



ΠΟΛΥΤΕΧΝΕΙΟ  
ΚΡΗΤΗΣ /  
TECHNICAL  
UNIVERSITY  
OF CRETE

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

---

Οικονομική Κατανομή Ηλεκτρικών Φορτίων σε  
Θερμοηλεκτρικούς Σταθμούς:  
Μοντελοποίηση και Βελτιστοποίηση



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

ΦΟΙΤΗΤΗΣ: ΑΝΑΣΤΑΣΙΟΣ ΠΟΛΥΧΡΟΝΟΠΟΥΛΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΓΥΑΚΗΣ ΔΗΜΗΤΡΗΣ

Χανιά, 2024



*Στα άτομα τα οποία στάθηκαν δίπλα μου  
στα εύκολα και στα δύσκολα  
και τα οποία δεν θα ξεχάσω ποτέ!*

# Ευχαριστίες

---

Ξεκινώντας, θα ήθελα να ευχαριστήσω πρωτίστως, τον επιβλέπων καθηγητή μου κ. Ιψάκη, για την εμπιστοσύνη την οποία μου έδειξε όλον αυτόν τον καιρό καθώς και για την καθοδήγηση του σε ό,τι χρειάστηκα κατά την διάρκεια της εκπόνησης της εργασίας αυτής αλλά και κατά την διάρκεια των μαθημάτων του. Ταυτόχρονα, δεν θα μπορούσα να μην ευχαριστήσω τον πατέρα μου Χρήστο, την μητέρα μου Αλεξάνδρα, καθώς και τα τρία μου αδέρφια, Νίκο, Γιώργο και Πανωραία, οι οποίοι αποτέλεσαν στήριγμα μου τα πέντε αυτά έτη όπου βρέθηκα στο Πολυτεχνείο Κρήτης και όπου χωρίς αυτούς δεν θα είχα καταφέρει τίποτα έως τώρα. Τέλος, ένα τεράστιο ευχαριστώ στα άτομα τα οποία με ανέχονται καθημερινά, τους φίλους και τις φίλες μου, όπου δίχως αυτούς δεν θα μπορούσα να περάσω καλύτερα φοιτητικά χρόνια.

## Περίληψη

---

Στην παρούσα εργασία επιλύεται ένα μέρος του προβλήματος της οικονομικής κατανομής ηλεκτρικού φορτίου, σε θερμικούς σταθμούς παραγωγής ηλεκτρικής ενέργειας. Το πρόβλημα αυτό εμφανίζεται, όταν δύο ή και περισσότεροι θερμικοί σταθμοί συνεργάζονται, ώστε να καλύψουν την ηλεκτρική ζήτηση κάποιας περιοχής, καθώς το ζήτημα το οποίο προκύπτει είναι πόση ενέργεια πρέπει να παραχθεί από τον κάθε σταθμό παραγωγής (λόγω του ότι κάθε σταθμός χαρακτηρίζεται από διαφορετικό δείκτη σταθμισμένης ενέργειας, €/kWh). Στόχος λοιπόν, της εργασίας αυτής είναι η επίλυση του προβλήματος κατανομής ηλεκτρικού φορτίου, το οποίο είναι ένα πρόβλημα γραμμικού τύπου, με τρόπο τέτοιο ώστε, η ηλεκτρική ζήτηση να καλύπτεται με το ελάχιστο δυνατό συνολικό κόστος λειτουργίας των σταθμών. Αρχικά παρουσιάζονται, ο τρόπος λειτουργίας των συστημάτων ηλεκτρικής ενέργειας καθώς και έννοιες που αφορούν την μελέτη και πρόβλεψη του ηλεκτρικού φορτίου. Γίνεται έπειτα, αναφορά στα θερμικά συστήματα παραγωγής ενέργειας αλλά και στις καμπύλες εισόδου-εξόδου, προκειμένου να γίνει ανάλυση της οικονομικής λειτουργίας των μονάδων. Στην συνέχεια, επιλύεται το πρόβλημα της οικονομικής κατανομής φορτίου στην απλή και σύνθετη εκδοχή του με την χρήση περιορισμών, όπως των λειτουργικών ορίων των μονάδων και των απωλειών ενέργειας λόγω μεταφοράς. Τέλος, επιλύεται και το πρόβλημα ένταξης ηλεκτρικών μονάδων παραγωγής, ως μέρος της βελτιστοποίησης του προβλήματος της οικονομικής κατανομής ηλεκτρικού φορτίου. Για την επίλυση των παραπάνω προβλημάτων, χρησιμοποιήθηκε το λογισμικό της MatLAB, στο οποίο αναπτύχθηκαν τα αντίστοιχα αρχεία. Τα αποτελέσματα τα οποία πήραμε από το λογισμικό της MatLAB ήταν τα αναμενόμενα με μικρές διαφοροποιήσεις από την κλασική επίλυση.

**Λέξεις κλειδιά:** Διαχείριση ενέργειας, Σύστημα Ηλεκτρικής Ενέργειας (ΣΗΕ), Σύστημα Ελέγχου Ενέργειας (ΣΕΕ), Θερμικοί σταθμοί, Ενέργεια, Βελτιστοποίηση

# Abstract

---

In this thesis, a part of the problem of economic distribution of electrical load for the thermal power plants, is being solved. This problem appears, when two or more thermal plants are cooperated in order to cover the electrical demand of an area, since the issue that arises is how much energy should be produced by each plant. The aim of this thesis is to solve the economic distribution problem in such a way, that the electrical demand is covered with the minimum possible total cost of operating the plant. Initially, the operation of electricity systems is presented, along with concepts related to the study and prediction of electrical load. After that, there is a reference about thermal power plants, as well as to the input – output curves, in order to analyze the economic operation of plants. Additionally, the problem of economic load distribution is solved in both its simple and complex versions by considering constraints such as the functional limits of the units and energy losses due to transmission. Finally, the problem of power plant integration is solved, as part of the optimization of economic load distribution problem. For all these problems, MatLAB software was utilized, in which the corresponding files were developed.

**Key words:** *Energy Management, Electricity System, Electricity Control System, Thermal Power Plants, Energy, Optimization*

# Πρόλογος

---

Στην καθημερινότητα μας, συναντάμε όλο και περισσότερο την βελτιστοποίηση. Αν αναλύσουμε το πως σκεφτόμαστε, από τον δρόμο τον οποίο θα ακολουθήσουμε για να πάμε από το ένα μέρος στο άλλο, από τον προγραμματισμό των καλοκαιρινών διακοπών, έως και το πλάνο δόσεων το οποίο θα βάλουμε για να αποκτήσουμε κάποιο υλικό αγαθό, ο άνθρωπος χρησιμοποιεί την βελτιστοποίηση. Έτσι γίνεται και στα συστήματα παραγωγής ενέργειας. Δεν αρκεί να παράξουμε μια ποσότητα ενέργειας η οποία μας ζητείται, αλλά χρειάζεται να την παράξουμε, με τον πιο οικονομικό και αποτελεσματικό τρόπο. Αυτού του είδους η βελτιστοποίηση, παρουσιάζεται στην εργασία αυτή με στόχο την οικονομική κατανομή ηλεκτρικών φορτίων (συμβατικών και ανανεώσιμων πηγών ενέργειας).

Το βιβλίο πάνω στο οποίο βασίζεται η εργασία αυτή, είναι του κ. Αναστάσιου Μπακιρτζή, με τίτλο Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας, ένα βιβλίο, αν και αρκετά προγενέστερο για τα δεδομένα του 2023, περιγράφει απόλυτα τον τρόπο με τον οποίο κατανέμεται η ηλεκτρική ενέργεια σε θερμοηλεκτρικούς σταθμούς. Για τις ανάγκες της εργασίας, χρησιμοποιήθηκε το λογισμικό MATLAB, στο οποίο και δημιουργήθηκαν κάποια προγράμματα προκειμένου να επιλυθούν τα προβλήματα τα οποία παρουσιάζονται στην συνέχεια.

Αρχικά, χρειάζεται να οριστεί η έννοια του θερμοηλεκτρικού σταθμού. Ένας θερμοηλεκτρικός σταθμός, είναι μια εγκατάσταση η οποία παράγει ηλεκτρική ενέργεια με την χρήση θερμικής ενέργειας. Η ενέργεια αυτή παράγεται συνήθως, από την καύση ορυκτών πόρων όπως ο λιγνίτης, ο άνθρακας, το πετρέλαιο, το αέριο αλλά και από πυρηνική αντίδραση, την οποία βέβαια δεν συναντάμε στην Ελλάδα.

Ο αναγνώστης στην εργασία αυτή, θα συναντήσει τέσσερα κεφάλαια. Στο πρώτο κεφάλαιο, θα μελετηθεί ο τρόπος λειτουργίας των συστημάτων ηλεκτρικής ενέργειας, καθώς και η πρόβλεψη του φορτίου, ενώ στο δεύτερο, θα αναλυθούν τα θερμικά συστήματα παραγωγής ενέργειας καθώς και οι καμπύλες εισόδου – εξόδου των μονάδων. Επιπρόσθετα, στο τρίτο κεφάλαιο, ο αναγνώστης συναντά το πρόβλημα οικονομικής κατανομής φορτίου αλλά και της ένταξης μονάδων παραγωγής. Αφού λοιπόν, παρουσιαστεί η βελτιστοποίηση συναρτήσεων, επιλύονται τρία παραδείγματα

του προβλήματος κατανομής και δύο της ένταξης μονάδων. Στο τέταρτο και τελευταίο κεφάλαιο, παρουσιάζονται και αναλύονται, οι κώδικες οι οποίοι δημιουργήθηκαν, για την επίλυση των παραδειγμάτων του κεφαλαίου 3.



# Περιεχόμενα

---

Περίληψη .....	5
Abstract .....	6
Κεφάλαιο 1: Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας, μελέτη και πρόβλεψη φορτίου.....	16
1.1. Εισαγωγή.....	16
1.2. Λογισμικό των ΣΕΕ.....	17
1.2.1. Υποσύστημα παραγωγής .....	18
1.2.2. Υποσύστημα εποπτικού ελέγχου και συλλογής πληροφοριών – SCADA .	18
1.2.3. Υποσύστημα ηλεκτρικού δικτύου μεταφοράς.....	19
1.3. Μελέτη ηλεκτρικής ζήτησης .....	19
1.4. Πρόβλεψη φορτίου .....	21
1.4.1. Στατικά μοντέλα πρόβλεψης.....	22
1.4.2. Δυναμικά μοντέλα πρόβλεψης.....	22
Κεφάλαιο 2: Θερμικά συστήματα παραγωγής ενέργειας και καμπύλες εισόδου – εξόδου .....	23
2.1. Εισαγωγή.....	23
2.2. Θερμικοί σταθμοί παραγωγής ενέργειας.....	24
2.2.1. Ατμοηλεκτρικός σταθμός .....	24
2.2.2. Αεριοστροβιλικός σταθμός .....	25
2.2.3. Σταθμός συνδυασμένου κύκλου .....	26
2.2.4. Ντιζελοηλεκτρικός σταθμός .....	27
2.3. Χαρακτηριστικές καμπύλες θερμικών σταθμών .....	28
2.3.1. Χαρακτηριστικές καμπύλες κατανάλωσης καυσίμου.....	29
2.3.2. Χαρακτηριστικές καμπύλες κατανάλωσης θερμότητας .....	31
2.3.3. Χαρακτηριστικές καμπύλες κόστους λειτουργίας.....	33
2.4. Κόστος εκκίνησης θερμικών μονάδων.....	35

Κεφάλαιο 3: Πρόβλημα οικονομικής κατανομής φορτίου και ένταξης μονάδων παραγωγής.....	38
3.1. Εισαγωγή.....	38
3.2. Βελτιστοποίηση συναρτήσεων.....	39
3.2.1. Ελαχιστοποίηση συναρτήσεων μιας μεταβλητής, χωρίς περιορισμούς.....	39
3.2.2. Ελαχιστοποίηση συναρτήσεων πολλών μεταβλητών, χωρίς περιορισμούς.....	41
3.3. Ελαχιστοποίηση συνάρτησης με εξισωτικούς περιορισμούς.....	42
3.3.1. Ελαχιστοποίηση με έναν εξισωτικό περιορισμό.....	43
3.3.2. Ελαχιστοποίηση με πολλούς εξισωτικούς περιορισμούς.....	44
3.4. Ελαχιστοποίηση συνάρτησης με ανισωτικούς περιορισμούς.....	46
3.5. Πρόβλημα οικονομικής κατανομής φορτίου.....	48
3.5.1. Η απλούστερη μορφή του προβλήματος οικονομικής κατανομής.....	48
Παράδειγμα 1.....	50
3.5.2. Οικονομική κατανομή φορτίου και λειτουργικά όρια μονάδων.....	54
Παράδειγμα 2.....	57
3.5.3. Οικονομική κατανομή φορτίου και απώλειες μεταφοράς.....	60
Παράδειγμα 3.....	66
3.6. Ένταξη μονάδων παραγωγής.....	73
Παράδειγμα 4.....	74
Παράδειγμα 5.....	78
Κεφάλαιο 4: Ανάπτυξη λογισμικού για επίλυση του προβλήματος οικονομικής κατανομής.....	82
4.1. Εισαγωγή.....	82
4.2. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 1.....	83
4.3. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 2.....	89
4.4. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 3.....	101
4.5. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 4.....	115
4.6. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 5.....	128

Συμπεράσματα .....	141
Βιβλιογραφία .....	142

## Περιεχόμενα Γραφημάτων

---

Γράφημα 1.1: Τυπική καμπύλη φορτίου. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας).....	20
Γράφημα 2.1: Μηνιαία παραγωγή ενέργειας για το έτος 2022. (Πηγή: ΑΔΜΗΕ) .....	23
Γράφημα 2.2: Καμπύλη ωριαίας κατανάλωσης καυσίμου. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	30
Γράφημα 2.3: Καμπύλη διαφορικής κατανάλωσης καυσίμου. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	30
Γράφημα 2.4: Καμπύλη ειδικής κατανάλωσης καυσίμου. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	31
Γράφημα 2.5: Καμπύλη ωριαίας κατανάλωσης θερμότητας. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	32
Γράφημα 2.6: Καμπύλη διαφορικής κατανάλωσης θερμότητας. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας).....	32
Γράφημα 2.7: Καμπύλη ειδικής κατανάλωσης θερμότητας. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	33
Γράφημα 2.8: Καμπύλη ωριαίου κόστους λειτουργίας. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	33
Γράφημα 2.9: Καμπύλη διαφορικού κόστους λειτουργίας. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	34
Γράφημα 2.10: Καμπύλη ειδικού κόστους λειτουργίας. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	34
Γράφημα 2.11: Χρονική μεταβολή της ισχύος, κατά την εκκίνηση μιας μονάδας. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	36
Γράφημα 3.1: Καμπύλη φορτίου. ....	78
Γράφημα 3.2: Πρόγραμμα Ένταξης Μονάδων.....	79
Γράφημα 3.1: Καμπύλη φορτίου. ....	129

## Περιεχόμενα Διαγραμμάτων

---

Διάγραμμα 1.1: Σύστημα Ελέγχου Ενέργειας. ....	17
Διάγραμμα 3.1: Καμπύλη ωριαίου κόστους, της πρώτης μονάδας που χρησιμοποιεί καύσιμο φυσικό αέριο.....	51
Διάγραμμα 3.2: Καμπύλη ωριαίου κόστους, της δεύτερης μονάδας που χρησιμοποιεί καύσιμο μαζούτ.....	52
Διάγραμμα 3.3: Καμπύλη ωριαίου κόστους, της τρίτης μονάδας που χρησιμοποιεί καύσιμο μαζούτ.....	52
Διάγραμμα 3.4: Αλγόριθμος Επαναλήψεων Λάμδα. ....	56
Διάγραμμα 3.5: Αλγόριθμος Οικονομικής Κατανομής Φορτίου με Απώλειες. ....	65

## Περιεχόμενα Εικόνων

---

Εικόνα 2.1: Ατμοηλεκτρική μονάδα. (Πηγή: <a href="https://www.shiksha.com">https://www.shiksha.com</a> ).....	25
Εικόνα 2.2: Αεριοστροβιλική μονάδα. (Πηγή: <a href="https://commons.wikimedia.org">https://commons.wikimedia.org</a> ).....	26
Εικόνα 2.3: Μονάδα συνδυασμένου κύκλου. (Πηγή: <a href="https://www.valmet.com">https://www.valmet.com</a> ).....	27
Εικόνα 2.4: Ντιζελοηλεκτρική μονάδα. (Πηγή: <a href="https://www.researchgate.net">https://www.researchgate.net</a> ) .....	28

## Περιεχόμενα Πινάκων

---

Πίνακας 1.1: Πρόβλεψη φορτίου ανάλογα τον χρονικό ορίζοντα τον οποίο αποσκοπούν. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	21
Πίνακας 2.1: Χαρακτηριστικά μεγέθη καμπυλών εισόδου – εξόδου. ....	29
Πίνακας 2.2: Ενδεικτικοί χρόνοι ψυχρής και θερμής εκκίνησης ατμοηλεκτρικής μονάδας, έως την ανάληψη του ονομαστικού του φορτίου. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας) .....	36
Πίνακας 2.3: Καταμερισμός κόστους εκκίνησης ατμοηλεκτρικής μονάδας. ....	37
Πίνακας 3.1: Αποτελέσματα παραδείγματος 4. ....	76
Πίνακας 3.2: Αποτελέσματα παραδείγματος 5. ....	80
Πίνακας 4.1: Δεδομένα του παραδείγματος 2, στο αρχείο data.xlsx. ....	99
Πίνακας 4.2: Δεδομένα του Σεναρίου Α, στο αρχείο data.xlsx. ....	100
Πίνακας 4.3: Αποτελέσματα παραδείγματος 4. ....	127
Πίνακας 4.4: Αποτελέσματα παραδείγματος 4, από το Matlab. ....	128
Πίνακας 4.5: Δεδομένα του παραδείγματος 5, στο αρχείο data.xlsx. ....	130
Πίνακας 4.6: Αποτελέσματα παραδείγματος 5. ....	139
Πίνακας 4.7: Αποτελέσματα παραδείγματος 4, από το Matlab. ....	139

# Κεφάλαιο 1: Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας, μελέτη και πρόβλεψη φορτίου

---

## 1.1. Εισαγωγή

Τα πρώτα συστήματα ηλεκτρικής ενέργειας (ΣΗΕ), έκαναν την εμφάνιση τους την δεκαετία του 1880. Με την πάροδο του χρόνου, λόγω των αυξημένων απαιτήσεων σε ζήτηση ηλεκτρικής ενέργειας, τα ΣΗΕ έκτοτε έγιναν, όλο και πιο μεγάλα σε μέγεθος αλλά και πιο περίπλοκα. Αυτή η συνεχόμενη αυξητική τάση, έφερε την ανάγκη να δημιουργηθεί ένα ενιαίο σύστημα ελέγχου ενέργειας (ΣΕΕ), το οποίο αποτελείται από μικρότερα περιφερειακά κέντρα ελέγχου ενέργειας (ΠΚΕΕ), ώστε να γίνεται ευκολότερα η διαχείριση της παραγόμενης ενέργειας για μεγάλες γεωγραφικές εκτάσεις. Στην Ελλάδα, σύμφωνα με τον ΑΔΜΗΕ, όλη η λειτουργία και ο έλεγχος του συστήματος γίνεται κυρίως από το εθνικό κέντρο ελέγχου ενέργειας (ΕΚΕΕ) καθώς και από τα τρία ΠΚΕΕ που βρίσκονται στην χώρα.

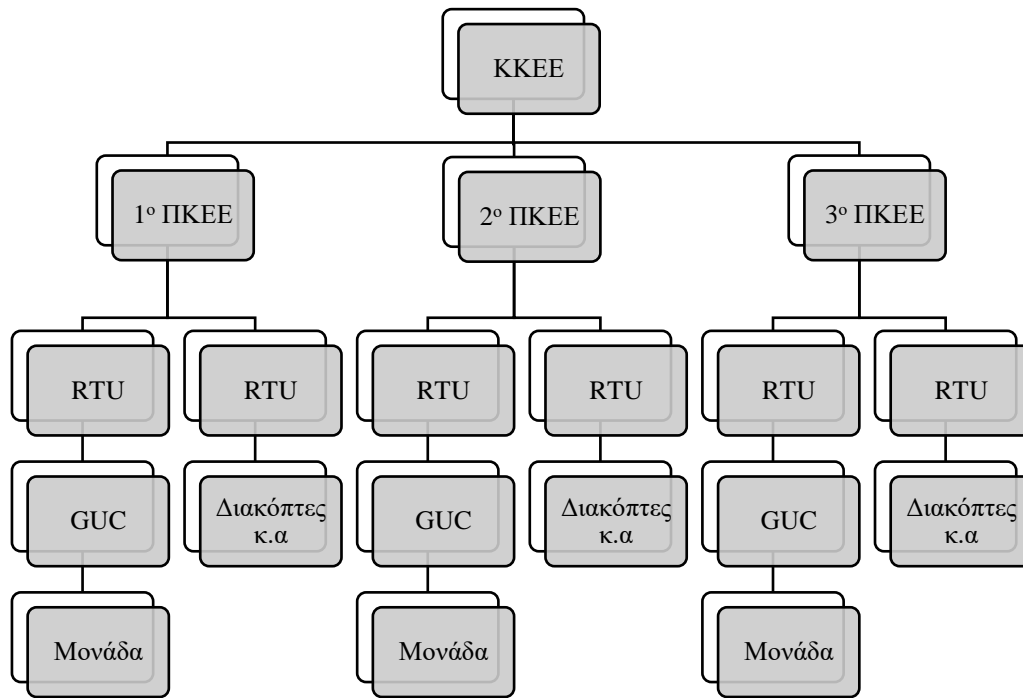
Στο ακόλουθο διάγραμμα παρουσιάζεται ένα ΣΕΕ, το οποίο απαρτίζεται από ένα κεντρικό κέντρο ελέγχου ενέργειας και τρία ΠΚΕΕ. Κάθε ΚΕΕ επικοινωνεί με τους σταθμούς παραγωγής, μέσω RTUs (τερματικών μονάδων) και GUCs (μονάδων ελέγχου παραγωγής), ενώ με τους υποσταθμούς παραγωγής μόνο με RTUs.

- **RTUs:** Σκοπός των RTUs είναι η συλλογή δεδομένων από σταθμούς και υποσταθμούς αλλά και η έκδοση εντολών που αποστέλλονται από τα ΚΕΕ προς τους σταθμούς.
- **GUCs:** Σκοπός των GUCs είναι η εξισορρόπηση της παραγόμενης ενέργειας με την ζήτηση εκείνης της στιγμής. Αν για παράδειγμα η παραγωγή εκείνη την στιγμή είναι μεγαλύτερη από την απαιτούμενη ενέργεια, η GUC ρίχνει τις στροφές του ρυθμιστή της μονάδας προκειμένου να έρθει η παραγόμενη ενέργεια στα θεμιτά επίπεδα.

Ο τρόπος με τον οποίο είναι σχεδιασμένα τα ΣΕΕ, δεν είναι μόνο για την καλύτερη διαχείριση της παραγόμενης ενέργειας, αλλά και για την καλύτερη διαχείριση σε περίπτωση εκτάκτων καταστάσεων, όπως για παράδειγμα η απώλεια



μιας μονάδας, ή απώλειας τάσης του δικτύου ή διατήρηση πόρων. Υπάρχουν λοιπόν, διάφορες δικλίδες ασφαλείας σε κάθε μέρος του ΣΕΕ, οι οποίες του επιτρέπουν να λειτουργεί αυτόνομα το σύστημα έως ότου αποκατασταθεί μια πιθανή βλάβη. Όμως για να λειτουργήσει σωστά ένα ΣΕΕ, χρειάζεται η υποστήριξη από ένα λογισμικό το οποίο επιτρέπει όλα τα παραπάνω.



Διάγραμμα 1.1: Σύστημα Ελέγχου Ενέργειας.

## 1.2. Λογισμικό των ΣΕΕ

Για την εύρυθμη λειτουργία των ΣΕΕ, δημιουργήθηκε εδώ και αρκετά έτη ένα λογισμικό το οποίο είναι υπεύθυνο για την ενεργειακή διαχείριση. Το λογισμικό αυτό, αποτελείται από τρία βασικά υποσυστήματα. Το υποσύστημα παραγωγής, το υποσύστημα εποπτικού ελέγχου και συλλογής πληροφοριών, γνωστό και ως SCADA (Supervisory Control and Data Acquisition) και τέλος το υποσύστημα ηλεκτρικού δικτύου μεταφοράς (Network).

### 1.2.1. Υποσύστημα παραγωγής

Το υποσύστημα παραγωγής, είναι υπεύθυνο για τον προγραμματισμό και έλεγχο της παραγωγής ενέργειας. Πιο συγκεκριμένα, χαρακτηρίζεται από λειτουργίες όπως, ο αυτόματος έλεγχος παραγωγής (ΑΕΠ), ο οποίος είναι υπεύθυνος για την ρύθμιση της παραγωγής με απώτερο σκοπό την κάλυψη των αναγκών ενέργειας, την οικονομική κατανομή φορτίου, η οποία είναι υπεύθυνη για την παραγωγή ενέργειας με το λιγότερο δυνατό κόστος, την ένταξη μονάδων παραγωγής ώστε να καλύπτεται η προβλεπόμενη ζήτηση της επόμενης ημέρας ή και των επόμενων ημερών – εβδομάδων. Ακόμα, χαρακτηρίζεται από την αξιολόγηση των ανταλλαγών, η οποία είναι υπεύθυνη για την βέλτιστη ένταξη των μονάδων παραγωγής και για την ανταλλαγή ενέργειας με τα γειτονικά συστήματα και τέλος την βραχυπρόθεσμη πρόβλεψη του φορτίου, όπου δίνει τα αποτελέσματα της πρόβλεψης στα υποσυστήματα της ένταξης μονάδων και αξιολόγησης ανταλλαγών.

### 1.2.2. Υποσύστημα εποπτικού ελέγχου και συλλογής πληροφοριών – SCADA

Το υποσύστημα SCADA, είναι υπεύθυνο για την επιτήρηση του ηλεκτρικού δικτύου καθώς και για τους τηλεχειρισμούς. Μέσω του SCADA, γίνεται συλλογή δεδομένων αλλά και επεξεργασία τους. Έχει σημαντικό ρόλο στην λειτουργία του ΣΕΕ, καθώς έχει την δυνατότητα να κρατά ιστορικό κατά την λειτουργία των μονάδων, να τοποθετεί εικονικές πινακίδες στον εξοπλισμό του συστήματος, να επεξεργάζεται την τοπολογία που προσδιορίζει την τρέχουσα κατάσταση ενός υποσταθμού και τέλος μπορεί να αποκόψει το φορτίο οποιαδήποτε στιγμή, από όλες τις μονάδες, με ένα ομαδικό κλείσιμο διακοπών.

Επιπλέον, μέσω ενός τέτοιου συστήματος γίνεται η καταχώρηση των απαιτούμενων τιμών παραγωγής. Από εκεί, οι επιθυμητές τιμές αποστέλλονται στα RTU's και έπειτα στους ελεγκτές μονάδων GUCs. Έτσι, επιτυγχάνεται η διαχείριση της παραγόμενης ενέργειας κάθε στιγμή.

### 1.2.3. Υποσύστημα ηλεκτρικού δικτύου μεταφοράς

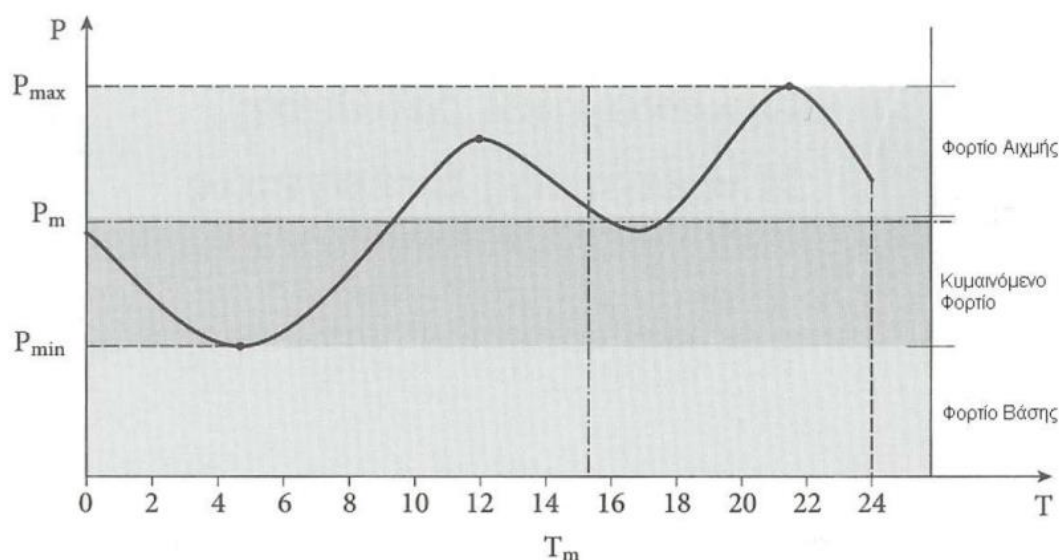
Το υποσύστημα του ηλεκτρικού δικτύου μεταφοράς, συνεργάζεται με το σύστημα μεταφοράς ενέργειας. Είναι υπεύθυνο για την εκτίμηση της λειτουργικής κατάστασης, μετά από χρήση μαθηματικών αλγορίθμων με τα δεδομένα που έχουν συλλεχθεί από το SCADA. Έτσι, οι χειριστές έχουν μια πλήρη εικόνα όχι μόνο για την απόδοση των μονάδων αλλά και για το κόστος των απωλειών μεταφοράς, τα οποία όπως θα μελετηθεί στην συνέχεια, έχουν σημαντικό ρόλο στην οικονομική κατανομή των φορτίων στις μονάδες παραγωγής. Επίσης, μέσω του υποσυστήματος αυτού, δίνεται η δυνατότητα να μελετηθεί η ροή του φορτίου καθώς και να αναλυθούν ενδεχόμενες διαταραχές.

Χάρη σε αυτό το ειδικά διαμορφωμένο λογισμικό, εφαρμόζεται κατά βέλτιστο τρόπο το πρόγραμμα παραγωγής που προκύπτει κατά τον ημερήσιο ή εβδομαδιαίο προγραμματισμό, αλλά και το πρόγραμμα ένταξης των μονάδων, με σκοπό την μείωση των λειτουργικών εξόδων, όπως τα κόστη καυσίμων, εκκίνησης, συντήρησης και λειτουργίας. Οι χειριστές επιπλέον, έχουν μια οπτική και για τα γειτονικά διασυνδεδεμένα συστήματα. Αυτό έχει ως αποτέλεσμα οι ανταλλαγές ενέργειας να γίνονται ευκολότερα. Έχοντας όλες αυτές τις λειτουργίες διαθέσιμες, σε συνεργασία με τον ΑΕΠ, υπολογίζεται το οικονομικότερο, πλέον, πρόγραμμα κατανομής φορτίου στις μονάδες παραγωγής ενέργειας.

## 1.3. Μελέτη ηλεκτρικής ζήτησης

Κύριος σκοπός μιας εταιρείας παραγωγής ηλεκτρικής ενέργειας, είναι η κάλυψη της ζήτησης σε ενέργεια. Για τον λόγο αυτό η εταιρεία λειτουργεί όσες μονάδες κρίνει απαραίτητο, προκειμένου να καλυφθεί η ζήτηση αυτή. Η ζήτηση όμως δεν είναι σταθερή. Σε διάστημα εικοσιτετράωρου, παρατηρείται μια αυξομείωση στην καμπύλη του φορτίου, κάτι το οποίο είναι αναμενόμενο, καθώς το βράδυ δεν υπάρχουν οι ίδιες απαιτήσεις σε ενέργεια, όσο τις μεσημεριανές ώρες. Στο γράφημα 1.1, παρουσιάζεται μια τυπική καμπύλη φορτίου σε εικοσιτετράωρη βάση. Στον

κάθετο άξονα βρίσκεται το φορτίο ή ζήτηση ισχύος, ενώ στον οριζόντιο, ο χρόνος. Το εμβαδόν το οποίο σχηματίζεται από την καμπύλη και τον άξονα του χρόνου εκφράζει και την κατανάλωση της ενέργειας.



Γράφημα 1.1: Τυπική καμπύλη φορτίου.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

Μια εταιρεία ενέργειας καλείται να καλύψει το εμβαδόν αυτό σε καθημερινή βάση. Για να επιτευχθεί αυτό με τον οικονομικότερο τρόπο, η κάλυψη του φορτίου χωρίζεται σε τρεις ζώνες. Την ζώνη του φορτίου βάσης, του κυμαινόμενου φορτίου και του φορτίου αιχμής. Η ενέργεια που καταναλώνεται στο φορτίο βάσης είναι συνεχόμενη και μεγάλη σε ποσότητα. Για τον λόγο αυτό, την ζώνη βάσης φορτίου την καλύπτουν μονάδες, οι οποίες παράγουν φθηνά ενέργεια. Από την άλλη, η ζώνη του κυμαινόμενου φορτίου παρουσιάζει συχνές αυξομειώσεις του φορτίου. Την κάλυψη της ζώνης αυτής την αναλαμβάνουν μονάδες με μικρούς χρόνους εκκίνησης, οι οποίες είναι πιο κοστοβόρες από τις μονάδες του φορτίου βάσης. Τέλος, η ζώνη του φορτίου αιχμής συναντάται τις λιγότερες ώρες μέσα στην ημέρα και την κάλυψη της, την αναλαμβάνουν μονάδες όπου παράγουν ακριβά την ενέργεια.

## 1.4. Πρόβλεψη φορτίου

Για την σωστή λειτουργία των συστημάτων ηλεκτρικής ενέργειας, οι εταιρείες είναι σημαντικό να λαμβάνουν πληροφορίες για την ηλεκτρική ζήτηση. Μάλιστα χρειάζεται να γνωρίζουν λεπτομέρειες που αφορούν την ζήτηση για τα επόμενα δευτερόλεπτα έως και αρκετά χρόνια μετά, ώστε να γίνεται αξιόπιστα και οικονομικά η λειτουργία των συστημάτων αυτών. Υπάρχουν διάφορα είδη προβλέψεων της ζήτησης, τα οποία χαρακτηρίζονται ανάλογα τον χρονικό ορίζοντα τον οποίο αποσκοπούν. Στον πίνακα 1.1 παρουσιάζονται τα είδη αυτά.

Πίνακας 1.1: Πρόβλεψη φορτίου ανάλογα τον χρονικό ορίζοντα τον οποίο αποσκοπούν.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

Είδος Πρόβλεψης	Χρονικός Ορίζοντας	Λειτουργίες στις οποίες χρειάζεται
<b>Μακροπρόθεσμη Πρόβλεψη</b>	10 έως 25 έτη	<ul style="list-style-type: none"><li>• Προγραμματισμός Ένταξης Νέων Μονάδων Παραγωγής</li></ul>
<b>Μεσοπρόθεσμη Πρόβλεψη</b>	Μερικοί μήνες έως 5 έτη	<ul style="list-style-type: none"><li>• Προγραμματισμός Καυσίμων</li><li>• Χρήση Υδάτινων Πόρων</li><li>• Προγραμματισμός Συντήρησης</li></ul>
<b>Βραχυπρόθεσμη Πρόβλεψη</b>	Μερικές ώρες έως 1 εβδομάδα	<ul style="list-style-type: none"><li>• Ένταξη μονάδων</li><li>• Υδροθερμική Συνεργασία</li><li>• Ανταλλαγές Ισχύος</li></ul>
<b>Πολύ Βραχυπρόθεσμη Πρόβλεψη</b>	Μερικά δευτερόλεπτα έως μερικά λεπτά	<ul style="list-style-type: none"><li>• Αυτόματος Έλεγχος Παραγωγής</li><li>• Οικονομική Κατανομή Φορτίου</li></ul>

Αυτό που απασχολεί περισσότερο το αντικείμενο της εργασίας αυτής, είναι η βραχυπρόθεσμη και η πολύ βραχυπρόθεσμη πρόβλεψη. Στις προβλέψεις αυτές συναντάται η καμπύλη φορτίου, η οποία αν προβλεφθεί, μπορούν να υπολογιστούν δεδομένα όπως η ημερήσια αιχμή και η ημερήσια ενέργεια αλλά τα μέσα ωριαία φορτία της επόμενης ημέρας ή της επόμενης εβδομάδας. Για τον υπολογισμό της καμπύλης φορτίου χρησιμοποιούνται στατικά αλλά και δυναμικά μοντέλα πρόβλεψης.

### 1.4.1. Στατικά μοντέλα πρόβλεψης

Τα στατικά μοντέλα πρόβλεψης είναι απλά σε δομή και έχουν εύκολη λύση, όμως δεν εξετάζουν την συσχέτιση του φορτίου με τις αμέσως προηγούμενες τιμές και δεν μοντελοποιούν το φορτίο με τις καιρικές μεταβολές, με αποτέλεσμα την παρουσία σφαλμάτων στις προβλέψεις τους. Ένα τέτοιο μοντέλο μπορεί να λειτουργεί ως εξής. Αρχικά, συλλέγονται διάφορα ιστορικά στοιχεία για το φορτίο και διάφορες τυπικές καμπύλες ανάλογα τις καιρικές συνθήκες όπως η θερμοκρασία και η υγρασία και στην συνέχεια συνδυάζονται με τις μετρήσεις των τελευταίων ημερών. Έτσι προκύπτει μια πρόβλεψη της επόμενης ημέρας ή της επόμενης εβδομάδας. Η παραπάνω διαδικασία εξαρτάται σε μεγάλο βαθμό από την εμπειρία του χειριστή του μοντέλου.

### 1.4.2. Δυναμικά μοντέλα πρόβλεψης

Τα δυναμικά μοντέλα πρόβλεψης είναι πιο περίπλοκα σε δομή και λαμβάνουν υπόψιν τους τις καιρικές συνθήκες, τυχαίους παράγοντες όπως για παράδειγμα μια γιορτή όπου η ζήτηση ενέργειας μπορεί να είναι πιο αυξημένη από ότι συνήθως, καθώς και τις πρόσφατες τιμές του φορτίου. Τα μοντέλα αυτά χωρίζονται σε δύο κατηγορίες, στα μοντέλα χρονοσειρών ARMA και στα μοντέλα χώρου κατάστασης. Ανάλογα τα δεδομένα που είναι διαθέσιμα, τα δυναμικά μοντέλα μπορεί να δώσουν προβλέψεις μεγαλύτερης ακρίβειας.

Αφού υπολογιστεί η καμπύλη φορτίου με την καταλληλότερη μέθοδο πρόβλεψης, οι χειριστές έχουν όλα τα δεδομένα για την διαχείριση και κατανομή της ενέργειας στους σταθμούς παραγωγής. Το διαθέσιμο λογισμικό, δίνει τις κατάλληλες εντολές για σωστό προγραμματισμό παραγωγής, προκειμένου να καλυφθεί η ζήτηση με τον πιο αξιόπιστο και οικονομικότερο δυνατό τρόπο. Τα δεδομένα μεταφέρονται από το SCADA, στα KEE, από κει στα ΠΚΕΕ και καταλήγουν στο Κεντρικό ΚΕΕ, από όπου και γίνεται ο κεντρικός έλεγχος του συστήματος ενέργειας.

## Κεφάλαιο 2: Θερμικά συστήματα παραγωγής ενέργειας και καμπύλες εισόδου – εξόδου

### 2.1. Εισαγωγή

Στην Ελλάδα, η ενέργεια παράγεται ως επί το πλείστον σε θερμικούς σταθμούς παραγωγής ενέργειας. Τα τελευταία χρόνια ωστόσο, ένα ποσοστό της έχει αντικατασταθεί από ανανεώσιμες πηγές ενέργειας και υδροηλεκτρικούς σταθμούς, όμως οι θερμικοί σταθμοί παραγωγής εξακολουθούν να κυριαρχούν στην παραγωγή. Στο ακόλουθο γράφημα, παρουσιάζεται η παραγωγή ενέργειας του έτους 2022 ανά μήνα. Ειδικότερα, το 48.68% της συνολικής ενέργειας, παράχθηκε από θερμικούς σταθμούς, το 42.93% από ΑΠΕ ενώ, από υδροηλεκτρική παραγωγή μόλις το 8.38%.



Γράφημα 2.1: Μηνιαία παραγωγή ενέργειας για το έτος 2022.  
(Πηγή: ΑΔΜΗΕ)

Παρατηρείται ότι η παραγόμενη ενέργεια, προέρχεται κατά κύριο λόγο από θερμικούς σταθμούς ενώ υπάρχουν μήνες, όπου το μεγαλύτερο ποσοστό παραγωγής έρχεται από ΑΠΕ.

## 2.2. Θερμικοί σταθμοί παραγωγής ενέργειας

Όπως αναφέρθηκε και στον πρόλογο, ο θερμικός ή ο θερμοηλεκτρικός σταθμός παραγωγής ενέργειας είναι μια εγκατάσταση η οποία παράγει ηλεκτρική ενέργεια με την χρήση θερμικής ενέργειας. Ανάλογα τις μηχανές και το μέσο το οποίο χρησιμοποιείται ως καύσιμο συναντώνται τρεις τύποι σταθμών:

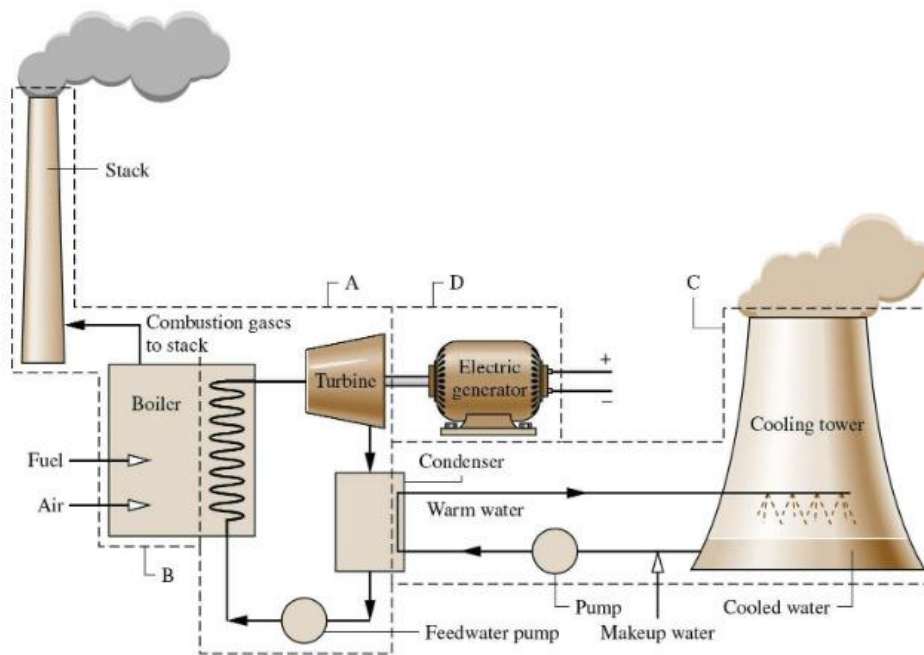
- ο ατμοηλεκτρικός
- ο αεριοστροβιλικός και
- ο ντιζελοηλεκτρικός

Πολλές φορές συναντάται και ο συνδυασμός σταθμών ή και διάφορες παραλλαγές αυτών με απώτερο σκοπό την καλύτερη απόδοση κατά την διάρκεια της παραγωγής.

### 2.2.1. Ατμοηλεκτρικός σταθμός

Ένας ατμοηλεκτρικός σταθμός (ΑΗΣ) χρησιμοποιεί ατμό και ατμοστρόβιλους. Κάθε ΑΗΣ αποτελείται από έναν λέβητα, ένα στρόβιλο, μια γεννήτρια, έναν πύργο ψύξης, μια καμινάδα και το δικό της ηλεκτρικό σύστημα μέχρι το δίκτυο. Υπάρχουν περιπτώσεις στις οποίες συνεργάζονται παράλληλα περισσότερες από μια μονάδες. Στις περιπτώσεις αυτές συναντώνται κοινές για το σύστημα οι μονάδες μεταφοράς καυσίμου και μερικές φορές οι καμινάδες.





Εικόνα 2.1: Ατμοηλεκτρική μονάδα.  
(Πηγή: <https://www.shiksha.com>)

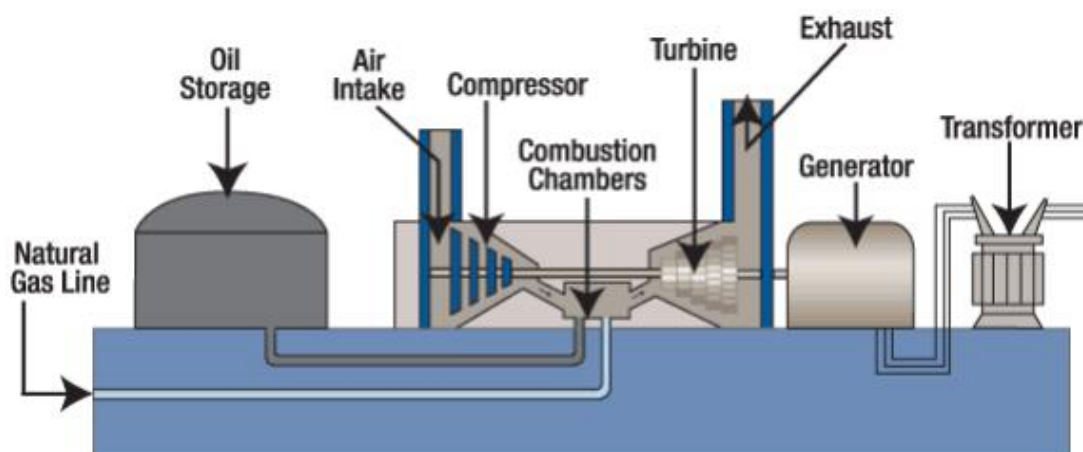
Στην εικόνα 2.1 παρουσιάζεται μια ατμοηλεκτρική μονάδα. Ο λέβητας (boiler) παράγει υπέρθερμο ατμό, ο οποίος με την σειρά του εκτονώνεται στον ατμοστρόβιλο (turbine) και κινεί την γεννήτρια (electric generator). Στην έξοδο του ατμοστρόβιλου, ο ατμός συμπυκνώνεται σε νερό στον συμπυκνωτή (condenser), όπου και γίνεται αποβολή θερμότητας στο περιβάλλον από τον πύργο ψύξης.

Αξίζει να σημειωθεί ότι οι ΑΗΣ, έχουν σχετικά μεγάλο βαθμό απόδοσης περίπου στο 30-45% και λόγω του φθηνού καυσίμου για την Ελλάδα (λιγνίτη), έχουν μικρό λειτουργικό κόστος για την παραγόμενη ενέργεια. Επίσης, η εκκίνηση και η ρύθμιση της ισχύος τους, είναι πολύπλοκη και αργή. Για τον λόγο αυτό, συναντώνται κυρίως ως σταθμοί βάσης.

### 2.2.2. Αεριοστροβιλικός σταθμός

Ένας αεριοστροβιλικός σταθμός χρησιμοποιεί καυσαέρια και αεριοστρόβιλους. Αποτελείται από τον συμπιεστή, το θάλαμο καύσης, τον αεριοστρόβιλο και τη

γεννήτρια. Στην εικόνα 2.2 παρουσιάζεται μια αεριοστροβιλική μονάδα. Αρχικά, εισέρχεται συμπιεσμένος αέρας στον θάλαμο καύσης στον οποίο ψεκάζεται ντίζελ ή φυσικό αέριο και γίνεται καύση. Τα καυσαέρια της διαδικασίας αυτής, οδηγούνται σε έναν αεριοστρόβilo όπου και εκτονώνονται. Ένα μέρος της ενέργειας κατά την εκτόνωση των καυσαερίων, μετατρέπεται σε μηχανική ενέργεια.



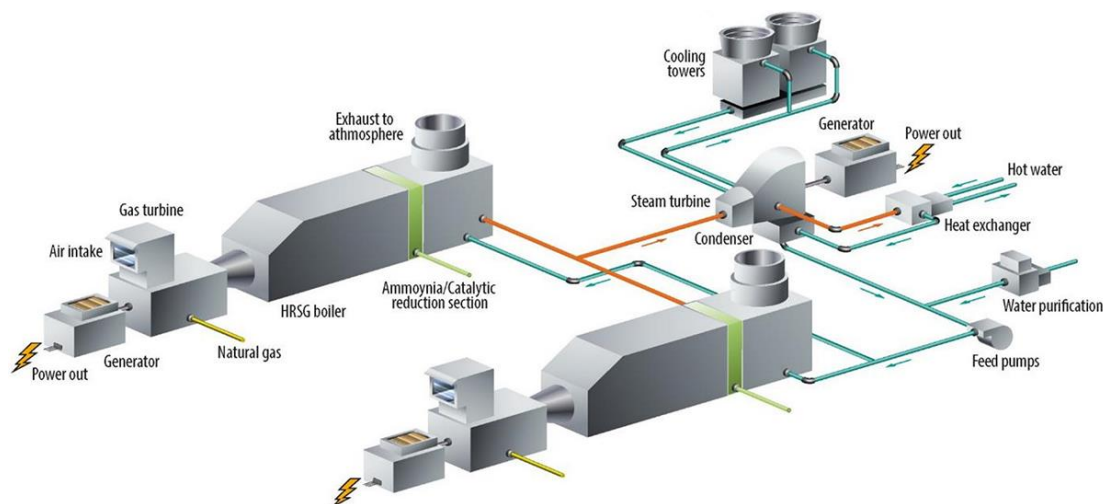
Εικόνα 2.2: Αεριοστροβιλική μονάδα.  
(Πηγή: <https://commons.wikimedia.org>)

Μια τέτοια μονάδα, έχει μικρό βαθμό απόδοσης περίπου στο 25-35%. Κύρια αιτία της χαμηλής αυτής απόδοσης, είναι η μεγάλη ισχύς που χρειάζεται ο αεροσυμπιεστής για να λειτουργήσει αλλά και η υψηλή θερμοκρασία των καυσαερίων στην έξοδο του συστήματος. Αυτές οι μονάδες έχουν μεγάλο κόστος λειτουργίας εξαιτίας καυσίμου, όμως η εκκίνηση τους είναι σύντομη. Για τον λόγο αυτόν χρησιμοποιούνται για να καλύψουν το φορτίο αιχμής.

### 2.2.3. Σταθμός συνδυασμένου κύκλου

Σε έναν τέτοιο σταθμό χρησιμοποιούνται ο συνδυασμός των ατμοηλεκτρικών και αεριοστροβιλικών σταθμών. Συνήθως, συναντώνται μια ή και παραπάνω αεριοστροβιλικές μονάδες και από ένα ζεύγος ατμοστρόβιλου – γεννήτριας. Τα καυσαέρια τα οποία παράγονται από τους αεριοστροβιλικούς σταθμούς, διοχετεύονται σε λέβητα καυσαερίων όπου εκεί παράγεται ατμός που κινεί τον

ατμοστρόβιλο. Στην εικόνα 2.3, παρουσιάζεται ένας σταθμός συνδυασμένου κύκλου με δύο αεριοστρόβιλους και έναν ατμοστρόβιλο.

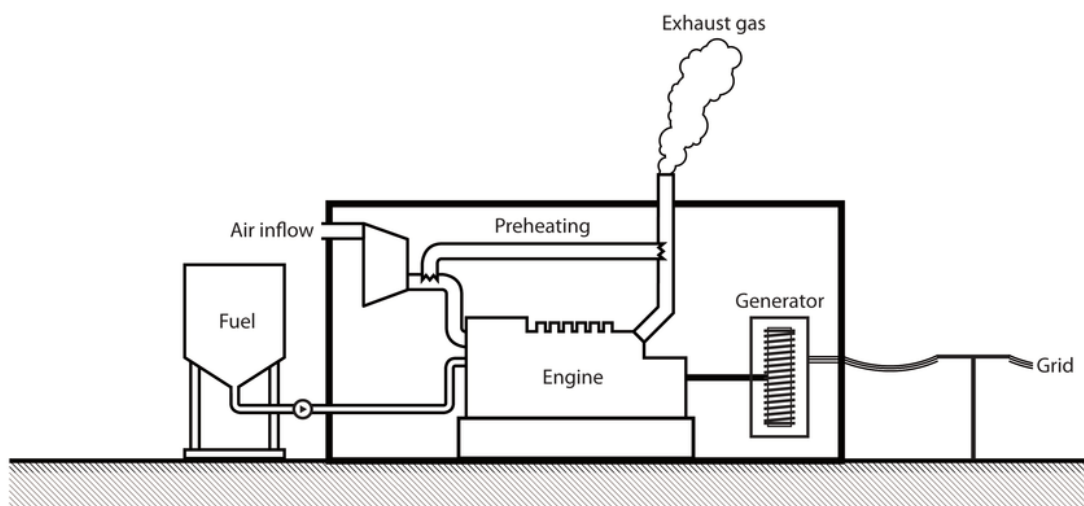


Εικόνα 2.3: Μονάδα συνδυασμένου κύκλου.  
(Πηγή: <https://www.valmet.com>)

Αυτού του είδους η μονάδα, έχει μεγάλο βαθμό απόδοσης περίπου στο 50%, αλλά η ρύθμιση της ισχύος τους δεν είναι τόσο άμεση όσο στους αεριοστροβιλικούς σταθμούς. Για αυτόν τον λόγο, χρησιμοποιούνται για να καλύψουν το κυμαινόμενο φορτίο.

#### 2.2.4. Ντιζελοηλεκτρικός σταθμός

Οι ντιζελοηλεκτρικοί σταθμοί, χρησιμοποιούν μηχανές εσωτερικής καύσης για την παραγωγή ενέργειας. Έχουν μεγάλο βαθμό απόδοσης περίπου στο 50% και χρησιμοποιούν ως καύσιμο, αποστάγματα του πετρελαίου. Η ρύθμιση της ισχύος τους επίσης, είναι γρήγορη και απροβλημάτιστη. Στην Ελλάδα δεν χρησιμοποιούνται στο κεντρικό δίκτυο ηλεκτρικής ενέργειας, αλλά σε μικρά αυτόνομα δίκτυα όπως για παράδειγμα σε κάποιο μικρό νησί. Στο εικόνα 2.4 παρουσιάζεται μια τέτοια μονάδα.



Εικόνα 2.4: Ντιζελοηλεκτρική μονάδα.  
(Πηγή: <https://www.researchgate.net>)

## 2.3. Χαρακτηριστικές καμπύλες θερμικών σταθμών

Για να γίνει η ανάλυση της οικονομικής λειτουργίας των συστημάτων ηλεκτρικής ενέργειας, χρησιμοποιούνται ορισμένες χαρακτηριστικές καμπύλες εισόδου – εξόδου των θερμικών μονάδων. Αυτού του είδους οι χαρακτηριστικές καμπύλες, ορίζουν την σχέση μεταξύ της συνολικής εισόδου και της καθαρής ισχύος εξόδου. Η καθαρή ισχύς εξόδου του συστήματος, είναι η ενέργεια οπου τελικά προσδίδεται στο ηλεκτρικό δίκτυο, έχοντας εξαιρέσει την ιδιοκατανάλωση της μονάδας, η οποία κυμαίνεται περίπου από 2% έως 8% της συνολικής παραγόμενης ενέργειας. Από την άλλη, η είσοδος του συστήματος είναι, είτε η παροχή καυσίμου, είτε η ωριαία κατανάλωση της θερμότητας, είτε το ωριαίο κόστος λειτουργίας.

Στον πίνακα 2.1 παρουσιάζονται τα μεγέθη τα οποία χρησιμοποιούνται για τον ορισμό των χαρακτηριστικών καμπυλών εισόδου – εξόδου. Παρατηρείται ότι στον πίνακα έχει προστεθεί και το μέγεθος της θερμογόνου δύναμης του καυσίμου η οποία αν είναι γνωστή μπορεί να υπολογιστεί το ωριαίο κόστος λειτουργίας της μονάδας.

Πίνακας 2.1: Χαρακτηριστικά μεγέθη καμπυλών εισόδου – εξόδου.

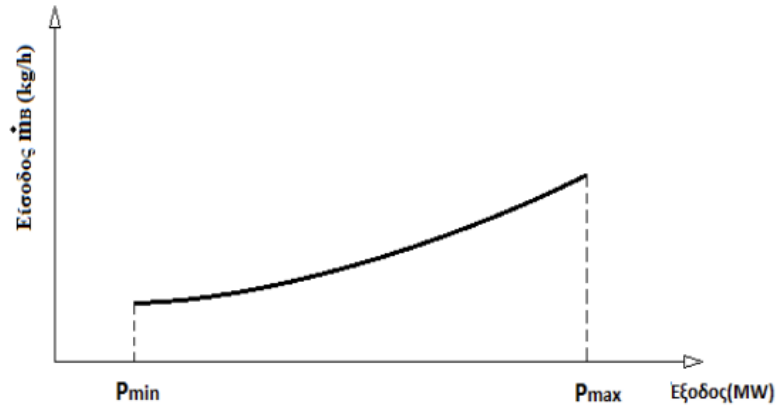
Σύμβολο	Περιγραφή	Μονάδα Μέτρησης
$\dot{m}_B$	Παροχή Καυσίμου	T/h
<b>H</b>	Ωριαία Κατανάλωση Θερμότητας	Gcal/h
<b>Q</b>	Θερμογόνος Δύναμη Καυσίμου	Kcal/kg
<b>T</b>	Τιμή Καυσίμου	€/kg
<b>F</b>	Ωριαίο Κόστος Λειτουργίας	€/h
<b>P</b>	Ηλεκτρική Ισχύς Εξόδου	MW

Ανάλογα με το αν η είσοδος του συστήματος είναι συνάρτηση της παροχής καυσίμου, της ωριαίας κατανάλωσης θερμότητας ή του ωριαίου κόστους λειτουργίας, παρουσιάζονται τρεις ομάδες χαρακτηριστικών καμπυλών των θερμικών μονάδων. Στις χαρακτηριστικές καμπύλες, στον κάθετο άξονα βρίσκεται η είσοδος στο σύστημα, ενώ στον οριζόντιο η καθαρή έξοδος της μονάδας σε MW. Η τιμή που παίρνει η έξοδος, κυμαίνεται μεταξύ μιας ελάχιστης τιμής  $P_{\min}$  και μιας μέγιστης  $P_{\max}$ . Η ελάχιστη τιμή της μονάδας,  $P_{\min}$ , επιβάλλεται από την ευστάθεια της καύσης στον καυστήρα και έχει τιμή περίπου το 30% της ονομαστικής ισχύος της μονάδας, ενώ η μέγιστη τιμή,  $P_{\max}$ , είναι ίση ή το πολύ 5% μεγαλύτερη της ονομαστικής μέγιστης ισχύος της μονάδας, καθώς οι θερμικές μονάδες δεν έχουν δυνατότητα υπερφόρτισης. Αξίζει να σημειωθεί πως στην ειδική περίπτωση των λιγνιτικών σταθμών, όταν η ισχύς εξόδου της είναι χαμηλή, χρειάζεται η παράλληλη καύση υγρού καυσίμου προκειμένου να παραμείνει η καύση στον καυστήρα.

### 2.3.1. Χαρακτηριστικές καμπύλες κατανάλωσης καυσίμου

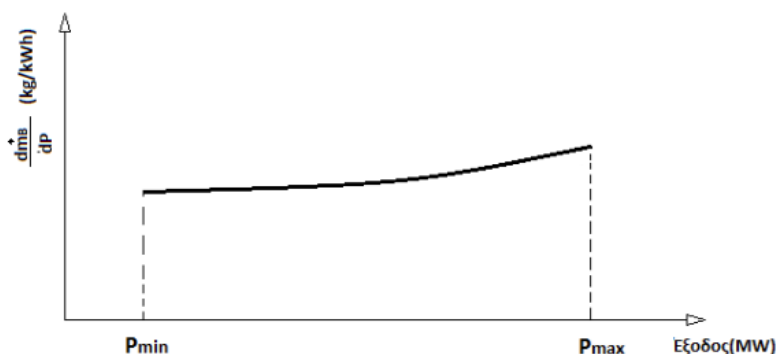
Η πρώτη ομάδα που συναντάται, είναι οι χαρακτηριστικές καμπύλες κατανάλωσης καυσίμου. Η ομάδα αυτή αποτελείται από τρεις χαρακτηριστικές καμπύλες, της ωριαίας κατανάλωσης καυσίμου, της διαφορικής κατανάλωσης καυσίμου και τέλος της ειδικής κατανάλωσης καυσίμου.

- **Καμπύλη ωριαίας κατανάλωσης καυσίμου:** Εκφράζει την παροχή καυσίμου  $\dot{m}_B$ , ως συνάρτηση της καθαρής ισχύος εξόδου  $P$ .



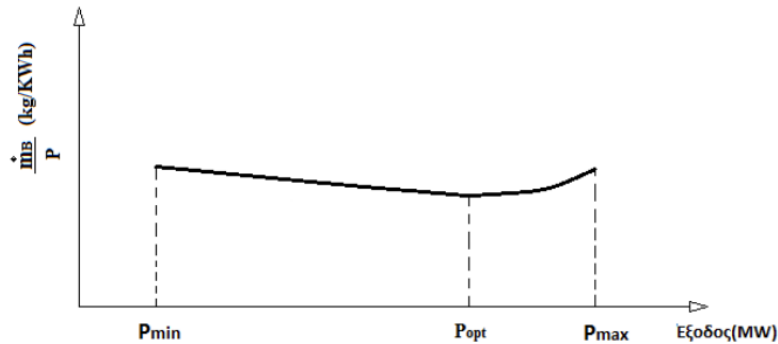
Γράφημα 2.2: Καμπύλη ωριαίας κατανάλωσης καυσίμου.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

- **Καμπύλη διαφορικής κατανάλωσης καυσίμου:** Εκφράζει την αύξηση της παροχής καυσίμου  $\Delta \dot{m}_B$ , που απαιτείται για την αύξηση της καθαρής ισχύος εξόδου κατά μια μονάδα,  $\Delta P=1\text{kW}$  ή αλλιώς την αύξηση της κατανάλωσης καυσίμου που απαιτείται για την αύξηση της παραγωγής ηλεκτρικής ενέργειας κατά  $\Delta E=1\text{kWh}$ . Ειδικότερα, η καμπύλη αυτή είναι η παράγωγος της ωριαίας κατανάλωσης καυσίμου ως προς την έξοδο,  $\frac{d\dot{m}_B}{dP}$  και έχει μονάδα μέτρησης  $\text{kg/kWh}$ . Αυτή η καμπύλη είναι αύξουσα συνάρτηση της ισχύος εξόδου.



Γράφημα 2.3: Καμπύλη διαφορικής κατανάλωσης καυσίμου.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

- **Καμπύλη ειδικής κατανάλωσης καυσίμου:** Εκφράζει τον λόγο της ωριαίας κατανάλωσης ή παροχής καυσίμου, ως προς την καθαρή έξοδο της μονάδας,  $\frac{\dot{m}_B}{P}$ . Η καμπύλη αυτή εκφράζεται συναρτήσει της ισχύος εξόδου και εμφανίζει ολικό ελάχιστο πριν το  $P_{max}$ . Στο σημείο εκείνο επιτυγχάνεται η οικονομικότερη λειτουργία του σταθμού.

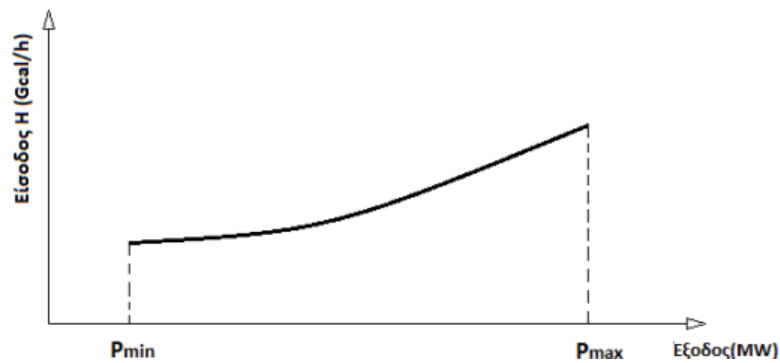


Γράφημα 2.4: Καμπύλη ειδικής κατανάλωσης καυσίμου.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

### 2.3.2. Χαρακτηριστικές καμπύλες κατανάλωσης θερμότητας

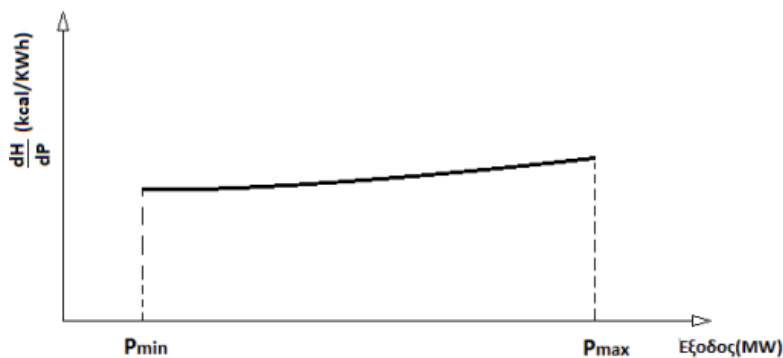
Η δεύτερη ομάδα που συναντάται, είναι οι καμπύλες κατανάλωσης θερμότητας. Όμοια και αυτή η ομάδα, αποτελείται από τρεις χαρακτηριστικές καμπύλες, της ωριαίας κατανάλωσης θερμότητας, της διαφορικής κατανάλωσης θερμότητας και τέλος της ειδικής κατανάλωσης θερμότητας.

- **Καμπύλη ωριαίας κατανάλωσης θερμότητας:** Εκφράζει τον ρυθμό παροχής θερμότητας,  $H$ , στον λέβητα, ως συνάρτηση της ισχύος εξόδου  $P$ .



Γράφημα 2.5: Καμπύλη ωριαίας κατανάλωσης θερμότητας.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

- **Καμπύλη διαφορικής κατανάλωσης θερμότητας:** Εκφράζει την πρόσθετη θερμότητα που χρειάζεται το σύστημα, για να αυξήσει την παραγωγή ηλεκτρικής ενέργειας κατά  $\Delta E=1$  kWh. Ειδικότερα, η καμπύλη αυτή είναι η παράγωγος της ωριαίας κατανάλωσης θερμότητας ως προς την έξοδο,  $\frac{dH}{dP}$  και έχει μονάδα μέτρησης kcal/kWh. Αυτή η καμπύλη, είναι αύξουσα συνάρτηση της ισχύος εξόδου.

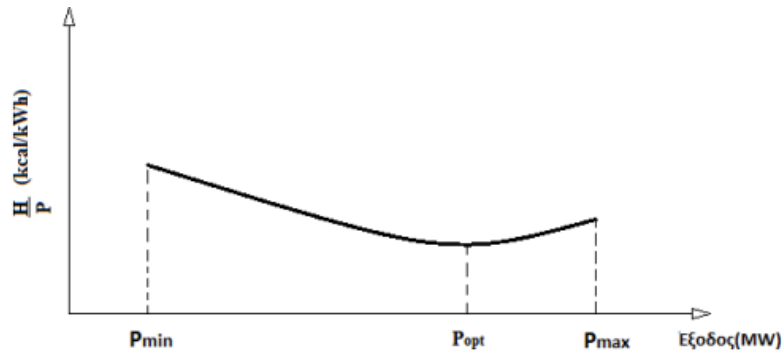


Γράφημα 2.6: Καμπύλη διαφορικής κατανάλωσης θερμότητας.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

- **Καμπύλη ειδικής κατανάλωσης θερμότητας:** Εκφράζει τον λόγο της ωριαίας κατανάλωσης θερμότητας, ως προς την έξοδο της μονάδας,  $\frac{H}{P}$ . Η καμπύλη αυτή εκφράζεται συναρτήσεως της ισχύος εξόδου της μονάδας και είναι το αντίστροφο του βαθμού απόδοσης της μονάδας. Έτσι, παρουσιάζει



ολικό ελάχιστο πριν το  $P_{\max}$ , όπου και επιτυγχάνεται ο μεγαλύτερος βαθμός απόδοσης της μονάδας με την λιγότερη κατανάλωση θερμότητας.

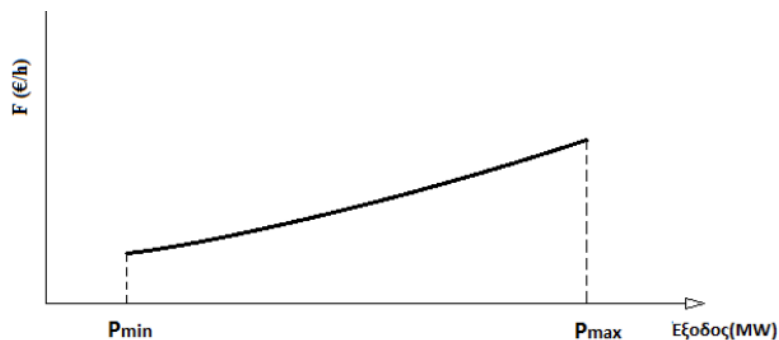


Γράφημα 2.7: Καμπύλη ειδικής κατανάλωσης θερμότητας.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

### 2.3.3. Χαρακτηριστικές καμπύλες κόστους λειτουργίας

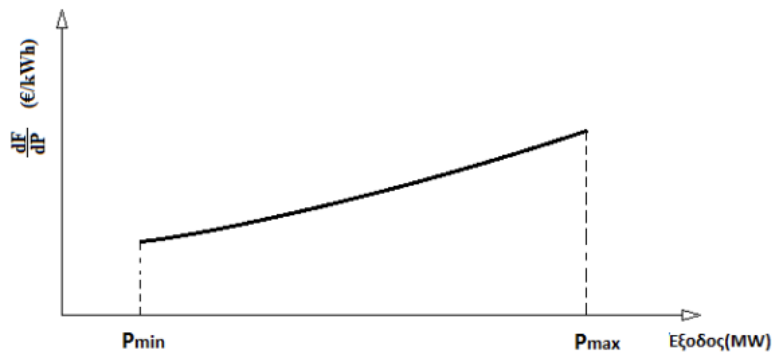
Η Τρίτη και τελευταία ομάδα που συναντάται, είναι οι καμπύλες κόστους λειτουργίας. Αποτελείται και αυτή από τρεις χαρακτηριστικές καμπύλες, του ωριαίου κόστους λειτουργίας, του διαφορικού κόστους λειτουργίας και του ειδικού κόστους λειτουργίας.

- **Καμπύλη ωριαίου κόστους λειτουργίας:** Εκφράζει το ωριαίο κόστος λειτουργίας  $F$  ως συνάρτηση της ισχύος εξόδου  $P$ .



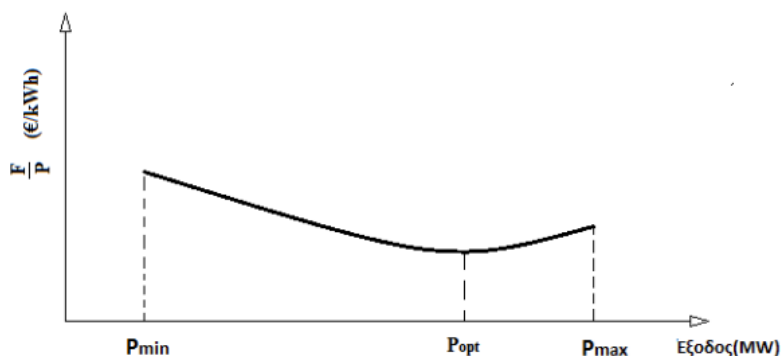
Γράφημα 2.8: Καμπύλη ωριαίου κόστους λειτουργίας.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

- **Καμπύλη διαφορικού κόστους λειτουργίας:** Εκφράζει την αύξηση του ωριαίου κόστους λειτουργίας  $\Delta F$ , που απαιτείται για την αύξηση της ισχύος εξόδου, κατά μια μονάδα,  $\Delta P=1\text{kW}$  ή αλλιώς την αύξηση του κόστους παραγωγής που προέρχεται από την παραγωγή μιας πρόσθετης κιλοβατώρας από την μονάδα. Ειδικότερα, η καμπύλη αυτή είναι η παράγωγος του ωριαίου κόστους ως προς την έξοδο,  $\frac{dF}{dP}$  και έχει μονάδα μέτρησης €/kWh. Αυτή η καμπύλη είναι αύξουσα συνάρτηση της ισχύος εξόδου.



Γράφημα 2.9: Καμπύλη διαφορικού κόστους λειτουργίας.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

- **Καμπύλη ειδικού κόστους λειτουργίας:** Εκφράζει τον λόγο του ωριαίου κόστους λειτουργίας ως προς την έξοδο της μονάδας,  $\frac{F}{P}$  ή αλλιώς την τιμή παραγωγής μιας κιλοβατώρας. Το ολικό ελάχιστο εμφανίζεται πριν το  $P_{\max}$ , όπου ελαχιστοποιείται το κόστος λειτουργίας.



Γράφημα 2.10: Καμπύλη ειδικού κόστους λειτουργίας.  
(Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

Αξίζει να αναφερθεί ότι, το ειδικό και το διαφορικό κόστος λειτουργίας, αναφέρονται και ως μέσο και οριακό κόστος λειτουργίας αντίστοιχα. Στο σημείο βέλτιστης λειτουργίας της μονάδας ή αλλιώς, στο σημείο ελαχιστοποίησης του κόστους λειτουργίας, το μέσο κόστος ισούται με το οριακό. Για τις τιμές εξόδου της μονάδας μικρότερες της βέλτιστης, το οριακό – διαφορικό κόστος είναι μικρότερο από το μέσο – ειδικό, ενώ για τιμές εξόδου μεγαλύτερες της βέλτιστης, το οριακό κόστος είναι μεγαλύτερο από το μέσο κόστος.

## 2.4. Κόστος εκκίνησης θερμικών μονάδων

Το κόστος εκκίνησης των θερμικών μονάδων αποτελεί σημαντικό μέγεθος για τον προγραμματισμό και την λειτουργία της. Η εκκίνηση της, δεν είναι μια απλή διαδικασία, ειδικά για τις ατμοηλεκτρικές μονάδες παραγωγής ενέργειας. Από την έναρξη της καύσης, έως την ένταξη της μονάδας στο δίκτυο, μεσολαβεί ένα χρονικό διάστημα και ένα υπολογίσιμο κόστος, το οποίο τελικά προστίθεται στο κόστος λειτουργίας. Για παράδειγμα, μια αεριοστροβιλική μονάδα έχει απλή εκκίνηση, με σχετικά ασήμαντο κόστος, αλλά με μεγάλα λειτουργικά κόστη. Αντίθετα μια ατμοηλεκτρική μονάδα, έχει πολύπλοκη εκκίνηση, με μεγάλο κόστος, αλλά με λίγα λειτουργικά κόστη. Αυτό έχει ως αποτέλεσμα τελικά, η ατμοηλεκτρική μονάδα να είναι οικονομικότερη, παρά το μεγάλο κόστος εκκίνησης.

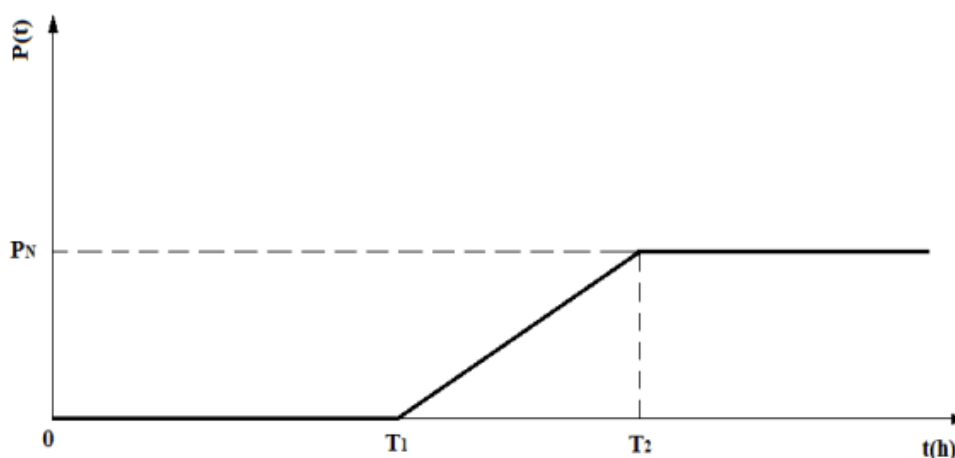
Το κόστος εκκίνησης μιας μονάδας, εξαρτάται πρωτίστως, από το μέγεθος της, το καύσιμο που χρησιμοποιεί και τέλος από τον χρόνο κράτησης. Ως χρόνος κράτησης, ορίζεται ο χρόνος στον οποίο η μονάδα παραμένει εκτός λειτουργίας. Ανάλογα τον χρόνο κράτησης, η εκκίνηση της μονάδας χαρακτηρίζεται ως θερμή ή ψυχρή. Η θερμή εκκίνηση, έχει μικρότερο κόστος από την ψυχρή. Γενικότερα, όσο αυξάνεται ο χρόνος κράτησης της μονάδας, τόσο αυξάνεται και το κόστος εκκίνησης, όπου τελικά παίρνει την μέγιστη του τιμή, κατά την ψυχρή εκκίνηση. Ενδεικτικά, στον πίνακα 2.2 παρουσιάζονται οι χρόνοι εκκίνησης μιας ατμοηλεκτρικής μονάδας,

με ισχύ 300MW. Παρατηρείται, πως η θερμή εκκίνηση διαφοροποιείται ανάλογα τον χρόνο κράτησης της μονάδας σε τρεις υποπεριπτώσεις.

Πίνακας 2.2: Ενδεικτικοί χρόνοι ψυχρής και θερμής εκκίνησης ατμοηλεκτρικής μονάδας, έως την ανάληψη του ονομαστικού του φορτίου. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

Εκκίνηση	Χρόνος κράτησης μονάδας (h)	Ενδεικτικός χρόνος εκκίνησης (h)
Ψυχρή	>48h	10h
Θερμή	24-40h	4,5h
	6-8h	3,5h
	1-2h	2h

Ιδιαίτερα αξιοσημείωτο είναι πως αποφεύγονται οι συχνές κρατήσεις και εκκινήσεις των μεγάλων λιγνιτικών μονάδων, εξαιτίας του υψηλού κόστους εκκίνησης αλλά και των καταπονήσεων που δέχονται. Στο ακόλουθο γράφημα παρουσιάζεται χρονικά, η μεταβολή της ισχύος της μονάδας, κατά την εκκίνηση. Θεωρείται ότι, την στιγμή  $t=0$  ξεκινά η διαδικασία της εκκίνησης, την στιγμή  $t=T_1$  συγχρονίζεται η μονάδα με το δίκτυο, ενώ την στιγμή  $t=T_2$ , η μονάδα φτάνει να παράγει την ονομαστική της ισχύ. Στο διάστημα μεταξύ της στιγμής  $T_1$  και  $T_2$ , φαίνεται κατά προσέγγιση η πραγματική μεταβολή ισχύος κατά την εκκίνηση.



Γράφημα 2.11: Χρονική μεταβολή της ισχύος, κατά την εκκίνηση μιας μονάδας. (Πηγή: Αν. Μπακιρτζής Οικονομική Λειτουργία Συστημάτων Ηλεκτρικής Ενέργειας)

Παρακάτω αναφέρονται μερικοί από τους παράγοντες οι οποίοι περιορίζουν την ταχύτητα εκκίνησης, σε μια ατμοηλεκτρική μονάδα:

- Η θερμική τάση που αναπτύσσεται μεταξύ των μετάλλων.
- Η διαφορική συστολή του στροφείου και του κελύφους.
- Η παραμόρφωση που εμφανίζουν τα μέταλλα των οριζόντιων αρμών του κελύφους.
- Οι ταλαντώσεις του ατμοστρόβιλου κατά την εκκίνηση του.

Στον ακόλουθο πίνακα γίνεται ενδεικτικός καταμερισμός του κόστους εκκίνησης μιας ατμοηλεκτρικής λιγνιτικής μονάδας 300MW, μέχρι την στιγμή ανάληψης του ονομαστικού του φορτίου. Τα δεδομένα τα οποία χρησιμοποιήθηκαν από το βιβλίο του κ. Μπακιρτζή για την δημιουργία του πίνακα, έχουν μονάδα μέτρησης τις δραχμές. Για τον λόγο αυτό, θα μετατραπεί η αξία τους σε ευρώ, σύμφωνα με την ισοτιμία του 2001, η οποία ήταν 1 ευρώ για 340,75 δραχμές.

Πίνακας 2.3: Καταμερισμός κόστους εκκίνησης ατμοηλεκτρικής μονάδας.

Αιτία	Κόστος (δρχ)	Κόστος (€)	Κόστος (%)
<b>Κάύσιμα πριν τον συγχρονισμό</b>	7,060,000	20,719	53%
<b>Κάύσιμα μετά τον συγχρονισμό</b>	4,543,000	13,332	34%
<b>Ηλεκτρική ενέργεια</b>	980,000	2,876	7%
<b>Αφαλατωμένο νερό</b>	197,000	578	1%
<b>Υπερωρίες προσωπικού</b>	476,000	1,397	4%
<b>Σύνολο</b>	13,256,000	38,902	100%

Της φαίνεται το κόστος εκκίνησης, κατέχει σημαντικό ρόλο στην οικονομική διαχείριση και προγραμματισμό λειτουργίας των μονάδων. Παρόλα αυτά, στα επόμενα κεφάλαια, δεν θα συμπεριληφθεί στην τελική λύση, καθώς ξεφεύγει από το αντικείμενο μελέτης της συγκεκριμένης εργασίας, αλλά είναι σημαντικό ο αναγνώστης να το γνωρίζει.

## Κεφάλαιο 3: Πρόβλημα οικονομικής κατανομής φορτίου και ένταξης μονάδων παραγωγής

---

### 3.1. Εισαγωγή

Στα προηγούμενα κεφάλαια παρουσιάστηκε ο τρόπος λειτουργίας του Συστήματος Ηλεκτρικής Ενέργειας αλλά και μελετήθηκαν τα θερμικά συστήματα, μαζί με τις χαρακτηριστικές καμπύλες εισόδου – εξόδου. Έχοντας ως υπόβαθρο τα παραπάνω, στο κεφάλαιο αυτό θα επιλυθεί το πρόβλημα οικονομικής κατανομής φορτίου σε θερμικούς σταθμούς παραγωγής ενέργειας. ***Το πρόβλημα αυτό εμφανίζεται, όταν δύο ή περισσότερες ηλεκτρικές μονάδες συνεργάζονται με σκοπό να ικανοποιήσουν την ηλεκτρική ζήτηση μιας περιοχής.*** Η λύση του, θα μπορούσε να αρκестεί στο ενδεχόμενο, ότι οι έξοδοι ισχύος των συνεργαζόμενων μονάδων, να ισούνται με το φορτίο που απαιτείται να καλυφθεί. Μια τέτοια προσέγγιση, θα μπορούσε να δώσει πολλές λύσεις στο πρόβλημα. Όταν όμως εισέρχεται ο περιορισμός της οικονομικότερης λειτουργίας του συστήματος, το πρόβλημα αποκτά μια βέλτιστη λύση. Ο στόχος της οικονομικής κατανομής φορτίου, είναι ο προσδιορισμός της λύσης αυτής.

Το πρόβλημα της οικονομικής κατανομής του φορτίου, είναι ένα πρόβλημα στιγμιαίας βελτιστοποίησης και έτσι, τα δεδομένα και ζητούμενα του προβλήματος, αναφέρονται σε μια δεδομένη στιγμή της λειτουργίας του συστήματος. Για να επιλυθεί το πρόβλημα αυτό, χρειάζεται να οριστούν έννοιες όπως η βελτιστοποίηση συναρτήσεων, η ελαχιστοποίηση με εξισωτικούς και ανισωτικούς περιορισμούς και τέλος να γίνει η επίλυση του προβλήματος.

## 3.2. Βελτιστοποίηση συναρτήσεων

Στα μαθηματικά, ένα πρόβλημα βελτιστοποίησης, συναντάται στην ελαχιστοποίηση ή μεγιστοποίηση μιας συνάρτησης,  $f = (x_1, \dots, x_N)$ , η οποία ονομάζεται αντικειμενική συνάρτηση ή συνάρτηση στόχου. Η επίλυση της, γίνεται με τον καθορισμό του διανύσματος  $x = [x_1, \dots, x_N]$ , που επιστρέφει αντίστοιχα την ελάχιστη ή μέγιστη τιμή της συνάρτησης  $f$ . Η μεταβλητή  $x_i$ , μπορεί να πάρει οποιεσδήποτε τιμές, όταν μιλάμε για ένα πρόβλημα βελτιστοποίησης, χωρίς περιορισμούς, ενώ μπορεί να περιοριστεί σε ορισμένες επιτρεπτές, για ένα πρόβλημα βελτιστοποίησης, με περιορισμούς. Ένα πρόβλημα βελτιστοποίησης, στην απλή του μορφή, θα περιέχει μόνο μια μεταβλητή  $x$ , ενώ στην σύνθετη του μορφή μπορεί να εμπλέκει περισσότερες από μια, μεταβλητές  $x$ . Στην παρούσα εργασία, παρουσιάζονται προβλήματα ελαχιστοποίησης.

### 3.2.1. Ελαχιστοποίηση συναρτήσεων μιας μεταβλητής, χωρίς περιορισμούς

Σε ένα σημείο  $x^*$ , η συνάρτηση  $f(x)$  παρουσιάζει ολικό ελάχιστο, αν για όλα τα  $x$  στο πεδίο ορισμού της ισχύει:

$$f(x^*) \leq f(x) \quad (3.1)$$

Είναι επίσης γνωστό, ότι μια συνάρτηση  $f(x)$ , παρουσιάζει τοπικό ακρότατο στα άκρα του πεδίου ορισμού της, αλλά και στα κρίσιμα σημεία, δηλαδή τα σημεία στα οποία δεν ορίζεται η πρώτη παράγωγος της συνάρτησης. Ακόμα, είναι γνωστό ότι, αν η  $f(x)$  είναι συνεχής συνάρτηση και έχει συνεχείς πρώτες και δεύτερες παραγώγους για τις τιμές του  $x$ , γύρω από το  $x^*$ , το  $x^*$  είναι αυστηρά τοπικό ελάχιστο της  $f(x)$ , αν ισχύουν οι ακόλουθες συνθήκες:

$$\frac{df(x^*)}{dx} = 0 \quad (3.2)$$

$$\frac{d^2f(x^*)}{dx^2} > 0 \quad (3.3)$$

Ένα απλό πρόβλημα ελαχιστοποίησης, με αντικειμενική συνάρτηση, την συνάρτηση  $f(x)$ , γράφεται ως εξής:

$$\text{Min } f(x) \quad (3.4)$$

Ο υπολογισμός της τιμής  $x^*$ , η οποία ικανοποιεί την (3.2), μπορεί να γίνει και με την χρήση μιας επαναληπτικής μεθόδου, όπως η Newton, ξεκινώντας από κάποια αρχική τιμή  $x^0$  για το  $x$ .

$$x^{k+1} = x^k - \left( \frac{d^2f(x^k)}{dx^2} \right)^{-1} \frac{df(x^k)}{dx}, k = 0, 1, 2, \dots \quad (3.5)$$

Αρκετές φορές η επίλυση της εξίσωσης (3.2), παρουσιάζει δυσκολίες όταν η συνάρτηση είναι πολύπλοκη. Υπάρχουν περιπτώσεις, που αν επιλυθεί με την μέθοδο των επαναλήψεων, ανάλογα την αρχική επιλογή του  $x^0$ , το αποτέλεσμα το οποίο παίρνουμε, θα είναι απλά ένα στάσιμο σημείο της συνάρτησης, κάτι το οποίο, δεν εγγυάται απαραίτητα την επίλυση του προβλήματος ελαχιστοποίησης (3.4). Για τον λόγο αυτό κρίνεται απαραίτητο να ελεγχθεί και η συνθήκη (3.3).

Η αριθμητική αυτή μέθοδος, ονομάζεται και μέθοδος κλίσεων η οποία εντοπίζει ένα τοπικό ελάχιστο της συνάρτησης  $f(x)$ . Η μέθοδος ξεκινάει από μια αρχική τιμή  $x^0$ , υπολογίζει την κλίση της,  $\frac{df(x^0)}{dx}$ , και έπειτα προχωράει κατά την αντίθετη κατεύθυνση η οποία έχει αρνητική κλίση,  $-\frac{df(x^0)}{dx}$  (άρα και μείωση της τιμής της), έως ότου να ικανοποιηθεί κάποιο κριτήριο τερματισμού. Ως κριτήριο τερματισμού, συνήθως λαμβάνεται η σχέση (3.6), στην οποία ο αριθμός  $\varepsilon$ , είναι ένας μικρός θετικός αριθμός. Με άλλα λόγια η διαδικασία τερματίζει όταν σχεδόν μηδενίζεται η πρώτη παράγωγος.



$$\left| \frac{df(x^k)}{dx} \right| < \varepsilon \quad (3.6)$$

### 3.2.2. Ελαχιστοποίηση συναρτήσεων πολλών μεταβλητών, χωρίς περιορισμούς

Στην παράγραφο 3.2.1, περιεγράφηκε η ελαχιστοποίηση συναρτήσεων για μια μεταβλητή, χωρίς περιορισμούς. Αν γενικευθούν τα παραπάνω, θα έχουμε ως αποτέλεσμα, την ελαχιστοποίηση συνάρτησης πολλών μεταβλητών,  $f(x_1, \dots, x_N)$ . Επίσης, χρειάζεται να θεωρηθεί η συνάρτηση, ως συνάρτηση διανύσματος,  $f(x)$ , με  $x = [x_1, \dots, x_N]$ .

Στην σχέση (3.7), ορίζεται η πρώτη παράγωγος ή αλλιώς κλίση της συνάρτησης  $f(x)$ , ενώ στην (3.8) ορίζεται η δεύτερη παράγωγος της  $f(x)$ , ή αλλιώς πίνακας Hess,  $H_f(x)$ :

$$\nabla_x f(x) = \frac{df(x)}{dx} = \left[ \frac{\partial f(x)}{\partial x_1} \dots \frac{\partial f(x)}{\partial x_N} \right] \quad (3.7)$$

και,

$$H_f(x) = \frac{d^2 f(x)}{dx^2} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & & \ddots & \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & & & \frac{\partial^2 f}{\partial x_N^2} \end{bmatrix} \quad (3.8)$$

Οι σχέσεις (3.9) και (3.10) είναι οι αντίστοιχες σχέσεις (3.2) και (3.3) της παραγράφου 3.2.1. Το διάνυσμα  $x^* = [x_1, \dots, x_N]$ , είναι τοπικό ελάχιστο της  $f(x) = f(x_1, \dots, x_N)$  όταν:

$$\nabla_x f(x) = 0_N \text{ ή } \frac{\partial f(x_1, \dots, x_N)}{\partial x_i} = 0 \quad (3.9)$$

και

$$H_f(x), \text{ θετικά ορισμένος πίνακας} \quad (3.10)$$

Όπου το  $0_N = [00 \dots 0]^T$ , είναι ένα διάνυσμα με  $N$ , μηδενικά στοιχεία.

Όμοια, η εξίσωση (3.7) μπορεί να επιλυθεί με την μέθοδο επαναλήψεων Newton, με σκοπό την εύρεση του στάσιμου σημείου, έχοντας ως  $x^0$ , ένα αρχικά επιλεγμένο διάνυσμα. Επειδή το σημείο το οποίο προκύπτει από την μέθοδο είναι στάσιμο, κρίνεται και εδώ απαραίτητο να ελεγχθεί η συνθήκη (3.10). Παρακάτω παρουσιάζεται η εξίσωση (3.11), ισοδύναμη της (3.5):

$$x^{k+1} = x^k - \nabla_x f(x^k)^T \cdot H_f(x^k)^{-1}, k = 0, 1, 2, \dots \quad (3.11)$$

Αφού βρεθεί το τοπικό ελάχιστο, μέσω της μεθόδου κλίσεων όπου περιεγράφηκε νωρίτερα, η μέθοδος τερματίζει με την σχέση (3.12) ως κριτήριο τερματισμού.

$$\|\nabla_x f(x^k)\| < \varepsilon \quad (3.12)$$

Ως  $\varepsilon$ , ορίζεται ένας μικρός θετικός αριθμός και ως  $\|\nabla_x f(x^k)\|$  μια νόρμα του διανύσματος  $\nabla f = \left[ \frac{\partial f(x)}{\partial x_1} \dots \frac{\partial f(x)}{\partial x_N} \right]$ , δηλαδή  $\|\nabla f\| = \frac{1}{N} \sum_{i=1}^N \left| \frac{\partial f}{\partial x_i} \right|$ .

### 3.3. Ελαχιστοποίηση συνάρτησης με εξισωτικούς περιορισμούς

Ένα πρόβλημα ελαχιστοποίησης, μπορεί να έχει από έναν έως και πολλούς εξισωτικούς περιορισμούς. Ασχέτως την ποσότητα των περιορισμών, ο τρόπος επίλυσης τέτοιων προβλημάτων είναι, με την χρήση της συνάρτησης Lagrange. Η συνάρτηση Lagrange, ενσωματώνει στην αντικειμενική συνάρτηση, τους εκάστοτε

περιορισμούς και βρίσκει το σημείο στο οποίο παρουσιάζει την μέγιστη ή την ελάχιστη τιμή.

### 3.3.1. Ελαχιστοποίηση με έναν εξισωτικό περιορισμό

Ένα πρόβλημα ελαχιστοποίησης, με έναν μόνο εξισωτικό περιορισμό, μπορεί να έχει την ακόλουθη μορφή:

$$\text{Min:} \quad f(x_1, x_2, \dots, x_N) \quad (3.13)$$

$$\text{Με περιορισμό:} \quad \varphi(x_1, x_2, \dots, x_N) = 0 \quad (3.14)$$

Για την επίλυση του προβλήματος, χρειάζεται να σχηματιστεί η συνάρτηση Lagrange. Για τον σχηματισμό της, αρχικά πολλαπλασιάζεται ο περιορισμός (3.14) με τον συντελεστή – πολλαπλασιαστή Lagrange  $\lambda$ . Στην συνέχεια προστίθεται ο πολλαπλασιασμένος, με τον συντελεστή  $\lambda$ , περιορισμός στην αντικειμενική συνάρτηση (3.13) και προκύπτει η συνάρτηση Lagrange.

Ο πολλαπλασιαστής  $\lambda$ , μπορεί να ερμηνευθεί ως εξής. Αν  $f$ , είναι το κόστος μια διαδικασίας, για την παραγωγή μιας ποσότητας αγαθού, ο πολλαπλασιαστής  $\lambda$  εκφράζει την αύξηση του ελάχιστου κόστους, κατά την μοναδιαία αύξηση της ποσότητας αγαθού.

$$\mathcal{L}(x_1, x_2, \dots, x_N) = f(x_1, x_2, \dots, x_N) + \lambda \cdot \varphi(x_1, x_2, \dots, x_N) \quad (3.15)$$

Η συνάρτηση  $N+1$  μεταβλητών,  $\mathcal{L}(x_1, x_2, \dots, x_N, \lambda)$ , χωρίς περιορισμούς, περιγράφει το πρόβλημα, με έναν περιορισμό, των σχέσεων (3.13) και (3.14). Το ελάχιστο,  $(x_1^*, x_2^*, \dots, x_N^*)$ , της συνάρτησης είναι και η βέλτιστη λύση του προβλήματος. Αν υποτεθεί ότι, οι συναρτήσεις  $f$  και  $\varphi$ , είναι συνεχείς και παραγωγίσιμες, οι αναγκαίες συνθήκες ελαχίστου της (3.15) είναι:

$$\frac{\partial \mathcal{L}}{\partial x_i} = 0, \quad i = 1, \dots, N \quad (3.16)$$

και

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \quad (3.17)$$

Έτσι έχουμε:

$$\frac{\partial f}{\partial x_i} + \lambda \frac{\partial \varphi}{\partial x_i} = 0, \quad i = 1, \dots, N \quad (3.18)$$

και

$$\varphi(x_1, x_2, \dots, x_N) = 0 \quad (3.19)$$

### 3.3.2. Ελαχιστοποίηση με πολλούς εξισωτικούς περιορισμούς

Ένα πρόβλημα ελαχιστοποίησης, με πάνω από έναν εξισωτικό περιορισμό, μπορεί να έχει την ακόλουθη μορφή:

$$\text{Min:} \quad f(x_1, x_2, \dots, x_N) \quad (3.20)$$

$$\begin{aligned} \text{Με περιορισμούς:} \quad & \varphi_1(x_1, x_2, \dots, x_N) = 0 \\ & \varphi_2(x_1, x_2, \dots, x_N) = 0 \\ & \vdots \\ & \varphi_M(x_1, x_2, \dots, x_N) = 0 \end{aligned} \quad (3.21)$$

Με αντίστοιχο τρόπο και εδώ, σχηματίζεται η συνάρτηση Lagrange, εισάγοντας της, έναν πολλαπλασιαστή Lagrange, για κάθε έναν από τους περιορισμούς. Έτσι έχουμε:

$$\begin{aligned} \mathcal{L}(x_1, x_2, \dots, x_N) = & f(x_1, x_2, \dots, x_N) + \lambda_1 \cdot \varphi_1(x_1, x_2, \dots, x_N) \\ & + \lambda_2 \cdot \varphi_2(x_1, x_2, \dots, x_N) + \dots + \lambda_M \cdot \varphi_M(x_1, x_2, \dots, x_N) \end{aligned} \quad (3.22)$$

Αν υποθεθεί ότι, οι συναρτήσεις  $f$  και  $\varphi_i$ , είναι συνεχείς και παραγωγίσιμες, οι αναγκαίες συνθήκες ελαχιστοποίησης, για την χωρίς περιορισμούς σχέση (3.22), είναι:

$$\frac{\partial \mathcal{L}}{\partial x_i} = 0, \quad i = 1, \dots, N \quad (3.23)$$

και

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = 0, \quad i = 1, \dots, M \quad (3.24)$$

Έτσι έχουμε:

$$\frac{\partial f(x_1, x_2, \dots, x_N)}{\partial x_i} + \sum_{j=1}^M \lambda_j \frac{\partial \varphi_j(x_1, x_2, \dots, x_N)}{\partial x_i} = 0, i = 1, \dots, N \quad (3.25)$$

και

$$\varphi_j(x_1, x_2, \dots, x_N) = 0, \quad j = 1, \dots, M \quad (3.26)$$

Γενικεύοντας τα παραπάνω, ένα πρόβλημα ελαχιστοποίησης μπορεί να έχει την εξής μορφή:

$$\text{Min:} \quad f(x) \quad (3.27)$$

$$\text{Με περιορισμό:} \quad \varphi(x) = 0_M^T \quad (3.28)$$

Όπου το  $x = [x_1, x_2, \dots, x_N]$  είναι ένα  $N$ -διάστατο διάνυσμα, το  $\varphi(x) = [\varphi_1(x), \varphi_2(x), \dots, \varphi_M(x)]^T$ , είναι ένα  $M$ -διάστατο διάνυσμα του  $N$ -διάστατου διανύσματος  $x$  και το  $0_M^T$  είναι ένα  $M$ -διάστατο διάνυσμα. Εισάγοντας τον συντελεστή Lagrange, σχηματίζεται η συνάρτηση Lagrange:

$$\mathcal{L}(x, \lambda) = f(x) + \vec{\lambda} \cdot \varphi(x) \quad (3.29)$$

Της οποίας οι αναγκαίες συνθήκες για την εύρεση του βέλτιστου είναι:

$$\nabla_x f(x) + \lambda \nabla_x \varphi(x) = 0_N^T \quad (3.30)$$

$$\varphi(x) = 0_M^T \quad (3.31)$$

$$\nabla_x f(x) = \left[ \frac{\partial f(x)}{\partial x_1} \dots \frac{\partial f(x)}{\partial x_N} \right] \quad (3.32)$$

και

$$\nabla_x \varphi(x) = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_1} & \dots & \frac{\partial \varphi_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial \varphi_M}{\partial x_1} & \dots & \frac{\partial \varphi_M}{\partial x_N} \end{bmatrix} \quad (3.33)$$

Χάρη στις παραπάνω σχέσεις, μπορεί να βρεθεί ένα στάσιμο σημείο της συνάρτησης Lagrange,  $\mathcal{L}(x, \lambda)$ , στο οποίο η συνάρτηση παίρνει, την ελάχιστη τιμή της, ως προς το  $x$ , ενώ την μέγιστη της, ως προς τον πολλαπλασιαστή  $\lambda$ .

### 3.4. Ελαχιστοποίηση συνάρτησης με ανισωτικούς περιορισμούς

Ένα πρόβλημα ελαχιστοποίησης με ανισωτικούς περιορισμούς είναι της ακόλουθης μορφής:

$$\text{Min:} \quad f(x) \quad (3.34)$$

$$\text{Με περιορισμούς:} \quad \varphi_i(x) = 0, \quad i = 1, \dots, N_\varphi \quad (3.35)$$

$$\omega_i(x) \leq 0, \quad i = 1, \dots, N_\omega \quad (3.36)$$

$$x = [x_1, \dots, x_N]$$

Για την επίλυση του, σχηματίζεται πάλι η συνάρτηση Lagrange, εισάγοντας έναν πολλαπλασιαστή,  $\lambda_i$ , για κάθε εξισωτικό περιορισμό, ενώ για κάθε ανισωτικό, εισάγεται ένας πολλαπλασιαστής  $\mu_i \geq 0$ . Έτσι η συνάρτηση  $\mathcal{L}(x, \lambda, \mu)$ , γίνεται:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{N_\varphi} \lambda_i \varphi_i(x) + \sum_{i=1}^{N_\omega} \mu_i \omega_i(x) \quad (3.37)$$

Η εύρεση της βέλτιστης λύσης θα γίνει με χρήση των συνθηκών Kuhn – Tucker (K-T). Οι συνθήκες αυτές, επιτρέπουν τον υπολογισμό των πολλαπλασιαστών Lagrange. Στην προκειμένη περίπτωση, οι αναγκαίες για την βέλτιστη λύση  $x^*$ ,  $\lambda^*$ ,  $\mu^*$ , K-T συνθήκες είναι:

$$\begin{array}{ll} \text{i.} & \frac{\partial \mathcal{L}}{\partial x_i}(x^*, \lambda^*, \mu^*) = 0 \quad \text{για } i = 1, \dots, N \\ \text{ii.} & \varphi_i(x^*) = 0 \quad \text{για } i = 1, \dots, N_\varphi \\ \text{iii.} & \omega_i(x^*) \leq 0 \quad \text{για } i = 1, \dots, N_\omega \\ \text{iv.} & \left. \begin{array}{l} \mu_i^* \cdot \omega_i(x^*) = 0 \\ \mu_i^* \geq 0 \end{array} \right\} \quad \text{για } i = 1, \dots, N_\omega \end{array}$$

Η πρώτη συνθήκη δηλώνει ότι, η κλίση της συνάρτησης  $\mathcal{L}$ , ως προς τις αρχικές μεταβλητές  $\nabla_x f$ , πρέπει να είναι μηδέν. Η δεύτερη συνθήκη, ότι είναι αναγκαίο να ικανοποιείται ο περιορισμός (3.35), ενώ η τρίτη, να ικανοποιείται ο περιορισμός (3.36). Τέλος, η τέταρτη συνθήκη δηλώνει ότι, πρέπει τουλάχιστον ένα από τα  $\mu_i^*$  και  $\omega_i(x^*)$  να είναι ίσο με το μηδέν κατά την βέλτιστη λύση. Ειδικότερα, αν στην βέλτιστη λύση, ο περιορισμός  $\omega_i(x^*) \leq 0$  ικανοποιείται οριακά σαν ισότητα, δηλαδή  $\omega_i(x^*) = 0$ , τότε αναγκαστικά,  $\mu_i^* > 0$ , δηλαδή το  $\mu_i^*$  παίρνει μια θετική τιμή. Αντίστοιχα, όταν στην βέλτιστη λύση ο περιορισμός  $\omega_i(x^*) \leq 0$ , γίνεται  $\omega_i(x^*) < 0$ , τότε αναγκαστικά  $\mu_i^* = 0$ . Ουσιαστικά η τέταρτη συνθήκη, εκφράζει τη συμπληρωματική απόκλιση μεταξύ των ανισωτικών περιορισμών και των πολλαπλασιαστών  $\mu_i$ .

Τέλος, ο συντελεστής  $\mu_i^*$ , εκφράζει την μείωση που επιτυγχάνεται στο ελάχιστο κόστος, κατά την μοναδιαία αύξηση της διαθέσιμης ποσότητας του πόρου  $i$ . Αλλιώς, μπορεί να ερμηνευθεί σαν μοναδιαία – οριακή αξία του πόρου  $i$ , ή σαν σκιά τιμή του  $i$ .

### 3.5. Πρόβλημα οικονομικής κατανομής φορτίου

Στις παραγράφους 3.2, 3.3 και 3.4, παρουσιάστηκε η θεωρία, πίσω από την επίλυση του προβλήματος οικονομικής κατανομής. Ένα πρόβλημα οικονομικής κατανομής μπορεί να επιλυθεί, ως ένα πρόβλημα βελτιστοποίησης με περιορισμούς. Συγκεκριμένα, το πρόβλημα στην απλή του μορφή, περιγράφεται από την αντικειμενική συνάρτηση, δηλαδή την ελαχιστοποίηση του κόστους λειτουργίας των θερμικών μονάδων παραγωγής ενέργειας,  $F_T$ , υπό τον περιορισμό ότι η ζήτηση,  $P_R$ , θα καλύπτεται από την παραγόμενη ισχύ των μονάδων,  $P_i$ .

Προφανώς, σε ένα διασυνδεδεμένο δίκτυο, η ζήτηση δεν καλύπτεται αποκλειστικά από θερμικές μονάδες, αλλά από υδροηλεκτρικούς σταθμούς, από ΑΠΕ και από ανταλλαγές ενέργειας με τα γειτονικά συνδεδεμένα δίκτυα. Στην εργασία αυτή, θα θεωρηθεί ότι η ενέργεια παράγεται αποκλειστικά από θερμικές μονάδες.

#### 3.5.1. Η απλούστερη μορφή του προβλήματος οικονομικής κατανομής

Παρακάτω παρουσιάζεται το πρόβλημα οικονομικής κατανομής, στην πιο απλή του μορφή. Στην ουσία, δεν λαμβάνονται υπόψιν, τα λειτουργικά όρια των μονάδων και οι απώλειες ενέργειας λόγω μεταφοράς. Έτσι το πρόβλημα μεταφράζεται ως εξής:

$$\begin{aligned} \text{Min: } F_T(P_1, P_2, \dots, P_N) \\ &= F_1(P_1) + F_2(P_2) + \dots + F_N(P_N) \\ &= \sum_{i=1}^N F_i(P_i) \end{aligned} \quad (3.38)$$

$$\text{Με περιορισμό: } \Phi(P_1, P_2, \dots, P_N) = P_R - \sum_{i=1}^N P_i = 0 \quad (3.39)$$

Όπου  $P_R$  δηλώνεται η ζητούμενη ενέργεια και  $P_i$  η παραγόμενη ισχύς των μονάδων.



Η σχέση (3.39), ονομάζεται και εξίσωση διατήρησης της ισχύος. Επιπλέον, τα  $F_i$ , είναι συναρτήσεις που εκφράζουν το ωριαίο κόστος λειτουργίας των μονάδων και τα  $F_i(P_i)$ , εκφράζουν το κόστος λειτουργίας μιας μονάδας  $i$ , για ισχύ λειτουργίας  $P_i$ . Τα  $F_i(P_i)$  είναι της μορφής:  $a_i P_i^2 + b_i P_i + c_i$  και τα  $a_i, b_i, c_i$ , είναι οι συντελεστές κόστους της μονάδας  $i$ .

Το παραπάνω πρόβλημα μπορεί να επιλυθεί με την συνάρτηση Lagrange, καθώς είναι ένα πρόβλημα ελαχιστοποίησης, με έναν εξισωτικό περιορισμό. Αρχικά, πολλαπλασιάζεται ο περιορισμός με τον πολλαπλασιαστή Lagrange,  $\lambda$ , και είναι:

$$\begin{aligned} \mathcal{L}(P_1, P_2, \dots, P_N, \lambda) \\ = F_T(P_1, P_2, \dots, P_N) + \lambda \cdot \Phi(P_1, P_2, \dots, P_N) \\ = \sum_{i=1}^N F_i(P_i) + \lambda \cdot (P_R - \sum_{i=1}^N P_i) \end{aligned} \quad (3.40)$$

Η απαραίτητη συνθήκη βελτίστου, της συνάρτησης (3.40), είναι ο μηδενισμός των μερικών παραγώγων της συνάρτησης, ως προς τις μεταβλητές  $P_i$  και  $\lambda$ . Η μερική παράγωγος της  $\mathcal{L}$ , ως προς τον πολλαπλασιαστή  $\lambda$ , δίνει τον περιορισμό, ενώ των μεταβλητών  $P_i$  τα ακόλουθα:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial P_i} = \frac{\partial F_i(P_i)}{\partial P_i} - \lambda = 0 \\ \text{ή} \\ \frac{\partial F_i}{\partial P_i} = \lambda \end{aligned} \quad (3.41)$$

Οι εξισώσεις της σχέσης (3.41), ονομάζονται εξισώσεις συνεργασίας και αποτελούν απαραίτητη συνθήκη για την ελαχιστοποίηση του συνολικού κόστους λειτουργίας των μονάδων. Οι εξισώσεις συνεργασίας εκφράζουν ότι, οι μονάδες πρέπει να λειτουργούν με το ίδιο διαφορικό κόστος λειτουργίας. Το διαφορικό αυτό κόστος, είναι ίσο με τον πολλαπλασιαστή Lagrange,  $\lambda$ . Η οικονομική του σημασία για το πρόβλημα οικονομικής κατανομής είναι, η ποσοτική αύξηση του ωριαίου κόστους λειτουργίας των μονάδων, με την αύξηση του φορτίου του συστήματος κατά ένα kW. Μονάδα μέτρησης του  $\lambda$ , είναι το €/h.

Συνεπώς, η επίλυση του προβλήματος οικονομικής κατανομής, γίνεται με χρήση της εξίσωσης διατήρησης ισχύος και των εξισώσεων συνεργασίας. Παρακάτω, παρουσιάζεται η επίλυση του προβλήματος οικονομικής κατανομής, με βάση όσα αναφέρθηκαν παραπάνω.

### Παράδειγμα 1

Έστω, μια μικρή εταιρεία παραγωγής ενέργειας, της οποίας το σύστημα παραγωγής, αποτελείται από τρεις θερμικές μονάδες. Χρειάζεται να υπολογιστεί το οικονομικό σημείο λειτουργίας των μονάδων, όταν η ζήτηση του φορτίου είναι 900 MW.

#### Μονάδα 1:

Μέγιστη Έξοδος:

$$P_{1\max} = 600 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{1\min} = 150 \text{ MW}$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_1 \left( \frac{\text{Gcal}}{h} \right) = 165.8 + 2.321P_1 + 0.0005P_1^2$$

Καύσιμο: Φυσικό αέριο

Κόστος καυσίμου: 3.48 k€/Gcal

#### Μονάδα 2:

Μέγιστη Έξοδος:

$$P_{2\max} = 400 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{2\min} = 100 \text{ MW}$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_2 \left( \frac{\text{Gcal}}{h} \right) = 78.3 + 2.018P_2 + 0.0005P_2^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

#### Μονάδα 3:

Μέγιστη Έξοδος:

$$P_{3\max} = 200 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{3\min} = 50 \text{ MW}$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_3 \left( \frac{\text{Gcal}}{h} \right) = 39.4 + 2.048P_3 + 0.0012P_3^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

Αρχικά, υπολογίζονται οι καμπύλες ωριαίου κόστους λειτουργίας των μονάδων.

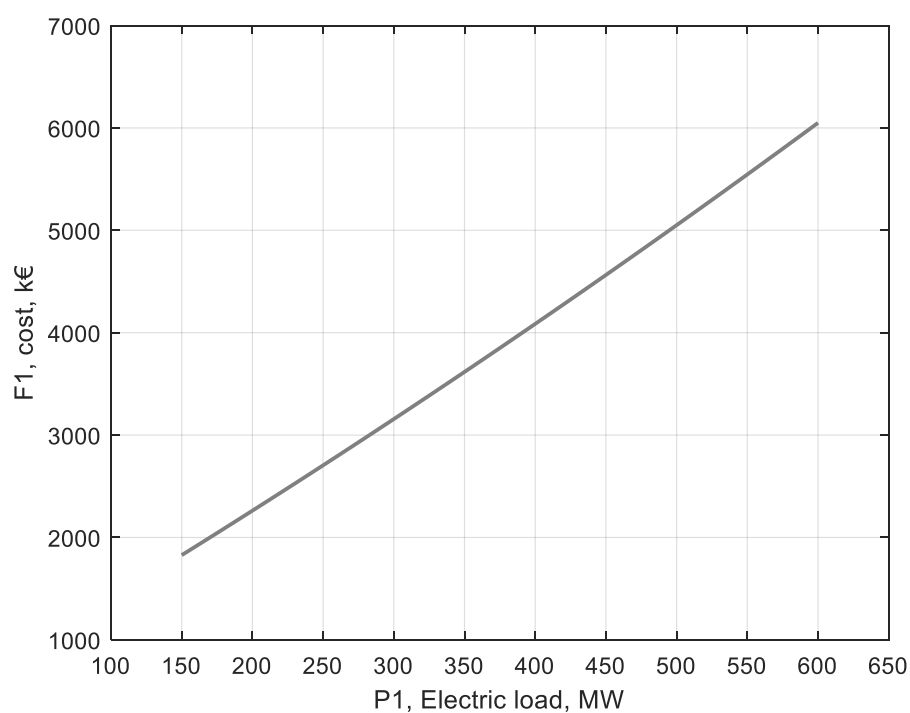
Είναι:

$$F_1(P_1) = H_1(P_1) * 3.48 = 577 + 8.078P_1 + 0.00173P_1^2 \quad \text{k€/h}$$

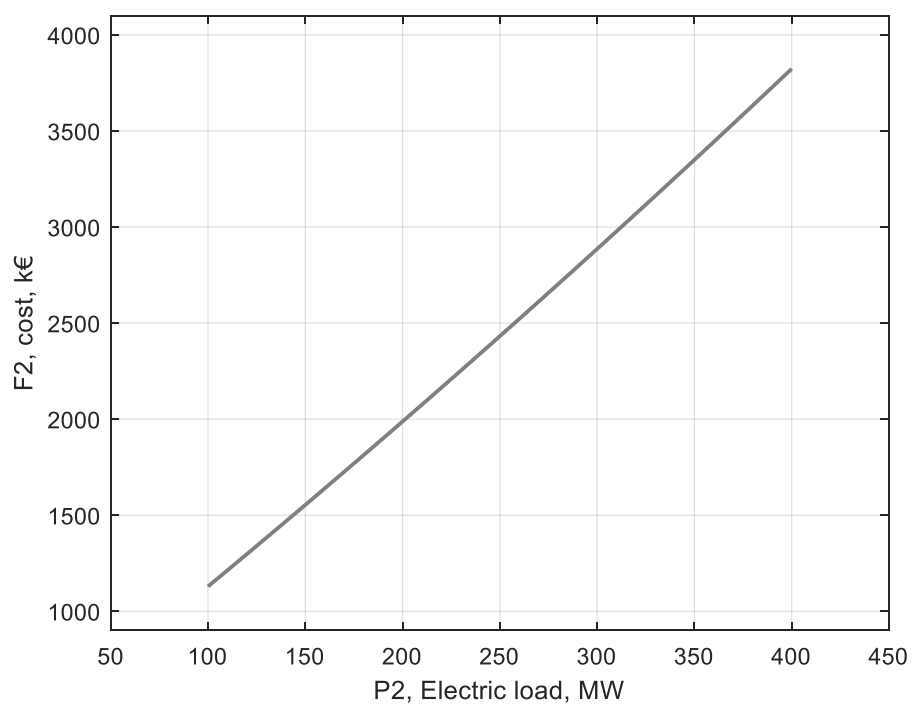
$$F_2(P_2) = H_2(P_2) * 3.96 = 310 + 7.992P_2 + 0.00200P_2^2 \quad \text{k€/h}$$

$$F_3(P_3) = H_3(P_3) * 3.96 = 156 + 8.110P_3 + 0.00474P_3^2 \quad \text{k€/h}$$

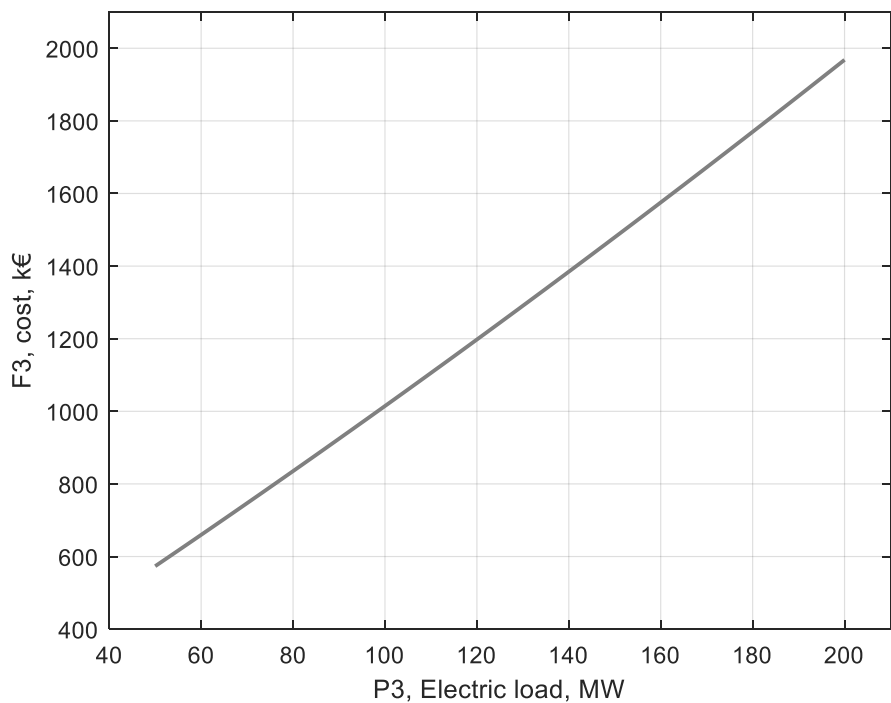
Ενδιαφέρον παρουσιάζει, η γραφική απεικόνιση των καμπυλών κόστους λειτουργίας των μονάδων. Ειδικότερα είναι:



Διάγραμμα 3.1: Καμπύλη ωριαίου κόστους, της πρώτης μονάδας που χρησιμοποιεί καύσιμο φυσικό αέριο



Διάγραμμα 3.2: Καμπύλη ωριαίου κόστους, της δεύτερης μονάδας που χρησιμοποιεί καύσιμο μαζούτ



Διάγραμμα 3.3: Καμπύλη ωριαίου κόστους, της τρίτης μονάδας που χρησιμοποιεί καύσιμο μαζούτ

Παρατηρείται πως και για τις τρεις μονάδες, οι καμπύλες ωριαίου κόστους λειτουργίας, έχουν σχεδόν γραμμική μορφή. Αυτό οφείλεται στον μικρό συντελεστή που έχουν οι εξισώσεις, στο τετράγωνο της ισχύος της κάθε μονάδας.

Έτσι, οι εξισώσεις συνεργασίας (3.41) γίνονται:

$$\begin{aligned}\frac{dF_1}{dP_1} &= 8.078 + 0.00346P_1 = \lambda \\ \frac{dF_2}{dP_2} &= 7.992 + 0.00400P_2 = \lambda \\ \frac{dF_3}{dP_3} &= 8.110 + 0.00948P_3 = \lambda\end{aligned}$$

και η εξίσωση διατήρησης της ισχύος (3.39), γίνεται:

$$P_1 + P_2 + P_3 = 900 \text{ MW}$$

Το παραπάνω σύστημα, των τεσσάρων γραμμικών εξισώσεων, λύνεται εύκολα ως προς τις μεταβλητές  $P_1$ ,  $P_2$ ,  $P_3$  και  $\lambda$  αντίστοιχα. Αρχικά, χρησιμοποιούνται οι εξισώσεις συνεργασίας, από τις οποίες υπολογίζονται τα  $P_1$ ,  $P_2$ ,  $P_3$ , συναρτήσει του  $\lambda$  και αντικαθίστανται στην εξίσωση διατήρησης της ισχύος, η οποία λύνεται ως προς την μεταβλητή  $\lambda$ . Έτσι, είναι:

$$\begin{aligned}P_1 &= \frac{\lambda - 8.078}{0.00346} \\ P_2 &= \frac{\lambda - 7.992}{0.00400} \\ P_3 &= \frac{\lambda - 8.110}{0.00948}\end{aligned}$$

και

$$\frac{\lambda - 8.078}{0.00346} + \frac{\lambda - 7.992}{0.00400} + \frac{\lambda - 8.110}{0.00948} = 900$$

Άρα έχουμε:

$$\lambda = 9.446 \frac{k\text{€}}{MW} \text{ ή } \lambda = 9.446 \frac{\text{€}}{kW}$$

και

$$P_1 = 395.5 MW$$

$$P_2 = 363.5 MW$$

$$P_3 = 141 MW$$

Παρατηρείται ότι, οι μονάδες λειτουργούν με το ίδιο διαφορικό κόστος λειτουργίας,  $\lambda = 9.446 \frac{\text{€}}{kW}$ , το οποίο εκφράζει το κόστος παραγωγής μιας πρόσθετης κιλοβατώρας από το σύστημα. Οι εξόδοι ισχύος των μονάδων, είναι εντός των ορίων λειτουργίας της και το άθροισμα του παραγόμενου φορτίου ισούται με το απαιτούμενο, των 900 MW.

### 3.5.2. Οικονομική κατανομή φορτίου και λειτουργικά όρια μονάδων

Για να γίνει πιο ρεαλιστική η επίλυση του προβλήματος οικονομικής κατανομής φορτίου, χρειάζεται να ληφθούν υπόψη τα λειτουργικά όρια των μονάδων. Ειδικότερα, το πρόβλημα σε αυτή του την μορφή, περιγράφεται από την αντικειμενική συνάρτηση, δηλαδή την ελαχιστοποίηση του συνολικού κόστους λειτουργίας,  $F_T$ , υπό τον εξισωτικό περιορισμό ότι η ζήτηση,  $P_R$ , θα καλύπτεται από την παραγόμενη ισχύ των μονάδων,  $P_i$  αλλά και ότι οι μονάδες, θα λειτουργούν εντός των ονομαστικών ορίων λειτουργίας της. Έτσι το πρόβλημα μεταφράζεται ως εξής:

$$\begin{aligned} \text{Min: } F_T(P_1, P_2, \dots, P_N) \\ &= F_1(P_1) + F_2(P_2) + \dots + F_N(P_N) \\ &= \sum_{i=1}^N F_i(P_i) \end{aligned} \quad (3.42)$$

$$\text{Με περιορισμούς: } \Phi(P_1, P_2, \dots, P_N) = P_R - \sum_{i=1}^N P_i = 0 \quad (3.43)$$

$$P_{i \min} \leq P_i \leq P_{i \max} \quad (3.44)$$

Το παραπάνω πρόβλημα, μπορεί να επιλυθεί με την χρήση των συνθηκών Kuhn – Tucker. Μετά από κάποια επεξεργασία των συνθηκών προκύπτει:

$$\begin{aligned} \frac{dF_i}{dP_i} &= \lambda && \text{για } P_{i \min} \leq P_i \leq P_{i \max} \\ \frac{dF_i}{dP_i} &\leq \lambda && \text{για } P_i = P_{i \max} \\ \frac{dF_i}{dP_i} &\geq \lambda && \text{για } P_i = P_{i \min} \end{aligned} \quad (3.45)$$

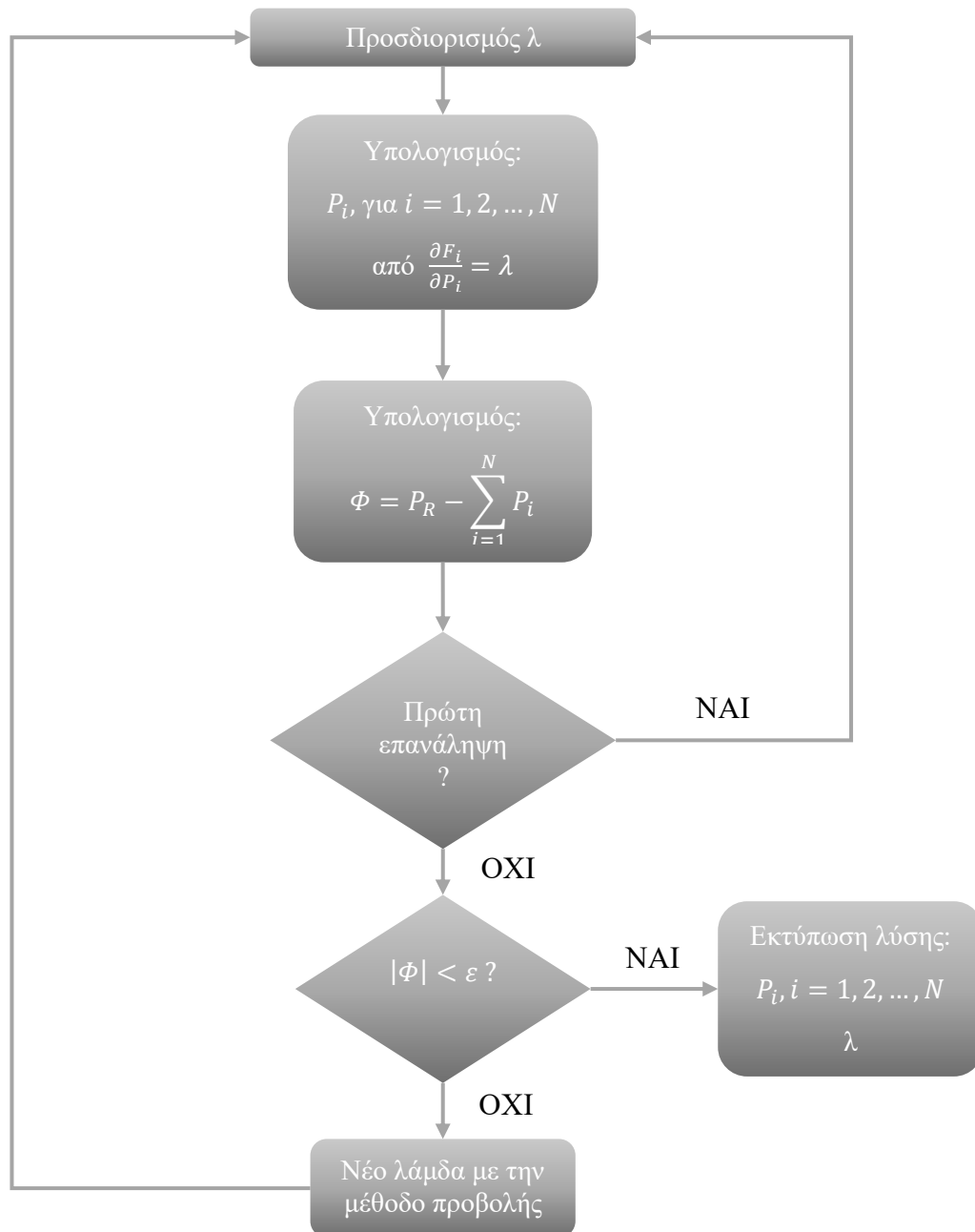
$$\sum_{i=1}^N P_i = P_R \quad (3.46)$$

$$P_{i \min} \leq P_i \leq P_{i \max} \quad (3.47)$$

Οι εξισώσεις (3.45 – 3.47) είναι οι συνθήκες βελτίστου του προβλήματος. Η (3.45), εκφράζει ότι αν μια μονάδα, λειτουργεί εντός των ορίων λειτουργίας της, τότε έχει διαφορετικό κόστος λειτουργίας ίσο με  $\lambda$ , ενώ αν λειτουργεί στο μέγιστο ή ελάχιστο του ονομαστικού ορίου λειτουργίας της, τότε έχει διαφορετικό κόστος λειτουργίας μικρότερο ή μεγαλύτερο ίσο του  $\lambda$ , αντίστοιχα. Με άλλα λόγια, αν η παραγόμενη ισχύς μιας μονάδας, για ένα συγκεκριμένο διαφορετικό κόστος λειτουργίας,  $\lambda$ , είναι μικρότερη από την ελάχιστη ονομαστική της ισχύ, την αναγκάζουμε να δουλέψει στο κάτω όριο της ισχύς της και αντίστοιχα αν είναι μεγαλύτερη από την μέγιστη ονομαστική της, την αναγκάζουμε να δουλέψει στο άνω όριο της ισχύς της. Η (3.46), αναφέρει ότι πρέπει να ισχύει η διατήρηση της ισχύος και η (3.47), εκφράζει τα όρια των μονάδων.

Το παραπάνω πρόβλημα, επιλύεται σε υπολογιστή, με επαναληπτικό αλγόριθμο. Στην προκειμένη περίπτωση, χρησιμοποιείται ο αλγόριθμος επαναλήψεων λάμδα. Επιλέγεται, αρχικά, ένας συντελεστής Lagrange,  $\lambda$  και υπολογίζονται τα  $P_i$ , από τις εξισώσεις συνεργασίας (3.41). Αν τα πλέον γνωστά  $P_i$ , δεν παραβιάζουν τα όρια λειτουργίας των μονάδων, συνεχίζεται ο αλγόριθμος, έως ότου η εξίσωση διατήρηση ισχύος  $\Phi = P_R - \sum_{i=1}^N P_i$ , έχει τιμή εντός κάποιου ορίου ανοχής  $\varepsilon$  (πχ.  $\varepsilon = 0.1 \text{ MW}$ ). Στην περίπτωση την οποία τα όρια λειτουργίας παραβιάζονται, τότε αν  $P_i < P_{i \min}$ ,  $P_i = P_{i \min}$  και αν  $P_i > P_{i \max}$ ,  $P_i = P_{i \max}$  και ο αλγόριθμος συνεχίζεται. Αν δεν ικανοποιείται η εξίσωση διατήρησης της ισχύος, επιλέγεται νέο  $\lambda$  ανάλογα την παραγόμενη ενέργεια. Αν είναι λιγότερη από την ζητούμενη, επιλέγεται μεγαλύτερο  $\lambda$ , ενώ αν είναι περισσότερη, επιλέγεται μικρότερο. Οι επαναλήψεις

συνεχίζουν μέχρι να επιτευχθεί σύγκλιση. Κάτι τέτοιο, σε προβλήματα με πολλές επαναλήψεις μπορεί να διαρκέσει, μεγάλο χρονικό διάστημα. Για τον λόγο αυτό, μετά την πρώτη επανάληψη, χρησιμοποιείται η μέθοδος προβολών, η οποία προέρχεται από την επαναληπτική μέθοδο Newton – Raphson. Στο ακόλουθο διάγραμμα, παρουσιάζεται ο αλγόριθμος επαναλήψεων λάμδα.



Διάγραμμα 3.4: Αλγόριθμος Επαναλήψεων Λάμδα.



Στην συνέχεια, παρουσιάζεται η λύση του προβλήματος οικονομικής κατανομής, λαμβάνοντας υπόψη τα λειτουργικά όρια των μονάδων.

## Παράδειγμα 2

Έστω το σύστημα του παραδείγματος 1. Υποθέτουμε ότι γίνεται μια αύξηση της τιμής του μαζούτ κατά 20%, οπότε η νέα τιμή καυσίμου για τις μονάδες 2 και 3 είναι:

$$3.96 * 1.2 = 4.752 \text{ k€/Gcal}$$

Χρειάζεται να υπολογιστεί το οικονομικό σημείο λειτουργίας των μονάδων, όταν η ζήτηση του φορτίου είναι 900 MW.

Οι καμπύλες ωριαίου κόστους των μονάδων είναι:

$$\begin{aligned} F_1(P_1) &= H_1(P_1) * 3.48 = 577 + 8.078P_1 + 0.00173P_1^2 & \text{k€/h} \\ F_2(P_2) &= H_2(P_2) * 4.752 = 372 + 9.590P_2 + 0.00240P_2^2 & \text{k€/h} \\ F_3(P_3) &= H_3(P_3) * 4.752 = 187 + 9.732P_3 + 0.00568P_3^2 & \text{k€/h} \end{aligned}$$

Έτσι, οι εξισώσεις συνεργασίας (3.41) γίνονται:

$$\begin{aligned} \frac{dF_1}{dP_1} &= 8.078 + 0.00346P_1 = \lambda \\ \frac{dF_2}{dP_2} &= 9.590 + 0.00480P_2 = \lambda \\ \frac{dF_3}{dP_3} &= 9.732 + 0.01136P_3 = \lambda \end{aligned}$$

και η εξίσωση διατήρησης της ισχύος (3.39), γίνεται:

$$P_1 + P_2 + P_3 = 900 \text{ MW}$$

Λύνοντας, το παραπάνω σύστημα όπως το παράδειγμα 1, το αποτέλεσμα το οποίο παίρνουμε είναι:

$$\lambda = 10.402 \frac{\text{€}}{\text{kW}}$$

$$P_1 = 671.8 \text{ MW}$$

$$P_2 = 169.2 \text{ MW}$$

$$P_3 = 59 \text{ MW}$$

Παρατηρείται, ότι παρόλο που οι παραπάνω τιμές ικανοποιούν την εξίσωση διατήρησης της ισχύος, η έξοδος ισχύος της μονάδας 1, είναι μεγαλύτερη από την μέγιστη ονομαστική ισχύ της μονάδας ( $P_{1 \max} = 600 \text{ MW}$ ). Αυτό, δεν είναι εφικτό, συνεπώς, για  $\lambda = 10,402 \frac{\text{€}}{\text{kW}}$ , η λύση γίνεται:

$$P_1 = \mathbf{600 \text{ MW}}$$

$$(\text{Στο άνω όριο με } \frac{dF_1}{dP_1} = 8.078 + 0.00346 \cdot P_1 = 10.154 < \lambda)$$

$$P_2 = 169.2 \text{ MW}$$

$$P_3 = 59 \text{ MW}$$

Ουσιαστικά, παραβίαση των ορίων λειτουργίας της πρώτης μονάδας προκύπτει για την τιμή του  $\lambda$  που βρήκαμε, δηλαδή 10,402. Για τον λόγο αυτό η μονάδα 1 θα λειτουργεί με μικρότερο  $\lambda$  από τις άλλες δύο. Με άλλα λόγια θα παράγει ενέργεια, της οποίας η μοναδιαία αύξηση του κόστους, θα είναι μικρότερη από την μοναδιαία αύξηση του κόστους των μονάδων 2 και 3.

Τώρα όμως,

$$P_1 + P_2 + P_3 = 828.2 < 900 \text{ MW}$$

Συνεπώς, πρέπει να αυξηθεί η παραγωγή προκειμένου να καλυφθεί η ζήτηση. Άρα χρειάζεται να αυξηθεί το  $\lambda$ . Έστω, ότι αυξάνεται κατά 10%. Γίνεται δηλαδή:

$$\lambda = 10.402 * 1.1 = 11.442 \frac{\text{€}}{\text{kW}}$$

και υπολογίζεται ότι:

$$\begin{aligned} P_1 &= 972.3 \text{ MW} \Rightarrow P_1 = \mathbf{600 \text{ MW}} \\ P_2 &= 385.8 \text{ MW} \\ P_3 &= 150.5 \text{ MW} \end{aligned}$$

με

$$P_1 + P_2 + P_3 = 1,136.3 > 900 \text{ MW}$$

Συνεπώς, η 10% αύξηση του  $\lambda$  (λάμδα), ήταν μεγάλη και χρειάζεται να ελαττωθεί. Η επιλογή του νέου  $\lambda$  (λάμδα), γίνεται με γραμμική παρεμβολή.

$$\Phi = P_R - (P_1 + P_2 + P_3)$$

$$\text{Για } \lambda_0 = 10.402$$

$$\Phi_0 = 900 - 828.2 = 71.8 \text{ MW}$$

$$\text{Για } \lambda_1 = 11.442$$

$$\Phi_1 = 900 - 1,136.3 = -236.3 \text{ MW}$$

Άρα ψάχνουμε την τιμή του  $\lambda_2$ , για την οποία  $\Phi_2 = 0$ . Αυτή είναι:

$$\begin{aligned} \frac{\lambda_2 - \lambda_1}{\lambda_1 - \lambda_0} &= \frac{\Phi_2 - \Phi_1}{\Phi_1 - \Phi_0} \rightarrow \lambda_2 = \lambda_1 - \frac{\lambda_1 - \lambda_0}{\Phi_1 - \Phi_0} \Phi_1 \\ &= 11.442 - \frac{11.442 - 10.402}{-236.3 - 71.8} (-236.3) = 10.64436 \end{aligned}$$

Για  $\lambda_2 = 10.64436 \frac{\text{€}}{\text{kW}}$ , υπολογίζουμε:

$$P_1 = 741.7 \text{ MW} \Rightarrow P_1 = \mathbf{600 \text{ MW}}$$

$$P_2 = 219.7 \text{ MW}$$

$$P_3 = 80.3 \text{ MW}$$

με

$$P_1 + P_2 + P_3 = 900 \text{ MW}$$

Άρα, ικανοποιείται ο περιορισμός φορτίου και δεν χρειάζονται άλλες επαναλήψεις. Συνοψίζοντας, η βέλτιστη λύση είναι:

$$\lambda = 10.64436 \frac{\text{€}}{\text{kW}}$$

$$P_1 = 600 \text{ MW}$$

$$P_2 = 219.7 \text{ MW}$$

$$P_3 = 80.3 \text{ MW}$$

με

$$\frac{dF_1}{dP_1} = 10.154 < \lambda,$$

$$\frac{dF_1}{dP_1} = \frac{dF_2}{dP_2} = 10.64436 = \lambda,$$

$$P_1 = P_{1 \max}$$

( $P_2, P_3$ , αυστηρά εντός ορίων)

Άρα ικανοποιούνται και οι συνθήκες βελτίστου Kuhn – Tucker.

### 3.5.3. Οικονομική κατανομή φορτίου και απώλειες μεταφοράς

Στην έως τώρα επίλυση του προβλήματος οικονομικής κατανομής, αμελήθηκαν οι απώλειες του συστήματος μεταφοράς. Κατά την μεταφορά ενέργειας, υπάρχουν απώλειες, οι οποίες οφείλονται κυρίως στην ωμική αντίσταση των γραμμών και έχουν σημαντικό ρόλο στην οικονομική κατανομή του φορτίου. Έστω για παράδειγμα, μια εταιρεία έχει στην κατοχή της δύο θερμικές μονάδες παραγωγή ενέργειας. Η πρώτη

μονάδα, παράγει ενέργεια με μικρό κόστος, αλλά βρίσκεται μακριά από την πόλη, ενώ η δεύτερη, παράγει πιο ακριβά την ενέργεια και βρίσκεται κοντά στην πόλη. Αν δεν προσμετρηθούν οι απώλειες, λόγω μεταφοράς και με βάση όσα αναφέρθηκαν παραπάνω, το κύριο μέρος της παραγωγής θα καλυφθεί από την πρώτη μονάδα, η οποία παράγει φθηνά ενέργεια. Όμως αν προσμετρηθούν οι απώλειες, δημιουργείται το ερώτημα, πόσο πρέπει να αυξηθεί η έξοδος της δεύτερης μονάδας και να μειωθεί της πρώτης, ώστε να μειωθούν οι απώλειες καθώς και η συνολική παραγωγή.

Σε ένα σύστημα μεταφοράς, οι απώλειες μπορούν να εκφραστούν ως συνάρτηση ισχύων εξόδου των μονάδων, δηλαδή  $P_L(P_1, P_2, \dots, P_N)$  και η εξίσωση διατήρησης της ισχύος εκφράζει, ότι η συνολική παραγωγή πρέπει να καλύπτει την συνολική ζήτηση,  $P_R$ , συν τις απώλειες λόγω μεταφοράς,  $P_L$ . Τελικά, το πρόβλημα μεταφράζεται ως εξής:

$$\begin{aligned} \text{Min:} \quad F_T(P_1, P_2, \dots, P_N) &= F_1(P_1) + F_2(P_2) + \dots + F_N(P_N) \\ &= \sum_{i=1}^N F_i(P_i) \end{aligned} \quad (3.48)$$

$$\begin{aligned} \text{Με περιορισμούς:} \quad \Phi &= P_R + P_L(P_1, P_2, \dots, P_N) - \sum_{i=1}^N P_i = 0 \\ &(\text{Φορτίο} + \text{Απώλειες} + \text{Παραγωγή} = 0) \end{aligned} \quad (3.49)$$

$$P_{i \min} \leq P_i \leq P_{i \max} \quad (3.50)$$

Σχηματίζοντας, την συνάρτηση Lagrange είναι:

$$\mathcal{L} = F_T + \lambda \Phi$$

στο σημείο βελτίστου θα πρέπει να ισχύει:

$$\frac{\partial \mathcal{L}}{\partial P_i} = 0, \text{ για κάθε } P_{i \min} \leq P_i \leq P_{i \max}$$

$$\frac{\partial \mathcal{L}}{\partial P_i} = \frac{\partial F_i}{\partial P_i} + \lambda \cdot \left( \frac{\partial P_L}{\partial P_i} - 1 \right) = 0 \quad (3.51)$$

και με μια ανακατανομή όρων θα γίνεται:

$$\frac{1}{1 - \frac{\partial P_L}{\partial P_i}} \cdot \frac{\partial F_i}{\partial P_i} = \lambda \quad (3.52)$$

Ο όρος  $\frac{\partial P_L}{\partial P_i}$ , εκφράζει τις διαφορικές απώλειες του ζυγού  $i$ , στον οποίο είναι συνδεδεμένη η μονάδα  $i$ , ενώ ο συντελεστής  $Pf_i = \frac{1}{1 - \frac{\partial P_L}{\partial P_i}}$ , ονομάζεται συντελεστής ποινής, του ζυγού  $i$ . Ειδικότερα, όταν αυξάνονται οι απώλειες μεταφοράς κατά την αύξηση της εξόδου της μονάδας, οι διαφορικές απώλειες είναι θετικές και ο συντελεστής ποινής μεγαλύτερος της μονάδας.

Η επίλυση του προβλήματος οικονομικής κατανομής με απώλειες, όπως και το πρόβλημα χωρίς απώλειες, επιτυγχάνεται όταν τα διαφορικά κόστη των μονάδων είναι ίδια. Η μόνη διαφορά μεταξύ των δύο προβλημάτων, με ή χωρίς απώλειες, είναι ο συντελεστής ποινής,  $Pf_i$ , ο οποίος επιδρά στην επίλυση, κατά τον ακόλουθο τρόπο.

- Αν  $Pf_i > 1$ , δηλαδή η αύξηση του  $P_i$ , αυξάνει τις απώλειες, ο όρος  $Pf_i \cdot \frac{\partial F_i}{\partial P_i}$ , λειτουργεί έτσι ώστε, το διαφορικό κόστος,  $\frac{\partial F_i}{\partial P_i}$ , να αυξάνεται.
- Αν  $Pf_i < 1$ , δηλαδή η αύξηση του  $P_i$ , μειώνει τις απώλειες, ο όρος  $Pf_i \cdot \frac{\partial F_i}{\partial P_i}$ , λειτουργεί έτσι ώστε, το διαφορικό κόστος,  $\frac{\partial F_i}{\partial P_i}$ , να μειώνεται.

Άρα, οι εξισώσεις (3.51) μετά τον ορισμό της (3.52) μετατρέπονται στην (3.53) και αποτελούν τις εξισώσεις συνεργασίας του προβλήματος.

$$P f_i \cdot \frac{\partial F_i}{\partial P_i} = \lambda, \text{ για κάθε } P_{i \min} \leq P_i \leq P_{i \max} \quad (3.53)$$

Το παραπάνω πρόβλημα λύνεται με τη μέθοδο των συντελεστών B, όπου οι απώλειες μεταφοράς, μπορούν να εκφραστούν ως συνάρτηση της εξόδου των γεννητριών, με την ακόλουθη εξίσωση:

$$P_L = P^T [B] P + P^T B_0 + B_{00} \quad (3.54)$$

Στην (3.54) το  $P$ , είναι το διάνυσμα εξόδου όλων των μονάδων ( $N \times 1$ ), το  $[B]$ , εκφράζει έναν τετραγωνικό πίνακα ( $N \times N$ ), το  $B_0$ , ένα διάνυσμα με διάσταση ( $N \times 1$ ) και τέλος το  $B_{00}$ , μια σταθερά. Η (3.54) γράφεται και ως εξής:

$$P_L = \sum_{i=1}^N \sum_{j=1}^N P_i B_{ij} P_j + \sum_{i=1}^N B_{0i} P_i + B_{00} \quad (3.55)$$

και έτσι, η εξίσωση διατήρησης της ισχύος είναι:

$$\Phi = P_R + \sum_{i=1}^N \sum_{j=1}^N P_i B_{ij} P_j + \sum_{i=1}^N B_{0i} P_i + B_{00} - \sum_{i=1}^N P_i \quad (3.56)$$

Οπότε στο σημείο βελτίστου είναι:

$$\mathcal{L} = F_T + \lambda \cdot P_R + \sum_{i=1}^N \sum_{j=1}^N P_i B_{ij} P_j + \sum_{i=1}^N B_{0i} P_i + B_{00} - \sum_{i=1}^N P_i$$

Άρα:

$$\frac{\partial \mathcal{L}}{\partial P_i} = \frac{\partial F_i}{\partial P_i} + \lambda \cdot \left( 2 \sum_{j=1}^N B_{ij} P_j - B_{0i} - 1 \right) = 0 \quad (3.57)$$

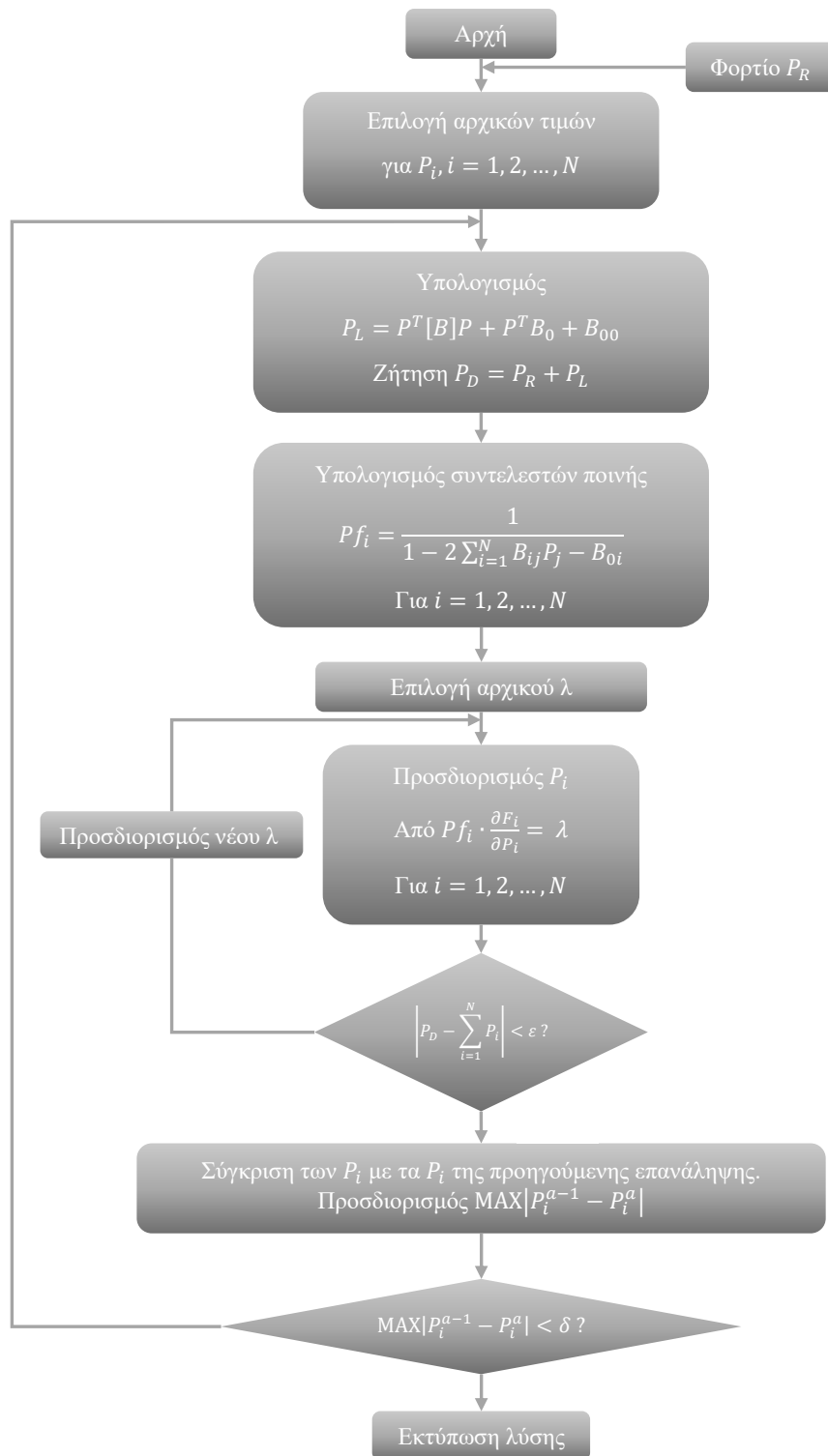
Και ο συντελεστής ποινής:

$$Pf_i = \frac{1}{1 - 2 \sum_{j=1}^N B_{ij} P_j - B_{0i}} \quad (3.58)$$

Σε έναν υπολογιστή, το πρόβλημα αυτό λύνεται με βάση τον αλγόριθμο οικονομικής κατανομής φορτίου με απώλειες. Αρχικά, εισάγεται η απαιτούμενη ενέργεια,  $P_R$  και επιλέγονται κάποιες τιμές για τα  $P_i$ . Στην συνέχεια, υπολογίζονται οι απώλειες μεταφοράς,  $P_L$ , και η νέα ανάγκη για ενέργεια, δηλαδή το άθροισμα της ζητούμενης ενέργειας και των απωλειών,  $P_D = P_L + P_R$ . Έπειτα, υπολογίζονται οι συντελεστές ποινής και επιλέγεται ένα αρχικό  $\lambda$ . Μετά τον προσδιορισμό του αρχικού  $\lambda$ , επιλύεται το πρόβλημα οικονομικής κατανομής χρησιμοποιώντας την σχέση (3.53). Εάν, η νέα πλέον εξίσωση διατήρησης της ισχύος,  $\Phi = P_D - \sum_{i=1}^N P_i$ , έχει τιμή εντός κάποιου ορίου ανοχής  $\varepsilon$  (πχ.  $\varepsilon = 0.1 MW$ ), τότε συγκρίνονται τα  $P_i$ , της νέας επανάληψης, με της προηγούμενης και αν είναι περίπου ίδια, δηλαδή η διαφορά τους  $MAX|P_i^{a-1} - P_i^a|$ , είναι εντός κάποιου ορίου ανοχής  $\delta$  (πχ.  $\delta = 1 MW$ ), τότε ο αλγόριθμος συγκλίνει και τερματίζει εκτυπώνοντας την λύση. Αν όμως, η εξίσωση  $\Phi$ , δεν βρίσκεται εντός κάποιου ορίου ανοχής  $\varepsilon$ , τότε επιλέγεται νέο  $\lambda$  και ακολουθεί η ίδια διαδικασία έως ότου ισχύει ο περιορισμός. Ταυτόχρονα, αν η διαφορά,  $MAX|P_i^{a-1} - P_i^a|$ , της νέας και προηγούμενης λύσης δεν έχει τιμή εντός κάποιου ορίου ανοχής  $\delta$ , τότε η διαδικασία επιστρέφει πίσω στο σημείο το οποίο υπολογίζονται οι απώλειες λόγω μεταφοράς.

Στο διάγραμμα 3.2, παρουσιάζεται ο παραπάνω αλγόριθμος και στο παράδειγμα 3 που ακολουθεί, γίνεται εφαρμογή του.





Διάγραμμα 3.5: Αλγόριθμος Οικονομικής Κατανομής Φορτίου με Απώλειες.

### Παράδειγμα 3

Έστω ξανά το σύστημα του παραδείγματος 1. Αυτή την φορά, υποθέτουμε ότι η τρίτη μονάδα βρίσκεται εγκατεστημένη πολύ κοντά στο κέντρο ζήτησης του φορτίου, έτσι ώστε η παραγόμενη ενέργεια, να μεταφέρεται στους καταναλωτές χωρίς απώλειες. Αντίθετα, η πρώτη και δεύτερη μονάδα, βρίσκονται μακριά από το κέντρο κατανάλωσης και έτσι υπάρχουν απώλειες. Αυτές οι απώλειες μεταφοράς, εκφράζονται ως συνάρτηση των  $P_1$  και  $P_2$ , με βάση την ακόλουθη σχέση.

$$P_L = 0.0003P_1^2 - 0.00005 P_1 \cdot P_2 + 0.0001P_2^2 - 0.01P_1 + 0.01P_2 + 4$$

Χρειάζεται να υπολογιστεί η βέλτιστη κατανομή του φορτίου στις μονάδες, όταν η ζήτηση είναι 900 MW.

Αρχικά, υπολογίζονται τα διαφορικά κόστη, όπως το παράδειγμα 1, τα οποία είναι:

$$\begin{aligned}\frac{dF_1}{dP_1} &= 8.078 + 0.00346P_1 \\ \frac{dF_2}{dP_2} &= 7.992 + 0.00400P_2 \\ \frac{dF_3}{dP_3} &= 8.110 + 0.00948P_3\end{aligned}$$

Στην συνέχεια υπολογίζονται οι διαφορικές απώλειες, οι οποίες είναι:

$$\begin{aligned}\frac{dP_L}{dP_1} &= 0.0006P_1 - 0.00005P_2 - 0.01 \\ \frac{dP_L}{dP_2} &= -0.00005P_1 + 0.0002P_2 + 0.01 \\ \frac{dP_L}{dP_3} &= 0\end{aligned}$$

Συνεπώς οι συντελεστές ποινής,  $Pf_i = \frac{1}{1 - \frac{\partial P_L}{\partial P_i}}$ , γίνονται:

$$Pf_1 = \frac{1}{1 - 0.0006P_1 + 0.00005P_2 + 0.01}$$

$$Pf_2 = \frac{1}{1 + 0.00005P_1 - 0.0002P_2 - 0.01}$$

$$Pf_3 = 1$$

Τέλος, οι εξισώσεις συνεργασίας,  $Pf_i \cdot \frac{\partial F_i}{\partial P_i} = \lambda$  και η εξίσωση διατήρησης της ισχύος είναι:

$$Pf_1 \cdot [8.078 + 0.00346P_1] = \lambda$$

$$Pf_2 \cdot [7.992 + 0.00400P_2] = \lambda$$

$$Pf_3 \cdot [8.110 + 0.00948] = \lambda$$

και

$$P_1 + P_2 + P_3 = 900 + P_L(P_1, P_2) \text{ MW}$$

### 1<sup>η</sup> Επανάληψη

Πρώτα, επιλέγουμε τις τιμές των ισχύων εξόδου των μονάδων και υποθέτουμε, ότι έχουμε απώλειες της τάξεως των 10 MW. Θα μπορούσαμε να επιλέξουμε διαφορετική τιμή για τις απώλειες, πχ. 50 MW ή 100 MW, αλλά για το παράδειγμα θεωρούμε ότι είναι 10 MW.

$$P_1 = 350 \text{ MW}$$

$$P_2 = 370 \text{ MW}$$

$$P_3 = 190 \text{ MW}$$

$$(P_1 + P_2 + P_3 = 910 \text{ MW, εκτιμώντας 10 MW απώλειες})$$

Υπολογίζουμε τις απώλειες, κάνοντας αντικατάσταση τις τιμές εξόδου των μονάδων στην σχέση των απωλειών,  $P_L$ , και η συνολική απαιτούμενη ενέργεια είναι το άθροισμα της ζήτησης και των απωλειών. Συνεπώς είναι οι σχέσεις:

$$P_L = 0.0003 \cdot 350^2 - 0.00005 \cdot 350 \cdot 370 + 0.0001 \cdot 370^2 - 0.01 \cdot 370 + 0.01 \cdot 350 + 4$$

άρα,

$$P_L = 48.17 \text{ MW}$$

και

$$P_D = 900 + 48.17 = 948.17 \text{ MW}$$

Έπειτα, υπολογίζονται και οι συντελεστές ποιής:

$$\begin{aligned} Pf_1 &= 1.22175 \\ Pf_2 &= 1.07124 \\ Pf_3 &= 1 \end{aligned}$$

Έτσι οι εξισώσεις συνεργασίας γίνονται:

$$Pf_1 \cdot \frac{dF_1}{dP_1} = 1.22175 \cdot [8.078 + 0.00346P_1] = \lambda$$

$$Pf_2 \cdot \frac{dF_2}{dP_2} = 1.07124 \cdot [7.992 + 0.00400P_2] = \lambda$$

$$Pf_3 \cdot \frac{dF_3}{dP_3} = 1 \cdot [8.110 + 0.00948] = \lambda$$

και η εξίσωση διατήρησης της ισχύος:

$$P_1 + P_2 + P_3 = P_D = 948.17 \text{ MW}$$

Στην συνέχεια επιλέγουμε, αρχική τιμή για το  $\lambda$ , έστω  $\lambda = 11 \text{ €/kWh}$  και υπολογίζουμε:

$$\begin{aligned}
P_1 &= 267.49 \text{ MW} \\
P_2 &= 569.12 \text{ MW} \Rightarrow \mathbf{P_2 = 400 \text{ MW}} \\
P_3 &= 80.3 \text{ MW} \Rightarrow \mathbf{P_3 = 200 \text{ MW}}
\end{aligned}$$

Η εξίσωση διατήρησης της ισχύος γίνεται:

$$P_1 + P_2 + P_3 = 867.49 < P_D = 948.17 \text{ MW}$$

$$\Phi = P_D - \sum_{i=1}^3 P_i = 80.68 \text{ MW}$$

Η παραγόμενη ενέργεια, είναι μικρότερη από την απαιτούμενη. Συνεπώς επιλέγεται μεγαλύτερο  $\lambda$ . Έστω,  $\lambda = 12.1$ , μεγαλύτερο δηλαδή κατά 10%, από το προηγούμενο.

Έτσι, η λύση των εξισώσεων συνεργασίας είναι:

$$\begin{aligned}
P_1 &= 527.70 \text{ MW} \\
P_2 &= 825.84 \text{ MW} \Rightarrow \mathbf{P_2 = 400 \text{ MW}} \\
P_3 &= 420.89 \text{ MW} \Rightarrow \mathbf{P_3 = 200 \text{ MW}}
\end{aligned}$$

και η εξίσωση διατήρησης της ισχύος:

$$\Phi = P_D - \sum_{i=1}^3 P_i = -179.537 \text{ MW}$$

Η παραγόμενη ενέργεια, είναι μεγαλύτερη από την απαιτούμενη. Συνεπώς επιλέγεται μικρότερο  $\lambda$ , αλλά αυτή την φορά μέσω γραμμικής παρεμβολής. Είναι:

$$\lambda_2 = \lambda_1 - \frac{\lambda_1 - \lambda_0}{\Phi_1 - \Phi_0} \Phi_1 = 12.1 - \frac{12.1 - 11}{-179.537 - 80.68} (-179.537) = 11.341 \text{ €/kWh}$$

Για  $\lambda = 11.341$ , οι εξισώσεις συνεργασίας δίνουν:

$$\begin{aligned} P_1 &= 348.17 \text{ MW} \\ P_2 &= 648.84 \text{ MW} \Rightarrow \mathbf{P_2 = 400 \text{ MW}} \\ P_3 &= 340.83 \text{ MW} \Rightarrow \mathbf{P_3 = 200 \text{ MW}} \end{aligned}$$

με

$$\Phi = P_D - \sum_{i=1}^3 P_i = 0 \text{ MW}$$

## 2<sup>η</sup> Επανάληψη

Για την δεύτερη επανάληψη, χρησιμοποιούμε τα  $P_1, P_2, P_3$ , της προηγούμενης επανάληψης, δηλαδή:

$$\begin{aligned} P_1 &= 348.17 \text{ MW} \\ P_2 &= 400 \text{ MW} \\ P_3 &= 200 \text{ MW} \end{aligned}$$

Η διαδικασία ξεκινά ξανά από την αρχή. Υπολογίζουμε τις απώλειες και την συνολική ζήτηση και έχουμε:

$$P_L = 49.92 \text{ MW}$$

$$P_D = 900 + 49.92 = 949.92 \text{ MW}$$

Στην συνέχεια, υπολογίζουμε τους συντελεστές ποιής:

$$\begin{aligned}Pf_1 &= 1.21788 \\Pf_2 &= 1.07827 \\Pf_3 &= 1\end{aligned}$$

και οι εξισώσεις συνεργασίας είναι:

$$Pf_1 \cdot \frac{dF_1}{dP_1} = 1.21788 \cdot [8.078 + 0.00346P_1] = \lambda$$

$$Pf_2 \cdot \frac{dF_2}{dP_2} = 1.07827 \cdot [7.992 + 0.00400P_2] = \lambda$$

$$Pf_3 \cdot \frac{dF_3}{dP_3} = 1 \cdot [8.110 + 0.00948] = \lambda$$

και η εξίσωση διατήρησης της ισχύος:

$$P_1 + P_2 + P_3 = 949.9 \text{ MW}$$

Επιλέγουμε, αρχική τιμή για το  $\lambda$ , έστω  $\lambda = 11 \text{ €/kWh}$  και μετά από τρεις επαναλήψεις λάμδα υπολογίζουμε:

$$\begin{aligned}\lambda &= 11.313 \text{ €/kWh} \\P_1 &= 349.92 \text{ MW} \\P_2 &= 400 \text{ MW} \\P_3 &= 200 \text{ MW}\end{aligned}$$

με

$$\Phi = P_D - \sum_{i=1}^3 P_i = 0 \text{ MW}$$

### 3<sup>η</sup> Επανάληψη

Όμοια και για την τρίτη επανάληψη, χρησιμοποιούμε τα  $P_1$ ,  $P_2$ ,  $P_3$ , της προηγούμενης επανάληψης, δηλαδή:

$$P_1 = 349.92 \text{ MW}$$

$$P_2 = 400 \text{ MW}$$

$$P_3 = 200 \text{ MW}$$

Υπολογίζονται τα παρακάτω:

$$P_L = 49.92 \text{ MW}$$

$$P_D = 900 + 49.92 = 949.92 \text{ MW}$$

$$Pf_1 = 1.21944$$

$$Pf_2 = 1.07817$$

$$Pf_3 = 1$$

οι εξισώσεις συνεργασίας είναι:

$$Pf_1 \cdot \frac{dF_1}{dP_1} = 1.21944 \cdot [8.078 + 0.00346P_1] = \lambda$$

$$Pf_2 \cdot \frac{dF_2}{dP_2} = 1.07817 \cdot [7.992 + 0.00400P_2] = \lambda$$

$$Pf_3 \cdot \frac{dF_3}{dP_3} = 1 \cdot [8.110 + 0.00948] = \lambda$$

η εξίσωση διατήρησης της ισχύος:

$$P_1 + P_2 + P_3 = 950.24 \text{ MW}$$



Τελικά, η λύση είναι:

$$\lambda = 11.328 \text{ €/kWh}$$

$$P_1 = 350.24 \text{ MW}$$

$$P_2 = 400 \text{ MW}$$

$$P_3 = 200 \text{ MW}$$

Παρατηρείται, ότι η λύση της τελευταίας επανάληψης, είναι περίπου ίδια με την λύση της προηγούμενης επανάληψης, δηλαδή  $\text{MAX}|P_i^3 - P_i^2| < 1 \text{ MW}$ . Άρα ο αλγόριθμος έχει συγκλίνει και τερματίζεται μετά από 3 επαναλήψεις.

### 3.6. Ένταξη μονάδων παραγωγής

Ένα μέρος της βελτιστοποίησης του προβλήματος οικονομικής κατανομής φορτίου, είναι και η ένταξη μονάδων παραγωγής. Ως ένταξη μιας μονάδας, ορίζεται ο συγχρονισμός και η σύνδεση της με το δίκτυο, ώστε να δώσει ισχύ στο σύστημα. Όπως αναφέρθηκε στο Κεφάλαιο 2, η ηλεκτρική ζήτηση, μεταβάλλεται κατά την διάρκεια της ημέρας, ανάλογα τον κύκλο της ανθρώπινης δραστηριότητας. Αν ο απώτερος σκοπός μιας εταιρείας ενέργειας ήταν απλά, η κάλυψη του ζητούμενου φορτίου, θα μπορούσαν να ενταχθούν όσες μονάδες είναι απαραίτητες, ώστε να καλύπτεται το μέγιστο φορτίο της ζήτησης και οι μονάδες να λειτουργούν όλη μέρα, ακόμα και όταν το φορτίο είναι χαμηλό. Όμως, κάτι τέτοιο κρίνεται ως αντισυμβαλλόμενο. Για τον λόγο αυτό, δημιουργήθηκε ένα πρόγραμμα ένταξης των μονάδων για το επόμενο χρονικό διάστημα, έστω εικοσιτετράωρο.

Για αμιγώς θερμικά συστήματα, το πρόβλημα της βέλτιστης ένταξης μονάδων, ορίζεται ως εξής. Έστω, ένα σύστημα με  $N$  θερμικές μονάδες στην διάθεση του, για το οποίο γνωρίζουμε την πρόβλεψη του φορτίου,  $P_D(t)$ ,  $t = 1, \dots, T$ , και την δοσμένη χρονική περίοδο  $T$ , η οποία είναι συνήθως  $T = 24 \text{ h}$  ή και  $T = 168 \text{ h}$ , δηλ. 1 ή 7 ημέρες. Ζητείται, το πρόγραμμα ένταξης των θερμικών αυτών μονάδων, ώστε το συνολικό κόστος λειτουργίας για μια χρονική περίοδο  $T$ , να ελαχιστοποιείται.

Γνωρίζοντας για τις καμπύλες κόστους,  $F_i(P_i)$ , των μονάδων, το πρόβλημα μοντελοποιείται ως εξής:

$$F_T = \sum_{t=1}^T \sum_{i=1}^N [e_i(t)F_i(P_i(t)) + e_i(t) \cdot (1 - e_i(t-1)) \cdot SU_i] \quad (3.59)$$

όπου,

$$e_i = \begin{cases} 1, & \text{αν η μονάδα } i \text{ λειτουργεί την ώρα } t \\ 0, & \text{αν η μονάδα } i \text{ δεν λειτουργεί την ώρα } t \end{cases}$$

$SU_i$ : το κόστος εκκίνησης της μονάδας  $i$

Ένα πρόγραμμα ένταξης μονάδων, βασίζεται στην πρόβλεψη του φορτίου. Έτσι, τυχόν σφάλματα στην πρόβλεψη, μπορεί να οδηγήσουν σε ένα μη οικονομικά βέλτιστο πρόγραμμα ένταξης. Γενικά, κατά το πρόγραμμα ένταξης των μονάδων, λαμβάνεται υπόψη και η παραγόμενη ενέργεια από ΑΠΕ, υδροηλεκτρικούς σταθμούς αλλά και η ενέργεια, που προέρχεται από ανταλλαγές με τα γειτονικά δίκτυα. Στην περίπτωση μας, δεν θα ληφθούν υπόψιν. Στην συνέχεια, παρουσιάζονται δύο παραδείγματα ένταξης μονάδων.

#### Παράδειγμα 4

Έστω, μια εταιρεία παραγωγής ενέργειας η οποία διαθέτει τις παρακάτω πετρελαϊκές μονάδες. Κάποια στιγμή το φορτίο της εταιρείας είναι  $550 \text{ MW}$ . Ζητείται η μονάδα ή ο συνδυασμός των μονάδων, που πρέπει να λειτουργήσουν, ώστε να τροφοδοτήσουν το δίκτυο κατά τον πλέον οικονομικό τρόπο.

Μονάδα 1:

Μέγιστη Έξοδος:

$$P_{1 \max} = 600 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{1 \min} = 150 \text{ MW}$$

Κόστος Λειτουργίας

Μονάδας:

$$H_1 \left( \frac{\text{Gcal}}{h} \right) = 122.5 + 1.729P_1 + 0.00034P_1^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου:  $3.96 \text{ k€/Gcal}$

Μονάδα 2:

Μέγιστη Έξοδος:

$$P_{2\max} = 400 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{2\min} = 100 \text{ MW}$$

Κόστος Λειτουργίας

Μονάδας:

$$H_2 \left( \frac{\text{Gcal}}{h} \right) = 67.7 + 1.714P_2 + 0.00042P_2^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

Μονάδα 3:

Μέγιστη Έξοδος:

$$P_{3\max} = 200 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{3\min} = 50 \text{ MW}$$

Κόστος Λειτουργίας

Μονάδας:

$$H_3 \left( \frac{\text{Gcal}}{h} \right) = 20.4 + 2.088P_3 + 0.0012P_3^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

Αρχικά υπολογίζονται οι καμπύλες κόστους λειτουργίας των μονάδων όπως έγινε και στα προηγούμενα παραδείγματα.

$$\begin{aligned} F_1(P_1) &= H_1(P_1) * 3.96 = 485.1 + 6.847P_1 + 0.00135P_1^2 & \text{k€/h} \\ F_2(P_2) &= H_2(P_2) * 3.96 = 268.1 + 6.787P_2 + 0.00166P_2^2 & \text{k€/h} \\ F_3(P_3) &= H_3(P_3) * 3.96 = 80.8 + 8.268P_3 + 0.00474P_3^2 & \text{k€/h} \end{aligned}$$

Για την επίλυση του προβλήματος, δοκιμάζονται όλοι οι πιθανοί συνδυασμοί των μονάδων. Ορισμένοι από αυτούς, θα οδηγήσουν σε μη εφικτή λύση, όπως για παράδειγμα το άθροισμα των μεγίστων εξόδων είναι μικρότερο από το φορτίο, ή όταν το άθροισμα των ελαχίστων εξόδων των μονάδων, είναι μεγαλύτερο από το φορτίο. Για κάθε έναν εφικτό συνδυασμό, γίνεται κατανομή των φορτίων με τις μεθόδους που αναλύθηκαν παραπάνω.

Πιο συγκεκριμένα, για την περίπτωση την οποία λειτουργεί μόνο μια μονάδα, γίνεται απλά κατανομή φορτίου στην μονάδα, ενώ για την περίπτωση την οποία λειτουργούν δύο και τρεις μονάδες αντίστοιχα, γίνεται εφαρμογή του αλγορίθμου επαναλήψεων λάμδα. Τέλος, από τις καμπύλες κόστους λειτουργίας της κάθε

μονάδας, υπολογίζεται και το κόστος ανάλογα το φορτίο. Στον ακόλουθο πίνακα, παρουσιάζονται τα αποτελέσματα του κάθε συνδυασμού.

Πίνακας 3.1: Αποτελέσματα παραδείγματος 4.

Κατάσταση μονάδων			Όρια Ισχύος Συνδυασμού		Έξοδος Μονάδων			Κόστος Μονάδων (κ€)			Συνολικό Κόστος (κ€)
$UN_1$	$UN_2$	$UN_3$	$P_{min}$	$P_{max}$	$P_1$	$P_2$	$P_3$	$F_1$	$F_2$	$F_3$	
OFF	OFF	OFF	0	0	Μη Εφικτή Λύση						
OFF	OFF	ON	50	200	Μη Εφικτή Λύση						
OFF	ON	OFF	100	400	Μη Εφικτή Λύση						
ON	OFF	OFF	150	600	550	0	0	4658	0	0	4658
OFF	ON	ON	150	600	0	400	150	0	3249	1428	4677
ON	OFF	ON	200	800	500	0	50	4245	0	506	4751
ON	ON	OFF	250	1000	294	256	0	2615	2114	0	4729
ON	ON	ON	300	1200	266	234	50	2405	1944	506	4855

Για παράδειγμα, έστω ο συνδυασμός στον οποίο λειτουργούν μόνο οι μονάδες 1 και 2. Η επίλυση του προβλήματος κατανομής θα γίνει όπως έγινε στο παράδειγμα 1.

Αρχικά, αφού έχουν υπολογιστεί οι καμπύλες ωριαίου κόστους λειτουργίας των μονάδων υπολογίζονται οι εξισώσεις συνεργασίας:

$$\frac{dF_1}{dP_1} = 6.847 + 0.00270P_1 = \lambda$$

$$\frac{dF_2}{dP_2} = 6.787 + 0.00332P_2 = \lambda$$

και η διατήρηση της ισχύος (3.39), γίνεται:

$$P_1 + P_2 = 550 \text{ MW}$$

Το παραπάνω σύστημα, των τριών γραμμικών εξισώσεων, λύνεται εύκολα ως προς τις μεταβλητές  $P_1$ ,  $P_2$  και  $\lambda$  αντίστοιχα. Αρχικά, χρησιμοποιούνται οι εξισώσεις συνεργασίας, από τις οποίες υπολογίζονται τα  $P_1$ ,  $P_2$ , συναρτήσει του  $\lambda$  και αντικαθίστανται στην εξίσωση διατήρησης της ισχύος, η οποία λύνεται ως προς την μεταβλητή  $\lambda$ . Έτσι, είναι:

$$P_1 = \frac{\lambda - 6.847}{0.00270}$$

$$P_2 = \frac{\lambda - 6.787}{0.00332}$$

και

$$\frac{\lambda - 6.847}{0.00270} + \frac{\lambda - 6.787}{0.00332} = 550$$

Άρα έχουμε:

$$\lambda = 7.639 \frac{k\text{€}}{MW} \text{ ή } \lambda = 7.639 \frac{\text{€}}{kW}$$

και

$$P_1 = 293.34 \text{ MW}$$

$$P_2 = 256.62 \text{ MW}$$

Αντίστοιχα το κόστος για την περίπτωση αυτή είναι:

$$F_1(293.34) = 485.1 + 6.847 * 293.34 + 0.00135 * 293.34^2 \quad k\text{€/h}$$

$$= 2,609.764$$

$$F_2(256.64) = 268.1 + 6.787 * 256.64 + 0.00166 * 256.64^2 \quad k\text{€/h}$$

$$= 2,119.250$$

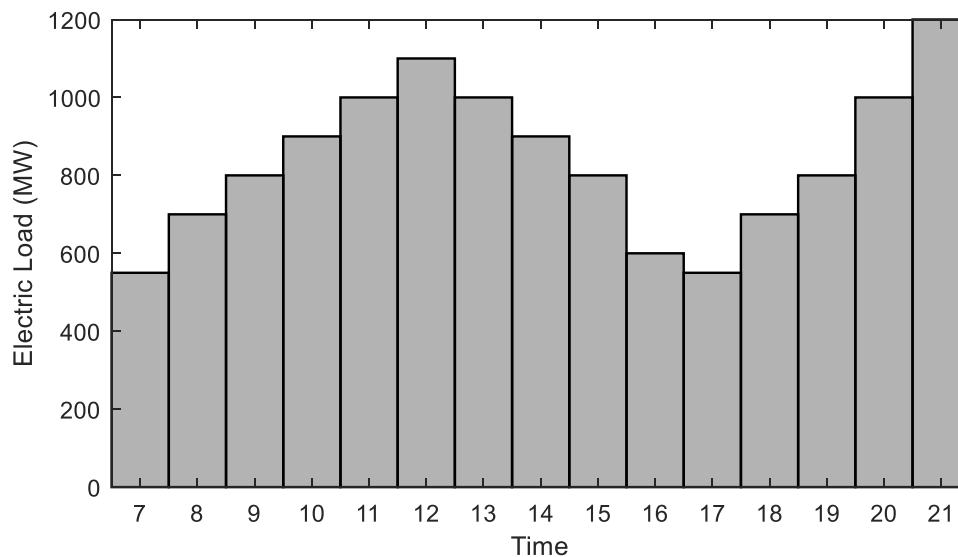
Τέλος το συνολικό κόστος ισούται με:

$$4,729.014 \text{ k€/h}$$

Η ίδια διαδικασία ακολουθήθηκε στις υπόλοιπες περιπτώσεις του παραδείγματος.

## Παράδειγμα 5

Έστω, ξανά η εταιρεία του παραδείγματος 4. Αυτή την φορά, δεν θα υπάρχει ένα στιγμιαίο φορτίο, αλλά μια καμπύλη φορτίου διάρκειας μιας ημέρας. Το ελάχιστο φορτίο της καμπύλης αυτής είναι  $550\text{ MW}$ , ενώ το μέγιστο της  $1200\text{ MW}$ .



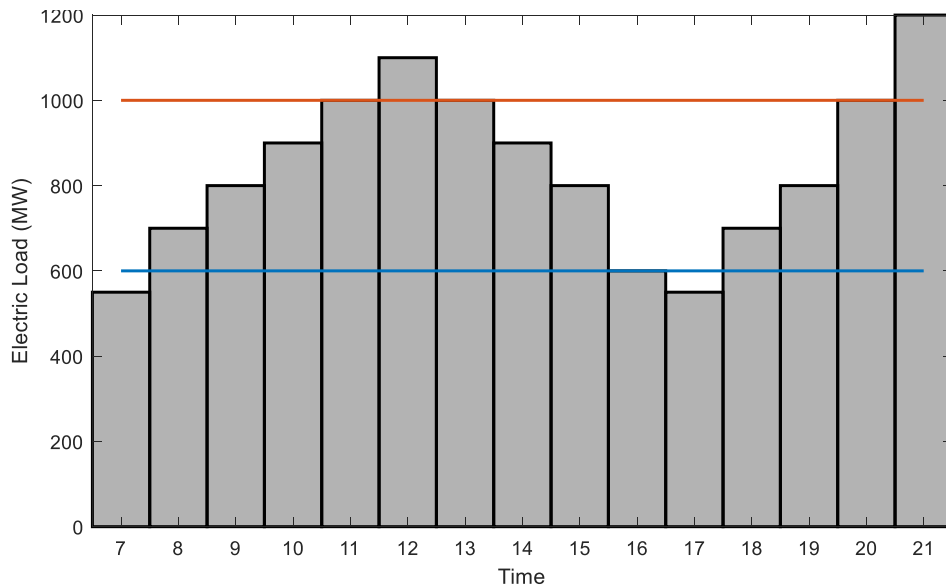
Γράφημα 3.1: Καμπύλη φορτίου.

Προκειμένου να καλυφθεί η καμπύλη φορτίου, κάποιες μονάδες κρατούνται εκτός λειτουργίας, όταν το φορτίο ελαττώνεται, ενώ εντάσσονται ξανά, όταν αυτό αυξάνεται. Αξίζει να σημειωθεί, ότι ένα πρόβλημα ένταξης έχει πολλούς περιορισμούς, οι οποίοι καθιστούν την λύση του περίπλοκη. Στην περίπτωση μας, ο μοναδικός περιορισμός, είναι να καλύπτεται η ζήτηση.

Ο κανόνας ένταξης που θα χρησιμοποιηθεί είναι:

- Αν το φορτίο είναι πάνω από  $1000\text{ MW}$ , τότε λειτουργούν όλες οι μονάδες μαζί.
- Αν το φορτίο είναι μεταξύ  $600\text{ MW}$  και  $1000\text{ MW}$ , τότε λειτουργούν οι μονάδες 1 και 2.
- Αν το φορτίο είναι μικρότερο από  $600\text{ MW}$ , τότε λειτουργεί μόνο, η μονάδα 1.

Ο κανόνας παρουσιάζεται στο γράφημα 3.2. Ειδικότερα, στην περιοχή κάτω από την μπλε γραμμή λειτουργεί μόνο η μονάδα 1. Έπειτα, στην περιοχή ανάμεσα από την μπλε και την πορτοκαλί γραμμή λειτουργούν οι μονάδες 1 και 2, ενώ στην περιοχή πάνω από την πορτοκαλί, λειτουργούν όλες μαζί.



Γράφημα 3.2: Πρόγραμμα Ένταξης Μονάδων.

Όπως και στο παράδειγμα 4, αρχικά υπολογίζονται οι καμπύλες κόστους λειτουργίας των μονάδων.

$$\begin{aligned}
 F_1(P_1) &= H_1(P_1) * 3.96 = 485.1 + 6.847P_1 + 0.00135P_1^2 & \text{k€/h} \\
 F_2(P_2) &= H_2(P_2) * 3.96 = 268.1 + 6.787P_2 + 0.00166P_2^2 & \text{k€/h} \\
 F_3(P_3) &= H_3(P_3) * 3.96 = 80.8 + 8.268P_3 + 0.00474P_3^2 & \text{k€/h}
 \end{aligned}$$

Στην συνέχεια παρουσιάζονται τα αποτελέσματα του προγράμματος ένταξης των μονάδων, κατά αύξουσα σειρά του φορτίου. Εδώ δεν υπάρχουν μη εφικτές λύσεις καθώς, τα φορτία και οι μονάδες οι οποίες θα λειτουργήσουν, είναι γνωστά. Όμοια και σε αυτή την λύση, χρησιμοποιήθηκαν οι μέθοδοι κατανομής φορτίου που παρουσιάστηκαν στα παραπάνω παραδείγματα. Για την περίπτωση που λειτουργεί μια μονάδα, έγινε κατανομή του φορτίου σε αυτή, ενώ για τις περιπτώσεις που λειτουργούν δύο ή και όλες οι μονάδες, έγινε εφαρμογή του αλγορίθμου επανάληψης

λάμδα. Τέλος, με τις καμπύλες κόστους, υπολογίστηκε το κόστος παραγωγής ενέργειας για την κάθε μονάδα.

Πίνακας 3.2: Αποτελέσματα παραδείγματος 5.

Φορτίο	Κατάσταση μονάδων			Όρια Ισχύος Συνδυασμού		Έξοδος Μονάδων			Κόστος Μονάδων (k€)			Συνολικό Κόστος
(MW)	$UN_1$	$UN_2$	$UN_3$	$P_{min}$	$P_{max}$	$P_1$	$P_2$	$P_3$	$F_1$	$F_2$	$F_3$	(k€)
550	1	0	0	150	600	550	0	0	4658	0	0	4658
600	1	0	0	150	600	600	0	0	5078	0	0	5078
650	1	1	0	250	1000	349	301	0	3041	2459	0	5500
700	1	1	0	250	1000	377	323	0	3258	2634	0	5892
<b>750</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>250</b>	<b>1000</b>	<b>405</b>	<b>345</b>	<b>0</b>	<b>3476</b>	<b>2811</b>	<b>0</b>	<b>6287</b>
800	1	1	0	250	1000	432	368	0	3698	2991	0	6689
850	1	1	0	250	1000	460	390	0	3918	3169	0	7087
900	1	1	0	250	1000	500	400	0	4245	3249	0	7494
950	1	1	0	250	1000	550	400	0	4658	3249	0	7907
1000	1	1	1	250	1000	600	400	0	5078	3249	0	8327
1050	1	1	1	300	1200	600	400	50	5078	3249	506	8833
1100	1	1	1	300	1200	600	400	100	5078	3249	955	9282
1150	1	1	1	300	1200	600	400	150	5078	3249	1428	9755
1200	1	1	1	300	1200	600	400	200	5078	3249	1925	10252

Για παράδειγμα, έστω ότι η ζήτηση είναι 700 MW. Συνεπώς, λειτουργούν μόνο οι μονάδες 1 και 2. Η επίλυση του προβλήματος κατανομής θα γίνει όπως το παράδειγμα 1. Αρχικά, αφού έχουν υπολογιστεί οι καμπύλες ωριαίου κόστους λειτουργίας των μονάδων υπολογίζονται οι εξισώσεις συνεργασίας:

$$\frac{dF_1}{dP_1} = 6.847 + 0.00270P_1 = \lambda$$

$$\frac{dF_2}{dP_2} = 6.787 + 0.00332P_2 = \lambda$$

και η διατήρηση της ισχύος (3.39), γίνεται:

$$P_1 + P_2 = 750 \text{ MW}$$



Το παραπάνω σύστημα, των τριών γραμμικών εξισώσεων, λύνεται εύκολα ως προς τις μεταβλητές  $P_1$ ,  $P_2$  και  $\lambda$  αντίστοιχα. Αρχικά, χρησιμοποιούνται οι εξισώσεις συνεργασίας, από τις οποίες υπολογίζονται τα  $P_1$ ,  $P_2$ , συναρτήσει του  $\lambda$  και αντικαθίστανται στην εξίσωση διατήρησης της ισχύος, η οποία λύνεται ως προς την μεταβλητή  $\lambda$ . Έτσι, είναι:

$$P_1 = \frac{\lambda - 6.847}{0.00270}$$

$$P_2 = \frac{\lambda - 6.787}{0.00332}$$

και

$$\frac{\lambda - 6.847}{0.00270} + \frac{\lambda - 6.787}{0.00332} = 750$$

Άρα έχουμε:

$$\lambda = 7.937 \frac{k\text{€}}{MW} \text{ ή } \lambda = 7.937 \frac{\text{€}}{kW}$$

και

$$P_1 = 403.704 \text{ MW}$$

$$P_2 = 346.386 \text{ MW}$$

Αντίστοιχα το κόστος για την περίπτωση αυτή είναι:

$$F_1(403.704) = 485.1 + 6.847 * 403.704 + 0.00135 * 403.704^2 \quad \text{k€/h}$$

$$= 3,469.28$$

$$F_2(346.386) = 268.1 + 6.787 * 346.386 + 0.00166 * 346.386^2 \quad \text{k€/h}$$

$$= 2,818.19$$

Τέλος το συνολικό κόστος ισούται με:

$$6,287.44 \text{ k€/h}$$

Η ίδια διαδικασία ακολουθήθηκε για τα υπόλοιπα φορτία του παραδείγματος.

## Κεφάλαιο 4: Ανάπτυξη λογισμικού για επίλυση του προβλήματος οικονομικής κατανομής

---

### 4.1. Εισαγωγή

Πολλές φορές η επίλυση του προβλήματος οικονομικής κατανομής, χωρίς την βοήθεια ηλεκτρονικού υπολογιστή, μπορεί να είναι χρονοβόρα και μάλιστα χωρίς να γνωρίζουμε ότι η λύση, θα είναι σίγουρα η βέλτιστη. Οι πράξεις, ανάλογα το πρόβλημα, γίνονται ολοένα και πιο περίπλοκες με αποτέλεσμα η πιθανότητα σφάλματος να αυξάνεται. Έτσι, η χρήση ηλεκτρονικού υπολογιστή καθίσταται αναγκαία, πόσο μάλλον σε περιπτώσεις, όπου εκείνη την στιγμή, πρέπει να κατανεμηθεί το φορτίο στις μονάδες παραγωγής.

Στο κεφάλαιο αυτό, παρουσιάζεται η επίλυση των παραδειγμάτων του κεφαλαίου 3 σε περιβάλλον ηλεκτρονικού υπολογιστή, με χρήση του λογισμικού MatLAB της MathWorks. Στόχος της ανάπτυξης του κώδικα είναι, η όσο το δυνατόν πιο αυτοματοποιημένη επίλυση των προβλημάτων, με ελάχιστη παραμετροποίηση από τον χρήστη. Αρχικά, δημιουργήθηκε ένας φάκελος με όνομα, `Economic_Distribution_Examples`, στον οποίο αποθηκεύτηκαν όλα απαιτούμενα τα αρχεία, για την επίλυση του εκάστοτε προβλήματος. Μέσα στον φάκελο αυτό, ο χρήστης συναντά υπό – φακέλους για κάθε ένα παράδειγμα ξεχωριστά. Ο διαχωρισμός αυτός σε υπό – φακέλους, γίνεται για την καλύτερη οργάνωση των αρχείων, καθώς χρησιμοποιούνται αρκετές συναρτήσεις και διαφορετικά δεδομένα. Στην συνέχεια αναλύονται οι κώδικες, που χρησιμοποιήθηκαν για την επίλυση των παραδειγμάτων.

## 4.2. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 1

### Εκφώνηση

Έστω, μια μικρή εταιρεία παραγωγής ενέργειας, της οποίας το σύστημα παραγωγής, αποτελείται από τρεις θερμικές μονάδες. Χρειάζεται να υπολογιστεί το οικονομικό σημείο λειτουργίας των μονάδων, όταν η ζήτηση του φορτίου είναι 900 MW.

#### Μονάδα 1:

Μέγιστη Έξοδος:

$$P_{1\max} = 600 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{1\min} = 150 \text{ MW}$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_1 \left( \frac{\text{Gcal}}{h} \right) = 165.8 + 2.321P_1 + 0.0005P_1^2$$

Καύσιμο: Φυσικό αέριο

Κόστος καυσίμου: 3.48 k€/Gcal

#### Μονάδα 2:

Μέγιστη Έξοδος:

$$P_{2\max} = 400 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{2\min} = 100 \text{ MW}$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_2 \left( \frac{\text{Gcal}}{h} \right) = 78.3 + 2.018P_2 + 0.0005P_2^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

#### Μονάδα 3:

Μέγιστη Έξοδος:

$$P_{3\max} = 200 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{3\min} = 50 \text{ MW}$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_3 \left( \frac{\text{Gcal}}{h} \right) = 39.4 + 2.048P_3 + 0.0012P_3^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

Ο χρήστης, ανοίγει τον φάκελο `Economic_Distribution_Examples` και αναζητά τον υπό – φάκελο με όνομα `Example_1`. Μόλις τον ανοίξει, συναντά το `m.file` με όνομα `Example_1.m`. Ανοίγοντας το `m.file`, ο χρήστης βρίσκεται πλέον στο περιβάλλον του λογισμικού MatLAB και βλέπει τον κώδικα του παραδείγματος 1 του οποίου η ανάλυση γίνεται παρακάτω.

Η επίλυση του παραδείγματος έγινε, όπως ακριβώς παρουσιάζεται στο Κεφάλαιο 3. Στο πρώτο τμήμα του κώδικα, με όνομα `Intro`, γίνεται καθαρισμός της μνήμης και του χώρου εργασίας της MatLAB. Η εντολή `format compact`, χρησιμοποιείται, ώστε τα αποτελέσματα να μην έχουν μεγάλα διαστήματα μεταξύ τους ενώ η `clc` και η `clear all` προκειμένου να καθαρίσει το `command window` και το `workspace` από μεταβλητές.

## Intro

```
clear all
format compact
clc
```

Στο δεύτερο τμήμα με όνομα, `Data`, γίνεται η καταχώρηση των δεδομένων του παραδείγματος καθώς και ορίζονται ως συμβολικές μεταβλητές, οι εξόδοι ισχύος των μονάδων,  $P_1$ ,  $P_2$ ,  $P_3$  και ο συντελεστής Lagrange,  $\lambda$ .

## Data

```
syms P1 P2 P3 lamda

%Unit_1---->Gas
P1_max=600;
P1_min=150;
H1=165.8 + 2.321*P1 + 0.0005*(P1^2);
cost1=3.48;

%Unit_2---->fuel oil
P2_max=400;
P2_min=100;
H2=78.3 + 2.018*P2 + 0.0005*(P2^2);
cost2=3.96;

%Unit_3---->fuel oil
P3_max=200;
P3_min=50;
```

```
H3=39.4 + 2.048*P3 + 0.0012*(P3^2);  
cost3=3.96;
```

Στο τρίτο τμήμα του κώδικα, με όνομα Necessary equations, υπολογίζονται όλες οι εξισώσεις που χρειάζονται για την επίλυση του προβλήματος οικονομικής κατανομής. Συγκεκριμένα, υπολογίζονται οι καμπύλες ωριαίου κόστους και ορίζονται οι εξισώσεις συνεργασίας, η εξίσωση διατήρησης της ενέργειας καθώς και η ηλεκτρική ζήτηση.

## Necessary equations

```
%hourly cost curves  
F1=H1*cost1;  
F2=H2*cost2;  
F3=H3*cost3;  
  
%Cooperation equations  
dF1=(diff(F1,P1)==lamda);  
dF2=(diff(F2,P2)==lamda);  
dF3=(diff(F3,P3)==lamda);  
  
%Power conservation equation  
Power_conservation_equation=P1+P2+P3;  
  
%Electrical load demand  
Electrical_load_demand=900;  
  
%Final power conservation equation  
Final_equation=(Power_conservation_equation==Electrical_load_demand);
```

Τέλος, στο τέταρτο κομμάτι του κώδικα με όνομα, Solving the economic distribution problem, γίνεται η επίλυση του προβλήματος κατανομής και εκτυπώνεται η λύση. Η επίλυση, γίνεται με την χρήση της συνάρτησης equationsToMatrix, η οποία μετατρέπει το πρόβλημα γραμμικών εξισώσεων, σε σύστημα μορφής πινάκων  $[A,b]$ . Ειδικότερα, δημιουργεί την εξίσωση  $Ax = b$ , όπου  $A$  είναι ένας πίνακας  $N \times N$  με τους συντελεστές της εκάστοτε εξίσωσης, όπου  $N$ , ο αριθμός των γραμμικών εξισώσεων. Το  $x$ , είναι ένα διάνυσμα  $N \times 1$  με τις μεταβλητές των εξισώσεων  $dF$  και  $b$ , ένα διάνυσμα  $N \times 1$  το οποίο περιέχει το αριστερό μέλος της εξίσωσης  $dF$ . Η συνάρτηση αυτή έχει ως όρισμα, τις εξισώσεις συνεργασίας, την εξίσωση διατήρησης της ισχύος (η γραμμική εξίσωση που αναφέραμε παραπάνω) αλλά και τις συμβολικές μεταβλητές του συστήματος.

Για το συγκεκριμένο παράδειγμα οι εξισώσεις συνεργασίας είναι:

$$\begin{aligned}\frac{dF_1}{dP_1} &= 8.078 + 0.00346P_1 = \lambda \\ \frac{dF_2}{dP_2} &= 7.992 + 0.00400P_2 = \lambda \\ \frac{dF_3}{dP_3} &= 8.110 + 0.00948P_3 = \lambda\end{aligned}$$

και η εξίσωση διατήρησης της ισχύος (3.39):

$$P_1 + P_2 + P_3 = 900 \text{ MW}$$

Άρα, ο πίνακας  $A$  και τα διανύσματα  $x$  και  $b$  γίνονται:

$$A = \begin{bmatrix} 0.00346 & 0 & 0 & -1 \\ 0 & 0.00400 & 0 & -1 \\ 0 & 0 & 0.00948 & -1 \\ 1 & 1 & 1 & 0 \end{bmatrix},$$

$$x = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \lambda \end{bmatrix},$$

$$b = \begin{bmatrix} -8.078 \\ -7.992 \\ -8.110 \\ 900 \end{bmatrix}$$

Αφού μετατραπεί το σύστημα, σε μορφή πινάκων, καλείται η συνάρτηση `linsolve`, η οποία λύνει το σύστημα αυτό. Η `linsolve`, δέχεται ως είσοδο το αποτέλεσμα της συνάρτησης `equationsToMatrix`, δηλαδή τον πίνακα  $A$ , και το διάνυσμα  $b$ , και επιστρέφει την λύση του συστήματος. Οι δύο αυτές συναρτήσεις χρησιμοποιούνται για την επίλυση και των επόμενων παραδειγμάτων.

## Solving the economic distribution problem

```
[A,B] = equationsToMatrix([dF1, dF2, dF3, Final_equation], [P1, P2, P3, lamda]);

solution=linsolve(A,B);

fprintf('P1 is equal to: <strong>%.2f MW</strong>',vpa(solution(1)))
fprintf('\nP2 is equal to: <strong>%.2f MW</strong>',vpa(solution(2)))
fprintf('\nP3 is equal to: <strong>%.2f MW</strong>',vpa(solution(3)))
fprintf('\nLamda is equal to: <strong>%.2f</strong>',vpa(solution(4)))
fprintf('\n')
```

Ο χρήστης για να δει το αποτέλεσμα, κατευθύνεται στην καρτέλα EDITOR και πατά το κουμπί RUN. Το αποτέλεσμα παρουσιάζεται παρακάτω.

P1 is equal to: 392.80 MW

P2 is equal to: 366.85 MW

P3 is equal to: 140.35 MW

Lamda is equal to: 9.44

Σύμφωνα με το Παράδειγμα 1 στο κεφάλαιο 3 οι λύσεις ήταν:

$$P_1 = 395.5 \text{ MW}$$

$$P_2 = 363.5 \text{ MW}$$

$$P_3 = 141 \text{ MW}$$

$$\lambda = 9.446 \frac{\text{€}}{\text{kW}}$$

Παρατηρούμε πως το αποτέλεσμα του κώδικα, είναι αρκετά κοντά με την επίλυση του παραδείγματος που γίνεται στο Κεφάλαιο 3. Μικρές αριθμητικές διαφορές είναι λογικές, λόγω στρογγυλοποιήσεων κατά των ορισμό των δεδομένων αλλά και κατά την χρήση των συναρτήσεων equationsToMatrix και linsolve.

Ας υποθέσουμε ένα σενάριο A, στο οποίο η τιμή του αερίου αυξάνεται κατά 10% ενώ η τιμή του μαζούτ μειώνεται κατά 5%. Είναι:

$$\text{Αέριο: } 3.48 * 1.1 = 3.828 \text{ k€/Gcal}$$

$$\text{Μαζούτ: } 3.96 * 0.95 = 3.762 \text{ k€/Gcal}$$

Ο χρήστης κατευθύνεται στο δεύτερο τμήμα του κώδικα και αλλάζει το κόστος της μονάδας 1, cost1, με το νέο κόστος για το αέριο, ενώ για τις μονάδες 2 και 3, αλλάζει τα cost2 και cost3, με το νέο κόστος του μαζούτ αντίστοιχα. Έτσι, ο κώδικας γίνεται:

## Data

```
syms P1 P2 P3 lamda

%Unit_1---->Gas
P1_max=600;
P1_min=150;
H1=165.8 + 2.321*P1 + 0.0005*(P1^2);
cost1=3.828;

%Unit_2---->fuel oil
P2_max=400;
P2_min=100;
H2=78.3 + 2.018*P2 + 0.0005*(P2^2);
cost2=3.762;

%Unit_3---->fuel oil
P3_max=200;
P3_min=50;
H3=39.4 + 2.048*P3 + 0.0012*(P3^2);
cost3=3.762;
```

Το αποτέλεσμα το οποίο παίρνει για το σενάριο αυτό, παρουσιάζεται παρακάτω.

P1 is equal to: **174.30 MW**

P2 is equal to: **521.08 MW**

P3 is equal to: **204.62 MW**

Lamda is equal to: **9.55**



Παρατηρούμε, ότι στο νέο σενάριο όπου το μαζούτ κοστίζει λιγότερο από το αέριο, οι μονάδες 2 και 3, που χρησιμοποιούν μαζούτ ως καύσιμο, παράγουν περισσότερη ενέργεια από ότι η μονάδα 1, που χρησιμοποιεί αέριο. Φαίνεται επίσης, ότι ο συντελεστής  $\lambda$ , έχει μεγαλύτερη τιμή από ότι προηγουμένως. Αυτό σημαίνει ότι το κόστος παραγωγής ενός kW, στο νέο σενάριο, κοστίζει περισσότερο από ότι στο παλιό, παρόλο που η παραγωγή γίνεται κατά κύριο λόγο στις μονάδες που χρησιμοποιούν το φθηνότερο καύσιμο. Ακόμα, στα δυο αυτά σενάρια, δεν συμπεριλάβαμε τους περιορισμούς των λειτουργικών ορίων των μονάδων. Αν είχε γίνει αυτό, το δεύτερο σενάριο δεν θα μπορούσε να πραγματοποιηθεί, καθώς οι μονάδες 2 και 3, λειτουργούν πάνω από το ανώτατο όριο λειτουργίας τους, ενώ η μονάδα 1, κάτω από το κατώτατο όριο λειτουργίας της. Επομένως, η αλλαγή που επιλέξαμε με το +10% (μονάδα 1) και -5% (μονάδες 2,3) είναι μη-εφικτό λόγω περιορισμών που δεν λήφθηκαν υπόψιν).

### 4.3. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 2

Όπως είδαμε και στην προηγούμενη παράγραφο, τα όρια λειτουργίας των μονάδων, έχουν σημαντικό ρόλο στο πρόβλημα οικονομικής κατανομής.

#### Εκφώνηση

Έστω το σύστημα του παραδείγματος 1. Υποθέτουμε, ότι γίνεται μια αύξηση της τιμής του μαζούτ κατά 20%, οπότε η νέα τιμή καυσίμου για τις μονάδες 2 και 3 είναι:

$$3.96 * 1.2 = 4.752 \text{ k€/Gcal}$$

Μονάδα 1:

Μέγιστη Έξοδος:

$$P_{1 \max} = 600 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{1 \min} = 150 \text{ MW}$$

Κόστος Λειτουργίας

Μονάδας:

$$H_1 \left( \frac{\text{Gcal}}{h} \right) = 165.8 + 2.321P_1 + 0.0005P_1^2$$

Καύσιμο: Φυσικό αέριο

Κόστος καυσίμου: 3.48 k€/Gcal

Μονάδα 2:

Μέγιστη Έξοδος:

$$P_{2\max} = 400 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{2\min} = 100 \text{ MW}$$

Κόστος Λειτουργίας

$$H_2 \left( \frac{\text{Gcal}}{h} \right) = 78.3 + 2.018P_2 + 0.0005P_2^2$$

Μονάδας:

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 4.752 k€/Gcal

Μονάδα 3:

Μέγιστη Έξοδος:

$$P_{3\max} = 200 \text{ MW}$$

Ελάχιστη Έξοδος:

$$P_{3\min} = 50 \text{ MW}$$

Κόστος Λειτουργίας

$$H_3 \left( \frac{\text{Gcal}}{h} \right) = 39.4 + 2.048P_3 + 0.0012P_3^2$$

Μονάδας:

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 4.752 k€/Gcal

Χρειάζεται να υπολογιστεί το οικονομικό σημείο λειτουργίας των μονάδων, όταν η ζήτηση του φορτίου είναι 900 MW. Για την επίλυση του παραδείγματος 2 σε ηλεκτρονικό υπολογιστή, χρησιμοποιείται ο αλγόριθμος επαναλήψεων λάμδα.

Ο χρήστης αυτή την φορά, βρίσκεται στον φάκελο με όνομα Economic\_Distribution\_Examples και αναζητά τον υπό – φάκελο με όνομα Example\_2. Μόλις τον ανοίξει, συναντά πέντε αρχεία, εκ των οποίων τα τρία είναι συναρτήσεις, με ονόματα array\_creator.m, constrains.m και equations\_solver.m, όπου καλούνται από το κύριο πρόγραμμα και θα αναλυθούν στην συνέχεια. Επίσης, ο χρήστης συναντά ένα αρχείο xlsx, με όνομα data.xlsx, στο οποίο καταχωρεί τα δεδομένα του παραδείγματος και τέλος συναντά το κύριο m.file, με όνομα Example\_2.m. Ανοίγοντας το Example\_2.m, ο χρήστης βρίσκεται ξανά στο περιβάλλον του λογισμικού MatLAB και βλέπει τον κώδικα του παραδείγματος 2 του οποίου η ανάλυση γίνεται παρακάτω.

Στο πρώτο τμήμα του κώδικα, με όνομα Intro, γίνεται καθαρισμός της μνήμης και του χώρου εργασίας της MatLAB, όπως έγινε και στον κώδικα του παραδείγματος 1.

## Intro

```
clear all
format compact
clc
```

Στο δεύτερο τμήμα του, με όνομα Importing data from excel, ορίζονται τα δεδομένα που έχουν καταχωρηθεί στο αρχείο data.xlsx, το οποίο θα παρουσιαστεί στην συνέχεια. Η διαδικασία αυτή επιτυγχάνεται, με την χρήση της συνάρτησης xlsread, η οποία διαβάζει ένα αρχείο excel και μεταφέρει τα δεδομένα του, στο περιβάλλον της MatLAB. Το αρχείο excel παρουσιάζεται στην συνέχεια στον πίνακα 4.1.

## Importing data from excel

```
Unit_min=xlsread('data.xlsx','Example_2','B2:D2');
Unit_max=xlsread('data.xlsx','Example_2','B3:D3');
Unit_cost=xlsread('data.xlsx','Example_2','B4:D4');
P_R=xlsread('data.xlsx','Example_2','B5');
```

Στο τρίτο τμήμα του κώδικα Declare the other parameters., ορίζονται ως συμβολικές μεταβλητές, οι εξόδοι ισχύος των μονάδων  $P_1, P_2, P_3$  και δημιουργείται ένα διάνυσμα με αυτές καθώς και για το κόστος λειτουργίας της κάθε μονάδας,  $H_1, H_2, H_3$ . Επίσης, ορίζεται ο αριθμός των μονάδων, με την συνάρτηση: length(Unit\_min), η οποία ως είσοδο δέχεται ένα διάνυσμα και επιστρέφει το μήκος του. Αν για παράδειγμα το διάνυσμα Unit\_min είναι ένα διάνυσμα με τέσσερις αριθμούς, τότε η συνάρτηση θα επιστρέψει 4. Το Unit\_min θα προκύψει από το αρχείο excel αυτόματα.

## Declare the other parameters

```
%Symbolic variables for the Output of each unit.
syms P1 P2 P3

%Creating an array with symbolic variables.
P=[P1 P2 P3];

%Creating an array with hourly heat consumption, of each unit.
H=[(165.8 + 2.321*P1 + 0.0005*(P1^2)) (78.3 + 2.018*P2 + 0.0005*(P2^2))
(39.4 + 2.048*P3 + 0.0012*(P3^2))];

%Defining the number of units.
Number_of_units=length(Unit_min);
fprintf('For this excersise we have <strong>%d</strong>
units.\n\n',Number_of_units)

%Calling the array_creator function which generates: P_min, P_max, F(hourly
cost curves) vectors.
[P_min,P_max,P_cost,F]=array_creator(Number_of_units,H,Unit_min,Unit_max,Uni
t_cost);
```

Στο τρίτο τμήμα, καλείται και η πρώτη συνάρτηση του προγράμματος, η `array_creator`.

```
%Calling the array_creator function which generates: P_min, P_max, F(hourly
cost curves) vectors.
[P_min,P_max,P_cost,F]=array_creator(Number_of_units,H,Unit_min,Unit_max,Uni
t_cost);
```

Η συνάρτηση αυτή, δέχεται ως ορίσματα, τον αριθμό των μονάδων `Number_of_units`, το διάνυσμα με το κόστος λειτουργίας, ή, και τα διανύσματα από το `data.xlsx`, που αφορούν τα μέγιστα και ελάχιστα όρια λειτουργίας των μονάδων αλλά και το κόστος καυσίμου. Η έξοδος της συνάρτησης αυτής είναι, ένα διάνυσμα με τα μέγιστα λειτουργικά όρια των μονάδων, ένα με τα ελάχιστα, ένα με τα κόστη των μονάδων και τέλος, ένα διάνυσμα με τις καμπύλες ωριαίου κόστους της κάθε μονάδας. Για να δει ο χρήστης την συνάρτηση αυτή, κατευθύνεται στο αριστερά μέρος του περιβάλλοντος της MatLAB και στο Current Folder, επιλέγει το αρχείο, με όνομα `array_creator.m`. Η συνάρτηση αυτή συναντάται και στα επόμενα παραδείγματα.

```

Function [P_min,P_max,P_cost,F]=array_creator(N,H,Unit_min,Unit_max,Unit_cost
)
%Creating an array for min values, max values and for the hourly cost of
each unit.
for i=1:N
    P_min(i)=Unit_min(i);           %min value array.
    P_max(i)=Unit_max(i);           %max value array.
    P_cost(i)=Unit_cost(i);         %each unit costs.
    F(i)=H(i)*P_cost(i);            %hourly cost of each unit.
end

```

Στο τέταρτο τμήμα του κώδικα, με όνομα lamda iteration algorithm, εφαρμόζεται ο αλγόριθμος επαναλήψεων λάμδα. Στην αρχή, αρχικοποιούνται το όριο ανοχής  $\varepsilon$ , καθώς και η αρχική τιμή του συντελεστή Lagrange,  $\lambda$ . Έπειτα εφαρμόζεται ο αλγόριθμος που παρουσιάστηκε στο διάγραμμα 3.1. Προτού σχολιαστεί ο αλγόριθμος όμως, χρειάζεται να επεξηγηθούν οι άλλες δύο συναρτήσεις τις οποίες ο χρήστης τις βρίσκει και αυτές στο Current Folder, με ονόματα constraints.m και equations\_solver.m.

```

sol=equations_solver(Number_of_units,F,P,Lamda);

```

```

[temp_sol,Total_Power_Production]=constraints(temp_sol,Number_of_units,P_min,
P_max);

```

Η συνάρτηση equations\_solver, δέχεται ως όρισμα τον αριθμό των μονάδων, τα διανύσματα καμπύλης ωριαίου κόστους και ισχύων εξόδου και τέλος τον συντελεστή Lagrange,  $\lambda$ . Ως έξοδο, επιστρέφει το διάνυσμα με την λύση του εκάστοτε συστήματος. Στην ουσία, η συνάρτηση αυτή, αφού πρώτα υπολογίσει τις εξισώσεις συνεργασίας, χρησιμοποιεί την equationsToMatrix και την linsolve, όπως ακριβώς παρουσιάστηκε στην παράγραφο 4.2, λύνοντας, κάθε φορά, το σύστημα εξισώσεων, για το λάμδα το οποίο επιθυμούμε.

```

function [solution]=equations_solver(N,F,P,lamda)

%Cooperation_equations

```

```

for i=1:N
    dF(i)=diff(F(i),P(i))==lamda;           %Creating cooperation equations.
end
% Solving the economic distribution problem.
[A,B] = equationsToMatrix([dF], [P]);
solution=linsolve(A,B);

```

Από την άλλη, η συνάρτηση `constrains.m`, δέχεται ως είσοδο το διάνυσμα, με την λύση από την `equations_solver`, τα διανύσματα με τα όρια λειτουργίας των μονάδων και τέλος τον αριθμό των μονάδων, ενώ επιστρέφει το διάνυσμα με την λύση που ικανοποιεί τους περιορισμούς για τα λειτουργικά όρια των μονάδων και την συνολική παραγόμενη ενέργεια. Πρακτικά, η συνάρτηση αυτή, ελέγχει αν η έξοδος κάποιας μονάδας, βρίσκεται εκτός ορίων λειτουργίας. Αν βρίσκεται εντός των ορίων, δεν κάνει κάποια αλλαγή, ενώ αν βρίσκεται εκτός των ορίων, δίνει στην έξοδο της μονάδας, την μέγιστη ή την ελάχιστη τιμή του ορίου λειτουργίας της, ανάλογα, με το αν υπερβαίνει το μέγιστο όριο ή βρίσκεται κάτω από το ελάχιστο όριο λειτουργίας της. Και στις δύο περιπτώσεις, μετά από τους ελέγχους, υπολογίζει την συνολική παραγόμενη ενέργεια.

```

function[solution,Total_Power_Production]=constrains(solution,N,P_min,P_max)

%Checking if each unit load constrains are fulfilled.
for i=1:N
    if solution(i)>P_max(i)           %If it's more.
        solution(i)=P_max(i);       %give the max possible value.
    elseif solution(i)<P_min(i)       %If it's less.
        solution(i)=P_min(i);       %give the min possible value.
    end
end
Total_Power_Production=vpa(sum(solution)); %Calculate the new produced
power after the power constrains.

```

Οι δύο αυτές συναρτήσεις, χρησιμοποιούνται στα επόμενα παραδείγματα.

Για την επίλυση του αλγορίθμου, αφού έχουν οριστεί το όριο ανοχής και η αρχική τιμή του συντελεστή λάμδα ή Lagrange, ορίζεται ένα διάστημα 100 επαναλήψεων στο οποίο αν ο αλγόριθμος, δεν συγκλίνει στην λύση, τερματίζει.

Στην πρώτη επανάληψη του αλγορίθμου, λύνεται το σύστημα για το αρχικό λάμδα που έχουμε επιλέξει και υπολογίζεται η τιμή της εξίσωση διατήρησης της ενέργειας, `fval`, όπου ουσιαστικά δηλώνει την υπολειπόμενη ενέργεια που χρειάζεται

να παραχθεί. Έπειτα αποθηκεύονται στις προσωρινές μεταβλητές, με όνομα prev\_fval και prev\_lamda, η τιμή της fval και του λάμδα αντίστοιχα. Έπειτα, υπολογίζεται το νέο λάμδα το οποίο είναι μεγαλύτερο κατά 10%.

## Lamda iteration algorithm

```
Error=0.1;Lamda=1;

for j=1:100
    if j==1

        %Solve the economic distribution problem for the first iteration.
        %We call the equation_solver function, to create the cooperation
        equations and then solve them.
        sol=equations_solver(Number_of_units,F,P,Lamda);

        %Calculating the energy conservation equation.
        fval=vpa(P_R - sum(sol(1:Number_of_units)));

        %Define Lamda and fval
        prev_Lamda=Lamda;
        prev_fval=fval;
        Lamda=Lamda*1.1;
```

Στην συνέχεια, το πρόγραμμα περνάει στην δεύτερη επανάληψη. Αρχικά υπολογίζει για το νέο λάμδα, την λύση του συστήματος και την αποθηκεύει σε ένα προσωρινό διάνυσμα με όνομα, temp\_sol. Υπολογίζει έπειτα, την τιμή της εξίσωσης διατήρησης της ισχύος, όπου και αυτή αποθηκεύεται σε προσωρινή μεταβλητή με όνομα temp\_fval. Στο σημείο αυτό, ελέγχεται η συνθήκη, αν η διαφορά της temp\_fval με την prev\_fval, είναι μικρότερη από το όριο ανοχής Error.

```
else
    %Solve the economic distribution problem for higher iterations.
    %We call the equation_solver function, to create the cooperation
    %equations and then solve them for higher Lamda.
    temp_sol=equations_solver(Number_of_units,F,P,Lamda);

    %Calculating the energy conservation equation.
    temp_fval=vpa(P_R - sum(temp_sol(1:Number_of_units)));

    if abs(prev_fval-temp_fval)<Error
```

Αν δεν ικανοποιείται η συνθήκη, τότε επιλέγεται νέο λάμδα, αυτή την φορά με την μέθοδο της προβολής και το πρόγραμμα συνεχίζει στην τρίτη επανάληψη (φαίνεται στο τέλος του κώδικα).

```

else
    %Calculating the new lamda with the Newton-Raphson method.
    temp_Lamda=Lamda;
    Lamda= Lamda-((Lamda-prev_Lamda)/(temp_fval-
    prev_fval))*temp_fval;
    prev_Lamda=temp_Lamda;
    prev_fval=temp_fval;
end
end
end

```

Αν όμως ικανοποιείται, τότε χρησιμοποιείται η συνάρτηση constrains.m, η οποία ελέγχει, αν η λύση υπερβαίνει τα όρια λειτουργίας των μονάδων και έπειτα ελέγχεται αν μετά την χρήση της, η συνολική παραγόμενη ενέργεια είναι περισσότερη ή λιγότερη από την απαιτούμενη.

```

if abs(prev_fval-temp_fval)<Error
    %Checking the load constrain.
    %This constrain function, checks if the load of each unit is
    %between P_min and P_max.

    [temp_sol,Total_Power_Production]=constrains(temp_sol,Number_of_units,P_min,
    P_max);

    %Printing the solution.
    fprintf('For lamda: <strong>%.2f</strong>, the optimal solution
    after the power constrains is:',Lamda)
    for i=1:Number_of_units
        fprintf('\nP%d is equal to: <strong>%.2f
        </strong>',i,vpa(temp_sol(i)))
    end
    fprintf('\n')

    if Total_Power_Production<P_R
        %We need to generate more power.
        fprintf('The produced power, is less than the
        requested(<strong>%.2f MW < %.2f
        MW</strong>).\n',Total_Power_Production,P_R)
        new_temp_fval=P_R-Total_Power_Production;
        new_temp_Lamda=Lamda;
        Lamda=Lamda*1.1;

        %Solving the problem with the new Lamda.
        temp_sol=equations_solver(Number_of_units,F,P,Lamda);

        [temp_sol,Total_Power_Production]=constrains(temp_sol,Number_of_units,P_min,
        P_max);

```



```

temp_fval=P_R-Total_Power_Production;

%Printing the solution.
fprintf('\nFor lamda: <strong>%.2f</strong>, the optimal
solution after the power constrains is:',Lamda)
for i=1:Number_of_units
    fprintf('\nP%d is equal to: <strong>%.2f
    MW</strong>',i,vpa(temp_sol(i)))
end
fprintf('\n')
end
if Total_Power_Production>P_R
    %We need to produce less power.
    fprintf('The produced power, is more than the
    requested(<strong>%.2f MW > %.2f
    MW</strong>).\n',Total_Power_Production,P_R)
    Lamda=Lamda -((Lamda-new_temp_Lamda)/(temp_fval-
    new_temp_fval))*temp_fval;

    %Solving the problem for the new Lamda.
    temp_sol=equations_solver(Number_of_units,F,P,Lamda);
    [temp_sol,Total_Power_Production]=constrains(temp_sol,Number_of_units,P_min,
    P_max);
end

```

Αν, μετά τους περιορισμούς η ενέργεια είναι λιγότερη από την απαιτούμενη επιλέγεται μεγαλύτερο λάμδα, κατά 10% και λύνεται ξανά το σύστημα. Αν με το νέο λάμδα, η ενέργεια είναι περισσότερη από την απαιτούμενη, χρησιμοποιείται γραμμική παρεμβολή για να βρεθεί το λάμδα το οποίο ικανοποιεί την λύση με τους περιορισμούς, όπως ακριβώς παρουσιάστηκε στο Κεφάλαιο 3.

Αν, όμως μετά τους περιορισμούς, η ενέργεια είναι περισσότερη από την απαιτούμενη επιλέγεται μικρότερο λάμδα, με την με γραμμική παρεμβολή.Τέλος, εκτυπώνεται το αποτέλεσμα και τερματίζει το πρόγραμμα.

```

%Printing the solution.

fprintf('\nFor lamda: <strong>%.2f</strong>, the optimal
solution after the power constrains is:',Lamda)

for i=1:Number_of_units
    fprintf('\nP%d is equal to: <strong>%.2f
    MW</strong>',i,vpa(temp_sol(i)))
End

fprintf('\nThe produced power is equal with the requested:
<strong>%.2f MW</strong>\n',Total_Power_Production)

break;

```

Πατώντας το κουμπί RUN, ο χρήστης βλέπει το αποτέλεσμα το οποίο παρουσιάζεται παρακάτω:

For this excersise we have 3 units.

For lamda: **10.41**, the optimal solution after the power constrains is:

P1 is equal to: **600.00 MW**

P2 is equal to: **171.77 MW**

P3 is equal to: **59.07 MW**

The produced power, is less than the requested(**830.84 MW < 900.00 MW**) .

For lamda: **11.45**, the optimal solution after the power constrains is:

P1 is equal to: **600.00 MW**

P2 is equal to: **390.74 MW**

P3 is equal to: **150.31 MW**

The produced power, is more than the requested(**1141.05 MW > 900.00 MW**) .

For lamda: **10.64**, the optimal solution after the power constrains is:

P1 is equal to: **600.00 MW**

P2 is equal to: **220.59 MW**

P3 is equal to: **79.41 MW**

The produced power is equal with the requested: **900.00 MW**

Σύμφωνα με το Παράδειγμα 2 στο κεφάλαιο 3 οι λύσεις ήταν:

$$\begin{aligned}P_1 &= 600 \text{ MW} \\P_2 &= 219.7 \text{ MW} \\P_3 &= 80.3 \text{ MW} \\ \lambda &= 10.64436 \frac{\text{€}}{\text{kW}}\end{aligned}$$

Το βέλτιστο αποτέλεσμα του κώδικα είναι και αυτό πολύ κοντά σε αυτό το οποίο φαίνεται στο Κεφάλαιο 3. Μικρές αριθμητικές διαφορές είναι λογικές, λόγω στρογγυλοποιήσεων κατά των ορισμό των δεδομένων αλλά και κατά την χρήση των συναρτήσεων equationsToMatrix και linsolve.

Ο κώδικας αυτός, όπως προαναφέρθηκε δημιουργήθηκε για να λαμβάνει υπόψη τα λειτουργικά όρια των μονάδων. Έστω ξανά το σενάριο A. Ο χρήστης, αυτή την φορά, ανοίγει από τον φάκελο Example\_2, το αρχείο με όνομα data.xlsx. Μόλις το ανοίξει, βρίσκεται στο πρώτο φύλλο εργασίας, του προγράμματος Excel, με όνομα Example\_2, το οποίο παρουσιάζεται παρακάτω.

Στα κελιά B2, C2 και D2, βρίσκονται τα κάτω όρια λειτουργίας της κάθε μονάδας, ενώ στα κελιά B3, C3 και D3 τα πάνω όρια λειτουργίας. Ταυτόχρονα, στα B4, C4 και D4 βρίσκονται τα κόστη καυσίμου της κάθε μονάδας. Ειδικότερα, η μονάδα 1 η οποία λειτουργεί με αέριο, έχει κόστος καυσίμου  $3.48 \frac{\text{k€}}{\text{Gcal}}$ , ενώ οι μονάδες 2 και 3 που λειτουργούν με μαζούτ, έχουν κόστος καυσίμου  $3.96 * 1.2 = 4.752 \frac{\text{k€}}{\text{Gcal}}$ , καθώς για το παράδειγμα 2, χρησιμοποιούνται τα δεδομένα του παραδείγματος 1, θεωρώντας ότι έχει γίνει αύξηση στην τιμή του μαζούτ κατά 20%.

Πίνακας 4.1: Δεδομένα του παραδείγματος 2, στο αρχείο data.xlsx.

	A	B	C	D
1		Unit 1	Unit 2	Unit 3
2	Min	150	100	50
3	Max	600	400	200
4	Cost	3.48	4.752	4.752
5	PR	900		
6				

Ο χρήστης για να βρεθεί στο Σενάριο A, κάνει τις ακόλουθες αλλαγές. Στο κελί B4, αντικαθιστά την τιμή 3.48 με την 3.828, ενώ στα κελιά C4 και D4, αλλάζει την τιμή 4.752 σε 3.762. Έτσι, το αρχείο έχει την ακόλουθη μορφή:

Πίνακας 4.2: Δεδομένα του Σεναρίου Α, στο αρχείο data.xlsx.

	A	B	C	D
1		Unit 1	Unit 2	Unit 3
2	Min	150	100	50
3	Max	600	400	200
4	Cost	3.828	3.762	3.762
5	PR	900		
6				

Κάνοντας αποθήκευση στο αρχείο data.xlsx και πατώντας RUN στο περιβάλλον της MatLAB, το αποτέλεσμα το οποίο παίρνει ο χρήστης είναι το ακόλουθο.

For lamda: **9.55**, the optimal solution after the power constrains is:

P1 is equal to: **174.30 MW**

P2 is equal to: **400.00 MW**

P3 is equal to: **200.00 MW**

The produced power, is less than the requested(**774.30 MW < 900.00 MW**) .

For lamda: **10.51**, the optimal solution after the power constrains is:

P1 is equal to: **423.83 MW**

P2 is equal to: **400.00 MW**

P3 is equal to: **200.00 MW**

The produced power, is more than the requested(**1023.83 MW > 900.00 MW**) .

For lamda: **10.03**, the optimal solution after the power constrains is:

P1 is equal to: **300.00 MW**

P2 is equal to: **400.00 MW**

P3 is equal to: **200.00 MW**

The produced power is equal with the requested: 900.00 MW.

Παρατηρούμε, πως και σε αυτή την περίπτωση οι μονάδες που χρησιμοποιούν το Μαζούτ ως καύσιμο, παράγουν την περισσότερη ενέργεια σε αντίθεση με την αρχική περίπτωση όπου το αέριο ήταν φθηνότερο.

Συγκρίνοντας, τα αποτελέσματα των παραδειγμάτων 1 και 2 για το σενάριο Α, παρατηρούμε, ότι αν λάβουμε υπόψη μας τα λειτουργικά όρια των μονάδων, το κόστος παραγωγής ενός kW διαφέρει. Η μονάδα 1, που πλέον παράγει ακριβά την ενέργεια, χρειάζεται να λειτουργήσει σε μεγαλύτερη ισχύ προκειμένου να καλύψει τις απαιτήσεις σε ενέργεια, αφού οι μονάδες 2 και 3, λειτουργούν στο μέγιστο της ισχύος τους, σε αντίθεση με το παράδειγμα 1, στο οποίο μπορούσαν να δώσουν στο σύστημα την περισσότερη δυνατή ισχύ.

#### 4.4. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 3

Το παράδειγμα 3, επιλύει το πρόβλημα οικονομικής κατανομής, λαμβάνοντας υπόψη τα λειτουργικά όρια των μονάδων αλλά και τις απώλειες λόγω μεταφοράς.

##### *Εκφώνηση*

Έστω ξανά το σύστημα του παραδείγματος 1. Αυτή την φορά, υποθέτουμε ότι η τρίτη μονάδα βρίσκεται εγκατεστημένη πολύ κοντά στο κέντρο ζήτησης του φορτίου, έτσι ώστε η παραγόμενη ενέργεια, να μεταφέρεται στους καταναλωτές χωρίς απώλειες. Αντίθετα, η πρώτη και δεύτερη μονάδα, βρίσκονται μακριά από το κέντρο κατανάλωσης και έτσι υπάρχουν απώλειες. Αυτές οι απώλειες μεταφοράς, εκφράζονται ως συνάρτηση των  $P_1$  και  $P_2$ , με βάση την ακόλουθη σχέση.

$$P_L = 0.0003P_1^2 - 0.00005 P_1 \cdot P_2 + 0.0001P_2^2 - 0.01P_1 + 0.01P_2 + 4$$

Χρειάζεται να υπολογιστεί η βέλτιστη κατανομή του φορτίου στις μονάδες, όταν η ζήτηση είναι 900 MW.

Ο χρήστης κατευθύνεται στον υπό – φάκελο με όνομα, Example\_3. Μέσα στον υπό – φάκελο, ο χρήστης συναντά έξι αρχεία. Ένα αρχείο xlsx με τα δεδομένα του παραδείγματος, το data.xlsx, τις τρεις γνωστές πλέον συναρτήσεις, equations\_solver.m, constrains.m και array\_creator.m, το κύριο αρχείο του παραδείγματος, Example\_3.m και μια νέα συνάρτηση, η equations\_creator.m η οποία θα αναλυθεί στην συνέχεια.

Ο χρήστης ανοίγει το αρχείο Example\_3.m. Στο πρώτο τμήμα του κώδικα, με όνομα Intro, γίνεται καθαρισμός της μνήμης και του χώρου εργασίας της MatLAB, όπως έγινε και στα δύο προηγούμενα παραδείγματα.

## Intro

```
format compact
clear all
clc
```

Στο δεύτερο τμήμα του, με όνομα Importing data from excel, ορίζονται τα δεδομένα που έχουν καταχωρηθεί στο αρχείο data.xlsx. Τα δεδομένα του παραδείγματος 3, αποθηκεύονται στο data.xlsx όπως ακριβώς παρουσιάστηκε και στο παράδειγμα 2.

## Importing data from excel

```
Unit_min=xlsread('data.xlsx','Example_3','B2:D2');
Unit_max=xlsread('data.xlsx','Example_3','B3:D3');
Unit_cost=xlsread('data.xlsx','Example_3','B4:D4');
P_R=xlsread('data.xlsx','Example_3','B5');
```

Στο τρίτο τμήμα του κώδικα με όνομα, Declare the other parameters, ορίζονται ως συμβολικές μεταβλητές, οι εξόδοι ισχύος των μονάδων  $P_1$ ,  $P_2$ ,  $P_3$  και δημιουργείται ένα διάνυσμα στο οποίο αποθηκεύονται με όνομα P.

## Declare the other parameters

```
%Symbolic variables for the Output of each unit
syms P1 P2 P3

%Creating an array with symbolic variables.
P=[P1 P2 P3];
```

Στην συνέχεια, δημιουργείται ένα διάνυσμα, το  $P_L$ , το οποίο εκφράζει τις απώλειες λόγω μεταφοράς. Υπενθυμίζεται ότι, η μονάδα 3, βρίσκεται δίπλα από το κέντρο κατανάλωσης, συνεπώς δεν έχει απώλειες.

```
%Creating an array with Transmission losses.
PL=0.0003*(P1^2)-0.00005*(P1*P2)+0.0001*(P2^2)-0.01*P1+0.01*P2+4;
```

Έπειτα, δημιουργείται ένα διάνυσμα για το κόστος λειτουργίας της κάθε μονάδας,  $H_1, H_2, H_3$ , με όνομα H, και καλούνται οι συναρτήσεις array\_creator και equations\_creator.

```
%Calling the array_creator function which generates: P_min, P_max, F(hourly
cost curves).
[P_min,P_max,P_cost,F]=array_creator(Number_of_units,H,Unit_min,Unit_max,Unit
t_cost);
%Calling the equation_creator function which generates:
dF(Differential_costs), dP(Differential_losses), Pf(Penalty_factors).
[dF,dP,Pf]=equations_creator(Number_of_units,P,F,PL);
```

Όσον αφορά την συνάρτηση equations\_creator, δέχεται ως όρισμα, τον αριθμό των μονάδων, το διάνυσμα των συμβολικών μεταβλητών P, το διάνυσμα F, με τις καμπύλες κόστους λειτουργίας των μονάδων από την array\_creator, καθώς και το

διάνυσμα με τις απώλειες,  $P_L$ . Ως έξοδο, επιστρέφει, τα διαφορικά κόστη παραγωγής των μονάδων,  $\frac{df_i}{dP_i}$ , τις διαφορικές απώλειες,  $\frac{dP_L}{dP_i}$  και τέλος τους συντελεστές ποινής  $PF_i$ .

```
function [dF,dP,PF]=equations_creator(N,P,F,PL)
%Creating an array for each: dF(Differential_costs),
dP(Differential_losses), Pf(Penalty_factors).
for i=1:N
    %Differential_Costs array.
    dF(i)=diff(F(i),P(i));
    %Differential_losses array.
    dP(i)=diff(PL,P(i));
    %Penalty_factors array.
    PF(i)=1/(1-dP(i));
end
```

Στο τέταρτο τμήμα του κώδικα, με όνομα Solve the economic distribution problem, ξεκινά η επίλυση του προβλήματος. Αυτή την φορά, θα εφαρμοστεί ο αλγόριθμος του διαγράμματος 3.2. Έτσι για την πρώτη επανάληψη, ορίζονται στο διάνυσμα,  $PP$  οι αρχικές τιμές για τις εξόδους των μονάδων και με την χρήση της συνάρτησης της MatLAB, subs, υπολογίζονται οι απώλειες για τις τιμές εξόδου που έχουμε ορίσει και αποθηκεύονται, στην μεταβλητή  $PPL$ . Η συνάρτηση subs, επιστρέφει το αποτέλεσμα μιας εξίσωσης για μια τιμή της μεταβλητής που έχει ορίσει ο χρήστης. Στην προκειμένη περίπτωση η subs, υπολογίζει τις απώλειες αντικαθιστώντας τις τιμές του διανύσματος  $PP$  στην εξίσωση  $P_L$ . Έπειτα υπολογίζεται η συνολική ζήτηση, μαζί με τις απώλειες λόγω μεταφοράς,  $PD$  και ταυτόχρονα, υπολογίζονται οι συντελεστές ποινής, με χρήση της subs, για τις αρχικές τιμές των ισχύων εξόδου,  $PP$ .

## Solve the economic distribution problem for the first iteration

```
%We choose the Data bellow randomly.
PP=[350 370 190];
PPL=vpa(subs(PL,[P],[PP]));
PD=P_R+PPL;

%Calculating the Penalty factors.
for i=1:Number_of_units
    PPF(i)=vpa(1/(1-subs(dP(i),[P],[PP])));
end
```



Επομένως, ξεκινά η επίλυση του προβλήματος για την πρώτη επανάληψη. Αφού, αρχικοποιείται ο συντελεστής Lagrange, χρησιμοποιούνται οι συναρτήσεις `equations_solver` και `constrains`, ώστε να επιλυθεί το πρόβλημα για την συγκεκριμένη τιμή του λάμδα, αλλά ταυτόχρονα να ισχύει ο περιορισμός για τα λειτουργικά όρια των μονάδων. Έτσι, υπολογίζεται η τιμή της εξίσωσης διατήρησης της ενέργειας, `fval` και εκτυπώνονται τα αποτελέσματα.

```
%Solving the problem for the first iteration.
Lamda=11;
[solution]=equations_solver(Number_of_units,dF,PPF,P,Lamda);
[solution,Total_Power_Production]=constrains(solution,Number_of_units,P_min,
P_max);
fval=vpa((PD - sum(solution(1:Number_of_units))),4);

%Printing
fprintf('For the <strong>1</strong> iteration we choose:')
for i=1:Number_of_units
    fprintf('\nP%d: %.2f MW',i,PP(i))
    if i==Number_of_units
        fprintf('\nlamda: %.2f',Lamda)
    end
end

fprintf('\n\nFor lamda: <strong>%.3f</strong>, the optimal solution after
the power constrains is:',Lamda)
for i=1:Number_of_units
    fprintf('\nP%d is equal to: %.2f MW',i,vpa(solution(i)))
    if i==Number_of_units
        fprintf('\nF value is equal to: %.2f MW',fval)
        fprintf('\n')
    end
end
```

Μετά τον ορισμό των παραπάνω, ορίζεται μια προσωρινή τιμή για την εξίσωση διατήρησης της ενέργειας, `temp_fval`, ίση με έναν μεγάλο αριθμό, έστω το άπειρο και δημιουργείται μια σειρά επαναλήψεων ως εξής. Για όσο, η εξίσωση διατήρησης της ενέργειας, `temp_fval`, δεν είναι μηδέν, το πρόβλημα οικονομικής κατανομής του φορτίου, θα λύνεται με την μέθοδο του παραδείγματος 2. Πιο συγκεκριμένα, εάν η παραγόμενη ενέργεια είναι μικρότερη από την ζητούμενη, το λάμδα παίρνει τιμή μεγαλύτερη κατά 10%, λύνεται ξανά το πρόβλημα, ελέγχεται ο περιορισμός των ορίων λειτουργίας και υπολογίζεται εκ' νέου η εξίσωση διατήρησης της ενέργειας. Τέλος, εκτυπώνεται η λύση. Εάν, όμως η παραγόμενη ενέργεια, είναι μεγαλύτερη από την απαιτούμενη, υπολογίζεται το νέο λάμδα, με γραμμική παρεμβολή και ακολουθείται η ίδια διαδικασία για το νέο λάμδα.

```

%Checking[...]
temp_fval=inf;
while abs(temp_fval)>0
    if Total_Power_Production<PD
        prev_lamda=Lamda;
        Lamda=Lamda*1.1;
        [solution]=equations_solver(Number_of_units,dF,PPF,P,Lamda);

[solution,Total_Power_Production]=constrains(solution,Number_of_units,P_min,
P_max);
        temp_fval=vpa((PD - sum(solution(1:Number_of_units))),4);

        %Printing
        fprintf('-----')
        fprintf('\n')
        fprintf('\nThe total production of power is less than the needed
one. So we choose a higher lamda')
        fprintf('\nFor lamda: <strong>%.3f</strong>, the optimal solution
after the power constrains is:',Lamda)
        for i=1:Number_of_units
            fprintf('\nP%d is equal to: %.2f MW',i,vpa(solution(i)))
            if i==Number_of_units
                fprintf('\nF value is equal to: %.2f MW',temp_fval)
                fprintf('\n')
            end
        end
    end
    if Total_Power_Production>PD
        Lamda= Lamda - ((Lamda-prev_lamda)/(temp_fval-fval))*temp_fval;
        [solution]=equations_solver(Number_of_units,dF,PPF,P,Lamda);

[solution,Total_Power_Production]=constrains(solution,Number_of_units,P_min,
P_max);
        temp_fval=vpa((PD - sum(solution(1:Number_of_units))),4);

        if abs(temp_fval)< 0.000001
            temp_fval=0;
        end

        %Printing
        fprintf('-----')
        fprintf('\n')
        fprintf('\nThe total production of power, is more than the needed.
So we choose a smaller lamda')
        fprintf('\nFor Lamda: <strong>%.3f</strong>, the optimal solution
after the power constrains is:',Lamda)
        for i=1:Number_of_units
            fprintf('\nP%d is equal to: %.2f MW',i,vpa(solution(i)))
            if i==Number_of_units
                fprintf('\nF value is equal to: %.2f MW',temp_fval)
                fprintf('\n')
            end
        end
    end
    if temp_fval==0
        break;
    end
end
end

```

Η παραπάνω διαδικασία, γίνεται όσο το πρόγραμμα βρίσκεται στην πρώτη επανάληψη. Για τις επαναλήψεις μεγαλύτερες της πρώτης, δημιουργήθηκε το πέμπτο τμήμα του κώδικα με όνομα, Solving the problem for iteration >1. Πρώτα, ορίζεται ξανά μια βοηθητική μεταβλητή, η last\_sol, με τιμή πολύ μεγάλη, έστω άπειρο, για το άθροισμα της παραγόμενης ενέργειας αλλά και μια μεταβλητή j, για τις επαναλήψεις. Αυτή την φορά, δεν έχει οριστεί μεταβλητή για το όριο ανοχής, αλλά θεωρούμε ότι η τιμή της είναι 1. Έτσι, δημιουργήθηκε μια συνθήκη επαναλήψεων στην οποία, για όσο η διαφορά της βοηθητικής μεταβλητής, last\_sol με την προηγούμενη λύση, solution, είναι μεγαλύτερη του 1, κάνε την παρακάτω διαδικασία. Αυτή η συνθήκη δημιουργήθηκε, ώστε το πρόγραμμα να σταματά, όταν συγκλίνει στην λύση.

## Solving the problem for iteration>1

```
last_sol=inf;  
  
j=2;  
  
while abs(sum(last_sol)-sum(solution))>1
```

Εισάγονται αρχικά, όλα τα δεδομένα της προηγούμενης επανάληψης και υπολογίζονται ξανά οι συντελεστές ποινής.

```
last_sol=solution;  
  
last_fval=temp_fval;  
  
%We use the data from the last iteration.  
PP=last_sol;  
PPL=vpa(subs(PL,[P],[PP']));  
PD=P_R+PPL;  
  
%Calculating the Penalty factors.  
for i=1:Number_of_units  
  
    PPF(i)=vpa(1/(1-substituted(dP(i),[P],[PP'])));  
  
end
```

Ορίζεται ξανά μια τιμή για το λάμδα, έστω 11. Έτσι, επαναλαμβάνεται η διαδικασία της πρώτης επανάληψης έως ότου ικανοποιηθεί το κριτήριο τερματισμού, δηλαδή η διαφορά της παραγόμενης ενέργειας των δύο τελευταίων επαναλήψεων να είναι μικρότερη του 1.

Η διαφορά με το κομμάτι κώδικα της πρώτης επανάληψης, είναι ότι στις επόμενες επαναλήψεις αποθηκεύει εκ' νέου τα νέα αποτελέσματα και τα συγκρίνει με την προηγούμενη της. Αν για παράδειγμα, ο κώδικας βρίσκεται στην δεύτερη επανάληψη, υπολογίζει τις εξόδους των μονάδων και συνεχίζει στην τρίτη επανάληψη, με την οποία συγκρίνει τα αποτελέσματα της δεύτερης. Αυτό επιτυγχάνεται με την χρήση βοηθητικών μεταβλητών και διανυσμάτων, για την εξίσωση διατήρησης της ενέργειας και για τις εξόδους ισχύος των μονάδων.

```
%Solving the problem.
Lamda=11;
[solution]=equations_solver(Number_of_units,dF,PPF,P,Lamda);

[solution,Total_Power_Production]=constrains(solution,Number_of_units,P_min,
P_max);
fval=vpa((PD - sum(solution(1:Number_of_units))),4);

%Printing
fprintf('-----')
fprintf('\n-----')
fprintf('\n-----')
fprintf('\nFor the <strong>%.d</strong> iteration we choose:',j)
for i=1:Number_of_units
    fprintf('\nP%d: %.2f MW',i,PP(i))
    if i==Number_of_units
        fprintf('\nlamda: %.2f',Lamda)
    end
end
fprintf('\n\nFor Lamda: <strong>%.3f</strong>, the optimal solution
after the power constrains is:',Lamda)
for i=1:Number_of_units
    fprintf('\nP%d is equal to: %.2f MW',i,vpa(solution(i)))
    if i==Number_of_units
        fprintf('\nF value is equal to: %.2f MW',fval)
        fprintf('\n')
    end
end

%Checking[...]
temp_fval=inf;
while abs(temp_fval)>0
    if Total_Power_Production<PD
        prev_lamda=Lamda;
        Lamda=Lamda*1.1;
        [solution]=equations_solver(Number_of_units,dF,PPF,P,Lamda);
    end
    [solution,Total_Power_Production]=constrains(solution,Number_of_units,P_min,
P_max);
end
```

```

temp_fval=vpa((PD - sum(solution(1:Number_of_units))),4);

%Printing
fprintf('-----
-----')
fprintf('\nThe total production of power is less than the needed
one. So we choose a higher lamda')
fprintf('\nFor lamda: <strong>%.3f</strong>, the optimal
solution after the power constrains is:',Lamda)
for i=1:Number_of_units
    fprintf('\nP%d is equal to: %.2f MW',i,vpa(solution(i)))
    if i==Number_of_units
        fprintf('\nF value is equal to: %.2f MW',temp_fval)
        fprintf('\n')
    end
end
end
if Total_Power_Production>PD
    Lamda= Lamda - ((Lamda-prev_lamda)/(temp_fval-fval))*temp_fval;
    [solution]=equations_solver(Number_of_units,dF,PPF,P,Lamda);

[solution,Total_Power_Production]=constrains(solution,Number_of_units,P_min,
P_max);
    temp_fval=vpa((PD - sum(solution(1:Number_of_units))),4);

    if abs(temp_fval)< 0.000001
        temp_fval=0;
    end

    %Printing
    fprintf('-----
-----')
    fprintf('\nThe total production of power, is more than the
needed. So we choose a smaller lamda')
    fprintf('\nFor Lamda: <strong>%.3f</strong>, the optimal
solution after the power constrains is:',Lamda)
    for i=1:Number_of_units
        fprintf('\nP%d is equal to: %.2f MW',i,vpa(solution(i)))
        if i==Number_of_units
            fprintf('\nF value is equal to: %.2f MW',temp_fval)
            fprintf('\n')
        end
    end
end
if temp_fval==0
    break;
end
end
j=j+1;
end
end

```

Αν ο χρήστης πατήσει το κουμπί RUN, στην καρτέλα του EDITOR, το αποτέλεσμα το οποίο θα πάρει είναι το ακόλουθο.

For this excersise we have 3 units.

For the 1 iteration we choose:

P1: 350.00 MW

P2: 370.00 MW

P3: 190.00 MW

lamda: 11.00

For lamda: **11.000**, the optimal solution after the power constrains is:

P1 is equal to: 266.21 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: 81.95 MW

-----

The total production of power is less than the needed one. So we choose a higher lamda

For lamda: **12.100**, the optimal solution after the power constrains is:

P1 is equal to: 524.93 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: -176.77 MW

-----

The total production of power, is more than the needed. So we choose a smaller lamda

For Lamda: **11.348**, the optimal solution after the power constrains is:

P1 is equal to: 348.17 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: 0.00 MW

-----

-----

For the **2** iteration we choose:

P1: 348.17 MW

P2: 400.00 MW

P3: 200.00 MW

lamda: 11.00

For Lamda: **11.000**, the optimal solution after the power constrains is:

P1 is equal to: 274.43 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: 75.49 MW

---

The total production of power is less than the needed one. So we choose a higher lamda

For lamda: **12.100**, the optimal solution after the power constrains is:

P1 is equal to: 533.98 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: -184.06 MW

---

The total production of power, is more than the needed. So we choose a smaller lamda

For Lamda: **11.320**, the optimal solution after the power constrains is:

P1 is equal to: 349.92 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: 0.00 MW

---

---

For the **3** iteration we choose:

P1: 349.92 MW

P2: 400.00 MW

P3: 200.00 MW

lamda: 11.00

For Lamda: **11.000**, the optimal solution after the power constrains is:

P1 is equal to: 271.10 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: 79.13 MW

---

The total production of power is less than the needed one. So we choose a higher lamda

For lamda: **12.100**, the optimal solution after the power constrains is:

P1 is equal to: 530.31 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: -180.08 MW

---

The total production of power, is more than the needed. So we choose a smaller lamda

For Lamda: **11.336**, the optimal solution after the power constrains is:

P1 is equal to: 350.24 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: 0.00 MW



Σύμφωνα με το Παράδειγμα 3 στο κεφάλαιο 3 οι λύσεις ήταν:

$$\begin{aligned}P_1 &= 350.24 \text{ MW} \\P_2 &= 400 \text{ MW} \\P_3 &= 200 \text{ MW} \\ \lambda &= 11.328 \frac{\text{€}}{\text{kW}}\end{aligned}$$

Παρατηρούμε ότι, η βέλτιστη λύση του προβλήματος είναι ίδια με αυτή του κεφαλαίου 3. Επίσης, φαίνεται ότι εφαρμόστηκε το κριτήριο τερματισμού που ορίστηκε προηγουμένως. Η λύση της δεύτερης επανάληψης, είναι σχεδόν ίδια με αυτή της βέλτιστης. Αν η αριθμητική διαφορά τους ήταν μεγαλύτερη του 1, τότε το πρόγραμμα θα έκανε άλλη μια επανάληψη.

Έστω, ένα σενάριο B, στο οποίο η ζήτηση αυξάνεται στα 920 MW. Ο χρήστης ανοίγει το αρχείο, data.xlsx και στο κελί, B5, πληκτρολογεί την τιμή 920 για το PR. Αποθηκεύοντας το αρχείο και πατώντας ξανά το RUN, προκύπτει η παρακάτω βέλτιστη λύση, στην τέταρτη επανάληψη.

For the 4 iteration we choose:

P1: 374.67 MW

P2: 400.00 MW

P3: 200.00 MW

lamda: 11.00

For Lamda: **11.000**, the optimal solution after the power constrains is:

P1 is equal to: 224.17 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: 150.70 MW

-----

The total production of power is less than the needed one. So we choose a higher lamda

For lamda: **12.100**, the optimal solution after the power constrains is:

P1 is equal to: 478.69 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: -103.81 MW

-----

The total production of power, is more than the needed. So we choose a smaller lamda

For Lamda: **11.651**, the optimal solution after the power constrains is:

P1 is equal to: 374.87 MW

P2 is equal to: 400.00 MW

P3 is equal to: 200.00 MW

F value is equal to: 0.00 MW

Πάλι αυτή την φορά, το πρόγραμμα σταμάτησε όταν υπήρξε σύγκλιση στην λύση. Παρατηρείται, ότι η προηγούμενη βέλτιστη λύση, με την νέα, έχουν διαφορά μικρότερη της μονάδας και πως στο σενάριο B, ο συντελεστής Lagrange, είναι μεγαλύτερος από ότι του παραδείγματος 3. Αυτό σημαίνει ότι η παραγωγή μιας μονάδας ενέργειας κοστίζει περισσότερο στο σενάριο B, κάτι το οποίο είναι λογικό, καθώς η μονάδα 1, χρειάζεται να παράξει περισσότερη ενέργεια. Επίσης παρατηρείται πως ανάλογα τα δεδομένα, επιτυγχάνεται πιο αργή ή πιο γρήγορη σύγκλιση.

## 4.5. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 4

Το παράδειγμα 4, ασχολείται με την ένταξη μονάδων παραγωγής, ως μέρος της βελτιστοποίησης του προβλήματος οικονομικής κατανομής.

### Εκφώνηση

Έστω, μια εταιρεία παραγωγής ενέργειας η οποία διαθέτει τις παρακάτω πετρελαϊκές μονάδες. Κάποια στιγμή το φορτίο της εταιρείας είναι  $550 MW$ . Ζητείται η μονάδα ή ο συνδυασμός των μονάδων, που πρέπει να λειτουργήσουν, ώστε να τροφοδοτήσουν το δίκτυο κατά τον πλέον οικονομικό τρόπο.

#### Μονάδα 1:

Μέγιστη Έξοδος:

$$P_{1\max} = 600 MW$$

Ελάχιστη Έξοδος:

$$P_{1\min} = 150 MW$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_1 \left( \frac{Gcal}{h} \right) = 122.5 + 1.729P_1 + 0.00034P_1^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

#### Μονάδα 2:

Μέγιστη Έξοδος:

$$P_{2\max} = 400 MW$$

Ελάχιστη Έξοδος:

$$P_{2\min} = 100 MW$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_2 \left( \frac{Gcal}{h} \right) = 67.7 + 1.714P_2 + 0.00042P_2^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

#### Μονάδα 3:

Μέγιστη Έξοδος:

$$P_{3\max} = 200 MW$$

Ελάχιστη Έξοδος:

$$P_{3\min} = 50 MW$$

Κόστος Λειτουργίας  
Μονάδας:

$$H_3 \left( \frac{Gcal}{h} \right) = 20.4 + 2.088P_3 + 0.0012P_3^2$$

Καύσιμο: Μαζούτ

Κόστος καυσίμου: 3.96 k€/Gcal

Ο χρήστης κατευθύνεται στον υπό – φάκελο με όνομα, Example\_4. Εκεί βρίσκει πέντε αρχεία, τα δεδομένα στην μορφή που παρουσιάστηκαν προηγουμένως στο αρχείο data.xlsx, την συνάρτηση array\_creator.m, η οποία είναι ίδια με τα προηγούμενα παραδείγματα, τις συναρτήσεις equations\_solver.m και constrains.m, στις οποίες έχουν γίνει μικρές τροποποιήσεις και αναλύονται παρακάτω και τέλος το αρχείο του παραδείγματος Example\_4.m.

Ανοίγοντας το Example\_4.m, ο χρήστης συναντά τον κώδικα για την επίλυση του παραδείγματος 4. Το πρώτο και δεύτερο τμήμα του κώδικα είναι ίδιο με τα προηγούμενα παραδείγματα. Στο τρίτο κομμάτι, Declare the other parameters, ορίζονται ως συμβολικές μεταβλητές, οι εξόδοι ισχύος των μονάδων καθώς και το διάνυσμα τους, P. Ακόμα, δημιουργείται το διάνυσμα H, με το κόστος λειτουργίας της κάθε μονάδας, ορίζεται ο αριθμός των μονάδων και στην συνέχεια χρησιμοποιείται η συνάρτηση array\_creator, προκειμένου να ορίσουμε την μέγιστη και ελάχιστη τιμή, του ορίου ισχύος λειτουργίας της κάθε μονάδας, του κόστους καυσίμου, αλλά και την καμπύλη ωριαίου κόστους.

Στο τέταρτο τμήμα του κώδικα, με όνομα defining the combinations, γίνεται καθορισμός των συνδυασμών των μονάδων. Αρχικά, βρίσκεται ο αριθμός των συνδυασμών ο οποίος είναι  $2^{\text{αριθμός των μονάδων}}$  και ταυτόχρονα, δημιουργείται ένας πίνακας με όνομα all\_combinations, στον οποίο ορίζουμε όλους τους δυνατούς συνδυασμούς, χρησιμοποιώντας τον αριθμό 1 όταν η μονάδα είναι ενεργή, ενώ το 0, όταν είναι ανενεργή. Στην συνέχεια, δημιουργούνται δυο κενά διανύσματα με όνομα current\_machine και all\_machines\_combination. Έπειτα, ο αλγόριθμος διαβάζει κάθε μια από τις γραμμές του πίνακα με τους συνδυασμούς και όταν συναντά τον αριθμό 1, τότε εισάγει στο βοηθητικό διάνυσμα, current\_machine, την μονάδα αυτή, ενώ όταν συναντά τον αριθμό μηδέν, εισάγει στο βοηθητικό διάνυσμα το μηδέν. Έτσι, μόλις τελειώνει ο εκάστοτε συνδυασμός, αποθηκεύονται τα στοιχεία του current\_machine στο βοηθητικό διάνυσμα, all\_machines\_combination. Στην ουσία, μετατρέπεται κατά αυτόν τον τρόπο, ο πίνακας με τους συνδυασμούς των μονάδων, σε πίνακα σε συνδυασμό των ισχύων εξόδου των μονάδων. Για παράδειγμα, αν στον αρχικό πίνακα υπάρχει ο συνδυασμός, [1 0 1] μετασχηματίζεται σε  $[P_1 \ 0 \ P_3]$ .

## Defining the combinations

```
%Defining the number of units.
Number_of_units=length(Unit_min);

%Defining the number of combinations.
N_combinations=(2^Number_of_units);

fprintf('We have <strong>%d</strong> combinations of collaborated
units.\n',N_combinations)

%Defining all the combinations.
all_combinations=[[0 0 0];
    [1 0 0];
    [0 1 0];
    [0 0 1];
    [1 1 0];
    [1 0 1];
    [0 1 1];
    [1 1 1]];
current_machine=[];
all_machines_combination=[];

%Creating an array with the P, instead of 0-1.

%For example [1 0 1] ---> [P1 0 P3]

for j=1:N_combinations
    current_combination=all_combinations(j,1:3);

    for j=1:length(current_combination)
        if current_combination(j)==1
            current_machine=[current_machine P(j)];
        else
            current_machine=[current_machine 0];
        end
        if j==length(current_combination)
            all_machines_combination=[all_machines_combination;
                current_machine];
            current_machine=[];
        end
    end

end

fprintf('These are:\n')
for j=1:length(all_machines_combination)
    fprintf('\t')
    fprintf('%s\t',all_machines_combination(j,1:3))
    fprintf(']')
    fprintf('\n')
end
```

Έτσι, ο χρήστης δεν χρειάζεται να πληκτρολογήσει τις εξόδους ισχύος της κάθε μονάδας σε πίνακα, παρά μόνο τους συνδυασμούς με 0 και 1. Το αποτέλεσμα που δίνει ο κώδικας για τους συνδυασμούς είναι:

We have **8** combinations of collaborated units.

These are:

[	0	0	0	]
[	P1	0	0	]
[	0	P2	0	]
[	0	0	P3	]
[	P1	P2	0	]
[	P1	0	P3	]
[	0	P2	P3	]
[	P1	P2	P3	]

Στο πέμπτο και τελευταίο τμήμα του κώδικα, γίνεται επίλυση του προβλήματος ένταξης των μονάδων. Ειδικότερα, χρησιμοποιούνται οι μεθοδολογίες και οι αλγόριθμοι που παρουσιάστηκαν στα προηγούμενα παραδείγματα.

Αρχικά, ο κώδικας θα επαναλαμβάνεται έως ότου επεξεργαστούν όλοι οι συνδυασμοί. Στο πρώτο κομμάτι του, ελέγχονται ποιες μονάδες υπάρχουν σε κάθε συνδυασμό. Στην ουσία, δημιουργούνται σε κάθε επανάληψη τα παρακάτω κενά, βοηθητικά διανύσματα, με όνομα: `all_solution_cost`, `new_current_combination`, `new_fvalue` και `j_value`. Στο πρώτο βοηθητικό διάνυσμα, θα αποθηκεύονται τα συνολικά κόστη λειτουργίας από όλες τις μονάδες του κάθε συνδυασμού, ενώ στο δεύτερο και τρίτο βοηθητικό διάνυσμα, θα αποθηκεύονται η ισχύς και το κόστος λειτουργίας των μονάδων που λειτουργούν σε κάθε συνδυασμό. Το διάνυσμα `j_value`, δημιουργήθηκε μόνο για να μετράει την τιμή που παίρνει η μεταβλητή `j`. Ακόμα, στα διανύσματα `new_current_combination` και `new_fvalue`, θα αποθηκεύονται μόνο τα στοιχεία των μονάδων που θα λειτουργούν κάθε φορά. Για παράδειγμα, για τον συνδυασμό  $[P_1 \ 0 \ P_3]$ , τα διανύσματα `new_current_combination` και `new_fvalue`, θα γίνουν  $[P_1 \ P_3]$  και  $[F_1 \ F_3]$  αντίστοιχα. Μια επιπλέον, χρησιμότητα των διανυσμάτων αυτών είναι για να γίνεται ο έλεγχος των περιορισμών για των ορίων λειτουργίας των μονάδων.

```

%For the solutions we use the same technics as the example_2 and example_3.
all_solution_costs=[];
for i=1:N_combinations
    current_combination=all_machines_combination(i,1:3);
    new_current_combination=[];
    new_f_value=[];
    j_value=[];
    for j=1:length(current_combination)
        if current_combination(j)==0
            new_current_combination=[new_current_combination []];
            new_f_value=[new_f_value []];
        else
            new_current_combination=[new_current_combination
current_combination(j)];
            new_f_value=[new_f_value F(j)];
            j_value=[j_value j];
        end
    end
end

```

Έπειτα, ο κώδικας χωρίζεται σε τέσσερις περιπτώσεις. Η πρώτη περίπτωση, ελέγχει αν δεν υπάρχει καμία μονάδα στον συνδυασμό, η δεύτερη, αν υπάρχει μόνο μια, η τρίτη, αν υπάρχουν δύο και τέλος η τέταρτη αν υπάρχουν και οι τρεις μονάδες στον συνδυασμό.

Στην πρώτη περίπτωση, το πρόγραμμα επιστρέφει ότι η λύση δεν είναι εφικτή, καθώς δεν υπάρχουν μονάδες για να καλύψουν την ζήτηση.

```

%Checking if there is no Unit.
if isempty(new_current_combination)
    fprintf('\nFor the <strong>%d</strong> combination: ',i)
    fprintf('<strong>[ ')
    fprintf(' %s',current_combination)
    fprintf(' ]</strong>')
    fprintf('\n')
    fprintf('\nThe solution <strong>is not possible</strong>.\n')
    all_solution_costs=[all_solution_costs nan];
end

```

Στην δεύτερη περίπτωση, ελέγχεται αν η απαιτούμενη ζήτηση, βρίσκεται εντός ορίων λειτουργίας. Αν βρίσκεται, τότε με την συνάρτηση subs, γίνεται επίλυση της εξίσωσης για την απαιτούμενη ενέργεια. Αν όχι, το πρόγραμμα επιστρέφει ότι δεν είναι εφικτή η λύση.

```

%Checking if there is one Unit.
elseif length(new_current_combination)==1
    fprintf('\nFor the <strong>%d</strong> combination: ',i)
end

```

```

fprintf('<strong>[ ')
fprintf(' %s',current_combination)
fprintf(' ]</strong>')
fprintf('\n')

if (Unit_min(j_value)<=P_R) && (P_R<=Unit_max(j_value))
    solution_cost=subs(F(j_value),P(j_value),P_R);
    fprintf('\nThe solution is possible and its cost is:
    <strong>%.3f</strong>\n',solution_cost)
    all_solution_costs=[all_solution_costs solution_cost];
else
    fprintf('\nThe solution <strong>is not possible</strong>.\n')
    all_solution_costs=[all_solution_costs nan];
end

```

Στην τρίτη περίπτωση, χρησιμοποιείται ο αλγόριθμος επαναλήψεων λάμδα, προκειμένου να βρεθεί η βέλτιστη κατανομή φορτίου στις δύο μονάδες. Στην συγκεκριμένη περίπτωση, χρειάζεται να χρησιμοποιηθεί η συνάρτηση clear, ώστε να διαγραφούν από την μνήμη του λογισμικού οι μεταβλητές new\_temp\_fval και new\_temp\_lamda καθώς το πρόγραμμα θα ελέγχει κάθε έναν από τους συνδυασμούς συνεχόμενα.

```

%Solving the economic distribution problem for 2 Units.
Error=0.1;
Lamda=1;
clear new_temp_fval new_temp_Lamda
for k=1:N_combinations
    %Solving the problem for the first iteration.
    if k==1

sol=vpa(equations_solver_example_4(length(new_current_combination),new_f_value,new_current_combination,Lamda));
        fval=vpa(P_R - sum(sol(1:length(sol))));

        %Define lamda and fval
        prev_Lamda=Lamda;
        prev_fval=fval;
        Lamda=Lamda*1.1;
    else
        %Solve the economic distribution problem for higher
iterations.

temp_sol=vpa(equations_solver_example_4(length(new_current_combination),new_f_value,new_current_combination,Lamda));
        temp_fval=vpa(P_R - sum(temp_sol(1:length(temp_sol))));

        if abs(prev_fval-temp_fval)<Error
            %Checking the load constrain.

[solution>Total_Power_Production]=constrains_example_4(temp_sol,length(temp_sol),j_value,P_R,P_min,P_max);
            while Total_Power_Production<P_R
                %We need to generate more power.
                new_temp_fval=P_R-Total_Power_Production;
                new_temp_Lamda=Lamda;
                Lamda=Lamda*1.1;
            end
        end
    end
end

```



```

        %Solving the problem with the new Lamda.

temp_sol=vpa(equations_solver_example_4(length(new_current_combination),new_
f_value,new_current_combination,Lamda));

[solution,Total_Power_Production]=constrains_example_4(temp_sol,length(temp_
sol),j_value,P_R,P_min,P_max);
    temp_fval=P_R-Total_Power_Production;
    end
    if Total_Power_Production>P_R

        %We need to produce less power.
        Lamda=Lamda -((Lamda-new_temp_Lamda)/(temp_fval-
new_temp_fval))*temp_fval;

        %Solving the problem for the new Lamda.

temp_sol=vpa(equations_solver_example_4(length(new_current_combination),new_
f_value,new_current_combination,Lamda));

[solution,Total_Power_Production]=constrains_example_4(temp_sol,length(temp_
sol),j_value,P_R,P_min,P_max);
    new_temp_fval=[];
    end
    %Printing the solution.
    cost_of_sol=[];
    fprintf('\n\nThe optimal solution, after the power
constrains is:')
    fprintf('\n\nLamda: <strong>%.2f</strong>',Lamda)
    for w=1:length(solution)
        fprintf('\nP%d is equal to: <strong>%.2f
MW</strong>',j_value(w),vpa(solution(w)))
        cost_of_sol=[cost_of_sol
subs(F(j_value(w)),P(j_value(w)),solution(w))];
    end
    cost_of_sol=sum(cost_of_sol);
    all_solution_costs=[all_solution_costs cost_of_sol];
    fprintf('\n\nThe total cost of the solution is:
<strong>%.3f</strong>\n',cost_of_sol)
    fprintf('\n\n')
    break;
    else
        temp_Lamda=Lamda;
        Lamda= Lamda-((Lamda-prev_Lamda)/(temp_fval-
prev_fval))*temp_fval;
        prev_Lamda=temp_Lamda;
        prev_fval=temp_fval;
    end

end

end
end

```

Στην πέμπτη και τελευταία περίπτωση, χρησιμοποιείται ξανά ο αλγόριθμος επαναλήψεων λάμδα, αλλά αυτή την φορά, για να κατανέμει το φορτίο σε τρεις μονάδες.

```

%Checking if there are 3 Units.
elseif length(new_current_combination)==3
    fprintf('\nFor the <strong>%d</strong> combination: ',i)
    fprintf('<strong>[ ')
    fprintf(' %s',current_combination)
    fprintf(' ]</strong>')
    fprintf('\n')

    Lamda=1;
    [solution]=equations_solver_example_4(Number_of_units,F,P,Lamda);

    [solution,Total_Power_Production]=constrains_example_4(solution,length(solut
ion),j_value,P_R,P_min,P_max);
    fval=vpa(P_R - Total_Power_Production);
    temp_fval=inf;

    while abs(temp_fval)>0
        if Total_Power_Production<P_R
            prev_lamda=Lamda;
            Lamda=Lamda*1.1;

[solution]=equations_solver_example_4(Number_of_units,F,P,Lamda);

[solution,Total_Power_Production]=constrains_example_4(solution,length(solut
ion),j_value,P_R,P_min,P_max);
            temp_fval=vpa(P_R - Total_Power_Production);
        end
        if Total_Power_Production>P_R
            Lamda= Lamda - ((Lamda-prev_lamda)/(temp_fval-
fval))*temp_fval;

[solution]=equations_solver_example_4(Number_of_units,F,P,Lamda);

[solution,Total_Power_Production]=constrains_example_4(solution,length(solut
ion),j_value,P_R,P_min,P_max);
            temp_fval=vpa(P_R - Total_Power_Production);
        end
        if abs(temp_fval)<0.0000001
            temp_fval=0;

            %Printing the solution.
            cost_of_sol=[];
            fprintf('\nThe optimal solution, after the power constrains
is:')

            fprintf('\nLamda: <strong>%.2f</strong>',Lamda)
            for w=1:length(solution)
                fprintf('\nP%d is equal to: <strong>%.2f
MW</strong>',j_value(w),vpa(solution(w)))
                cost_of_sol=[cost_of_sol
subs(F(j_value(w)),P(j_value(w)),solution(w))];
            end
            cost_of_sol=sum(cost_of_sol);
            all_solution_costs=[all_solution_costs cost_of_sol];
            fprintf('\nThe total cost of the solution is:
<strong>%.3f</strong>\n',cost_of_sol)
            fprintf('\n')
            break;
        end
        temp_Lamda=Lamda;
        Lamda= Lamda- ((Lamda-prev_Lamda)/(temp_fval-
prev_fval))*temp_fval;
        prev_Lamda=temp_Lamda;
        prev_fval=temp_fval;
    end
end

```

Τέλος, βρίσκεται η οικονομικότερη λύση από όλους τους συνδυασμούς και εκτυπώνεται.

```
%finding the best sol
all_solution_costs=vpa(all_solution_costs);
temp_best_sol=inf;
for w=1:length(all_solution_costs)
    if all_solution_costs(w)<temp_best_sol
        temp_best_sol=all_solution_costs(w);
        w_value=w;
    end
end
fprintf('\n\nThe best solution is for the combination ')
fprintf('<strong>[ ')
fprintf(' %s',all_machines_combination(w_value,1:3))
fprintf(' ]</strong>,\n')
fprintf(' with the total cost of
<strong>%.3f</strong>.\n',temp_best_sol)
end
end
```

Στην τέταρτη και πέμπτη περίπτωση, χρησιμοποιούνται οι τροποποιημένες πλέον συναρτήσεις `equations_solver_example_4` και `constrains_example_4`.

Αναφορικά με την `equations_solver_example_4`, προστέθηκε μια εντολή η οποία ορίζει ως μηδέν την λύση, για την περίπτωση όπου το διάνυσμα με τον τρέχον συνδυασμό είναι κενό.

```
function [solution]=equations_solver_example_4(N,F,P,lamda)
%Cooperation equations
if N==0
    solution=0;
else
    for i=1:N
        dF(i)=diff(F(i),P(i))==lamda; %Creating cooperation equations.
    end
    % Solving the economic distribution problem
    [A,B] = equationsToMatrix([dF], [P]);
    solution=linsolve(A,B);
end
```

Από την άλλη, στην συνάρτηση `constrains_example_4`, υπάρχουν αρκετές αλλαγές. Στο πρώτο τμήμα της, ελέγχονται όρια λειτουργίας και υπολογίζεται η συνολική παραγόμενη ενέργεια, μετά τους περιορισμούς. Επίσης, δημιουργούνται δύο κενά, βοηθητικά διανύσματα το διάνυσμα `changed_value` και `changed_element`.

Στο πρώτο διάγραμμα, αποθηκεύεται η νέα πλέον τιμή της ισχύος μετά τον περιορισμό, ενώ στο δεύτερο, η θέση του στοιχείου που οποίο άλλαξε τιμή.

```
function
[solution,Total_Power_Production]=constrains_example_4(solution,N,value_in_v
ector,P_R,P_min,P_max)

sol_before_constrains=solution;
%Saving the solution before the power constrains.
Total_Power_Before_constrains=vpa(sum(sol_before_constrains));
%Saving produced power before the power constrains.
changed_value=[];
%Creating an array for the value which is going to change.
changed_element=[];
%Creating an array for the element which is going to change.
for i=1:N
    if solution(i)>P_max(value_in_vector(i))
%If the power is more.
        solution(i)=P_max(value_in_vector(i));
%give the max possible value.
        changed_value=[changed_value [solution(i)]];
%we save the value that changed in an array.
        changed_element=[changed_element [i]];
%we save the element that changed in an array.
    elseif solution(i)<P_min(value_in_vector(i))
%If the power is less.
        solution(i)=P_min(value_in_vector(i));
%give the min possible value.
        changed_value=[changed_value [solution(i)]];
%we save the value that changed in an array.
        changed_element=[changed_element [i]];
%we save the element that changed in an array.
    end
end
Total_Power_Production=double(sum(solution));
%Calculate the new produced power after the power constrains.
```

Στο δεύτερο τμήμα της, εισάγεται ένας περιορισμός, για την συγκεκριμένη περίπτωση την οποία, υπάρχουν δύο μονάδες στον συνδυασμό και η μια από τις δύο δέχεται τροποποίηση στην ισχύ εξόδου της, λόγω των ορίων λειτουργίας. Αν η παραγόμενη ενέργεια πριν την τροποποίηση, είναι μικρότερη από την παραγόμενη, μετά την τροποποίηση, τότε αναγκάζουμε την μη τροποποιημένη από περιορισμούς μονάδα, να παράξει την ενέργεια που απαιτείται, για να καλυφθεί η ζήτηση.

```

%Constrain for the specific occasion in which we forge a unit to produce
less or more power, to cover the
%difference between the requested power and the produced power.
if N==2 && Total_Power_Before_constrains<Total_Power_Production &&
length(changed_value)==1
    new_P_R=P_R-sum(changed_value);
    value_in_vector(changed_element)=[];
    if (P_min(value_in_vector)<=new_P_R) &&
(new_P_R<=P_max(value_in_vector))
        solution(value_in_vector)=new_P_R;
%We keep the changed value from the constrains the same, but we forge the
other
        Total_Power_Production=double(sum(solution));
%to get the value: Requested_power-Changed_value_after_constrains.
end

```

Στο τρίτο τμήμα της, προστίθεται μια τελευταία συνθήκη για την περίπτωση όπου υπάρχουν δύο μονάδες και η παραγόμενη ενέργεια μετά τους περιορισμούς των ορίων λειτουργίας των μονάδων, είναι περισσότερη από την ζητούμενη. Όμοια στην περίπτωση αυτή, αναγκάζουμε την μονάδα, που δεν έχει τροποποιηθεί από τον περιορισμό του ορίου λειτουργίας, να παράξει την υπόλοιπη απαιτούμενη ενέργεια, προκειμένου να καλυφθεί η ζήτηση.

```

%Constrain for the specific occasion in which we forge a unit to produce
some specific power, in order to cover the requested power.
%We achieve this with the same technic as before.
elseif N==2 && Total_Power_Production>P_R
    for k=1:length(solution)
        if solution(k) == P_min(value_in_vector(k)) || solution(k) ==
P_max(value_in_vector(k))
            sol_for_replace=solution(k);
        else
            solution(k)=P_R-sol_for_replace;
        end
    end
    Total_Power_Production=double(sum(solution));
end

```

Έτσι, για την ζητούμενη ενέργεια των 550 MW, το αποτέλεσμα το οποίο βλέπει ο χρήστης είναι το ακόλουθο.

For the 1 combination: [ 0 0 0 ]

The solution **is not possible**.

For the 2 combination: [ P1 0 0 ]

The solution is possible and its cost is: **4658.148**

For the 3 combination: [ 0 P2 0 ]

The solution **is not possible**.

For the 4 combination: [ 0 0 P3 ]

The solution **is not possible**.

For the 5 combination: [ P1 P2 0 ]

The optimal solution, after the power constrains is:

Lamda: **7.64**

P1 is equal to: **294.08 MW**

P2 is equal to: **255.92 MW**

The total cost of the solution is: **4729.124**

For the 6 combination: [ P1 0 P3 ]

The optimal solution, after the power constrains is:

Lamda: **8.31**

P1 is equal to: **500.00 MW**

P3 is equal to: **50.00 MW**

The total cost of the solution is: **4751.208**

For the 7 combination: [ 0 P2 P3 ]

The optimal solution, after the power constrains is:

Lamda: 9.75

P2 is equal to: 400.00 MW

P3 is equal to: 150.00 MW

The total cost of the solution is: 4677.156

For the 8 combination: [ P1 P2 P3 ]

The optimal solution, after the power constrains is:

Lamda: 7.56

P1 is equal to: 266.45 MW

P2 is equal to: 233.55 MW

P3 is equal to: 50.00 MW

The total cost of the solution is: 4855.136

The best solution is for the combination [ P1 0 0 ], with the total cost of 4658.148.

Σύμφωνα με το Παράδειγμα 4 στο κεφάλαιο 3 οι λύσεις ήταν:

Πίνακας 4.3: Αποτελέσματα παραδείγματος 4.

Κατάσταση μονάδων			Όρια Ισχύος Συνδυασμού		Έξοδος Μονάδων			Κόστος Μονάδων (κ€)			Συνολικό Κόστος
$UN_1$	$UN_2$	$UN_3$	$P_{min}$	$P_{max}$	$P_1$	$P_2$	$P_3$	$F_1$	$F_2$	$F_3$	(κ€)
OFF	OFF	OFF	0	0	Μη Εφικτή Λύση						
OFF	OFF	ON	50	200	Μη Εφικτή Λύση						
OFF	ON	OFF	100	400	Μη Εφικτή Λύση						
ON	OFF	OFF	150	600	550	0	0	4658	0	0	4658
OFF	ON	ON	150	600	0	400	150	0	3249	1428	4677
ON	OFF	ON	200	800	500	0	50	4245	0	506	4751
ON	ON	OFF	250	1000	294	256	0	2615	2114	0	4729
ON	ON	ON	300	1200	266	234	0	2405	1944	506	4855

Από την άλλη, οι λύσεις του κώδικα συγκεντρωτικά είναι:

Πίνακας 4.4: Αποτελέσματα παραδείγματος 4, από το Matlab.

Κατάσταση μονάδων			Όρια Ισχύος Συνδυασμού		Έξοδος Μονάδων			Κόστος Μονάδων (k€)			Συνολικό Κόστος
$UN_1$	$UN_2$	$UN_3$	$P_{min}$	$P_{max}$	$P_1$	$P_2$	$P_3$	$F_1$	$F_2$	$F_3$	(k€)
OFF	OFF	OFF	0	0	Μη Εφικτή Λύση						
OFF	OFF	ON	50	200	Μη Εφικτή Λύση						
OFF	ON	OFF	100	400	Μη Εφικτή Λύση						
ON	OFF	OFF	150	600	550	0	0	4658	0	0	4658
OFF	ON	ON	150	600	0	400	150	0	3249	1428	4677
ON	OFF	ON	200	800	500.00	0	50.00	4245	0	506	4751
ON	ON	OFF	250	1000	294.08	255.92	0	2615	2114	0	4729
ON	ON	ON	300	1200	266.45	233.55	0	2405	1944	506	4855

Παρατηρούμε, ότι ο κώδικας, έχει τα ίδια αποτελέσματα, με εκείνα του πίνακα στο παράδειγμα 4. Αν ο χρήστης, επιθυμεί να δει τον πιο οικονομικό τρόπο για την κάλυψη ενός άλλου φορτίου, κατευθύνεται στο αρχείο data.xlsx και αλλάζει την τιμή της ζητούμενης ενέργειας. Αν για παράδειγμα, αλλάξει την ζητούμενη ενέργεια από 550 MW σε 450 MW, βρίσκει την ακόλουθη καλύτερη λύση.

The best solution is for the combination [ 0 P2 P3 ], with the total cost of 3755.268.

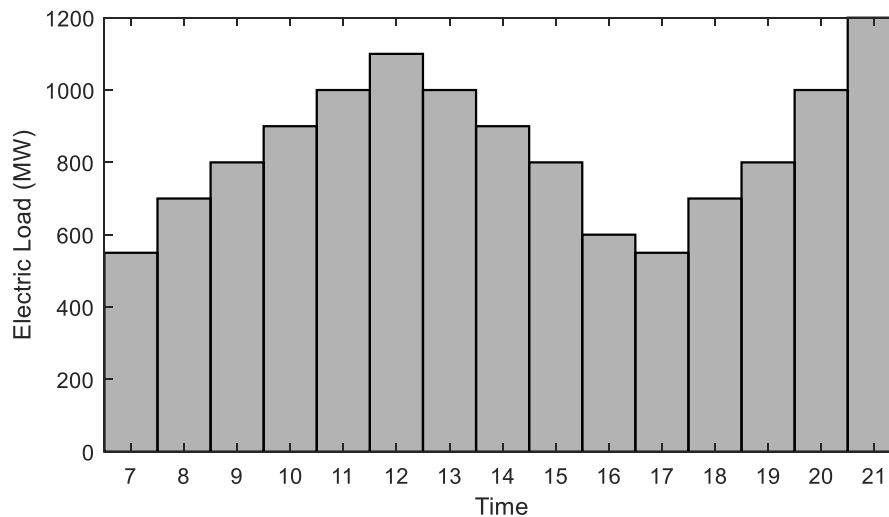
## 4.6. Ανάπτυξη κώδικα για την επίλυση του παραδείγματος 5

Το παράδειγμα 5, αναφέρεται και αυτό στην ένταξη μονάδων παραγωγής. Στο συγκεκριμένο παράδειγμα, δεν έχουμε μια σταθερή τιμή για την ζητούμενη ενέργεια, αλλά μια καμπύλη φορτίου την οποία πρέπει να καλύψουν οι μονάδες παραγωγής.



## Εκφώνηση

Έστω, ξανά η εταιρεία του παραδείγματος 4. Αυτή την φορά, δεν θα υπάρχει ένα στιγμιαίο φορτίο, αλλά μια καμπύλη φορτίου διάρκειας μιας ημέρας. Το ελάχιστο φορτίο της καμπύλης αυτής είναι  $550\text{ MW}$ , ενώ το μέγιστο της  $1200\text{ MW}$ .



Γράφημα 4.1: Καμπύλη φορτίου.

Προκειμένου να καλυφθεί η καμπύλη φορτίου, κάποιες μονάδες κρατούνται εκτός λειτουργίας, όταν το φορτίο ελαττώνεται, ενώ εντάσσονται ξανά, όταν αυτό αυξάνεται. Αξίζει να σημειωθεί, ότι ένα πρόβλημα ένταξης έχει πολλούς περιορισμούς, οι οποίοι καθιστούν την λύση του περίπλοκη. Στην περίπτωση μας, ο μοναδικός περιορισμός, είναι να καλύπτεται η ζήτηση.

Ο κανόνας ένταξης που θα χρησιμοποιηθεί είναι:

- Αν το φορτίο είναι πάνω από  $1000\text{ MW}$ , τότε λειτουργούν όλες οι μονάδες μαζί.
- Αν το φορτίο είναι μεταξύ  $600\text{ MW}$  και  $1000\text{ MW}$ , τότε λειτουργούν οι μονάδες 1 και 2.
- Αν το φορτίο είναι μικρότερο από  $600\text{ MW}$ , τότε λειτουργεί μόνο, η μονάδα 1.

Η επίλυση του παραδείγματος θα γίνει, με βάση τον παραπάνω κανόνα.

Ο χρήστης κατευθύνεται στον υπό – φάκελο με όνομα, Example\_5, όπου και βρίσκει τα πέντε αρχεία του παραδείγματος. Το αρχείο των δεδομένων data.xlsx, την συνάρτηση array\_creator, τον κύριο κώδικα του παραδείγματος και τις συναρτήσεις equations\_solver\_example\_5 και constrains\_example\_5, οι οποίες θα σχολιαστούν στην συνέχεια.

Στο παράδειγμα αυτό, υπάρχει και μια αλλαγή στον τρόπο εκχώρησης των δεδομένων. Συγκεκριμένα, η ζήτηση της ενέργειας, επειδή είναι καμπύλη φορτίου, καταχωρήθηκε στο αρχείο, με αύξουσα σειρά ανά 50 MW, από την ελάχιστη, έως την μέγιστη τιμή, από τα κελιά B5, έως O5. Το φύλλο example\_5, με τα δεδομένα παρουσιάζεται παρακάτω.

Πίνακας 4.5: Δεδομένα του παραδείγματος 5, στο αρχείο data.xlsx.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		Unit 1	Unit 2	Unit 3											
2	Min	150	100	50											
3	Max	600	400	200											
4	Cost	3.96	3.96	3.96											
5	PR	550	600	650	700	750	800	850	900	950	1000	1050	1100	1150	1200

Ο συγκεκριμένος κώδικας, χωρίζεται σε τέσσερα τμήματα. Τα τρία πρώτα, με ονόματα, Intro, Importing data from excel και declare the other parameters, είναι τα ίδια με του κώδικα του παραδείγματος 4.

Το τέταρτο τμήμα, στο οποίο γίνεται και η επίλυση του προβλήματος ένταξης μονάδων, αποτελείται από τρία κομμάτια. Το πρώτο, λύνει το πρόβλημα για τα φορτία ζήτησης μικρότερα ή ίσα των 600 MW, στα οποία λειτουργεί μόνο μονάδα 1, το δεύτερο, για φορτία μεγαλύτερα των 600 MW, έως και φορτία μικρότερα ή ίσα των 1000 MW, στα οποία λειτουργούν ταυτόχρονα, οι μονάδες 1 και 2 και τέλος το τρίτο, για φορτία μεταξύ 1000 MW και 1200 MW, όπου λειτουργούν και οι τρεις μονάδες ταυτόχρονα.

Ο κώδικας, χρησιμοποιεί τις μεθόδους που παρουσιάστηκαν στα προηγούμενα παραδείγματα και τερματίζει όταν καλυφθούν όλες οι τιμές της καμπύλης φορτίου.

Στο πρώτο κομμάτι του κώδικα, όπως προαναφέρθηκε λειτουργεί μόνο μια μονάδα. Για την επίλυση του προβλήματος κατανομής, χρησιμοποιείται η subs, η οποία λύνει την εξίσωση για την εκάστοτε τιμή του ζητούμενου φορτίου.

## Solving the problem.

```
%For the solutions we use the same technics as the example_4B, example_4G
and example_5A.
for i=1:length(P_R)
    if P_R(i)<=600
        %if the requested power is less than 600MW, we use only the first
        machine.
        cost_of_sol=vpa(subs(F(1),P(1),P_R(i)));

        fprintf('For the requested power of <strong>%d MW</strong>, the
        solution is:',P_R(i))
        fprintf('\nP1 is equal to: <strong>%d MW</strong>, with the cost of:
        <strong>%.2f</strong>\n\n',P_R(i),cost_of_sol)
```

Στο δεύτερο κομμάτι, λειτουργούν οι μονάδες 1 και 2. Η κατανομή του φορτίου στις μονάδες αυτές, γίνεται με χρήση του αλγορίθμου επαναλήψεων λάμδα.

```
elseif P_R(i)>600 && P_R(i)<=1000
    %if the requested power is less than 1000MW and more than 600MW, we
    use the first and the second machine.
    Lamda=1;
    Error=0.1;
    for j=1:10
        if j==1
            sol=vpa(equations_solver_example_5(2,F,P,Lamda));
            fval=P_R(i)-sum(sol(1:length(sol)));

            %Define lamda and fval
            prev_Lamda=Lamda;
            prev_fval=fval;
            Lamda=Lamda*1.1;
        else
            temp_sol=vpa(equations_solver_example_5(2,F,P,Lamda));
            temp_fval=P_R(i)-sum(temp_sol(1:length(temp_sol)));
            if abs(prev_fval-temp_fval)<Error
                %Checking the load constrain.

[solution,Total_Power_Production]=constrains_example_5(temp_sol,length(temp_
sol),P_R(i),P_min,P_max);
                while Total_Power_Production<P_R(i)
                    %We need to generate more power.
                    new_temp_fval=P_R(i)-Total_Power_Production;
                    new_temp_Lamda=Lamda;
                    Lamda=Lamda*1.1;

                    %Solving the problem with the new Lamda.
```

```

temp_sol=vpa(equations_solver_example_5(2,F,P,Lamda));

[solution,Total_Power_Production]=constrains_example_5(temp_sol,length(temp_
sol),P_R(i),P_min,P_max);
    temp_fval=P_R(i)-Total_Power_Production;
    end
    if Total_Power_Production>P_R(i)
        %We need to produce less power.
        Lamda=Lamda - ((Lamda-new_temp_Lamda)/(temp_fval-
new_temp_fval))*temp_fval;

        %Solving the problem for the new Lamda.

temp_sol=vpa(equations_solver_example_5(2,F,P,Lamda));

[solution,Total_Power_Production]=constrains_example_5(temp_sol,length(temp_
sol),P_R(i),P_min,P_max);
    end
    cost_of_sol=[];
    fprintf('\n\nFor the requested power of <strong>%d
MW</strong>, the solution is:\n',P_R(i))
    for w=1:length(solution)
        cost_of_sol=[cost_of_sol
double(subs(F(w),P(w),solution(w)))];
        fprintf('\nP%d is equal to: <strong>%.2f
MW</strong>, with the cost of: <strong>%.2f
MW</strong>',w,double(solution(w)),cost_of_sol(w))
    end
    cost_of_sol=sum(cost_of_sol);
    fprintf('\nThe total cost of this solution is:
<strong>%.2f</strong>\n',cost_of_sol)
    break;
    else

        temp_Lamda=Lamda;
        Lamda= Lamda- ((Lamda-prev_Lamda)/(temp_fval-
prev_fval))*temp_fval;
        prev_Lamda=temp_Lamda;
        prev_fval=temp_fval;

    end

end

end

end

```

Στον παραπάνω κώδικα, γίνεται ταυτόχρονος υπολογισμός του κόστους για το εκάστοτε φορτίο. Για να επιτευχθεί αυτό, χρησιμοποιείται ένα βοηθητικό διάνυσμα με όνομα, `cost_of_sol`, το οποίο είναι, αρχικά κενό. Στην συνέχεια, υπολογίζεται το κόστος παραγωγής της κάθε μονάδας ξεχωριστά με την εντολή `subs` και το προστίθεται στο βοηθητικό διάνυσμά. Έπειτα, υπολογίζεται το άθροισμα των στοιχείων του διανύσματος και το αποτέλεσμα, δεν είναι άλλο, από το κόστος της συγκεκριμένης λύσης.

```

cost_of_sol=[];

fprintf('\n\nFor the requested power of <strong>%d
MW</strong>, the solution is:\n',P_R(i))
for w=1:length(solution)
    cost_of_sol=[cost_of_sol
double(subs(F(w),P(w),solution(w)))];
    fprintf('\nP%d is equal to: <strong>%.2f
MW</strong>, with the cost of: <strong>%.2f
MW</strong>',w,double(solution(w)),cost_of_sol(w))
end
cost_of_sol=sum(cost_of_sol);
fprintf('\nThe total cost of this solution is:
<strong>%.2f</strong>\n',cost_of_sol)

```

Επίσης, γίνεται χρήση και των συναρτήσεων, equations\_solver\_example\_5 και constrains\_example\_5. Η συνάρτηση equations\_solver\_example\_5, είναι ίδια με τα προηγούμενα παραδείγματα, με την διαφορά, ότι χρησιμοποιεί ένα βοηθητικό διάνυσμα, με όνομα temp\_P, το οποίο βρίσκει τις μονάδες που λειτουργούν και έτσι δημιουργεί τις συναρτήσεις συνεργασίας, μόνο για αυτές.

```

function [solution]=equations_solver_example_5(N,F,P,lamda)
temp_P=[];
%Cooperation equations
for i=1:N
    dF(i)=diff(F(i),P(i))==lamda;    %Creating cooperation equations.
    temp_P=[temp_P P(i)];            %Creating a temporary array only for the
    units we have.
end

% Solving the economic distribution problem
[A,B] = equationsToMatrix([dF], [temp_P]);
solution=linsolve(A,B);
end

```

Η συνάρτηση constrains\_example\_5, χωρίζεται σε τέσσερα μέρη. Το πρώτο μέρος, είναι ίδιο με του παραδείγματος 4 στο οποίο παράλληλα με τους περιορισμούς ισχύος, αποθηκεύει την μονάδα, της οποίας την έξοδο, τροποποίησε ο περιορισμός. Το δεύτερο μέρος, αναφέρεται στην συγκεκριμένη περίπτωση στην οποία, μόνο μιας μονάδας, η έξοδος ισχύος, έχει τροποποιηθεί και η συνολική παραγωγή μετά τους περιορισμούς είναι περισσότερη από την απαιτούμενη. Έτσι, αναγκάζουμε την άλλη μονάδα, να παράξει λιγότερη ενέργεια προκειμένου να καλυφθεί ακριβώς το ζητούμενο φορτίο. Κάτι τέτοιο μπορεί να συμβεί, αν μια μονάδα λειτουργεί κάτω από το ελάχιστο όριο λειτουργίας της και με τους περιορισμούς αυξάνεται η παραγόμενη

ενέργεια, αφού αναγκάζεται να λειτουργήσει στο ελάχιστο δυνατό, το οποίο είναι σαφώς περισσότερο από το φορτίο που λειτουργούσε. Έτσι, η παραγόμενη ενέργεια μετά τους περιορισμούς βρίσκεται περισσότερη από την ζητούμενη.

```
if Total_Power_Production>P_R && length(changed_value)==1
%Constrain for the specific occasion in which we forge a unit to produce
less or more power,
    new_P_R=P_R-changed_value;
%to cover the difference between the requested power and the produced power.
    temp_array=solution;
    temp_array(changed_element)=[];
    available_element_for_change(changed_element)=[];
    if (P_min(available_element_for_change)<=new_P_R) &&
    (new_P_R<=P_max(available_element_for_change))
        solution(available_element_for_change)=new_P_R;
%We keep the changed value from the constrains the same, but we forge the
other
        Total_Power_Production=double(sum(solution));
%to get the value: Requested_power-Changed_value_after_constrains.
end
```

Το τρίτο μέρος, αναφέρεται στην περίπτωση, όπου λειτουργούν δύο μονάδες και ο περιορισμός ορίων λειτουργίας τους, τις αναγκάζει να δουλέψουν είτε στο μέγιστο της ισχύς τους, είτε στο ελάχιστο. Συγκεκριμένα, συναντάται η περίπτωση όπου οι μονάδες, λειτουργούν πάνω από το μέγιστο όριο λειτουργίας τους και έτσι ο περιορισμός τους δίνει την μέγιστη επιτρεπτή τιμή. Παρόλα αυτά, η παραγόμενη ενέργεια είναι περισσότερη από την απαιτούμενη. Έτσι, κρατάμε την μονάδα με την μικρότερη τιμή, από τις μέγιστες, και αναγκάζουμε την άλλη να λειτουργήσει σε μικρότερη ισχύ για να καλυφθεί η ζήτηση. Αν οι μονάδες δεν είχαν το ίδιο κόστος καυσίμου, ο περιορισμός αυτός δεν θα λειτουργούσε σωστά, καθώς θα έδινε ένα αποτέλεσμα το οποίο απλά ικανοποιεί την ζήτηση, χωρίς να λαμβάνει υπόψη του ποια μονάδα παράγει φθηνότερα ενέργεια.

```
elseif Total_Power_Production>P_R && length(changed_value)==2 && N==2
    if solution(1)>solution(2)
%Constrain for the specific occasion in which we forge a unit to produce
less or more power,
        new_P_R=P_R-solution(2);
%to cover the difference between the requested power and the produced power.
        solution(1)=new_P_R;
%This works only for 2 units.
    elseif solution(1)<solution(2)
%We keep the smaller changed value after the constrains the same, but we
forge the higher
```

```

        new_P_R=P_R-solution(1);
    %to get the value: Requested_power-Changed_value_after_constrains.
    solution(2)=new_P_R;
    Total_Power_Production=double(sum(solution));
end

```

Τέλος, το τέταρτο μέρος, αναφέρεται στην περίπτωση την οποία λειτουργούν και οι τρεις μονάδες, και οι μονάδες 1 και 2 λειτουργούν στο μέγιστο της ισχύος τους και με την παραγόμενη ενέργεια από την μονάδα 3, παραγόμενη ενέργεια είναι περισσότερη από την απαιτούμενη. Έτσι, αναγκάζουμε την μονάδα 3 να παράξει το φορτίο που απαιτείται για να καλυφθεί ακριβώς η ζήτηση.

```

elseif Total_Power_Production>P_R && solution(1)==P_max(1) &&
solution(2)==P_max(2)
    new_P_R=P_R-sum(solution(1:2));
    %Constrain for the specific occasion in which we forge a unit to produce
    less or more power,
    solution(3)=new_P_R;
    %to cover the difference between the requested power and the produced power.
    Total_Power_Production=double(sum(solution));
    %It works, only for the occasion in which we have the 2 first units on their
    max load.
end
%We forge the 3rd unit to produce the remaining power.
end

```

Αφού επεξηγήθηκαν οι συναρτήσεις, στο τρίτο και τελευταίο κομμάτι του κώδικα, επιλύεται το πρόβλημα κατανομής αυτή την φορά και για τις τρεις μονάδες, όταν το φορτίο είναι μεταξύ των 1000 MW και 1200 MW. Ταυτόχρονα και εδώ υπολογίζεται το κόστος της κάθε λύσης.

```

elseif P_R(i)>1000 && P_R(i)<=1200
    %if the requested power is less than 1200MW and more than 1000MW, we
    use all the machines.
    Lamda=1;
    [solution]=equations_solver_example_5(Number_of_units,F,P,Lamda);

    [solution,Total_Power_Production]=constrains_example_5(solution,length(solution),P_R(i),P_min,P_max);
    fval=double(P_R(i) - Total_Power_Production);
    temp_fval=inf;
    while abs(temp_fval)>0
        if Total_Power_Production<P_R(i)
            prev_lamda=Lamda;
            Lamda=Lamda*1.5;

```

```

[solution]=equations_solver_example_5(Number_of_units,F,P,Lamda);

[solution,Total_Power_Production]=constrains_example_5(solution,length(solution),P_R(i),P_min,P_max);
    temp_fval=double((P_R(i) - Total_Power_Production));
    end

    if Total_Power_Production>P_R(i)
        Lamda= Lamda - ((Lamda-prev_lamda)/(temp_fval-fval))*temp_fval;

[solution]=equations_solver_example_5(Number_of_units,F,P,Lamda);

[solution,Total_Power_Production]=constrains_example_5(solution,length(solution),P_R(i),P_min,P_max);
    temp_fval=double((P_R(i) - Total_Power_Production));
    end

    if abs(temp_fval)<0.0000001
        temp_fval=0;
        cost_of_sol=[];
        fprintf('\n\nFor the requested power of <strong>%d
MW</strong>, the solution is:\n',P_R(i))
        for w=1:length(solution)
            cost_of_sol=[cost_of_sol
double(subs(F(w),P(w),solution(w)))];
            fprintf('\nP%d is equal to: <strong>%.2f MW</strong>,
with the cost of: <strong>%.2f
MW</strong>',w,double(solution(w)),cost_of_sol(w))
        end
        cost_of_sol=sum(cost_of_sol);
        fprintf('\nThe total cost of this solution is:
<strong>%.2f</strong>\n',cost_of_sol)
        break;
    end
end
end
end
end

```

Το αποτέλεσμα που δίνει ο κώδικας είναι το παρακάτω.

For the requested power of **550 MW**, the solution is:

P1 is equal to: **550 MW**, with the cost of: **4658.15**

For the requested power of **600 MW**, the solution is:

P1 is equal to: **600 MW**, with the cost of: **5077.91**

For the requested power of **650 MW**, the solution is:

P1 is equal to: **349.34 MW**, with the cost of: **3041.30 MW**



P2 is equal to: **300.66 MW**, with the cost of: **2459.13 MW**

The total cost of this solution is: **5500.44**

For the requested power of **700 MW**, the solution is:

P1 is equal to: **376.97 MW**, with the cost of: **3257.51 MW**

P2 is equal to: **323.03 MW**, with the cost of: **2634.16 MW**

The total cost of this solution is: **5891.68**

For the requested power of **750 MW**, the solution is:

P1 is equal to: **404.61 MW**, with the cost of: **3475.78 MW**

P2 is equal to: **345.39 MW**, with the cost of: **2810.85 MW**

The total cost of this solution is: **6286.63**

For the requested power of **800 MW**, the solution is:

P1 is equal to: **432.24 MW**, with the cost of: **3696.10 MW**

P2 is equal to: **367.76 MW**, with the cost of: **2989.21 MW**

The total cost of this solution is: **6685.31**

For the requested power of **850 MW**, the solution is:

P1 is equal to: **459.87 MW**, with the cost of: **3918.48 MW**

P2 is equal to: **390.13 MW**, with the cost of: **3169.23 MW**

The total cost of this solution is: **7087.71**

For the requested power of **900 MW**, the solution is:

P1 is equal to: **500.00 MW**, with the cost of: **4245.12 MW**

P2 is equal to: **400.00 MW**, with the cost of: **3249.18 MW**

The total cost of this solution is: **7494.30**

For the requested power of **950 MW**, the solution is:

P1 is equal to: **550.00 MW**, with the cost of: **4658.15 MW**

P2 is equal to: **400.00 MW**, with the cost of: **3249.18 MW**

The total cost of this solution is: **7907.33**

For the requested power of **1000 MW**, the solution is:

P1 is equal to: **600.00 MW**, with the cost of: **5077.91 MW**

P2 is equal to: **400.00 MW**, with the cost of: **3249.18 MW**

The total cost of this solution is: **8327.09**

For the requested power of **1050 MW**, the solution is:

P1 is equal to: **600.00 MW**, with the cost of: **5077.91 MW**

P2 is equal to: **400.00 MW**, with the cost of: **3249.18 MW**

P3 is equal to: **50.00 MW**, with the cost of: **506.09 MW**

The total cost of this solution is: **8833.18**

For the requested power of **1100 MW**, the solution is:

P1 is equal to: **600.00 MW**, with the cost of: **5077.91 MW**

P2 is equal to: **400.00 MW**, with the cost of: **3249.18 MW**

P3 is equal to: **100.00 MW**, with the cost of: **955.15 MW**

The total cost of this solution is: **9282.24**

For the requested power of **1150 MW**, the solution is:

P1 is equal to: **600.00 MW**, with the cost of: **5077.91 MW**

P2 is equal to: **400.00 MW**, with the cost of: **3249.18 MW**

P3 is equal to: **150.00 MW**, with the cost of: **1427.98 MW**

The total cost of this solution is: **9755.06**

For the requested power of **1200 MW**, the solution is:

P1 is equal to: **600.00 MW**, with the cost of: **5077.91 MW**

P2 is equal to: **400.00 MW**, with the cost of: **3249.18 MW**

P3 is equal to: **200.00 MW**, with the cost of: **1924.56 MW**

The total cost of this solution is: **10251.65**

Σύμφωνα με το Παράδειγμα 5 στο κεφάλαιο 3 οι λύσεις ήταν:

Πίνακας 4.6: Αποτελέσματα παραδείγματος 5.

Φορτίο	Κατάσταση μονάδων			Όρια Ισχύος Συνδυασμού		Έξοδος Μονάδων			Κόστος Μονάδων (κ€)			Συνολικό Κόστος
(MW)	$UN_1$	$UN_2$	$UN_3$	$P_{min}$	$P_{max}$	$P_1$	$P_2$	$P_3$	$F_1$	$F_2$	$F_3$	(κ€)
550	1	0	0	150	600	550	0	0	4658	0	0	4658
600	1	0	0	150	600	600	0	0	5078	0	0	5078
650	1	1	0	250	1000	349	301	0	3041	2459	0	5500
700	1	1	0	250	1000	377	323	0	3258	2634	0	5892
750	1	1	0	250	1000	405	345	0	3476	2811	0	6287
800	1	1	0	250	1000	432	368	0	3698	2991	0	6689
850	1	1	0	250	1000	460	390	0	3918	3169	0	7087
900	1	1	0	250	1000	500	400	0	4245	3249	0	7494
950	1	1	0	250	1000	550	400	0	4658	3249	0	7907
1000	1	1	1	250	1000	600	400	0	5078	3249	0	8327
1050	1	1	1	300	1200	600	400	50	5078	3249	506	8833
1100	1	1	1	300	1200	600	400	100	5078	3249	955	9282
1150	1	1	1	300	1200	600	400	150	5078	3249	1428	9755
1200	1	1	1	300	1200	600	400	200	5078	3249	1925	10252

Από την άλλη, οι λύσεις του κώδικα συγκεντρωτικά είναι:

Πίνακας 4.7: Αποτελέσματα παραδείγματος 4, από το Matlab.

Φορτίο	Κατάσταση μονάδων			Όρια Ισχύος Συνδυασμού		Έξοδος Μονάδων			Κόστος Μονάδων (κ€)			Συνολικό Κόστος
(MW)	$UN_1$	$UN_2$	$UN_3$	$P_{min}$	$P_{max}$	$P_1$	$P_2$	$P_3$	$F_1$	$F_2$	$F_3$	(κ€)
550	1	0	0	150	600	550	0	0	4658.15	0	0	4658
600	1	0	0	150	600	600	0	0	5077.91	0	0	5078
650	1	1	0	250	1000	349.34	300.66	0	3041.30	2459.13	0	5500.44
700	1	1	0	250	1000	376.97	323.03	0	3257.51	2634.16	0	5891.68
750	1	1	0	250	1000	404.61	345.39	0	3475.78	2810.85	0	6286.63
800	1	1	0	250	1000	432.24	367.76	0	3696.10	2989.21	0	6685.31
850	1	1	0	250	1000	459.87	390.13	0	3918.48	3169.23	0	7087.71
900	1	1	0	250	1000	500	400	0	4245.12	3249.18	0	7494.30

950	1	1	0	250	1000	550	400	0	4658.15	3249.18	0	7907.33
1000	1	1	1	250	1000	600	400	0	5077.91	3249.18	0	8327.09
1050	1	1	1	300	1200	600	400	50	5077.91	3249.18	506.09	8833.18
1100	1	1	1	300	1200	600	400	100	5077.91	3249.18	955.15	9282.24
1150	1	1	1	300	1200	600	400	150	5077.91	3249.18	1427.98	9755.06
1200	1	1	1	300	1200	600	400	200	5077.91	3249.18	1924.56	10251.65

Παρατηρούμε, ότι τα αποτελέσματα, είναι σχεδόν ίδια με τα εκείνα του πίνακα του παραδείγματος 5. Μικρές αριθμητικές διαφορές, οφείλονται κυρίως σε στρογγυλοποιήσεις που έγιναν κατά την επίλυση.

## Συμπεράσματα

---

Στόχος της εργασίας αυτής, ήταν η επίλυση, ενός μέρους, του προβλήματος οικονομικής κατανομής φορτίου στις μονάδες παραγωγής ενέργειας. Σκοπός του προβλήματος κατανομής είναι η παραγωγή ενέργειας, κατά τον πιο οικονομικό τρόπο. Το πρόβλημα αυτό, είναι σύνθετο και πολλές φορές περίπλοκο και έτσι η επίλυση του καθίσταται όλο και δυσκολότερη. Αφού λοιπόν, αναλύθηκε η θεωρία πίσω από τα συστήματα ενέργειας και πίσω από την οικονομική κατανομή, δημιουργήθηκε ένα λογισμικό το οποίο προσπαθεί να επιλύσει κατά τον βέλτιστο δυνατό τρόπο μερικά από τα προβλήματα που εμφανίζονται κατά την κατανομή ενέργειας, όπως τα λειτουργικά όρια, οι απώλειες λόγω μεταφοράς και τέλος, η ένταξη των μονάδων.

Οι κώδικες που παρουσιάζονται παραπάνω, αναπτύχθηκαν κυρίως για την επίλυση παραδειγμάτων από σχετικό σύγγραμμα. Συνεπώς για πολλά δεδομένα, διαφορετικά από εκείνα των παραδειγμάτων του βιβλίου, οι κώδικες, μπορεί να μην επιστρέφουν λύση, ή η λύση την οποία επιστρέφουν, να μην είναι η βέλτιστη. Παρόλα αυτά, αναπτύχθηκαν ώστε να είναι όσο το δυνατόν πιο γενικευμένοι.

Σε συνέχεια της εργασίας αυτής, θα μπορούσαν να βελτιωθούν οι κώδικες που παρουσιάστηκαν, προκειμένου να διορθωθούν λογικά και συντακτικά λάθη. Επίσης θα μπορούσε να δοθεί ένα γραφικό περιβάλλον, πάνω στο οποίο, ο χρήστης θα ορίζει μόνο τις παραμέτρους και θα βλέπει το αποτέλεσμα, ενώ στο παρασκήνιο θα επιλύεται το πρόβλημα οικονομικής κατανομής.

## Βιβλιογραφία

---

- [1] Μπακιρτζής Γ. Αναστάσιος (1998). “Οικονομική λειτουργία συστημάτων ηλεκτρικής ενέργειας”, Θεσσαλονίκη, Εκδόσεις Ζήτη
- [2] ΑΔΜΗΕ (2023). “Περιγραφή και σχεδιασμός λειτουργίας”,  
“<https://www.admie.gr/systima/leitourgia/perigrafi>”
- [3] ΑΔΜΗΕ (2023). “Monthly Energy Reports”,  
“<https://www.admie.gr/en/market/reports/monthly-energy-balance>”
- [4] Shiksha (2021). “Thermal Power Plant diagram”,  
“<https://www.shiksha.com/work-energy-and-power-preparation/thermal-power-plant-diagram-2943>”
- [5] WIKIPEDIA COMMONS (2022). “Combustion turbine diagram”,  
“<https://commons.wikimedia.org>”
- [6] Valmet forward (2023). “Combined cycle power plants”,  
“<https://www.valmet.com/>”
- [7] ResearchGate (2015). “Schematic of a diesel power plant”,  
“<https://www.researchgate.net>”
- [8] Παπαϊωάννου Β. Γρηγόριος (2014). “Βέλτιστος προγραμματισμός λειτουργίας θερμικών μονάδων παραγωγής με ελαχιστοποίηση κόστους λειτουργίας και παραγόμενων ρύπων”, Διπλωματική εργασία, Πολυτεχνείο Κρήτης.
- [9] Ψαρράς Στ. Ιωάννης (2018). “Σύστημα Ελέγχου Φόρτισης Ηλεκτρικών Οχημάτων με Βέλτιστη Λειτουργία του Δικτύου Διανομής”, Διπλωματική εργασία, Πολυτεχνείο Κρήτης.
- [10] Performance evaluation for scientific balloon station-keeping strategies considering energy management strategy  
“<https://www.sciencedirect.com/science/article>”

- [11] Electric Vehicle Enhanced Fast Charging Enabled by Battery Thermal Management and Model Predictive Control  
“<https://www.sciencedirect.com/science/article>”
- [12] Distributed optimization of station-network collaborative operation and alliance benefit allocation for integrated energy system  
“<https://www.sciencedirect.com/science/article>”
- [13] Power-to-heat plants in district heating and electricity distribution systems: A techno-economic analysis  
“<https://www.sciencedirect.com/science/article>”