρονικό διάστημα και τελικά να το ελαχιστοποιήσουμε, με παράλληλη μέγιστη εκμετάλλευση (χρήση) του πόρου ενώ το έργο καθυστερεί όσο το δυνατόν λιγότερο.

Οι παραπάνω ιδέες με τη βοήθεια ηλεκτρονικού υπολογιστή μπορούν να τυποποιηθούν σε μοντέλα ευρετικών μεθόδων με παράλληλη αλληλεπίδραση του χρήστη. Έχουμε λοιπόν την παρακάτω γενική μορφή αλγόριθμου:

Βήμα 1: Δώσε τα δεδομένα του προβλήματος

Βήμα 2: Ο υπολογιστής δίνει τη λύση του χωρίς περιορισμούς πόρων προβλήματος.

Βήμα3: Ο υπολογιστής δίνει τη χρονική κατανομή των πόρων και διαπιστώνει αν είναι εφικτή.

Βήμα 4: Αν η αναζήτηση τελείωσε πηγαινε στο βήμα 7.

Βήμα5: Ο υπολογιστής προτρέπει τον χρήστη να προσθέσει / αφαιρέσει νέους περιορισμούς προτεραιότητας.

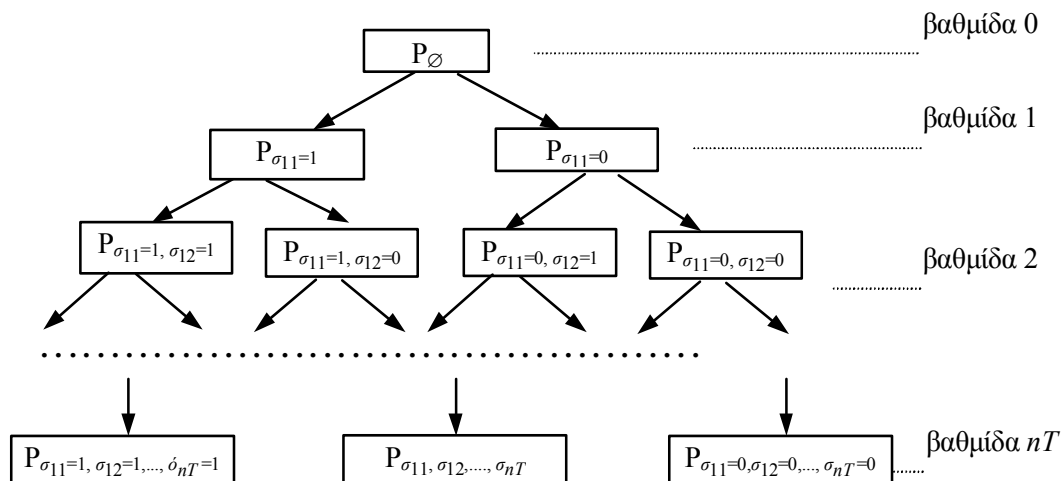
Βήμα 6: Πήγαινε στο βήμα 2

Βήμα 7: Δείξε και / ή τύπωσε τις καλύτερες και εφικτές λύσεις.

Σε μερικές περιπτώσεις είναι χρήσιμο να προστίθενται περιορισμοί της μορφής: η δραστηριότητα να μην ξεκινάει πριν την χρονική στιγμή 48. Αυτό εύκολα πραγματοποιείται προσθέτοντας μία δραστηριότητα που δεν χρησιμοποιεί πόρους διάρκειας 48 χρονικών μονάδων και χωρίς περιορισμούς προαπαιτούμενων δραστηριοτήτων. Τότε προστίθεται ένας περιορισμός προτεραιότητας υποχρεώνοντας την πραγματική δραστηριότητα να ξεκινάει μετά το πέρας αυτής που δεν χρησιμοποιεί πόρους.

2.13 Μέθοδος Διακλάδωσης και Φράγματος (branch-and-bound method)

Μία βολική ιδιότητα προβλημάτων βελτιστοποίησης είναι η συνέχεια των μεταβλητών αποφάσεως. Πράγματι, σε αυτήν την περίπτωση, το πρόβλημα υπολογισμού του ακρότατου μίας κυρτής συνάρτησης $f(x)$ ανάγεται στην επίλυση της εξίσωσης $df(x)/dx=0$. Αντίθετα, το πρόβλημα προγραμματισμού παραγωγής είναι δυσεπίλυτο όταν περιλαμβάνει τις ακέραιες (μη συνεχείς) μεταβλητές αποφάσεως σ_{it} . Μία μέθοδος επίλυσης τέτοιων προβλημάτων είναι η μέθοδος διακλάδωσης και φράγματος.



Σχήμα 2.5 Δένδρο αποφάσεων: αρχικό, ενδιάμεσα και τελικά προβλήματα βελτιστοποίησης.

Η μέθοδος διακλάδωσης και φράγματος (ΔΦ) εφαρμόζεται σε πολλούς τύπους συνδυαστικών προβλημάτων. Η ιδέα της *διακλάδωσης* είναι να ιδεί κανείς το πρόβλημα σαν δένδρο αποφάσεων. Κάθε κόμβος του δέντρου αντιστοιχεί σε μία σειρά "ειλημμένων αποφάσεων" για τις τιμές κάποιων σ_{it} . Από κάθε κόμβο αναφύονται νέα κλαδιά που αντιστοιχούν σε νέες αποφάσεις $\sigma_{i't'}=0$ και $\sigma_{i't'}=1$ για (i', t') που δεν έχουν ληφθεί ακόμη και οδηγούν σε κατάντεις κόμβους που αντιστοιχούν σε πιο πλήρεις σειρές αποφάσεων.

Η εξέταση όλων των κόμβων ωστόσο θα ισοδυναμούσε με την πλήρη αξιολόγηση όλων των δυνατών αποφάσεων και αυτό έχει μεγάλο υπολογιστικό κόστος. Πράγματι, στην τελευταία βαθμίδα, το πλήθος των προβλημάτων που πρέπει να επιλυθούν είναι 2^{nT} .

Παρατηρήστε ότι αν όλα τα σ_{it} είναι γνωστά τότε το πρόβλημα (2.1) λύνεται με τη μέθοδο simplex η οποία είναι βολική. Ακόμη και έτσι όμως, για $n=2$ προϊόντα και $T=12$ μήνες, θα πρέπει να επιλύσουμε περίπου $2^{24} \approx 17$ εκατομμύρια προβλήματα γραμμικού προγραμματισμού.

Η μέθοδος ΔΦ περιλαμβάνει τη διαδικασία *φραγμού* με την οποία διαγράφονται αρκετά ημιτελή προβλήματα (βαθμίδων $< nT$) και τα παρακλάδια τους και έτσι μειώνεται το υπολογιστικό φορτίο. Μία σχηματική παράσταση

του δένδρου αποφάσεων φαίνεται στο Σχ. 2.5.

Η μέθοδος ΔΦ βασίζεται στις ακόλουθες παρατηρήσεις:

A) *Πρόταση*: Για τα ακόλουθα προβλήματα ακέραιου προγραμματισμού

$P_{\Sigma}: \quad z_{\Sigma} = f(\Sigma) = \min f(\sigma_{11}, \dots, \sigma_{it}, \dots)$ υ.π. $\sigma_{it} \in \{0, 1\}$, για κάθε i, t	$P_{\Sigma'}: \quad z_{\Sigma'} = f(\Sigma') = \min f(\sigma_{11}, \dots, \sigma_{it}, \dots)$ υ.π. $0 \leq \sigma_{it} \leq 1$, για κάθε i, t
--	--

ισχύει ότι $z_{\Sigma'} \leq z_{\Sigma}$. Γενικότερα, αν σε πρόβλημα βελτιστοποίησης *χαλαρώσουν* ένας ή περισσότεροι περιορισμοί, τότε *βελτιώνεται* η αντικειμενική συνάρτηση.

Απόδειξη. Υποθέστε ότι δεν ισχύει η πρόταση. Ήτοι $z_{\Sigma'} > z_{\Sigma}$. Τότε το $z_{\Sigma'}$ δεν μπορεί να είναι το ελάχιστο (min) του προβλήματος $P_{\Sigma'}$, αφού υπάρχει ένας συνδυασμός Σ που αποτελείται από δυαδικές μεταβλητές σ_{it} ο οποίος δίνει μικρότερη τιμή z_{Σ} της συνάρτησης f και ταυτοχρόνως ικανοποιεί τους περιορισμούς $0 \leq \sigma_{it} \leq 1$, για κάθε i, t , του προβλήματος $P_{\Sigma'}$. Το ίδιο αποτέλεσμα ισχύει ακόμη και αν η f είναι συνάρτηση όχι μόνο των δυαδικών σ_{it} αλλά και άλλων *συνεχών* μεταβλητών αποφάσεως (π.χ. $f(\sigma_{it}, X_{it}, Y_t, \dots)$) όπως στην προηγούμενη παράγραφο).

B) Επιστρέφουμε τώρα στο δένδρο αποφάσεων του Σχ. 2.5. Θεωρούμε το αρχικό πρόβλημα P_{\emptyset} , όπου αντί των περιορισμών (Ζ) έχουμε τους περιορισμούς

(Ζ') $0 \leq \sigma_{it} \leq 1$, για κάθε i και t .

Συμβολίζουμε το πρόβλημα αυτό $P_{\emptyset'}$. Το $P_{\emptyset'}$ ονομάζεται *χαλαρό* του P_{\emptyset} επειδή οι περιορισμοί (Ζ') είναι λιγότερο αυστηροί από τους (Ζ). Τότε

- | |
|--|
| (i) το νέο πρόβλημα $P_{\emptyset'}$ είναι ένα κλασσικό πρόβλημα γραμμικού προγραμματισμού, και
(ii) η λύση $z_{\emptyset'}$ του $P_{\emptyset'}$ έχει τιμή <i>μικρότερη</i> ή, στη χειρότερη περίπτωση, <i>ίση</i> με την λύση z_{\emptyset} του αρχικού προβλήματος P_{\emptyset} (η απόδειξη προκύπτει αμέσως από την παρατήρηση Α). |
|--|

Γ) Ας θεωρήσουμε τώρα κάποιο από τα ενδιάμεσα προβλήματα του Σχ. 2.5. Έστω Σ ένα σύνολο που περιέχει μερικά από τα σ_{it} για τα οποία οι τιμές είναι *καθορισμένες* (0 ή 1). Για παράδειγμα $\Sigma = \{\sigma_{11}=0, \sigma_{12}=0\}$. Το πρόβλημα P_{Σ} είναι ένα πρόβλημα μικτού ακεραίου προγραμματισμού του οποίου η βέλτιστη λύση z_{Σ} είναι *χειρότερη (μεγαλύτερη)* ή, το πολύ, *ίση* με την λύση z_{\emptyset} του αρχικού προβλήματος που θέλουμε να λύσουμε. Αυτό συμβαίνει επειδή οι τιμές $\sigma_{11}=0, \sigma_{12}=0$ του Σ που έχουμε καθορίσει δεν είναι απαραίτητα οι

βέλτιστες. Αν συμβεί να είναι οι βέλτιστες, τότε το πρόβλημα P_Σ που προέκυψε από διακλαδώσεις θα έχει ίσο κόστος με το z_\emptyset . Άρα:

(iii) Καθώς αυξάνεται η βαθμίδα από $0 \rightarrow 1 \rightarrow 2 \dots$ τα προβλήματα που προκύπτουν θα δίδουν ολοένα και μεγαλύτερο ή, το πολύ, ίσο κόστος από τη λύση του προβλήματος της αμέσως ανώτερης βαθμίδας. Την ίδια ιδιότητα ικανοποιούν και τα χαλαρά προβλήματα των βαθμίδων αυτών.

Θεωρήστε τώρα τις δυαδικές μεταβλητές σ_{it} του προβλήματος P_Σ οι οποίες *δεν έχουν καθορισθεί ακόμη*. Αν χαλαρώσουμε τους περιορισμούς (ζ) για αυτές τις μεταβλητές, όπως και πριν, θα προκύψει το χαλαρό πρόβλημα P_Σ' με λύση $z_\Sigma' \leq z_\Sigma$. Η ιδιότητα αυτή είναι συνέπεια της πρότασης (ii). Οι λύσεις των χαλαρών προβλημάτων είναι *κάτω φράγματα* των λύσεων των αντίστοιχων δυαδικών.

Δ) Τώρα μπορούμε να δώσουμε την ιδέα της μεθόδου ΔΦ. Αν υποθέσουμε ότι, με κάποιες διακλαδώσεις, έχουμε φθάσει σε ένα πρόβλημα της *τελευταίας* βαθμίδας (nT), έστω το P_S , όπου S είναι ένα *πλήρες σύνολο* τιμών σ_{it} . Το πρόβλημα αυτό επιλύεται με τη μέθοδο simplex κατά τα γνωστά και έστω ότι έχει λύση z_S . Σημειώστε ότι η λύση z_S είναι μία *εφικτή λύση* του z_\emptyset γιατί ικανοποιεί τους περιορισμούς (ϵ) και, επομένως και, τους (ζ). Ας υποθέσουμε ότι υπάρχουν μερικά ενδιάμεσα προβλήματα P_Σ με (χαλαρά) κάτω φράγματα z_Σ' , τα οποία πρέπει να διακλαδίσουμε περαιτέρω. Τότε

(iv) αν $z_\Sigma' \geq z_S$, *διαγράφουμε* τα P_Σ , αφού, λόγω της πρότασης (iii), θα έδιναν χειρότερα αποτελέσματα ακόμη και αν τα διακλαδίζαμε (και στα χαλαρά και στα αυστηρά παρακλάδια τους).

Στην πρόταση αυτή οφείλεται η οικονομία της μεθόδου ΔΦ.

Ο αλγόριθμος ΔΦ συντηρεί μία λίστα προβλημάτων που εκκρεμούν. Στη συνέχεια περιγράφουμε μία παραλλαγή του.

Αλγόριθμος ΔΦ με έρευνα βέλτιστου (best-first search)

- (0) Θέσε $S=\emptyset$. Λύσε το χαλαρό πρόβλημα P_S' .
- (1) Αποθήκευσε τα αποτελέσματα του προβλήματος P_S' στη λίστα.
Αν η λύση του, z_S' , συμβεί να περιλαμβάνει *μόνον* δυαδικές μεταβλητές σ_{it} τότε
 - (i) αυτή είναι μία εφικτή λύση του P_\emptyset
 - (ii) ψάξε στη λίστα και *διάγραψε* όλα τα άλλα προβλήματα των οποίων οι λύσεις είναι μεγαλύτερες από την z_S'
 - (iii) αν η λίστα μείνει με το πρόβλημα P_S' *μόνον*, τότε ΠΕΡΑΤΩΣΕ ΤΟΝ ΑΛΓΟΡΙΘΜΟ με $z_\emptyset=z_S'$.
- (3) Ψάξε στη λίστα και βρες το χαλαρό πρόβλημα με το μικρότερο κόστος (*έρευνα βέλτιστου*). Ονόμασέ το Σ . Διάλεξε μία μεταβλητή σ_{it} που έχει *μη-δυαδική* τιμή ($0 < \sigma_{it} < 1$). Η επιλογή μπορεί να γίνει τυχαία ή με ανάλυση ευαισθησίας του πίνακα simplex. Θεώρησε τώρα τα σύνολα $S_0=\Sigma \cup \{\sigma_{it}=0\}$ και $S_1=\Sigma \cup \{\sigma_{it}=1\}$ και λύσε τα αντίστοιχα χαλαρά προβλήματα. Διάγραψε το P_S' από τη λίστα.
- (4)
 - (i) Θεωρώντας $S=S_0$ εκτέλεσε το βήμα (1),
 - (ii) θεωρώντας $S=S_1$ εκτέλεσε το βήμα (1),
 - (iii) πήγαινε στο βήμα (3).

Παρατηρούμε ότι ο αλγόριθμος ΔΦ **δεν** αναπτύσσει το δένδρο υποπροβλημάτων όπως στο Σχ. 2.5. Στο βήμα (3), αν υπάρχουν πολλές μη δυαδικές μεταβλητές, η επιλογή γίνεται τυχαία ή με ανάλυση ευαισθησίας της simplex. Στη δεύτερη περίπτωση, υπολογίζουμε την ευαισθησία της αντικειμενικής συνάρτησης ως προς τους περιορισμούς $\sigma_{it} \geq 0$ και $\sigma_{it} \leq 1$ για κάθε i, t . Από αυτήν ευρίσκουμε τη μεταβλητή εκείνη που δεν είναι δυαδική αλλά αν την αναγκάσαμε να πάρει τιμή 0 ή 1 θα είχαμε τη *μικρότερη* επιβάρυνση στο κόστος. Διακλαδίζοντας ως προς αυτή τη μεταβλητή προκύπτουν τα σύνολα S_0 και S_1 .

ΚΕΦΑΛΑΙΟ 3

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

3.1 Ο Αλγόριθμος

Το βασικό περίγραμμα του αλγορίθμου αποτυπώνεται στα εξής τρία επόμενα βήματα :

Βήμα 0:

Ξεκίνα διαβάζοντας τα δεδομένα του προς εξέταση έργου και φτιάξε μια λύση λαμβάνοντας υπ' όψη μόνον τους χρονικούς περιορισμούς προηγούμενων εργασιών. Δηλαδή μια λύση κατά την οποία όταν υπάρχει εξάρτηση μεταξύ των εργασιών η αρχή της εξαρτώμενης εργασίας θα τοποθετείται στο τέλος της εργασίας από την οποία εξαρτάται. Θεώρησε τη λύση αυτή σαν τρέχουσα πάνω στην οποία θα γίνουν οι όποιες μετατροπές για να οδηγηθούμε σε μια εναλλακτική.

Βήμα 1:

Τώρα άλλαξε την αρχή κάθε εργασίας της τρέχουσας λύσης τυχαία κατά δt_i όπου $\delta t_i = \text{random} * \text{diverse}$. Φτιάξε πάλι το έργο έτσι ώστε να πληρούνται οι περιορισμοί προηγούμενων εργασιών. Το βήμα αναζήτησης μπορεί να είναι τυχαίο αλλά κάθε φορά θα αυξάνει, ξεκινώντας από +1 ή -1 που αντιπροσωπεύουν μεταβολές μια μέρα μετά ή πριν αντίστοιχα από την ημέρα έναρξης της τρέχουσας λύσης.

Αυτό το βήμα επανέλαβε το αρκετές φορές και βρες ποια λύση δίνει το καλύτερο σενάριο για αριθμό εργατών. Το κατά πόσο μια λύση θα θεωρείται ενδεδειγμένη θα φαίνεται από την τιμή μιας συνάρτησης που θα ονομάσουμε αντικειμενική.

Βήμα 2:

Σε αυτό το βήμα αύξησε το diverse, θεώρησε σαν τρέχουσα λύση την καλύτερη του προηγούμενου βήματος και πήγαινε να επανεκτελέσεις το βήμα 1. Συνέχισε μέχρι να μην υπάρχει βελτίωση στη λύση.

3.2 Ο Κώδικας

Ακολουθεί ο κώδικας σε γλώσσα προγραμματισμού MATLAB:

[illegible]

```

no_workers=[3;3;8;8;7;8;16;16;10;15;12;14;18;22;14;24;28;15;21;21;27;15;34;6;12;12
;18;16;22;26;16;35;18];
% play User's maximum number of workers
Max_workers=100;
% play User %
percentage=0.3;
% play User maximum allowable time
MMAX_time=200;

%random search step size
Diverse=0;
%Value of objective function
Objective=10000000;

start_t=zeros(N,1);
best_start_t=start_t;
first=0;
OBJ=zeros(no_trials,1);
for lloop=1:no_trials,

for loop=1:no_of_iterations,
    start_t=start_t+round(Diverse*randn(size(start_t)));
    checked=zeros(N,1);
    previous=previous;
    startt=zeros(N,1);
    arrange(startt);

    end_t=start_t+duration;
    clear Max_time
    Max_time=max(max(end_t));
    if first==0
        min_time=Max_time;
        first=1;
    end
    clear Time
    clear WORK
    for t=1:Max_time,
        WORK(t,1)=0;
        Time(t,1)=t;
        for i=1:N,
            if t>= start_t(i,1) & t<= end_t(i,1)
                WORK(t,1)=WORK(t,1)+no_workers(i,1);
            end
        end
    end
    clear N_workers
    N_workers=zeros(Max_time,1);
    for i=1:floor(percentage*Max_time),
        N_workers(i,1)=Max_workers*i/floor(percentage*Max_time);
    end

    N_workers(floor(percentage*Max_time)+1:floor((1-percentage)*Max_time),1)=
Max_workers;

    for i=floor((1-percentage)*Max_time):Max_time,

```

```

        alpha=Max_workers/(floor((1-percentage)*Max_time)-Max_time);
        beta=-alpha*Max_time;
        N_workers(i,1)=alpha*i+beta;
    end

    ss=0;
    if MMAX_time < max(max(end_t))
        ss=1;
    end
    obj=norm(WORK(1:Max_time,1)-N_workers(1:Max_time,1))/Max_time+weight*ss;
    if obj<Objective
        Objective=obj;
        best_start_t=start_t;
    end

end

start_t=best_start_t;
end_t=start_t+duration;
clear Time
clear WORK
Max_time=max(max(end_t));
for t=1:Max_time,
    WORK(t,1)=0;
    Time(t,1)=t;
    for i=1:N,
        if t>= start_t(i,1) & t<= end_t(i,1)
            WORK(t,1)=WORK(t,1)+no_workers(i,1);
        end
    end
end
clear N_workers
N_workers=zeros(Max_time,1);
for i=1:floor(percentage*Max_time),
    N_workers(i,1)=Max_workers*i/floor(percentage*Max_time);
end

N_workers(floor(percentage*Max_time)+1:floor((1-percentage)*Max_time),1)=
Max_workers;

for i=floor((1-percentage)*Max_time):Max_time,
    alpha=Max_workers/(floor((1-percentage)*Max_time)-Max_time);
    beta=-alpha*Max_time;
    N_workers(i,1)=alpha*i+beta;
end

ss=0;
if MMAX_time < max(max(end_t))
    ss=1;
end
obj=norm(WORK(1:Max_time,1)-N_workers(1:Max_time,1))/Max_time+weight*ss

OBJ(lloop,1)=obj;
Diverse=Diverse+0.1;
end

figure
hold

```

```

    for i=1:N,
        line([start_t(i,1) end_t(i,1)], [i i])
    end
    hold
    grid
    xlabel('time (days)');
    ylabel('task no');

print -dtiff fig1.tiff
figure

plot(Time, WORK(:,1), 'r')
hold;
plot(Time, N_workers(1:Max_time, 1));
hold
grid;
xlabel('time (days)');
ylabel('no of workers');

print -dtiff fig2.tiff

figure
plot(OBJ(:,1), 'r')

grid
    xlabel('trial no');
    ylabel('objective function');

print -dtiff fig3.tiff

```


3.2.1 Η Συνάρτηση Arrange

```
function arrange(startt)
global duration previous previousss N_workers no_workers start_t end_t

N=max(size(duration));

for i=1:N,
    if sum(previous(i,:))==0
        start_t(i,1)=startt(i,1);
        end_t(i,1)=start_t(i,1)+duration(i,1);
        checked(i,1)=1;
    end
end
while (sum(sum(previousss)))>0
for i=1:N,
    for j=1:N,
        if previous(i,j)==1
            if checked(j,1)==1
                previousss(i,j)=0;
                if start_t(i,1)<=end_t(j,1)
                    start_t(i,1)= end_t(j,1);
                end
                end_t(i,1)=start_t(i,1)+duration(i,1);
                checked(i,1)=1;
            end
        end
    end
end
end
end
```

3.3 Αναλυτική επεξήγηση του κώδικα

Επεξηγούμε αναλυτικά τον κώδικα γραμμή προς γραμμή:

```
clear all  
close all  
global duration previous previousss N_workers no_workers start_t end_t
```

Είναι η αρχή του κώδικα όπου με την εντολή "clear all" γίνεται εκκένωση της μνήμης από μεταβλητές ενώ με "close all" κλείνουμε όλα τα ενεργά παράθυρα που μπορεί να υπάρχουν. Με την εντολή "global" δηλώνουμε όλες εκείνες τις μεταβλητές οι οποίες επιθυμούμε να είναι ορατές δηλαδή αναγνωρίσιμες από όλους τους κώδικες.

```
% play user  
no_of_iterations=100;           ( Δήλωση αριθμού επαναλήψεων )  
  
% User weight  
weight=1000;  
  
%play User choose number of trials  
no_trials=30;                  ( Δήλωση αριθμού trials )
```

Στη συνέχεια δηλώνουμε κάποιες σταθερές μεταβλητές. Κατ' αρχάς δηλώνουμε τον αριθμό των επαναλήψεων που θέλουμε να τρέξει ο κώδικας, καθώς και τις δοκιμές που θα κάνει μέσα σε αυτές τις επαναλήψεις ο κώδικας ("iterations" και " trials"). Δηλώνουμε και ένα βάρος "weight" το οποίο θα λάβει μέρος στην αντικειμενική συνάρτηση, από την οποία θα αντιλαμβανόμαστε κατά πόσο είμαστε κοντά σε επιθυμητά αποτελέσματα.

Το σχόλιο " % play user " υποδηλώνει στο χρήστη του προγράμματος να εισαγάγει την επιθυμητή τιμή.

```
% User Number of tasks  
N=33;  
% User duration of tasks  
duration=[1;5;3;3;11;14;3;4;8;10;13;5;15;13;5;9;19;17;7;6;6;8;13;7;6;5;  
7;13;12;8;21;14;12];          (Πίνακας Στήλη [33,1])
```

Τώρα πλέον εισάγουμε τα πραγματικά δεδομένα του έργου μας. Ζητείται από τον χρήστη να δηλώσει τον αριθμό των εργασιών N που απαρτίζουν το έργο του και σε έναν πίνακα στήλη [N, 1] τη διάρκεια κάθε εργασίας αντίστοιχα.

προγραμματισμού είναι “Finish to Start”).Ενώ, με το μηδενικό ψηφίο δηλώνεται ότι δεν υπάρχει χρονική εξάρτηση μεταξύ των δύο εργασιών.

Για παράδειγμα στον παρακάτω πίνακα :

No of task	1	2	3
1	0	0	0
2	1	0	0
3	0	1	0

Η εργασία task1 δεν έχει εξάρτηση από καμία εργασία (λογικό αφού βρίσκεται στην αρχή του έργου).Η δεύτερη εργασία task2 απαιτείται να έχει τελειώσει η πρώτη για να μπορεί να αρχίσει και τέλος το task3 δεν εξαρτάται από το task1 αλλά για να αρχίσει πρέπει να τελειώσει το task2.

Μετά απ αυτόν τον πίνακα δηλώνεται άλλος ένας πίνακας στήλη [N,1] ο οποίος εμπεριέχει τον αριθμό των εργατών που απασχολεί ημερησίως (ή για κάποια άλλη μονάδα χρόνου) κάθε εργασία για να ολοκληρωθεί σε χρονικές διάρκειες ίσες με αυτές που δηλώθηκαν στον Πίνακα “duration”.

```
% User maximum number of workers
```

```
Max_workers=130;
```

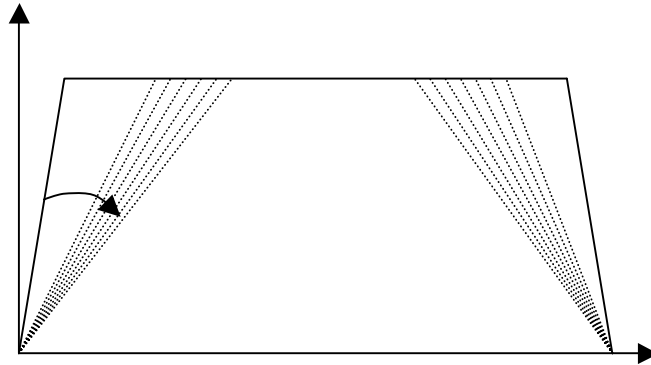
```
% play User %
```

```
percentage=0.4;
```

```
% play User maximum allowable time
```

```
MMAX_time=100;
```

Σε αυτό το σημείο του κώδικα ο εκάστοτε χρήστης του, εισάγει τα επιθυμητά δεδομένα για ένα διαφορετικό σενάριο πραγματοποίησης του έργου. Κατά σειρά επιλέγει πρώτα τον μέγιστο αριθμό εργατών που θα μπορούν να δουλεύουν ημερησίως (δηλ. το μέγιστο άθροισμα από εργάτες διαφορετικών εργασιών)και μετά το “percentage” μια μεταβλητή της οποίας η τιμή καθορίζει την κλίση των μη παράλληλων πλευρών ενός τραπεζίου, το οποίο επιθυμούμε να προσεγγίζει όσο το δυνατόν περισσότερο η γραφική παράσταση εργατικού δυναμικού-ημερών κι έτσι οι εργάτες να κατανέμονται ομαλά και σύμφωνα με τις απαιτήσεις του χρήστη. Το percentage μπορεί να παίρνει τιμές στο διάστημα (0, 0.5) και ενδεικτικά η λειτουργία του φαίνεται στο παρακάτω σχήμα. Καθώς το percentage αυξάνει, μικραίνει η τιμή της



Σχήμα 3.1

Σχήμα τραπεζίου που επηρεάζεται από το percentage.

```
%random search step size
Diverse=0;
%Value of objective function
Objective=10000000;

start_t=zeros(N,1);
best_start_t=start_t;
first=0;
OBJ=zeros(no_trials,1);
```

Δηλώνουμε το μέγεθος του τυχαίου βήματος αναζήτησης το οποίο αρχικά είναι πολύ μικρό, τείνει στο μηδέν και δίνουμε μια τιμή στην αντικειμενική συνάρτηση "objective function" πολύ μεγάλη. Δημιουργούμε ένα νέο πίνακα στήλη $[N, 1]$ με όλα τα στοιχεία μηδέν και τον ονομάζω "start_t" και αρχικοποιώ τη μεταβλητή "best_start_t" να δείχνει την τιμή "start_t". Κατόπιν δημιουργώ άλλον ένα πίνακα στήλη με όλα τα στοιχεία μηδενικά τον OBJ[no trials,1].

```
for lloop=1:no_trials,
for loop=1:no_of_iterations,
start_t=start_t+round(Diverse*randn(size(start_t)));
```

Ξεκινάμε να τρέξουμε τον κώδικα τόσες φορές όσες και ο αριθμός των "trials" που έχει δηλωθεί στην αρχή. Μέσα σ αυτόν θα τρέξει κι άλλος ένας υποκώδικας τόσες φορές όσο είναι το πλήθος των επαναλήψεων "iterations". Ο πίνακας στήλη "start_t" που ορίστηκε με στοιχεία μηδενικά τώρα παίρνει τυχαίες τιμές ενώ και η μεταβλητή diverse αυξάνεται κατά 0.1.

```
checked=zeros(N,1);
previous=previous;
startt=zeros(N,1);
arrange(startt);
```

Δημιουργούμε ένα αντίγραφο του πίνακα "previous" όπου φαίνονται οι εξαρτήσεις των εργασιών και τον ονομάζουμε "previous" καθώς επίσης και ένα αντίγραφο του πίνακα "start_t" τον οποίο και αρχικοποιούμε με μηδενικά και ονομάζουμε "startt". Στη συνέχεια τρέχω τη συνάρτηση "arrange" με όρισμα τον νέο αυτό πίνακα.

3.3.1 Αναλυτική Επεξήγηση της Συνάρτησης "Arrange"

```
function arrange(startt)
global duration previous previousss N_workers no_workers start_t end_t`
```

Αρχικά δηλώνουμε τη συνάρτηση "arrange" που δέχεται ως όρισμα τον μηδενικό πίνακα στήλη[N,1] startt καθώς και σαν μεταβλητές "global" όλες εκείνες τις μεταβλητές που θέλουμε να είναι κοινές να φαίνονται δηλαδή σε όλα τα αρχεία που χρησιμοποιεί ο κώδικας.

```
N=max(size(duration));
for i=1:N,
    if sum(previous(i,:))==0
        start_t(i,1)=startt(i,1);
        end_t(i,1)=start_t(i,1)+duration(i,1);
        checked(i,1)=1;
    end
end
```

Επιλέγεται η εργασία με τη μεγαλύτερη διάρκεια για να καθορίσει το πλήθος των επαναλήψεων της συνάρτησης και την ονομάζουμε N. Για 1 έως N λοιπόν φορές ελέγχουμε στον πίνακα των εξαρτήσεων αν υπάρχει κάποια γραμμή ή στήλη με μηδενικό άθροισμα που σημαίνει αυτόματα ότι σε εκείνη τη γραμμή ή τη στήλη δεν υπάρχει καμία εξάρτηση χρονική για την πραγματοποίηση της εργασίας. Εάν συμβαίνει αυτό τότε πήγαινε και βάλε στον πίνακα startt τις τιμές από τον πίνακα start_t. Τα "end_t", οι χρόνοι τέλους των εργασιών δηλαδή προκύπτουν από την αρχή συν τη διάρκεια της κάθε εργασίας.

```
while (sum(sum(previousss)))>0
    for i=1:N,
        for j=1:N,
            if previous(i,j)==1
                if checked(j,1)==1
                    previousss(i,j)=0;
```

Στη συνέχεια ελέγχουμε το άθροισμα όλων των στοιχείων του πίνακα "previous" (αντίγραφο του πίνακα των εξαρτήσεων) και το συγκρίνουμε με το 0. Πρέπει να είναι μεγαλύτερο γιατί αλλιώς δεν υπάρχει κανένας περιορισμός μεταξύ των εργασιών και μπορούμε να τις τοποθετήσουμε στο χρονικό διάγραμμα όπως επιθυμούμε χωρίς αντίκτυπο στο έργο. Εάν υπάρχει στοιχείο (i,j) που παίρνει την τιμή της μονάδας δηλαδή η εργασία i να έχει εξάρτηση από τη j και εάν παράλληλα το στοιχείο (j,1) είναι επίσης μονάδα τότε πήγαινε στον "previous" και μηδένισε το στοιχείο (i,j), καθώς τα δύο προηγούμενα γεγονότα είναι αδύνατον να συμβαίνουν ταυτόχρονα.

```

if start_t(i,1)<=end_t(j,1)
    start_t(i,1)= end_t(j,1);
end
end_t(i,1)=start_t(i,1)+duration(i,1);
checked(i,1)=1;
end
end
end
end

```

Επειδή υπάρχει εξάρτηση της εργασίας i από τη εργασία j χρονικά αυτό σημαίνει ότι δεν μπορεί η i να αρχίσει νωρίτερα από το τέλος της j. Εάν με τις τυχαίες μεταθέσεις και δοκιμές του κώδικα εμφανιστεί αυτό το σενάριο τότε εξίσωσε την αρχή της i και το τέλος της j, έτσι ώστε η αρχή της i να συμπίπτει με το τέλος της j. Τέλος, αφού τοποθετήσεις την εργασία i μετά το τέλος της j υπολόγισε και το χρονικό σημείο του τέλους της i που θα είναι το σημείο που αρχίζει συν τη διάρκειά της. Σε αυτό το σημείο τελειώνει η συνάρτηση "arrange" η οποία στην ουσία βλέπει αν υπάρχουν χρονικοί περιορισμοί μεταξύ των εργασιών και μόλις αυτό εξασφαλιστεί αρχίζει τις τυχαίες δοκιμές για να μεταθέσει τις αρχές των εργασιών και να μας δώσει εναλλακτικές προτάσεις για τη πραγματοποίηση του έργου.

(επιστροφή στο locate)

```

end_t=start_t+duration;
clear Max_time
Max_time=max(max(end_t));
if first==0
    min_time=Max_time;
    first=1;
end

```

Επιστρέφοντας στον κώδικα θέτουμε το χρόνο τέλους μιας εργασίας ως το άθροισμα της τιμής χρόνου εκκίνησης και της διάρκειας της εργασίας. Παράλληλα, μηδενίζω τη μεταβλητή "Max_time" και της δίνω τιμή ίση με το μέγιστο των "end times" των τιμών τέλους των εργασιών. Για τον πρώτο βρόγχο(όπου δηλώθηκε το first=0) εξισώνω τον μέγιστο χρόνο με τον

ελάχιστο. Κατόπιν, θέτω το `first=1` για να δηλώσω ότι πέρασα από τον πρώτο βρόγχο και ότι από εδώ και στο εξής δεν ισχύει η ισότητα "`min_time=max_time`".

```
clear Time
clear WORK
```

Μηδενίζω τις τιμές των μεταβλητών `time` και `work` έτσι ώστε να μην έχουν κρατήσει στη μνήμη παλιές τιμές.

```
for t=1:Max_time,
    WORK(t,1)=0;
    Time(t,1)=t;
    for i=1:N,
        if t>= start_t(i,1) & t<= end_t(i,1)
            WORK(t,1)=WORK(t,1)+no_workers(i,1);
        end
    end
end
```

Για `t=1` έως και τον μέγιστο χρόνο δημιουργώ τον πίνακα στήλη `WORK` ο οποίος έχει διαστάσεις `[Max_time,1]` με όλα του τα στοιχεία μηδενικά και τον πίνακα `Time` ιδίων διαστάσεων με τα στοιχεία του να λαμβάνουν την τιμή `t`. Τώρα για αριθμό επαναλήψεων ίσο με τον αριθμό των εργασιών του έργου ελέγχουμε αν το `t` βρίσκεται μεταξύ του χρονική στιγμή εκκίνησης και τέλους της κάθε εργασίας `i`. Αν αυτό επιβεβαιώνεται τότε τοποθετούνται στον πίνακα `WORK` οι εργάτες της κάθε εργασίας `i` που τελικά προκύπτει από το άθροισμα του μηδενικού πίνακα στήλη `WORK[t,1]` και του πίνακα "`no_workers`" `[i,1]` που δηλώνει ο χρήστης στην αρχή του κώδικα.

```
clear N_workers
N_workers=zeros(Max_time,1);
for i=1:floor(percentage*Max_time),
    N_workers(i,1)=Max_workers*i/floor(percentage*Max_time);
end
```

Στη συνέχεια γίνεται μηδενισμός της μεταβλητής `N_Workers` και δημιουργώ ένα πίνακα `N_Workers[Max_Time,1]` με μηδενικά. Στρογγυλοποιώ το γινόμενο `max_time * percentage` προς το μείον άπειρον και επαναλαμβάνω για τόσες φορές, όσες δηλαδή είναι και ο στρογγυλοποιημένος αριθμός, την ακόλουθη διαδικασία :

Κάθε εργασία `i` θα λάβει ως αριθμό εργατών το αποτέλεσμα που δίδεται από το γινόμενο `max_workers*i` διαιρούμενο δια τον παραπάνω στρογγυλοποιημένο αριθμό.


```

N_workers(floor(percentage*Max_time)+1:floor((1-percentage)*
Max_time),1) = Max_workers;
    for i=floor((1-percentage)*Max_time):Max_time,
        alpha=Max_workers/(floor((1-percentage)*Max_time)-Max_time);
        beta=-alpha*Max_time;
        N_workers(i,1)=alpha*i+beta;
    end

```

Ορίζω τον μέγιστο αριθμό εργατών βάσει του N_workers για ένα εύρος τιμών που δίδεται από τον τύπο:

Max_workers=

N_workers(floor(percentage*Max_time)+1:floor((1-percentage)*Max_time),1)

Τα alpha και beta που υπολογίζονται από του μαθηματικούς τύπους μας δίνουν τα χαρακτηριστικά των ευθειών των δύο μη παράλληλων πλευρών του τραπεζίου.

```

ss=0;
    if MMAX_time <max(max(end_t))
        ss=1;
    end

```

Αρχικοποιούμε μια μεταβλητή ss με την τιμή του μηδενός. Εάν το MMAX_time δηλαδή η τιμή του μέγιστου επιθυμητού χρόνου που εισαγάγει ο εκάστοτε χρήστης είναι μικρότερη του μεγίστου των χρόνων τέλους "end times" των εργασιών τότε το ss λαμβάνει την τιμή της μονάδας. Γίνεται στην ουσία ένας έλεγχος που θα χρησιμοποιηθεί στη συνέχεια.

```

obj=norm(WORK(1:Max_time,1)N_workers(1:Max_time,1))/
Max_time+weight*ss;

    if obj<Objective
        Objective=obj;
        best_start_t=start_t;
    end

```

Παίρνουμε όλα τα στοιχεία του πίνακα με τον αριθμό των εργατών από 1 έως "Max_Time" και τα αφαιρούμε από τα αντίστοιχα του WORK και όλα αυτά τα διαιρούμε με ένα γινόμενο Max_Time+weight*ss. Μετά παίρνουμε τη νόρμα του αποτελέσματος $NORM(x)=\max(\text{sum}(\text{abs}(X)))$. Το αποτέλεσμα "obj" αυτής της νόρμας μας δίνει την τιμή της αντικειμενικής συνάρτησης. Το "obj" συγκρίνεται με το "objective" που έχει οριοθετηθεί στην αρχή (Objective=10000000;) και αν βρεθεί μικρότερο εξισώνεται με το ανώτατο όριο. Τέλος, θέτουμε ως βέλτιστη τιμή έναρξης "best_start_t" τη νέα τιμή του "start_t" που προκύπτει.

```

end
start_t=best_start_t;
end_t=start_t+duration;
    clear Time
    clear WORK

```

Εδώ κλείνει ο βρόχος των επαναλήψεων "iterations" και την βέλτιστη τιμή που βρήκαμε παραπάνω την εξισώνουμε αντίστροφα τώρα με start_t για να τη χρησιμοποιήσουμε στην επόμενη loop. Θέτουμε το τέλος του έργου ως το άθροισμα των χρόνων έναρξης συν τη διάρκεια του έργου και ξαναδιαγράφουμε τις μεταβλητές "Time" και "Work" για να τους αποδώσουμε νέες τιμές.

```

OBJ(lloop,1)=obj;
Diverse=Diverse+0.1;
end

figure
hold
    for i=1:N,
        line([start_t(i,1) end_t(i,1)], [i i])
    end
    hold
    grid
    xlabel('time (days)');
    ylabel('task no');

print -dtiff fig1.tiff
figure

    plot(Time, WORK(:,1), 'r')
    hold;
    plot(Time, N_workers(1:Max_time, 1));
    hold
    grid;
    xlabel('time (days)');
    ylabel('no of workers');

print -dtiff fig2.tiff

figure
plot(OBJ(:,1), 'r')
grid
xlabel('trial no');
ylabel('objective function');

print -dtiff fig3.tiff

```

Εδώ κλείνει και βρόχος του αριθμού των δοκιμών `no_trials`. Δηλώνονται στη συνέχεια τα χαρακτηριστικά των γραφικών παραστάσεων που ακολουθήσουν σαν παρουσίαση του αποτελέσματος εφαρμογής του κώδικα. Για την πρώτη γραφική παράσταση για παράδειγμα, σχεδιάζουμε γραμμές που θα έχουν σαν αρχή και τέλος τις τιμές των πινάκων `start_t` και `end_t` και δίνουμε μια εντολή που να κρατάει ανοιχτό το παράθυρο για να μην αντικατασταθεί από τα επόμενα που θα ακολουθήσουν, αλλά αυτά να βγουν δίπλα του. Δηλώνουμε τα ονόματα των αξόνων των γραφικών παραστάσεων και στέλνουμε το γράφημα προς εκτύπωση στον εκτυπωτή. Ανάλογα ακολουθεί και το γράφημα εργατών-ημερών και τέλος το γράφημα της αντικειμενικής συνάρτησης συναρτήσει του αριθμού δοκιμών που θα μας δείχνει σε ποιο σημείο του κώδικα οδηγούμαστε σε καλές λύσεις.

ΚΕΦΑΛΑΙΟ 4

ΤΟ ΕΡΓΟ ΤΗΣ ΔΟΚΙΜΗΣ

4.1 Περιγραφή του έργου δοκιμής

Το έργο δοκιμής είναι ένα κομμάτι κατασκευής σύγχρονου αυτοκινητόδρομου πάνω στο ήδη υπάρχον οδικό δίκτυο. Επιλέξαμε ένα κατασκευαστικό έργο καθώς ανήκουν στην κατηγορία των έργων που συχνά παρατηρούνται χρονικές υπερβάσεις, χρήση μεγάλου εργατικού δυναμικού και συνεπώς κρίνεται αναγκαία η διαχείριση έργου. Είναι μια κρίσιμη δοκιμασία για τον κώδικα που προτείνουμε. Άλλωστε, πάντα τα κατασκευαστικά έργα λόγω κάποιων κρίσιμων εργασιών που εμπεριείχαν παρουσίαζαν ιδιαίτερη ευαισθησία. Το έργο αποτελείται από 29 κύριες εργασίες οι οποίες παρουσιάζονται στον παρακάτω πίνακα:

A/A	Περιγραφή Εργασίας
1	Αρχή του έργου
2	Διαμόρφωση γενικών συνθηκών
3	Δημιουργία εναλλακτικών δρόμων κίνησης
4	Τοποθέτηση πρόχειρων προστατευτικών δοκών
5	Αφαίρεση παλαιάς σήμανσης-τοποθέτηση καινούριας
6	Καταστροφή παλαιού οδοστρώματος
7	Εκσκαφή 1- Εκβραχισμός
8	Εκσκαφή 2
9	Εκσκαφή 3
10	Υποδομή πλευρικών υποστηριγμάτων Στάδιο Α΄
11	Υποδομή πλευρικών υποστηριγμάτων Στάδιο Β΄
12	Πλήρωση πλευρικών υποστηριγμάτων
13	Εκσκαφή για τοποθέτηση ηλεκτρικών
14	Ηλεκτρική Υποδομή – Προεγκατάσταση
15	Υδραυλικές – Αποχετευτικές Εγκαταστάσεις
16	Προετοιμασία υπεδάφους – Επιχωματώσεις
17	Προσθήκη ειδικού στρώματος πάχους 2΄΄
18	Προσθήκη ειδικού υλικού θερμομονωτικού υποστρώματος
19	Τελευταίο υπόστρωμα
20	Συμπύεση – Προετοιμασία για Ασφαλτόστρωση
21	Τσιμεντόστρωση Στάδιο Α΄
22	Τσιμεντόστρωση Στάδιο Β΄
23	Επίστρωση πίσσας
24	Ασφαλτόστρωση Α΄
25	Ασφαλτόστρωση Β΄
26	Τοποθετήσεις Σωληνώσεων – Αποχετευτικός Εξοπλισμός Ειδικές εργασίες
27	Πλάγια τοιχία Υποστήριξη
28	Βαφή - Οδική Σήμανση – Καλλωπισμός
29	Αφαίρεση Πρόχειρων Προστατευτικών Δοκών – Καθαρισμός – Παράδοση σε κυκλοφορία

Στο έργο υπάρχουν δύο ειδών περιορισμοί ανάμεσα στις εργασίες. Ο ένας περιορισμός είναι τύπου "Finish-to-Start"(FS) constraint δηλαδή η έναρξη μιας δραστηριότητας εξαρτάται από τη λήξη της προηγούμενης της και ο άλλος ήταν τύπου "Start-to-Start" (SS) δηλαδή η έναρξη μιας δραστηριότητας εξαρτάται από την έναρξη μιας άλλης. Αυτή η ορολογία χρησιμοποιείται κυρίως στα κομβικά δίκτυα κατά το χρονικό σχεδιασμό του έργου. Στο έργο της δοκιμής βέβαια οι εργασίες με περιορισμό "Start-to-Start" χρειάστηκαν τροποποίηση γιατί ο πίνακας "previous" των εξαρτήσεων που υπάρχει στο κώδικα αναγνωρίζει περιορισμούς "Finish-to-Start". Εύκολα όμως μπορούμε να περάσουμε από τον ένα τύπο στον άλλο δημιουργώντας περισσότερες εργασίες από μια υπάρχουσα. Δηλαδή για παράδειγμα, αν μια εργασία task2 έχει ως περιορισμό $SS(1,2) = 3$ σημαίνει ότι η εργασία task2 μπορεί να αρχίσει 3 ημέρες αφότου αρχίσει η εργασία task1. Αν τώρα η εργασία task1 έχει διάρκεια μεγαλύτερη των τριών ημερών μπορεί να διαιρεθεί σε δύο φάσεις. Η μια φάση της εργασίας task1a να διαρκεί τρεις μέρες και η άλλη task1b το υπόλοιπο της διάρκειας του task1. Έτσι, η εργασία task2 θα έχει πλέον ως περιορισμό $FS(1a,2)$ και θα μπορεί να αρχίσει με το πέρας της εργασίας task1a. Η μετατροπή που περιγράψαμε κρίνεται αναγκαία για τον κώδικα και ο χρήστης του θα πρέπει να την πραγματοποιεί πάντα πριν αρχίσει να εισάγει τα υπόλοιπα δεδομένα. Σχηματικά βέβαια στο διάγραμμα Gantt που ακολουθεί φαίνεται και η χρησιμότητά του.

			ΕΡΓΑΣΙΜΕΣ ΗΜΕΡΕΣ											
			1	2	3	4	5	6	7	8	9	10	11	
			Task1											
						Task2								
Μετά την μετατροπή Task1= task1a(3) + task1b(4)														
			task1a							task1b				
						Task2								

Μετά την δημιουργία δυο εργασιών από μια όπως φαίνεται και στο διάγραμμα αποδεσμεύουμε το υπόλοιπο task1b της εργασίας task1 το οποίο μπορεί να τοποθετηθεί χρονικά με άλλο τρόπο εφ' όσον το επιτρέπει βέβαια και η φύση της εργασίας. Και επειδή ο κώδικας χρησιμοποιεί αυτή ακριβώς τη λογική των μεταθέσεων πάνω σε ένα διάγραμμα Gantt για να μας δώσει μια εναλλακτική πρόταση για την πραγματοποίηση του έργου πρέπει να πράξαμε ομοίως για όλες τις εργασίες του έργου. Τελικά οι εργασίες έφθασαν να γίνουν 33 και για ευκολία συνοψίσαμε τα κύρια χαρακτηριστικά του έργου στον εξής παρακάτω πίνακα :

Ο Πίνακας με τα κύρια Χαρακτηριστικά του Έργου

TaskNo	Early start	Total previous	Constraints	Duration	Workers
1	1	0		1	3
2	2	1	FS1	5	3
3	2	1	FS1	3	8
4	5	2	FS3(1)	3	8
5	5	2	FS3(1)	11	7
6	5	2	FS3(1)	14	8
7	7	2	FS2(1)	3	16
8	10	3	FS7(2,1)	4	16
9	8	3	FS4(3,1)	8	10
10	8	3	FS4(3,1)	10	15
11	10	3	FS7(2,1)	13	12
12	10	3	FS7(2,1)	5	14
13	14	4	FS8(7,2,1)	15	18
14	16	4	FS9(4,3,1)	13	22
15	16	4	FS9(4,3,1)	5	14
16	16	4	FS9(4,3,1)	9	24
17	19	3	FS6(3,1)	19	28
18	21	5	FS15(9,4,3,1)	17	15
19	23	4	FS11(7,2,1)	7	21
20	30	5	FS19(11,7,2,1)	6	21
21	25	5	FS16(9,4,3,1)	6	27
22	29	5	FS13(8,7,2,1)	8	15
23	29	5	FS14(9,4,3,1)	13	34
24	30	5	FS19(11,7,2,1)	7	6
25	31	6	FS21(16,9,4,3,1)	6	12
26	31	6	FS21(16,9,4,3,1)	5	12
27	36	6	FS20(19,11,7,2,1)	7	18
28	37	6	FS22(13,8,7,2,1)	13	16
29	37	6	FS24(19,11,7,2,1)	12	22
30	39	6	FS18(15,9,4,3,1)	8	26
31	39	4	FS17(6,3,1)	21	16
32	42	6	FS23(14,9,4,3,1)	14	35
33	46	7	FS30(18,15,9,4,3,1)	12	18

Με βάση τους περιορισμούς φτιάχνουμε και τον πίνακα των εξαρτήσεων previous που θα χρησιμοποιήσει ο κώδικας για να μπορεί να δοκιμάζει νέες λύσεις. Ο συμβολισμός FS3(1) για την εργασία Νο 4 σημαίνει ότι για να αρχίσει πρέπει πρώτα να τελειώσει η εργασία Νο 3 αλλά και η εργασία Νο 1 επειδή η τελευταία είναι προηγούμενη της εργασίας Νο3.

Ο Πίνακας των εξαρτήσεων previous

[illegible]

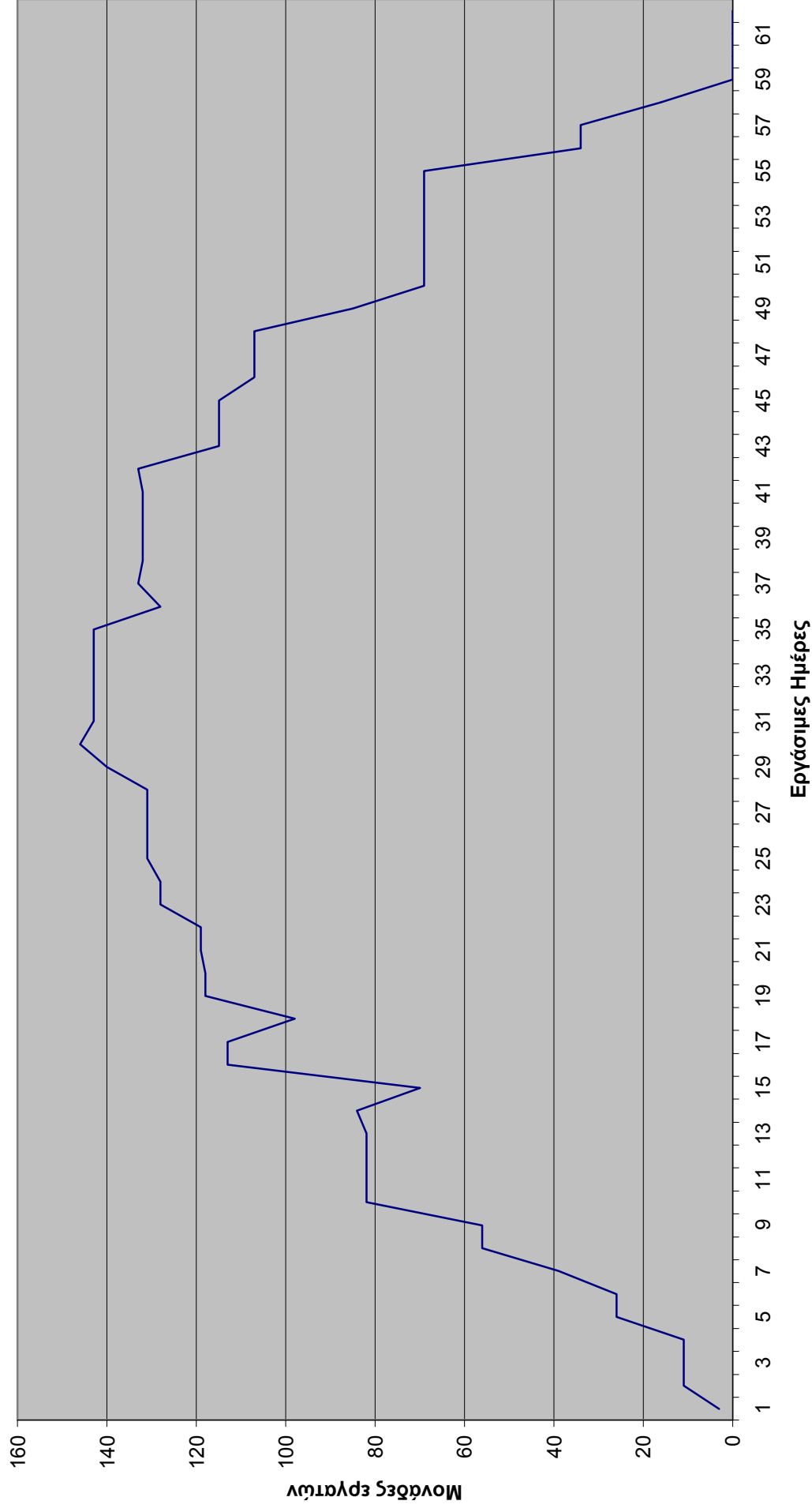
Όσον αφορά τους εργάτες θεωρήσαμε από εμπειρία ότι συγκεκριμένος αριθμός εργατών σε συγκεκριμένες ημέρες μπορεί να εκτελέσει ολοκληρωμένα μια εργασία. Το έργο αφού επιδέχτηκε μελέτες χρονικού προγραμματισμού και σύμφωνα με τις απαιτήσεις και προδιαγραφές της εργολήπτριας εταιρείας είχε σαν πιθανότερη εκδοχή αυτή του αρχείου [ergo.xls](#)

Διάγραμμα GANTT

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΡΓΑΣΙΩΝ

task	duration	early start	constraints	ΜΑΡΤΙΟΣ (ΕΡΓΑΣΙΜΕΣ ΗΜΕΡΕΣ)																			
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1		3																			
2	5	2	FS1	3	3	3	3																
3	3	2	FS1	8	8	8																	
4	3	5	FS3(1)	8	8	8																	
5	11	5	FS3(1)	7	7	7	7	7	7	7	7	7	7	7	7								
6	14	5	FS3(1)	8	8	8	8	8	8	8	8	8	8	8	8	8							
7	3	7	FS2(1)	16	16	16																	
8	4	10	FS7(2,1)				16	16	16	16	16												
9	8	8	FS4(3,1)				10	10	10	10	10	10	10	10									
10	10	8	FS4(3,1)				15	15	15	15	15	15	15	15	15								
11	13	10	FS7(2,1)				12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
12	5	10	FS7(2,1)				14	14	14	14	14												
13	15	14	FS8(7,2,1)																				
14	13	16	FS9(4,3,1)																				
15	5	16	FS9(4,3,1)																				
16	9	16	FS9(4,3,1)																				
17	19	19	FS6(3,1)																				
18	17	21	FS15(9,4,3,1)																				
19	7	23	FS11(7,2,1)																				
20	6	30	FS19(11,7,2,1)																				
21	6	25	FS16(9,4,3,1)																				
22	8	29	FS13(8,7,2,1)																				
23	13	29	FS14(9,4,3,1)																				
24	7	30	FS19(11,7,2,1)																				
25	6	31	FS21(16,9,4,3,1)																				
26	5	31	FS21(16,9,4,3,1)																				
27	7	36	FS20(19,11,7,2,1)																				
28	13	37	FS22(13,8,7,2,1)																				
29	12	37	FS24(19,11,7,2,1)																				
30	8	39	FS18(15,9,4,3,1)																				
31	21	39	FS17(6,3,1)																				
32	14	42	FS23(14,9,4,3,1)																				
33	12	46	FS30(18,15,9,4,3,1)																				
ΣΥΝΟΛΙΚΟΣ ΑΡΙΘΜΟΣ ΑΠΑΣΧΟΛΟΥΜΕΝΩΝ ΕΡΓΑΤΩΝ				3	11	11	11	26	26	39	56	56	82	82	82	82	84	70	113	113	98	118	119

Διάγραμμα Εργατικού Δυναμικού-Ημερών Έργου



ΚΕΦΑΛΑΙΟ 5

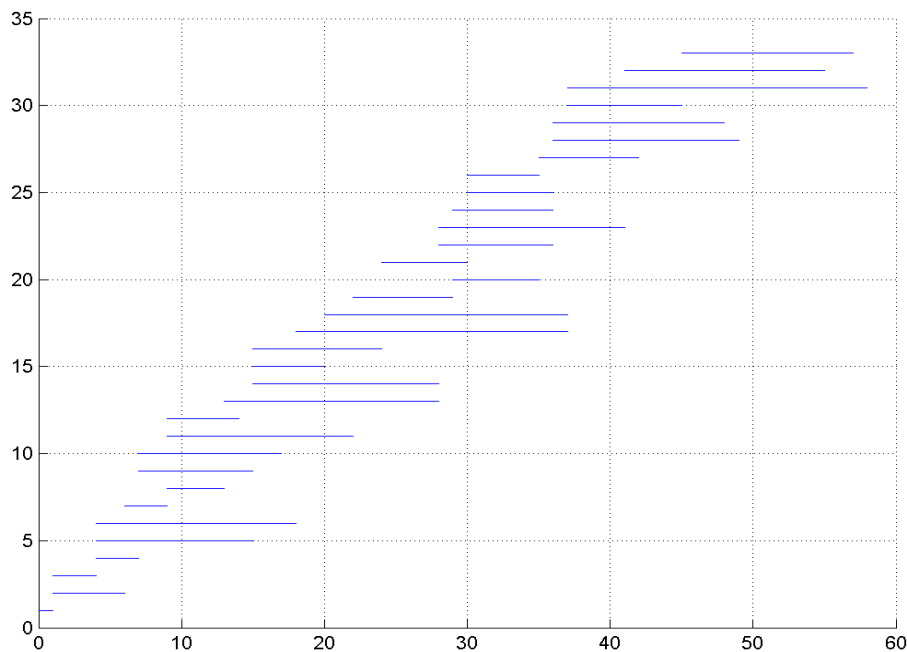
ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

5.1 Ενδεικτικά αποτελέσματα και γραφήματα

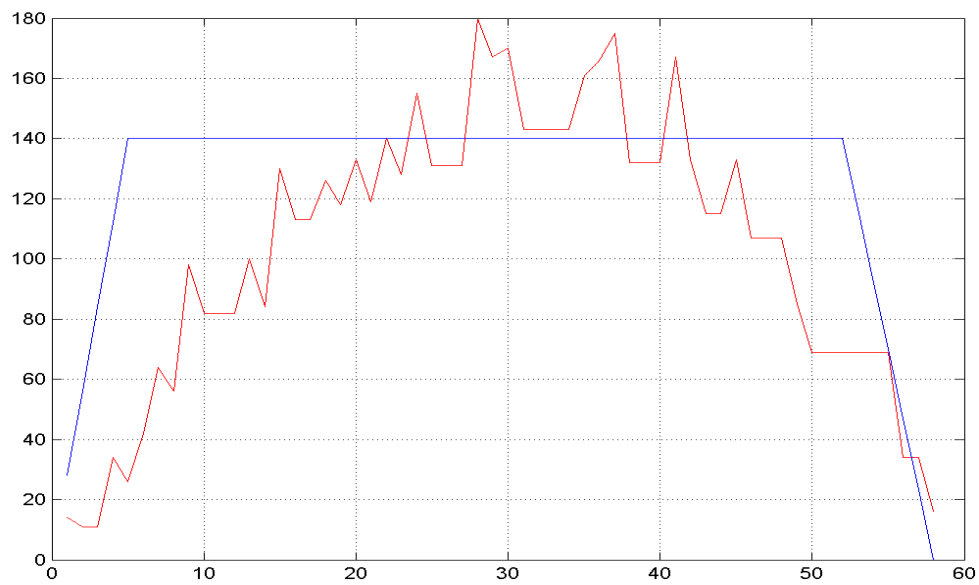
Μετά την εισαγωγή των δεδομένων του προβλήματός μας, θέτουμε σε εφαρμογή το πρόγραμμα για να δούμε εάν λειτουργεί σωστά, εάν δίνει καλές λύσεις και πόσο σύντομα, και πως επιδρούν σε αυτές οι διάφοροι παράμετροι (iterations, trials, percentage, MMAX Time) στους οποίους ο κάθε χρήστης μπορεί να δώσει διαφορετικές τιμές. Αρχικά, ξεκινούμε αυτή την αναζήτηση δίνοντας στο πρόγραμμα τιμές μεταβλητών (αριθμού εργατών και μέγιστης χρονικής διάρκειας έργου) κοντά στις τιμές που προέκυψαν από τη βελτιστοποίηση μέσω του χρονικού προγραμματισμού ($N = 146$ εργάτες ,max time= 59) για να δούμε ποιες εναλλακτικές λύσεις προσφέρονται, εάν βέβαια υπάρχουν, κοντά στη λύση της βέλτιστης χρονικής διάρκειας του έργου. Διατηρούμε τον αριθμό iterations σταθερό και ίσο με εκατό και τον αριθμό των trials ίσο με 40.

- **1^η Δοκιμή**

$N = 140$, max time= 64, percentage = 0.1, iterations= 100, trials= 40



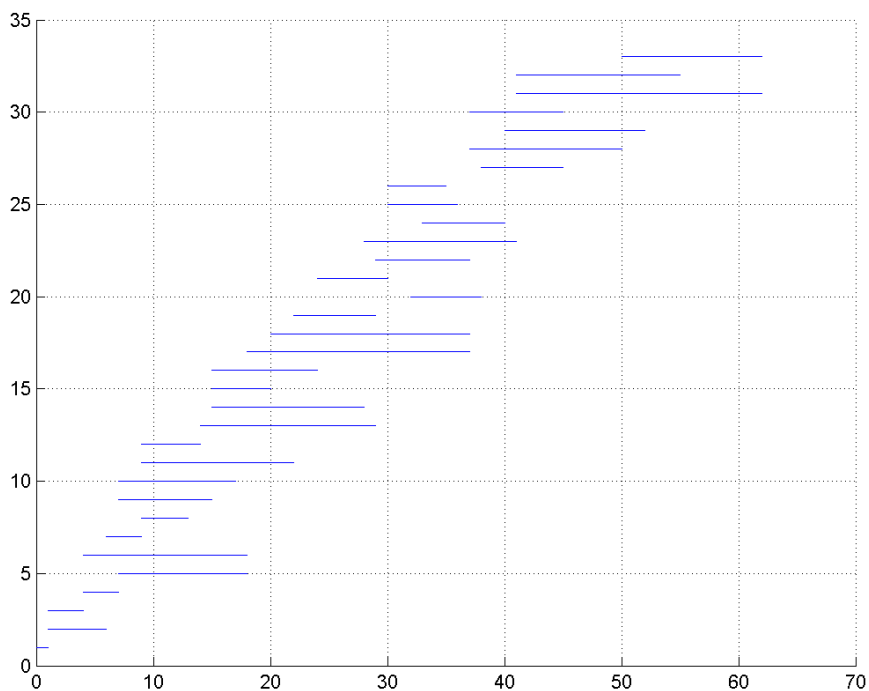
Σχήμα 5.1 Χρονοδιάγραμμα Εργασιών
($N = 140$, max time = 64, percentage = 0.1, iterations= 100, trials= 40)



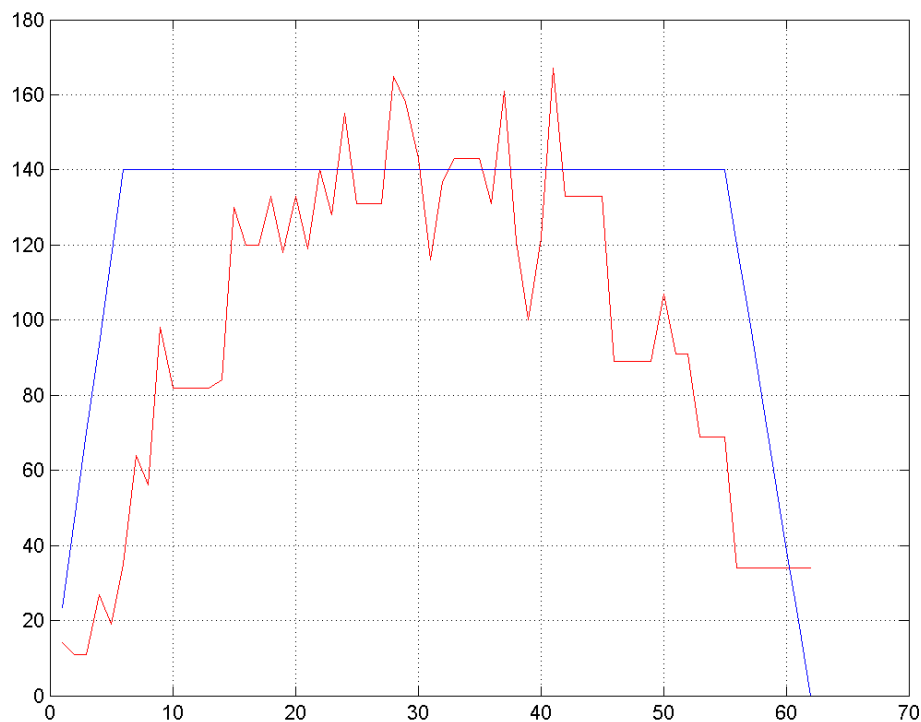
Σχήμα 5.2 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
($N = 140$, max time = 64, percentage = 0.1, iterations= 100, trials= 40)

- **2^η Δοκιμή**

$N = 140$, max time= 64, percentage = 0.1, iterations= 100, trials= 40



Σχήμα 5.3 Χρονοδιάγραμμα Εργασιών
($N = 140$, max time = 64, percentage = 0.1, iterations= 100, trials= 40)

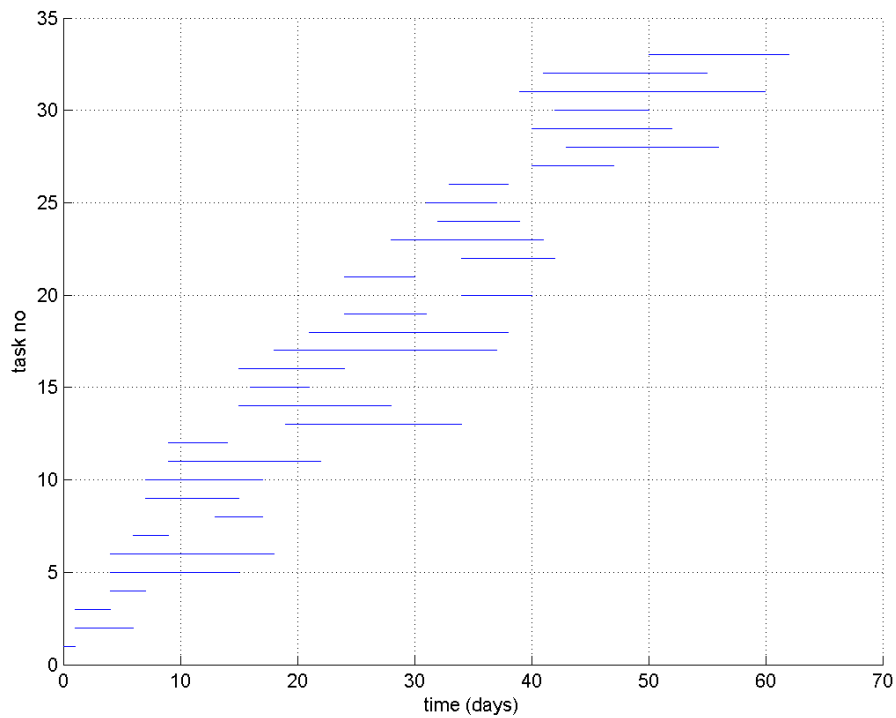


Σχήμα 5.4 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
($N = 140$, max time = 64, percentage = 0.1, iterations= 100, trials= 40)

Παρατηρούμε ότι το percentage εκτοξεύει τη τιμή της objective function και κρατάει συμπιεσμένο το έργο διατηρώντας μεγάλες τιμές στον αριθμό των εργατών άρα θα πρέπει να το αυξήσουμε για να επιτύχουμε πιο ομαλή κατανομή. Πριν όμως συνεχίσουμε, ας δούμε κάποιες δοκιμές που έγιναν με σκοπό τη αποσαφήνιση της σχέσης της τιμής του percentage με την τιμή της αντικειμενικής συνάρτησης objective function και τι επίδραση έχει η πρώτη στο κατά πόσο το έργο μπορεί να αποσυμπιεστεί. Διατηρούμε το ανώτατο όριο των εργατών στους 140, τον αριθμό των trials στο 40 αυξάνουμε το iterations στις 200 επαναλήψεις το percentage στο 0,1 και το max time στις 150 ημέρες δίνοντας τη δυνατότητα στο πρόγραμμα να παρατείνει άνετα το έργο. Παραθέτουμε τις δέκα πρώτες δοκιμές που έγιναν στο πρόγραμμα με αυτά τα δεδομένα.

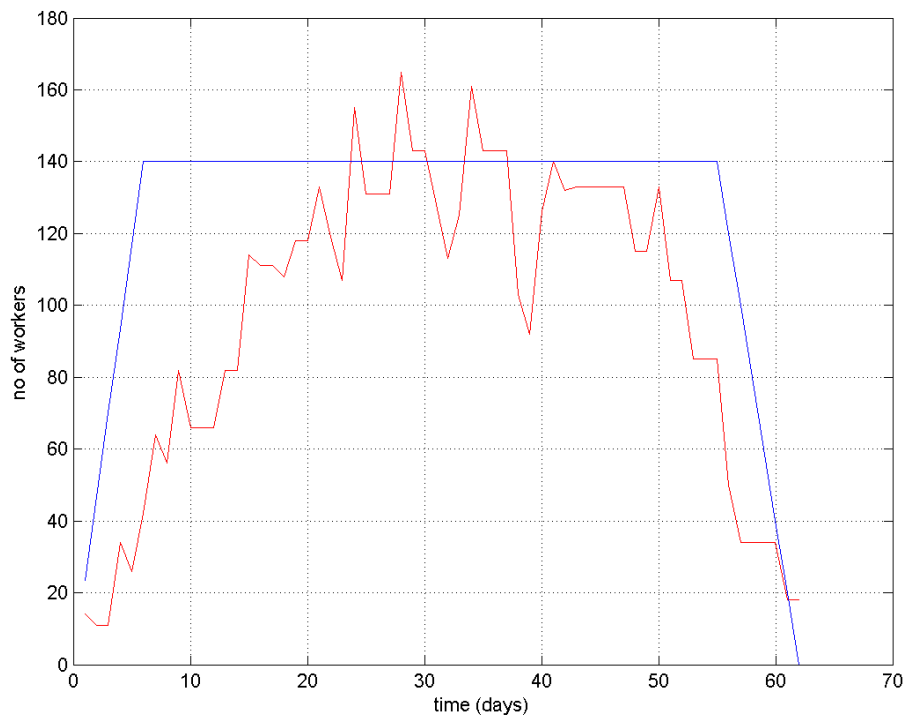
- **1^η Δοκιμή**

N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



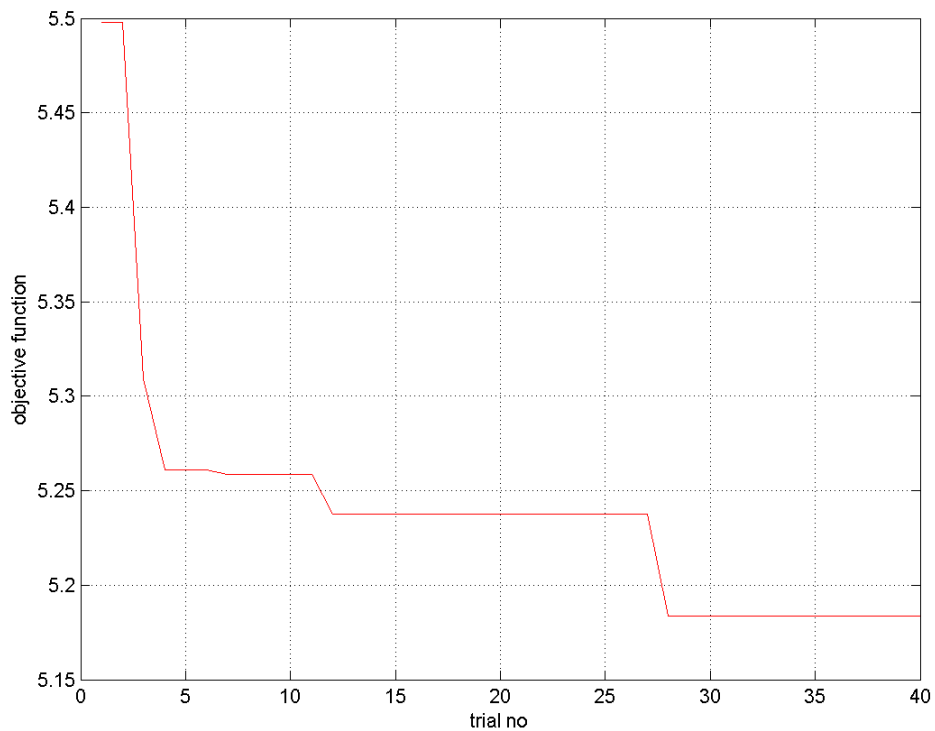
Σχήμα 5.5 Χρονοδιάγραμμα Εργασιών

(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.6 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

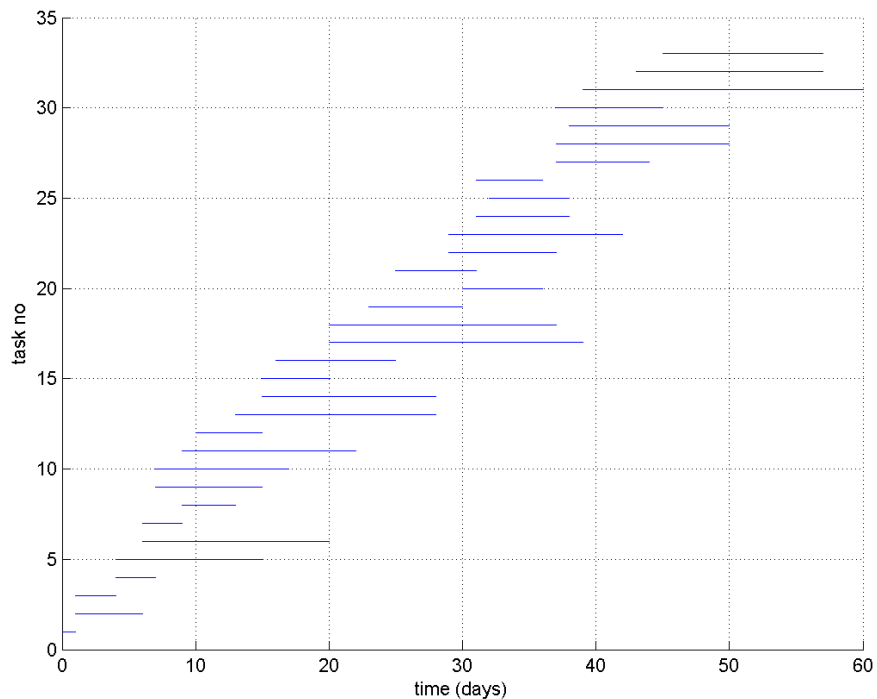
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



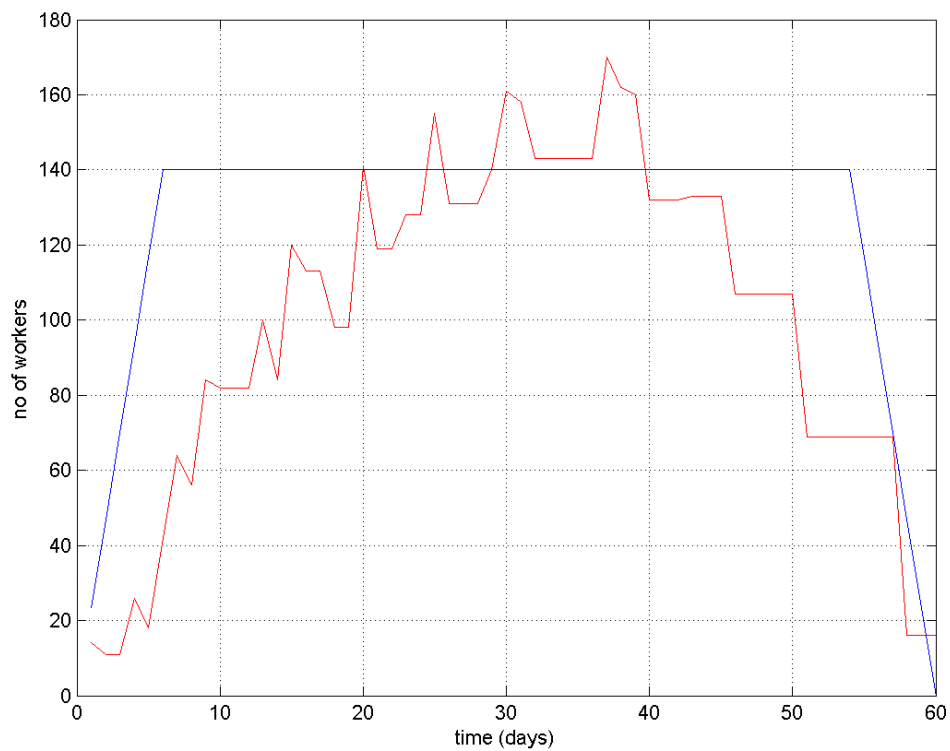
Σχήμα 5.7 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)

- **2^η Δοκιμή**

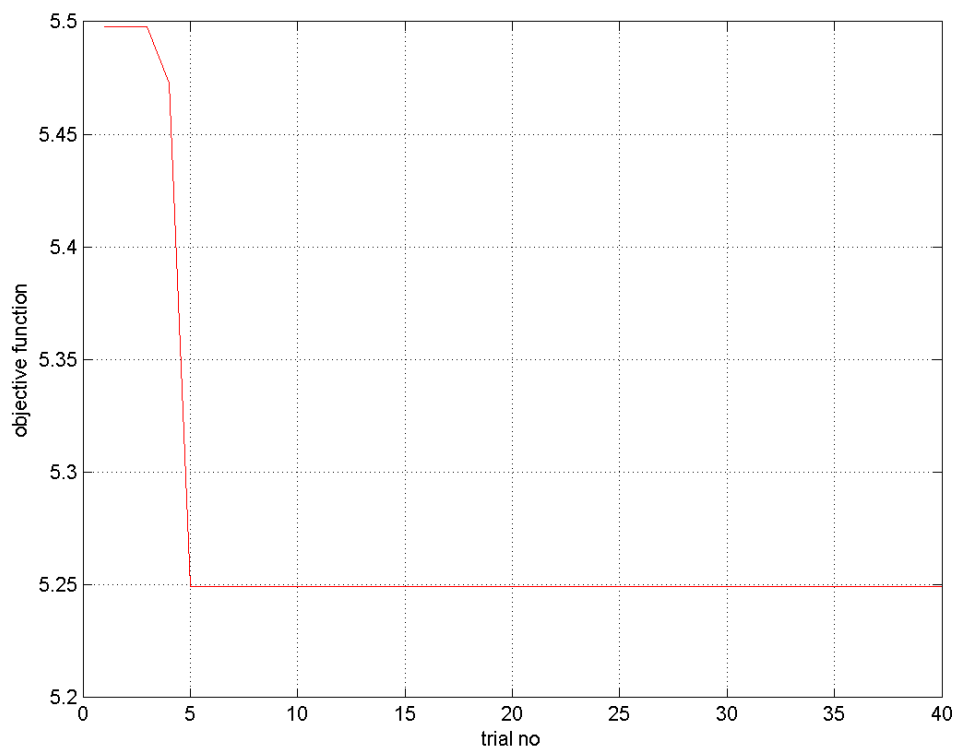
N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



Σχήμα 5.8 Χρονοδιάγραμμα Εργασιών
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



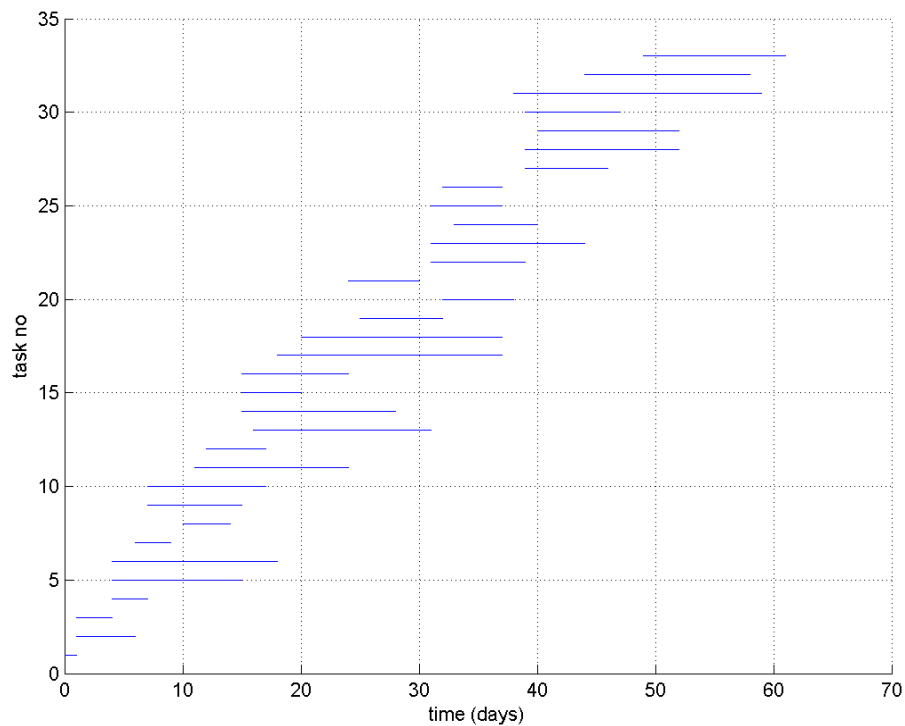
Σχήμα 5.9 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.10 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)

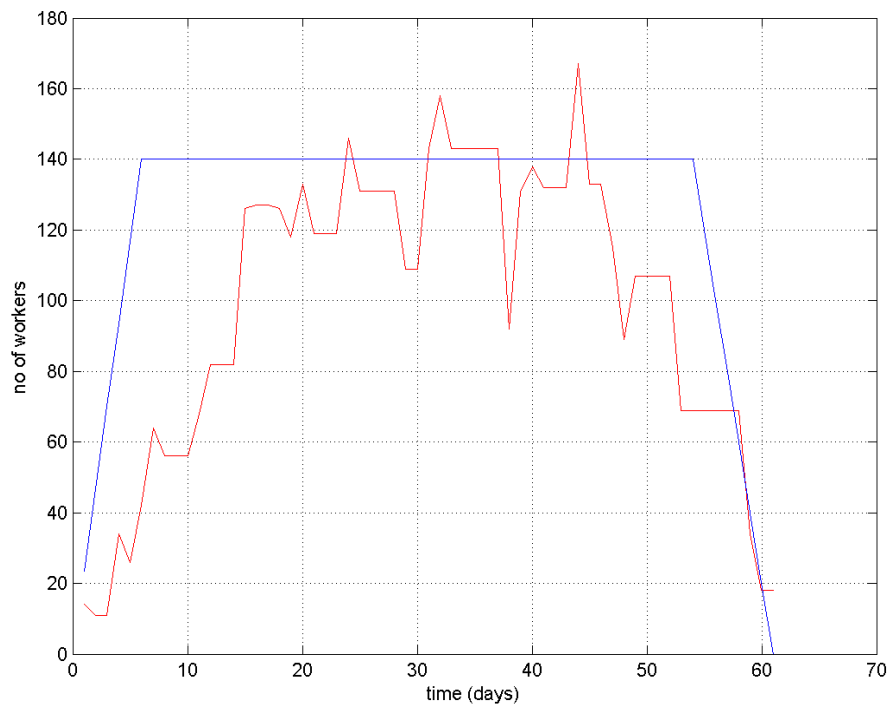
- **3^η Δοκιμή**

N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



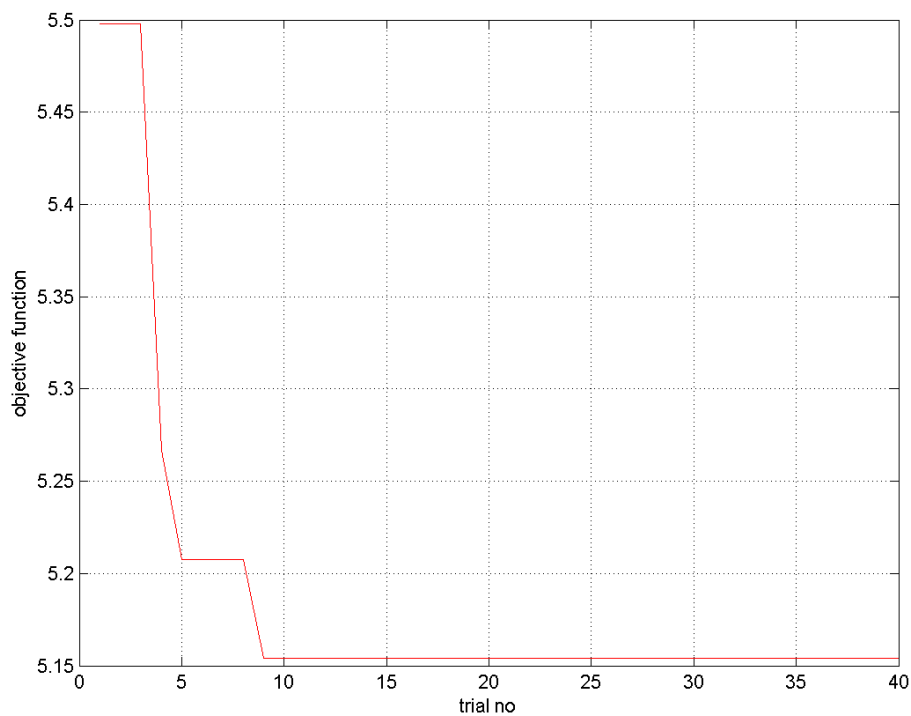
Σχήμα 5.11 Χρονοδιάγραμμα Εργασιών

(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.12 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

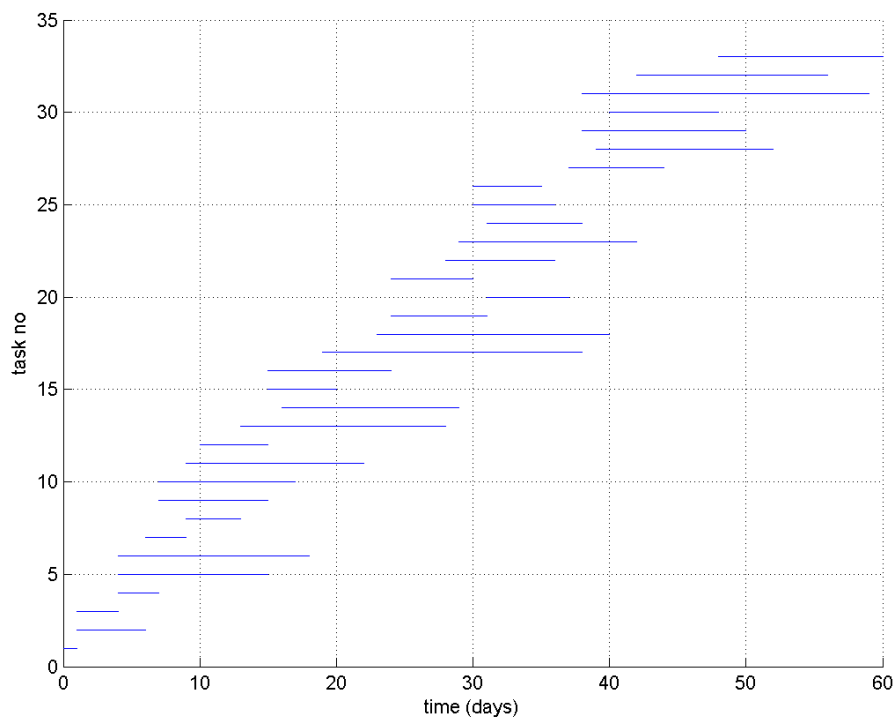
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



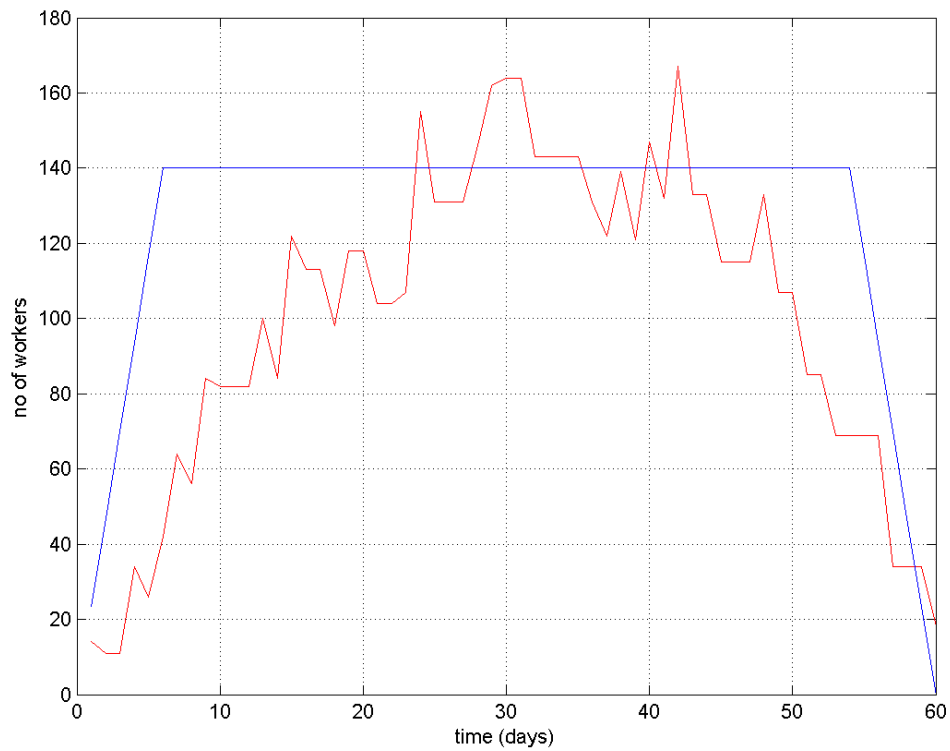
Σχήμα 5.13 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)

- **4^η Δοκιμή**

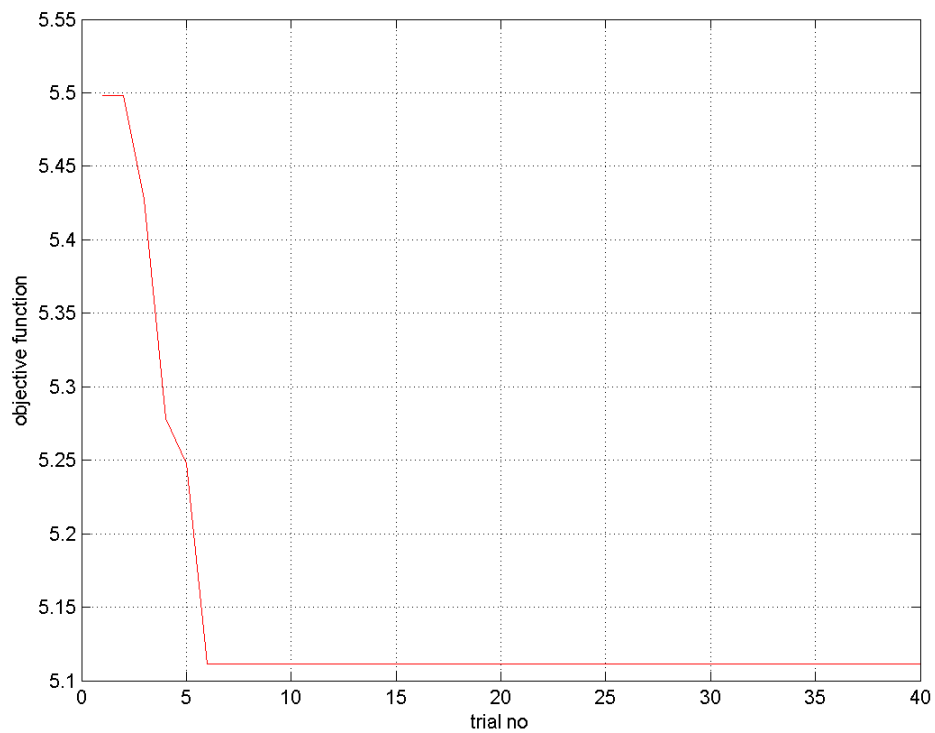
N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



Σχήμα 5.14 Χρονοδιάγραμμα Εργασιών
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



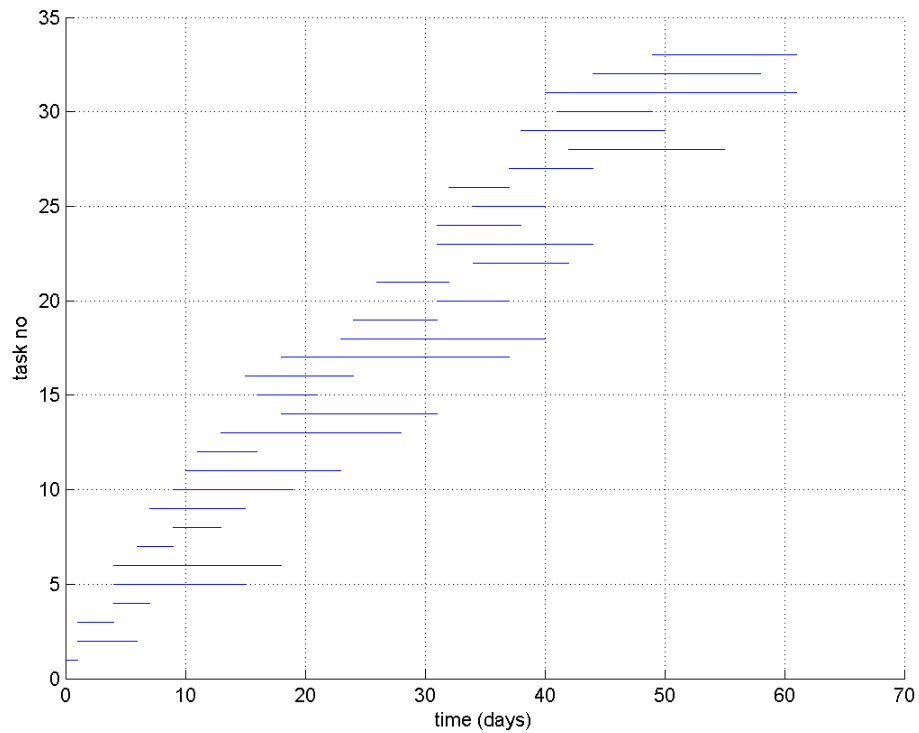
Σχήμα 5.15 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.16 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)

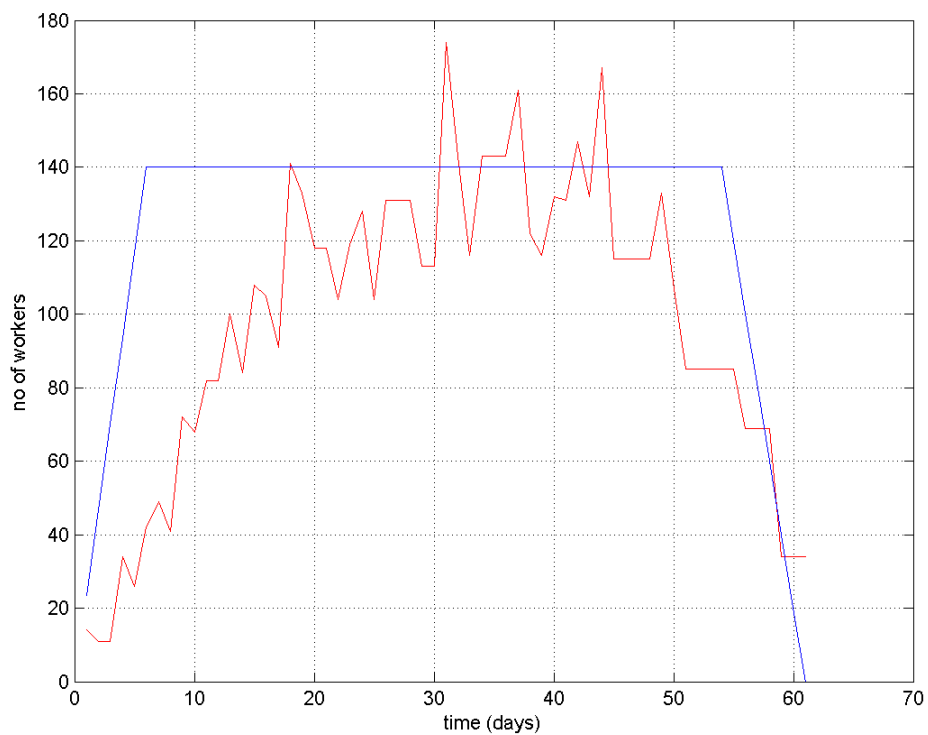
- **5^η Δοκιμή**

N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



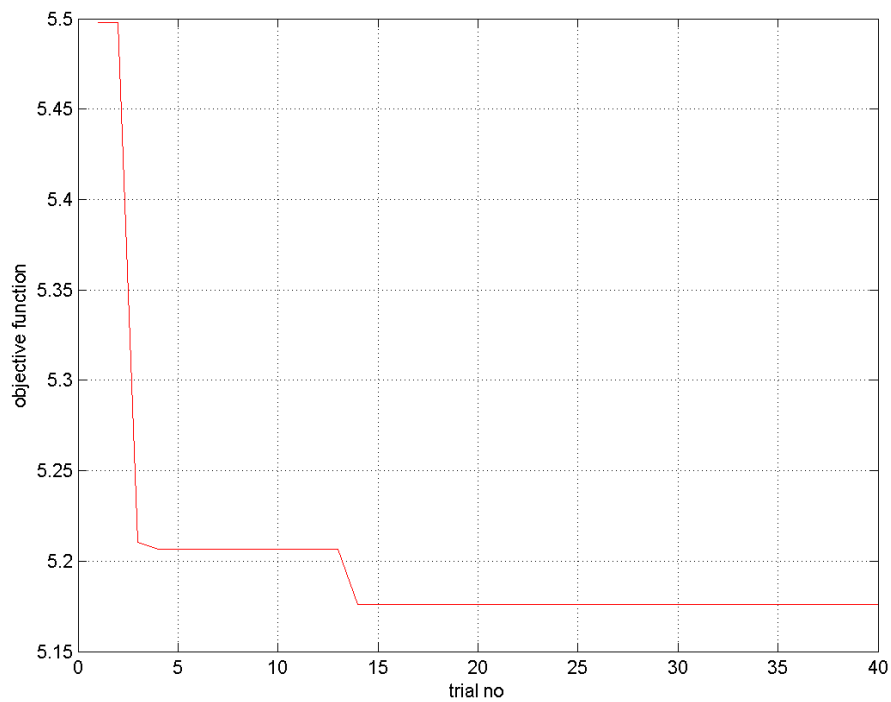
Σχήμα 5.17 Χρονοδιάγραμμα Εργασιών

(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.18 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

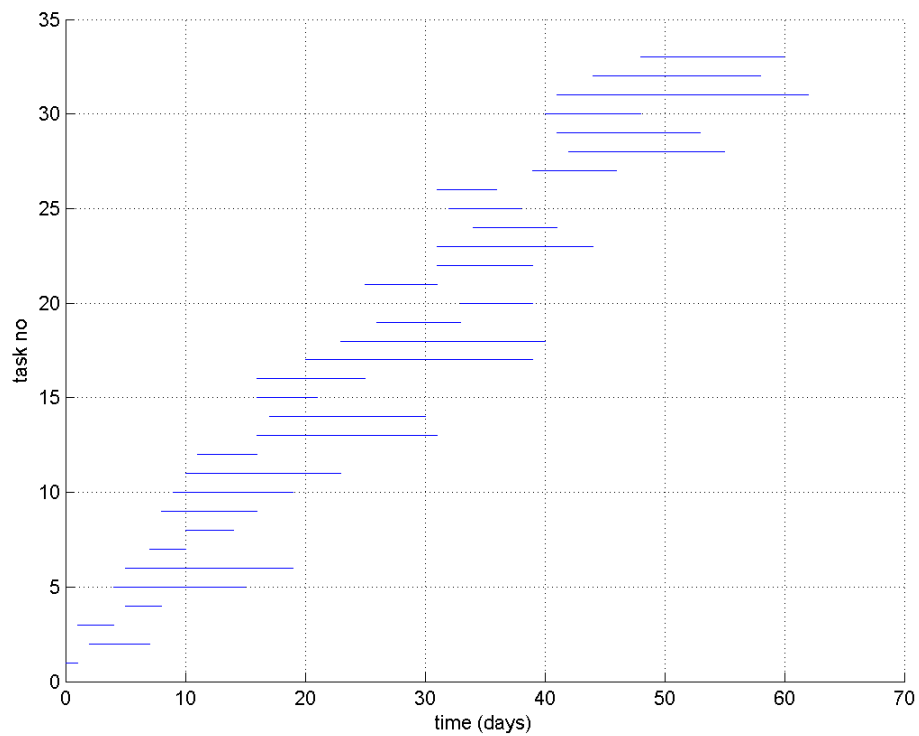
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



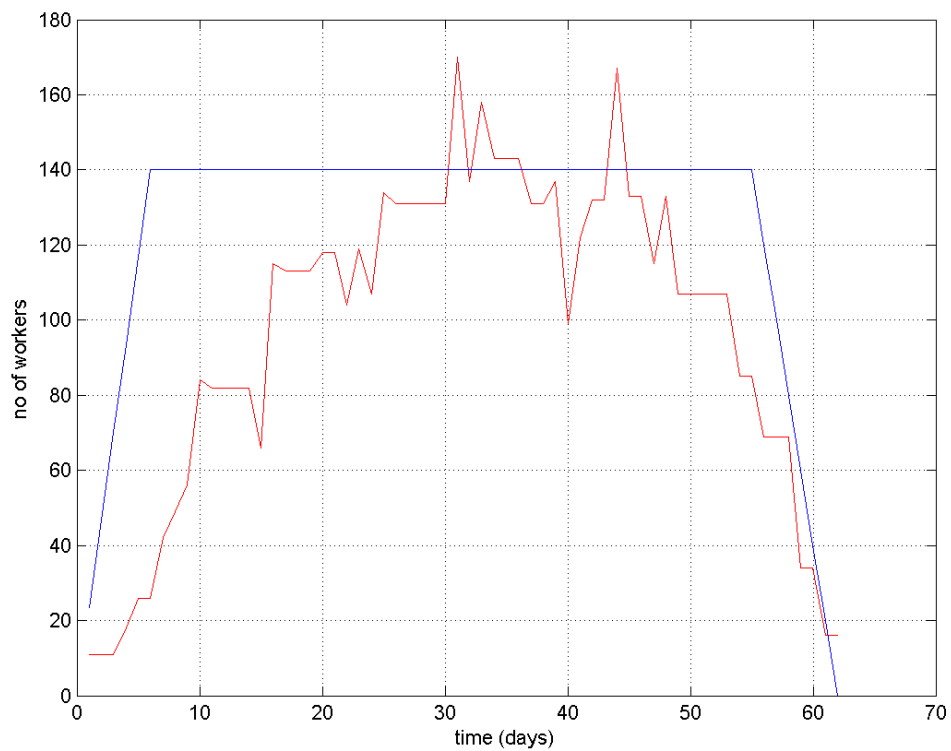
Σχήμα 5.19 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

- **6^η Δοκιμή**

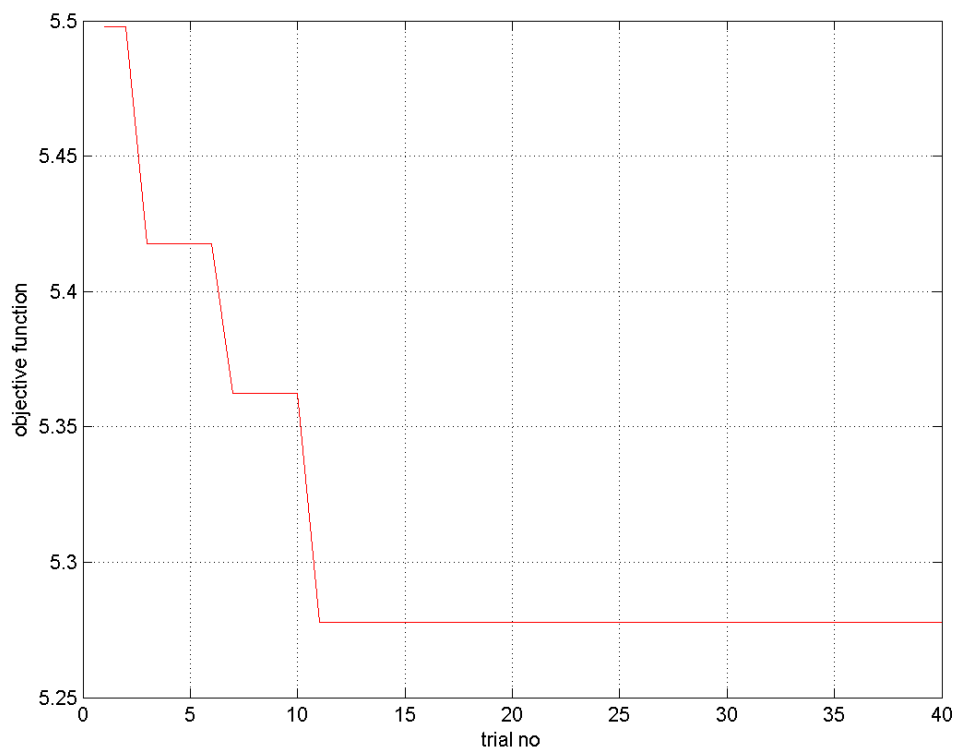
N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



Σχήμα 5.20 Χρονοδιάγραμμα Εργασιών
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



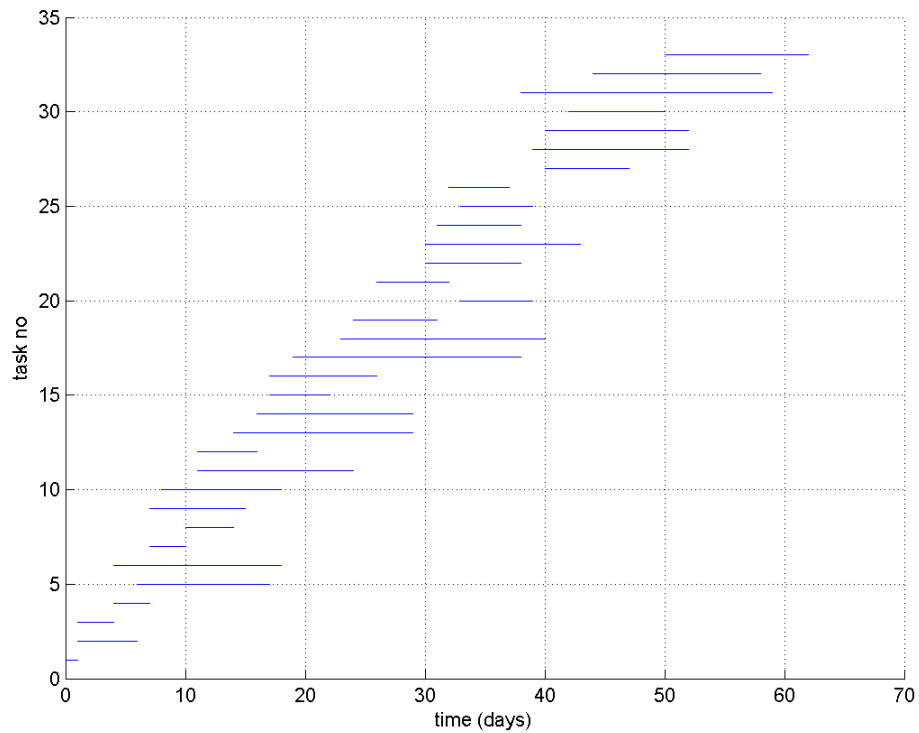
Σχήμα 5.21 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
($N = 140$, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.22 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
($N = 140$, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

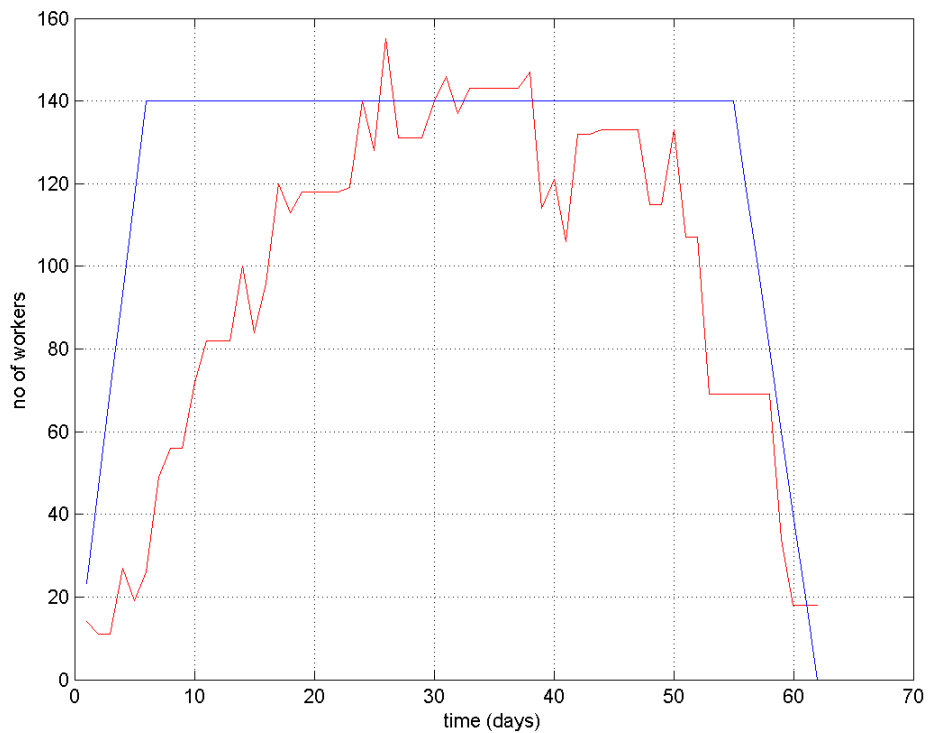
- **7^η Δοκιμή**

N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



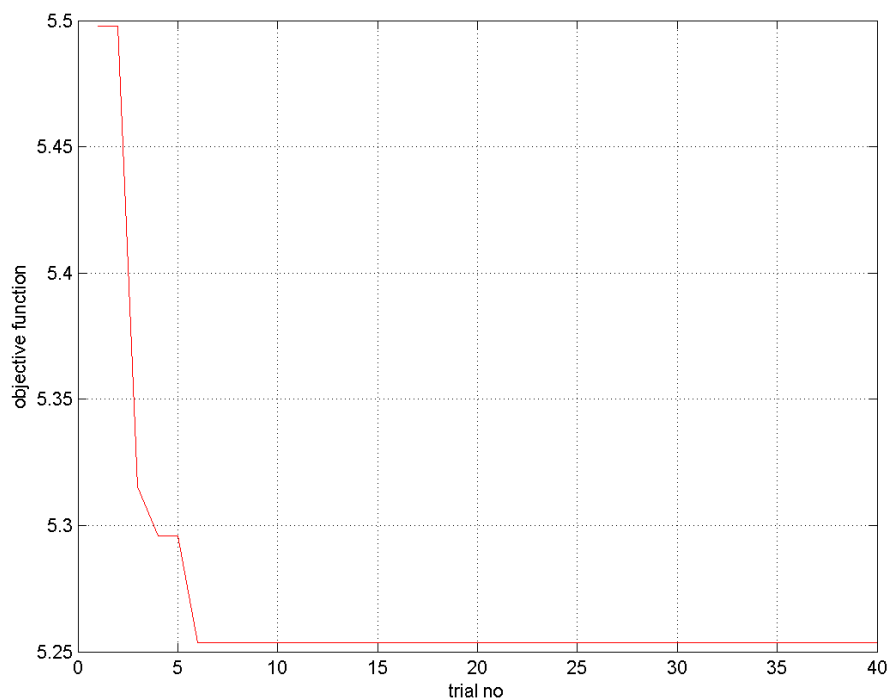
Σχήμα 5.23 Χρονοδιάγραμμα Εργασιών

(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.24 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

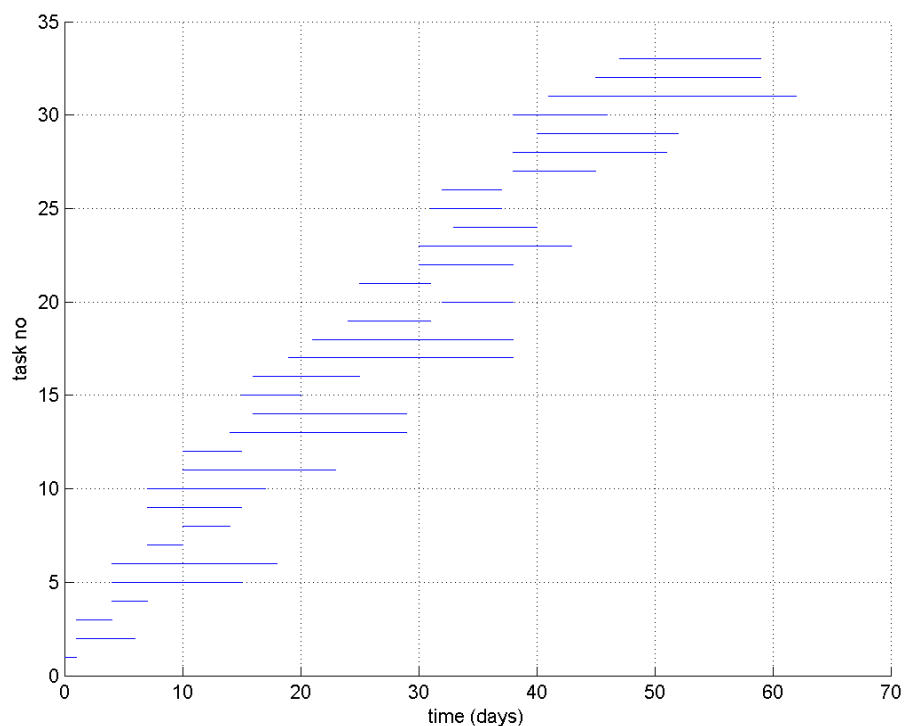
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



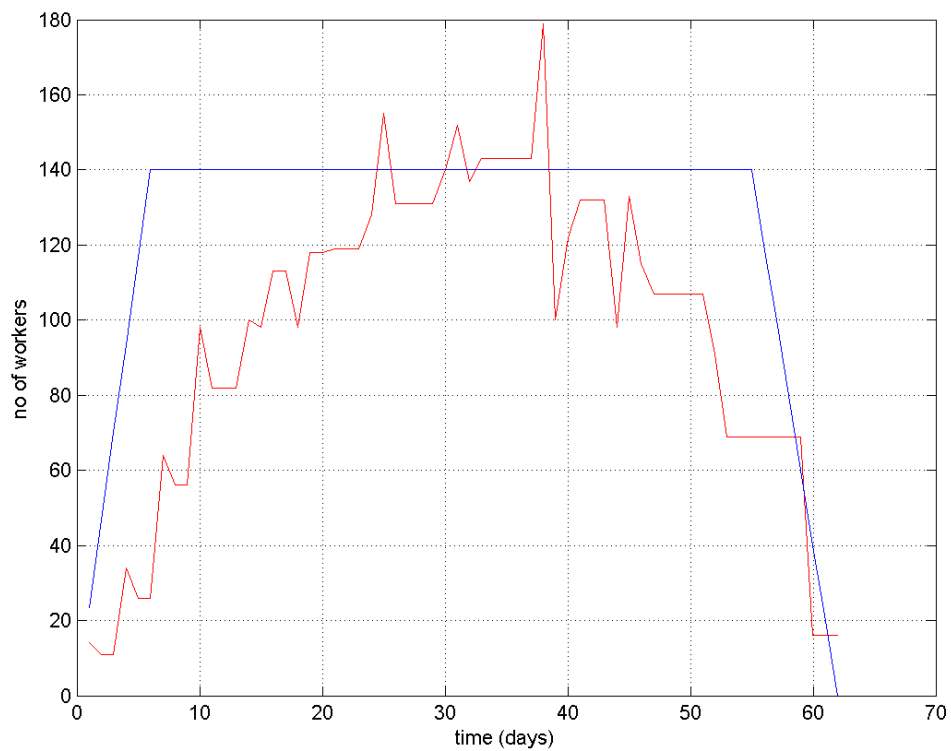
Σχήμα 5.25 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

- **8^η Δοκιμή**

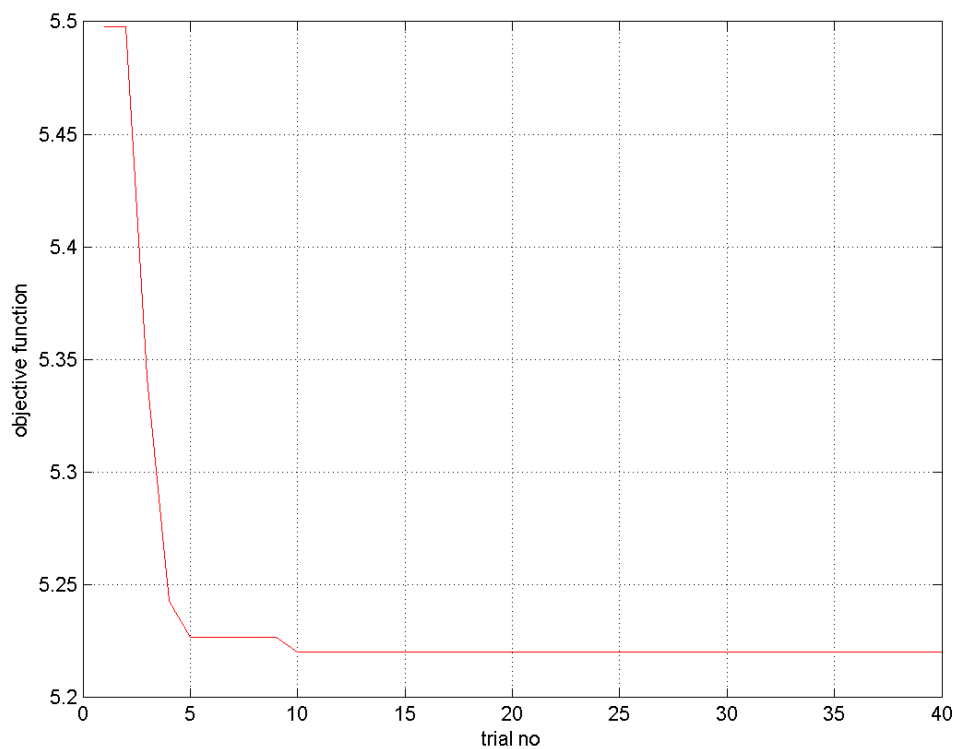
N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



Σχήμα 5.26 Χρονοδιάγραμμα Εργασιών
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



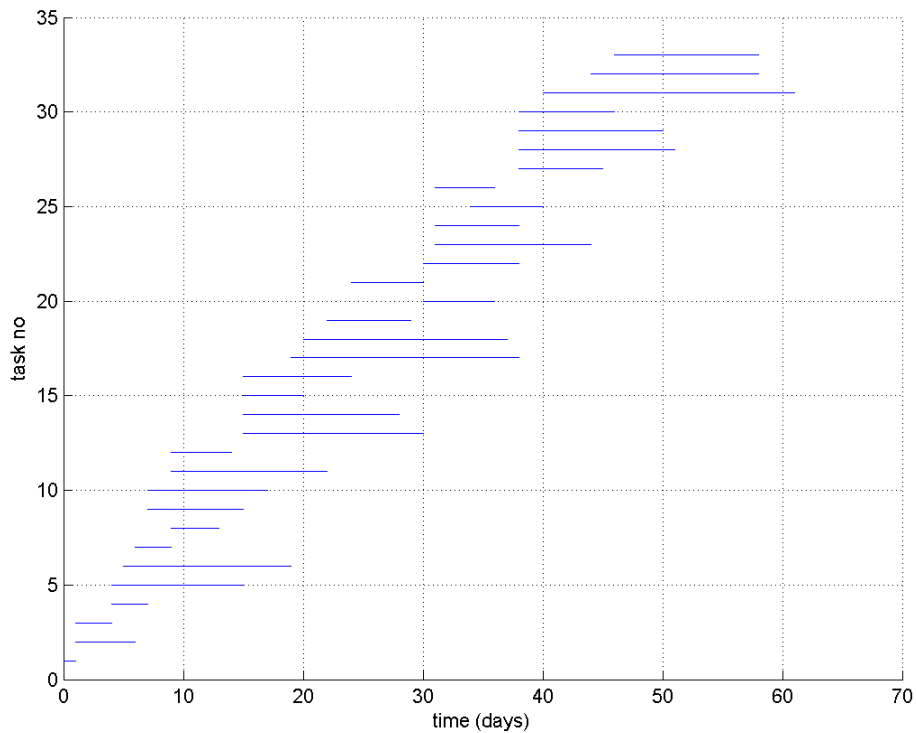
Σχήμα 5.27 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.28 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

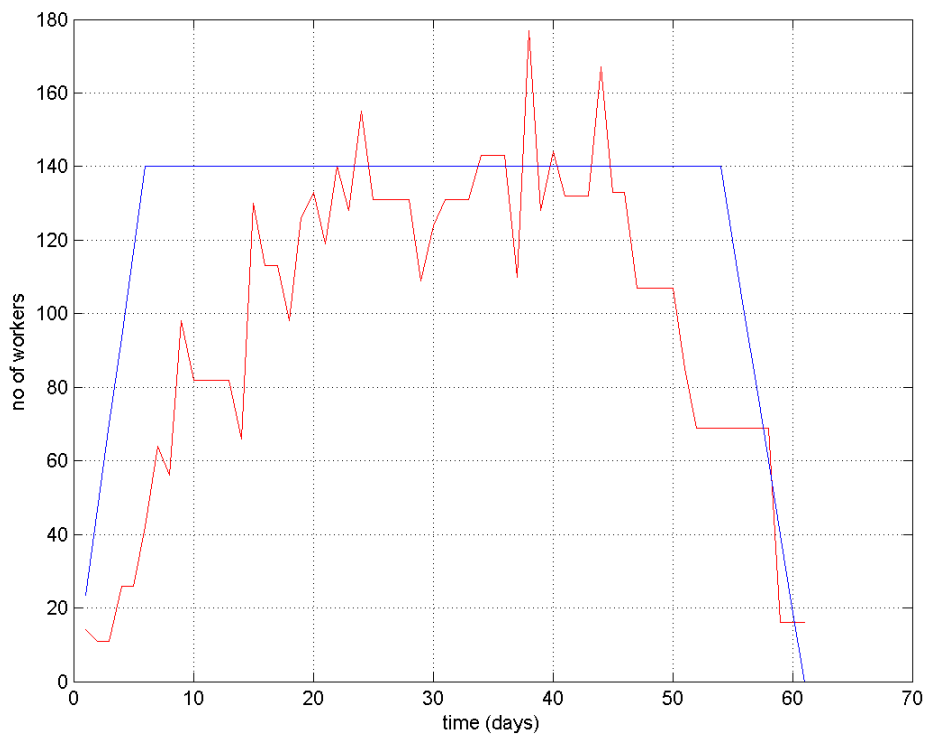
- **9^η Δοκιμή**

N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



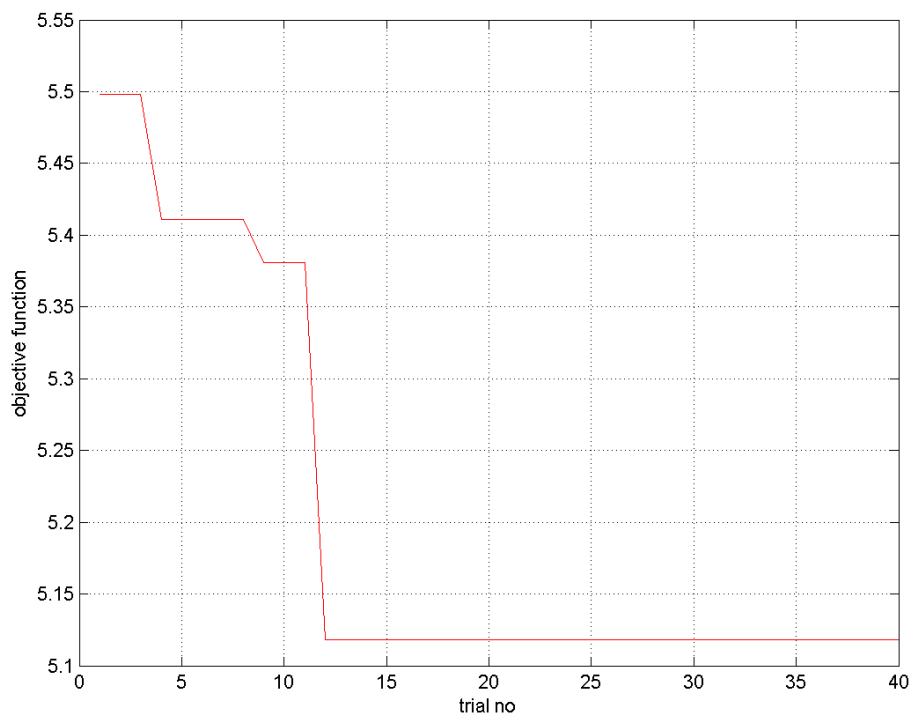
Σχήμα 5.29 Χρονοδιάγραμμα Εργασιών

(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.30 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

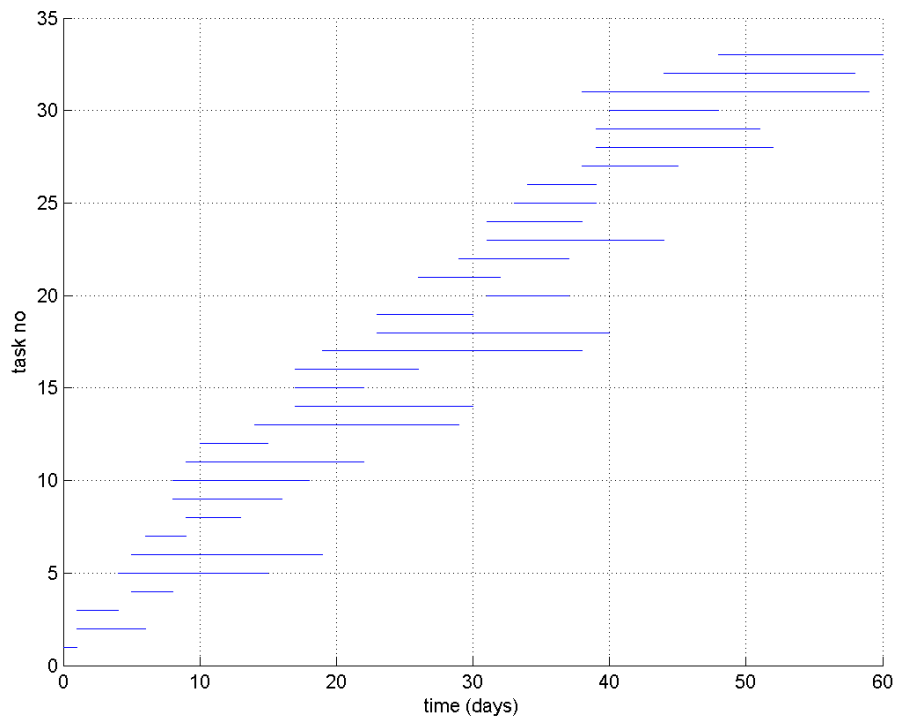
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



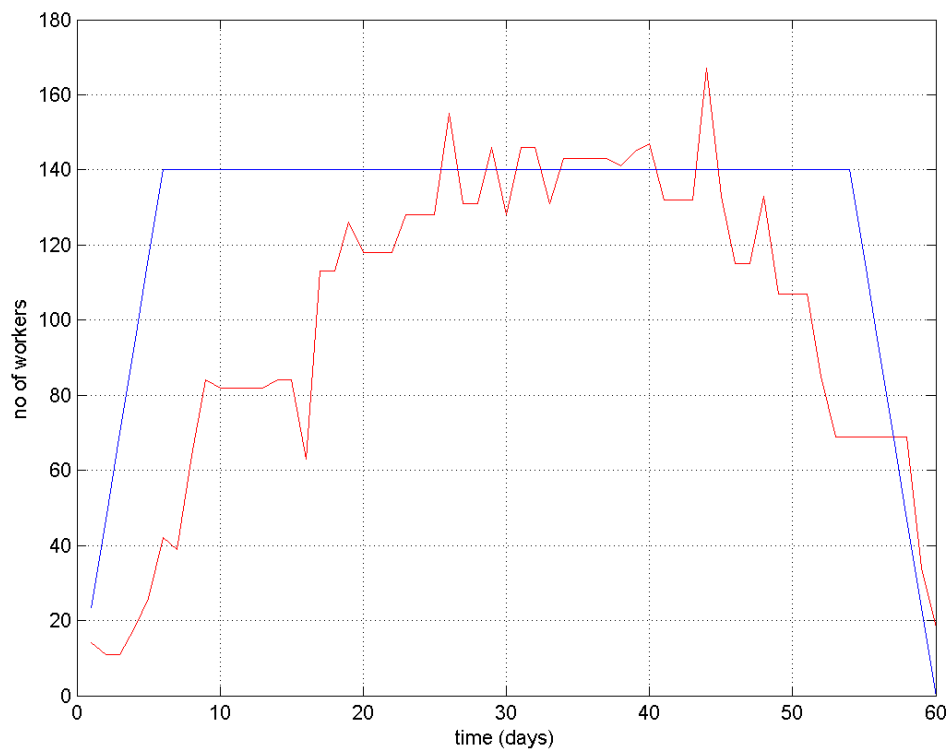
Σχήμα 5.31 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

- **10^η Δοκιμή**

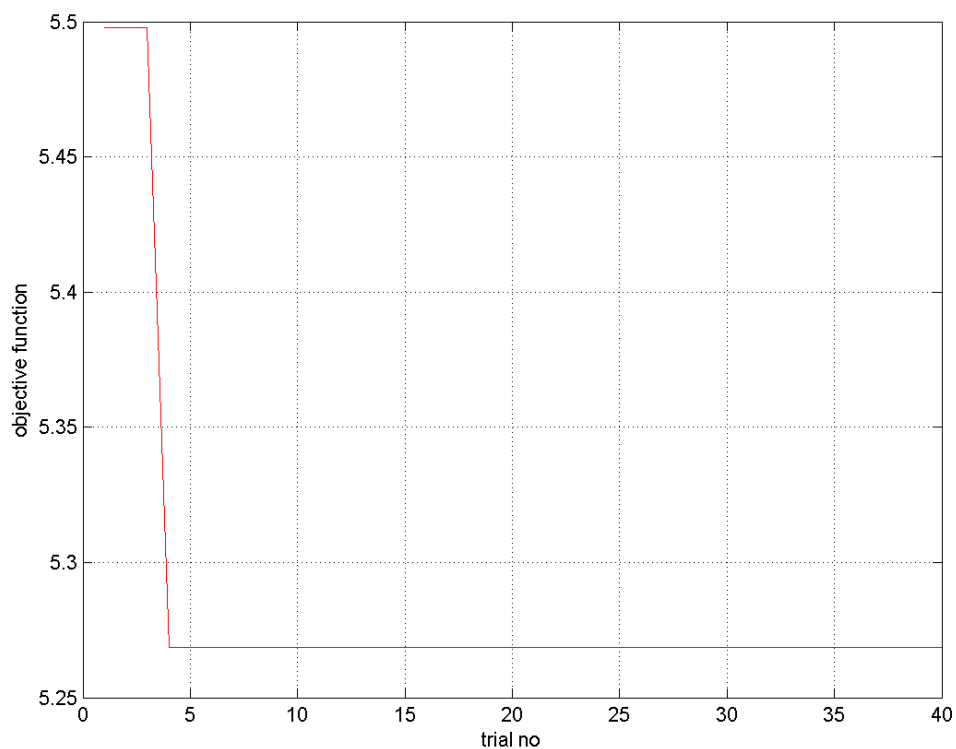
N = 140, max time= 150, percentage = 0.1, iterations= 200, trials= 40



Σχήμα 5.32 Χρονοδιάγραμμα Εργασιών
(N = 140, max time = 150, percentage = 0.1, iterations= 200, trials= 40)



Σχήμα 5.33 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
($N = 140$, max time = 150, percentage = 0.1, iterations= 200, trials= 40)

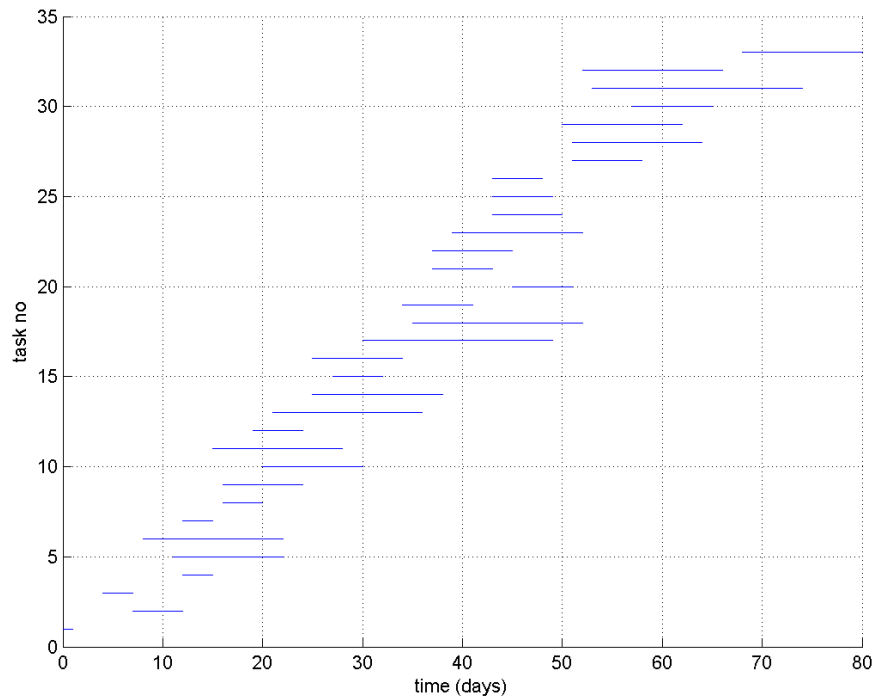


Σχήμα 5.34 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
($N = 140$, max time = 150, percentage = 0.1, iterations= 200, trials= 40)

Διατηρούμε όλες τις μεταβλητές σταθερές αυξάνουμε το percentage σε 0,4.

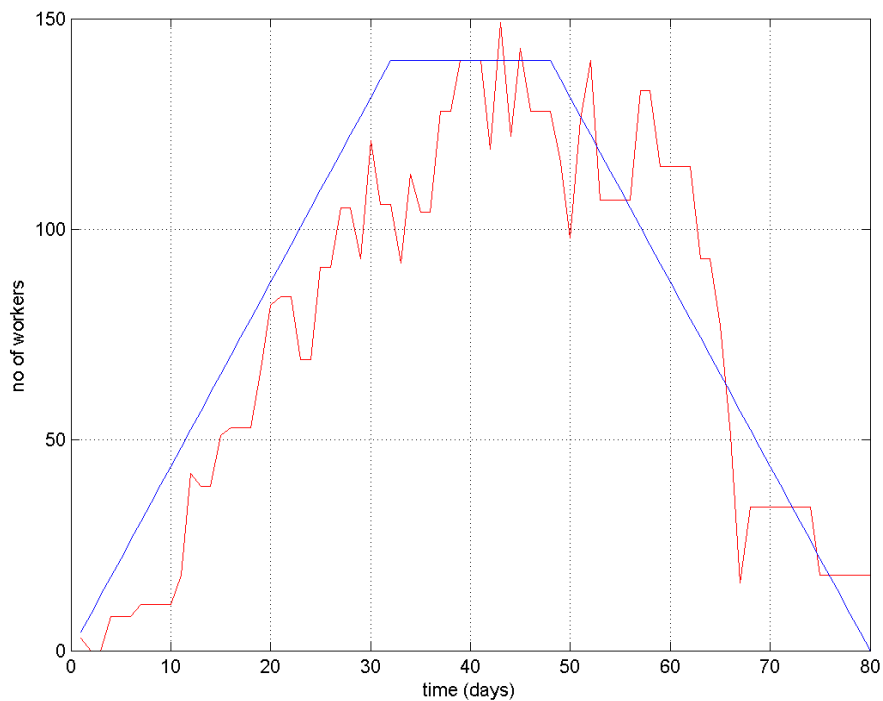
- **1^η Δοκιμή**

N = 140, max time= 150, percentage = 0.4, iterations= 200, trials= 40

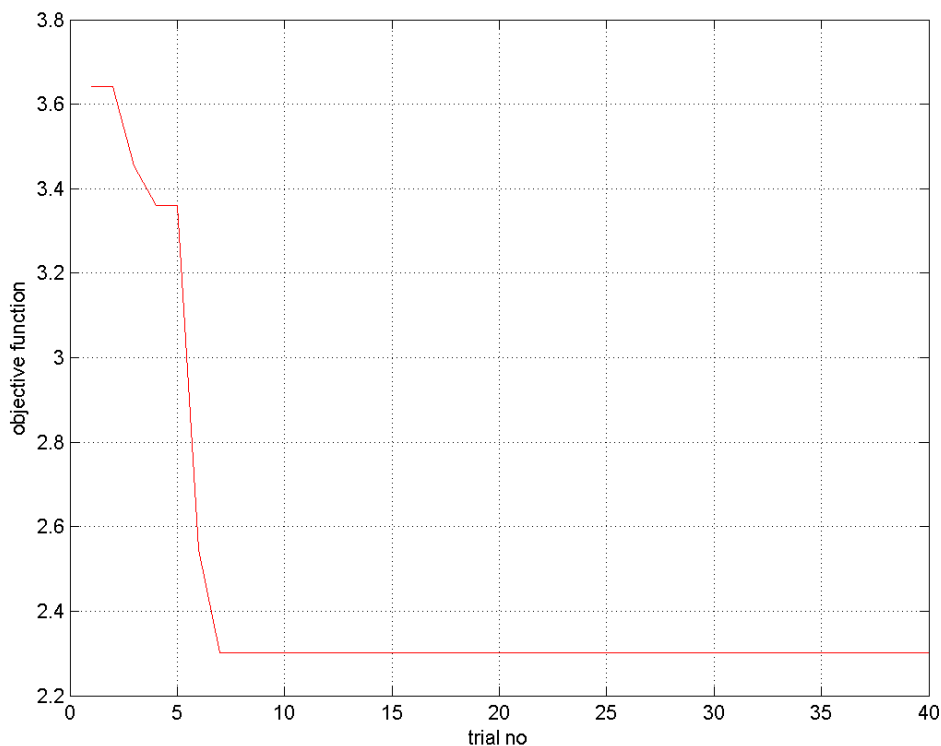


Σχήμα 5.35 Χρονοδιάγραμμα Εργασιών

(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)



Σχήμα 5.36 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

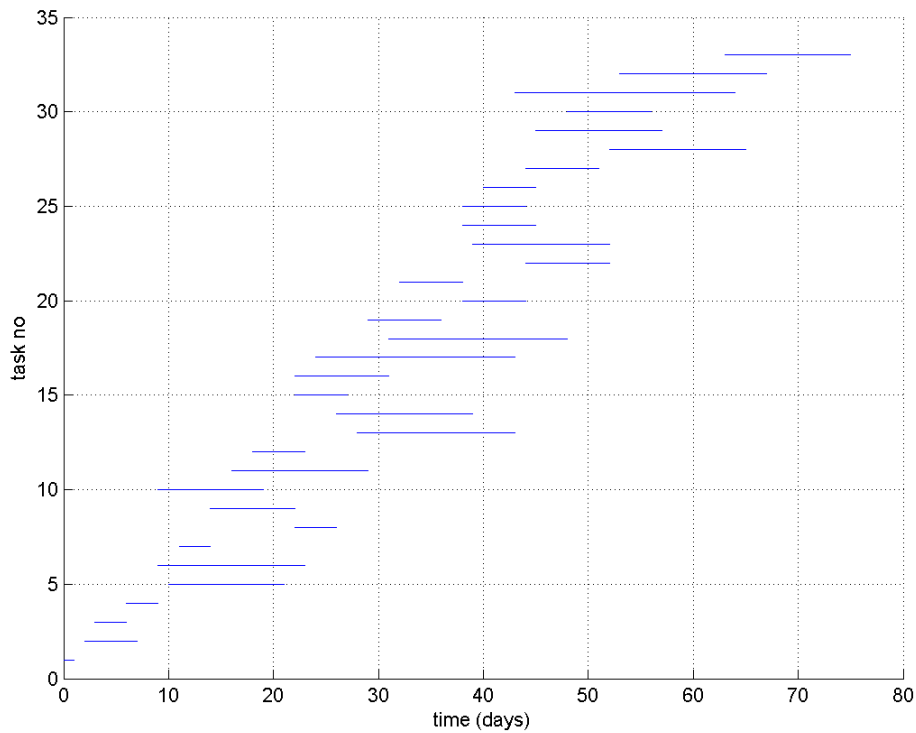


Σχήμα 5.37 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

Το έργο αποσυμπίεζεται αμέσως με τη πρώτη εφαρμογή κι αυτό γιατί με μεγαλύτερο percentage παίρνουμε καλύτερες τιμές στην objective Function και παράλληλα μας δίνει και μια καλή λύση. Στις προηγούμενες εφαρμογές με per 0,1 λόγω της απότομης κλίσης είχαμε μεγάλη διαφορά στη γραφική παράσταση του τραπεζίου από τη γραφική παράσταση των εργατών, με αποτέλεσμα η τιμή της objective Function να παίρνει μεγάλες τιμές και ο κώδικας να μην προσπαθεί να επεκτείνει άλλο το έργο καθώς η αύξηση της τιμής θα συνεχιστεί. Ας τρέξουμε όμως μερικές φορές ακόμα το πρόγραμμα για να δούμε τη ποιότητα των λύσεων που προσφέρονται τώρα από τον κώδικα.

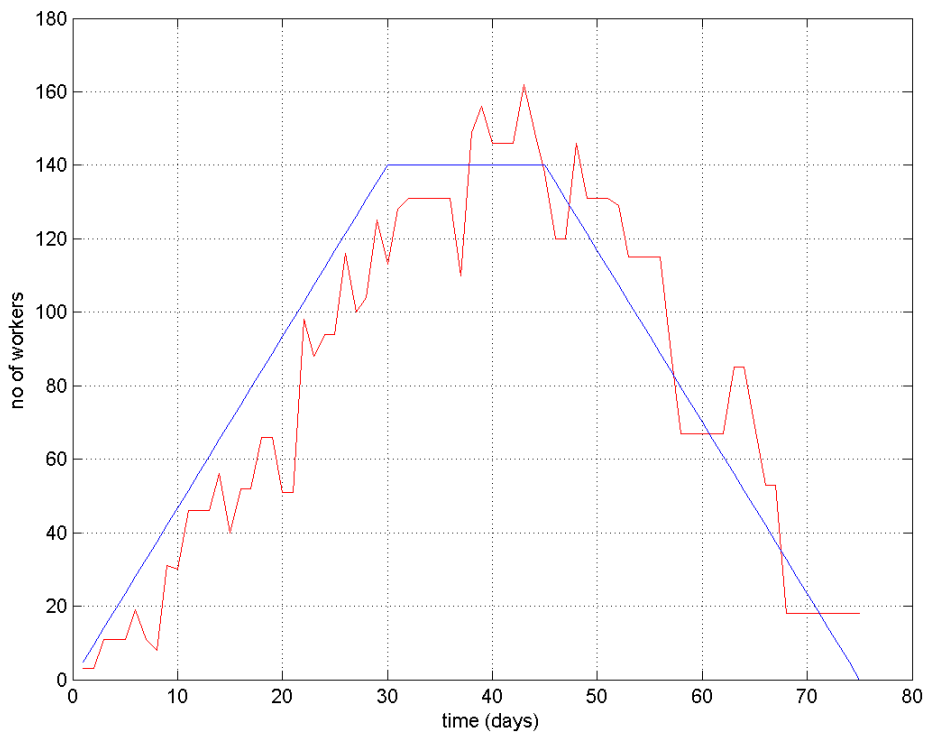
- **2^η Δοκιμή**

N = 140, max time= 150, percentage = 0.4, iterations= 200, trials= 40



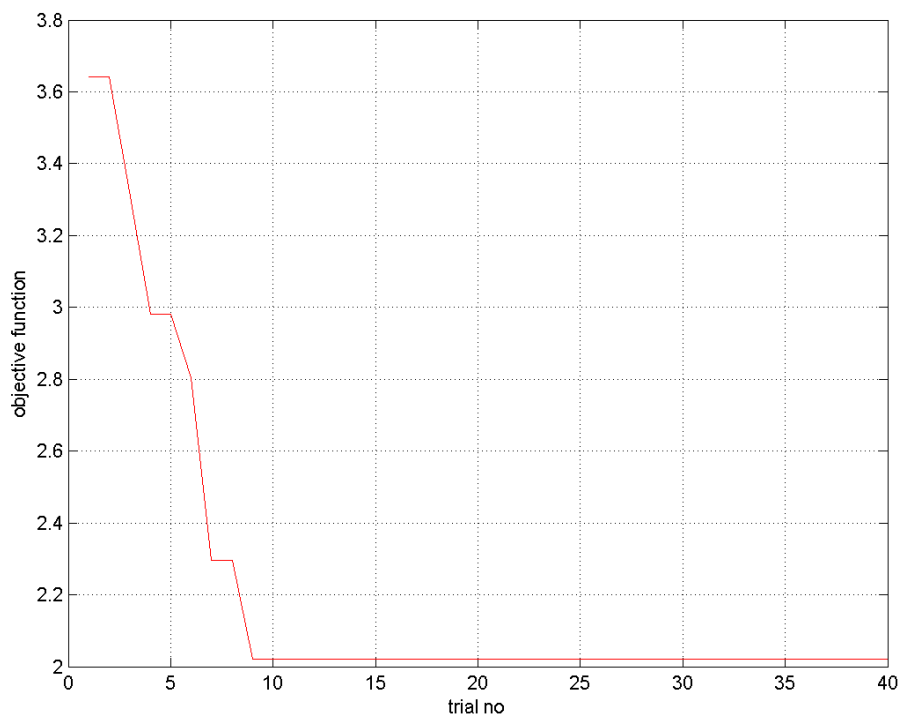
Σχήμα 5.38 Χρονοδιάγραμμα Εργασιών

(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)



Σχήμα 5.39 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

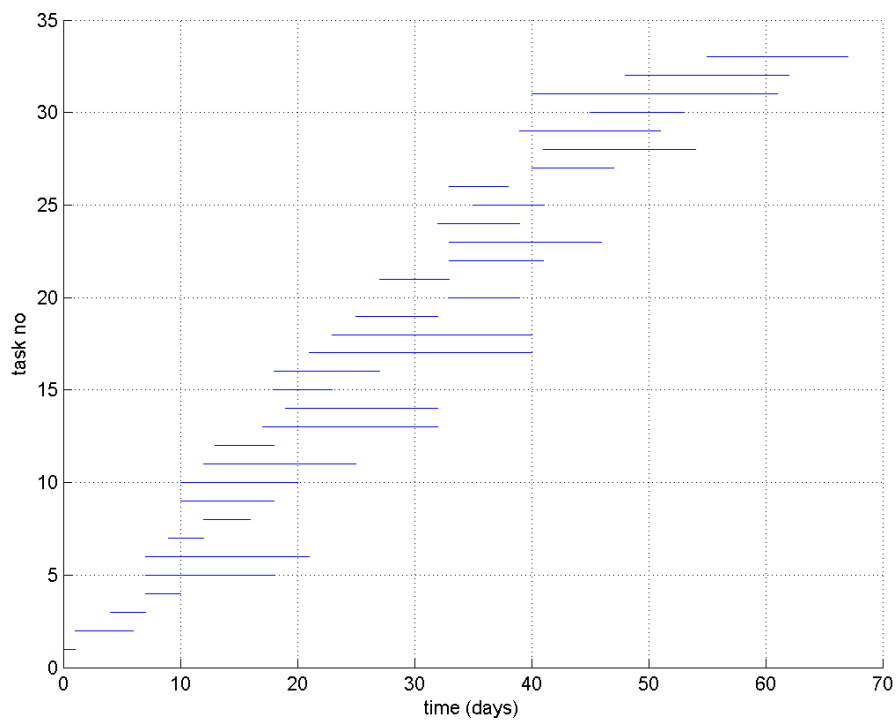
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)



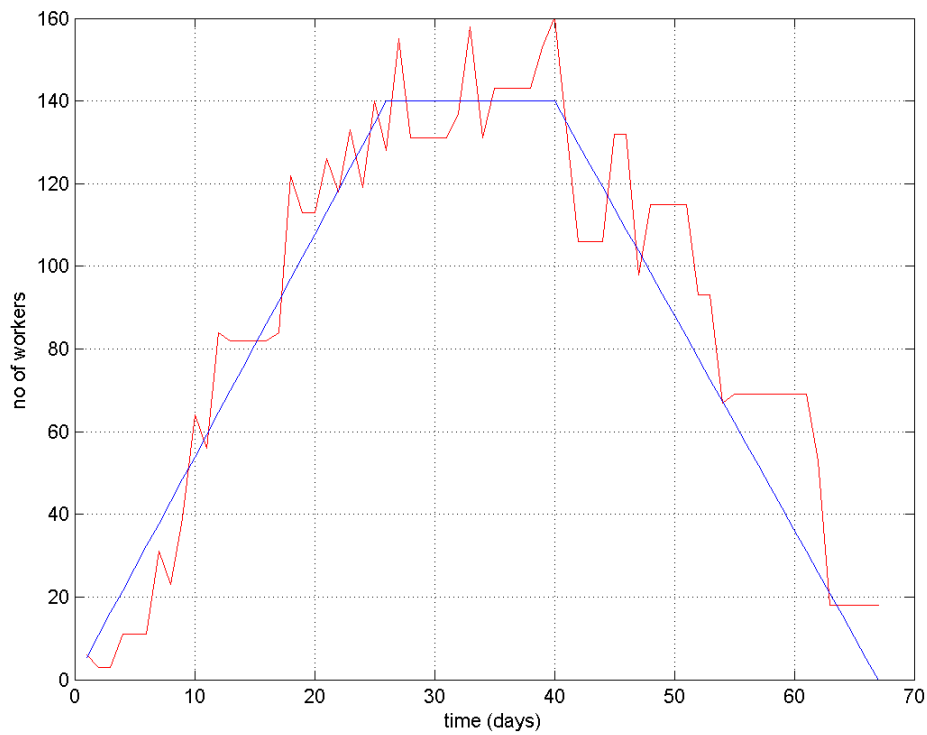
Σχήμα 5.40 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

- **3^η Δοκιμή**

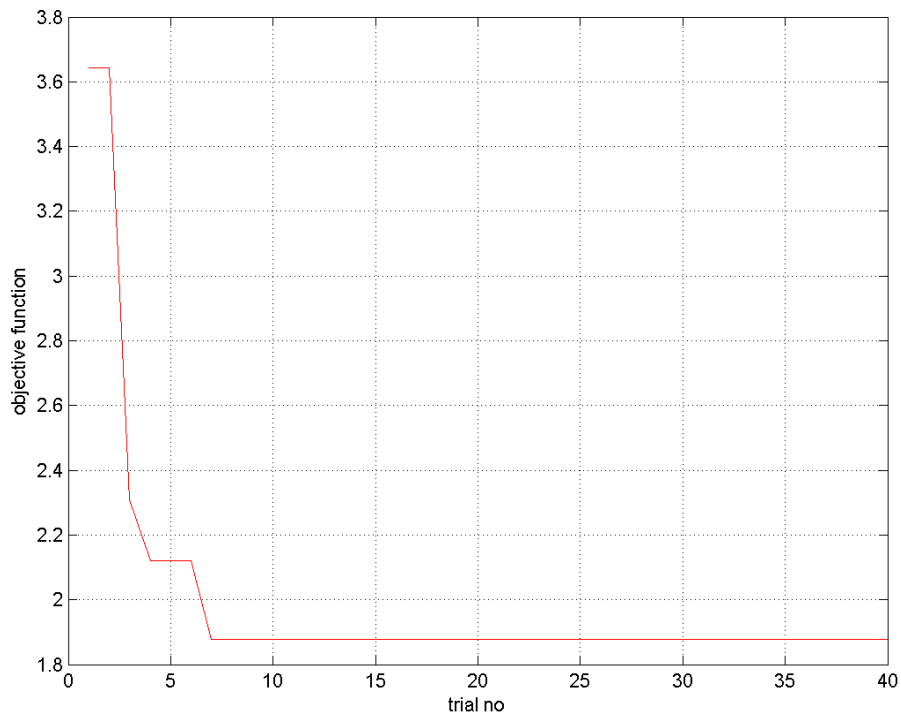
N = 140, max time= 150, percentage = 0.4, iterations= 200, trials= 40



Σχήμα 5.41 Χρονοδιάγραμμα Εργασιών
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)



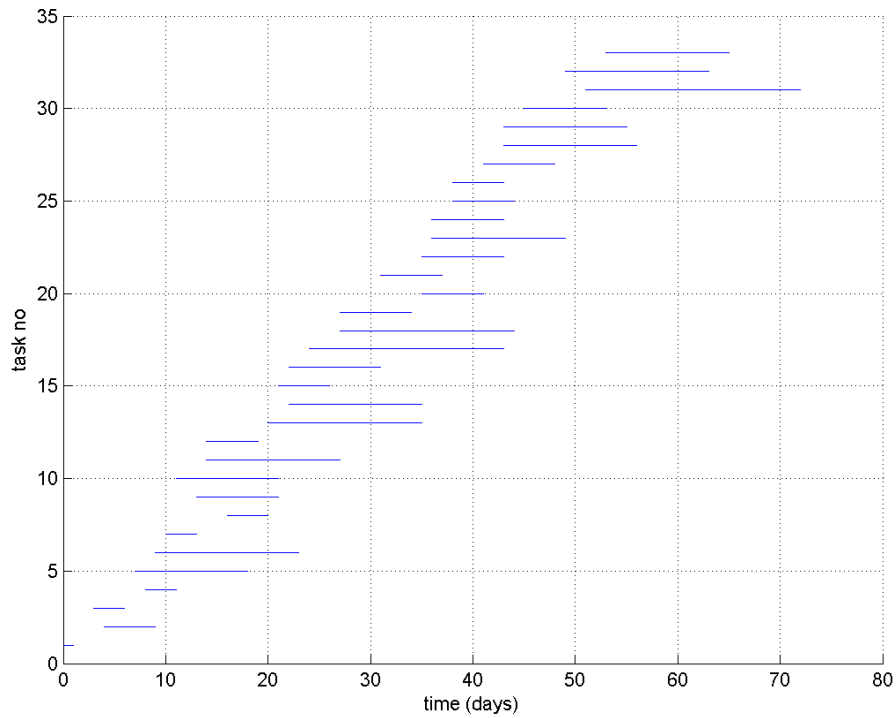
Σχήμα 5.42 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)



Σχήμα 5.43 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

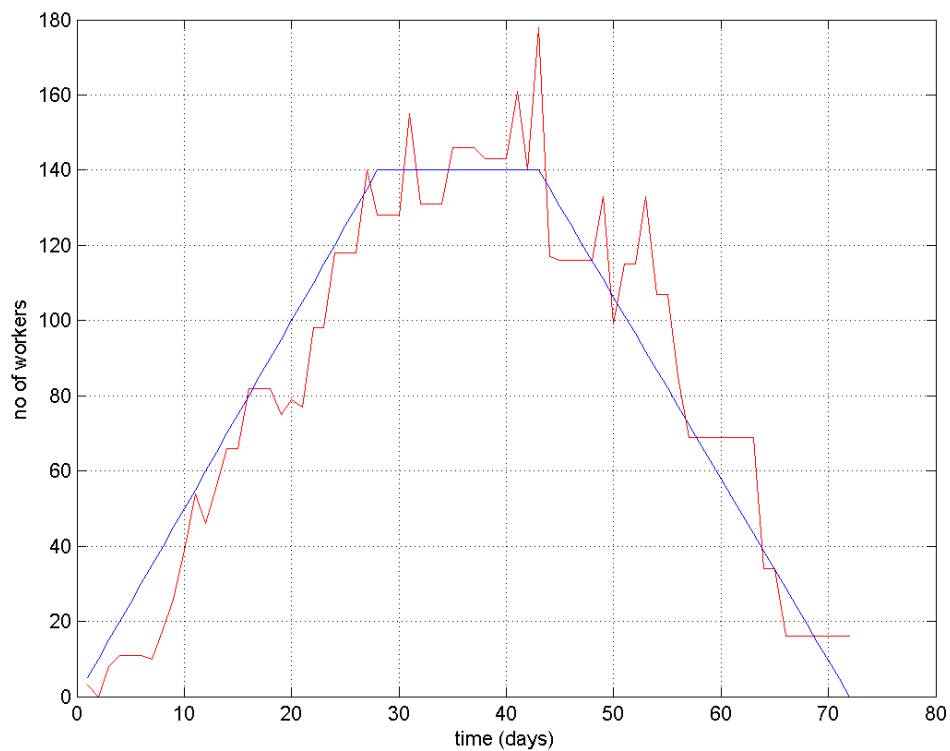
- **4^η Δοκιμή**

N = 140, max time= 150, percentage = 0.4, iterations= 200, trials= 40



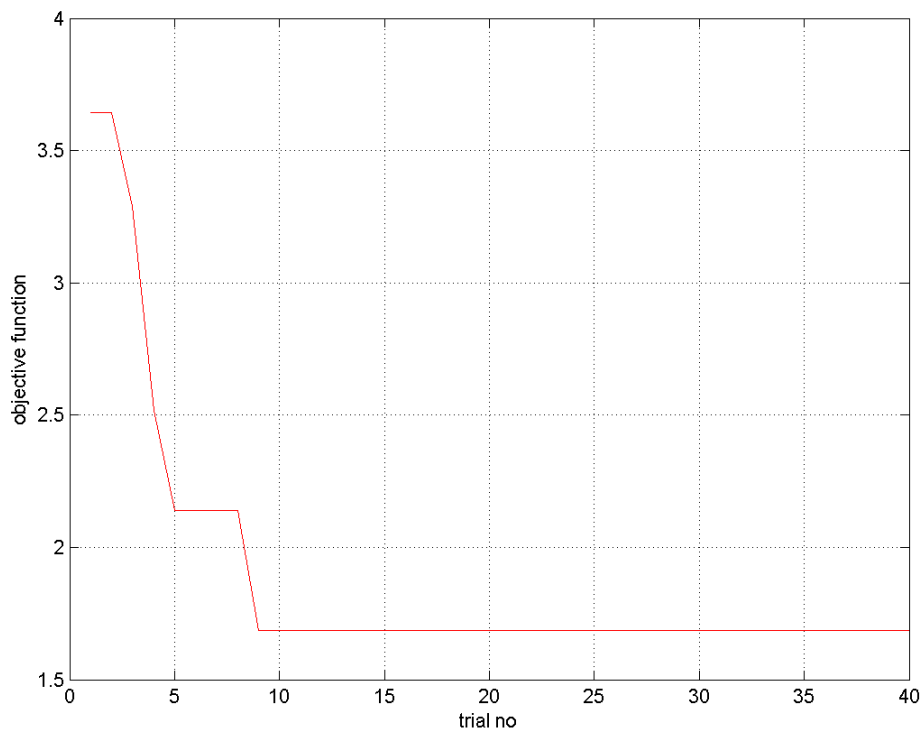
Σχήμα 5.44 Χρονοδιάγραμμα Εργασιών

(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)



Σχήμα 5.45 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

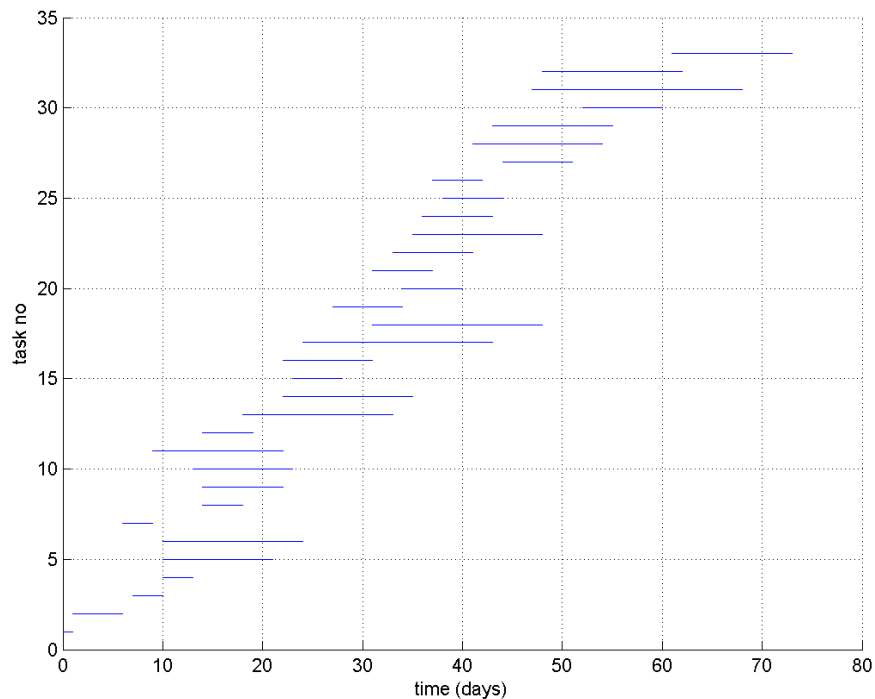
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)



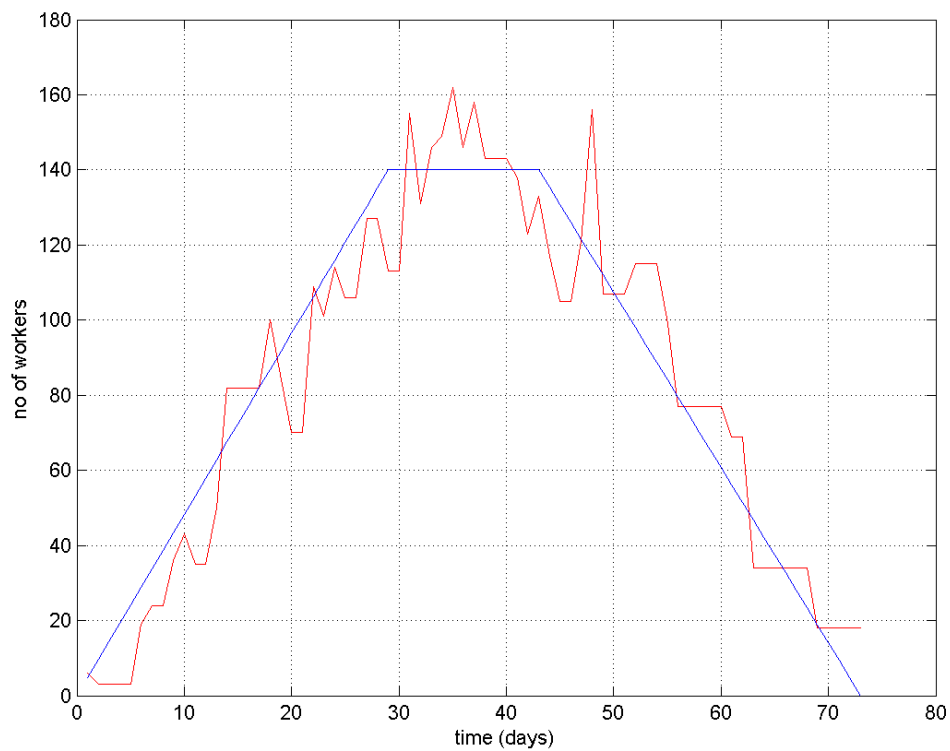
Σχήμα 5.46 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

- **5^η Δοκιμή**

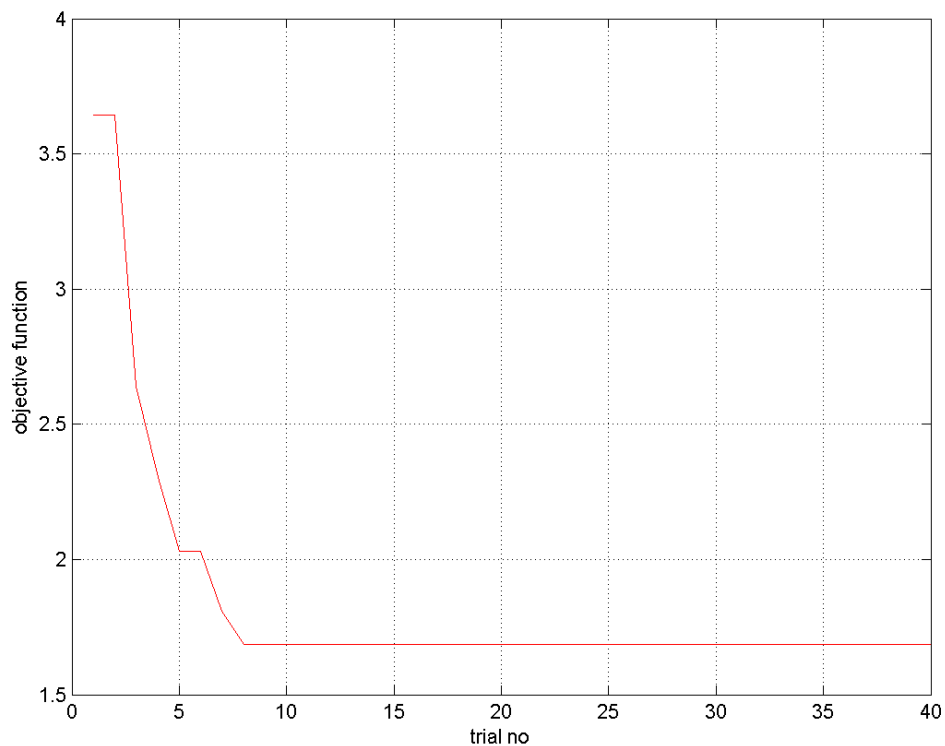
N = 140, max time= 150, percentage = 0.4, iterations= 200, trials= 40



Σχήμα 5.47 Χρονοδιάγραμμα Εργασιών
(N = 140, max time = 150, percentage = 0.4, iterations= 200, trials= 40)



Σχήμα 5.48 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
($N = 140$, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

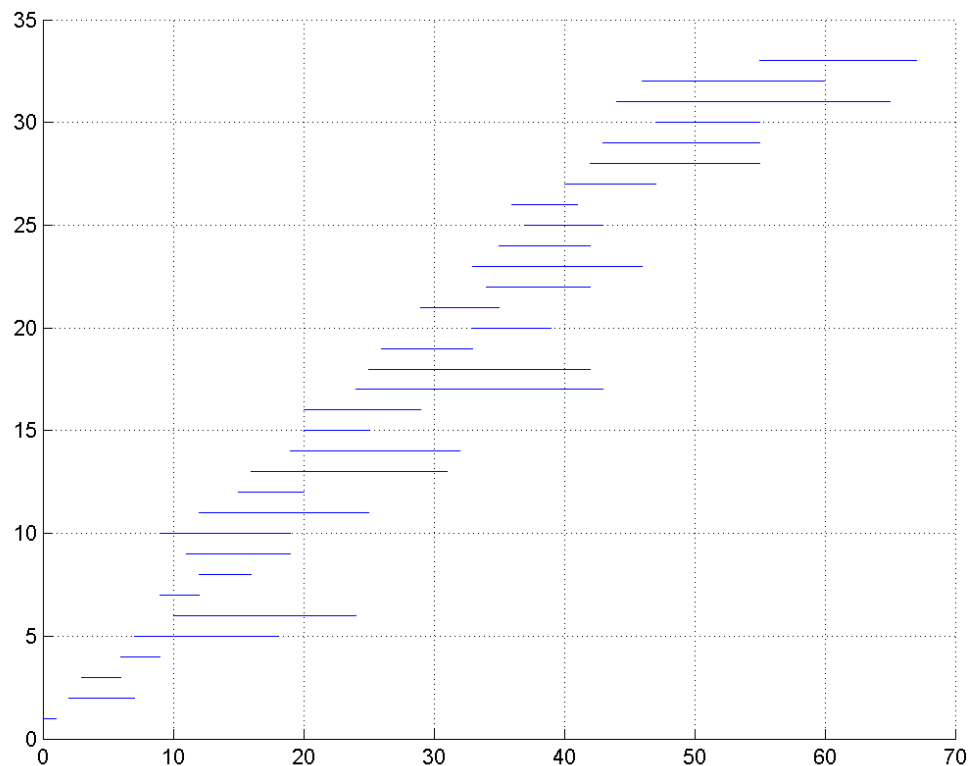


Σχήμα 5.49 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
($N = 140$, max time = 150, percentage = 0.4, iterations= 200, trials= 40)

Συνεχίζουμε τώρα τις αρχικές δοκιμές αυξάνοντας το percentage σταδιακά και για τα δεδομένα των $N = 140$, max time = 64, iterations= 100, trials= 40 και οι λύσεις που λαμβάνουμε εκτοξεύουν κατά πολύ τον αριθμό των εργατών άρα αυξάνουμε και το μέγιστο χρόνο κατά 5 μέρες maxtime=69. Με αυτά τα δεδομένα λαμβάνουμε μια καλή λύση κατά την πέμπτη επαναληπτική δοκιμή.

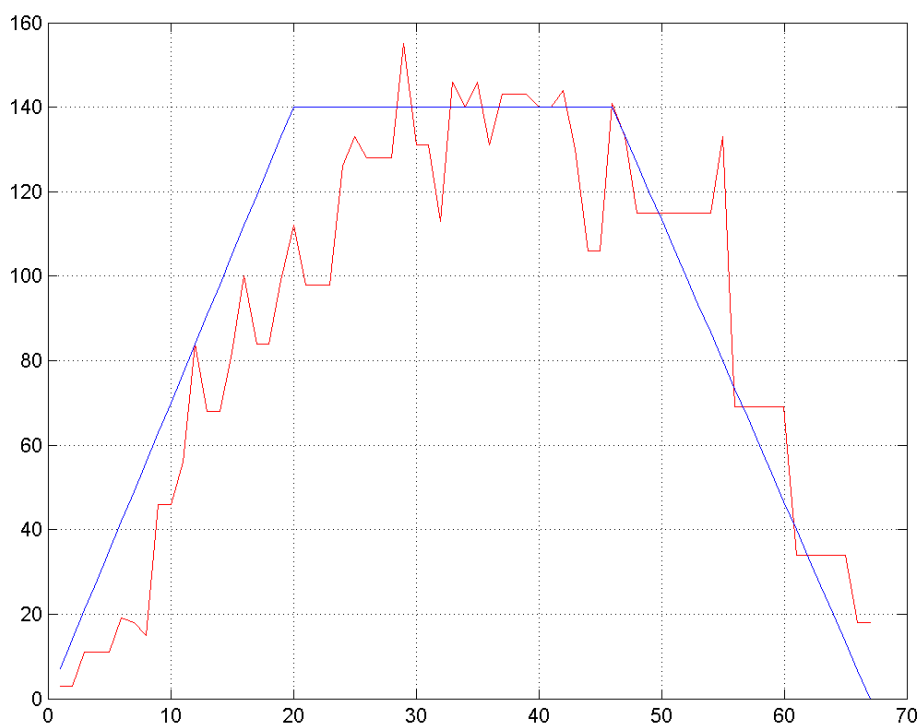
- **5^η Δοκιμή**

$N = 140$, max time= 69, percentage = 0.3, iterations= 100, trials= 40



Σχήμα 5.50 Χρονοδιάγραμμα Εργασιών

($N = 140$, max time= 69, percentage = 0.3, iterations= 100, trials= 40)



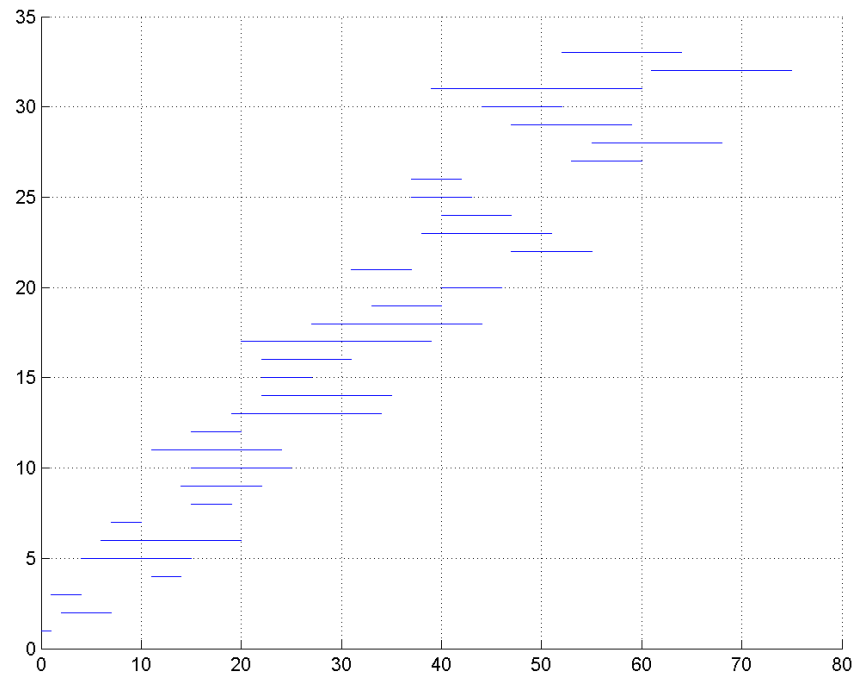
(καλή λύση, κοντά στο επιθυμητό διάγραμμα των εργατών)

Σχήμα 5.51 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
(N = 140, max time= 69, percentage = 0.3, iterations= 100, trials= 40)

Συμπεράσματα : Ως τώρα κοντά στο max time της βέλτιστης, από τον χρονικό προγραμματισμό, λύσης δεν μπορούμε να περιορίσουμε δραστικά τον αριθμό των εργατών κρατώντας παράλληλα μια ομαλή προεπιλεγμένη γραμμή. Ακόμα, το percentage ,ο συντελεστής δηλαδή που μεταβάλλει την επιθυμητή κατανομή των εργατών, για ακραίες τιμές ,μικρότερες του 0,15 και μεγαλύτερες του 0,35 συσσωρεύει μεγάλο αριθμό εργατών στην αρχή και το τέλος, και στη μέση του έργου αντίστοιχα. Άρα σαν δεύτερο βήμα θα προσπαθήσουμε να “ανοίξουμε” χρονικά περισσότερο το έργο δίδοντας μεγαλύτερο περιθώριο στο max time του έργου, μειώνοντας τους εργάτες που ουσιαστικά μας ενδιαφέρει, και με το percentage να κυμαίνεται μεταξύ του 0,2 και του 0,4.

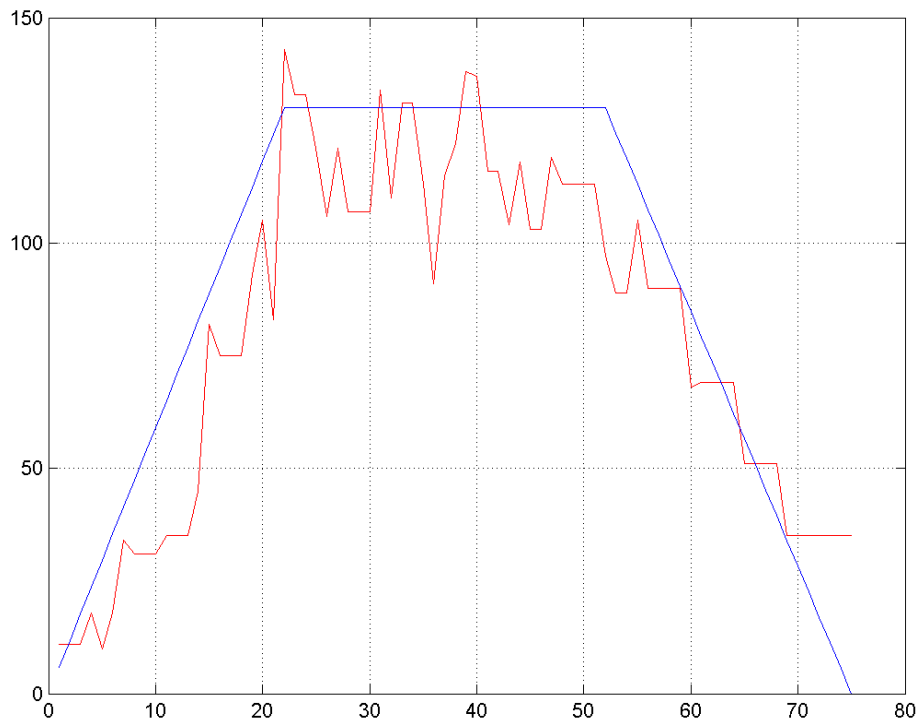
- **19^η Δοκιμή**

N = 130, max time= 80, percentage = 0.3, iterations= 100, trials= 40



Σχήμα 5.52 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 80, percentage = 0.3, iterations= 100, trials= 40)

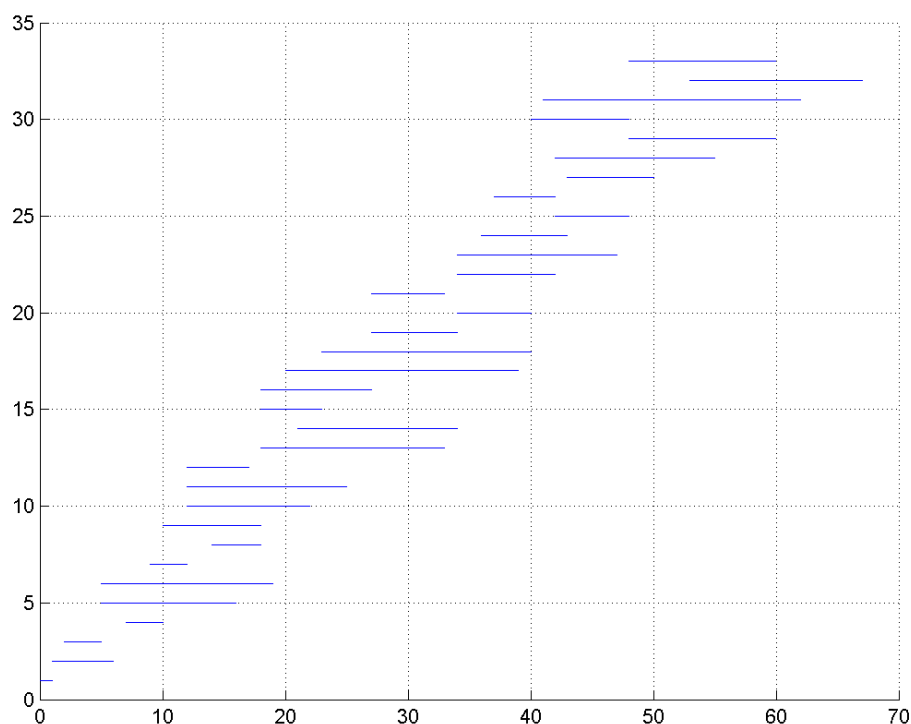


Σχήμα 5.53 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 80, percentage = 0.3, iterations= 100, trials= 40)

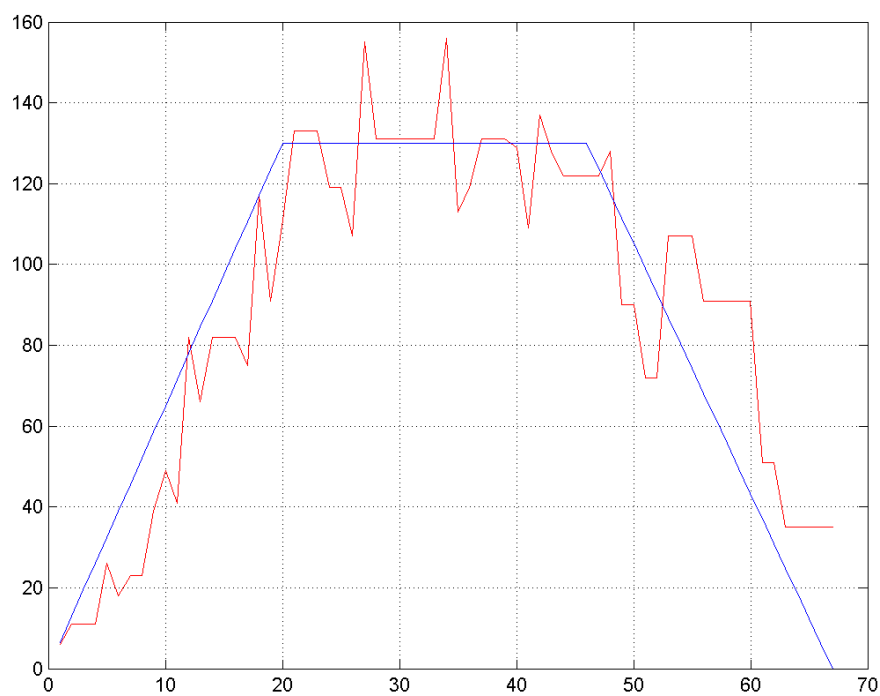
• **34^η Δοκιμή**

(N = 130, max time= 90, percentage = 0.3, iterations= 100, trials= 40)



Σχήμα 5.54 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 90, percentage = 0.3, iterations= 100, trials= 40)

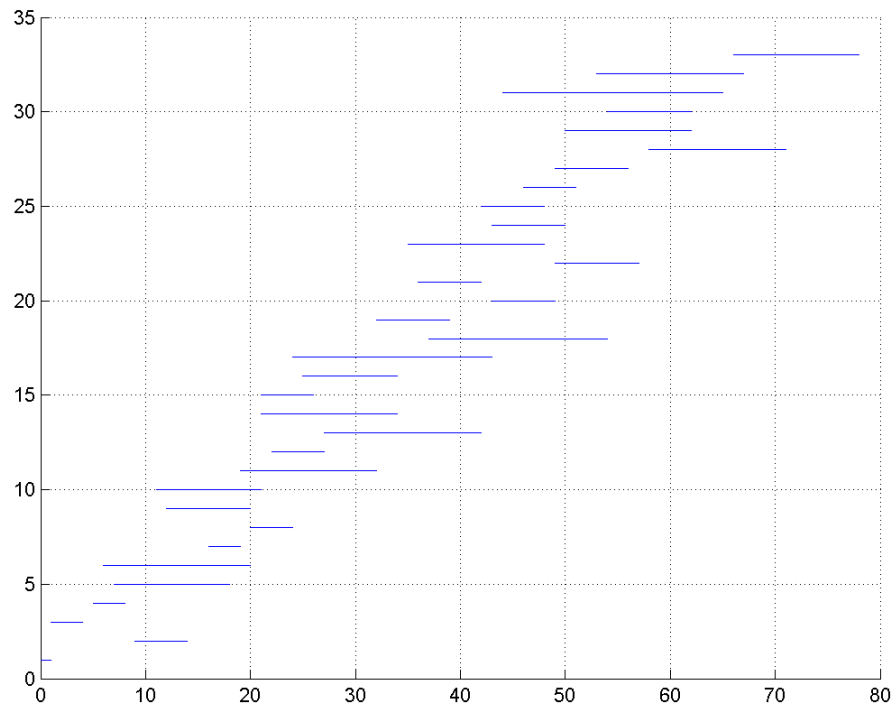


Σχήμα 5.55 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 90, percentage = 0.3, iterations= 100, trials= 40)

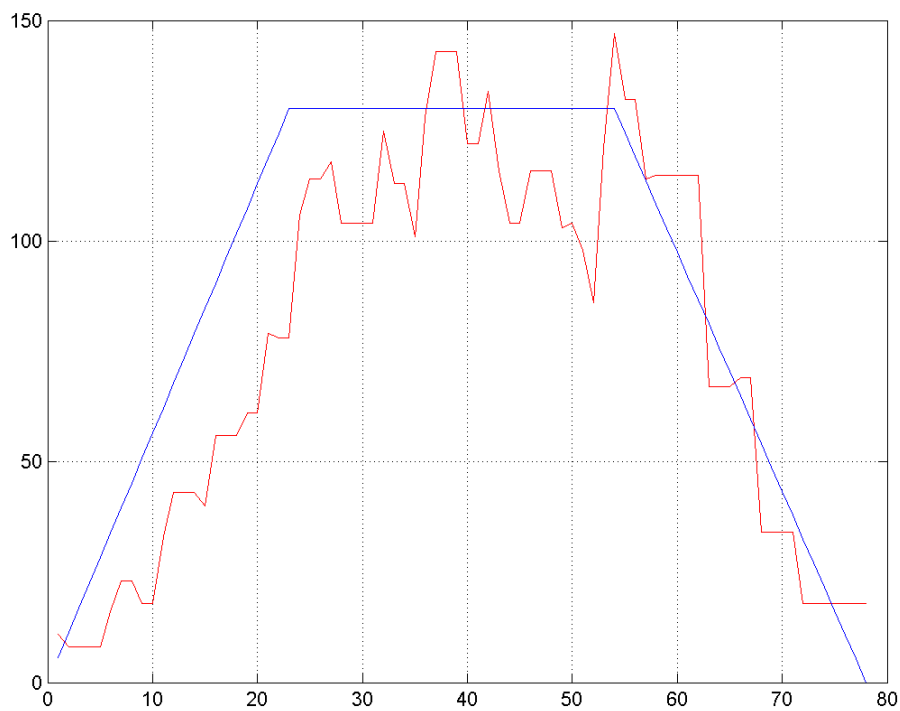
- **28^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)



Σχήμα 5.56 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)

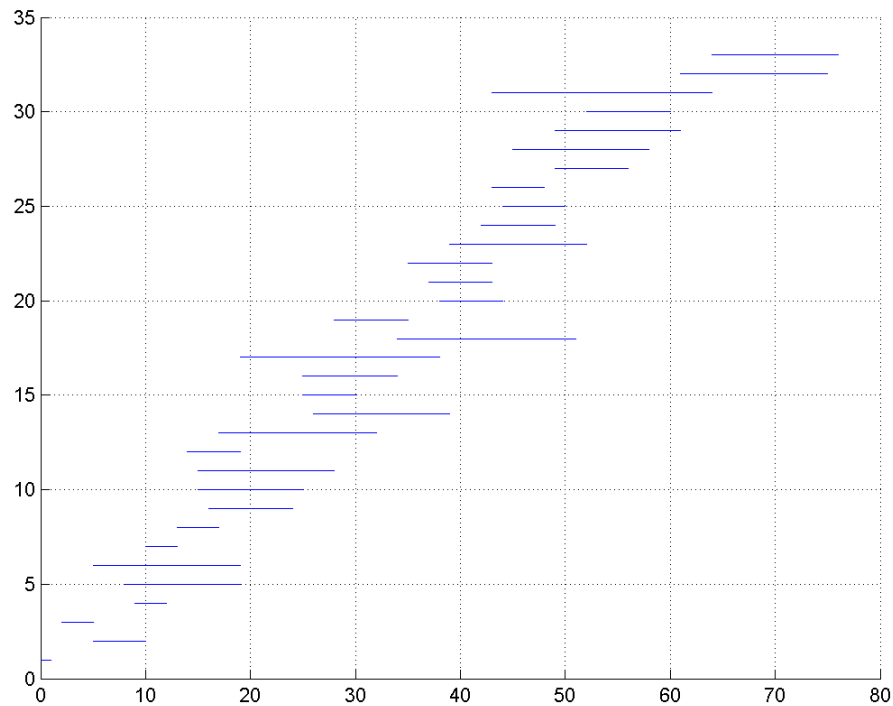


Σχήμα 5.57 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)

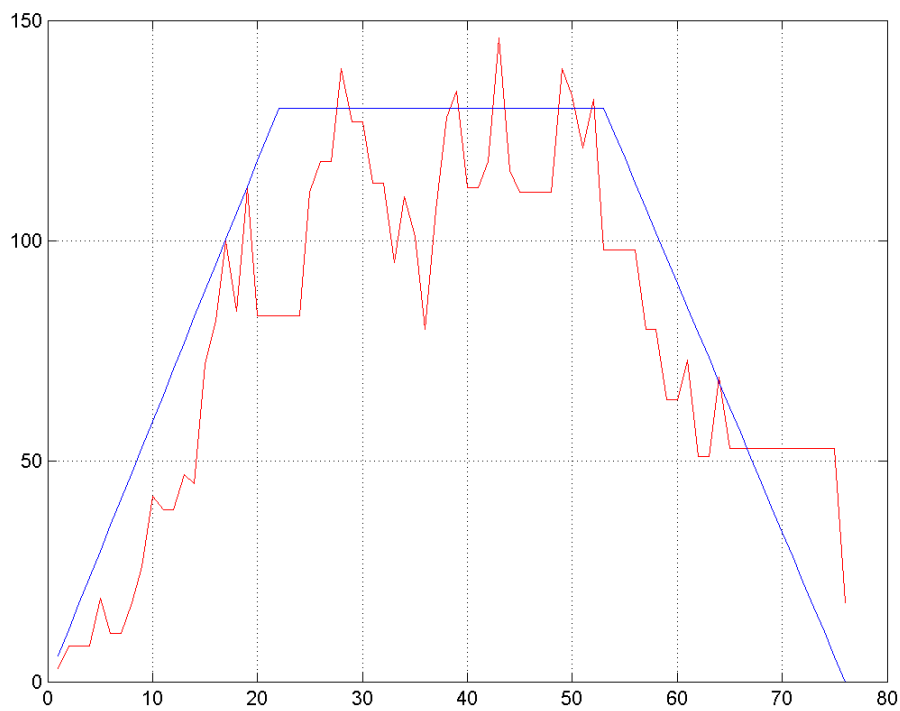
- **36^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)



Σχήμα 5.58 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)

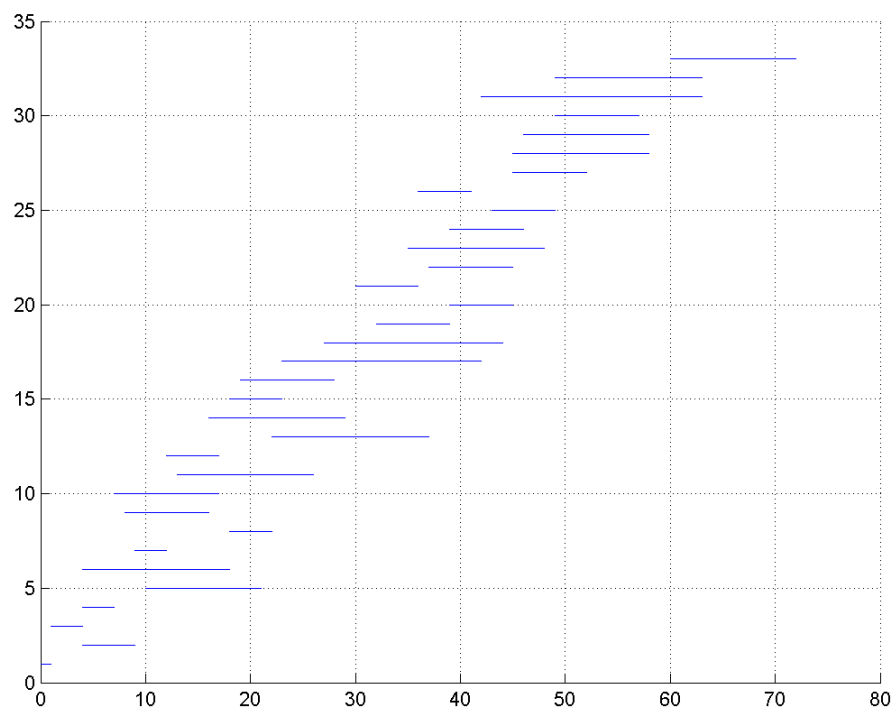


Σχήμα 5.59 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)

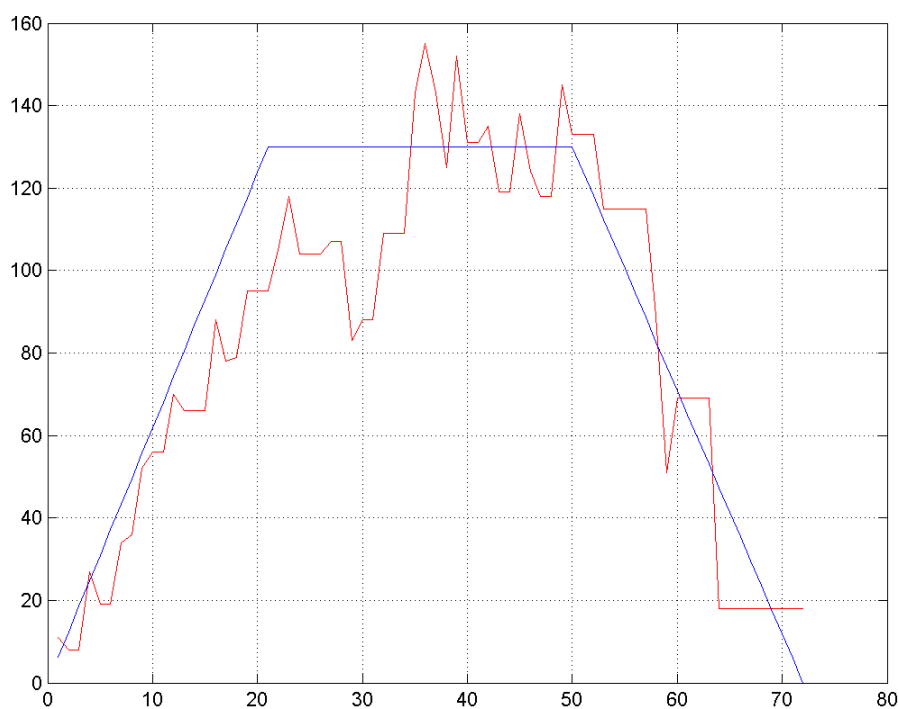
• **39^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)



Σχήμα 5.60 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)

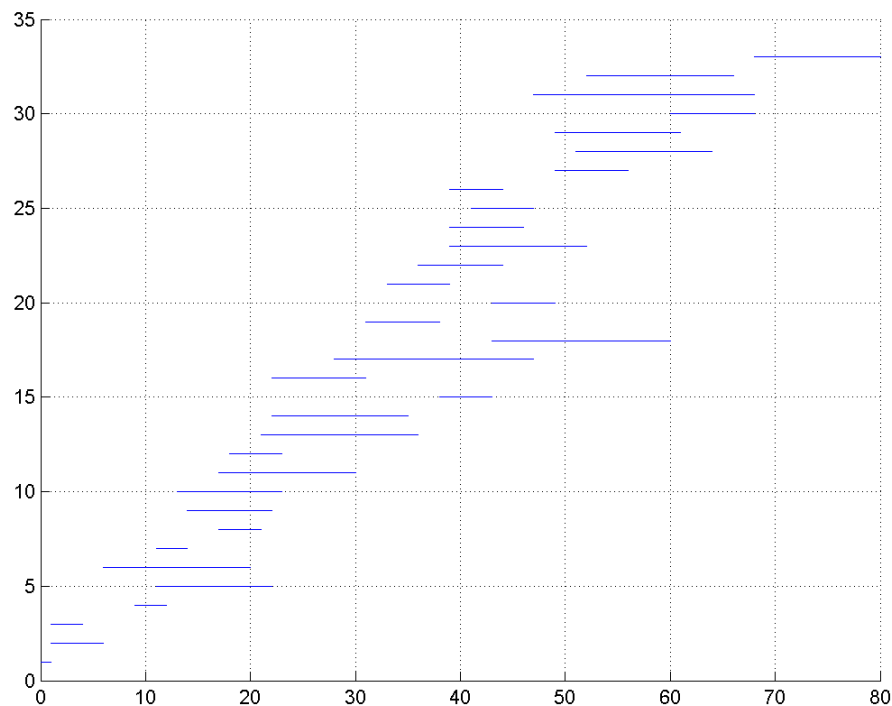


Σχήμα 5.61 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)

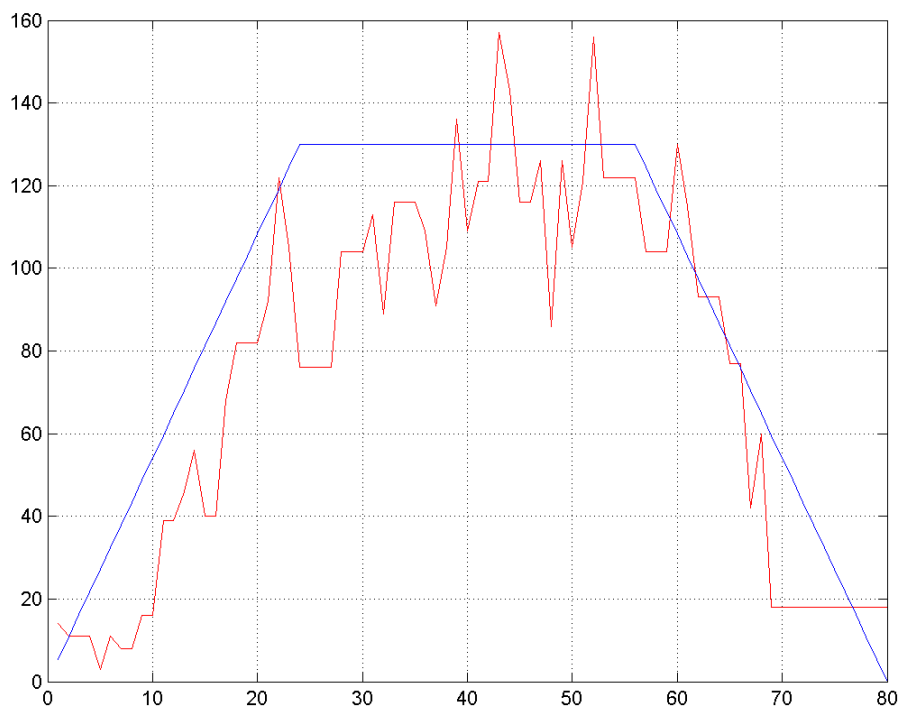
- **44^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)



Σχήμα 5.62 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)

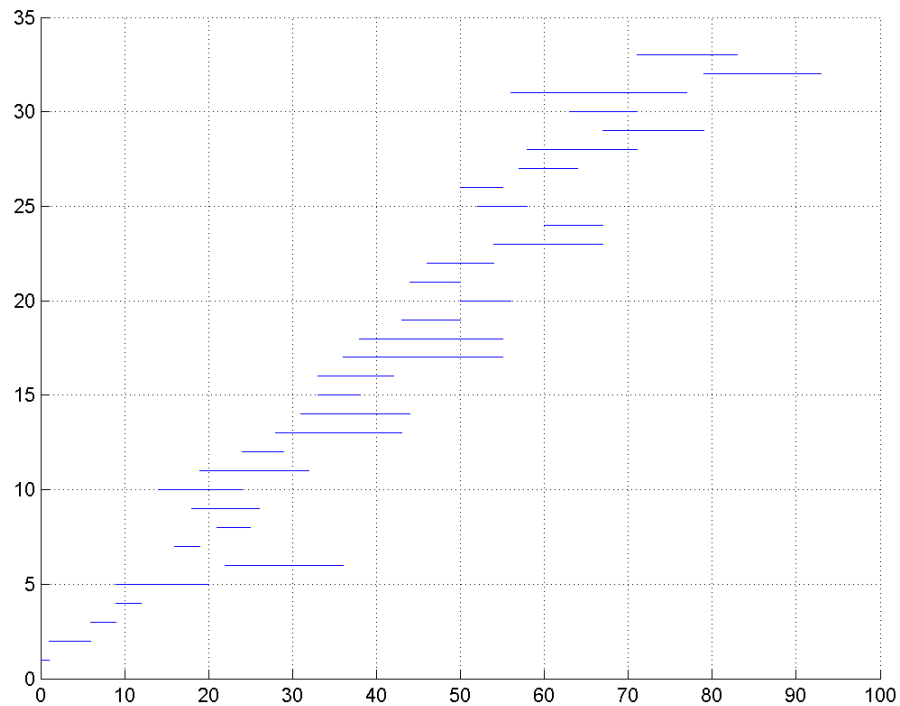


Σχήμα 5.63 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.3, iterations= 100, trials= 40)

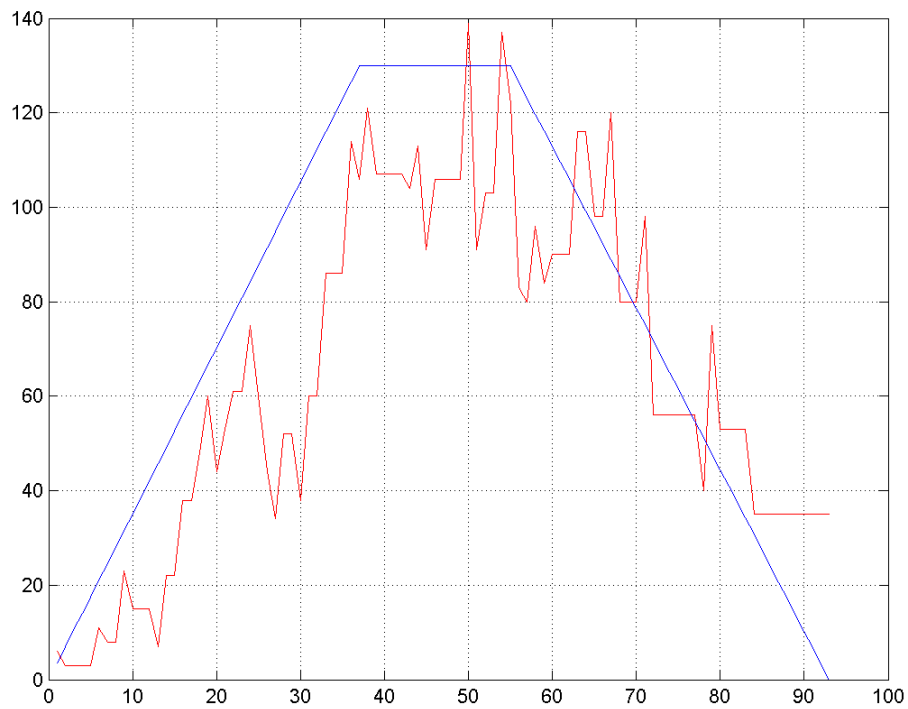
- **2^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.64 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

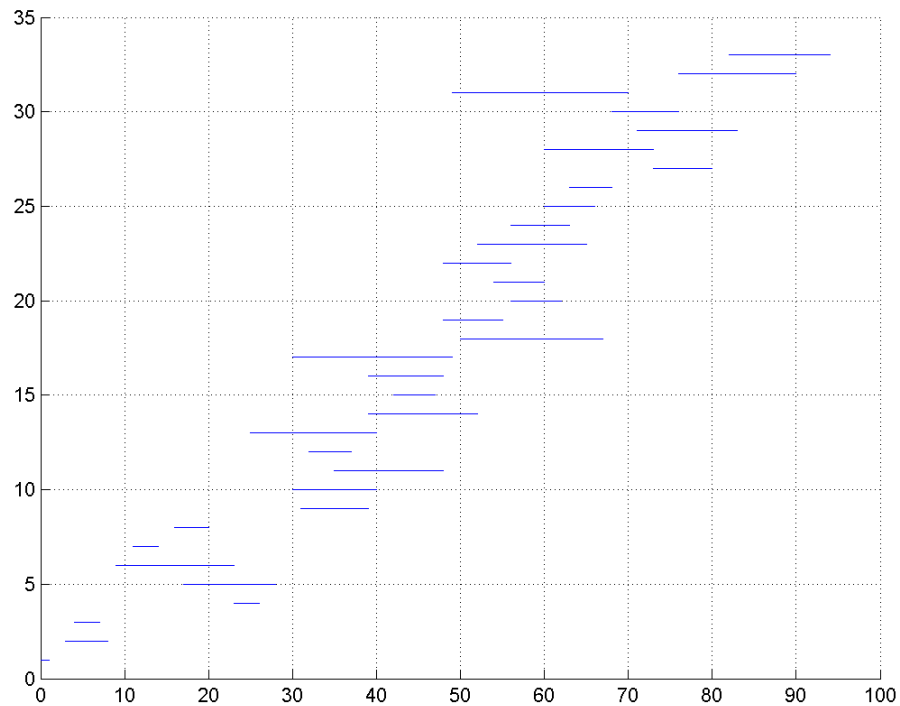


Σχήμα 5.65 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

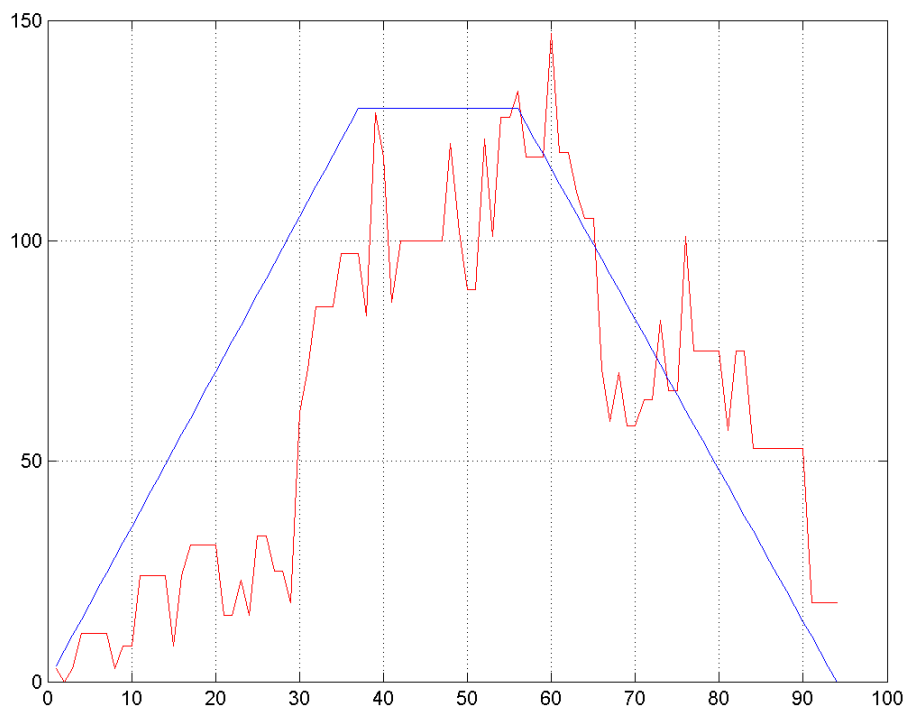
• **3^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.66 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

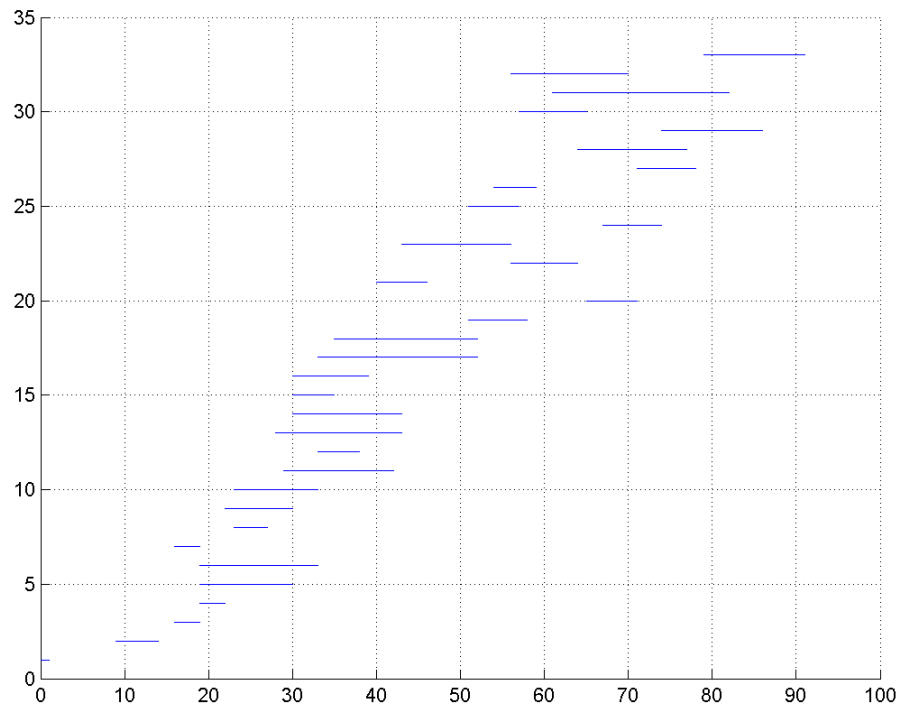


Σχήμα 5.67 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

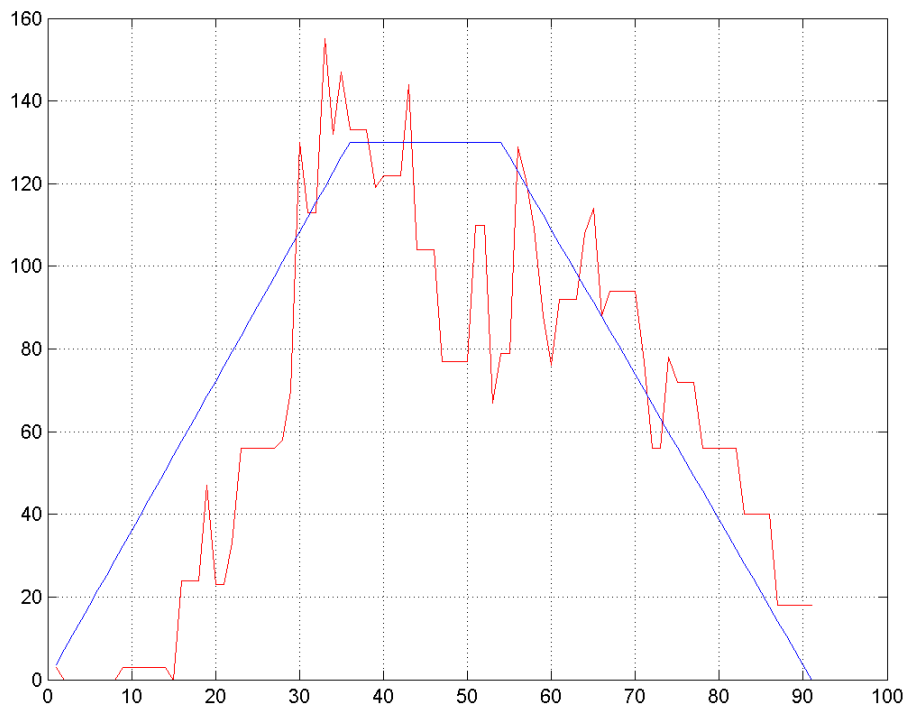
• **4^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.68 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

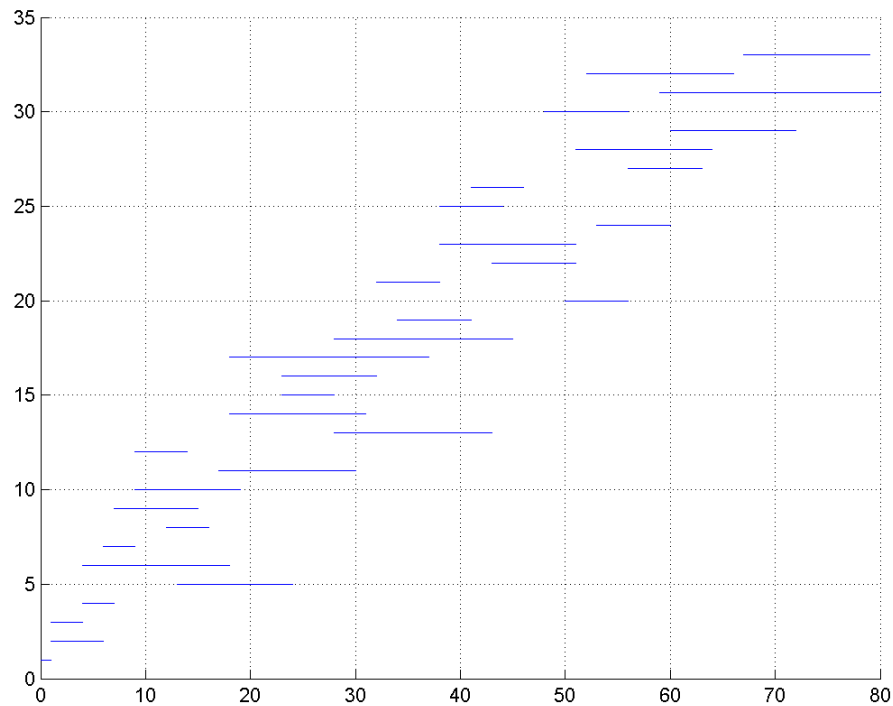


Σχήμα 5.69 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

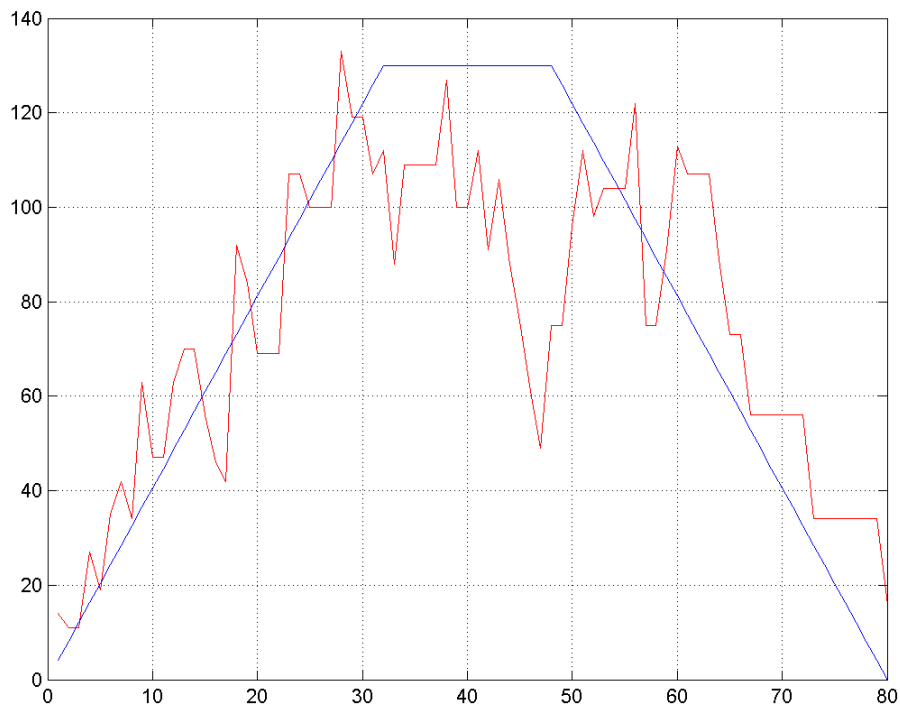
- **7^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.70 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

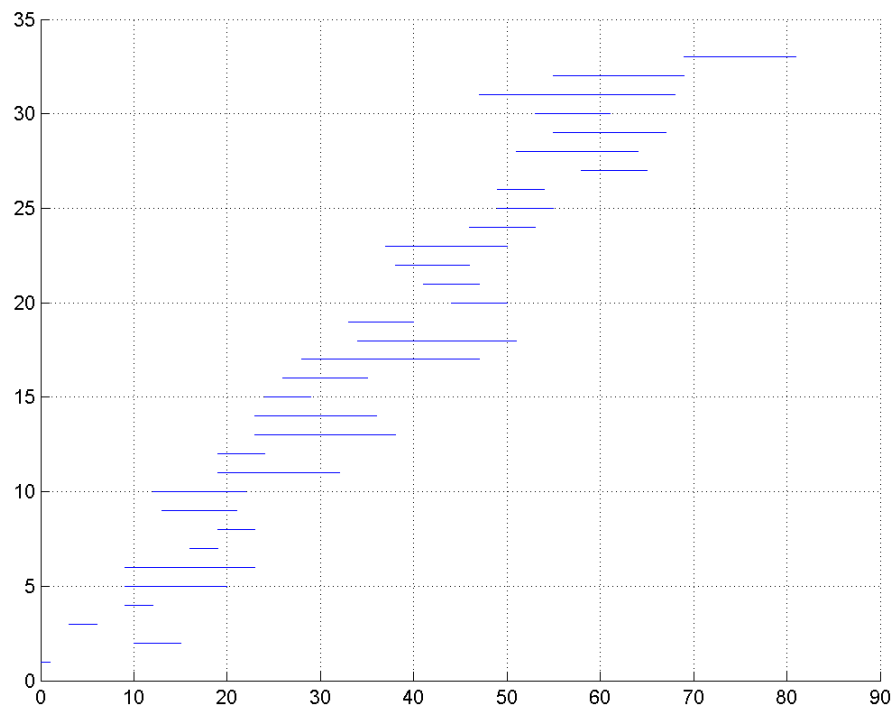


Σχήμα 5.71 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

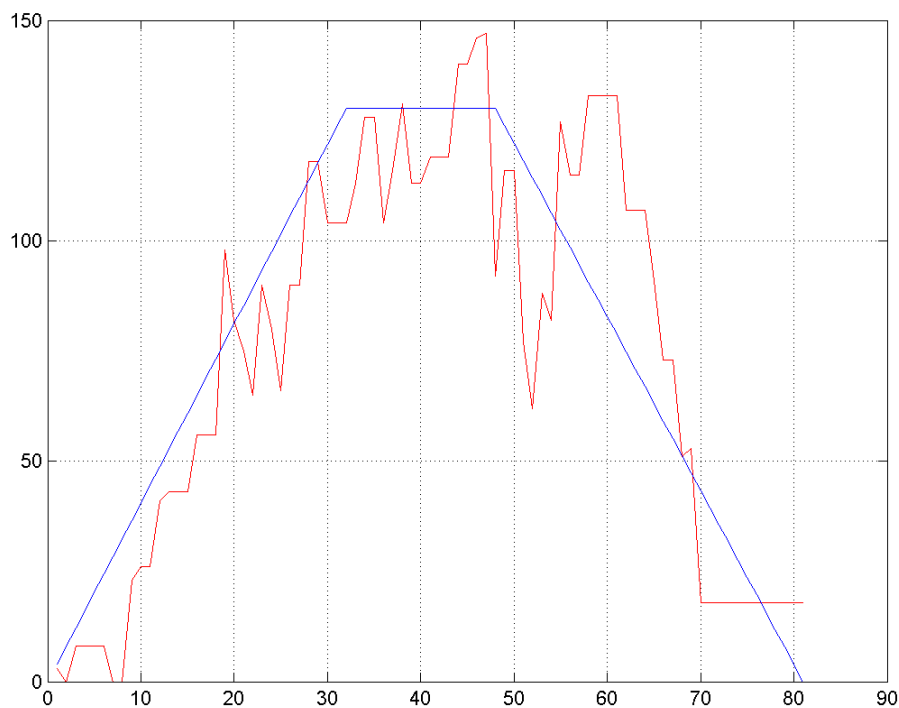
- **13^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.72 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

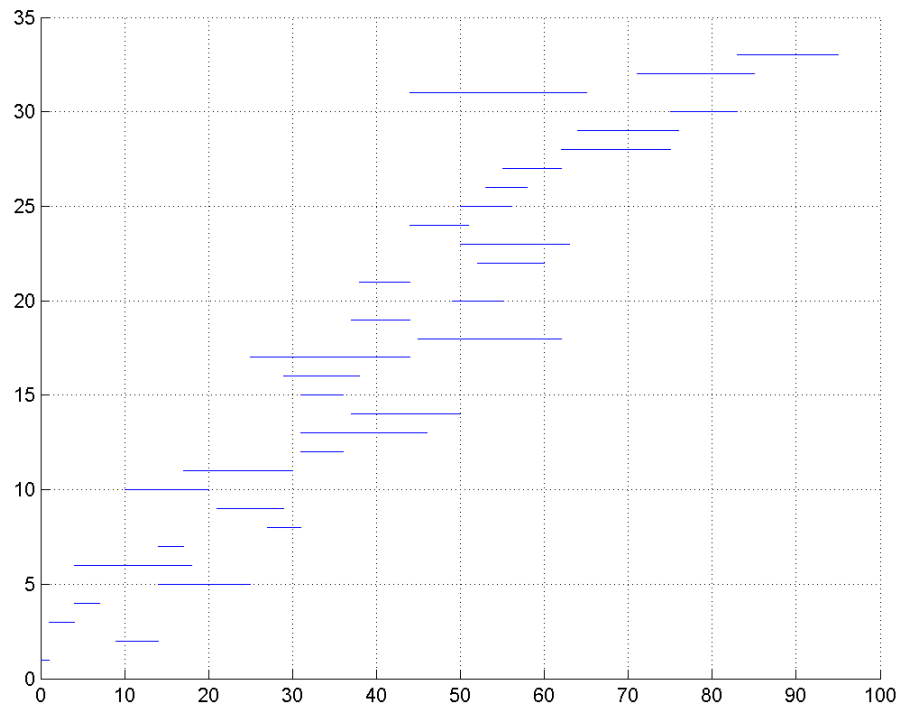


Σχήμα 5.73 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

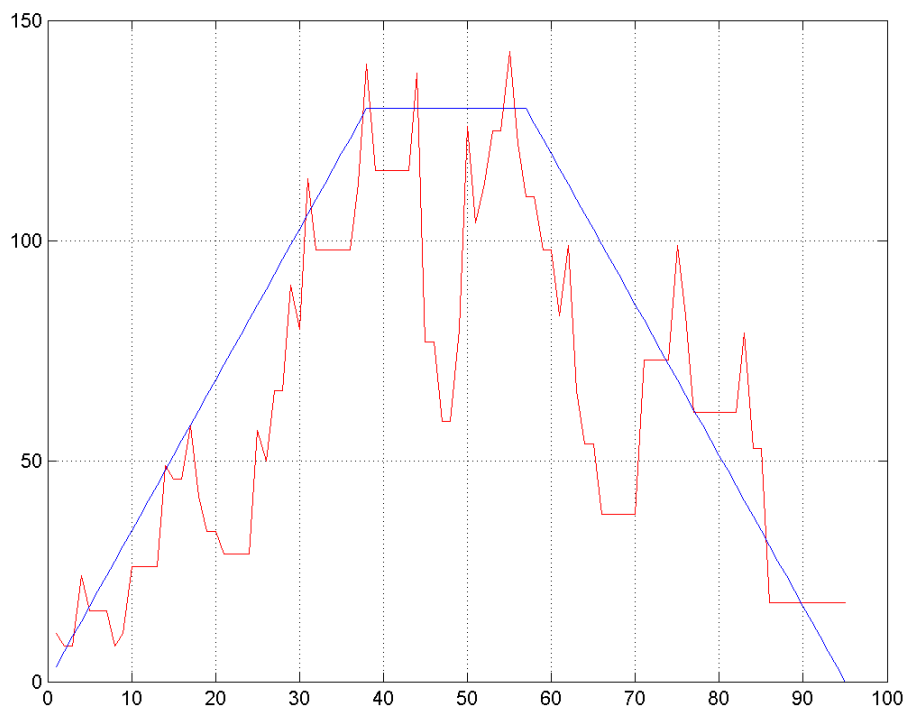
- **16^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.74 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

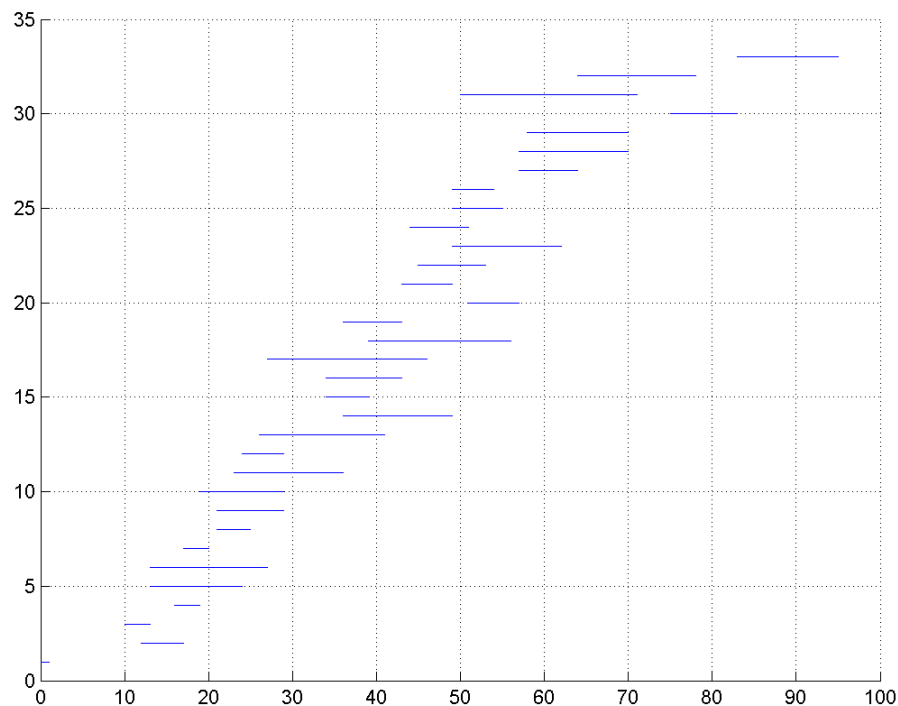


Σχήμα 5.75 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

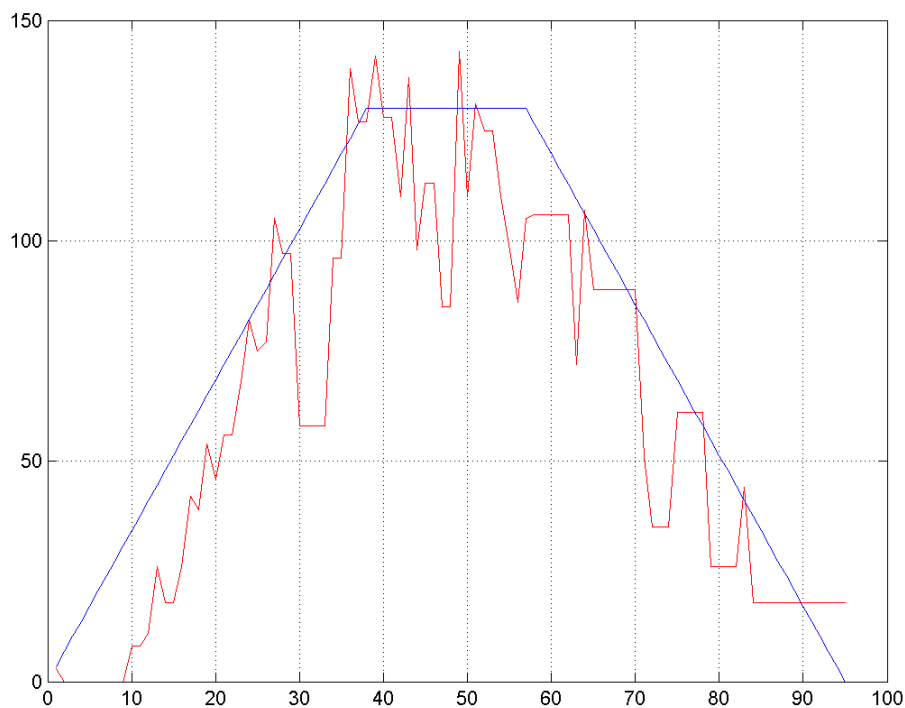
- **17^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.76 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

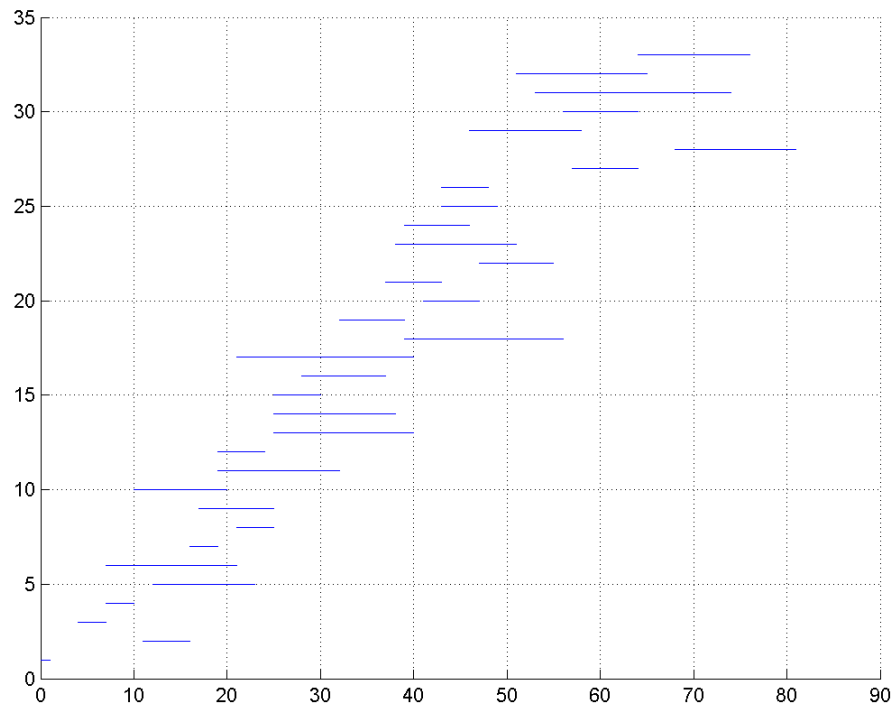


Σχήμα 5.77 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

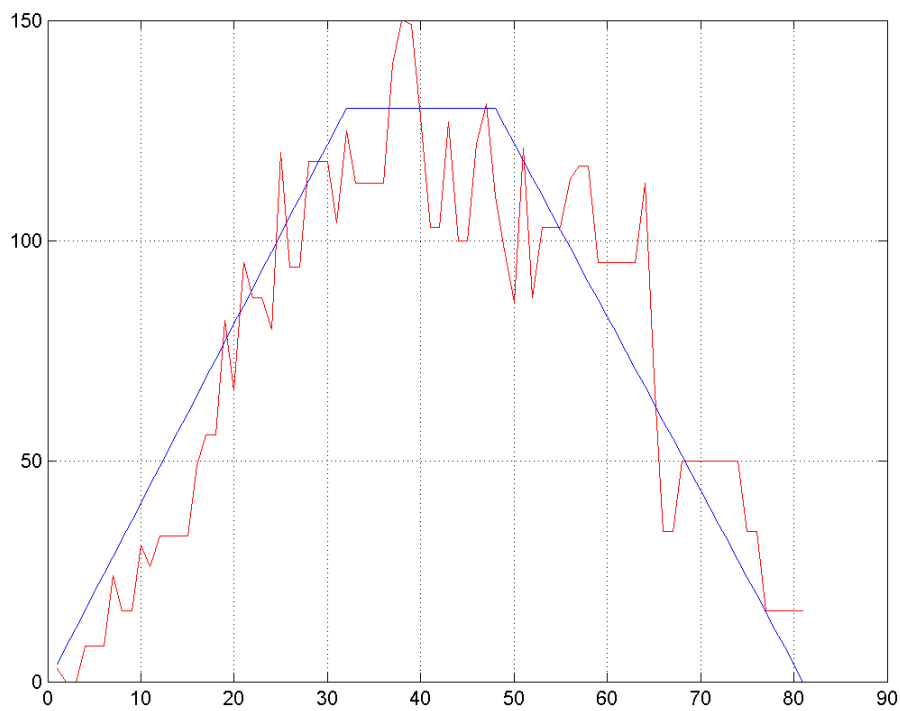
- **21^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.78 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

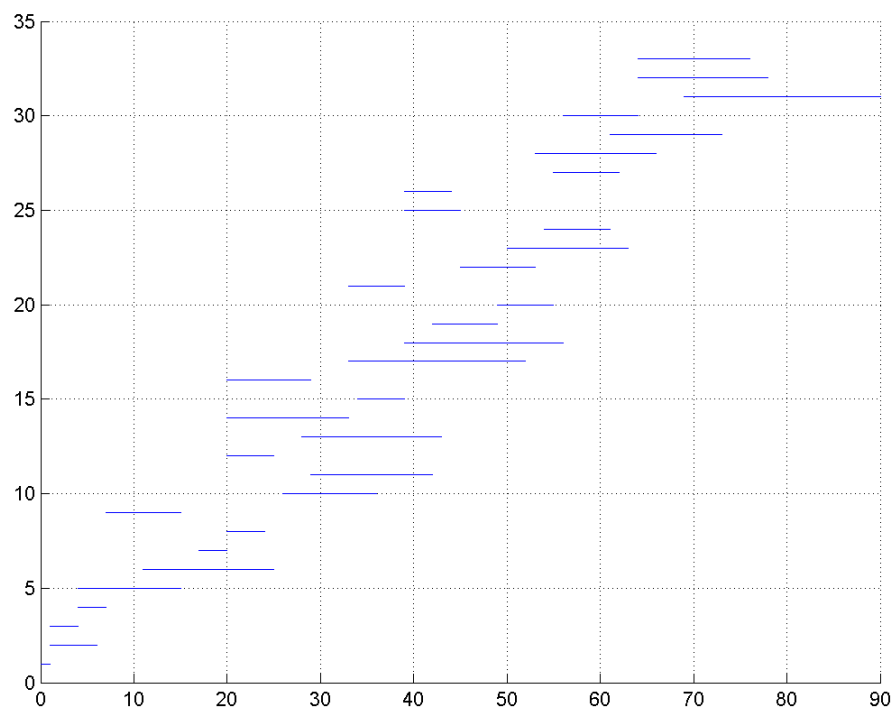


Σχήμα 5.79 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

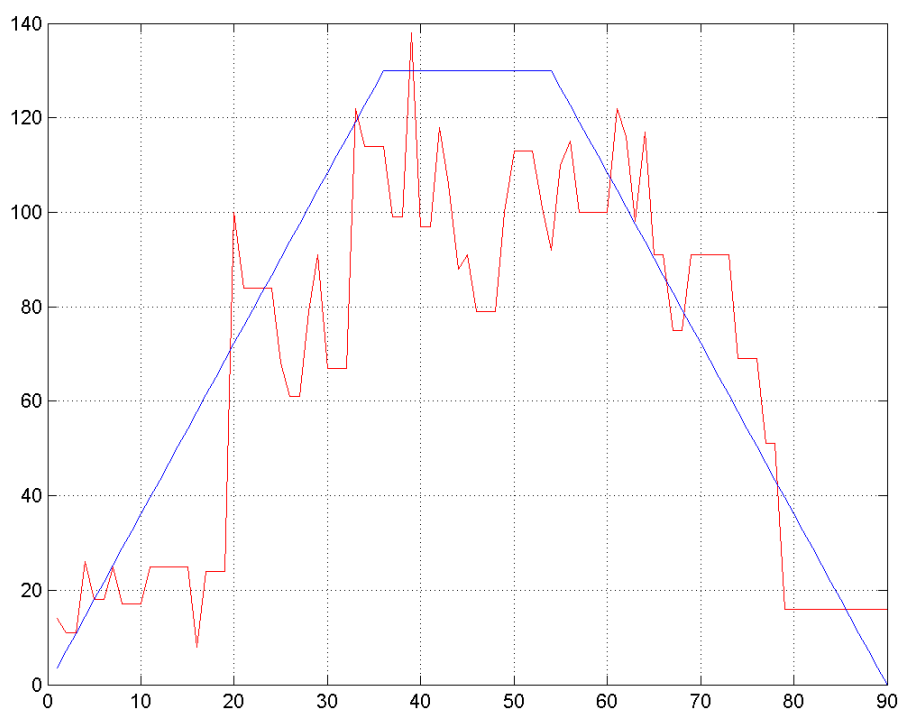
- **24^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.80 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

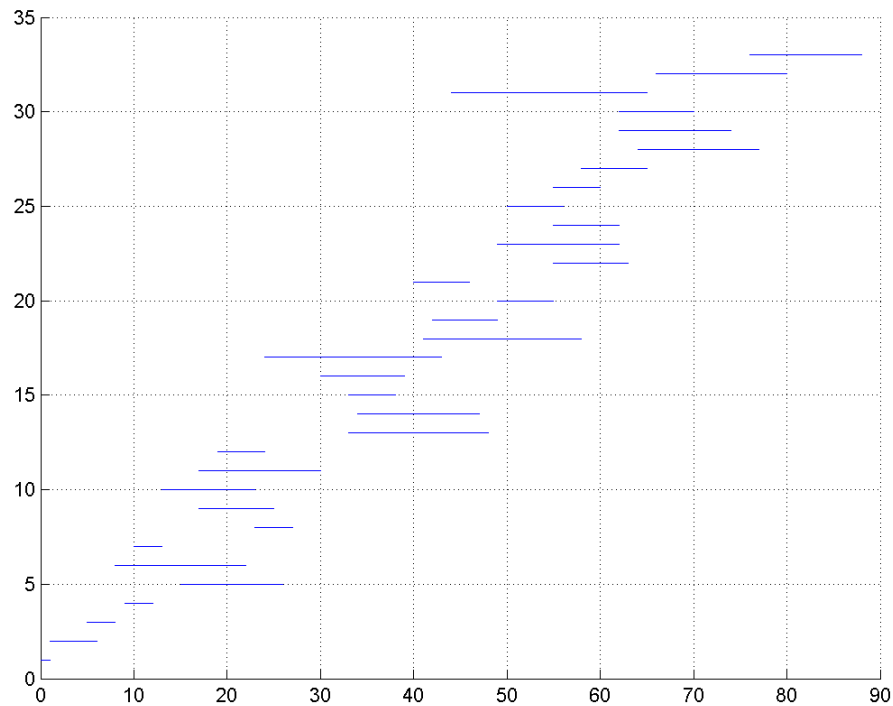


Σχήμα 5.81 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

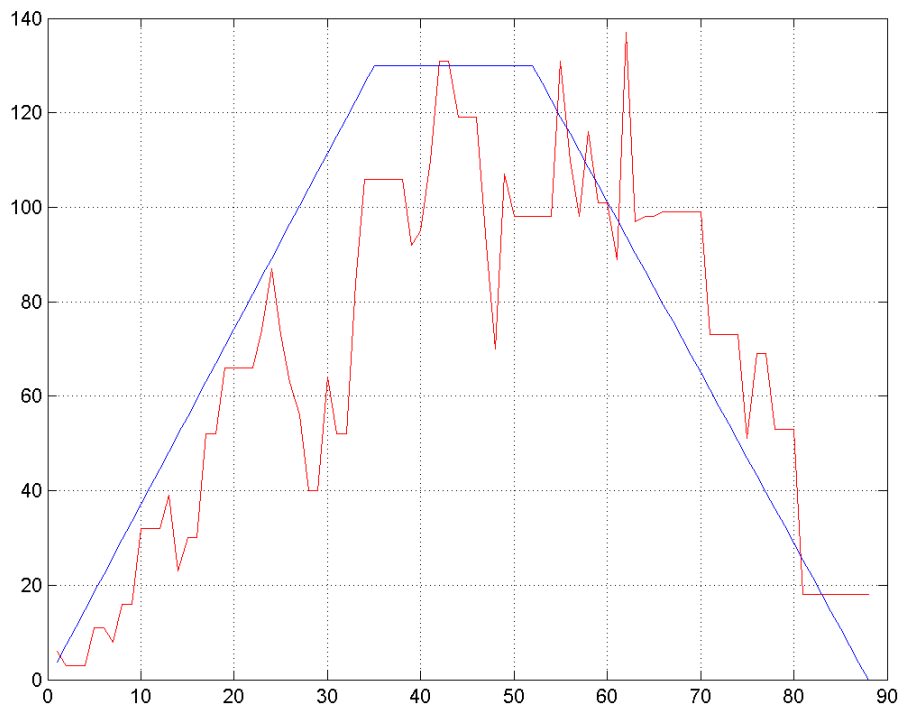
• **30^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.82 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

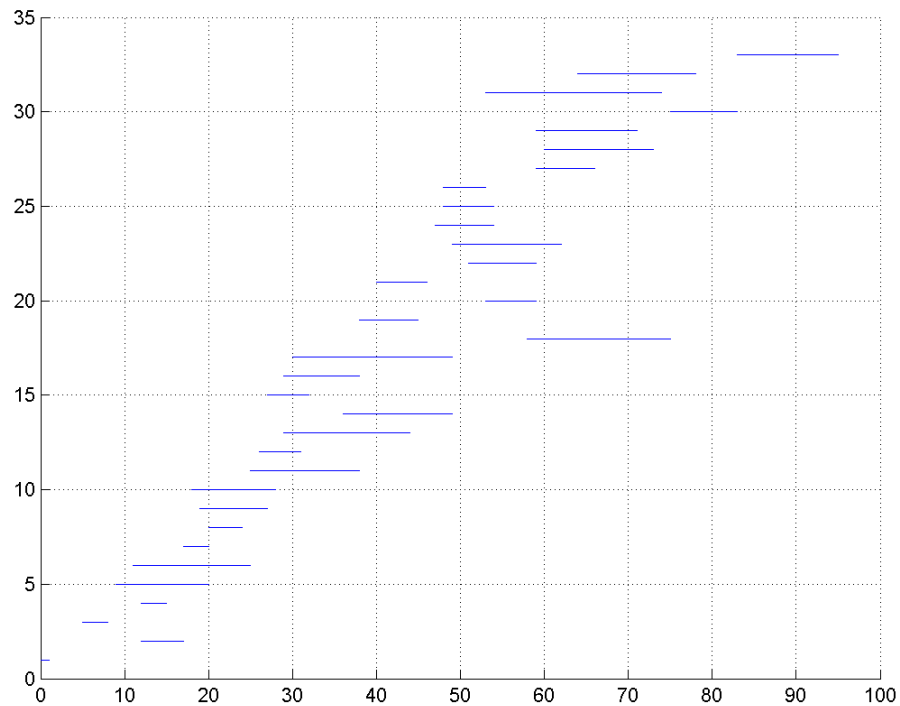


Σχήμα 5.83 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

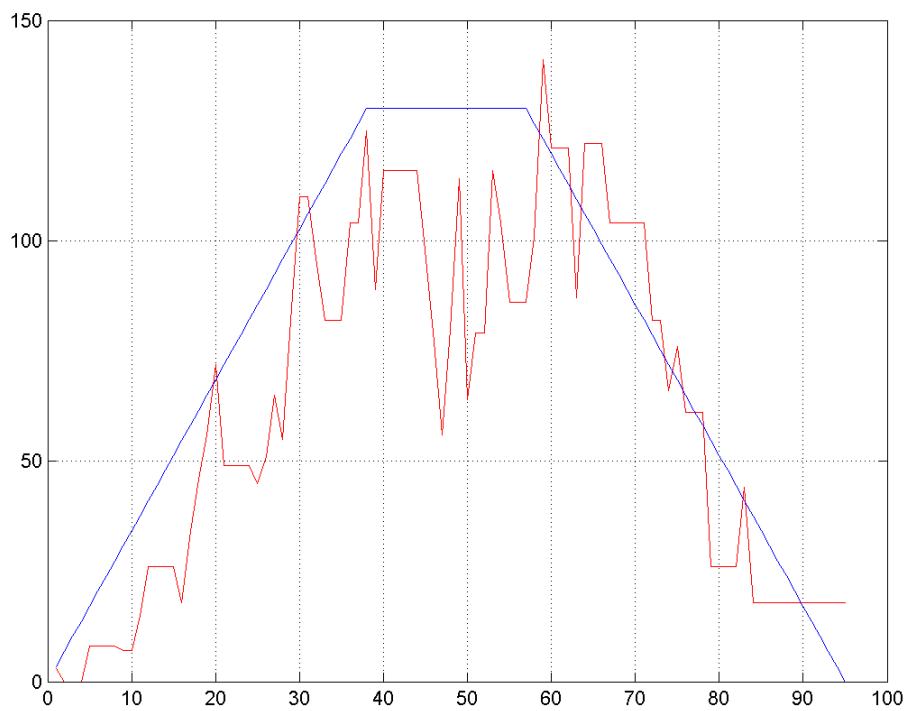
- **35^η Δοκιμή**

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.84 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

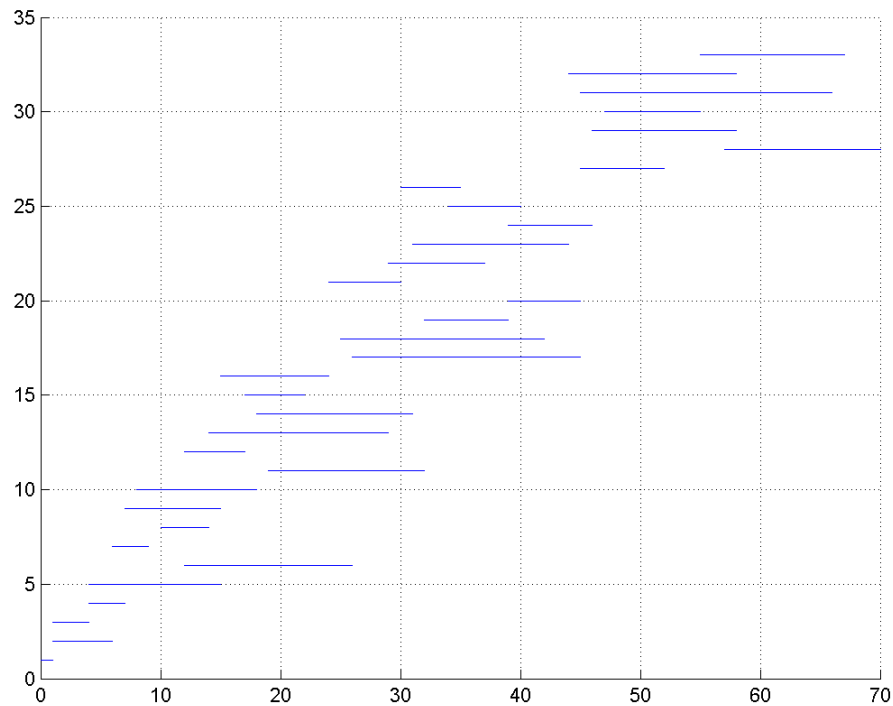


Σχήμα 5.85 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 95, percentage = 0.4, iterations= 100, trials= 40)

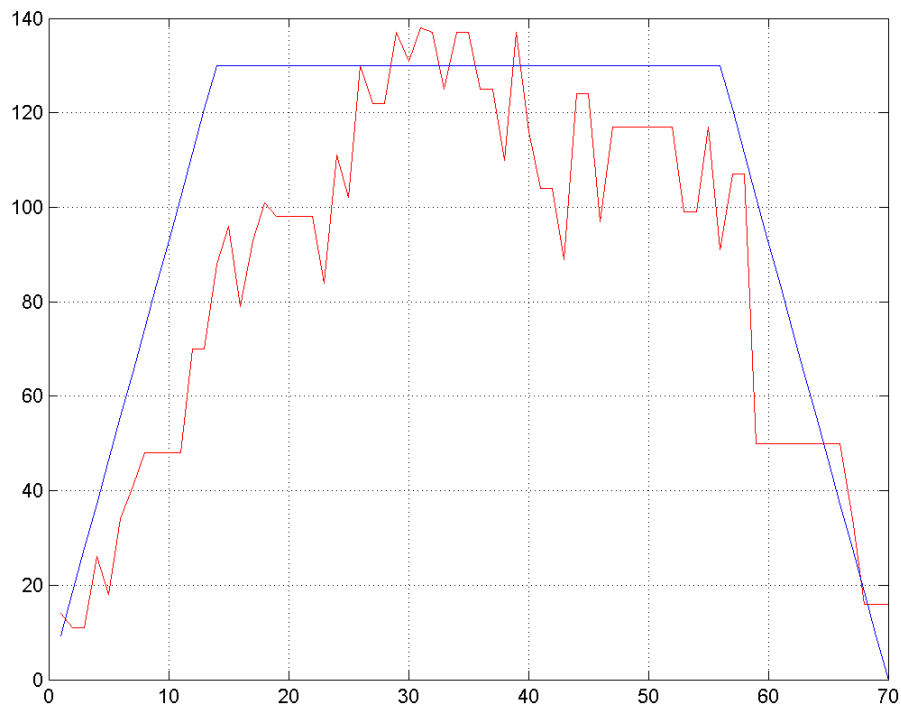
• **20^η Δοκιμή**

(N = 130, max time= 100, percentage = 0.2, iterations= 100, trials= 40)



Σχήμα 5.86 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 100, percentage = 0.2, iterations= 100, trials= 40)

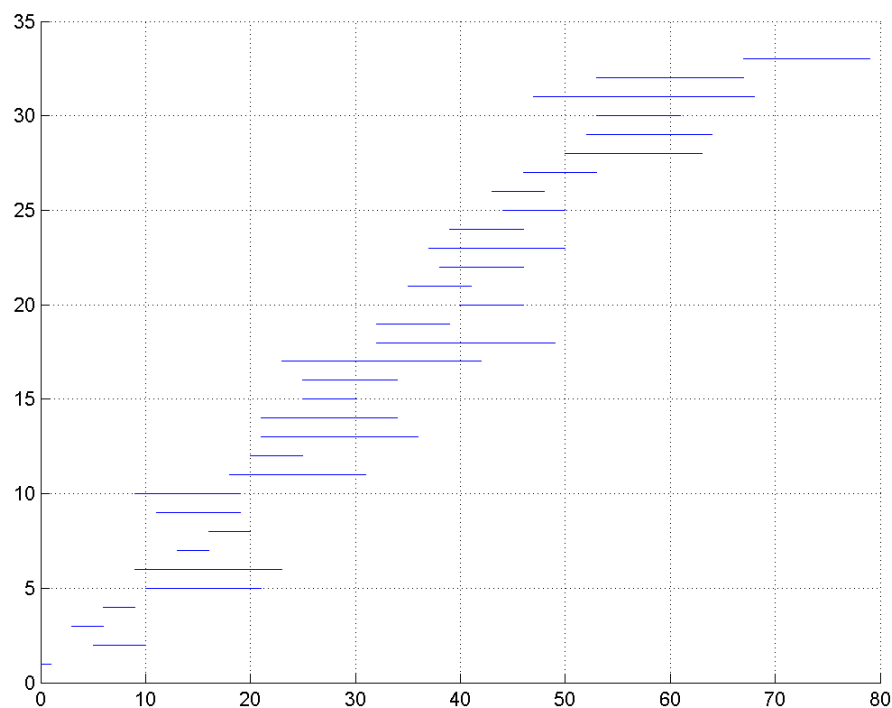


Σχήμα 5.87 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 100, percentage = 0.2, iterations= 100, trials= 40)

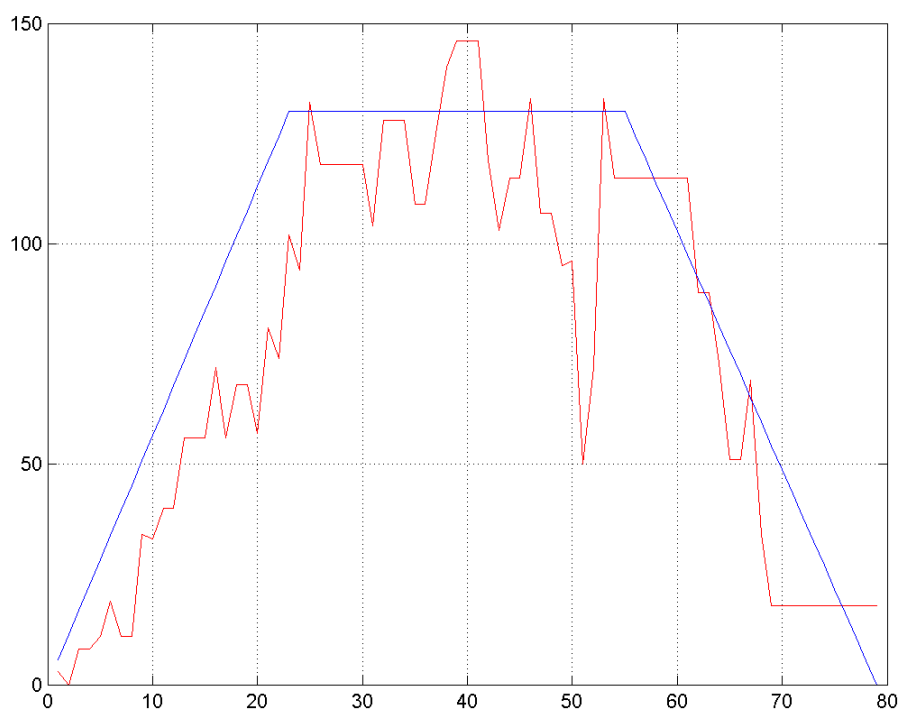
- **12^η Δοκιμή**

(N = 130, max time= 100, percentage = 0.3, iterations= 100, trials= 40)



Σχήμα 5.88 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 100, percentage = 0.3, iterations= 100, trials= 40)

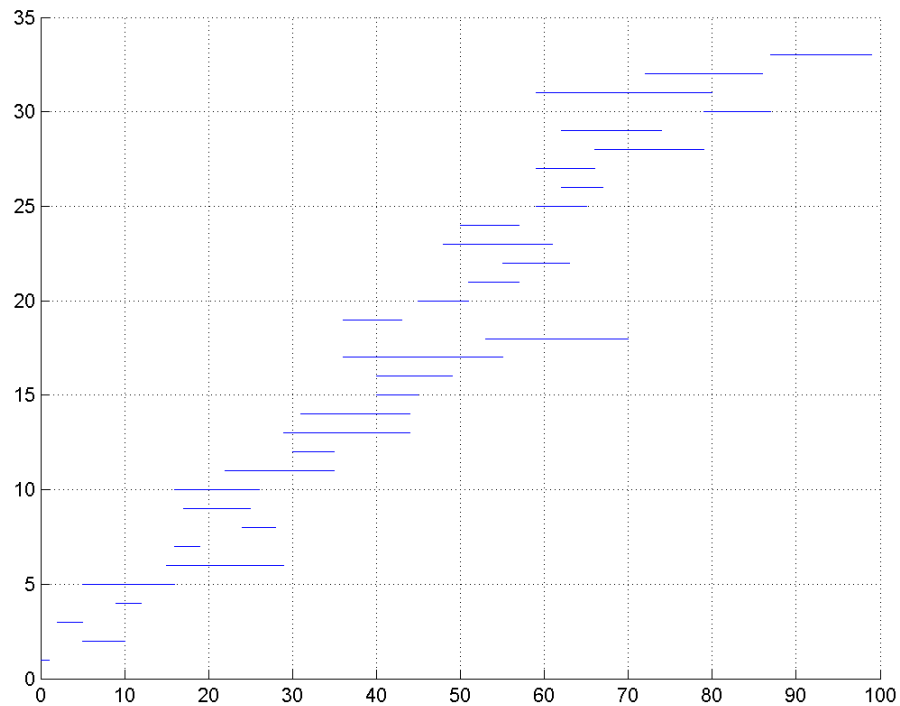


Σχήμα 5.89 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 100, percentage = 0.3, iterations= 100, trials= 40)

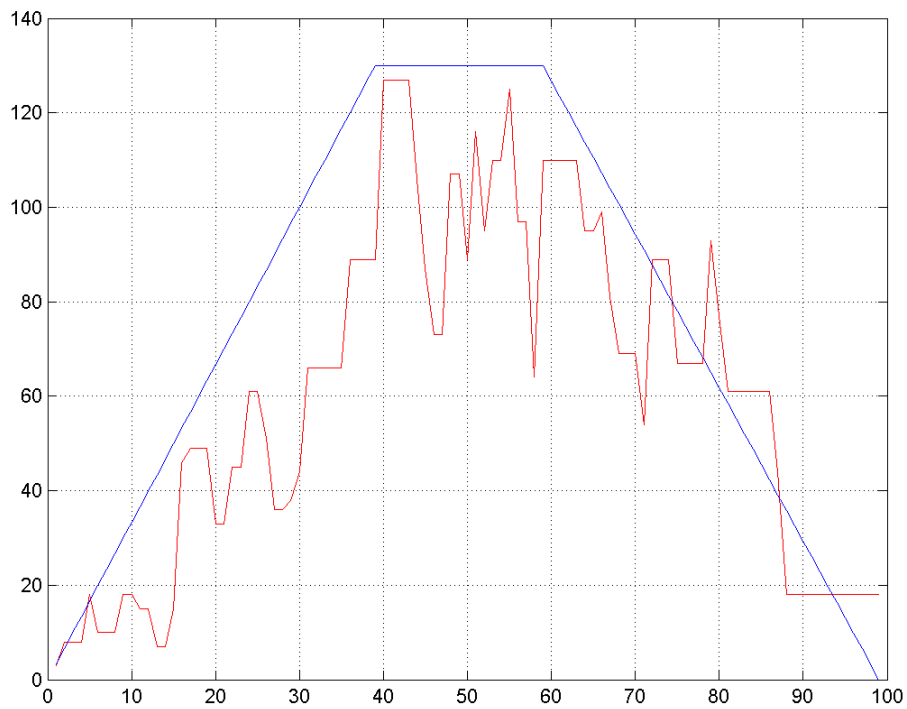
- **2^η Δοκιμή**

(N = 130, max time= 100, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.90 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 100, percentage = 0.4, iterations= 100, trials= 40)

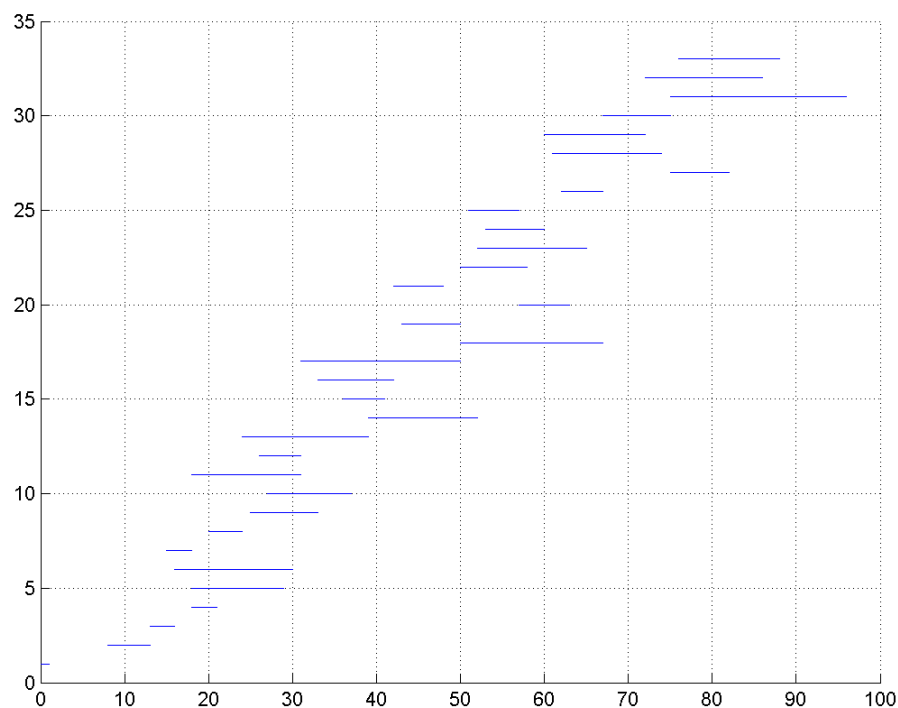


Σχήμα 5.91 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 100, percentage = 0.4, iterations= 100, trials= 40)

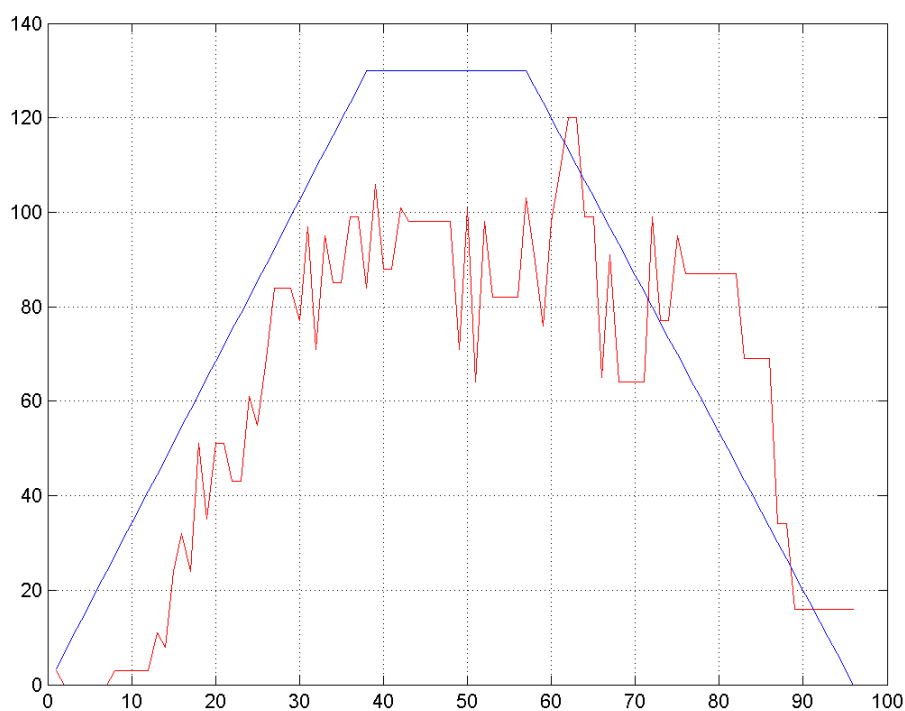
- **26^η Δοκιμή**

(N = 130, max time= 100, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.92 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 100, percentage = 0.4, iterations= 100, trials= 40)

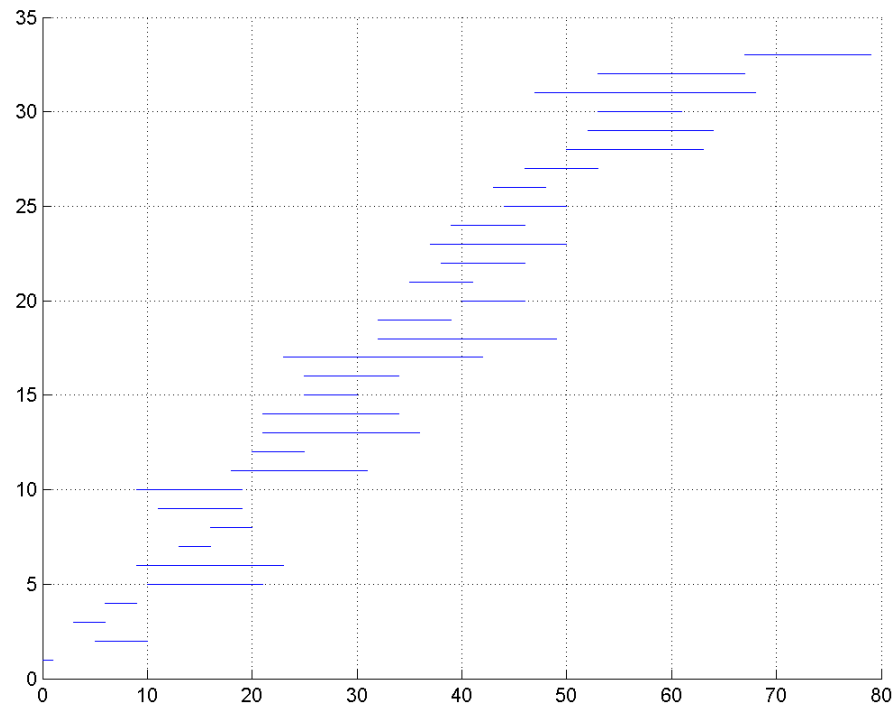


Σχήμα 5.93 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 100, percentage = 0.4, iterations= 100, trials= 40)

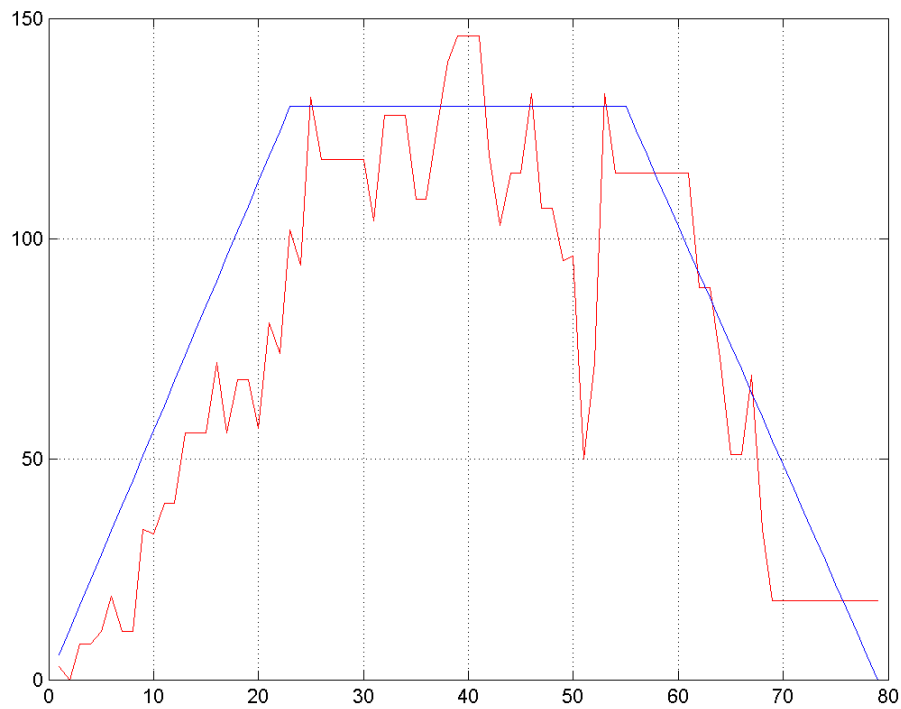
- **9^η Δοκιμή**

(N = 130, max time= 105, percentage = 0.3, iterations= 100, trials= 40)



Σχήμα 5.94 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 105, percentage = 0.3, iterations= 100, trials= 40)

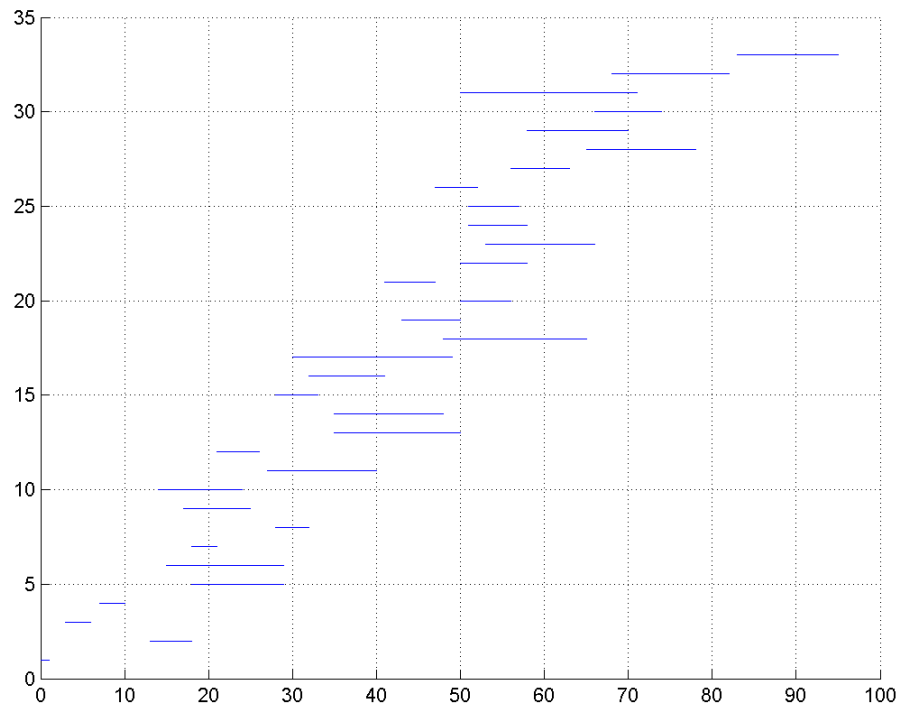


Σχήμα 5.95 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 105, percentage = 0.3, iterations= 100, trials= 40)

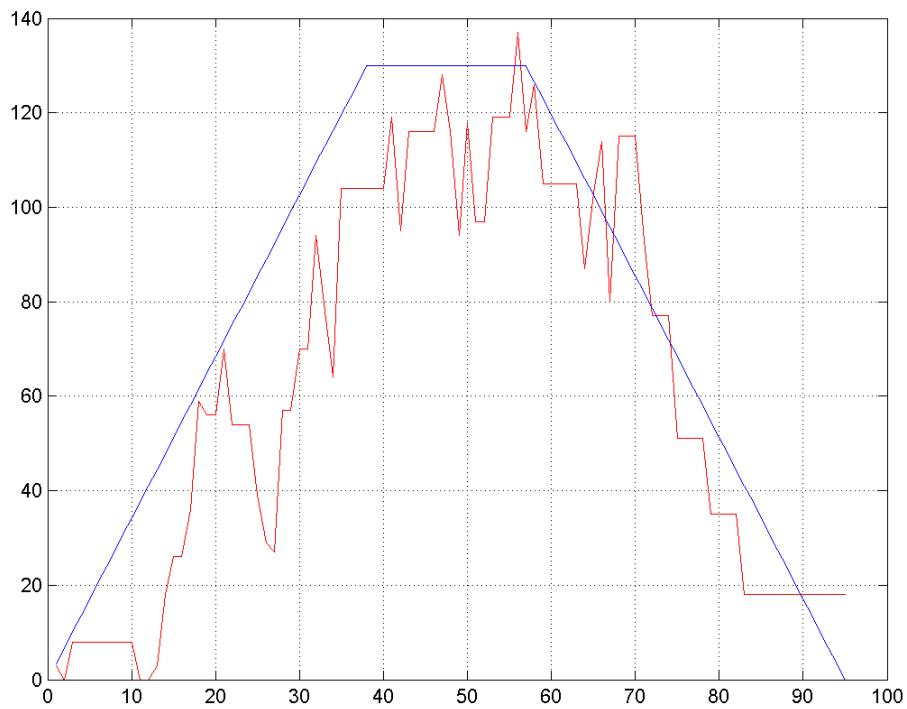
• **5^η Δοκιμή**

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.96 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

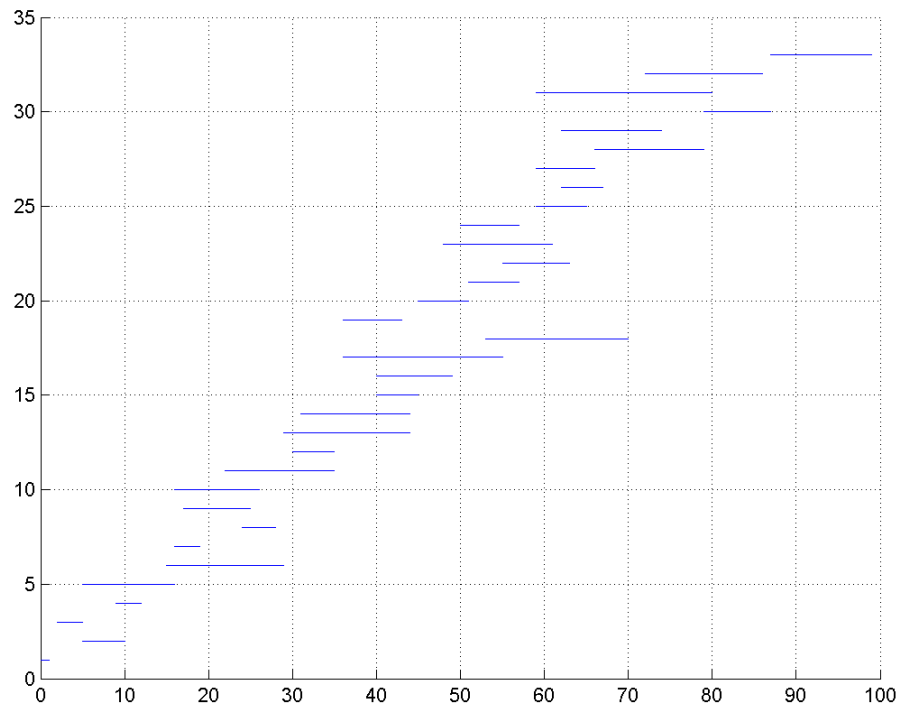


Σχήμα 5.97 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

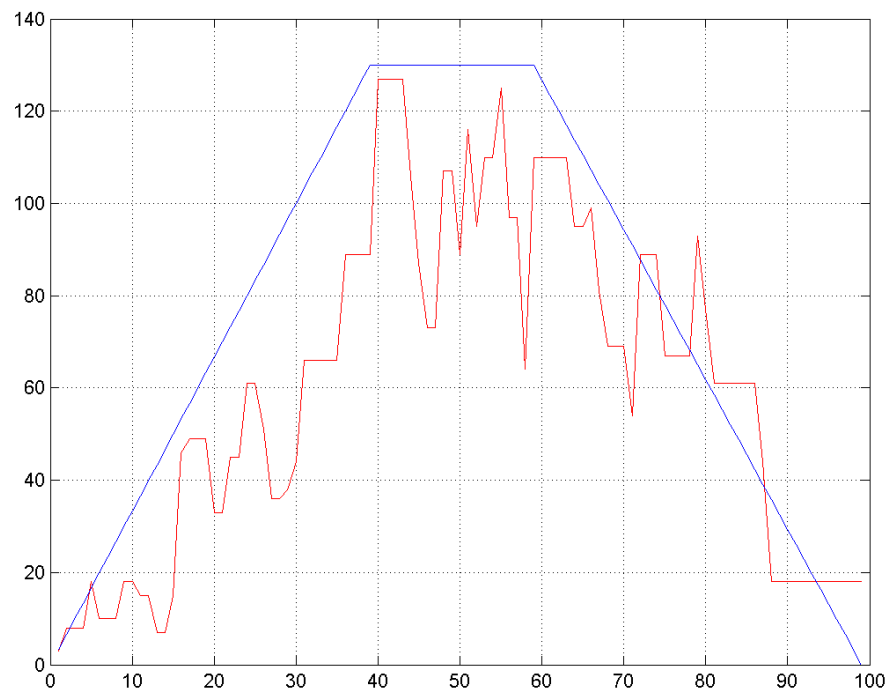
• **7^η Δοκιμή**

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.99 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

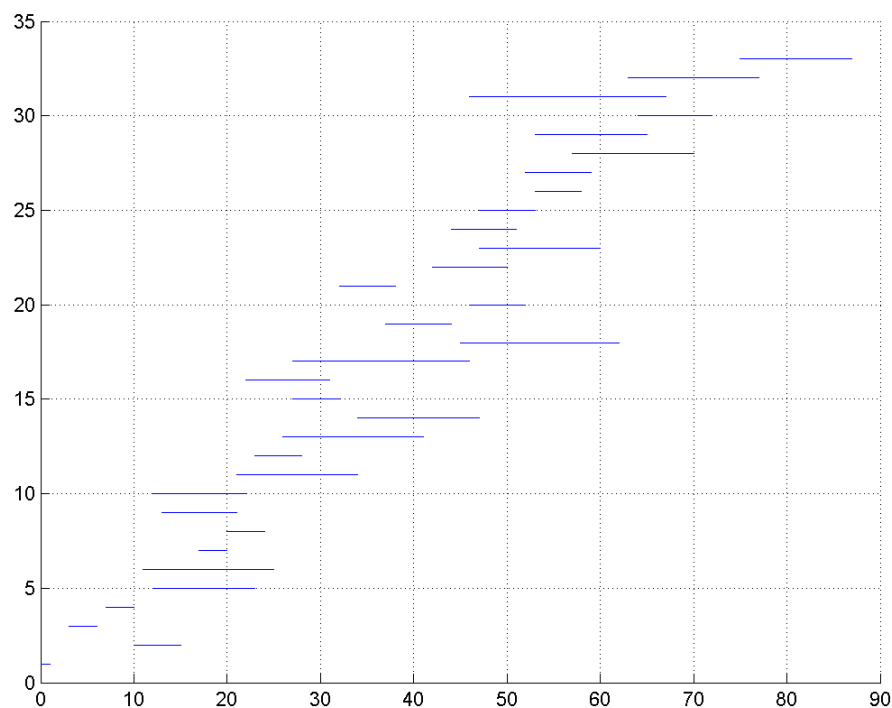


Σχήμα 5.100 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

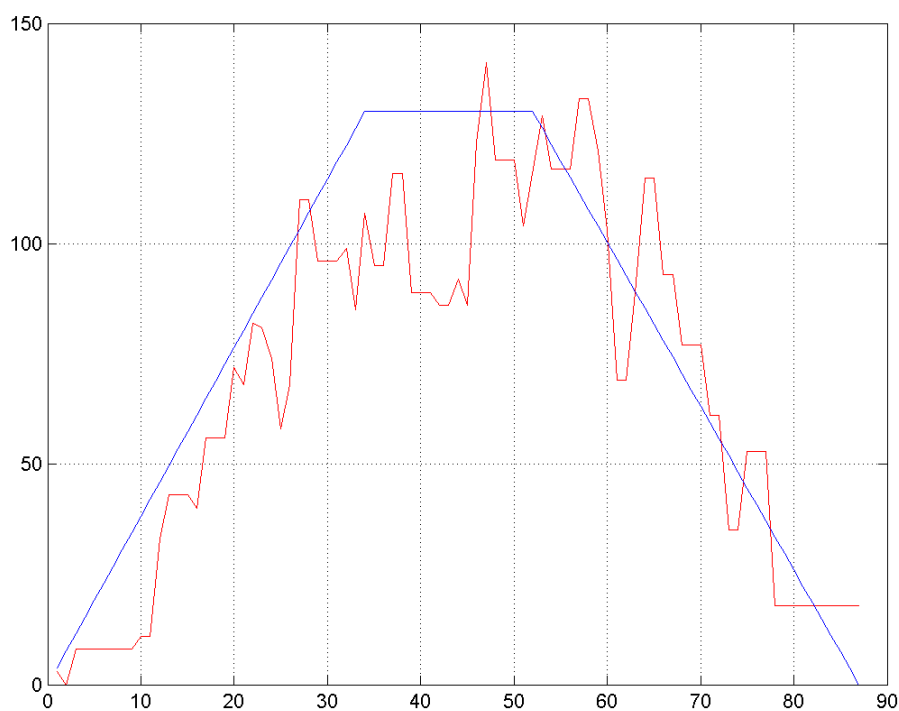
- **26^η Δοκιμή**

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.101 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

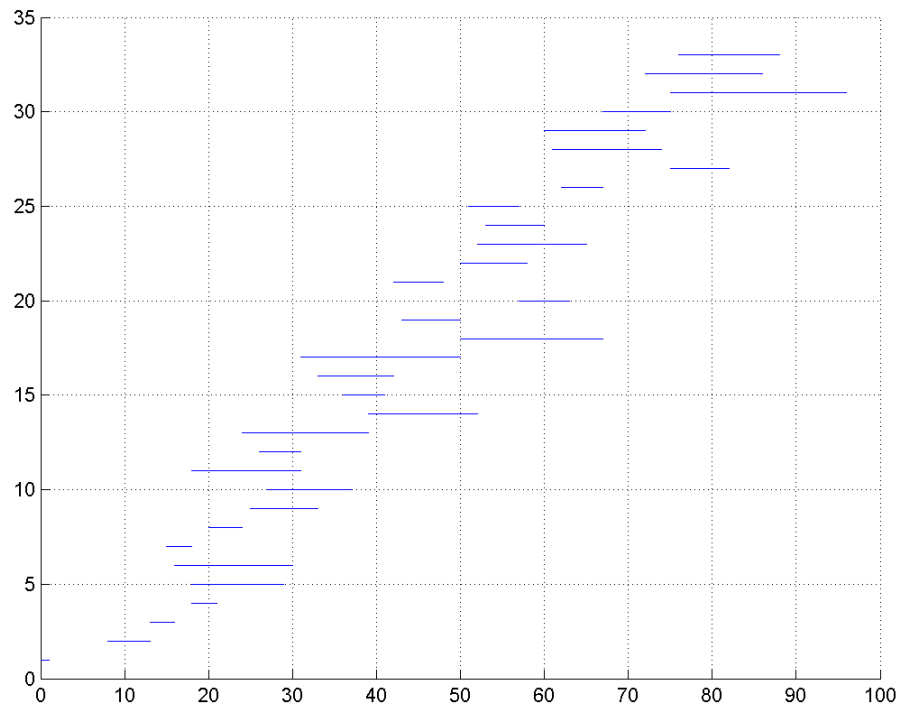


Σχήμα 5.102 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

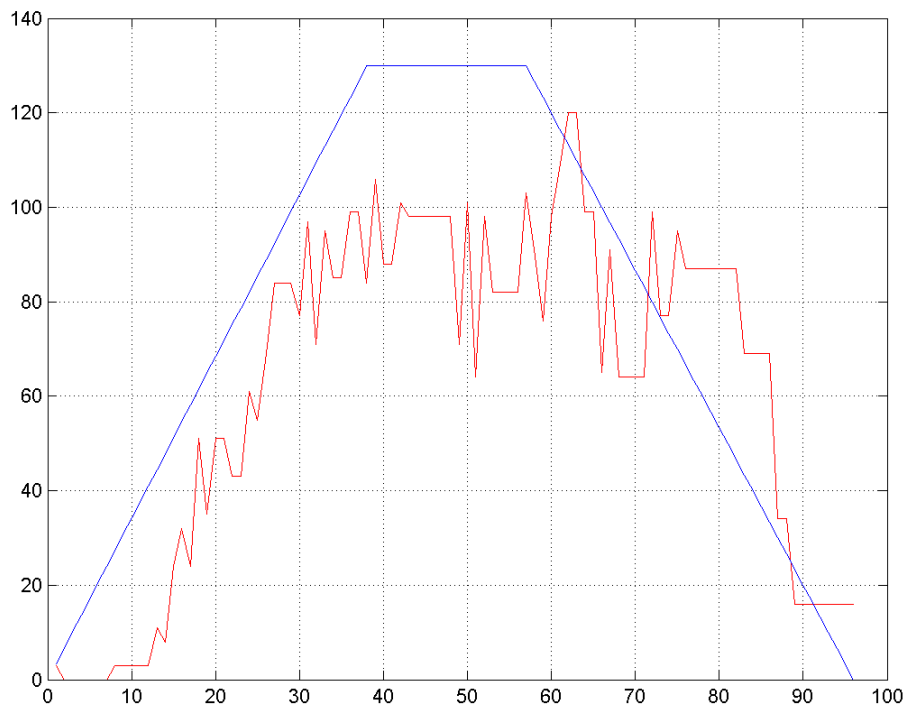
• **32^η Δοκιμή**

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.103 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

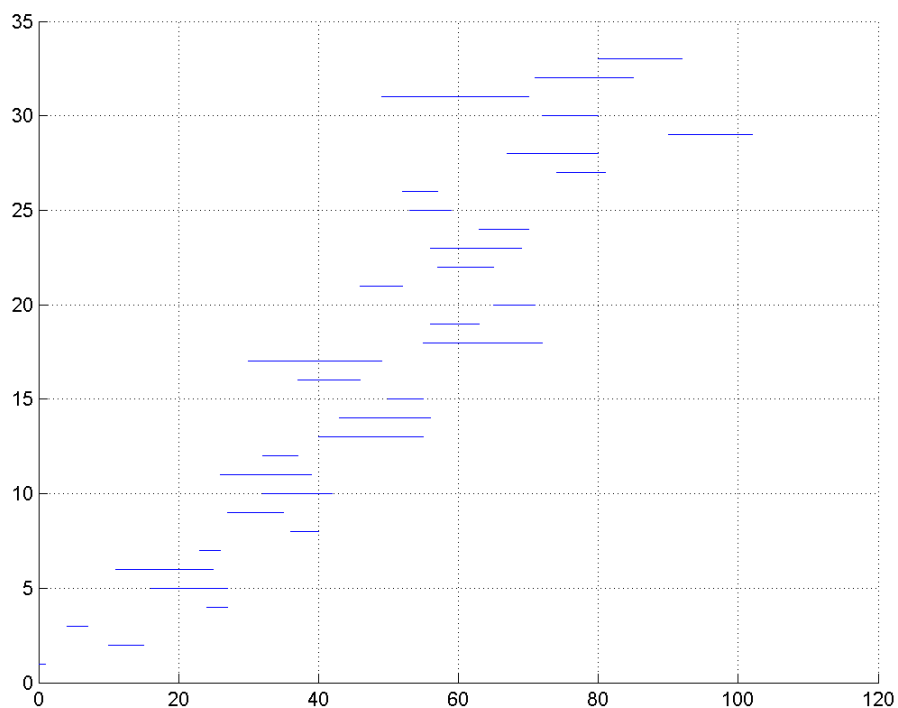


Σχήμα 5.104 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

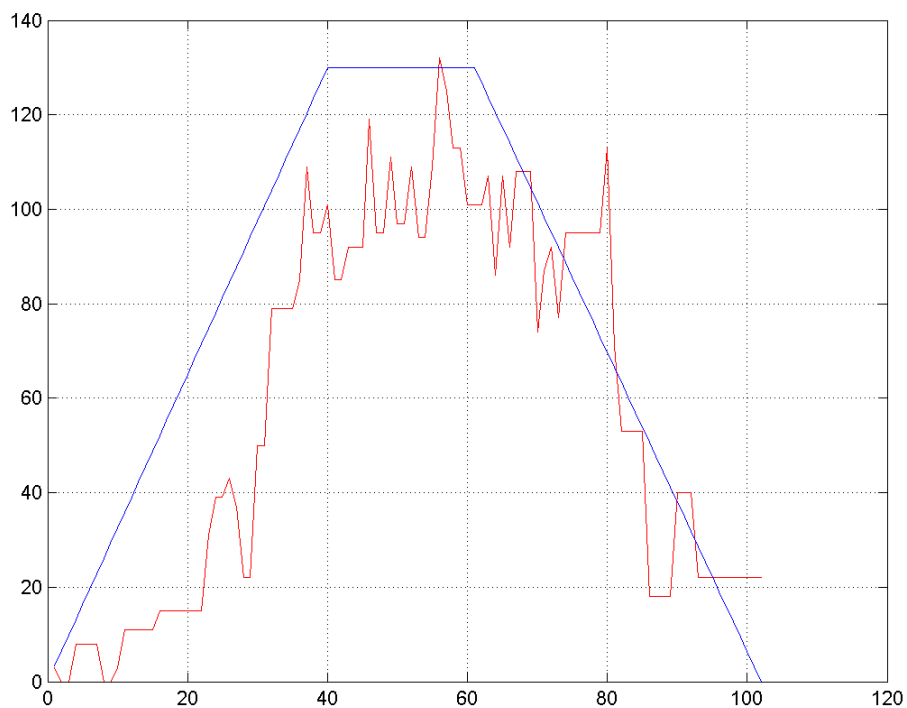
• **34^η Δοκιμή**

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)



Σχήμα 5.105 Χρονοδιάγραμμα Εργασιών

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)



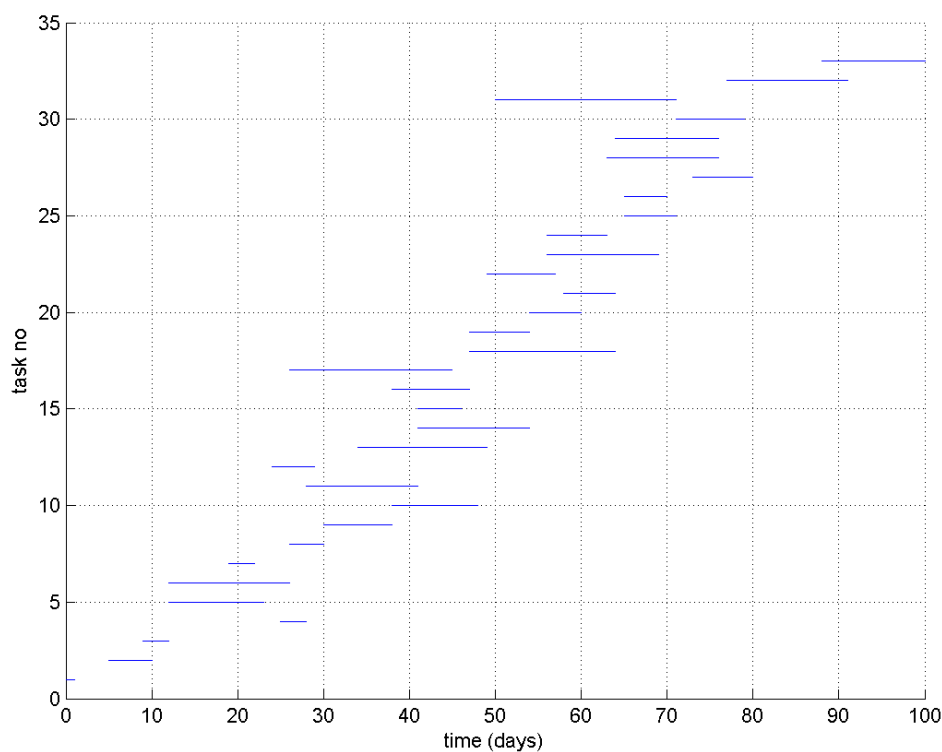
Σχήμα 5.106 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

(N = 130, max time= 105, percentage = 0.4, iterations= 100, trials= 40)

Έχοντας λάβει αρκετές και ικανοποιητικές λύσεις για τα διάφορα δεδομένα που εισάγαμε στο πρόγραμμα προχωρούμε πραγματοποιώντας δοκιμές με εξαιρετικά μεγάλο αριθμό επαναλήψεων iterations για να δούμε πως αυτό επηρεάζει την ποιότητα των λύσεων. Οι λύσεις που θα λάβουμε εξ' αρχής, φαίνεται ότι είναι πολύ καλές τόσο στο συνολικό χρόνο του έργου όσο και στη γραφική παράσταση εργατών-διάρκειας. Ωστόσο, θα μπορούσαμε να πούμε ότι το πρόγραμμα με αυτές τις λύσεις μας ανταμείβει για την υπομονή μας, καθώς είναι χαρακτηριστικά μεγαλύτερος ο χρόνος που χρειάζεται για να τελειώσει μια δοκιμή. Ας δούμε όμως μερικές ενδεικτικές λύσεις.

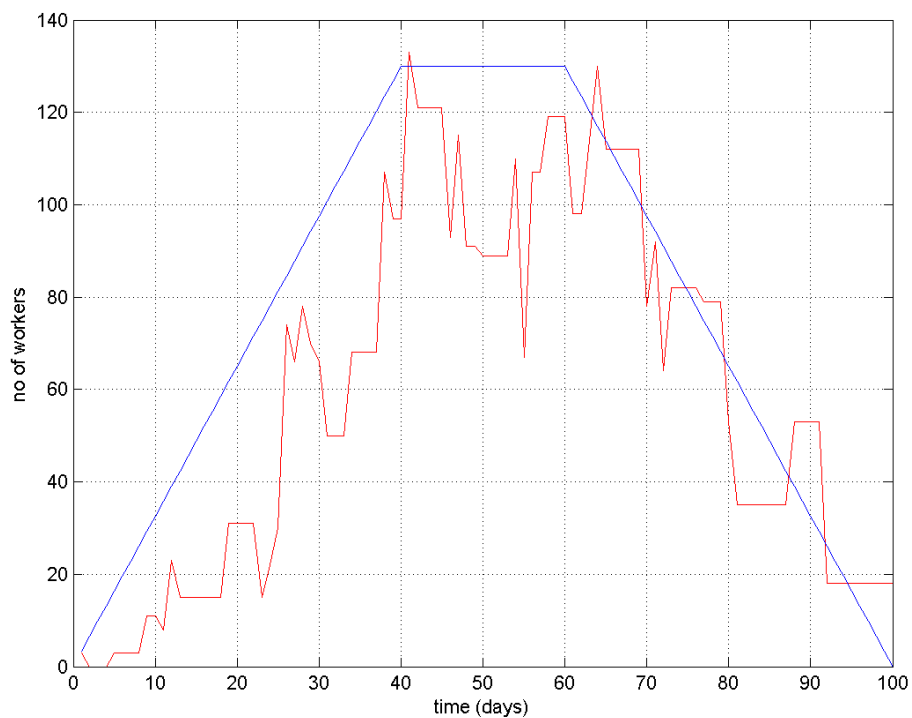
- **3^η Δοκιμή**

(N= 130, max time= 105, percentage= 0.4, iterations= 10.000, trials= 20)



Σχήμα 5.107 Χρονοδιάγραμμα Εργασιών

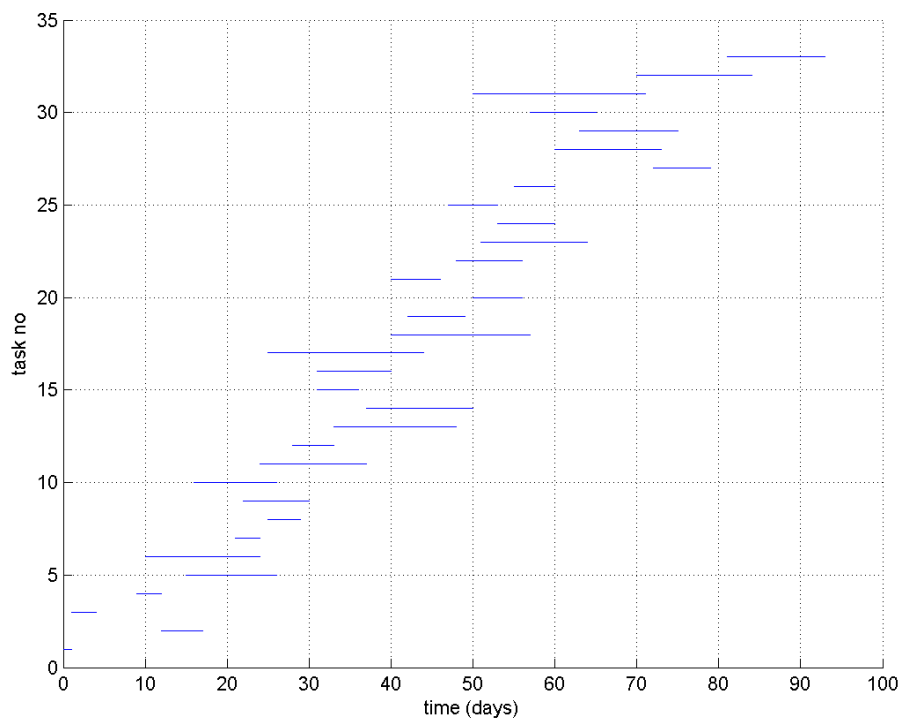
(N= 130, max time= 105, percentage= 0.4, iterations= 10.000, trials= 20)



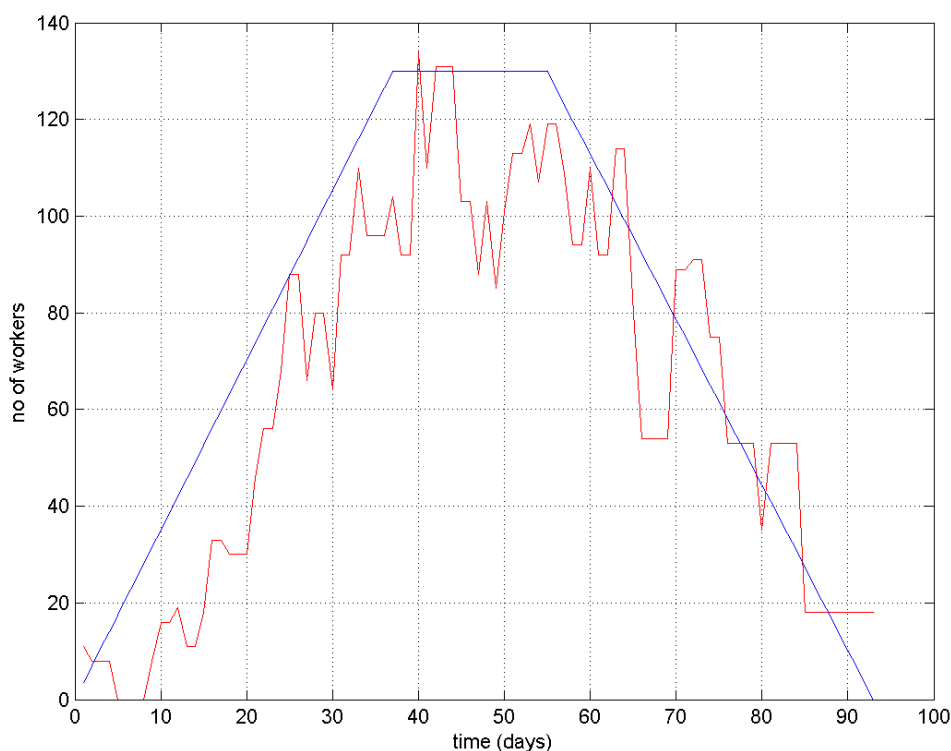
Σχήμα 5.108 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
(N= 130, max time= 105, percentage= 0.4, iterations= 10.000, trials= 20)

• **5^η Δοκιμή**

(N= 130, max time= 105, percentage= 0.4, iterations= 10.000, trials= 20)



Σχήμα 5.109 Χρονοδιάγραμμα Εργασιών
(N= 130, max time= 105, percentage= 0.4, iterations= 10.000, trials= 20)



Σχήμα 5.110 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
(N= 130, max time= 105, percentage= 0.4, iterations= 10.000, trials= 20)

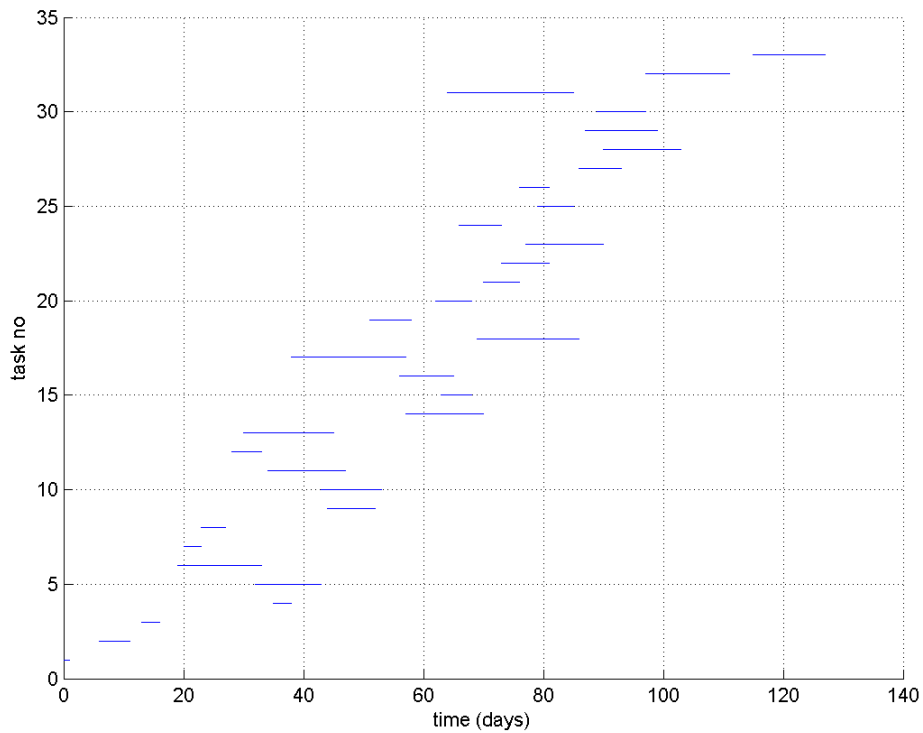
Πραγματοποιήσαμε κάποιες δοκιμές με αριθμό iterations 500, αριθμό trials 30, τον μέγιστο αριθμό των εργατών στους 100 και δίνοντας περιθώριο στο πρόγραμμα να φτάσει μέχρι και στις 200 ημέρες το χρόνο τέλους του έργου. Στο percentage διατηρήσαμε το 0,3 καθώς αυτή είναι η τιμή που συνήθως μας οδηγεί στις καλύτερες λύσεις. Ας δούμε τα χαρακτηριστικά των δοκιμών αυτών στο συγκεντρωτικό πίνακα των αποτελεσμάτων που ακολουθεί και τα γραφήματα των καλύτερων λύσεων.

Χαρακτηριστικά Προτεινόμενων Λύσεων

Αριθμός Δοκιμής	Max Time	No Trial που κατέληξε ο κώδικας	Τιμή Κριτηρίου Βελτιστοποίησης	Παρατηρήσεις Στη Λύση
1	96	13	1,7098	Καλή
2	120	7	2,5017	Μέτρια
3	94	25	2,4949	Μη αποδεκτή
4	89	13	1,7494	Μέτρια
5	99	17	1,7605	Μέτρια
6	126	27	2,5485	Αρκετά Καλή
7	104	17	2,2851	Μέτρια
8	118	11	2,3224	Καλή
9	94	6	1,5991	Μέτρια
10	103	17	1,9090	Μέτρια

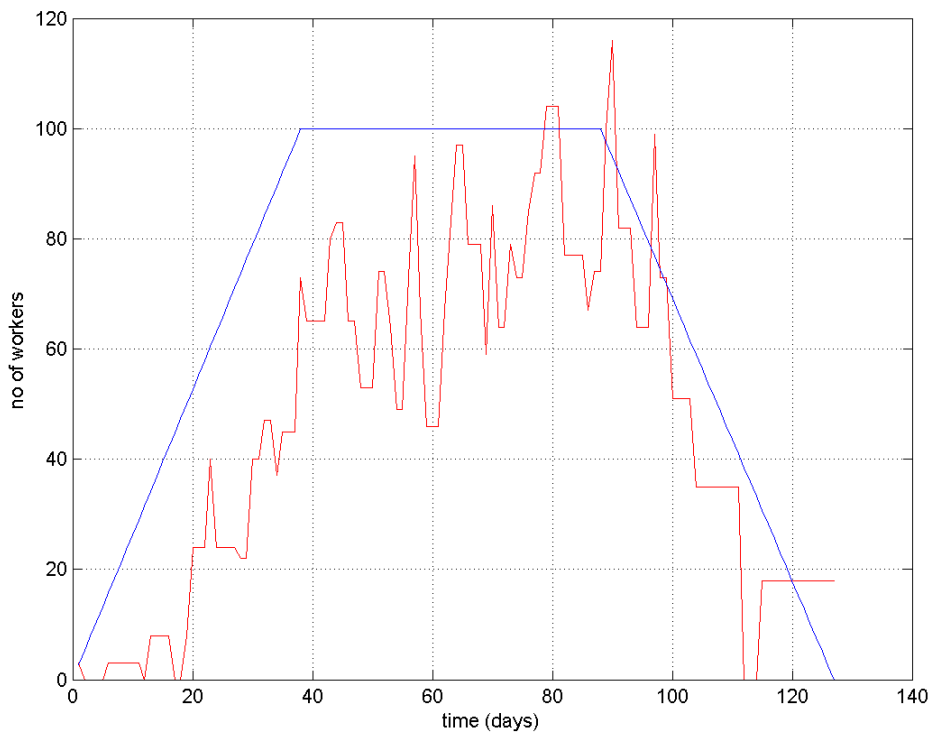
- **6^η Δοκιμή**

(N= 100, max time= 200, percentage= 0.3, iterations= 500, trials= 30)



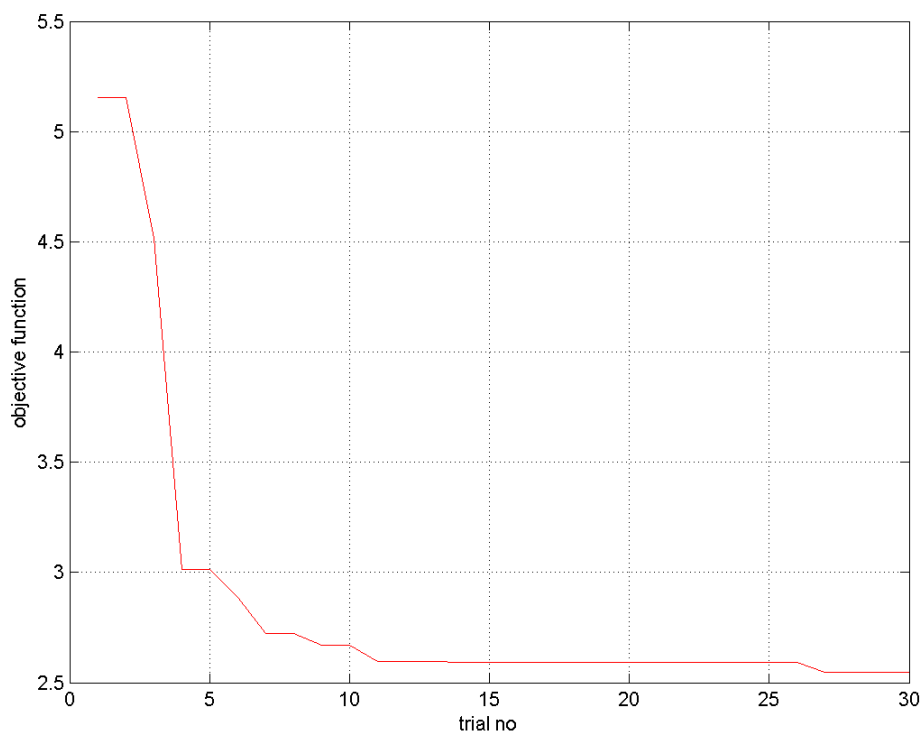
Σχήμα 5.111 Χρονοδιάγραμμα Εργασιών

(N= 100, max time= 200, percentage= 0.3, iterations= 500, trials= 30)



Σχήμα 5.112 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου

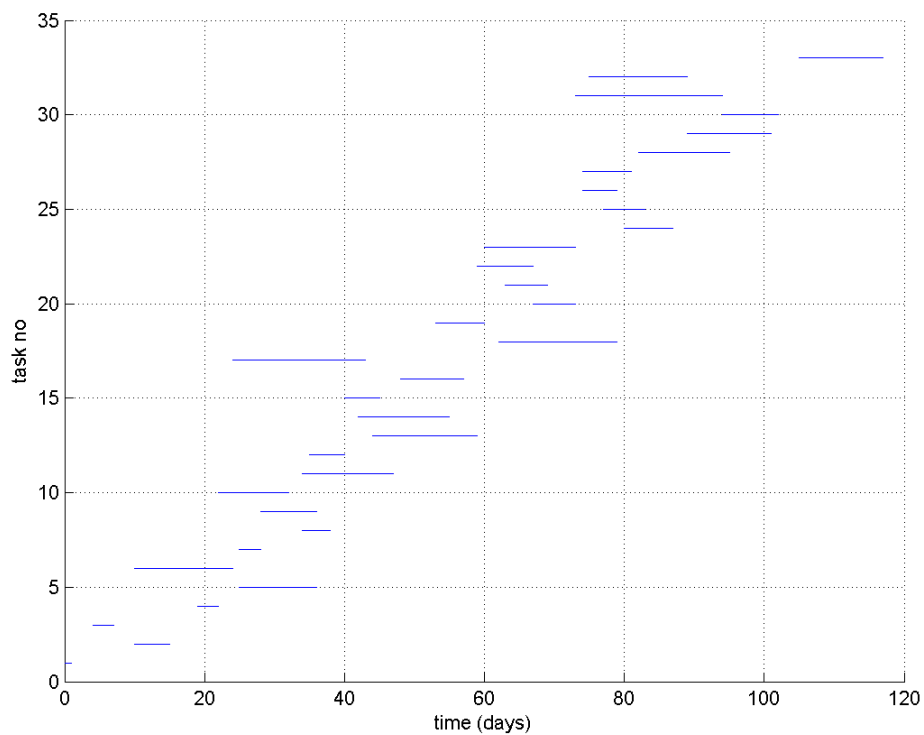
(N= 100, max time= 200, percentage= 0.3, iterations= 500, trials= 30)



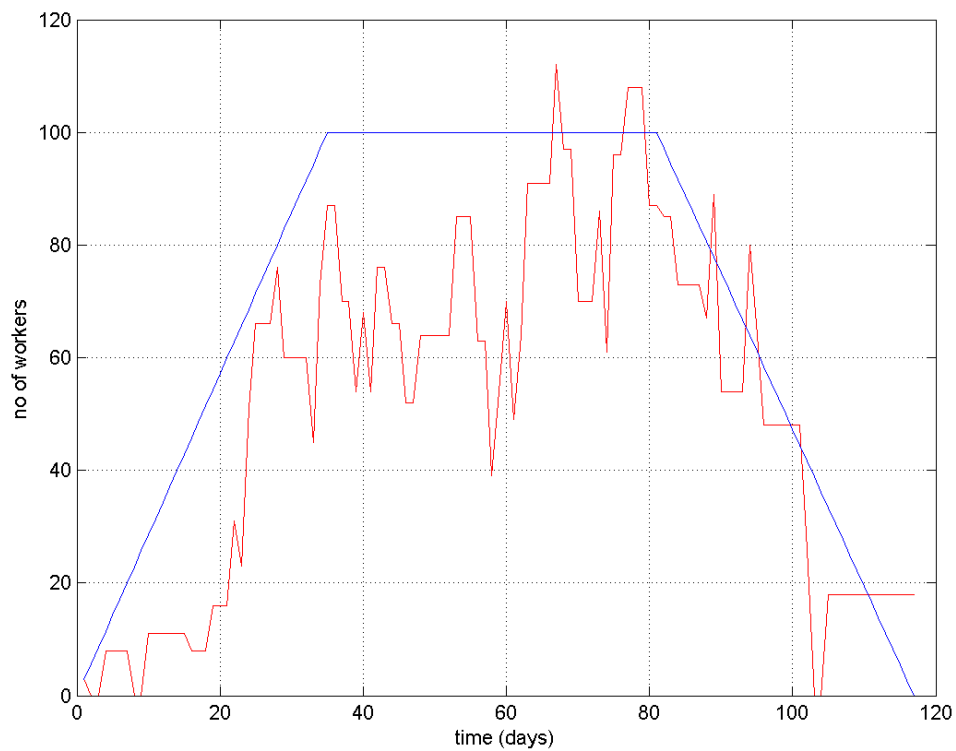
Σχήμα 5.113 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
(N= 100, max time= 200, percentage= 0.3, iterations= 500, trials= 30)

• **8^η Δοκιμή**

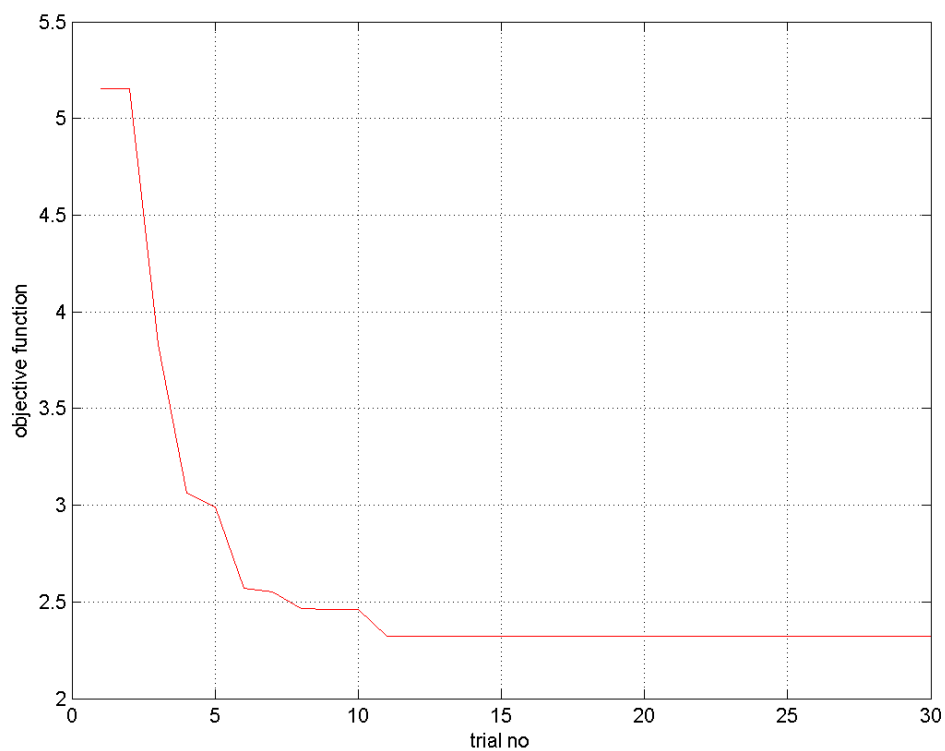
(N= 100, max time= 200, percentage= 0.3, iterations= 500, trials= 30)



Σχήμα 5.114 Χρονοδιάγραμμα Εργασιών
(N= 100, max time= 200, percentage= 0.3, iterations= 500, trials= 30)



Σχήμα 5.115 Γράφημα Εργατικού Δυναμικού – Διάρκειας Έργου
($N = 100$, max time= 200, percentage= 0.3, iterations= 500, trials= 30)



Σχήμα 5.116 Γράφημα Τιμής Αντικειμενικής Συνάρτησης – Αριθμού Trial
($N = 100$, max time= 200, percentage= 0.3, iterations= 500, trials= 30)

ΚΕΦΑΛΑΙΟ 6

ΣΥΜΠΕΡΑΣΜΑΤΑ

6.1 Συμπεράσματα

Μετά από πληθώρα δοκιμών και επαναλήψεων του προγράμματος που δημιουργήσαμε καταλήγουμε στο συμπέρασμα ότι λειτουργεί ικανοποιητικά προσφέροντας πολλαπλές, διαφορετικές και εφικτές εναλλακτικές λύσεις σε ένα πρόβλημα ιδιαίτερης δυσκολίας. Και η δυσκολία έγκειται στο γεγονός ότι το έργο αυτό που επιλέχθηκε είχε βεβαρημένο αριθμό εργατών σε πολλές από τις εργασίες οι οποίες χρονικά είχαν εξαρτήσεις μεταξύ τους. Πραγματοποιήσαμε πολλές αναγνωριστικές δοκιμές για να εξετάσουμε την συμπεριφορά των διάφορων παραμέτρων που υπεισέρχονται στο πρόγραμμα και τελικά καταλήξαμε σε κάποιες τιμές τους που αποδίδουν καλύτερες λύσεις πάντα βέβαια για το συγκεκριμένο έργο της δοκιμής. Αναλυτικότερα, σε τιμές του percentage που κυμαίνονταν μεταξύ του 0,3 και 0,4 σε μέγιστο δυνατό χρόνο εκτέλεσης του έργου MMax Time από 80 μέχρι 100 ημέρες λάβαμε αρκετές λύσεις που περιόριζαν τον αριθμό των εργατών στους 120-130 και στην επιθυμητή βέβαια γραφικά κατανομή που υποδεικνυόταν από τον κώδικα. Δηλαδή, πετύχαμε αρχικά μείωση μέχρι και 20% στον μέγιστο αριθμό εργατών αποσυμπιέζοντας χρονικά το έργο από 30% ως και 50%. Χαρακτηριστικό ήταν βέβαια ότι σε όλες δοκιμές που πραγματοποιούσαμε για διαφορετικά δεδομένα οι καλύτερες και περισσότερες λύσεις προέκυπταν για τις συγκεκριμένες τιμές του percentage (0,3-0,4) γεγονός που μας έκανε να επικεντρωθούμε από ένα σημείο και μετά μόνο σε αυτές τις τιμές. Προσπαθήσαμε να ελαττώσουμε περισσότερο την τιμή του αριθμού των απασχολούμενων εργατών και επιτύχαμε μείωση μεγαλύτερη του 30% (ως και 38%) περιορίζοντας την στους 100 ακόμα και 90 εργάτες. Η αύξηση του χρόνου εκτέλεσης του έργου ξεκινούσε από 50% και έφτανε μέχρι και 100%. Σαν ακραία περίπτωση πειραματιστήκαμε να βρούμε λύσεις ακόμα και στην περίπτωση των 70 εργατών δηλαδή μείωση της τάξης του 50%. Μόνο που στη περίπτωση αυτή χρειάστηκε υπερδιπλασιασμός τουλάχιστον της χρονικής διάρκειας του έργου και στην πραγματικότητα είναι δύσκολο να προτιμηθεί σαν εναλλακτική δράση. Γενικότερα, παρατηρήσαμε ότι μια μείωση σε επί τοις εκατό ποσοστό στον αρχικό αριθμό των εργατών είχε σαν αντίκρισμα σχεδόν την διπλάσια ποσοστιαία αύξηση σε μέρες στην αρχική εκτίμηση για τη χρονική εκτέλεση του έργου. Παράλληλα, στις διάφορες δοκιμές υπήρξε πειραματισμός για τον αριθμό των επαναλήψεων των iterations και trials. Παρόλο που η διαδικασία έγινε εξαιρετικά χρονοβόρα, δίνοντας αρκετά υψηλές τιμές για το iterations (5.000, 10.000, 15.000) λάβαμε εξ αρχής καλές λύσεις και χρονικά αλλά και από σκοπιάς γραφικής παράστασης εργατικού δυναμικού συναρτήσεως διάρκειας έργου. Στον αριθμό των trials ικανοποιητικές λύσεις λαμβάνουμε στην πλειοψηφία των περιπτώσεων σε τιμές 40 ως και 50. Άλλωστε, η εξέταση των γραφημάτων της αντικειμενικής συνάρτησης σε συνάρτηση με τον αριθμό των trials μαρτυρεί ότι ελάχιστες φορές έχουμε μεταβολές στην τιμή της πρώτης μετά από τον αριθμό 50 trials που στην ουσία σημαίνει ότι το πρόγραμμα έχει καταλήξει σε μια λύση και γι αυτό το

λόγο η τιμή της objective function παραμένει αμετάβλητη. Κατά συνέπεια, τιμές στον αριθμό των trials άνω του 70 μόνο χρονοβόρες καταλήγουν να είναι. Επίσης, στις δοκιμές που αυξήσαμε κατά πολύ τις απαιτήσεις μας για μείωση του εργατικού δυναμικού για παράδειγμα στους 90 εργάτες με αρκετό περιθώριο χρονικής αποσυμπίεσης, λάβαμε σαν πρώτες λύσεις τέτοιες που μπορούσαν να ικανοποιήσουν τις απαιτήσεις μας σε προηγούμενες αναζητήσεις όπως για παράδειγμα στους 130 εργάτες. Αυτό συμβαίνει γιατί η αναζήτηση του προγράμματος για εναλλακτικές λύσεις γίνεται σταδιακά δηλαδή εξετάζει πως μπορεί να μειώσει τους εργάτες χρησιμοποιώντας σε κάθε επανάληψη κάποιες ημέρες περισσότερες μέχρι να φτάσει στον αριθμό MMax Time που έχει δηλωθεί από τον χρήστη σαν μέγιστος δυνατός χρόνος αποπεράτωσης του έργου.

6.2 Συμβουλές προς χρήστες

Προτείνονται στο χρήστη οι παρακάτω τιμές, οι οποίες προέκυψαν συμπερασματικά από τη διαδικασία διαχείρισης του προγράμματος μας και μπορούν να βοηθήσουν στην αποτελεσματικότερη χρήση του.

Προτεινόμενες τιμές Μεταβλητών

	Τυπική αναζήτηση	Εξεζητημένη Αναζήτηση
Number of Iterations	700 -1.500	10.000 - 20.000
Number of trials	40 – 70	100 -200
percentage	0,3 -0,4	0,25 -0,45

Η τυπική αναζήτηση μπορεί να ολοκληρωθεί μέσα σε λίγα λεπτά ενώ η εξεζητημένη μπορεί να διαρκέσει ακόμα και αρκετές ώρες.

Επίσης, ο χρήστης μπορεί να επιτύχει γρηγορότερα λύσεις αν ,εκτός από τις προτεινόμενες τιμές μεταβλητών, στα δεδομένα του προβλήματος του εισάγει :

A) τιμή για μέγιστο αριθμό εργατών μειωμένη κατά 20 - 30 % από τις πραγματικές του απαιτήσεις και παράλληλα

B) τιμή για το μέγιστο χρόνο εκτέλεσης του έργου αυξημένη κατά 50 – 70% από την επιθυμητή.

6.3 Παρεμβάσεις- Βελτιώσεις

Τα πρώτα μηνύματα λειτουργίας του κώδικα είναι θετικά ωστόσο κάποιες παρατηρήσεις μας μπορούν να υπάρξουν απαρχή βελτιώσεων για το μέλλον. Πιο συγκεκριμένα, παρατηρήσαμε σε γραφήματα Εργατικού Δυναμικού – Διάρκειας Έργου σε αρκετές από τις λύσεις, ακόμα και σε πολύ καλές από αυτές, να υπάρχουν έντονες κορυφές είτε ανοδικές είτε καθοδικές τα λεγόμενα "peak". Αυτό πρακτικά σημαίνει έντονες διακυμάνσεις και θα ήταν σίγουρα ενδιαφέρον να μπορούσε να ελεγχθεί το μέγεθος των κορυφών

ή να περιοριστεί το ύψος τους σε επιθυμητές τιμές παράλληλα με τις διεργασίες του κώδικα αναζήτησης καλύτερης λύσης.

Επίσης, μια άλλη περίπτωση που μπορεί να συναντήσει ένας χρήστης είναι οι προτεινόμενες από το πρόγραμμα εναλλακτικές για την εκτέλεση του έργου να τον ικανοποιούν σε μεγάλο βαθμό, εκτός από τη χρονική τοποθέτηση συγκεκριμένων εργασιών πάνω στο χρονοδιάγραμμα. Σαν πρόταση βελτίωσης λοιπόν θα μπορούσε να αποτελέσει η ύπαρξη της δυνατότητας χειρονακτικής παρέμβασης από το χρήστη πάνω στη λύση που του προσφέρει το πρόγραμμα. Ασφαλώς, η τροποποίηση της λύσης θα πρέπει να επιδρά αυτόματα και στα υπόλοιπα γραφήματα έτσι ώστε ο χρήστης να μπορεί να συμπεράνει αν ωφέλησε η παρέμβασή του.

Ακόμα μια πρόταση για παρέμβαση του χρήστη στο πρόγραμμα είναι να υπάρχει η δυνατότητα επιλογής μιας ή περισσότερων εργασιών, εκ των προτέρων, ως προτιμητέες για μεταθέσεις στο χρονοδιάγραμμα των εργασιών.

Το πρόγραμμα, αν υπολογίσει κάποιος τα μεγέθη και τον αριθμό των παραμέτρων που υπεισέρχονται στην προσπάθεια διαχείρισης ενός έργου, επιδέχεται αρκετές βελτιώσεις και επεκτάσεις έτσι ώστε να είναι πιο ολοκληρωμένο και να μπορεί να συνυπολογίζει και άλλους παράγοντες πριν να προτείνει μια λύση. Με αυτόν τον τρόπο θα μπορεί να υποστηρίξει ισχυρότερα τη λήψη μιας απόφασης που είναι και το ζητούμενο.

Βιβλιογραφία

- Οργάνωση Παραγωγής και Προγραμματισμός Έργων, Σημειώσεις Μαθήματος Πολυτεχνείου Κρήτης, **Β. Κουϊκόγλου, Ε. Γρηγορούδης, Η. Κοσματοπούλος**, 2003
- Διοίκηση – Διαχείριση Έργου (Project Management), **Δημητριάδης Α.**, Εκδόσεις Νέων Τεχνολογιών, 1999
- Διοίκηση και Διαχείριση Έργων μέσω του Ειδικού Λογαριασμού Έρευνας, **Γ. Ντέτσικας**, Πανεπιστήμιο Ιωαννίνων, 2002
- Τεχνικές Μείωσης του Κόστους και Αύξησης της Απόδοσης, **Δημητριάδης Α.**, Εκδόσεις Νέων Τεχνολογιών, 1996