



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ  
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΤΗΣ ΔΙΟΙΚΗΣΗΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

Ανάπτυξη και Συγκριτική Ανάλυση  
μετα-Ευρετικών (Υβριδικών)  
Τεχνικών: Εφαρμογή στις Επενδύσεις  
Εναλλακτικών Μορφών Ενέργειας

*Αλέξανδρος Τζανέτος*



Επιβλέπων καθηγητής: Γεώργιος Δούνιας  
10/31/2013



## Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους όσους βοήθησαν για την ολοκλήρωση αυτής της εργασίας. Αρχικά, τον κ. Δούνια, τον επιβλέποντα καθηγητή μου, καθώς και τον κ. Νίκο Θωμαΐδη, τον καθηγητή που πρότεινε το χώρο εφαρμογής που επιλέχθηκε στην εργασία: τις ανανεώσιμες πηγές ενέργειας. Επιπλέον, τον Βασίλη Βασιλειάδη, ο οποίος με καθοδήγησε στα πρώτα στάδια της εργασίας, αλλά και τον Θεοδωρή Εμμανουηλίδη, που με καθοδήγησε στο αλγοριθμικό κομμάτι της εργασίας. Επίσης τον Michele Benini, από το Energy System Scenarios Research Group, RSE (Ricerca sul Systema Energetico S.p.A.), που παρείχε τα δεδομένα που χρειαζόμουν. Θα ήταν τουλάχιστον άδικο, παρόλο που θα φανεί κοινότυπο, να μην αναφέρω τον πατέρα μου, που λόγω συγγενούς αντικειμένου σπουδών μου έλυσε κάποιες απορίες. Τέλος, πέρα από τον Αποστόλη Μισαηλίδη που βοήθησε στην επεξεργασία μερικών εικόνων, νομίζω ότι αξίζει ένα (ανώνυμο) ευχαριστώ και στον άνθρωπο που με στήριξε κυρίως προς το τέλος της εργασίας, ώστε να ολοκληρωθεί παρά τις όποιες δυσκολίες!

## Εισαγωγή

Στη σημερινή εποχή, η επιστήμη της λήψης απόφασης εξελίσσεται με ταχείς ρυθμούς. Η διαδικασία εύρεσης της βέλτιστης απόφασης παρουσιάζει αυξημένη πολυπλοκότητα, καθώς προκύπτουν νέες ανάγκες, αλλά και περιορισμοί. Σε αυτό το πλαίσιο, τα προβλήματα βελτιστοποίησης απαιτούν πιο αποτελεσματικές τεχνικές, οι οποίες να έχουν την ικανότητα αντιμετώπισης μεγάλων χώρων λύσεων, μη-γραμμικών αντικειμενικών στόχων και περιορισμών. Σε αυτήν την κατεύθυνση, ο χώρος της Τεχνητής Νοημοσύνης προσφέρει έναν αριθμό από ευφυή συστήματα και αλγορίθμους, κατάλληλους για τέτοιου είδους προβλήματα. Εμβαθύνοντας στις ευφυείς τεχνικές, οι ίδιες αποτελούν πλέον ένα πολύ σημαντικό πεδίο έρευνας και τα τελευταία χρόνια παρατηρείται συνεχής ανάπτυξη στους αλγόριθμους που στηρίζονται σε φαινόμενα της φύσης (nature-inspired intelligent algorithms). Στόχος πολλών ερευνητών αποτελεί η συνεχής βελτίωση τους, έτσι ώστε να αντιμετωπίζουν πιο αποτελεσματικά νέες προκλήσεις στο χώρο λήψης απόφασης. Στα πλαίσια της παρούσας διπλωματικής, πραγματοποιείται μια μελέτη και ανάλυση αυτών των αλγορίθμων. Ο κύριος στόχος είναι η ανάπτυξη νέων συστημάτων, είτε αναβαθμίζοντας τη λειτουργία των υπάρχοντων αλγορίθμων, είτε συνδυάζοντας μερικούς από αυτούς σε έναν υβριδικό νέο αλγόριθμο. Κατ'αυτόν τον τρόπο, ο νέος αλγόριθμος ενδέχεται να θεωρηθεί πιο «έξυπνος» και να προσεγγίζει με μεγαλύτερη ακρίβεια τη βέλτιστη λύση ενός προβλήματος. Στο πλαίσιο της πρακτικής μελέτης, τα συστήματα αυτά θα εφαρμοστούν σε συγκριτικά δείγματα δεδομένων (test datasets) από το χώρο των ανανεώσιμων πηγών ενέργειας και ειδικότερα την αιολική ενέργεια. Η εργασία εστιάζει στο αλγοριθμικό κομμάτι και τον προγραμματισμό των προτεινόμενων αλγορίθμων, ώστε να δημιουργηθεί μια σωστή βάση για περαιτέρω μελέτη στο χώρο εφαρμογής, λαμβάνοντας υπόψη περισσότερες παραμέτρους. Τέλος, δίνεται η δυνατότητα τροποποίησης αυτών των συστημάτων με απώτερο σκοπό την εφαρμογή τους σε προβλήματα βελτιστοποίησης γενικού τύπου.

## ΔΗΛΩΣΗ ΠΕΡΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

Έχω διαβάσει και κατανοήσει τους κανόνες για τη λογοκλοπή και τον τρόπο σωστής αναφοράς των πηγών που περιέχονται στον Οδηγό συγγραφής διπλωματικών εργασιών του ΤΜΟΔ. Δηλώνω ότι, από όσα γνωρίζω, το περιεχόμενο της παρούσας διπλωματικής εργασίας είναι προϊόν δικής μου δουλειάς και υπάρχουν αναφορές σε όλες τις πηγές που χρησιμοποίησα.

Ο δηλών

Αλέξανδρος Τζανέτος

# Περιεχόμενα

Ευχαριστίες.....	3
Εισαγωγή.....	4
<b>ΔΗΛΩΣΗ ΠΕΡΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ.....</b>	<b>5</b>
Περιεχόμενα.....	6
<b>ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....</b>	<b>8</b>
<b>ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ.....</b>	<b>9</b>
<b>ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ.....</b>	<b>10</b>
<b>ΚΑΤΑΛΟΓΟΣ ΑΛΓΟΡΙΘΜΩΝ.....</b>	<b>11</b>
<b>1. Παρουσίαση της εργασίας.....</b>	<b>12</b>
1.1. Γενικά.....	12
1.1.1. Ανανεώσιμες πηγές ενέργειας.....	12
1.1.2. Αιολική ενέργεια.....	15
1.1.3. Τεχνικές πληροφορίες αιολικής ενέργειας.....	17
1.1.4. Ενεργειακές πληροφορίες ανεμογεννητριών.....	21
1.1.5. Τεχνικές πληροφορίες ανεμογεννητριών.....	22
1.1.6. Είδη αιολικών πάρκων.....	25
1.2. Σκοπός της εργασίας.....	26
1.3. Δομή της εργασίας.....	26
<b>2. Βιβλιογραφική επισκόπηση.....</b>	<b>28</b>
2.1. Γενικά.....	28
2.2. Βασικές Έννοιες.....	29
2.3. Υποθέσεις/Παραδοχές του προβλήματος.....	29
<b>3. Αλγόριθμοι βελτιστοποίησης και εφαρμογές.....</b>	<b>31</b>
3.1. Ευρετικοί και μετα-ευρετικοί αλγόριθμοι.....	31
3.1.1. Simulated Annealing.....	32
3.1.2. Particle Swarm Optimization.....	34
3.1.3. Harmony Search.....	35
3.1.4. Tabu Search.....	37

3.1.5. <i>Threshold Accepting</i> .....	38
3.2. <i>Υβριδικά σχήματα</i> .....	38
<b>4. Πεδίο εφαρμογής: το πρόβλημα της κατανομής ενέργειας</b> .....	<b>39</b>
4.1. <i>Το πρόβλημα του ενεργειακού χαρτοφυλακίου</i> .....	39
4.2. <i>Αντικειμενικοί στόχοι</i> .....	41
4.3. <i>Δεδομένα</i> .....	41
4.4. <i>Μεθοδολογία</i> .....	50
4.5. <i>Παράμετροι</i> .....	53
<b>5. Αποτελέσματα</b> .....	<b>54</b>
<b>6. Συμπεράσματα</b> .....	<b>58</b>
<b>Βιβλιογραφία</b> .....	<b>59</b>
<b>ΠΑΡΑΡΤΗΜΑ: ΑΛΓΟΡΙΘΜΟΙ</b> .....	<b>62</b>

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Περιγραφή	Σελ.
<b>Πίνακας 4-1:</b> Εγκατεστημένη ισχύς στην Ιταλία (ανά ζώνη) (πηγή: <a href="http://www.terna.it/">http://www.terna.it/</a> )	-42-
<b>Πίνακας 4-2:</b> Παράμετροι αλγορίθμων	-53-
<b>Πίνακας 5-1:</b> Αποτελέσματα Simulated Annealing	-54-
<b>Πίνακας 5-2:</b> Αποτελέσματα Particle Swarm Optimization	-54-
<b>Πίνακας 5-3:</b> Αποτελέσματα Threshold Accepting	-54-
<b>Πίνακας 5-4:</b> Αποτελέσματα Harmony Search	-55-
<b>Πίνακας 5-5:</b> Αποτελέσματα Tabu Search	-55-
<b>Πίνακας 5-6:</b> Αποτελέσματα υβριδικού σχήματος Simulated Annealing και Harmony Search	-56-
<b>Πίνακας 5-7:</b> Αποτελέσματα ελέγχου αποδοτικότητας	-57-
<b>Πίνακας 6-1:</b> Προτεινόμενα χαρτοφυλάκια	-58-



## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Περιγραφή	Σελ.
<b>Σχήμα 1-1:</b> Ηλεκτρική παραγωγή από ανανεώσιμες πηγές ενέργειας (Πηγή: Renewable Energy: Sources For Fuels and Electricity)	-12-
<b>Σχήμα 1-2:</b> Μηχανή Smith-Putnam (Πηγή: Μαρούδας, 2008)	-15-
<b>Σχήμα 1-3:</b> Ανεμογεννήτρια Mod-5B (Πηγή: Μαρούδας, 2008)	-16-
<b>Σχήμα 1-4:</b> Ανεμογεννήτρια Darrieus (Πηγή: Μαρούδας, 2008)	-16-
<b>Σχήμα 1-5:</b> Συνάρτηση πυκνότητας πιθανότητας ταχυτήτων ανέμου (Πηγή: Μαρούδας, 2008)	-18-
<b>Σχήμα 1-6:</b> Συνάρτηση πυκνότητας ισχύος ανέμου και ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)	-19-
<b>Σχήμα 1-7:</b> καμπύλη ισχύος ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)	-21-
<b>Σχήμα 1-8:</b> Βαθμός απόδοσης ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)	-21-
<b>Σχήμα 1-9:</b> Σχηματική αναπαράσταση ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)	-23-
<b>Σχήμα 1-10:</b> Βασικά δομικά χαρακτηριστικά ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)	-24-
<b>Σχήμα 1-11:</b> Χερσαίο αιολικό πάρκο (Πηγή: Μαρούδας, 2008)	-25-
<b>Σχήμα 1-12:</b> Αιολικό πάρκο κοντά στην ακτή (Πηγή: Μαρούδας, 2008)	-25-
<b>Σχήμα 1-13:</b> Υπεράκτιο αιολικό πάρκο (Πηγή: Μαρούδας, 2008)	-25-
<b>Σχήμα 4-1:</b> Ενεργειακή απόδοση ανεμογεννήτριας για δεδομένες τιμές ανέμου (Πηγή: www.wind-power-program.com)	-39-
<b>Σχήμα 4-2:</b> Απεικόνιση περιοχών βάση TERNA (πηγή: <a href="http://www.terna.it/">http://www.terna.it/</a> )	-43-

## ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Περιγραφή	Σελ.
<b>Διάγραμμα 4-1:</b> Χρονοσειρά δεδομένων της CNOR	-44-
<b>Διάγραμμα 4-2:</b> Χρονοσειρά δεδομένων της CSUD	-45-
<b>Διάγραμμα 4-3:</b> Χρονοσειρά δεδομένων της NORD	-46-
<b>Διάγραμμα 4-4:</b> Χρονοσειρά δεδομένων της SARD	-47-
<b>Διάγραμμα 4-5:</b> Χρονοσειρά δεδομένων της SICI	-48-
<b>Διάγραμμα 4-6:</b> Χρονοσειρά δεδομένων της SUD	-49-

## ΚΑΤΑΛΟΓΟΣ ΑΛΓΟΡΙΘΜΩΝ

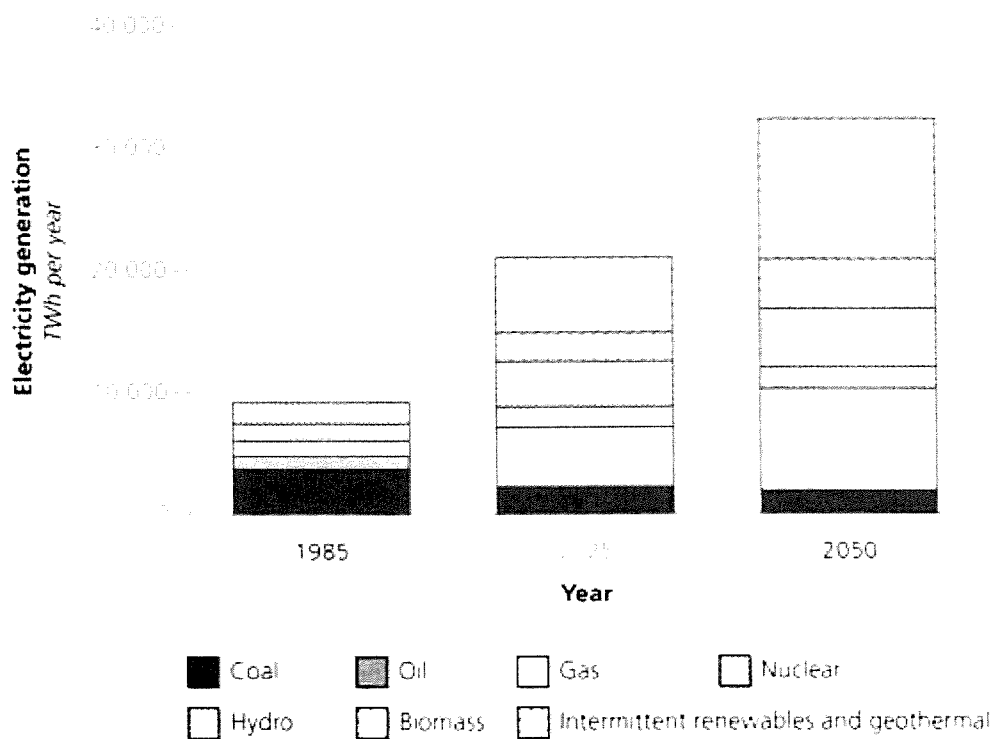
Περιγραφή	Σελ.
Αλγόριθμος Π1: Συνάρτηση randfixedsum	-62-
Αλγόριθμος Π2: Συνάρτηση SA_energy_dispatch	-64-
Αλγόριθμος Π3: Συνάρτηση PSO_energy_dispatch	-66-
Αλγόριθμος Π4: Συνάρτηση HS_energy_dispatch	-70-
Αλγόριθμος Π5: Συνάρτηση TS_energy_dispatch	-72-
Αλγόριθμος Π6: Συνάρτηση TA_energy_dispatch	-75-
Αλγόριθμος Π7: Συνάρτηση choose_tac	-78-
Αλγόριθμος Π8: Συνάρτηση test_energy_dispatch	-79-

# 1. Παρουσίαση της εργασίας

## 1.1. Γενικά

### 1.1.1. Ανανεώσιμες πηγές ενέργειας

Τα τελευταία χρόνια παρατηρείται μια συνεχής αύξηση στη χρήση των ανανεώσιμων πηγών ενέργειας. Αυτό είναι απόρροια τόσο της σταδιακής μείωσης των αποθεμάτων πετρελαίου όσο και της ανάγκης για ανακάλυψη νέων μορφών ενέργειας. Όπως φαίνεται και στο παρακάτω σχήμα (Johansson T.B., Bumham L. 1993) σταδιακά θα αυξηθεί και τελικά θα γίνει η πιο σημαντική πηγή ενέργειας η αιολική και γενικότερα οι ανανεώσιμες.



Σχήμα 1-1: Ηλεκτρική παραγωγή από ανανεώσιμες πηγές ενέργειας (Πηγή: Renewable Energy: Sources For Fuels and Electricity)

Οι ανανεώσιμες μορφές ενέργειας (ΑΠΕ) ή ήπιες μορφές ενέργειας, είναι μορφές εκμεταλλεύσιμης ενέργειας που προέρχονται από διάφορες φυσικές διαδικασίες, όπως ο άνεμος, η γεωθερμία, η κυκλοφορία του νερού και άλλες. Ο όρος «ήπιες» αναφέρεται σε δυο βασικά χαρακτηριστικά τους. Πρώτον, για την εκμετάλλευσή τους δεν απαιτείται κάποια ενεργητική

παρέμβαση, όπως εξόρυξη, άντληση ή καύση, όπως με τις μέχρι τώρα χρησιμοποιούμενες πηγές ενέργειας, αλλά απλώς η εκμετάλλευση της ήδη υπάρχουσας ροής ενέργειας στη φύση. Δεύτερον, πρόκειται για «καθαρές» μορφές ενέργειας, πολύ «φιλικές» στο περιβάλλον, που δεν αποδесμεύουν υδρογονάνθρακες, διοξείδιο του άνθρακα ή τοξικά και ραδιενεργά απόβλητα, όπως οι υπόλοιπες πηγές ενέργειας που χρησιμοποιούνται σε μεγάλη κλίμακα. Έτσι οι ΑΠΕ θεωρούνται από πολλούς μία αφετηρία για την επίλυση των οικολογικών προβλημάτων που αντιμετωπίζει η Γη.

Ως «ανανεώσιμες πηγές» θεωρούνται γενικά οι εναλλακτικές των παραδοσιακών πηγών ενέργειας (π.χ. του πετρελαίου ή του άνθρακα), όπως η ηλιακή και η αιολική. Ο χαρακτηρισμός «ανανεώσιμες» είναι κάπως καταχρηστικός, μιας και ορισμένες από αυτές τις πηγές, όπως η γεωθερμική ενέργεια δεν ανανεώνονται σε κλίμακα χιλιετιών. Σε κάθε περίπτωση οι ΑΠΕ έχουν μελετηθεί ως λύση στο πρόβλημα της αναμενόμενης εξάντλησης των (μη ανανεώσιμων) αποθεμάτων ορυκτών καυσίμων. Τελευταία από την Ευρωπαϊκή Ένωση, αλλά και από πολλά μεμονωμένα κράτη, υιοθετούνται νέες πολιτικές για τη χρήση ανανεώσιμων πηγών ενέργειας, που προάγουν τέτοιες εσωτερικές πολιτικές και για τα κράτη μέλη. Οι ΑΠΕ αποτελούν τη βάση του μοντέλου οικονομικής ανάπτυξης της πράσινης οικονομίας και κεντρικό σημείο εστίασης της σχολής των οικολογικών οικονομικών, η οποία έχει κάποια επιρροή στο οικολογικό κίνημα.

Γενικά, οι ήπιες μορφές ενέργειας είναι:

- **Αιολική ενέργεια:** Χρησιμοποιήθηκε παλιότερα για την άντληση νερού από πηγάδια καθώς και για μηχανικές εφαρμογές (π.χ. την άλεση στους ανεμόμυλους). Έχει αρχίσει να χρησιμοποιείται ευρέως για ηλεκτροπαραγωγή.
- **Ηλιακή ενέργεια:** Χρησιμοποιείται περισσότερο για θερμικές εφαρμογές (ηλιακοί θερμοσίφωνες και φούρνοι) ενώ η χρήση της για την παραγωγή ηλεκτρισμού έχει αρχίσει να κερδίζει έδαφος, με την βοήθεια της πολιτικής προώθησης των Ανανεώσιμων Πηγών Ενέργειας από το ελληνικό κράτος και την Ευρωπαϊκή Ένωση.
- **Υδραυλική ενέργεια:** Είναι τα γνωστά υδροηλεκτρικά έργα, που στο πεδίο των ήπιων μορφών ενέργειας εξειδικεύονται περισσότερο στα μικρά υδροηλεκτρικά. Είναι η πιο διαδεδομένη μορφή ανανεώσιμης ενέργειας.

- **Βιομάζα:** Χρησιμοποιεί τους υδατάνθρακες των φυτών (κυρίως αποβλήτων της βιομηχανίας ξύλου, τροφίμων και ζωοτροφών και της βιομηχανίας ζάχαρης) με σκοπό την αποδέσμευση της ενέργειας που δεσμεύτηκε απ' το φυτό με τη φωτοσύνθεση. Ακόμα μπορούν να χρησιμοποιηθούν αστικά απόβλητα και απορρίμματα. Μπορεί να δώσει βιοαιθανόλη και βιοαέριο, που είναι καύσιμα πιο φιλικά προς το περιβάλλον από τα παραδοσιακά. Είναι μια πηγή ενέργειας με πολλές δυνατότητες και εφαρμογές που θα χρησιμοποιηθεί πλατιά στο μέλλον.
- **Γεωθερμική ενέργεια:** Προέρχεται από τη θερμότητα που παράγεται απ' τη ραδιενεργό αποσύνθεση των πετρωμάτων της γης. Είναι εκμεταλλεύσιμη εκεί όπου η θερμότητα αυτή ανεβαίνει με φυσικό τρόπο στην επιφάνεια, π.χ. στους θερμοπίδακες ή στις πηγές ζεστού νερού. Μπορεί να χρησιμοποιηθεί είτε απευθείας για θερμικές εφαρμογές είτε για την παραγωγή ηλεκτρισμού. Η Ισλανδία καλύπτει το 80-90% των ενεργειακών της αναγκών, όσον αφορά τη θέρμανση, και το 20%, όσον αφορά τον ηλεκτρισμό, με γεωθερμική ενέργεια.
- **Ενέργεια από τη θάλασσα**
- **Ενέργεια από παλίρροιες:** Εκμεταλλεύεται τη βαρύτητα του Ήλιου και της Σελήνης, που προκαλεί ανύψωση της στάθμης του νερού. Το νερό αποθηκεύεται καθώς ανεβαίνει και για να ξανακατέβει αναγκάζεται να περάσει μέσα από μια τουρμπίνα, παράγοντας ηλεκτρισμό. Έχει εφαρμοστεί στην Αγγλία, τη Γαλλία, τη Ρωσία και αλλού.
- **Ενέργεια από κύματα:** Εκμεταλλεύεται την κινητική ενέργεια των κυμάτων της θάλασσας.
- **Ενέργεια από τους ωκεανούς:** Εκμεταλλεύεται τη διαφορά θερμοκρασίας ανάμεσα στα στρώματα του ωκεανού, κάνοντας χρήση θερμικών κύκλων. Βρίσκεται στο στάδιο της έρευνας.

### 1.1.2. Αιολική ενέργεια<sup>1</sup>

Η αιολική ενέργεια χρησιμοποιούνταν ανέκαθεν για διάφορους σκοπούς: ναυτιλία (κίνηση ιστιοφόρων πλοίων), άρδευση, αγροτικές καλλιέργειες, εμπορικές συναλλαγές, συγκοινωνία, εξερευνήσεις.

Για την παραγωγή ενέργειας, πρώτη φορά, χρησιμοποιήθηκε από τους Άραβες, οι οποίοι και εγκατέστησαν τους πρώτους ανεμόμυλους με κάθετους άξονες. Αργότερα, πέρασε στην Ευρώπη μέσω των σταυροφόρων, η ιδέα των ανεμόμυλων. Εκεί (κυρίως το Μεσαίωνα) χρησιμοποιήθηκαν ανεμόμυλοι οριζόντιου άξονα.

Αυτή τη στιγμή, σύμφωνα με τον ιστότοπο της European Wind Energy Association, πάνω από 300.000 ανεμόμυλοι, είναι εγκατεστημένοι στην Ευρώπη.

Το 19ο αιώνα εγκαταστάθηκαν στην Αμερική ανεμόμυλοι τύπου «Western Wheel», οι οποίοι είχαν περίπου 20 πτερύγια, ήταν φτιαγμένοι από ατσάλι και χρησίμευαν κυρίως για άρδευση. Μέχρι τα μέσα του 20ου αιώνα οι εγκαταστάσεις τέτοιου τύπου ανεμόμυλων είχαν φτάσει τα 8 εκατομμύρια.

Στα τέλη του 19ου αιώνα έγινε η πρώτη απόπειρα να χρησιμοποιηθεί αιολική μηχανή για την παραγωγή ηλεκτρικής ενέργειας, με την ανεμογεννήτρια συνεχούς ρεύματος ισχύος 12 kW, που κατασκευάστηκε στις ΗΠΑ από τον Brush. Παρόλα αυτά, δεν υπήρξε μεγάλο ενδιαφέρον για την αξιοποίηση της αιολικής ενέργειας κατά τον 20ο αιώνα, με μόνη εξαίρεση ίσως την ανεμογεννήτρια των Smith-Putnam (Μαρούδας, 2008), που κατασκευάστηκε το 1941 στις ΗΠΑ. Αξιοσημείωτο για την εποχή ήταν το γεγονός ότι η μηχανή αυτή είχε ατσάλινο ρότορα διαμέτρου 53 m και ονομαστική ισχύ 1250 kW.



Σχήμα 1-2: Μηχανή Smith-Putnam (Πηγή: Μαρούδας, 2008)

<sup>1</sup> Μαρούδας (2008)

Ύστερα, στη Δανία, ο Gedser το 1957 εγκατέστησε μια ανεμογεννήτρια ισχύος 200 kW και διάμετρο ρότορα 24 m και ο Hütter την ανεμογεννήτρια Hütter W34, το 1958, η οποία είχε ισχύ 100 kW και διάμετρο ρότορα 34 m.



**Σχήμα 1-3:**  
Ανεμογεννήτρια  
Mod-5B (Πηγή:  
Μαρούδας, 2008)

Αργότερα, τη δεκαετία του '70, η συνεχής αύξηση της τιμής του πετρελαίου οδήγησε τις περισσότερες χώρες να προωθήσουν κυβερνητικά χρηματοδοτούμενα προγράμματα για έρευνα και ανάπτυξη ανεμογεννητριών (Μαρούδας, 2008). Έτσι στις ΗΠΑ, το 1975 δημιουργήθηκε μια σειρά ανεμογεννητριών με πρώτη την Mod-0 (διαμέτρου 38 m και ισχύος 100kW) και τελευταία την Mod-5B (διαμέτρου 97,5 m και ισχύος 2,5 MW).

Οι κατασκευές συνεχίστηκαν, αλλά πλέον με σκοπό να βρεθεί ποια κατασκευή θα ήταν πιο οικονομική, οπότε στις χώρες που υπήρχαν ανεμογεννήτριες (Σουηδία, Γερμανία, Βρετανία) επιχορηγήθηκαν οι έρευνες.

Στον Καναδά, ωστόσο, κατασκευάστηκε η πρώτη ανεμογεννήτρια με κάθετο άξονα, η Darrieus με ισχύ 4 MW και σηματοδότησε την αρχή της εκμετάλλευσης της αιολικής ενέργειας σε μεγαλύτερη κλίμακα. Χαρακτηριστικά, στην Καλιφόρνια, το 1980, εγκαταστάθηκαν ανεμογεννήτριες συνολικής ισχύος 1500 MW, οι πρώτες εξ αυτών ήταν ισχύος 50kW η κάθε μία και ύστερα εμφανίστηκαν άλλες των 100, 150, 250 και 500 kW. Αυτό συντέλεσε στη μαζική παραγωγή ανεμογεννητριών και σταδιακά στη βελτίωση στον τομέα της αξιοπιστίας αυτών των συστημάτων. Τελικά, αργότερα κατασκευάστηκαν ανεμογεννήτριες ισχύος από 500 έως 1500 kW με πολλές δυνατότητες.



**Σχήμα 1-4:**  
Ανεμογεννήτρια Darrieus  
(Πηγή: Μαρούδας, 2008)



### 1.1.3. Τεχνικές πληροφορίες αιολικής ενέργειας<sup>2</sup>

Μας ενδιαφέρει και εξετάζουμε τη χαμηλότερη περιοχή της ατμόσφαιρας, το ατμοσφαιρικό οριακό στρώμα, όπου εκεί η κίνηση του ανέμου επιβραδύνεται από δυνάμεις τριβής και μεγάλα εμπόδια που συναντά στην επιφάνεια της γης, καθώς και από φαινόμενα που οφείλονται στον στροβιλισμό. Ο στροβιλισμός (μηχανικής ή θερμικής προέλευσης) προκαλεί απότομες μεταβολές στην ταχύτητα του ανέμου σε ένα μεγάλο εύρος συχνοτήτων και πλατών, γνωστές ως ριπές. Επιπλέον, σημαντική πληροφορία είναι το γεγονός ότι όσο απομακρυνόμαστε από το έδαφος, τόσο αυξάνεται η ταχύτητα του ανέμου και άρα εκμεταλλευόμαστε καλύτερα την αιολική ενέργεια.

Θεωρούμε αντιπροσωπευτική τιμή των ταχυτήτων του ανέμου την μέση τιμή, που δίνεται από τον τύπο:

$$U = \frac{1}{T} \int_{t_0-T/2}^{t_0+T/2} u(t) dt$$

Όπου:

$u(t)$  η στιγμιαία ταχύτητα του ανέμου

$T$  το χρονικό διάστημα των μετρήσεων των ταχυτήτων του ανέμου

Ωστόσο, για να υπολογίσουμε το αξιοποιήσιμο αιολικό δυναμικό χρησιμοποιούμε μια άλλη συνάρτηση, τη συνάρτηση Weibull:

$$P(U) = \frac{k}{V} \left(\frac{U}{C}\right)^{k-1} e^{-\left(\frac{U}{C}\right)^k}$$

Όπου:

$P(U)$  η συχνότητα εμφάνισης της ταχύτητας του ανέμου

$C$  ονομάζεται παράμετρος κλίμακας ή χαρακτηριστική ταχύτητα ανέμου

$k$  ονομάζεται παράμετρος μορφής

Για τα νησιά του Αιγαίου, να σημειωθεί ότι η παράμετρος μορφής κυμαίνεται από 1,4 μέχρι 2. Όσο μικρότερη η τιμή της παραμέτρου, τόσο η κατανομή γίνεται πιο ομοιόμορφη και οι ταχύτητες κοντά στη μέση είναι συχνότερες.

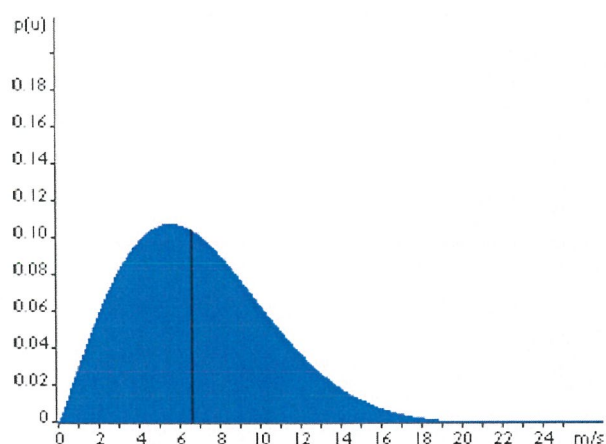
---

<sup>2</sup> Μαρούδας (2008)

Για να υπολογίσουμε την καμπύλη πυκνότητας πιθανότητας των ταχυτήτων του ανέμου χρειάζεται να διεξαχθούν πειράματα και να γνωρίζουμε τη μορφολογία της περιοχής έτσι ώστε οι ταχύτητες να αναχθούν από το ύψος των μετεωρολογικών μετρήσεων στο ύψος του ρότορα της ανεμογεννήτριας. Αυτή η αναγωγή γίνεται με τη βοήθεια του τύπου:

$$u = u_0 \left( \frac{z}{z_0} \right)^{\alpha}$$

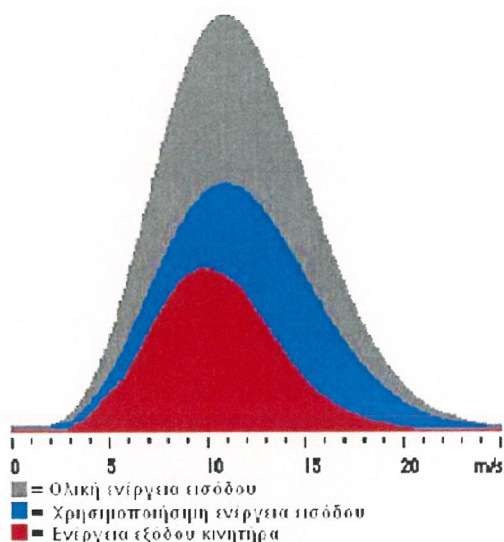
όπου  $\alpha$  παράμετρος που αυξάνεται όσο πιο έντονη είναι η μορφολογία και η τραχύτητα του εδάφους.



**Σχήμα 1-5:** Συνάρτηση πυκνότητας πιθανότητας ταχυτήτων ανέμου (Πηγή: Μαρούδας, 2008)

Η παραπάνω εικόνα δείχνει το πλήθος των ταχυτήτων του ανέμου που παρατηρούνται. Για να δημιουργηθεί το γράφημα χρησιμοποιούμε τον τύπο της ισχύος του ανέμου , όπου  $\rho$  η πυκνότητα του αέρα,  $A$  η επιφάνεια από την οποία διέρχεται ο άνεμος και  $u$  η ταχύτητα του ανέμου.

Για τον υπολογισμό της αιολικής ισχύος δε λαμβάνουμε υπ' όψιν την μέση ταχύτητα, αλλά πολλαπλασιάζουμε κάθε πιθανότητα μιας ορισμένης ταχύτητας ανέμου (από τη συνάρτηση Weibull) με την ισχύ που παράγεται σε αυτή την ταχύτητα. Έτσι σχηματίζεται η θεωρητική κατανομή της αιολικής ισχύος για κάθε ταχύτητα:



Σχήμα 1-6: Συνάρτηση πυκνότητας ισχύος ανέμου και ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)

Μια ανεμογεννήτρια, όμως, είναι αδύνατο να παραλάβει όλη αυτή την ισχύ, διότι τότε ο αέρας που θα περνούσε πίσω από την έλικα της ανεμογεννήτριας δε θα

απομακρυνόταν γιατί θα είχε μηδενική κινητική ενέργεια στην έξοδο. Έτσι δε παραγόταν ενέργεια, αφού ο στάσιμος αέρας στην έξοδο δε θα άφηνε νέο αέρα να εισέλθει από την είσοδο.

Έχει αποδειχθεί ότι μια ιδανική ανεμογεννήτρια θα πρέπει να μειώσει την αρχική ταχύτητα του αέρα κατά τα 2/3 της. Επιπλέον, σύμφωνα με τον νόμο του Albert Betz ισχύει: Η μέγιστη μηχανική ισχύς που μπορεί να παραχθεί από αιολική ενέργεια είναι ίση με το 59,3% αυτής.

Έτσι, στο σχήμα (1-5) παρουσιάζεται η θεωρητική ισχύς ανά ροής αέρα (γκρι περιοχή), το 59,3% της θεωρητικής ισχύς (μπλε περιοχή) και η πραγματική ηλεκτρική ισχύς που παράγεται από την ανεμογεννήτρια (κόκκινη περιοχή).

Οπότε συμπεραίνουμε ότι το μεγαλύτερο μέρος της ηλεκτρικής ενέργειας από ανεμογεννήτρια κατά τη διάρκεια ενός έτους το παίρνουμε από ταχύτητες ανέμου μεγαλύτερες της μέσης. Για να προσδιορίσουμε τώρα την ισχύ που αντιπροσωπεύει η κόκκινη περιοχή, πρέπει να λάβουμε υπ' όψιν την καμπύλη ισχύος της ανεμογεννήτριας (power curve). Η καμπύλη αυτή καθορίζει την απόδοση της ανεμογεννήτριας συναρτήσει της ταχύτητας του ανέμου και ποικίλει ανάλογα τον τύπο της ανεμογεννήτριας.

Τώρα, αν πολλαπλασιάσουμε την πιθανότητα εμφάνισης μιας ταχύτητας ανέμου με την ισχύ που αποδίδει αυτή η ανεμογεννήτρια σε αυτή την ταχύτητα, προσδιορίζεται η συνεισφορά της κάθε ταχύτητας του ανέμου στη μέση ετήσια ισχύ της ανεμογεννήτριας. Εν συνεχεία, το άθροισμα αυτών των γινομένων (η μέση ετήσια ισχύς δηλαδή) πολλαπλασιάζεται με τη διαθεσιμότητα της ανεμογεννήτριας και με τις 8760 ώρες του έτους. Η διαθεσιμότητα των σύγχρονων ανεμογεννητριών είναι τουλάχιστον 98%.

Ένας άλλος παράγοντας που πρέπει να λάβουμε υπ' όψιν είναι η μέση θερμοκρασία της περιοχής που μας ενδιαφέρει, καθώς κάθε καμπύλη ισχύος της ανεμογεννήτριας έχει υπολογιστεί για 15°C. Μια αύξηση της θερμοκρασίας θα οδηγούσε σε μείωση της πυκνότητας του αέρα και κατ' επέκταση μείωση της ισχύος της ανεμογεννήτριας.

Αυτό εύκολα συμπεραίνεται και από τον τύπο της ισχύος  $P = 0,5 \cdot \rho \cdot u^3 \cdot A \cdot C_f$ , όπου  $C_f$  είναι ο συντελεστής άνωσης, ο οποίος είναι μοναδικός για κάθε ανεμογεννήτρια,  $\rho$  η πυκνότητα του αέρα,  $A$  η επιφάνεια που διέρχεται ο άνεμος και  $u$  η ταχύτητα αυτού. Ο συντελεστής άνωσης είναι ένας τρόπος έκφρασης του ποσοστού ενέργειας που έχει ο άνεμος λίγο πριν τον ανεμοκινητήρα και δίνεται από τη σχέση  $C_f = 4a(1 - a)^2$ , όπου ο συντελεστής  $a$  ισούται με το λόγο της διαφοράς των ταχυτήτων του ανέμου πριν και μετά το δρομέα, προς την ταχύτητα που έχει πριν το δρομέα. Η μέγιστη τιμή του  $C_f$  είναι 59%, το λεγόμενο όριο του Betz. Ο συντελεστής άνωσης είναι συνάρτηση της γεωμετρίας των πτερυγίων και του λόγου της ταχύτητας προς την ταχύτητα του αέρα:

$$\lambda = \frac{\omega R}{V}$$

Όπου:

$\omega$  είναι η γωνιακή ταχύτητα των πτερυγίων

$R$  η ακτίνα των πτερυγίων

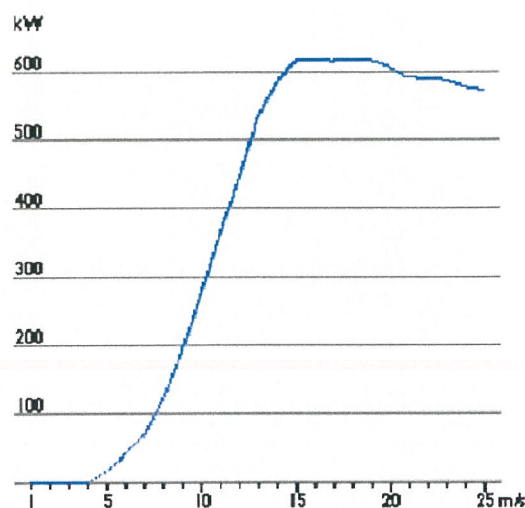
$V$  η μέση τιμή της ταχύτητας του ανέμου στο ύψος του άξονα του δρομέα

#### 1.1.4. Ενεργειακές πληροφορίες ανεμογεννητριών

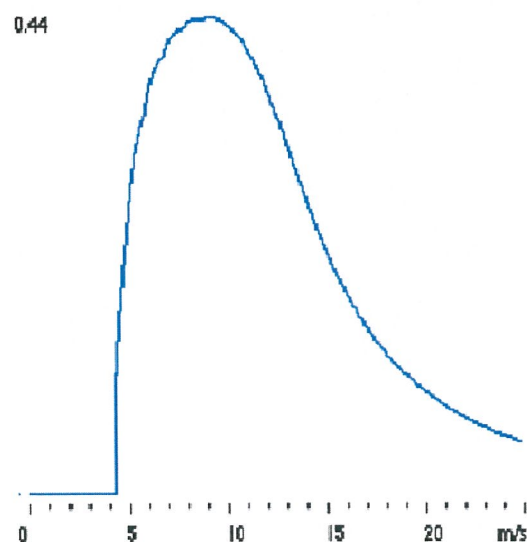
Για να προσδιοριστεί ο βαθμός απόδοσης μιας ανεμογεννήτριας σε συνάρτηση με την ταχύτητα του ανέμου, πρέπει να διαιρεθεί κάθε ισχύς του ανέμου στην αντίστοιχη ταχύτητα (σχήμα 1-7) με τη θεωρητική ισχύ του ανέμου σε αυτή την ταχύτητα. Έτσι προκύπτει ο βαθμός απόδοσής της, που φαίνεται στο σχήμα (1-8).

Επίσης ένα άλλο μέγεθος μιας ανεμογεννήτριας που εξαρτάται από τις ταχύτητες του ανέμου κατά τη διάρκεια του έτους, τον τόπο εγκατάστασής της και τα τεχνικά χαρακτηριστικά της είναι ο συντελεστής εκμετάλλευσής της, που είναι η έκφραση του λόγου της ετήσιας παραγόμενης ενέργειας προς αυτήν που θα παραγόταν αν η ανεμογεννήτρια λειτουργούσε στην ονομαστική ισχύ της συνεχώς στη διάρκεια του έτους. Με άλλα λόγια, ο συντελεστής εκμετάλλευσής, δείχνει πόσο αξιοποιείται η ανεμογεννήτρια στον τόπο εγκατάστασής της.

Τυχόν αλλαγές στα μεγέθη της ανεμογεννήτριας επιφέρουν βελτιστοποιήσεις αυτής. Αλλάζοντας το μέγεθος των πτερυγίων ή το μέγεθος της γεννήτριας μεταβάλλεται η καμπύλη ισχύος της, ενώ μια μεταβολή του ύψους επηρεάζει τις τιμές των ταχυτήτων ανέμου που θα λειτουργεί η ανεμογεννήτρια. Γενικά, μείζον θέμα είναι να επιτυγχάνεται ο μεγαλύτερος βαθμός απόδοσης μιας ανεμογεννήτριας.



Σχήμα 1-7: καμπύλη ισχύος ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)



Σχήμα 1-8: Βαθμός απόδοσης ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)

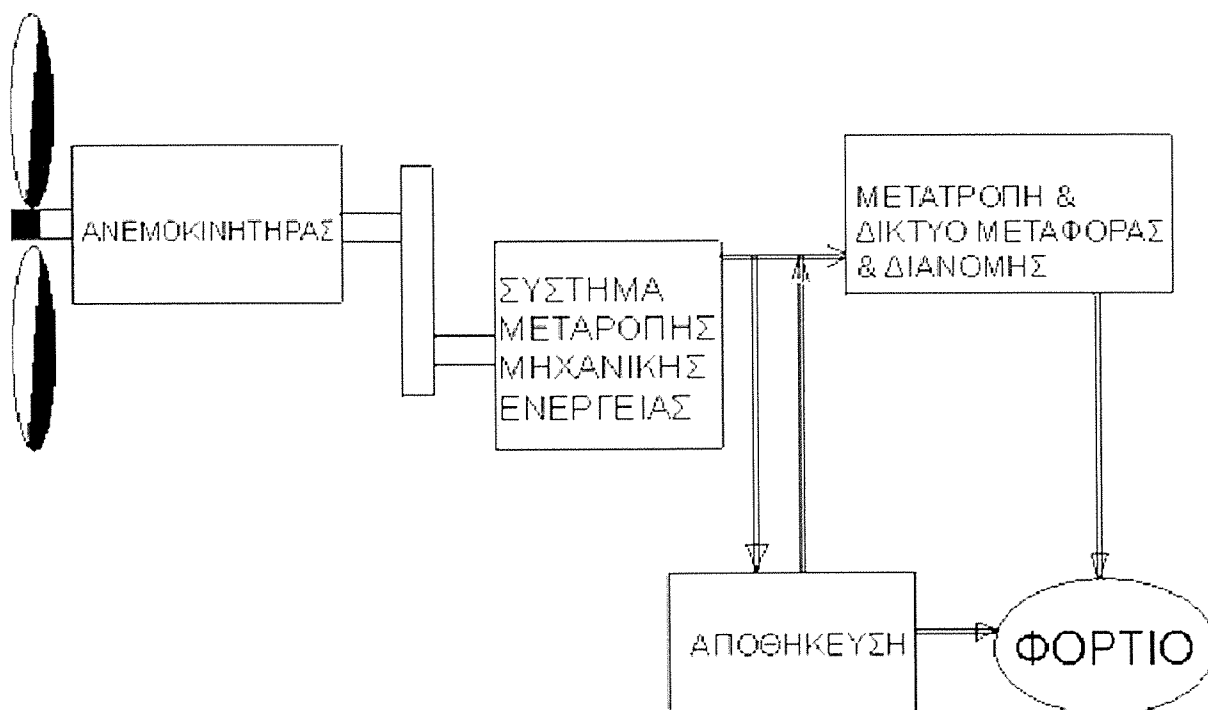
Σημασία έχουν και χαρακτηριστικά όπως το μέγεθος της έλικας αλλά και το μέγεθος της ανεμογεννήτριας συνολικά. Ωστόσο, για μια μεγάλη έλικα, όσο μεγαλύτερη είναι η γεννήτρια και οι ταχύτητες του ανέμου, τόσο περισσότερη ενέργεια παράγεται, όμως δε θα αποδίδει τόσο σε μέσες και μικρές τιμές ταχυτήτων. Αντίστροφα, για μικρότερη γεννήτρια, θα αποδίδει καλύτερα σε μικρές τιμές ανέμου και όχι τόσο καλά σε μεγαλύτερες.

Επιπρόσθετα, οι μικρές ανεμογεννήτριες παρόλο που παράγουν μικρά ποσά ενέργειας, δεν απαιτούν ισχυρό δίκτυο, παράγουν σχετικά πιο σταθερή ισχύ, απαιτούν μικρότερο κόστος θεμελίωσης, είναι αισθητικά πιο αποδεκτές από τους κατοίκους και καθιστούν πολλές μαζί ένα αιολικό πάρκο πιο αξιόπιστο.

Βλέπει κανείς, λοιπόν, ότι δεν είναι πάντα καλύτερη λύση μια μεγαλύτερη ανεμογεννήτρια.

#### ***1.1.5. Τεχνικές πληροφορίες ανεμογεννητριών***

Η αξιοποίηση της ενέργειας που παράγει μια ανεμογεννήτρια, γίνεται μέσω ενός συστήματος που μετατρέπει την κινητική ενέργεια του άξονα της γεννήτριας σε ηλεκτρική και από εκεί διοχετεύεται στο σύστημα.

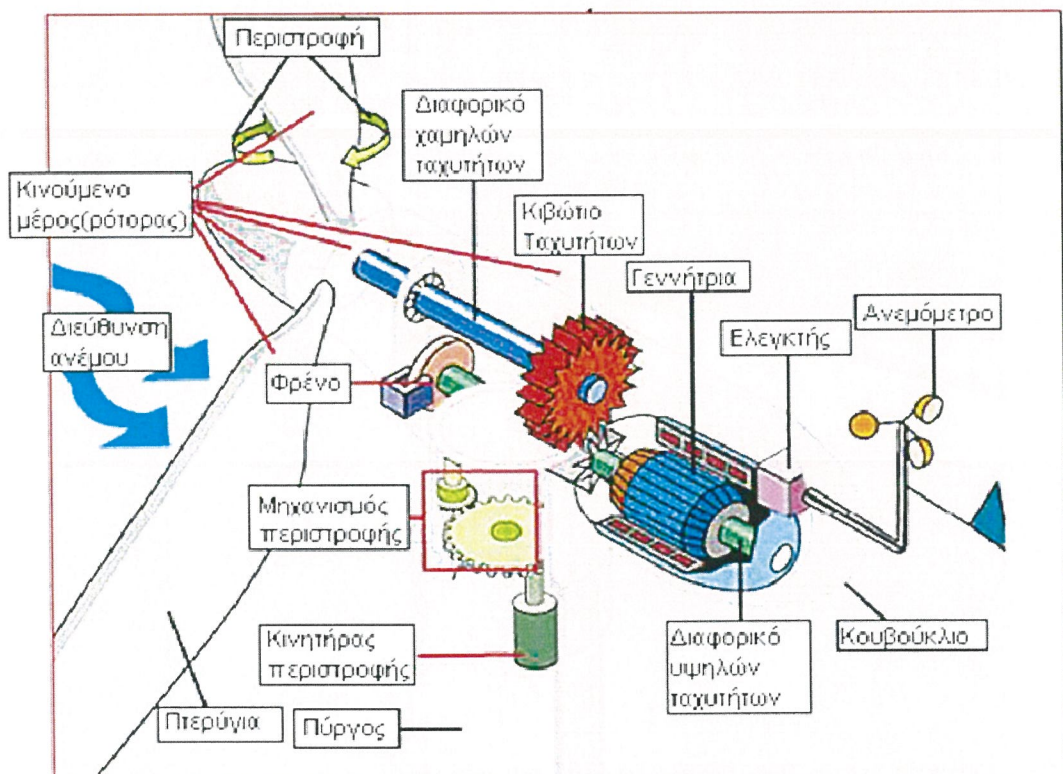


Σχήμα 1-9: Σχηματική αναπαράσταση ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)

Τα βασικά δομικά χαρακτηριστικά μιας ανεμογεννήτριας είναι:

1. Κουβούκλιο, όπου περιέχονται όλα τα βασικά στοιχεία της ανεμογεννήτριας
2. Πτερύγια (μέχρι και 27m μπορεί να είναι το καθένα)
3. Κεφαλή
4. Διαφορικό χαμηλών ταχυτήτων, το οποίο συνδέει την κεφαλή του ρότορα με το κιβώτιο ταχυτήτων
5. Κιβώτιο ταχυτήτων, που μεταφέρει την κίνηση από το διαφορικό χαμηλών ταχυτήτων στο διαφορικό υψηλών ταχυτήτων, περιστρέφοντάς το μέχρι και 50 γρηγορότερα από το διαφορικό χαμηλών στροφών
6. Διαφορικό υψηλών ταχυτήτων, το οποίο περιστρέφεται με σχεδόν 1500 στροφές ανά λεπτό και οδηγεί την ηλεκτρική γεννήτρια
7. Ηλεκτρική γεννήτρια (μεταξύ 600 και 3000 kW)
8. Μηχανισμός περιστροφής
9. Ηλεκτρονικός ελεγκτής, που περιέχει έναν υπολογιστή που παρακολουθεί συνεχώς την κατάσταση της ανεμογεννήτριας, με δικαίωμα παύσης της λειτουργίας της εφόσον υπάρχει πρόβλημα

10. Ανεμόμετρο και ανεμοδείκτης, που μετρούν κατεύθυνση και ένταση ανέμου
11. Πύργος, που στηρίζει το κουβούκλιο και τα κινούμενα μέρη της
12. Μονάδα ψύξης, που περιέχει έναν ηλεκτρικό ανεμιστήρα για να ψύχει την ηλεκτρική γεννήτρια
13. Υδραυλικό σύστημα, που χρησιμοποιείται για να επαναφέρει τα αεροδυναμικά φρένα της ανεμογεννήτριας



Σχήμα 1-10: Βασικά δομικά χαρακτηριστικά ανεμογεννήτριας (Πηγή: Μαρούδας, 2008)

Όσον αφορά τα πτερόγια, στις μεγάλες ανεμογεννήτριες, έχουν τη δυνατότητα περιστροφής γύρω από τον άξονά τους. Δέχονται τον άνεμο κατά μήκος τους υπό διαφορετικές γωνίες, που πιο κοντά στη βάση τους αυξάνεται η γωνία πρόσπτωσης. Προκειμένου να υπάρχει ενιαία γωνία πρόσπτωσης, τα πτερόγια έχουν περιστροφική μορφή.



### 1.1.6. Είδη αιολικών πάρκων

Υπάρχουν 3 είδη αιολικών πάρκων:

- Χερσαία αιολικά πάρκα (onshore)



Σχήμα 1-11: Χερσαίο αιολικό πάρκο (Πηγή: Μαρούδας, 2008)

- Αιολικά πάρκα κοντά στην ακτή (onshore)



Σχήμα 1-12: Αιολικό πάρκο κοντά στην ακτή (Πηγή: Μαρούδας, 2008)

- Υπεράκτια αιολικά πάρκα (offshore)



Σχήμα 1-13: Υπεράκτιο αιολικό πάρκο (Πηγή: Μαρούδας, 2008)

## ***1.2. Σκοπός της εργασίας***

Στην παρούσα διπλωματική εργασία καλούμαστε να μελετήσουμε το πρόβλημα της επιλογής του βέλτιστου τρόπου παροχής ενέργειας σε ένα διασυνδεδεμένο δίκτυο ανεμογεννητριών με αντικειμενικό σκοπό την αύξηση της συνολικής απόδοσης, δηλαδή της συνολικής ενέργειας που παράγει το δίκτυο. Ως περίπτωση εφαρμογής επιλέχθηκε η ευρύτερη περιοχή της Ιταλίας.

Η πρωτοτυπία της βρίσκεται στο ότι θα χρησιμοποιηθούν πέντε διαφορετικοί νοήμονες αλγόριθμοι, οι οποίοι θα συγκριθούν αρχικά για τα αποτελέσματα που δίνουν, άρα και θα βρεθεί ο πιο αποτελεσματικός όσον αφορά το χώρο εφαρμογής που εξετάζεται. Επιπλέον, θα γίνει μια πρώτη απόπειρα να δημιουργηθεί ένα υβριδικό σχήμα (hybrid scheme) που θα περιέχει τουλάχιστον δύο εξ αυτών των αλγορίθμων, με σκοπό την εκμετάλλευση των πλεονεκτημάτων τους και τελικά την διερεύνηση αν συνδυασμός αυτών δίνει καλύτερα αποτελέσματα από μεμονωμένα τρεξίματα (simulations) κάθε ενός εκ των αλγορίθμων που παρουσιάζονται παρακάτω.

Ο στόχος της εργασίας είναι να εστιάσει κανείς στο αλγοριθμικό κομμάτι του προβλήματος, δημιουργώντας εκ του μηδενός τους αλγόριθμους που θα αναφερθούν. Μας ενδιαφέρει κυρίως να συγκρίνουμε τους προτεινόμενους αλγόριθμους, καθώς και τον υβριδικό, ώστε να καλυφθεί το βιβλιογραφικό κενό που υπάρχει στην επίλυση τέτοιου είδους προβλημάτων από νοήμονα συστήματα.

## ***1.3. Δομή της εργασίας***

Η εργασία αποτελείται από 6 κεφάλαια.

Στο Κεφάλαιο 1 περιέχεται μια σύντομη περιγραφή του προβλήματος, καθώς και η συνεισφορά της εν λόγω διπλωματικής στη διερεύνηση λύσης στο πρόβλημα.

Στο Κεφάλαιο 2 παρουσιάζεται μια βιβλιογραφική επισκόπηση στην έρευνα που έχει γίνει μέχρι τώρα σε θέματα τεχνητής νοημοσύνης και ειδικότερα στα προβλήματα πολλαπλών μεταβλητών (ή αλλιώς προβλήματα χαρτοφυλακίου).

Στο Κεφάλαιο 3 παρουσιάζονται οι μεθοδολογίες των αλγορίθμων και η μαθηματική λογική που ακολουθεί ο καθένας από αυτούς.

Στο Κεφάλαιο 4 αναλύεται το πρόβλημα κατανομής ενέργειας και ορίζονται οι συναρτήσεις που θα χρησιμοποιηθούν στην αλγοριθμική μοντελοποίησή του.

Στο Κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα κάθε αλγόριθμου και γίνεται σύγκριση μεταξύ αυτών. Επιπρόσθετα, ο αλγόριθμος με τα καλύτερα αποτελέσματα συγκρίνεται με το προτεινόμενο υβριδικό σχήμα.

Στο Κεφάλαιο 6 καταγράφονται τα συμπεράσματα που απορρέουν από την εργασία.

## 2. Βιβλιογραφική επισκόπηση

### 2.1. Γενικά

Στο παρελθόν έχουν γίνει παρόμοιες εργασίες γενικότερα σε αυτόν το χώρο εφαρμογής, τις ανανεώσιμες πηγές ενέργειας. Αρκετές από αυτές τις εργασίες καλούνται να δώσουν λύση στο πρόβλημα της κατανομής φορτίου της ενέργειας από ανεμογεννήτριες (Kusiak A. and Zheng H., 2009) και (Gonzales et. al., 2010). Επίσης υπάρχουν ακόμη εργασίες που ασχολούνται με προβλήματα από τον ίδιο χώρο εφαρμογής, σχετικά με υδροθερμική ενέργεια (Basu M., 2004), με φωτοβολταϊκά (Κουτρουλής Ε., Κολοκότσα Δ., Ποτηράκης Α., Καλαϊτζάκης Κ., 2005) και συνδυασμένης παραγωγής θερμότητας (Vasebi A., Fesanghary M., Bathae S.M.T., 2007). Όλες χρησιμοποιούν κάποια εκδοχή νοημών αλγόριθμων όπως ο Simulated Annealing ή ο Harmony Search.

Στην εργασία τους, οι Grothe και Schnieders (Grothe O. and Schieders J. 2011) παρουσίασαν τη σχέση μεταξύ ανέμου και κατανομής αιολικής ενέργειας, βασιζόμενοι σε ένα Copula model, δηλαδή ένα στατιστικό μοντέλο που περιγράφει μη γραμμικές σχέσεις μεταξύ ενός διανύσματος μεταβλητών. Σαν δεδομένα χρησιμοποίησαν μετρήσεις ταχύτητας ανέμου σε διάφορες περιοχές της Γερμανίας. Επιπρόσθετα, εργασίες όπως του (Cassola et al, 2007) έχουν ασχοληθεί με το ίδιο πρόβλημα βελτιστοποίησης αλλά με διαφορετικά δεδομένα (η συγκεκριμένη της Κορσικής).

Ωστόσο, μια σχετική εργασία που μελετάει το ίδιο πρόβλημα με την παρούσα είναι του Θωμαΐδη (2012), η οποία χρησιμοποιεί σαν δεδομένα τις μετρήσεις ημερήσιας απόδοσης των ανεμογεννητριών σε διάφορα μέρη της Ολλανδίας.

Συνολικά, σχεδόν καμία εργασία δεν έχει στοχεύσει στη συγκριτική μελέτη μεταξύ των πέντε αλγορίθμων που χρησιμοποιούνται εδώ, ούτε έχει γίνει κάποια απόπειρα για υβριδικό σχήμα μεταξύ αυτών. Κατ' επέκταση, τα αποτελέσματα αυτής της εργασίας μπορούν να χρησιμοποιηθούν για περαιτέρω έρευνα σχετικά με τη μορφή ή το είδος των νοημών αλγορίθμων που θα μπορούσαν να καταστήσουν πιο αποδοτική την τελική λύση που θα επιλέξει κάποιος λήπτης απόφασης.

## **2.2. Βασικές Έννοιες**

Μέγιστη ισχύς (capacity) μιας μονάδας είναι η μέγιστη δυνατή τιμή ενέργειας που μπορεί να δίνει σαν απόδοση η μονάδα (π.χ. ανεμογεννήτρια).

Συνολική διαθέσιμη δυναμικότητα ονομάζεται το σύνολο της μέγιστης ισχύος που έχει ένα σύνολο μονάδων (π.χ. αιολικό πάρκο).

Ονομαστική ισχύς (ή δυναμικότητα) είναι η ενέργεια που παράγει μια μονάδα για μια δεδομένη χρονική στιγμή και σε δεδομένες συνθήκες.

Συνολική ονομαστική ισχύς είναι η συνολική ενέργεια που παράγει ένα σύνολο μονάδων (π.χ. αιολικό πάρκο) για μια δεδομένη χρονική διάρκεια και σε δεδομένες συνθήκες.

Παραγώμενο δυναμικό (generating capacity) ονομάζεται η τιμή της ενέργειας που παράγεται ανά εγκατεστημένο KW.

## **2.3. Υποθέσεις/Παραδοχές του προβλήματος**

Στο πρόβλημα έχουμε σαν δεδομένα την ενέργεια που παίρνουμε ανά εγκατεστημένο MW, η οποία προέρχεται από το άθροισμα των πολλαπλασιασμών της πραγματικής ενέργειας κάθε μονάδας (ανεμογεννήτριας) επί την ενεργειακή μονάδα (ανεμογεννήτρια). Για να αποφύγουμε την πολυπλοκότητα που θα γεννήσει η λήψη τιμών ενέργειας από όλες τις ανεμογεννήτριες, καθώς η κάθε μια είναι διαφορετικού τύπου και άρα ισχύος, δεχόμαστε στη συνολική ενέργεια ως έχει χωρίς να την αναλύουμε επί μέρους.

Ένα ακόμη γεγονός που δε θεωρούμε παράμετρο του προβλήματος εδώ είναι η ζήτηση που θα υπάρχει ανά τόπους. Δηλαδή, ο διαχειριστής του δικτύου θα πρέπει κάθε φορά να λαμβάνει υπ' όψιν του τη ζήτηση κάθε περιοχής, ώστε να διαμορφώσει το μίγμα με τέτοιο τρόπο που όλες

οι περιοχές να λαμβάνουν την ελάχιστη ενέργεια που χρειάζονται. Για λόγους απλότητας, αποφεύγουμε και αυτό να μπει στους περιορισμούς του προβλήματος.

Επιπρόσθετα, ένα ακόμη σημείο που θα ανέβαζε το βαθμό δυσκολίας θα ήταν να κάνουμε πιο δυναμικό το μοντέλο μας και να λαμβάνουμε τιμές ταχύτητας ανέμου από μετρήσεις σε καθημερινή βάση και να γίνονται οι υπολογισμοί βάση της καμπύλης ισχύος που θα δημιουργούταν κάθε φορά. Επειδή η παρούσα εργασία στοχεύει κυρίως στην παρουσίαση ενός αλγοριθμικού μοντέλου με εισαγωγή τεχνητής νοημοσύνης για την επίλυση του προβλήματος, ο πιο δυναμικός χαρακτήρας του αλγορίθμου αφέθηκε για περαιτέρω έρευνα.

### 3. Αλγόριθμοι βελτιστοποίησης και εφαρμογές

Αλγόριθμους βελτιστοποίησης ονομάζουμε κάθε επαναληπτική διαδικασία, η οποία προγραμματίζεται βασισμένη σε κάποιο φαινόμενο που παρατηρείται στη φύση και εκμεταλλευόμενη αυτό το ιδιαίτερο χαρακτηριστικό που “αντιγράφει”, έχει στόχο την εύρεση της οριακά καλύτερης λύσης (optimal best solution). Λέγοντας οριακά καλύτερη εννοείται μια λύση που προσεγγίζει κατά πολύ την καλύτερη. Αυτό συμβαίνει καθώς οι αλγόριθμοι αυτοί εντοπίζουν “περιοχές” που περιέχουν τέτοιες τιμές των μεταβλητών ενός προβλήματος που χρήζει επίλυση και ύστερα ερευνούν ποια από αυτές είναι η καλύτερη.

Το ερώτημα, όμως, που παραμένει είναι γιατί να επιλέξει κανείς τέτοιου είδους αλγόριθμους και να μην ψάξει τη λύση σε κάποιο πρόβλημα με το γνωστό τρόπο: λύνοντας μαθηματικές εξισώσεις. Εκτός του ότι πολλές φορές οι εξισώσεις έχουν τόσες πολλές μεταβλητές που τις κάνουν πολύ δύσκολο να λυθούν, επιπλέον για να βρεθεί η καλύτερη λύση όλων, θα πρέπει να βρεθούν πρώτα όλες οι πιθανές λύσεις. Κάτι που σημαίνει ότι αν έχουμε π.χ. ένα πρόβλημα με 8 μεταβλητές και 100 διαφορετικούς συνδυασμούς αυτών, ώστε να μας δώσουν κάποια λύση, για να κάνει κανείς όλους τους πιθανούς συνδυασμούς αυτών πρέπει να υπολογίσει στην ουσία η σχέση  $\frac{n!}{k!(n-k)!}$  δηλαδή το μέγεθος  $\frac{100!}{8!92!}$  που δίνει ότι θα είχαμε 186.087.894.300 πιθανούς συνδυασμούς. Είναι εύκολα κατανοητό ότι αυτοί οι υπολογισμοί θα διαρκούσαν για πολύ καιρό, κάτι που αυξάνει ακόμη περισσότερο τη δυσκολία εύρεσης της καλύτερης λύσης. Ένας αλγόριθμος βελτιστοποίησης, λοιπόν, είναι μια καλύτερη και γρηγορότερη επιλογή ώστε να προσεγγίσει κανείς κατά πολύ την επιθυμητή λύση.

#### 3.1. Ευρετικοί και μετα-ευρετικοί αλγόριθμοι

Σύμφωνα με τον T. Weise, οι αλγόριθμοι βελτιστοποίησης χωρίζονται (βάση της μεθόδου που ακολουθούν) σε ευρετικούς και μετα-ευρετικούς (Weise, 2009). Επιπλέον, οι ορισμοί αυτών των δύο κατηγοριών στην ίδια εργασία δίνονται ως εξής “Ευρετικός είναι μέρος ενός αλγορίθμου βελτιστοποίησης που χρησιμοποιεί την πληροφορία που έχει συλλεχθεί ως εκείνο το

σημείο από τον αλγόριθμο, ώστε να αποφασισθεί ποιος συνδυασμός παραμέτρων θα ελεγχθεί ή ποιος νέος συνδυασμός μπορεί να παραχθεί” και “Ένας μετα-ευρετικός είναι μια μέθοδος για την επίλυση προβλημάτων, συνδυάζει αντικειμενικές συναρτήσεις και ευρετικούς αλγόριθμους με έναν αφηρημένο και ενδεχομένως αποτελεσματικό τρόπο, συνήθως χωρίς να χρειάζεται βαθύτερη κατανόηση της δομής τους”.

Στην παρούσα εργασία θα ασχοληθούμε με τους μετα-ευρετικούς αλγόριθμους Simulated Annealing, Particle Swarm Optimization, Harmony Search, Tabu Search και Threshold Accepting.

### ***3.1.1. Simulated Annealing***

Η λέξη annealing προέρχεται από τη μεταλλουργία, όπου είναι ένας τρόπος θέρμανσης των υλικών σε μεγάλες θερμοκρασίες μαζί με ελεγχόμενη ψύξη με σκοπό την αύξηση του μεγέθους των κρυστάλλων του υλικού και μείωση των ατελειών του. Η αύξηση της θερμοκρασίας προκαλεί αποκόλληση των ατόμων του υλικού με αποτέλεσμα να ψάχνουν να επικολληθούν σε θέση με υψηλότερη ενέργεια, ενώ η ελεγχόμενη ψύξη βοηθά στην εύρεση αυτών των θέσεων ευκολότερα.

Αντίστοιχα λοιπόν, αυτή η μέθοδος λήψης αποφάσεων στοχεύει στην «εναλλαγή θέσεων» των λύσεων μιας εξίσωσης, ώστε να βρεθεί το ελάχιστο αυτής. Κάθε βήμα του αλγορίθμου της μεθόδου αυτής, αντικαθιστά μια λύση με κάποια τυχαία νέα λύση. Η νέα λύση θα απορριφθεί ή θα γίνει δεκτή σύμφωνα με μια πιθανότητα που εξαρτάται από τη διαφορά μεταξύ αντίστοιχων με τη λύση τιμών της συνάρτησης, καθώς και από μια παράμετρο που σε όλη τη διάρκεια του αλγορίθμου μειώνεται, την  $T$  (από το temperature). Όσο μειώνεται η παράμετρος  $T$ , τόσο καλύτερη επιλογή γίνεται για την αποδοχή ή την απόρριψη των τιμών, ενώ όσο είναι μεγαλύτερη, η επιλογή είναι λίγο τυχαία.

Η μέθοδος αυτή προτάθηκε πρώτη φορά από τους Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi το 1983 και πρόσθεσε ορισμένες βελτιώσεις ο Vlado Černý το 1985. Ο ψευδοκώδικας έχει ως εξής:



```

for i έως N επαναλήψεις
    1. Αρχικοποίηση συνθηκών
    2. Δημιουργία πίνακα με αρχικές τιμές των
    3. παραμέτρων/μεταβλητών
    4. Δημιουργία νέου πίνακα με νέες τιμές παραμέτρων
    5. Εφαρμογή κριτηρίου αποδοχής - Metropolis criterion
    6. Αποθήκευση ελάχιστης τιμής και αλλαγή τιμών παραμέτρων
end

```

Ο συγκεκριμένος αλγόριθμος έχει ένα πολύ καλό κριτήριο αποδοχής μιας νέας τιμής, το Metropolis criterion. Το συγκεκριμένο κριτήριο αποδοχής προήλθε από την μαρκοβιανή αλυσίδα Monte Carlo (Markov chain Monte Carlo) Metropolis-Hastings algorithm, η οποία είναι μια μέθοδος λήψης μιας ακολουθίας τυχαίων δειγμάτων από μια κατανομή πιθανοτήτων, για την οποία δεν μπορεί να γίνει άμεση δειγματοληψία. Αν, για παράδειγμα, έχουμε μια τιμή  $x_t$  και ελέγχουμε την επόμενη τιμή  $x'$  με συνάρτηση πυκνότητας πιθανότητας  $Q(x'|x_t)$ , υπολογίζουμε

μια τιμή  $a = a_1 \cdot a_2$ , όπου  $a_1 = \frac{P(x')}{P(x_t)}$  και  $a_2 = \frac{Q(x_t|x')}{Q(x'|x_t)}$ . Δηλαδή  $a_1$  ορίζουμε το λόγο της πιθανότητας

της προτεινόμενης τιμής προς της υπάρχουσας τιμής και  $a_2$  ορίζουμε τον λόγο των συναρτήσεων πυκνότητας πιθανότητας των δύο τιμών, που αυτός ο λόγος θα είναι 1 εάν η προτεινόμενη τιμή έχει συμμετρική συνάρτηση πυκνότητας πιθανότητας ως προς την υπάρχουσα τιμή. Έτσι επιλέγουμε μια νέα τιμή  $x_{t+1}$  ως εξής:

Αν  $a \geq 1$  τότε  $x_{t+1} = x'$

Αλλιώς  $x_{t+1} = x_t$  με πιθανότητα  $a$

ή  $x_t$  με πιθανότητα  $1-a$

Εδώ, ωστόσο, αυτή η πιθανότητα έχει την τιμή  $p = \exp\left[-\frac{\Delta E}{k_b T}\right]$ , όπου  $k_b$  είναι η σταθερά του

Boltzmann και  $T$  είναι η θερμοκρασία ελέγχου της διαδικασίας ανόπτισης (annealing process).

$\Delta E$  είναι η διαφορά επιπέδου ενέργειας, όπου αν  $\gamma$  μια πραγματική σταθερά, τότε  $\Delta E = \gamma \cdot \Delta f$ .

Χωρίς τον κίνδυνο της γενικότητας και για απλοποίηση, αν θεωρήσουμε  $k_b = 1$  και  $\gamma = 1$ , η

συνάρτηση πιθανότητας απλοποιείται σε  $p(\Delta f, T) = e^{-\Delta f/T}$ .

Έτσι, ξεκινώντας από υψηλές τιμές θερμοκρασίας  $T$ , η πιθανότητα  $p$  προσεγγίζει τη μονάδα, άρα ο αλγόριθμος αποδέχεται ευκολότερα τις νέες τιμές. Όσο η τιμή της θερμοκρασίας  $T$  μειώνεται, τόσο η πιθανότητα  $p$  προσεγγίζει το μηδέν και άρα ο αλγόριθμος επιλέγει μόνο τις πολύ καλύτερες νέες τιμές. Γι' αυτό η σταδιακή μείωση της θερμοκρασίας  $T$  στον αλγόριθμο, μας προσφέρει την ασφάλεια του ότι αρχικά θα ερευνηθούν περισσότερες περιοχές λύσεων και όσο εκτελεί περισσότερες επαναλήψεις ο αλγόριθμος, θα γίνεται τοπική αναζήτηση γύρω από αυτές τις τιμές με αποτέλεσμα να μπορεί να βρεθεί μια λύση κοντά στη βέλτιστη.

### 3.1.2. Particle Swarm Optimization

Ο εν λόγω αλγόριθμος παρουσιάστηκε το 1995 πρώτη φορά από τους Kennedy και Eberhart, βασιζόμενοι στον τρόπο που κινούνται τα σμήνη (πουλιών, ψαριών κλπ) στη φύση. Ο PSO γέννησε ένα νέο πεδίο έρευνας, το οποίο ονομάστηκε “νοημοσύνη σμήνους” (swarm intelligence). Στην πορεία εμφανίστηκαν πολλές διαφορετικές εκδοχές του παραπάνω αλγόριθμου, καθώς χρησιμοποιήθηκε σε όλα τα είδη προβλημάτων βελτιστοποίησης, υπολογιστικής νοημοσύνης και εφαρμογών σχεδιασμού/προγραμματισμού.

Ο αλγόριθμος ψάχνει το χώρο λύσεων της αντικειμενικής συνάρτησης, προσαρμόζοντας τις τροχιές επιμέρους παραγόντων (παράγοντες του προβλήματος), οι οποίοι ονομάζονται σωματίδια (particles), καθώς σχηματίζονται μονοπάτια (paths) από κάθε σωματίδιο μέσω διανυσμάτων θέσης, δηλαδή σημείων και φοράς που έχει κάθε στιγμή κάθε σωματίδιο. Η κίνηση κάθε σωματιδίου αποτελείται από δύο βασικές συνιστώσες: μια στοχαστική και μια ντετερμινιστική. Κάθε σωματίδιο “έλκεται” από τη θέση του ολικού βέλτιστου  $g^*$  και από τη δική του βέλτιστη θέση  $x_i^*$  από όσες έχει καταλάβει μέχρι εκείνη τη στιγμή στον αλγόριθμο. Ωστόσο, το σωματίδιο έχει και την τάση να κινείται τυχαία (λόγω της στοχαστικής συνιστώσας).

Υποθέτοντας ότι  $x_i$  και  $v_i$  είναι τα διανύσματα θέσης και ταχύτητας για το σωματίδιο  $i$ , το νέο διάνυσμα ταχύτητας θα είναι:

$$v_i^{t+1} = v_i^t + \alpha \varepsilon_1 \odot [g^* - x_i^t] + \beta \varepsilon_2 \odot [x_i^* - x_i^t]$$

όπου τα  $\varepsilon_1$  και  $\varepsilon_2$  είναι δύο τυχαία διανύσματα που παίρνουν τιμές μεταξύ 0 και 1. Το εσωτερικό γινόμενο δύο πινάκων  $u \odot v$  δίνεται από το  $[u \odot v]_{ij} = u_{ij}v_{ij}$ . Οι παράμετροι  $\alpha$  και  $\beta$  λέγονται παράμετροι μάθησης ή σταθερές επιτάχυνσης και συνήθως ισχύει  $\alpha \approx \beta \approx 2$ .

Οι αρχικές θέσεις των σωματιδίων πρέπει να κατανέμονται σχετικά ομοιόμορφα στο πεδίο των λύσεων, ώστε να ελεγχθεί ο μέγιστος δυνατός χώρος, κάτι που είναι σημαντικό κυρίως σε πολυδιάστατα προβλήματα. Η αρχική ταχύτητα κάθε σωματιδίου μπορεί να θεωρηθεί μηδενική, άρα  $v_i^{t=0} = 0$  και η νέα θέση βρίσκεται κάθε φορά από τον τύπο

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

παρόλο που το διάνυσμα ταχύτητας  $v_i$  μπορεί να πάρει οποιαδήποτε τιμή, συνήθως το ορίζουμε μεταξύ  $[0, v_{\max}]$ .

Ο ψευδοκώδικας του συγκεκριμένου αλγόριθμου είναι:

1. Τυχαίος ορισμός των σωματιδίων στο πεδίο ορισμού, ορισμός (μικρής) τυχαίας ταχύτητας (velocity)
- while** το κριτήριο τερματισμού δεν είναι αληθές
  2. Ενημέρωση του gbest για το σμήνος (swarm) και του pbest για το σωματίδιο
  - if** εφαρμόζουμε το lbest (local best) **then**
    - Εύρεση της υπο-ομάδας που ανήκει το σωματίδιο
    - Ενημέρωση του lbest για όλα τα σωματίδια που ανήκουν στην υπο-ομάδα
  3. Υπολογισμός νέας ταχύτητας του σωματιδίου
  4. Κίνηση του σωματιδίου σε νέα θέση
- end**

### 3.1.3. Harmony Search

Το 2001 σχεδιάστηκε από τον Z. W. Geem ένας νέος ευρετικός αλγόριθμος, ο οποίος στηρίχθηκε στο πρότυπο της διαδικασίας που ακολουθεί ένας μουσικός όταν αυτοσχεδιάζει. κατά τη συγκεκριμένη διαδικασία, έχει τρεις επιλογές: α) να παίζει κάποια γνωστή μελωδία καλώντας τη στη μνήμη του, β) να παίζει μια μελωδία παρόμοια με κάποια που είναι γνωστή και γ) να συνθέσει νέα τυχαία μελωδία. Οπότε, μοντελοποιώντας αυτή τη λογική, έχουμε τρεις

αντίστοιχες συνιστώσες: χρήση της αρμονίας από τη μνήμη, ρύθμιση μιας υπαρκτής αρμονίας με στόχο τη δημιουργία μιας διαφορετικής και τυχαίο τρόπο σύνθεσης.

Αρχικά, η χρήση μιας αρμονίας που υπάρχει εξασφαλίζει ότι μια καλή αρμονία (καλός συνδυασμός από νότες) θα μεταφερθεί στη νέα σύνθεση που θα δημιουργηθεί. Για να εφαρμόσουμε αυτή τη συνιστώσα της μνήμης, θέτουμε στον αλγόριθμο μια παράμετρο  $r_{\text{accept}} \in [0,1]$ , η οποία καλείται μνήμη αρμονίας (harmony memory) και μας βοηθά να αποδεχόμαστε ή να απορρίπτουμε αρμονίες (ή καλύτερα συνδυασμούς λύσεων). Όσο πιο μικρό είναι το  $r_{\text{accept}}$ , τόσο πιο δύσκολα αποδέχεται ο αλγόριθμος νέες αρμονίες και άρα επιλέγει μόνο τις καλύτερες, ενώ όσο πιο μεγάλη τιμή έχει η παράμετρος, τόσο σχεδόν όλες οι αρμονίες επιλέγονται και δεν γίνεται περαιτέρω αναζήτηση για καλές λύσεις. Έχει βρεθεί ότι ο αλγόριθμος εμφανίζει καλύτερη επίδοση για τιμές  $r_{\text{accept}} = 0.7 \sim 0.95$ .

Περνώντας στη δεύτερη συνιστώσα, αυτή της ρύθμισης μιας υπάρχουσας μελωδίας με σκοπό τη δημιουργία μιας νέας, χρησιμοποιούμε έναν γραμμικό (linear) τύπο ρύθμισης. Θεωρώντας το  $x_{\text{old}}$  μια υπάρχουσα λύση/μελωδία, η ρυθμισμένη λύση/μελωδία θα είναι:

$$x_{\text{new}} = x_{\text{old}} + b_p(2\varepsilon - 1),$$

όπου  $\varepsilon$  είναι ένας τυχαίος αριθμός που παίρνουμε από την ομοιόμορφη κατανομή  $[0,1]$ . Το  $b_p$  είναι ένα εύρος που ελέγχει τη ρύθμιση, ώστε να είναι μεταξύ ενός εύρους τιμών. Η συγκεκριμένη συνιστώσα είναι ανάλογη του φορέα μετάλλαξης (mutation operator) του γενετικού αλγόριθμου. Μπορούμε να εφαρμόσουμε ένα ποσοστό ρύθμισης (pitch-adjusting rate)  $r_{\text{pa}}$  για να ελέγξουμε το ποσοστό της ρύθμισης. Όσο μικρότερο αυτό το ποσοστό, τόσο μικρότερη η πιθανότητα αλλαγής από την αρχική λύση. Αν είναι πολύ υψηλό, τότε ο αλγόριθμος μπορεί να μην συγκλίνει σχεδόν καθόλου σε μια ικανοποιητική λύση/αρμονία. Γι' αυτό και συνήθως παίρνει τιμές  $r_{\text{pa}} = 0.1 \sim 0.5$ .

Τέλος, η τρίτη συνιστώσα είναι υπεύθυνη για τη διαφοροποίηση των λύσεων που θα ελεγχθούν. Η συνιστώσα της ρύθμισης, παρόλο που διαφοροποιεί τις λύσεις, δεν κάνει τοπική

αναζήτηση γύρω από αυτές, κάτι που προσπαθούμε να πετύχουμε με τη συνιστώσα της τυχαιότητας. Έτσι ο αλγόριθμος ελέγχει μια μεγαλύτερη περιοχή λύσεων με αποτέλεσμα να αυξάνεται η πιθανότητα να προσεγγίσουμε περισσότερο το ολικό βέλτιστο. Ο Harmony Search έχει χρησιμοποιηθεί σε πολλά πεδία εφαρμογής, όπως η βελτιστοποίηση συναρτήσεων, αλλά και προβλήματα εξοικονόμησης ενέργειας, όπως αυτό που παρουσιάζεται σε αυτή τη διπλωματική.

### 3.1.4. Tabu Search

Πρόκειται για μέθοδο τοπικής αναζήτησης λύσεων, η οποία προτάθηκε πρώτη φορά από τον Glover το 1986. Η κεντρική ιδέα της μεθόδου Tabu Search είναι ότι απαγορεύει να ερευνηθούν λύσεις που έχουν ήδη ερευνηθεί. Οι λύσεις που ερευνώνται για πρώτη φορά αποθηκεύονται σε μία λίστα που ονομάζεται *tabulist* (“απαγορευμένη λίστα”). Σε κάθε βήμα της διαδικασίας η μέθοδος ερευνά τοπικά μία διαφορετική γειτονιά λύσεων και επαναληπτικά προχωρεί από μία λύση  $x$  σε μία λύση  $x'$  έως ότου ικανοποιηθεί κάποιο κριτήριο τερματισμού. Η γενική ιδέα του αποτυπώνεται στον ψευδοκώδικα:

1. Επιλογή ενός  $x$  που ανήκει στο πεδίο τιμών
2. Εύρεση του καλύτερου γείτονα (*best neighbor*)  
**if** το διάνυσμα που προκύπτει από τον καλύτερο γείτονα  
είναι μηδενικό **then**  
    Άλλαξε την καλύτερη λύση για τη συγκεκριμένη γειτονιά  
    λύσεων  
**end**
3. Ενημέρωσε την *tabu list*
4. Επέστρεψε στο βήμα 1

Ο Tabu Search, λοιπόν, έχει ένα σημαντικό πλεονέκτημα έναντι άλλων, όσων αφορά το “ψάξιμο” πολλών πιθανών λύσεων. Κατ’ επέκταση, αυτό τον καθιστά τον πρώτο υποψήφιο για υβριδικά σχήματα. Δεδομένου ότι δημιουργεί μια λίστα με τις λύσεις που ερευνήθηκαν, θα μπορούσε να αποτρέπει τον άλλο αλγόριθμο, που θα συμμετέχει στο υβριδικό σχήμα, να επαναλάβει την αναζήτησή του στην ίδια περιοχή λύσεων. Όντας ένας από τους πρώτους αλγόριθμους με τεχνητή νοημοσύνη που κατασκευάστηκαν, η λογική του δε στηρίζεται σε κάποιο μαθηματικό μοντέλο. Αυτό τον κάνει εξαιρετικά απλό, παρόλο που η χρησιμότητά του είναι τεράστια, ειδικά σε προβλήματα με τεράστιους χώρους λύσεων.

### 3.1.5. *Threshold Accepting*

Ο Threshold Accepting στηρίχτηκε στη λογική του Simulated Annealing και παρουσιάστηκε πρώτη φορά το 1985 από τον Barr και μια ομάδα ερευνητών (Barr and others 1985). Ακολουθώντας τη λογική των hill climbing αλγόριθμων, όπως ο simulated annealing και ο tabu search, δηλαδή αλγόριθμων που με κάποιο τρόπο ερευνούν για καλύτερες λύσεις κοντά στα τοπικά ακρότατα που βρίσκουν και έτσι “ανεβαίνουν” όλο και περισσότερο (στα προβλήματα μεγιστοποίησης προφανώς), έτσι και ο threshold accepting ελέγχει αν κάποια από τις γειτονικές λύσεις είναι καλύτερη. Η λογική του φαίνεται στον ψευδοκώδικα:

```
1. Αρχικοποίηση τιμών και ορίων (thresholds)
while δεν συναντώνται τα όρια (thresholds)
    2. Επιλογή μιας καλής (optimal) τιμής τυχαία
    3. Εύρεση τιμής διανυσμάτων (παλαιάς και νέας λύσης)
       if η νέα καλή τιμή είναι καλύτερη της παλαιότερης
       then
           Άλλαξε την καλύτερη λύση
       end
    4. Επιστροφή στο βήμα 2
end
```

### 3.2. *Υβριδικά σχήματα*

Υβριδικό σχήμα (Hybrid scheme) καλείται κάθε πιθανός συνδυασμός δύο ή περισσότερων αλγόριθμων βελτιστοποίησης με σκοπό την αξιοποίηση του “δυνατού” σημείου του καθένα και άρα καλύτερη επίδοση, δηλαδή προσέγγιση κατά πολύ της καλύτερης λύσης.

Στην παρούσα εργασία επιλέχθηκε να μελετηθεί το υβριδικό σχήμα του Simulated Annealing και του Harmony Search. Επιλέχθηκε βάση των αποτελεσμάτων που έδωσαν οι δύο αλγόριθμοι ξεχωριστά, τα οποία ήταν εκ των υψηλότερων τιμών μεταξύ των πέντε αλγορίθμων που διερευνήθηκαν και αναλύθηκαν παραπάνω. Μάλιστα ο Simulated Annealing και για τα τρία κριτήρια είχε μικρή διαφορά από την καλύτερη λύση. Το υβριδικό σχήμα περιέχει τη λογική του Harmony Search και απλώς το κριτήριο αποδοχής προέρχεται από τον Simulated Annealing.

## 4. Πεδίο εφαρμογής: το πρόβλημα της κατανομής ενέργειας

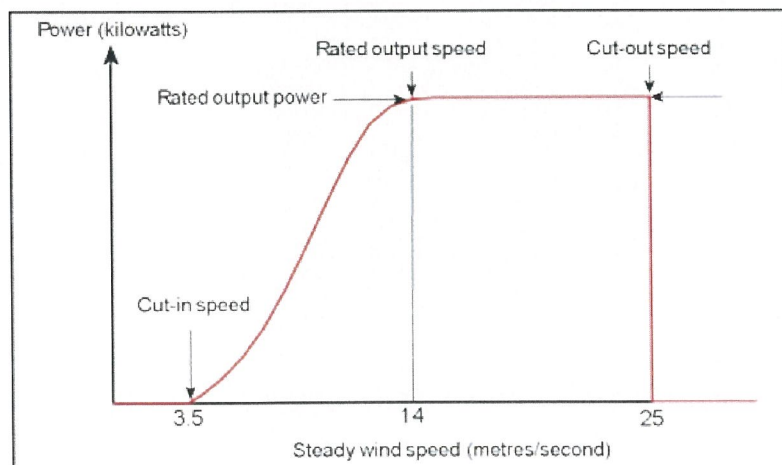
### 4.1. Το πρόβλημα του ενεργειακού χαρτοφυλακίου

Όπως θα είχαμε τις μετοχές ( $x_i$ ) και την απόδοσή τους σε ένα οικονομικό χαρτοφυλάκιο για τη στιγμή που ερευνούμε το βέλτιστο χαρτοφυλάκιο, έτσι και στο ενεργειακό χαρτοφυλάκιο ορίζουμε  $N_i$  τις ενεργειακές μονάδες (ανεμογεννήτρια εδώ) και  $E_i$  την πραγματική ενέργεια που παράγει η μονάδα  $i$  τη χρονική στιγμή  $t$ . Έτσι δημιουργούμε το άθροισμα

$$E_{p,t} = N_1 E_{p_1,t} + N_2 E_{p_2,t} + \dots + N_m E_{p_m,t} \quad (1)$$

το οποίο μας δίνει την ενέργεια που αποδίδει το συνολικό μείγμα (ολόκληρο το αιολικό πάρκο δηλαδή, αποτελούμενο από  $m$  μονάδες) τη χρονική στιγμή  $t$ .

Θεωρώντας  $C_i$  τη δυναμικότητα (μέγιστη ισχύς) των μονάδων  $i$ , δηλαδή τη μέγιστη τιμή που μπορούν να προσφέρει κάθε ανεμογεννήτρια, η οποία βρίσκεται συνήθως στο διάγραμμα ισχύος, όπως φαίνεται στην παρακάτω εικόνα, καταλήγουμε στη σχέση  $0 \leq E_p \leq C_p$ , η οποία περιορίζει τις τιμές της συνολικής ενέργειας, καθώς μπορεί να είναι από μηδέν έως και τη μέγιστη συνολική δυναμικότητα του μείγματος.



Σχήμα 4-1: Ενεργειακή απόδοση ανεμογεννήτριας για δεδομένες τιμές ανέμου (Πηγή: [www.wind-power-program.com](http://www.wind-power-program.com))

Αν διαιρέσουμε τώρα κάθε μέλος της (1) με τη συνολική διαθέσιμη δυναμικότητα (total available capacity), δηλαδή  $\sum_1^m N_j C_j$ , δημιουργούμε το πολυώνυμο:

$$\frac{N_1 E_{p_1 t}}{\sum_1^m N_j C_j} + \frac{N_2 E_{p_2 t}}{\sum_1^m N_j C_j} + \dots + \frac{N_m E_{p_m t}}{\sum_1^m N_j C_j} = CF_{pt}$$

όπου  $CF_{pt}$  είναι η ενέργεια που παίρνουμε ανά εγκατεστημένο KW (generating capacity). Άρα από τη σχέση που έχει δημιουργηθεί, αν πολλαπλασιάσουμε και διαιρέσουμε τον αριθμητή με τη δυναμικότητα κάθε μονάδας, παίρνουμε:

$$CF_{pt} = \frac{\sum_1^m N_i E_{i t}}{\sum_1^m N_j C_j} = \frac{\sum_1^m N_i C_i [E_{i t}/C_i]}{\sum_1^m N_j C_j} = \frac{\sum_1^m w_i \cdot CF_{i t}}{\sum_1^m N_j C_j}$$

όπου το  $w_i$  είναι το ποσοστό της συνολικής ισχύος που εγκαθίσταται στην τοποθεσία  $i$ , που βγαίνει από τον αριθμό των ενεργειακών μονάδων επί τη δυναμικότητα κάθε μονάδας. Ο λόγος της πραγματικής ενέργειας που παράγεται από τη μονάδα  $i$  τη χρονική στιγμή  $t$  προς τη δυναμικότητα κάθε μονάδας, δίνει το capacity factor κάθε μονάδας, δηλαδή την ενέργεια που παίρνουμε ανά εγκατεστημένο KW για κάθε μονάδα. Άρα καταλήγουμε στη σχέση

$$CF_{pt} = \sum_1^m \frac{w_i \cdot CF_{i t}}{N_j C_j} \quad (2)$$

η οποία μετριέται σε μονάδες  $\left[\frac{\text{KWh}}{\text{KWp}}\right]$ , δηλαδή μέση ενέργεια που παράγεται την περίοδο  $t$  προς μέγιστη ισχύ. Οπότε, μιας και στην εργασία στόχος είναι να βρεθούν τα κατάλληλα βάρη ( $w_i$ ) που θα συνθέτουν το ενεργειακό χαρτοφυλάκιο, τα δεδομένα που χρησιμοποιούνται είναι χρονολογικές σειρές από τους λόγους των παραπάνω μεγεθών.

Τέλος, ένας περιορισμός είναι η χρησιμοποίηση όλης της εγκατεστημένης ισχύος που είναι διαθέσιμη, δηλαδή το άθροισμα της στάθμισης του χαρτοφυλακίου πρέπει να ισούται με ένα (1), άρα χρησιμοποιούνται οι αποδόσεις και των έξι (6) περιοχών. Αυτός ο περιορισμός, μαθηματικά ορίζεται από τη σχέση:

$$\sum_{i=1}^6 w_i = 1$$



## **4.2. Αντικειμενικοί στόχοι**

Στο ενεργειακό χαρτοφυλάκιο, δεν μας ενδιαφέρει να περιορίσουμε τον αριθμό των μονάδων που θα συμμετέχουν (βλ. Παραδοχές παραπάνω), οπότε ο μόνος περιορισμός που βάζουμε είναι ο περιορισμός των ορίων για τις τιμές που θα λαμβάνουν τα βάρη, όπως και στο οικονομικό χαρτοφυλάκιο.

Στο παρόν πρόβλημα θέτουμε τρεις (3) αντικειμενικούς στόχους. Πρώτον, τη μεγιστοποίηση της μέσης παραγωγικότητας, δηλαδή του λόγου της ενέργειας που παράγεται ως προς την εγκατεστημένη ισχύ. Δεύτερον, τη μεγιστοποίηση του Information Ratio, του λόγου της μέσης τιμής της παραγωγικότητας προς την τυπική απόκλιση αυτής. Τέλος, τη μεγιστοποίηση της τιμής των εκατοστημορίων του διαγράμματος σχετικών συχνοτήτων της παραγωγικότητας.

## **4.3. Δεδομένα**

Στο παρόν πρόβλημα χρησιμοποιήθηκαν δεδομένα που συλλέχθηκαν από τον ιστότοπο της TERNA, της ιταλικής εταιρείας παροχής ηλεκτρικής ενέργειας. Τα δεδομένα που συλλέχθηκαν είναι τιμές ενέργειας (σε MWh) που αποδίδει συνολικά κάθε μια από τις 6 περιοχές της Ιταλίας. Οι τιμές σχηματίζουν χρονολογικές σειρές με ωριαίες τιμές για κάθε μήνα, κατά τα έτη 2011, 2012 και 2013. Οπότε συνολικά το διάστημα που εξετάζουμε είναι από 1/1/2011 έως 22/10/2013.

Τα δεδομένα επιλέχθηκαν λόγω του χωρισμού τους σε επιμέρους περιοχές, κάτι που δίνει νόημα στην στάθμιση (weights) που χρησιμοποιούμε στη μέθοδό μας.

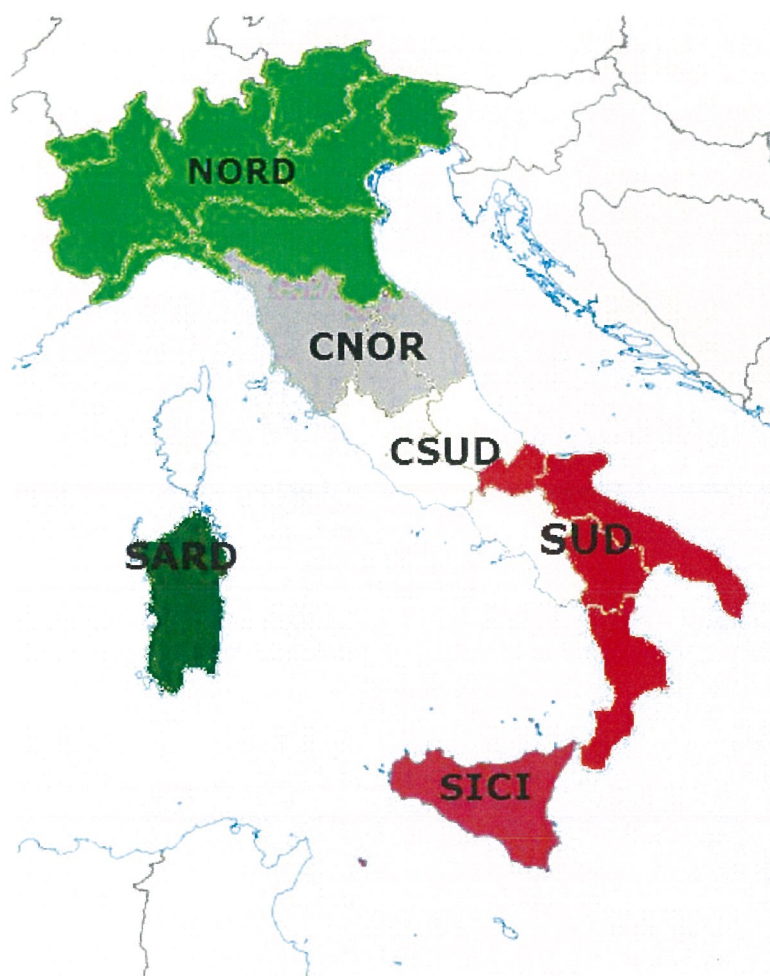
Η τιμή της συνολικής διαθέσιμης ισχύος (σε MW) για κάθε περιοχή, σύμφωνα με το Energy System Scenarios Research Group, RSE (Ricerca sul Sistema Energetico S.p.A.), δίνεται στον παρακάτω πίνακα:

**Πίνακας 4-1:** Εγκατεστημένη ισχύς στην Ιταλία (ανά ζώνη) (πηγή: <http://www.terna.it/>)

	CNOR	CSUD	NORD	SARD	SICI	SUD
1/11/2010	20	816,79	22,9	659,02	1261,62	2404,84
1/12/2010	20	816,79	22,9	659,02	1319,62	2536,24
1/1/2011	20	816,79	22,9	659,02	1319,62	2612,19
1/3/2011	20	882,79	22,9	659,02	1319,62	2612,19
1/4/2011	20	917,34	22,9	659,02	1319,62	2612,19
1/5/2011	20	917,34	22,9	659,02	1319,62	2709,69
1/6/2011	20	934,34	22,9	659,02	1426,17	2709,69
1/7/2011	20	934,34	22,9	817,72	1426,17	2725,69
1/8/2011	20	934,34	22,9	817,72	1426,17	2763,19
1/9/2011	20	934,34	22,9	817,72	1490,77	2763,19
1/11/2011	20	934,34	22,9	817,72	1490,77	2786,99
1/12/2011	20	934,34	22,9	817,72	1554,77	2835,19
1/1/2012	20	934,34	32,5	817,72	1554,77	2865,94
1/5/2012	20	934,34	43,3	817,72	1554,77	2865,94
1/1/2013	20	934,34	55,6	817,72	1554,77	2990,34
1/2/2013	69,4	934,34	55,6	817,72	1554,77	2990,34

Παρατηρούμε ότι η εγκατεστημένη ισχύς αυξάνεται με την πάροδο του χρόνου, γεγονός που δικαιολογεί και την αλματώδη αύξηση στην παραγωγή, όπως αυτή φαίνεται στα διαγράμματα που έπονται.

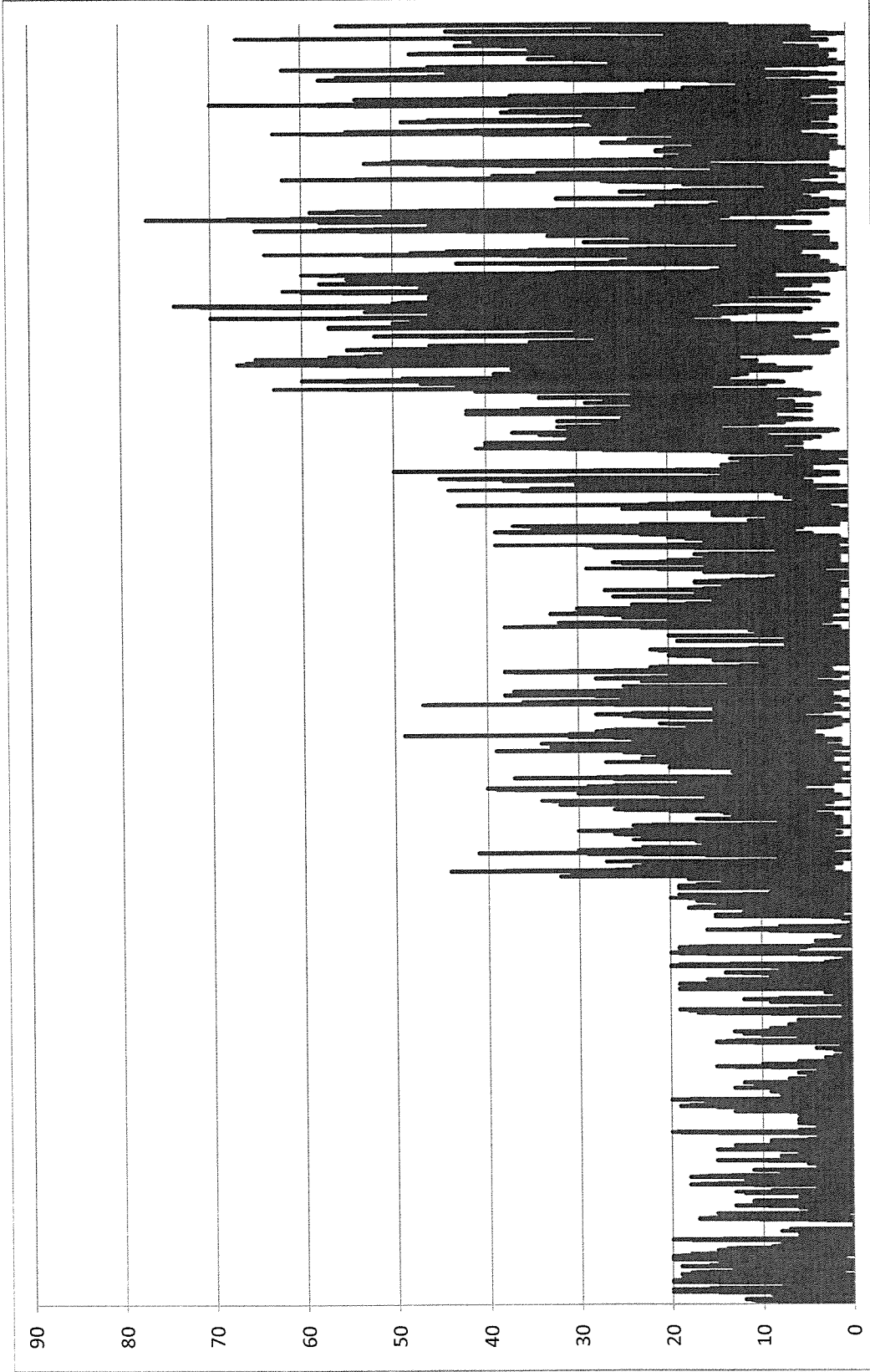
Οι περιοχές που έχει χωριστεί η Ιταλία, σύμφωνα με την TERNΑ, φαίνονται στην παρακάτω εικόνα:



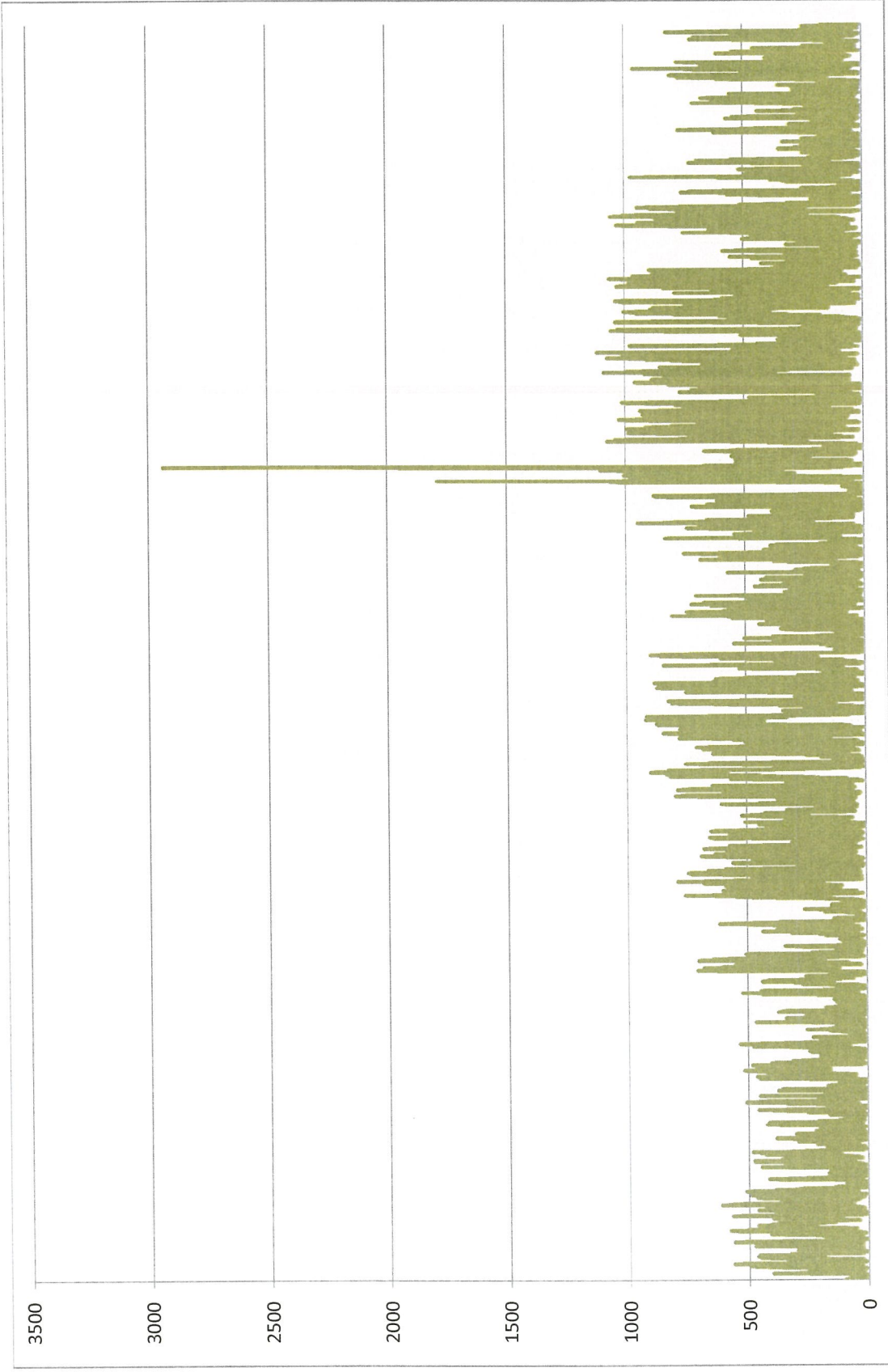
Σχήμα 4-2: Απεικόνιση περιοχών βάση Terna (πηγή: <http://www.terna.it/>)

Όλα τα δεδομένα είναι προσβάσιμα στον ιστότοπο της Terna ([http://www.terna.it/default/Home/SISTEMA\\_ELETTRICO/transparency\\_report.aspx](http://www.terna.it/default/Home/SISTEMA_ELETTRICO/transparency_report.aspx)).

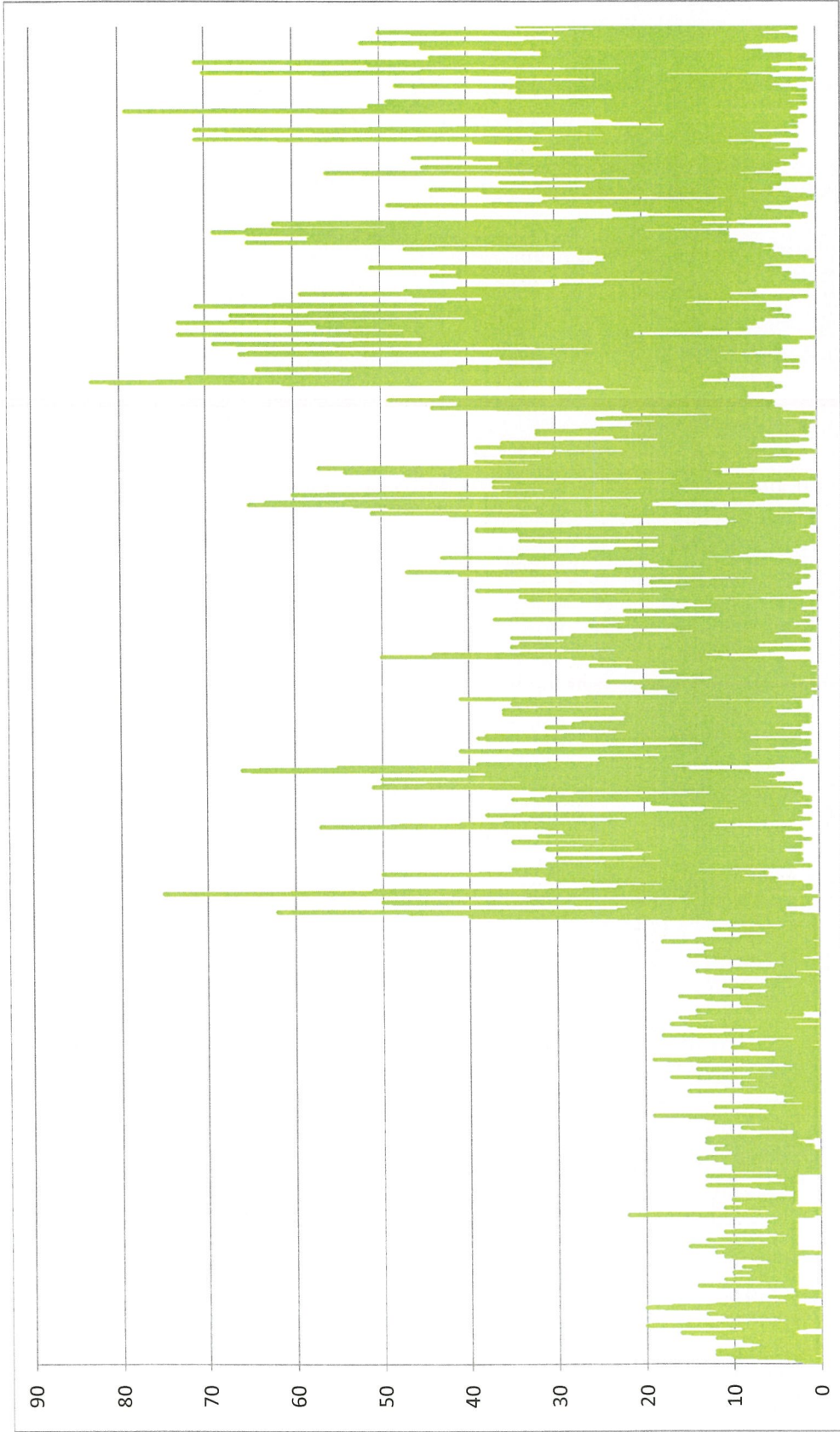
Στα παρακάτω διαγράμματα φαίνονται οι χρονοσειρές που έχουμε σαν δεδομένα. Ο οριζόντιος άξονας είναι ο χρόνος (ωριαίες τιμές) και ο κάθετος προσδιορίζεται από τις τιμές αποδιδόμενης ενέργειας (MWh) κάθε περιοχής.



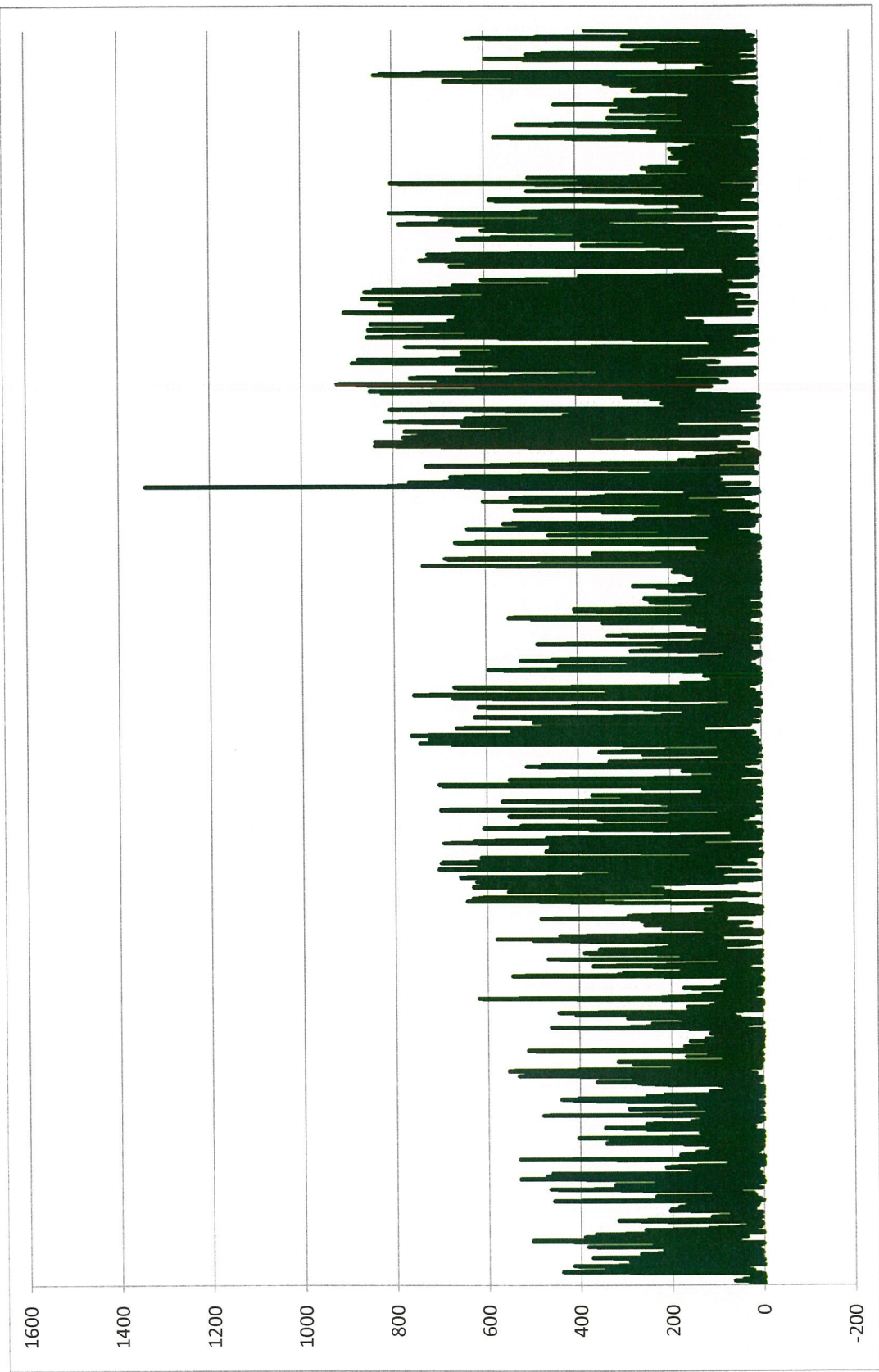
Διάγραμμα 4-1: Χρονοσειρά δεδομένων της CNOR



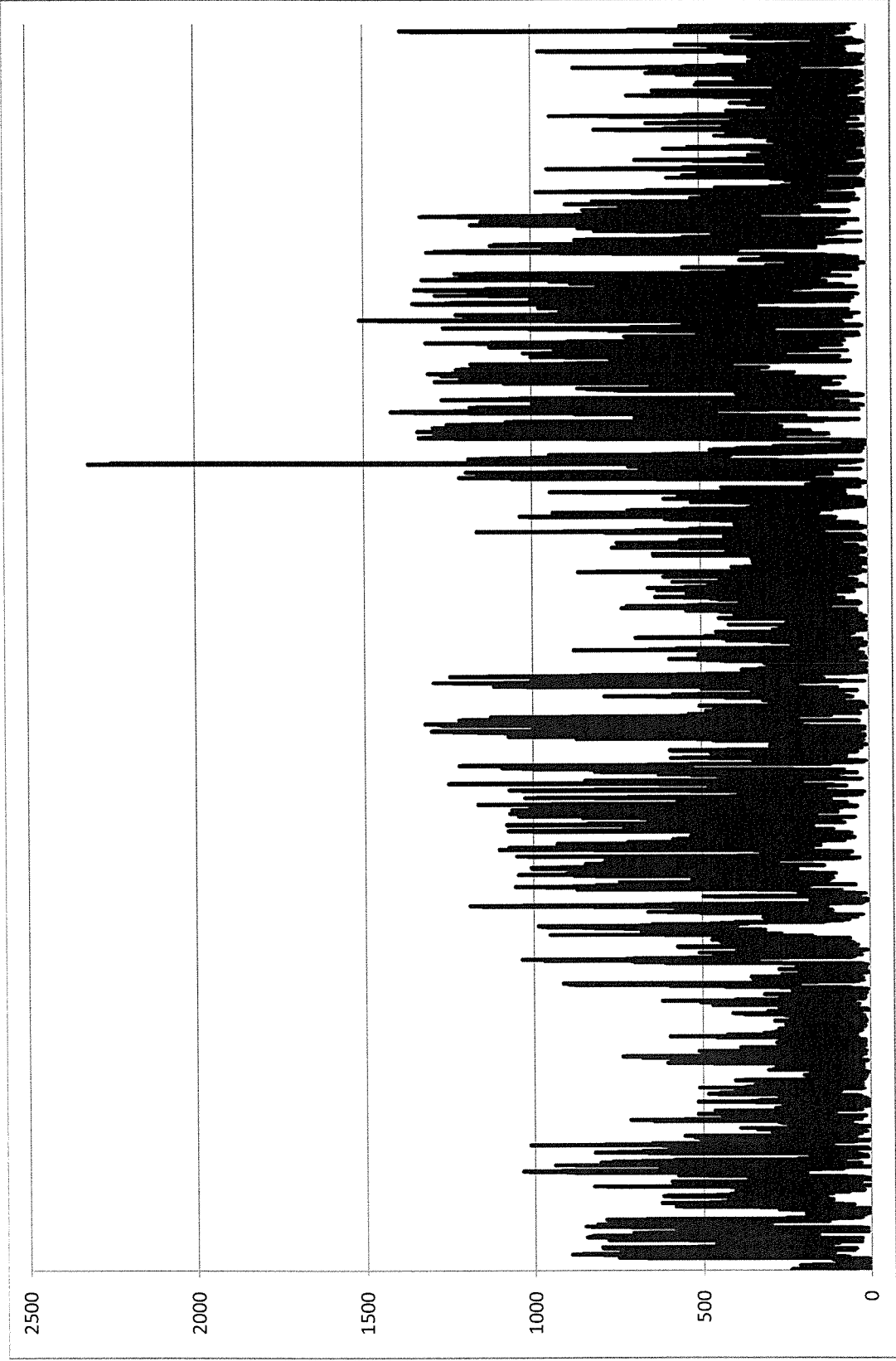
Διάγραμμα 4-2: Χρονοσειρά δεδομένων της CSUD



**Διάγραμμα 4-3:** Χρονοσειρά δεδομένων της NORD

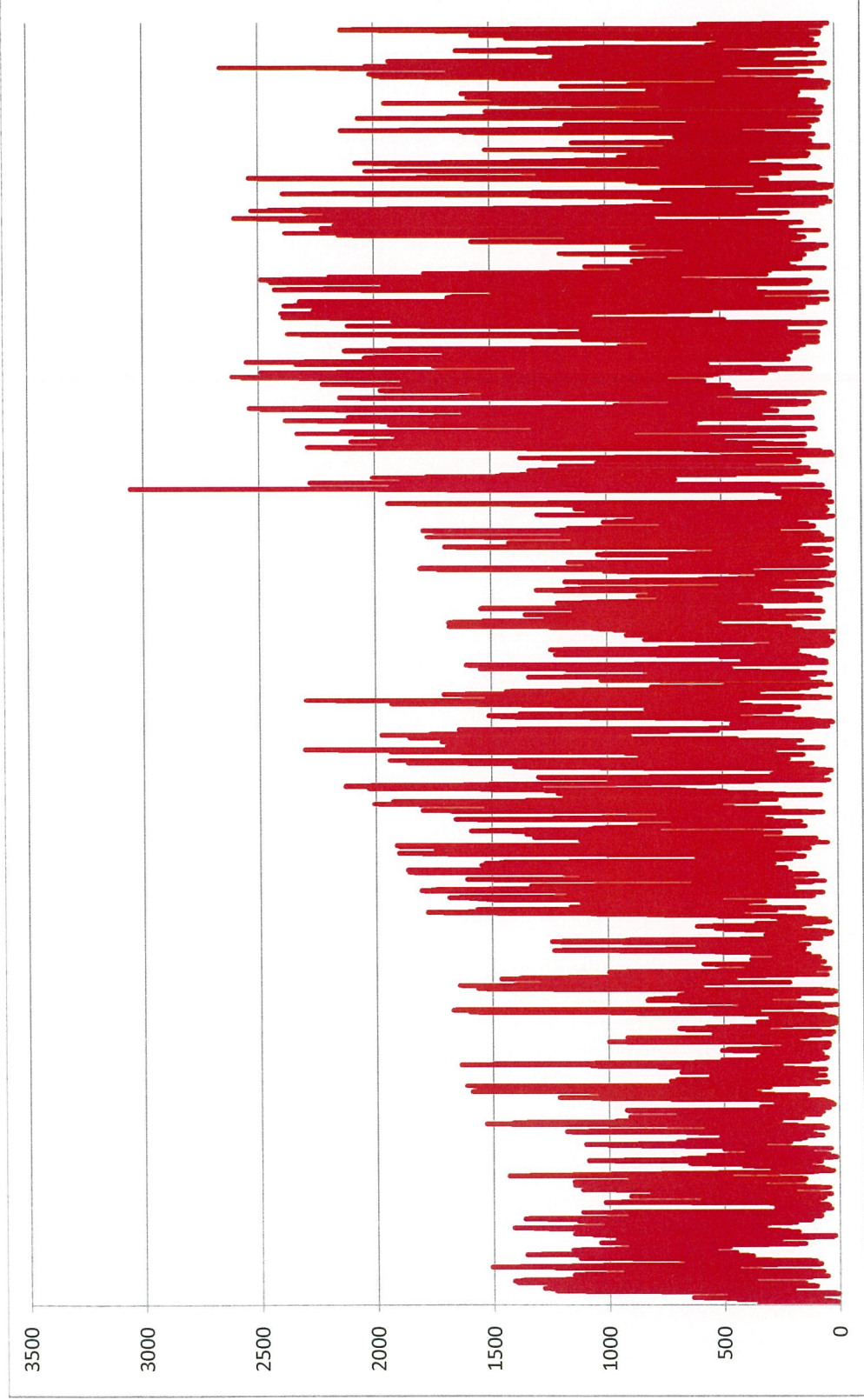


Διάγραμμα 4-4: Χρονοσειρά δεδομένων της SARD



Διάγραμμα 4-5: Χρονοσειρά δεδομένων της SIC1





Διάγραμμα 4-6: Χρονοσειρά δεδομένων της SUD

#### 4.4. Μεθοδολογία

Η προτεινόμενη μεθοδολογία βασίστηκε σε σχετικές εργασίες με προβλήματα πολλαπλών στόχων (multi-objective problems), όπως στην εργασία των Αναγνωστόπουλου Κ.Π. και Μαμάνη Γ. (2009). Ο απλοϊκός (naive) στόχος θα ήταν η μεγιστοποίηση της μέσης παραγωγικότητας, κάτι που όμως λύνεται εύκολα, λόγω του ότι μπορεί κανείς να διακρίνει πως η εξίσωση (2) είναι μια γνησίως αύξουσα συνάρτηση (με δεδομένα capacity factors κάθε περιοχής και ως άγνωστο τα ποσοστά στάθμισης). Έτσι η λύση είναι εμφανής και θα είναι κάθε φορά η περιοχή με τη μεγαλύτερη παραγωγικότητα, δηλαδή το αντίστοιχο  $w_i$  θα λαμβάνει την τιμή 1 (άρα 100% του χαρτοφυλακίου θα επενδύεται εκεί) και τα υπόλοιπα θα λαμβάνουν την τιμή 0.

Γι' αυτό προσθέτουμε ένα στατιστικό μέτρο που εκφράζει τη στάθμιση του κινδύνου ως προς την απόδοση του χαρτοφυλακίου, το Information Ratio<sup>3</sup>. Το εν λόγω μέτρο είναι ο λόγος της μέσης τιμής της παραγωγικότητας προς την τυπική απόκλιση αυτής. Ο στόχος μας είναι η μεγιστοποίηση (maximize) του λόγου αυτού, καθώς αυτό θα σημαίνει εξίσου μεγαλύτερες τιμές παραγωγικότητας και μικρή διασπορά από τη μέση τιμή.

Τέλος, κάνοντας κανείς το διάγραμμα σχετικών συχνοτήτων, μπορεί να βρει ένα κακό σενάριο (worst case scenario) και να ελέγξει ένα άλλο στατιστικό μέτρο, το εκατοστημόριο του διαγράμματος σχετικών συχνοτήτων της παραγωγικότητας. Δηλαδή να δει ελέγχοντας κάποιο (υψηλό) ποσοστό του δείγματος (των δεδομένων), την πιθανότητα να εμφανιστεί το σενάριο εκείνο και ο στόχος του είναι η ελαχιστοποίηση της πιθανότητας αυτής. Στόχος είναι η μεγιστοποίηση της τιμής κάθε εκατοστημορίου, δηλαδή να εμφανίζεται μεγαλύτερο μέρος των συχνοτήτων δεξιά από το εκατοστημόριο που ελέγχουμε (π.χ. το  $p_5$  είναι το εκατοστημόριο του 5% και άρα θέλουμε στο υπόλοιπο 95% του διαγράμματος να συγκεντρώνεται όσο το δυνατόν περισσότερες τιμές του δείγματος).

Σε κάθε αλγόριθμο, λοιπόν, έχουμε εφαρμόσει αυτούς τους 3 υπολογισμούς και παραθέτονται τα αποτελέσματα παρακάτω για συγκριτική μελέτη.

---

<sup>3</sup> [http://en.wikipedia.org/wiki/Information\\_ratio](http://en.wikipedia.org/wiki/Information_ratio)

Αρχικά φτιάχνουμε τη συνάρτηση `randfixedsum`, η οποία είναι μια συνάρτηση που δημιουργεί ένα δεδομένο πλήθος τυχαίων αριθμών μεταξύ ενός διαστήματος που ορίζουμε. Οι αριθμοί αυτοί αθροίζουν στο ένα (1). Δίνοντας, λοιπόν, εμείς σαν ορίσματα ότι θέλουμε το πλήθος να είναι έξι (6), γιατί έχουμε τις έξι ζώνες-περιοχές και ορίζοντας το διάστημα από μηδέν (0) έως ένα (1), μιας και θέλουμε να έχουμε θετικές σταθμίσεις, η συνάρτηση μας φτιάχνει διανύσματα με εξάδες από βάρη που θα πάμε να τα αντιστοιχίσουμε σε κάθε σειρά δεδομένων που έχουμε. Στα ορίσματα βάζουμε και το πλήθος των διανυσμάτων που θέλουμε να φτιάξει, οπότε για κάθε γενιά (generation) που θα τρέχει ο αλγόριθμος, να έχει διαφορετικό διάνυσμα.

Σχεδιάζοντας τους αλγόριθμους, πρέπει να λάβει κανείς υπόψη σχετικές εργασίες που παρουσιάζουν διάφορες εκδοχές των πέντε παραπάνω αλγορίθμων, κατά τις οποίες λύνονται προβλήματα χαρτοφυλακίου.

Για τον Simulated Annealing χρησιμοποιήθηκε η δημοσιευμένη εργασία των (Βασιλειάδη Β., Δούνια Γ. και Τζανέτου Α., 2012) που παρουσιάζει μια εκδοχή του εν λόγω αλγορίθμου να λύνει το πρόβλημα του χαρτοφυλακίου με σταθμίσεις σε χαρτοφυλάκια των πέντε (5) μετοχών κάθε φορά.

Για τον Harmony Search χρησιμοποιήθηκε σαν υπόδειγμα η εργασία των (Chakraborty, P., Roy, G. G., Panigrahi, B. K., Bansal, R. C., Mohapatra, A., 2011) που ασχολείται με την επίλυση ενός προβλήματος ενεργειακού χαρτοφυλακίου. Ωστόσο, σε εκείνη την εργασία υπάρχουν περισσότεροι περιορισμοί και λαμβάνεται υπόψη και η ζήτηση κάθε φορά, ώστε ο διαχειριστής του δικτύου να κάνει ένα καταμερισμό ενέργειας βάση των κατώτατων ορίων ζήτησης που πρέπει να πληροί. Το μόνο κομμάτι που αλλάζει στην παρούσα εργασία είναι στο δεύτερο μέρος του αλγορίθμου, όπου με κάποια τυχειότητα γίνεται μια αλλαγή ενός όρου από το αρχικό διάνυσμα (pitch adjustment), η πιθανότητα ρύθμισης (pitch adjustment rate) μηδενίζεται. Αυτό πρέπει να συμβεί, καθώς πρέπει αυστηρά το άθροισμα των όρων του διανύσματος (δηλαδή των  $w_i$ ) να είναι ίσο με ένα (1). Συνθήκη, η οποία θα παραβιαστεί αν αλλάξει μόνο ο ένας όρος του διανύσματος. Έτσι, για να διατηρηθεί και η λογική του Harmony Search (να μην απαλειφθεί το

βήμα αυτό ή να μην κάνουμε αλλαγή δύο όρων, κάτι που θα δυσκόλευε την ομαλή λειτουργία του αλγόριθμου), διατηρείται το βήμα και απλώς γίνεται κανονικοποίηση του διανύσματος που προκύπτει.

Μια ομολογουμένως ενδιαφέρουσα εργασία, αυτή των (Wang, L., Singh, C., 2006) αποτέλεσε παράδειγμα για το σχεδιασμό του Particle Swarm Optimization, καθώς και αυτή ασχολείται με το πρόβλημα του ενεργειακού χαρτοφυλακίου, αλλά είναι και multi-objective, κάτι που συναντά κανείς και στην παρούσα διπλωματική. Ωστόσο, μια ακόμη φορά χρειάζεται να γίνει μια τροποποίηση στην πορεία του κώδικα. Ο τρόπος που δημιουργούμε τα διανύσματα ταχύτητας, μια και είναι σημαντικά για την ανακατεύθυνση του αρχικού διανύσματος και κατ' επέκταση της προσέγγισης της λύσης, δεν μπορεί πλέον να είναι εντελώς τυχαίος. Όπως προαναφέρθηκε, για να δημιουργηθεί το νέο διάνυσμα, σε κάθε όρο του παλιού διανύσματος προστίθεται το διάνυσμα της ταχύτητας (velocity). Αυτά τα διανύσματα στην παρούσα εργασία δημιουργούνται με τυχαίο τρόπο. Για να μην παραβιαστεί η συνθήκη, η οποία λέει ότι το διάνυσμα των  $w_i$  πρέπει να αθροίζει στο ένα (1), αλλά ωστόσο πρέπει να μεταβληθεί η θέση των σωματιδίων (εδώ των weights) σύμφωνα με τη λογική του αλγόριθμου, αφού προστεθούν τα διανύσματα της ταχύτητας, προστίθεται ένα βήμα κανονικοποίησης. Σε αυτό το βήμα βγαίνει το άθροισμα των νέων  $w_i$  και διαιρείται κάθε  $w_i$  με το άθροισμα αυτό, έτσι ώστε να δημιουργηθεί ένα διάνυσμα που θα είναι ξανά μεταξύ μηδέν (0) και ένα (1).

Δυσκολότερα βρίσκει κανείς εκδοχές του Tabu Search για επίλυση προβλημάτων χαρτοφυλακίου, μιας και είναι αλγόριθμος τοπικής αναζήτησης (local search algorithm), ωστόσο μια σχετική εργασία είναι του (Schaefer, A., 2002). Η μόνη αλλαγή που γίνεται είναι ότι ο έλεγχος που εκτελεί για την καλύτερη γειτονική λύση, δεν ψάχνει να βρει διανύσματα που να είναι μηδέν (0), αλλά διανύσματα που δίνουν καλύτερες λύσεις βάσει των τριών κριτηρίων.

Εξίσου δύσκολα βρίσκει κανείς εκδοχές του Threshold Accepting Algorithm για portfolio problems, αλλά το κεφάλαιο που έγραψαν οι (Kuhn, D., Parpas, P., Rustem, B., 2008) στο βιβλίο Computational Methods in Financial Engineering, είναι μια καλή αναφορά.

Όλοι οι αλγόριθμοι παραθέτονται στο παράρτημα.

#### 4.5. Παράμετροι

Για την εύρυθμη λειτουργία των αλγόριθμων, καθώς και για να υπάρχει μια κοινή βάση σύγκρισης των αποτελεσμάτων αυτών, χρησιμοποιήθηκαν οι παρακάτω παράμετροι από κοινού. Σε όσους αλγόριθμους υπήρχαν τοπικές παράμετροι (παρουσιάζονταν μόνο στον έναν αλγόριθμο), ακολουθήθηκαν οι προτεινόμενες τιμές της βιβλιογραφίας.

Πίνακας 4-2: Παράμετροι αλγορίθμων

Παράμετρος	Επεξήγηση	Τιμή	Τοπική
<i>pop</i>	Population: πληθυσμός ή αλλιώς πλήθος τιμών	24.622	
<i>gen</i>	Generations: γενιές ή αλλιώς επαναλήψεις του αλγόριθμου	1000	
<i>k</i>	Αριθμός ζωνών	6	
<i>T</i>	Temperature: παράμετρος κριτηρίου απόρριψης	10.000	Simulated annealing
<i>r</i>	Ρυθμός μείωσης του T	0,85	Simulated annealing
<i>Probability</i>	Πιθανότητα αποδοχής νέων λύσεων	0,6	Simulated annealing
<i>w<sub>l</sub></i>	Κατώτατο όριο βαρών (weights)	0	
<i>w<sub>u</sub></i>	Ανώτατο όριο βαρών (weights)	1	
<i>a</i>	Κατώτατο όριο ταχυτήτων (velocities)	0	Particle Swarm Optimization
<i>b</i>	Ανώτατο όριο ταχυτήτων (velocities)	1	Particle Swarm Optimization
<i>G</i>	Ανώτατο όριο έλξης (attraction)	0,9	Particle Swarm Optimization
<i>L</i>	Κατώτατο όριο έλξης (attraction)	0,5	Particle Swarm Optimization
<i>M</i>	Ορμή (momentum)	0,9	Particle Swarm Optimization
<i>hms</i>	Harmony Memory Size: αριθμός ζωνών (ίσο με k)	6	Harmony Search
<i>hmcr</i>	Harmony Memory Consideration Rate: πιθανότητα διατήρησης πληροφορίας από τη μνήμη	0,8	Harmony Search
<i>par</i>	Pitch Adjusting Rate: πιθανότητα ρύθμισης κάποιας παλιάς τιμής	0,3	Harmony Search
<i>cmp</i>	Components: πλήθος τιμών (ίσο με pop)	24.622	Harmony Search
<i>u_bnd</i>	Κατώτατο όριο πιθανότητας	0	Harmony Search
<i>l_bnd</i>	Ανώτατο όριο πιθανότητας	1	Harmony Search
<i>list_size</i>	Μέγεθος της tabu list	100	Tabu Search

## 5. Αποτελέσματα

Τα αποτελέσματα που βγάξει κάθε αλγόριθμος, παρατίθενται στους παρακάτω πίνακες:

Πίνακας 5-1: Αποτελέσματα Simulated Annealing

Simulated annealing										
wbest						1st criterion	2nd criterion	3rd criterion		
$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	CFpbest	IRbest	$p_5$	$p_{10}$	$p_{15}$
0,0184	0,0432	0,4454	0,0087	0,4582	0,0262	0,9826				
0,0207	0,0256	0,1101	0,0610	0,3507	0,4318		1,3496			
0,0207	0,0256	0,1101	0,0610	0,3507	0,4318			0,0046	0,0064	0,0075

Πίνακας 5-2: Αποτελέσματα Particle Swarm Optimization

Particle Swarm Optimization										
wbest						1st criterion	2nd criterion	3rd criterion		
$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	CFpbest	IRbest	$p_5$	$p_{10}$	$p_{15}$
0,1459	0,2123	0,0936	0,2152	0,2539	0,0792	0,9833				
0,1655	0,1693	0,1624	0,1695	0,1717	0,1616		1,2814			
0,1592	0,1831	0,1404	0,1841	0,1980	0,1352			0,0036	0,0046	0,0054

Πίνακας 5-3: Αποτελέσματα Threshold Accepting

Threshold Accepting										
wbest						1st criterion	2nd criterion	3rd criterion		
$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	CFpbest	IRbest	$p_5$	$p_{10}$	$p_{15}$
0,0522	0,3093	0,0113	0,4053	0,0074	0,2145	0,8360				
0,0598	0,0009	0,0597	0,0159	0,3031	0,5606		1,3454			
0,0034	0,1852	0,0003	0,1300	0,3734	0,3077			0,0051	0,0062	0,0067

Πίνακας 5-4: Αποτελέσματα Harmony Search

Harmony Search										
wbest						1st criterion	2nd criterion	3rd criterion		
$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	CFpbest	IRbest	Pqbest		
								$p_5$	$p_{10}$	$p_{15}$
0,0892	0,7031	0,0000	0,0225	0,1050	0,0802	0,9670				
0,0242	0,0244	0,1617	0,0418	0,4108	0,3371		1,3419			
0,0290	0,1742	0,0772	0,1092	0,2650	0,3454			0,0053	0,0064	0,0075

Πίνακας 5-5: Αποτελέσματα Tabu Search

Tabu Search										
wbest						1st criterion	2nd criterion	3rd criterion		
$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	CFpbest	IRbest	Pqbest		
								$p_5$	$p_{10}$	$p_{15}$
0,3291	0,0209	0,2897	0,1370	0,2109	0,0124	0,8704				
0,0260	0,0154	0,1590	0,0555	0,2645	0,4797		1,3540			
0,0066	0,2255	0,0626	0,1765	0,1289	0,3999			0,0051	0,0065	0,0074

Παρατηρούμε, λοιπόν, ότι δεν υπάρχει κάποιος αλγόριθμος που δίνει την καλύτερη λύση όλων και για τα τρία κριτήρια, αλλά ότι ο κάθε ένας έχει διαφορετική συμπεριφορά σε κάθε κριτήριο.

Έχοντας ως αντικειμενική συνάρτηση τη μεγιστοποίηση της μέσης παραγωγικότητας, καλύτερη λύση δίνει ο Particle Swarm Optimization, με μικρή απόκλιση, ωστόσο, από τον Simulated Annealing.

Αν η αντικειμενική συνάρτηση είναι η μεγιστοποίηση του Information Ratio, τότε παρατηρείται η καλύτερη λύση από τον Tabu Search. Δεύτερος είναι και πάλι με μικρή απόκλιση ο Simulated Annealing.

Θέτοντας ως αντικειμενικό στόχο τη μεγιστοποίηση των εκατοστημορίων (percentiles), καλύτερη λύση δίνει ο Harmony Search και για ακόμη μια φορά η λύση του Simulated Annealing είναι πολύ κοντά.

Ωστόσο, αυτό που αξίζει να εξετάσει κανείς είναι αν οι τιμές των σταθμίσεων έχουν κάποια λογική βάση της εγκατεστημένης ισχύος. Τα  $w_1$  αντιστοιχούν στην περιοχή της (βόρειας) κεντρικής Ιταλίας (CNOR), όπου είναι εγκατεστημένο το 1,08% της συνολικής εγκατεστημένης ισχύος, τα  $w_2$  αντιστοιχούν στην (νότια) κεντρική Ιταλία (CSUD - 14,55% της συνολικής εγκατεστημένης ισχύος), τα  $w_3$  στην περιοχή της Βόρειας Ιταλίας (NORD - 0,87%), τα  $w_4$  στην περιοχή της Σαρδηνίας (SARD - 12,74%), τα  $w_5$  στην περιοχή της Σικελίας (SICI - 24,21%) και

τα  $w_6$  στη Νότια Ιταλία (SUD), όπου είναι συγκεντρωμένο το 46,56% της συνολικής εγκατεστημένης ισχύος. Βάση των παραπάνω θα έπρεπε να παρατηρείται μια τάση για μεγαλύτερη τιμή στάθμισης στην περιοχή της Νότιας Ιταλίας.

Ο Simulated Annealing ακολουθεί αυτή τη λογική μόνο για το δεύτερο και το τρίτο κριτήριο, ενώ στο πρώτο μοιράζει μεγαλύτερο μέρος του χαρτοφυλακίου στην περιοχή της Βόρειας Ιταλίας και στη Σαρδηνία, κάτι που μπορεί να αποδοθεί καθαρά στην παραγωγικότητα των δύο περιοχών, καθώς τείνει να είναι υψηλότερη (βλ. διαγράμματα).

Ο Particle Swarm Optimization σχεδόν ισομοιράζει τα χαρτοφυλάκια, γεγονός που εξηγεί το γιατί δίνει το χαμηλότερο Information Ratio και τα χαμηλότερα percentiles συγκριτικά με τους υπόλοιπους αλγόριθμους.

Ο Threshold Accepting επιλέγει τις περιοχές της νότιας Κεντρικής και Νότιας Ιταλίας, καθώς και τη Σαρδηνία, όσων αφορά τη μεγιστοποίηση της μέσης παραγωγικότητας. Ωστόσο, στα άλλα δύο κριτήρια ακολουθεί την τάση που αναφέρθηκε πριν ότι είναι η λογικότερη βάση της εγκατεστημένης ισχύος.

Ο Harmony Search δίνει ένα πολύ μεγάλο ποσοστό του χαρτοφυλακίου (το 70%) στη (νότια) Κεντρική Ιταλία και παράλληλα μηδενίζει το ποσοστό της Βόρειας Ιταλίας, ωστόσο με αυτό το συνδυασμό παρατηρείται μια πολύ καλή τιμή μέσης παραγωγικότητας. Στα άλλα δύο κριτήρια ακολουθεί κι αυτός την τάση που προαναφέρθηκε.

Τέλος, ο Tabu Search επιλέγει κατά κύριο λόγο τη (βόρεια) Κεντρική Ιταλία στο πρώτο κριτήριο, κάτι που εξηγεί και τη χαμηλή τιμή της μέσης παραγωγικότητας έναντι άλλων αλγορίθμων. Στα επόμενα κριτήρια, τείνει και αυτός να ακολουθήσει την προαναφερόμενη τάση.

**Πίνακας 5-6:** Αποτελέσματα υβριδικού σχήματος Simulated Annealing και Harmony Search

Hybrid Simulated Annealing and Harmony Search										
						1st criterion	2nd criterion	3rd criterion		
wbest						CFpbest	IRbest	Pqbest		
$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$			$p_5$	$p_{10}$	$p_{15}$
0,1577	0,0955	0,1596	0,3504	0,0721	0,1646	1,0197				
0,0866	0,0238	0,2106	0,0751	0,2247	0,3792		1,3405			
0,1106	0,0474	0,0857	0,1563	0,1493	0,4506			0,0042	0,0058	0,0069

Αυτό που παρατηρεί κανείς απευθείας για το υβριδικό σχήμα που ερευνήθηκε, είναι πως προτείνει ένα χαρτοφυλάκιο που μεγιστοποιεί την μέση παραγωγικότητα στο 100%, δηλαδή αξιοποιείται κάθε μονάδα εγκατεστημένης ισχύος. Σε αυτό το χαρτοφυλάκιο ενδιαφέρον παρατηρείται στο ότι το 35% αυτού προέρχεται από την Σαρδηνία, ενώ από 16% σχεδόν στη



Νότια Ιταλία. Ωστόσο, βάση του δεύτερου κριτηρίου, το χαρτοφυλάκιο διαμορφώνεται διαφορετικά με 37% να προέρχεται από τη Νότια Ιταλία, 22% από τη Σικελία και 21% από τη Βόρεια Ιταλία, δηλαδή 59% του χαρτοφυλακίου συγκεντρώνεται στο νοτιότερο άκρο της χώρας, κάτι που έχει λογική, καθώς βρέχεται από θάλασσα (αρκετά θαλάσσια αιολικά πάρκα) και συγκεντρώνει υψηλές τιμές ανέμου. Τέλος, αξιοσημείωτο είναι ότι βάση του τρίτου κριτηρίου, το 45% του χαρτοφυλακίου προέρχεται από τη Νότια Ιταλία. Γενικότερα, δηλαδή παρατηρείται αυτή η τάση που προαναφέρθηκε, μιας και η περισσότερη εγκατεστημένη ισχύς είναι συγκεντρωμένη εκεί.

Για να ελέγξουμε την αποτελεσματικότητα των λύσεων που παρήχθησαν, κρατήθηκαν δεδομένα δύο (2) μηνών, μεταξύ 1/11/2010 και 31/12/2010, τα οποία και δόθηκαν σαν ορίσματα εισόδου στην συνάρτηση *test\_energy\_dispatch*. Για κάθε κριτήριο κρατήθηκε το καλύτερο διάνυμα μεταξύ όσων παρήχθησαν από τους αλγόριθμους και το διάνυμα του υβριδικού.

**Πίνακας 5-7:** Αποτελέσματα ελέγχου αποδοτικότητας

	<b>Best algorithm</b>	<b>Hybrid scheme</b>
<b><i>Capacity Factor</i></b>	0,6396	0,6441
<b><i>Information Ratio</i></b>	2,1788	2,2018
<b><i>p<sub>5</sub></i></b>	0,0165	0,0200
<b><i>p<sub>10</sub></i></b>	0,0168	0,0204
<b><i>p<sub>15</sub></i></b>	0,0171	0,0210

Η πρώτη παρατήρηση που μπορεί να γίνει είναι ότι και στις δύο περιπτώσεις ο δείκτης της μέσης παραγωγικότητας ξεπερνά το 60%, γεγονός θετικό μιας και δείχνει ότι οι αλγόριθμοι είναι αποδοτικοί. Το Information Ratio και στις δύο περιπτώσεις ξεπερνά το δύο (2), κάτι που δείχνει ότι μειώνεται κατά πολύ η διασπορά γύρω από τη μέση τιμή της κατανομής των Capacity Factors. Τέλος, παρατηρείται μια ανάλογη βελτίωση των εκατοστημορίων σε σχέση με τις τιμές που έδιναν οι αλγόριθμοι κατά τη μάθηση.

## 6. Συμπεράσματα

Ενδεχομένως το βέλτιστο χαρτοφυλάκιο να διαφοροποιείται από μέρα σε μέρα και από ώρα σε ώρα. Ωστόσο, τα προτεινόμενα χαρτοφυλάκια που προκύπτουν από την παρούσα μελέτη είναι:

Πίνακας 6-1: Προτεινόμενα χαρτοφυλάκια

	CNOR	CSUD	NORD	SARD	SICI	SUD
<b>Max CF</b>	14,59%	21,23%	9,36%	21,52%	25,39%	7,92%
	15,77%	9,55%	15,96%	35,04%	7,21%	16,46%
<b>Max IR</b>	2,60%	1,54%	15,90%	5,55%	26,45%	47,97%
	8,66%	2,38%	21,06%	7,51%	22,47%	37,92%
<b>Max Pq</b>	2,90%	17,42%	7,72%	10,92%	26,50%	34,54%
	11,06%	4,74%	8,57%	15,63%	14,93%	45,06%

Αξίζει να γίνει περαιτέρω έρευνα, ώστε να αναλυθούν περισσότερο τα χαρτοφυλάκια.

Ενδεικτικά, μελλοντικές εργασίες θα μπορούσαν να έχουν στόχους, όπως:

- ✓ Ανάλυση χαρτοφυλακίου στο επίπεδο του μείγματος από αιολικά πάρκα
- ✓ Εύρεση βέλτιστου χαρτοφυλακίου με βάση και τη ζήτηση
- ✓ Παραμετροποίηση των αλγορίθμων για καλύτερα αποτελέσματα
- ✓ Ανάλυση περισσότερων υβριδικών σχημάτων
- ✓ Ανάλυση μορφολογικών παραμέτρων που επηρεάζουν το χαρτοφυλάκιο

Στην παρούσα διπλωματική, πέρα από το γεγονός ότι σε γενικές γραμμές προτείνει χαρτοφυλάκια με λογικούς καταμερισμούς (σύμφωνα με την μορφολογία της κάθε περιοχής), παρουσιάζει πέντε (5) διαφορετικές τεχνικές εύρεσης βέλτιστου ενεργειακού χαρτοφυλακίου, βασισμένες σε μεθόδους που εισάγουν τεχνητή νοημοσύνη. Δηλαδή δημιουργήθηκαν (σε πρωταρχικό στάδιο φυσικά) αλγόριθμοι που μελλοντικά θα μπορούσαν να λαμβάνουν αποφάσεις και να τροποποιούν το χαρτοφυλάκιο κάθε φορά βάσει των παραμέτρων που το επηρεάζουν.

Τέλος, πρέπει να σημειωθεί ότι ο λήπτης απόφασης, δηλαδή ο διαχειριστής του δικτύου, με το μοντέλο που προτείνεται, μπορεί να λάβει ένα προτεινόμενο χαρτοφυλάκιο για την επόμενη μέρα, βάσει της χρονοσειράς όλων των προηγούμενων ημερών. Με τις προσθήκες που προτείνονται παραπάνω, το προτεινόμενο χαρτοφυλάκιο μπορεί να γίνει πιο δυναμικό και να δίνεται ανά ώρα ή ανά εποχή.

## Βιβλιογραφία

### Έντυπα

**Anagnostopoulos, K.P., Mamanis, G. 2009**, *Multiobjective evolutionary algorithms for complex portfolio optimization problems*, Springer-Verlag

**Chakraborty, P., Roy, G. G., Panigrahi, B. K., Bansal, R. C., Mohapatra, A., 2011**, 'Dynamic economic dispatch using harmony search algorithm with modified differential mutation operator' in *Electrical Engineering*, vol. 94, issue 4, pp. 197-205

**Corana, A., Marchesi, M., Martini, C., Ridella, S., 1987**, 'Minimizing Multimodal Functions of Continuous Variables with the 'Simulated Annealing' Algorithm' in *ACM Transactions on Mathematical Software*, Vol. 13, No. 3, September 1987, pp. 262-280

**Dueck, G., Winker, P., 1992**, 'New concepts and algorithms for portfolio choice' in *Applied Stochastic Models and Data Analysis*, Vol. 8, pp. 159-178

**Gao, X.Z. , Wang, X., Ovaska, S.J., 2009**, 'Harmony Search Methods for Multi-modal and Constrained Optimization' in *Music-Inspired Harmony Search Algorithm*, eds. Z.W. Geem, Berlin, SCI 191, pp. 39-51

**Geem, Z. W., Kim, J. H., Loganathan, G.V., 2001**, *A New Heuristic Optimization Algorithm: Harmony Search*, Simulation

**Gendreau, M. 2003**, 'An introduction to Tabu Search' in *Handbook of Metaheuristics*, Springer, pp. 37-54

**Gendreau, M., Potvin, J. Y., 2010**, 'Tabu Search' in *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, Vol. 146, Springer Science+Business Media, LLC 2010, pp. 41-59

**Glover, F., 1989**, 'Tabu Search-Part I' in *ORSA Journal on Computing*, Vol. 1, No. 3, Summer 1989, pp. 190-206

**Glover, F., 1989**, 'Tabu Search-Part II' in *ORSA Journal on Computing*, Vol. 2, No. 1, Winter 1990, pp. 4-32

**Glover, F., 1993**, 'A user's guide to tabu search' in *Annals of Operations Research*, Vol. 41 (1993)

**Gonzalez, J. S., Rodriguez, A. G. G., Mora, J. C., Santos, J. R., Payan, M. B. 2010**, *Optimization of wind farm turbines layout using an evolutive algorithm*, Renewable Energy

**Hendtlash, T. 2008**, *The Particle Swarm Algorithm*, Springer-Verlag

**Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., 1983**, 'Optimization by Simulated Annealing' in *Science*, New Series, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680

**Koutroulis, E., Kolokotsa, D., Potirakis, A., Kalaitzakis, K., 2005**, 'Methodology for optimal sizing of stand-alone photovoltaic/wind-generator systems using genetic algorithms' in *Solar Energy*, vol. 80, issue 9, pp. 1072-1088

**Kuhn, D., Parpas, P., Rustem, B., 2008**, 'Threshold Accepting Approach to Improve Bound-based Approximations for Portfolio Optimization' in *Computational Methods in Financial Engineering*, pp. 3-26

**Kusiak, A., Zheng, H. 2009**, *Optimization of wind turbine energy and power factor with an evolutionary computation algorithm*, Energy

**Parsopoulos, K. E., Vrahatis, M. N., 2002**, 'Recent approaches to global optimization problems through Particle Swarm Optimization' in *Natural Computing 1*, pp. 235–306

**Paterlini, S., 2010**, *Evolutionary Computation for Modelling and Optimization in Finance*, Springer-Verlag

**Sarvari, H., Zamanifar, K., 2011**, *Improvement of harmony search algorithm by using statistical analysis*, Springer Science+Business Media

**Schaerf, A., 2002**, 'Local Search Techniques for Constrained Portfolio Selection Problems' in *Computational Economics*, Vol. 20, pp. 177–190

**Thomaidis, N. (ed.) 2012**, 'Designing strategies for optimal spatial distribution of wind power', *Proceedings of Conference on "Energy and Climate Change"*, Athens, pp. 33-41

**Valdez, F., Melin, P., Castillo, O., 2007**, 'Evolutionary Computing for the Optimization of Mathematical Functions' in *Anal. and Des. of Intel. Sys. using SC Tech.*, eds. P. Melin et al., ASC 41, pp. 463–472

**Vanderbilt, D., Eouie, S., 1984**, 'A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables' in *Journal of computational physics*, Vol. 56, pp. 259-271

**Vasebi A., Fesanghary M., Bathaee S.M.T., 2007**, 'Combined heat and power economic dispatch by harmony search algorithm' in *International Journal of Electrical Power & Energy Systems*, vol. 29, issue 10, pp. 713-719

**Vassiliadis, V., Dounias, G., Tzanetos, A. 2012**, 'Parameter tuning of hybrid nature-inspired intelligent metaheuristics for solving financial portfolio optimization problems', *Proceedings of 7th Hellenic Conference on AI*, SETN 2012, Lamia, pp. 198-205

**Wang, L., Singh, C., 2007**, *Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm*, *Electric Power Systems Research*, Vol. 77, pp. 1654-1664

**Winker, P., Maringer, D., 2007**, 'The Threshold Accepting Optimisation Algorithm in Economics and Statistics' in *Optimization, Econometrics and Financial Analysis*, Part II, pp. 107-125

**Yang, X.-S. , 2009**, 'Harmony Search as a Metaheuristic Algorithm' in *Music-Inspired Harmony Search Algorithm*, eds. Z.W. Geem, Berlin, SCI 191, pp. 1–14

**Yang, X.-S., 2011** , 'Optimization Algorithms' in *Comput. Optimization, Methods and Algorithms*, eds S. Koziel & X.-S. Yang, Berlin, pp 13-31

**Zafar, H., Chowdhury, A., Panigrahi, 2011**, 'Solution of Economic Load Dispatch Problem Using Lbest-Particle Swarm Optimization with Dynamically Varying Sub-swarms' in *SEMCCO 2011*, eds. B. K., B.K. Panigrahi et al. , Part I, LNCS 7076, pp. 191–198

**Μαρούδας Δ., 2008**, Συγκριτική Ανάλυση Αιολικών Συστημάτων και Έλεγχος Ισχύος σε Σύστημα με ΑΜΔΤ (Ασύγχρονη Μηχανή Διπλής Τροφοδοσίας)

### *Ιστότοποι*

[http://en.wikipedia.org/wiki/Information\\_ratio](http://en.wikipedia.org/wiki/Information_ratio)

[http://www.terna.it/default/Home/SISTEMA\\_ELETTRICO/transparency\\_report.aspx](http://www.terna.it/default/Home/SISTEMA_ELETTRICO/transparency_report.aspx)

<http://www.wind-power-program.com>

<http://www.ewea.org/policy-issues/wind-energy-and-the-grid/>

## ΠΑΡΑΡΤΗΜΑ: ΑΛΓΟΡΙΘΜΟΙ

### Αλγόριθμος Π1: Συνάρτηση randfixedsum

```
function [x,v] = randfixedsum(n,m,s,a,b)

% [x,v] = randfixedsum(n,m,s,a,b)
%
% This generates an n by m array x, each of whose m columns
% contains n random values lying in the interval [a,b], but
% subject to the condition that their sum be equal to s. The
% scalar value s must accordingly satisfy  $n*a \leq s \leq n*b$ . The
% distribution of values is uniform in the sense that it has the
% conditional probability distribution of a uniform distribution
% over the whole n-cube, given that the sum of the x's is s.
%
% The scalar v, if requested, returns with the total
% n-1 dimensional volume (content) of the subset satisfying
% this condition. Consequently if v, considered as a function
% of s and divided by  $\sqrt{n}$ , is integrated with respect to s
% from  $s = a$  to  $s = b$ , the result would necessarily be the
% n-dimensional volume of the whole cube, namely  $(b-a)^n$ .
%
% This algorithm does no "rejecting" on the sets of x's it
% obtains. It is designed to generate only those that satisfy all
% the above conditions and to do so with a uniform distribution.
% It accomplishes this by decomposing the space of all possible x
% sets (columns) into n-1 dimensional simplexes. (Line segments,
% triangles, and tetrahedra, are one-, two-, and three-dimensional
% examples of simplexes, respectively.) It makes use of three
% different sets of 'rand' variables, one to locate values
% uniformly within each type of simplex, another to randomly
% select representatives of each different type of simplex in
% proportion to their volume, and a third to perform random
% permutations to provide an even distribution of simplex choices
% among like types. For example, with n equal to 3 and s set at,
% say, 40% of the way from a towards b, there will be 2 different
% types of simplex, in this case triangles, each with its own
% area, and 6 different versions of each from permutations, for
% a total of 12 triangles, and these all fit together to form a
% particular planar non-regular hexagon in 3 dimensions, with v
% returned set equal to the hexagon's area.
%
% Roger Stafford - Jan. 19, 2006

% Check the arguments.
if (m~=round(m))|(n~=round(n))|(m<0)|(n<1)
    error('n must be a whole number and m a non-negative integer.')
elseif (s<n*a)|(s>n*b)|(a>=b)
    error('Inequalities  $n*a \leq s \leq n*b$  and  $a < b$  must hold.')
end

% Rescale to a unit cube:  $0 \leq x(i) \leq 1$ 
s = (s-n*a)/(b-a);
```

```

% Construct the transition probability table, t.
% t(i,j) will be utilized only in the region where j <= i + 1.
k = max(min(floor(s),n-1),0); % Must have 0 <= k <= n-1
s = max(min(s,k+1),k); % Must have k <= s <= k+1
s1 = s - [k:-1:k-n+1]; % s1 & s2 will never be negative
s2 = [k+n:-1:k+1] - s;
w = zeros(n,n+1); w(1,2) = realmax; % Scale for full 'double' range
t = zeros(n-1,n);
tiny = 2^(-1074); % The smallest positive matlab 'double' no.
for i = 2:n
    tmp1 = w(i-1,2:i+1).*s1(1:i)/i;
    tmp2 = w(i-1,1:i).*s2(n-i+1:n)/i;
    w(i,2:i+1) = tmp1 + tmp2;
    tmp3 = w(i,2:i+1) + tiny; % In case tmp1 & tmp2 are both 0,
    tmp4 = (s2(n-i+1:n) > s1(1:i)); % then t is 0 on left & 1 on right
    t(i-1,1:i) = (tmp2./tmp3).*tmp4 + (1-tmp1./tmp3).*(~tmp4);
end

% Derive the polytope volume v from the appropriate
% element in the bottom row of w.
v = n^(3/2)*(w(n,k+2)/realmax)*(b-a)^(n-1);

% Now compute the matrix x.
x = zeros(n,m);
if m == 0, return, end % If m is zero, quit with x = []
rt = rand(n-1,m); % For random selection of simplex type
rs = rand(n-1,m); % For random location within a simplex
s = repmat(s,1,m);
j = repmat(k+1,1,m); % For indexing in the t table
sm = zeros(1,m); pr = ones(1,m); % Start with sum zero & product 1
for i = n-1:-1:1 % Work backwards in the t table
    e = (rt(n-i,:) <= t(i,j)); % Use rt to choose a transition
    sx = rs(n-i,:).^ (1/i); % Use rs to compute next simplex coord.
    sm = sm + (1-sx).*pr.*s/(i+1); % Update sum
    pr = sx.*pr; % Update product
    x(n-i,:) = sm + pr.*e; % Calculate x using simplex coords.
    s = s - e; j = j - e; % Transition adjustment
end
x(n,:) = sm + pr.*s; % Compute the last x

% Randomly permute the order in the columns of x and rescale.
rp = rand(n,m); % Use rp to carry out a matrix 'randperm'
[ig,p] = sort(rp); % The values placed in ig are ignored
x = (b-a)*x(p+repmat([0:n*(m-1)],n,1))+a; % Permute & rescale x

```

## Αλγόριθμος Π2: Συνάρτηση SA\_energy\_dispatch

```
function [IRbest wbest CFpbest aux Pqbest]=SA_energy_dispatch(data);
%this functions implements a type of Simulated Annealing algorithm for
%(energy portfolio's) weight optimization
%INPUT wind energy (MWh)
%returns: Information Ratio, max Mean CF, percentiles

%% Parameter Initialization
T=10000; %Temperature parameter
r=0.85; %Parameter for temperature (T) reduction
gen=1000; %number of generations
pop=24622; %population (data are from 1026 days for 24 hours/day)
k=6; %number or zones
TAC=zeros(1,6);
aux=zeros(5,gen);

%% Solution Initialization
wl=0;
wu=1;
[w,v]=randfixedsum(k,gen,1,wl,wu);
w=w'; % inversion of w's matrix
CF=zeros(pop,6);
CFp=zeros(pop,1);
CFpn=zeros(pop,1);
wbest=zeros(3,6);
Pqbest=zeros(1,3);

for i=1:pop
    CFS=0;
    for j=1:6
        TAC=choose_tac(pop,j);
        CF(i,j)=(data(i,j)*w(1,j))/TAC; % capacity factor of every unit
        CFS=CFs+CF(i,j);
    end
    CFp(i)=CFS; % total capacity factor
end

E_CF=mean(CFp); % mean of capacity factor
S_CF=std(CFp); % standard deviation of capacity factor
IR=(E_CF/S_CF); % objective/fitness function
Pq0 = prctile(CFp,[.05 .1 .15]); % Percentiles (p5,p10,p15)

for c=1:3
    for zn=1:6
        wbest(c,zn)=w(1,zn);
    end
end
IRbest=max(IR);
CFpbest=max(CFp);
Pqbest=Pq0';
aux(:,1)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];

%% Find solution
for m=1:gen
```



```

%% Create new solutions

for i=1:pop
    CFs=0;
    for j=1:6
        TAC=choose_tac(pop,j);
        CF(i,j)=(data(i,j)*w(m,j))/TAC; % capacity factor of every unit
        CFs=CFs+CF(i,j);
    end
    CFpn(i)=CFs; % total capacity factor
end

E_CFn=mean(CFpn); % mean of capacity factor
S_CFn=std(CFpn); % standard deviation of capacity factor
IRn=(E_CFn/S_CFn); % objective/fitness function
Pqn = prctile(CFpn,[.05 .1 .15]); % Percentiles (p5,p10,p15)
CFpnbest=max(CFpn);

%% Acceptance criterion (Metropolis criterion)
if T||0
    pwr=(IRn-IRbest)/T;
    Probability=exp(pwr)/100;
end

if Probability>=0.60 && IRn>IRbest
    IRbest=IRn;
    wbest(2,:)=w(m,:);
end

if Probability>=0.60 && CFpnbest>CFpbest
    wbest(1,:)=w(m,:);
    CFpbest=CFpnbest;
end

if Probability>=0.60 && Pqn(1)>Pqbest(1)
    wbest(3,:)=w(m,:);
    Pqbest=Pqn';
end

aux(:,m)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];
T=r*T;
end
end

```

### Αλγόριθμος Π3: Συνάρτηση PSO\_energy\_dispatch

```
function [IRbest wbest CFpbest aux Pqbest]=PSO_energy_dispatch(data);
%this functions implements a type of Particle Swarm Optimization algorithm
for
%(portfolio's) weight optimization
%INPUT wind energy (MWh)
%returns: Information Ratio, max Mean CF, percentiles

%% Parameter Initialization
pop=24622; %number of population
gen=1000; %number of generations
a=0;
b=1; %velocity's borders
G=0.9; %high point of attraction
L=0.5; %low point of attraction
M=0.9; %momentum of the particle
k=6; %number of zones
TAC=zeros(1,6);
R1=rand(100,1);
R2=rand(100,1); %both are in the range [0,1]
aux=zeros(5,gen);

%% Solution Initialization
wl=0;
wu=1;
[w,v]=randfixedsum(k,1,1,wl,wu);
w=w'; % inversion of w's matrix
CF=zeros(pop,6);
CFp=zeros(pop,1);
CFpn=zeros(pop,1);
wbest=zeros(3,6);
d=zeros(gen,1);
pbest=zeros(pop,1);

for i=1:pop
    CFs=0;
    for j=1:6
        TAC=choose_tac(pop,j);
        CF(i,j)=(data(i,j)*w(1,j))/TAC; % capacity factor of every unit
        CFs=CFs+CF(i,j);
    end
    CFp(i)=CFs; % total capacity factor
end

E_CF=mean(CFp); % mean of capacity factor
S_CF=std(CFp); % standard deviation of capacity factor
IR=(E_CF/S_CF); % objective/fitness function
Pq0 = prctile(CFp,[.05 .1 .15]); % Percentiles (p5,p10,p15)

IRbest=max(IR);
for c=1:3
    for zn=1:6
        wbest(c,zn)=w(1,zn);
    end
end
```

```

end
CFbest=CFs;
Pqbest=Pq0';
vel=rand(1,k); %Calculate velocity for each particle
                %Small initial velocity needed

for i=1:pop
    pbest(i)=CFp(i); % initialize particle's best
end

d(1)=inf; % distance from the particle itself
min_d=inf;
for i=1:gen
    if i||1
        d(i)=CFp(i)-CFp(i+1); % find best neighbor
    end
    if min_d>d(i)
        min_d=d(i);
        pos=i; % find position of best neighbor
    end
end
lbest=CFp(pos);
fbest=max(pbest);
CFpbest=max(CFp);
aux(:,1)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];

%% Find solution
for m=1:gen
    t=gen; % time goes by like the generations
    sw=0;
    for x=1:6
        w(x)=w(x)+t*(vel(x)); % update particle's position
        sw=sw+w(x);
    end

    for x=1:6
        w(x)=w(x)/sw; % normalize w's [in bandwidth (0,1)]
    end

    if m||1

        for i=1:pop
            CFs=0;
            for j=1:6
                TAC=choose_tac(pop,j);
                CF(i,j)=(data(i,j)*w(1,j))/TAC; % capacity factor of every
unit
                CFs=CFs+CF(i,j);
            end
            CFpn(i)=CFs; % total capacity factor
        end

        E_CFn=mean(CFpn); % mean of capacity factor
        S_CFn=std(CFpn); % standard deviation of capacity factor
        IRn=(E_CFn/S_CFn); % objective/fitness function

```

```

Pqn = prctile(CFpn,[.05 .1 .15]); % Percentiles (p5,p10,p15)
Pqn=Pqn';
CFpnbest=max(CFpn);

if pbest(m)>CFpn(m)
    pbest(m)=CFpn(m); %finding the best position of each particle
end

d(m)=inf;% distance from the particle itself
min_d=inf;
for y=1:gen-1
    if y|m
        d(y)=CFpn(y)-CFpn(y+1); % find best neighbor
    end
    if min_d>d(y)
        min_d=d(y);
        pos=y; % find position of best neighbor
    end
end
lbest=CFpn(pos);
fbest=max(pbest);

if CFpnbest>CFpbest
    CFpbest=CFpnbest;
    wbest(1,:)=w(1,:);
end

if IRn>IRbest
    IRbest=IRn;
    wbest(2,:)=w(1,:);
end

if Pqn(1)>Pqbest(1)
    Pqbest=Pqn;
    wbest(3,:)=w(1,:);
end

aux(:,m)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];

%% Update particle's position

R1=rand(1,1);
R2=rand(1,1); %both are in the range [0,1]
for x=1:6
    vel(x)=(M*vel(x))+((1-M)*((G*R1*(fbest-CFpn(m))/t)+(L*R2*(lbest-
CFpn(m))/t))); %Calculate the velocity of the particle
end

sw=0;
for x=1:6
    w(x)=w(x)+t*(vel(x)); % update particle's position
    sw=sw+w(x);
end
for x=1:6
    w(x)=w(x)/sw; % normalize w's [in bandwidth (0,1)]
end

```

end

end

end

end

#### Αλγόριθμος Π4: Συνάρτηση HS\_energy\_dispatch

```
function [fbest wbest aux Pqn]=HS_energy_dispatch(data);
%this functions implements a type of Harmony Search algorithm for
%(portfolio's) weight optimization

%% Parameter Initialization
gen=1000; %number of generations
cmp=24622; %number of components = pop
hms=6; %harmony memory size = k
hmcr=0.8; %harmony memory consideration rate (0.7~0.95)
par=0.3; %pitch adjusting rate (0.1~0.5)
u_bnd=1; %upper bound for probabilities
l_bnd=0; %lower bound for probabilities

%% Harmony memory initialization

wl=0;
wu=1;
[w,v]=randfixedsum(hms,1,1,wl,wu);
w=w'; % inversion of w's matrix
CF=zeros(cmp,6);
CFp=zeros(cmp,1);
CFpn=zeros(cmp,1);
wbest=zeros(3,6);
Pqbest=zeros(1,3);

for i=1:cmp
    CFs=0;
    for j=1:6
        TAC=choose_tac(cmp,j);
        CF(i,j)=(data(i,j)*w(1,j))/TAC; % capacity factor of every unit
        CFs=CFs+CF(i,j);
    end
    CFp(i)=CFs; % total capacity factor
end

E_CF=mean(CFp); % mean of capacity factor
S_CF=std(CFp); % standard deviation of capacity factor
IR=(E_CF/S_CF); % objective/fitness function
Pq0 = prctile(CFp,[.05 .1 .15]); % Percentiles (p5,p10,p15)

for c=1:3
    for zn=1:6
        wbest(c,zn)=w(1,zn);
    end
end
IRbest=max(IR);
CFpbest=max(CFp);
Pqbest=Pq0';
aux(:,1)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];

%% Find solution
for m=1:gen
    %% New harmony improvisation
```

```

[wn,v]=randfixedsum(hms,1,1,wl,wu);
wn=wn'; % inversion of wn's matrix
pr_hmcr=l_bnd+rand*(u_bnd-l_bnd);
pr_par=l_bnd+rand*(u_bnd-l_bnd);
sw=0;
for i=1:hms
    if hmcr<pr_hmcr %memory consideration
        w(i)=wn(i);
    end
    if par<pr_par %pitch adjustment
        w(i)=w(i)+wn(i);
    end
    sw=sw+w(i);
end

for i=1:hms
    w(i)=w(i)/sw; % normalize w's [in bandwidth (0,1)]
end

%% Harmony memory update

for i=1:cmp
    CFs=0;
    for j=1:6
        TAC=choose_tac(cmp,j);
        CF(i,j)=(data(i,j)*w(1,j))/TAC; % capacity factor of every unit
        CFs=CFs+CF(i,j);
    end
    CFpn(i)=CFs; % total capacity factor
end

E_CFn=mean(CFpn); % mean of capacity factor
S_CFn=std(CFpn); % standard deviation of capacity factor
IRn=(E_CFn/S_CFn); % objective/fitness function
Pqn = prctile(CFpn,[.05 .1 .15]); % Percentiles (p5,p10,p15)
CFpnbest=max(CFpn);

if CFpnbest>CFpbest
    CFpbest=CFpnbest;
    wbest(1,:)=wn(1,:);
end

if IRn>IRbest
    IRbest=IRn;
    wbest(2,:)=wn(1,:);
end

if Pqn(1)>Pqbest(1)
    Pqbest=Pqn';
    wbest(3,:)=wn(1,:);
end
aux(:,m)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];
end
end

```

### Αλγόριθμος Π5: Συνάρτηση TS\_energy\_dispatch

```
function [IRbest wbest CFpbest aux Pqbest]=TS_energy_dispatch(data);
%this functions implements a type of Tabu Search algorithm for
%(energy portfolio's) weight optimization
%INPUT wind energy (MWh)
%returns: Information Ratio, max Mean CF, percentiles

%% Parameter Initialization
gen=1000; %number of generations
pop=24622; %population (data are from 1026 days for 24 hours/day)
k=6; %number or zones
TAC=zeros(1,6);
aux=zeros(5,gen);
cl_nbg=zeros(gen,3);
list_size=100; %size of the tabu list
Tabu_list_1=zeros(list_size,6);
Tabu_list_2=zeros(list_size,6);
Tabu_list_3=zeros(list_size,6); % one list for every criterion
counter_1=2;
counter_2=2;
counter_3=2; % counters are for moving to the next row of the tabu list

%% Solution Initialization
wl=0;
wu=1;
[w,v]=randfixedsum(k,gen,1,wl,wu);
w=w'; % inversion of w's matrix
CF=zeros(pop,6);
CFp=zeros(pop,1);
CFpn=zeros(pop,1);
wbest=zeros(3,6);
Pqbest=zeros(1,3);
IR=zeros(gen,1);
Pq=zeros(gen,3);

for i=1:pop
    CFs=0;
    for j=1:6
        TAC=choose_tac(pop,j);
        CF(i,j)=(data(i,j)*w(1,j))/TAC; % capacity factor of every unit
        CFs=CFs+CF(i,j);
    end
    CFp(i)=CFs; % total capacity factor
end

E_CF=mean(CFp); % mean of capacity factor
S_CF=std(CFp); % standard deviation of capacity factor
IR(1)=(E_CF/S_CF); % objective/fitness function
Pq(1,:) = prctile(CFp,[.05 .1 .15]); % Percentiles (p5,p10,p15)

for c=1:3
    for zn=1:6
        wbest(c,zn)=w(1,zn);
    end
end
```



```

end
Tabu_list_1(1,:)=wbest(1,:);
Tabu_list_2(1,:)=wbest(2,:);
Tabu_list_3(1,:)=wbest(3,:);
IRbest=max(IR);
CFpbest=max(CFp);
Pqbest=(Pq(1,:))';
aux(:,1)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];

%% Find solution
for m=1:gen

    %% Checks for clearing out the tabu lists if they are full
    if counter_1==list_size
        Tabu_list_1=zeros(list_size,6);
        counter_1=1;
    end
    if counter_2==list_size
        Tabu_list_2=zeros(list_size,6);
        counter_2=1;
    end
    if counter_3==list_size
        Tabu_list_3=zeros(list_size,6);
        counter_3=1;
    end

    %% Check the tabu list if there's the proposed w vector
    status=0; % binary variable for checking the solution existence in Tabu
list
    for x=1:tabu_list
        if Tabu_list(x,:)==w(m,:)
            status=1;
            break
        end
    end
    if status==1
        [w1,v]=randfixedsum(k,1,1,wl,wu);
        w(m,:)=w1';
    end

    %% Create new solutions

    for i=1:pop
        CFs=0;
        for j=1:6
            TAC=choose_tac(pop,j);
            CF(i,j)=(data(i,j)*w(m,j))/TAC; % capacity factor of every unit
            CFs=CFs+CF(i,j);
        end
        CFpn(i)=CFs; % total capacity factor
    end

    E_CFn=mean(CFpn); % mean of capacity factor
    S_CFn=std(CFpn); % standard deviation of capacity factor
    IR(m)=(E_CFn/S_CFn); % objective/fitness function
    Pq(m,:)=prctile(CFpn,[.05 .1 .15]); % Percentiles (p5,p10,p15)

```

```

CFpbest=max(CFpn);

%% Neighborhood search
nbg(m,:)=inf; % distance from itself
cl_nbg(m,:)=inf;
for y=1:gen
    if y|m
        nbg(m,1)=abs(CFpn(m)-CFp(y));
        if cl_nbg(m,1)>nbg(m,1)
            cl_nbg(m,1)=nbg(m,1); % find best neighbor (1st criterion)
            pos(1)=m;
        end
        nbg(m,2)=abs(IR(m)-IR(y));
        if cl_nbg(m,2)>nbg(m,2)
            cl_nbg(m,2)=nbg(m,2); % find best neighbor (2nd criterion)
            pos(2)=m;
        end
        nbg(m,3)=abs(sum(Pq(m,:))-sum(Pq(y,:)));
        if cl_nbg(m,3)>nbg(m,3)
            cl_nbg(m,3)=nbg(m,3); % find best neighbor (3rd criterion)
            pos(3)=m;
        end
    end
end

if CFpn(pos(1))>CFpbest
    wbest(1,:)=w(pos(1),:);
    CFpbest=CFpn(pos(1));
    Tabu_list_1(counter_1,:)=wbest(1,:);
    counter_1=counter_1+1;
end

if IR(pos(2))>IRbest
    wbest(2,:)=w(pos(2),:);
    IRbest=IR(pos(2));
    Tabu_list_2(counter_2,:)=wbest(2,:);
    counter_2=counter_2+1;
end

if Pq(pos(3),1)>Pqbest(1)
    wbest(3,:)=w(pos(3),:);
    Pqbest=(Pq(pos(3),:))';
    Tabu_list_3(counter_3,:)=wbest(3,:);
    counter_3=counter_3+1;
end

aux(:,m)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];
end
end

```

## Αλγόριθμος Π6: Συνάρτηση TA\_energy\_dispatch

```
function [IRbest wbest CFpbest aux Pqbest]=TA_energy_dispatch(data);
%this functions implements a type of Threshold Accepting algorithm for
%(portfolio's) weight optimization
%INPUT wind energy (MWh)
%returns: Information Ratio, max Mean CF, percentiles

%% Parameter Initialization
wl=0;
wu=1;
gen=1000; %number of generations
pop=24622; %population (data are from 661 days)
k=6; %number or zones
TAC=zeros(1,6);
aux=zeros(5,gen);
cl_nbg=zeros(gen,3);

%% Solution Initialization
[w,v]=randfixedsum(k,gen,1,wl,wu);
w=w'; % inversion of w's matrix
CF=zeros(pop,6);
CFp=zeros(pop,1);
CFpn=zeros(pop,1);
wbest=zeros(1,6);
IR=zeros(gen,1);
Pq=zeros(gen,3);

for i=1:pop
    CFS=0;
    for j=1:6
        TAC=choose_tac(pop,j);
        CF(i,j)=(data(i,j)*w(1,j))/TAC; % capacity factor of every unit
        CFS=CFs+CF(i,j);
    end
    CFp(i)=CFS; % total capacity factor
end

E_CF=mean(CFp); % mean of capacity factor
S_CF=std(CFp); % standard deviation of capacity factor
IR(1)=(E_CF/S_CF); % objective/fitness function
Pq(1,:) = prctile(CFp,[.05 .1 .15]); % Percentiles (p5,p10,p15)
IRbest=max(IR);

for c=1:3
    for zn=1:6
        wbest(c,zn)=w(1,zn);
    end
end
CFpbest=max(CFp);
Pqbest=(Pq(1,:))';
aux(:,1)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];

%% Find solution
for m=1:gen
```

```

%% Create new solutions

for i=1:pop
    CFs=0;
    for j=1:6
        TAC=choose_tac(pop,j);
        CF(i,j)=(data(i,j)*w(m,j))/TAC; % capacity factor of every unit
        CFs=CFs+CF(i,j);
    end
    CFpn(i)=CFs; % total capacity factor
end

E_CFn=mean(CFpn); % mean of capacity factor
S_CFn=std(CFpn); % standard deviation of capacity factor
IR(m)=(E_CFn/S_CFn); % objective/fitness function
Pq(m,:)=prctile(CFpn,[.05 .1 .15]); % Percentiles (p5,p10,p15)
CFpbest=max(CFpn);

%% Neighborhood search
nbg(m,:)=inf; % distance from itself
cl_nbg(m,:)=inf;
for y=1:gen
    if y|m
        nbg(m,1)=abs(CFpn(m)-CFp(y));
        if cl_nbg(m,1)>nbg(m,1)
            cl_nbg(m,1)=nbg(m,1); % find best neighbor (1st criterion)
            pos(1)=m;
        end
        nbg(m,2)=abs(IR(m)-IR(y));
        if cl_nbg(m,2)>nbg(m,2)
            cl_nbg(m,2)=nbg(m,2); % find best neighbor (2nd criterion)
            pos(2)=m;
        end
        nbg(m,3)=abs(sum(Pq(m,:))-sum(Pq(y,:)));
        if cl_nbg(m,3)>nbg(m,3)
            cl_nbg(m,3)=nbg(m,3); % find best neighbor (3rd criterion)
            pos(3)=m;
        end
    end
end

if CFpn(pos(1))>CFpbest
    wbest(1,:)=w(pos(1),:);
    CFpbest=CFpn(pos(1));
end

if IR(pos(2))>IRbest
    wbest(2,:)=w(pos(2),:);
    IRbest=IR(pos(2));
end

if Pq(pos(3),1)>Pqbest(1)
    wbest(3,:)=w(pos(3),:);
    Pqbest=(Pq(pos(3),:))';
end
end

```

```
    aux(:,m)= [CFpbest; IRbest; Pqbest(1); Pqbest(2); Pqbest(3)];  
end  
end
```

### Αλγόριθμος Π7: Συνάρτηση choose\_tac

```
function [tac_i]=choose_tac(pr_time,area);
% This function is for choosing the exact total available capacity
% according to the day (or the pop value) we are (in 2011-2013)
% INPUT: pop value

tac=zeros(1,6);
if pr_time>=18288
    tac=[69.4 934.34 55.6 817.72 1554.77 2990.34];
    tac_i=tac(area);
elseif pr_time>=17544
    tac=[20 934.34 55.6 817.72 1554.77 2990.34];
    tac_i=tac(area);
elseif pr_time>=11664
    tac=[20 934.34 43.3 817.72 1554.77 2865.94];
    tac_i=tac(area);
elseif pr_time>=8760
    tac=[20 934.34 32.5 817.72 1554.77 2865.94];
    tac_i=tac(area);
elseif pr_time>=8016
    tac=[20 934.34 22.9 817.72 1554.77 2835.19];
    tac_i=tac(area);
elseif pr_time>=7296
    tac=[20 934.34 22.9 817.72 1490.77 2786.99];
    tac_i=tac(area);
elseif pr_time>=5832
    tac=[20 934.34 22.9 817.72 1490.77 2763.19];
    tac_i=tac(area);
elseif pr_time>=5088
    tac=[20 934.34 22.9 817.72 1426.17 2763.19];
    tac_i=tac(area);
elseif pr_time>=4344
    tac=[20 934.34 22.9 817.72 1426.17 2725.69];
    tac_i=tac(area);
elseif pr_time>=3624
    tac=[20 934.34 22.9 659.02 1426.17 2709.69];
    tac_i=tac(area);
elseif pr_time>=2880
    tac=[20 917.34 22.9 659.02 1319.62 2709.69];
    tac_i=tac(area);
elseif pr_time>=2160
    tac=[20 917.34 22.9 659.02 1319.62 2612.19];
    tac_i=tac(area);
elseif pr_time>=1416
    tac=[20 882.79 22.9 659.02 1319.62 2612.19];
    tac_i=tac(area);
else
    tac=[20 816.79 22.9 659.02 1319.62 2612.19];
    tac_i=tac(area);
end

end
```

## Αλγόριθμος Π8: Συνάρτηση test\_energy\_dispatch

```
% This m-file is for testing the results of the algorithms

%% Parameter Initialization
pop=888; %population (data are from 1026 days for 24 hours/day)
TAC=[20 816.79 22.9 659.02 1261.62 2404.84];
TAC1=[20 816.79 22.9 659.02 1319.62 2536.24];

%% Test Capacity Factor

CF_al=zeros(pop,6);
CF_hy=zeros(pop,6);
CFp_al=zeros(pop,1);
CFp_hy=zeros(pop,1);
wbest_1=[0.1459 0.2123 0.0936 0.2152 0.2539 0.0792];
whybr_1=[0.1577 0.0955 0.1596 0.3504 0.0721 0.1646];

for i=1:pop
    CFs_al=0;
    CFs_hy=0;
    for j=1:6
        if i<720
            CF_al(i,j)=(data(i,j)*wbest_1(1,j))/TAC(j); % capacity factor of
every unit
            CF_hy(i,j)=(data(i,j)*whybr_1(1,j))/TAC(j);
        else
            CF_al(i,j)=(data(i,j)*wbest_1(1,j))/TAC1(j); % capacity factor of
every unit
            CF_hy(i,j)=(data(i,j)*whybr_1(1,j))/TAC1(j);
        end
        CFs_al=CFs_al+CF_al(i,j);
        CFs_hy=CFs_hy+CF_hy(i,j);
    end
    CFp_al(i)=CFs_al; % total capacity factor
    CFp_hy(i)=CFs_hy;
end

CFp_al=max(CFp_al);
CFp_hy=max(CFp_hy);

%% Test Information Ratio

CF_al=zeros(pop,6);
CF_hy=zeros(pop,6);
CFp_al=zeros(pop,1);
CFp_hy=zeros(pop,1);
wbest_2=[0.0260 0.0154 0.1590 0.0555 0.2645 0.4797];
whybr_2=[0.0866 0.0238 0.2106 0.0751 0.2247 0.3792];

for i=1:pop
    CFs_al=0;
    CFs_hy=0;
    for j=1:6
        if i<720
```

```

        CF_al(i,j)=(data(i,j)*wbest_2(1,j))/TAC(j); % capacity factor of
every unit
        CF_hy(i,j)=(data(i,j)*whybr_2(1,j))/TAC(j);
    else
        CF_al(i,j)=(data(i,j)*wbest_2(1,j))/TAC1(j); % capacity factor of
every unit
        CF_hy(i,j)=(data(i,j)*whybr_2(1,j))/TAC1(j);
    end
    CFs_al=CFs_al+CF_al(i,j);
    CFs_hy=CFs_hy+CF_hy(i,j);
end
CFp_al(i)=CFs_al; % total capacity factor
CFp_hy(i)=CFs_hy;
end

E_CF_al=mean(CFp_al); % mean of capacity factor
E_CF_hy=mean(CFp_hy);
S_CF_al=std(CFp_al); % standard deviation of capacity factor
S_CF_hy=std(CFp_hy);
IR_al=(E_CF_al/S_CF_al);
IR_hy=(E_CF_hy/S_CF_hy); % objective/fitness function

%% Test Percentiles

CF_al=zeros(pop,6);
CF_hy=zeros(pop,6);
CFp_al=zeros(pop,1);
CFp_hy=zeros(pop,1);
wbest_3=[0.0290 0.1742 0.0772 0.1092 0.2650 0.3454];
whybr_3=[0.1106 0.0474 0.0857 0.1563 0.1493 0.4506];

for i=1:pop
    CFs_al=0;
    CFs_hy=0;
    for j=1:6
        if i<720
            CF_al(i,j)=(data(i,j)*wbest_3(1,j))/TAC(j); % capacity factor of
every unit
            CF_hy(i,j)=(data(i,j)*whybr_3(1,j))/TAC(j);
        else
            CF_al(i,j)=(data(i,j)*wbest_3(1,j))/TAC1(j); % capacity factor of
every unit
            CF_hy(i,j)=(data(i,j)*whybr_3(1,j))/TAC1(j);
        end
        CFs_al=CFs_al+CF_al(i,j);
        CFs_hy=CFs_hy+CF_hy(i,j);
    end
    CFp_al(i)=CFs_al; % total capacity factor
    CFp_hy(i)=CFs_hy;
end

Pq_al = prctile(CFp_al,[.05 .1 .15]); % Percentiles (p5,p10,p15)
Pq_hy = prctile(CFp_hy,[.05 .1 .15]);

```