



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΦΑΡΜΟΣΜΕΝΑ ΗΛΕΚΤΡΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη αλγορίθμου για την εύρεση του ολικού σημείου μέγιστης ισχύος, σε φωτοβολταϊκή συστοιχία υπό συνθήκες μερικής σκίασης»

Του φοιτητή
Ιωάννη Κατσαρώνα
Αρ. Μητρώου: 51908Μ

Επιβλέπων
Κιοσκερίδης Ιορδάνης
Καθηγητής

Μάρτιος 2022

Ανάπτυξη αλγορίθμου για την εύρεση του ολικού σημείου μέγιστης ισχύος, σε φωτοβολταϊκή
συστοιχία υπό συνθήκες μερικής σκίασης

Κωδικός Δ.Ε. 21345

Ιωάννης Κατσαρώνας

Ιορδάνης Κιοσκερίδης

Ανάληψη: Σεπτέμβριος 2020

Περάτωση: Φεβρουάριος 2022

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς, είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Μεταπτυχιακό Πρόγραμμα Σπουδών «Εφαρμοσμένα Ηλεκτρονικά Συστήματα» στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ιωάννη Κατσαρώνα που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Στη σύζυγό μου Άννα και στους μικρούς μου, Σταύρο και Μιχάλη.

Πρόλογος

Ένας μεγάλος αριθμός ηγετών στον τομέα της τεχνολογίας έχει επισημάνει ότι υπάρχει μια τεράστια διαφορά μεταξύ της τεχνολογικής σχεδίασης και της επιστημονικής σχεδίασης ενός συστήματος. Η διαφορά τους εντοπίζεται στην ανάλυση και στη βελτιστοποίηση. Η διαδικασία της επιστημονικής ανάλυσης χρησιμοποιεί μαθηματικά μοντέλα και επιστημονικές μεθόδους, έτσι ώστε να βοηθήσει το σχεδιαστή του συστήματος να εκτιμήσει αποτελέσματα και καταστάσεις που θα παρουσιάζει το προς υλοποίηση σύστημα. Αντιθέτως, η διαδικασία της βελτιστοποίησης είναι μια επαναλαμβανόμενη διαδικασία, η οποία μέσα από τους σχεδιαστικούς περιορισμούς και κριτήρια επιτρέπει στο σχεδιαστή να εντοπίσει τη βέλτιστη εκείνη λύση ή τεχνική, που μπορεί να χρησιμοποιηθεί έτσι ώστε να επιλυθούν τα προβλήματα που προκύπτουν κατά τα πρώιμα στάδια της ανάπτυξης του συστήματος. Ένας από τους βασικούς λόγους επιλογής της συγκεκριμένης διπλωματικής ήταν η δυνατότητα που δίνεται στο σχεδιαστή του συστήματος να επηρεάσει την απόδοση, την ποιότητα και την αξιοπιστία του τελικού συστήματος χρησιμοποιώντας τη διαδικασία της βελτιστοποίησης. Το όφελος που αποκόμισε ο συγγραφέας της παρούσης διπλωματικής, από την ενασχόληση του με το πρόβλημα της εύρεσης μεγίστου σημείου MPP, ήταν η δημιουργία και η εφαρμογή δύο νέων μεθόδων που χρησιμοποιούνται για πρώτη φορά σε τέτοιου είδους προβλήματα.

Περίληψη

Η συγκεκριμένη διατριβή πραγματεύεται την ανάπτυξη αλγορίθμου για την εύρεση του ολικού σημείου μέγιστης ισχύος, σε φωτοβολταϊκή συστοιχία υπό συνθήκες μερικής σκίασης. Αρχικά, έγινε μια μελέτη και καταγραφή των υφιστάμενων αλγορίθμων που χρησιμοποιούνται στο συγκεκριμένο τομέα και καταγράφηκαν τα πλεονεκτήματα και τα μειονεκτήματά τους. Στη συνέχεια επιλέχθηκαν προς υλοποίηση δύο αλγόριθμοι. Ο πρώτος αλγόριθμος που επιλέχθηκε είναι ο P&O λόγω της δημοτικότητας που έχει αποκτήσει και λόγω της απλότητας που παρουσιάζει η υλοποίησή του. Ο δεύτερος αλγόριθμος που επιλέχθηκε είναι ο PSO, ο οποίος εμφανίζει υψηλή απόδοση, μικρό χρόνο σύγκλισης και εμφανίζει μεγάλη δημοτικότητα. Οι συγκεκριμένοι δύο αλγόριθμοι χρησιμοποιήθηκαν ως βάση για τη δημιουργία τριών νέων αλγορίθμων. Το σύνολο των αλγορίθμων που υλοποιήθηκαν εκτελέστηκαν σε εννέα διαφορετικά μοντέλα μερικής σκίασης και πραγματοποιήθηκε σύγκριση των αποτελεσμάτων τους, με σκοπό την επιλογή του βέλτιστου αλγορίθμου για την εφαρμογή του σε ένα πραγματικό σύστημα.

Ο πρώτος αλγόριθμος που δημιουργήθηκε ονομάζεται PSO Advanced και αποτελεί παραλλαγή του κλασικού αλγορίθμου PSO. Η διαφορά τους εντοπίζεται στον τρόπο ανανέωσης των σωματιδίων που κινούνται στο χώρο των λύσεων. Μια επιπλέον διαφορά του με τον PSO, είναι ότι ο PSO Advanced εφαρμόζει τη μέθοδο της διαγραφής λύσεων με χαμηλό ποσοστό απόδοσης. Οι παραπάνω διαφορές οδήγησαν τον PSO Advanced σε ταχύτερη σύγκλιση από τον PSO, εμφανίζοντας ταυτόχρονα τον ίδιο υψηλό βαθμό απόδοσης.

Ο δεύτερος αλγόριθμος που υλοποιήθηκε, ονομάζεται NDBE και αποτελεί παραλλαγή της DE. Θεωρείται γενετικός αλγόριθμος και χρησιμοποιεί ένα διαφορετικό τελεστή μετάλλαξης από αυτό της DE. Χάρη στο διαφορετικό αυτό τελεστή μετάλλαξης, η αρχική λύση ευνοείται στις πρώτες γενιές, αυξάνοντας το ρυθμό σύγκλισης, αλλά σε επόμενες γενιές η βέλτιστη λύση λαμβάνει μεγαλύτερη βαρύτητα, αποφεύγοντας την παγίδευση σε τοπικά μέγιστα. Έτσι, στα αρχικά στάδια του αλγορίθμου ευνοείται η εξερεύνηση, ενώ καθώς τα χρωμοσώματα συγκλίνουν σε μια λύση ευνοείται η εκμετάλλευση. Ο συγκεκριμένος αλγόριθμος εμφανίζει το μεγαλύτερο ποσοστό ακρίβειας και απόδοσης, αλλά αντίθετα απαιτείται περισσότερος χρόνος για τον εντοπισμό της βέλτιστης λύσης.

Ο τρίτος αλγόριθμος που υλοποιήθηκε ο AIA και βασίζεται στον κλασικό αλγόριθμο ACO. Τροποποιώντας κάποια από τα τμήματα του ACO, δημιουργήθηκε ο αλγόριθμος AIA, ο οποίος παρουσίασε αξιολογικά αποτελέσματα. Εμφανίζει παρόμοιο χρόνο σύγκλισης με αυτόν του PSO ενώ ταυτόχρονα παρουσιάζει μεγαλύτερη απόδοση.

Από τη σύγκριση των αλγορίθμων που υλοποιήθηκαν, ο αλγόριθμος NDBE και ο AIA εμφανίζουν αξιολογικά αποτελέσματα και ενδιαφέρουσα συμπεριφορά. Συνεπώς, συνίσταται η τοποθέτησή τους σε σύστημα το οποίο λειτουργεί κάτω από πραγματικές συνθήκες για να ολοκληρωθεί ο έλεγχος της λειτουργικότητας και της αξιοπιστίας τους.

Maximum power point tracking algorithm implementation for PV systems under partial shading conditions

«Ioannis Katsaronas»

Abstract

The present thesis deals with the development of algorithms for finding the total maximum power point (MPP) in a photovoltaic array under partial shading conditions. Initially, a study of existing algorithms used in this field was made and the goal of this analysis was to identify the advantages and disadvantages of each algorithm. After the analysis was made, two algorithms were selected for implementation. The first one was P&O due to its popularity and the simplicity of its implementation and the second algorithm chosen was PSO. PSO was selected due to its high performance, short convergence time, and also its popularity. These two algorithms were used as a basis for the creation of three new algorithms. The set of algorithms implemented was executed on nine different partial shading experiments and a comparison of their results was carried out in order to select the best algorithm/s to be applied in to a real system.

The first algorithm created is called PSO Advanced and is a variant of the classical PSO algorithm. Their difference lies in the way of updating the particle's position in the solution space during each generation. Additionally, PSO advanced has utilized a new methodology that is the deletion of the solutions which score the lowest performance. The above differences led PSO Advanced to faster convergence than PSO while producing the same level of performance.

The second algorithm implemented is called NBDE and it is a variant of DE. It is considered a genetic algorithm and it uses a different mutation operator from DE. This mutation operator favors the initial solution in the first generations by increasing the convergence rate, but in subsequent next generations, the optimal solution receives more weight, avoiding getting trapped in local optima. Thus, in the early generations, exploration is favored, and while chromosomes converge to a solution, exploitation is favored. As a result, NBDE achieved the performance but on the contrary, it needs more time to locate the optimal solution.

The third algorithm implemented is called AIA and is based on the classical ACO algorithm. AIA was created by modifying some of the parts of ACO and the results were quite impressive. The convergence time was similar to that of PSO while the performance was similar to that of NBDE.

Having developed these five algorithms, a comparison of their performance has been carried out. As a result, NBDE and AIA algorithms showed promising results and interesting behavior. Therefore, it is recommended to be placed in a real world operating system and tests their performance under various environmental conditions in order to complete the functionality and reliability testing. Only after comparing real word data with simulation data, one can extract a reliable verdict regarding the performance of these two new algorithm implementations in the field of MPP tracking.

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον καθηγητή του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΠΜΣ Εφαρμοσμένα Ηλεκτρονικά Συστήματα, κ. Ιορδάνη Κιοσκερίδη, για την εμπιστοσύνη της ανάθεσης και την επίβλεψη της παρούσης διπλωματικής εργασίας. Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου και ιδιαίτερα την σύζυγο μου, Άννα, για τη στήριξη, την εμπιστοσύνη και την υπομονή τους.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Εικόνων	xii
Κατάλογος Πινάκων.....	xv
Συντομογραφίες.....	xvii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Πρόλογος.....	1
1.2 Σύγκριση μεθόδων για εντοπισμό σημείο MPP	4
1.3 Συγκριτικό τεχνικών εύρεσης σημείου MPP.....	4
1.4 Προσομοίωση και μοντελοποίηση συστήματος.....	5
1.5 PID-based MPPT ελεγκτής	7
1.6 FL-based MPPT ελεγκτής	8
1.7 ANN-based MPPT ελεγκτής	9
1.8 GA-based MPPT ελεγκτής.....	12
1.8.1 Γενικές πληροφορίες για τους γενετικούς αλγορίθμους.....	12
1.8.2 Εφαρμογή γενετικών αλγορίθμων στο πρόβλημα MPPT	12
1.9 PSO-based MPPT ελεγκτής	14
1.10 Αποτελέσματα προσομοιώσεων.....	15
1.10.1 Σύγκριση βάση απόδοσης	15
1.10.2 Απόκριση λόγω περιβαλλοντικών αλλαγών.....	16
1.10.3 Απαιτήσεις υλισμικού – λογισμικού	17
1.11 Συμπεράσματα.....	17
Κεφάλαιο 2ο: Αλγόριθμος P&O	19
2.1 Εισαγωγή.....	19
2.2 Ταλαντώσεις σε σταθερή κατάσταση (σημείο MPP).....	19
2.3 Απόκλιση από το σημείο MPP	19
2.4 Αδυναμία εύρεσης βέλτιστου σημείου MPP υπό συνθήκες μερικής σκίασης.....	20
2.5 Διάγραμμα ροής αλγορίθμου P&O	22
2.6 Ψευδό – κώδικας P&O	24

Κεφάλαιο 3ο: Αλγόριθμος PSO	25
3.1 Βασικές αρχές συλλογικής συμπεριφοράς	25
3.2 Ο βασικός αλγόριθμος PSO	26
3.3 Αλγόριθμος PSO για MPPT	28
3.3.1 Μοντελοποίηση του προβλήματος	28
3.3.2 Στάδια Αλγορίθμου	30
3.3.3 Αλγόριθμος PSO_advanced	33
3.3.4 Βάρος συντελεστή αδράνειας	33
3.3.5 Απόρριψη χειρότερης λύσης	35
3.3.6 Στάδια Αλγορίθμου	35
3.3.7 Ψευδό-κώδικας αλγορίθμων	36
Κεφάλαιο 4ο: Αλγόριθμος DE	38
4.1 Εισαγωγή	38
4.2 Αλγόριθμος DE για MPPT	40
4.2.1 Διαφορική εξέλιξη (DE) για MPP	41
4.2.2 Πειραματισμοί	41
4.3 NBDE	44
4.3.1 Εισαγωγή	44
4.3.2 Κωδικοποίηση χρωμοσωμάτων	44
4.3.3 Μετάλλαξη, διασταύρωση και συνάρτηση κόστους	45
4.3.4 Υλοποίηση διωνυμικής διασταύρωσης	46
4.3.5 Διαδικασία επιλογής	48
4.3.6 Διάγραμμα ροής	49
4.3.7 Ψευδό – κώδικας	50
4.3.8 Πίνακας τιμών που χρησιμοποιήθηκαν	50
Κεφάλαιο 5ο: Ant Inspired Algorithm	52
5.1 Εισαγωγή	52
5.2 Σύλληψη ιδέας για δημιουργία αλγορίθμου	52
5.3 Αρχικοποίηση αλγορίθμου	52
5.4 Ανανέωση θέσης ατόμων πληθυσμού	56
5.5 Συντελεστής μετακίνησης	56
5.6 Levy Random Walks	56
5.7 Διαχείριση και απόρριψη λύσεων	58
5.8 Αλγόριθμος ΑΙΑ	58
5.9 Διάγραμμα ροής	60

5.10	Ψευδό – κώδικας	61
5.11	Πίνακας τιμών που χρησιμοποιήθηκαν	62
Κεφάλαιο 6ο: Πειραματική διαδικασία.....		63
6.1	Εισαγωγή.....	63
6.2	Γραφικό περιβάλλον (GUI).....	66
6.3	Πείραμα 1 ^ο	70
6.4	Πείραμα 2 ^ο	77
6.5	Πείραμα 3 ^ο	83
6.6	Πείραμα 4 ^ο	90
6.7	Πείραμα 5 ^ο	95
6.8	Πείραμα 6 ^ο	100
6.9	Πείραμα 7 ^ο	106
6.10	Πείραμα 8ο.....	112
6.11	Πείραμα 9ο.....	118
Κεφάλαιο 7ο: Αποτελέσματα - Συμπεράσματα.....		124
7.1	Απόδοση.....	125
7.2	Ταχύτητα σύγκλισης.....	126
7.3	Ποσοστό επιτυχούς σύγκλισης.....	128
7.4	Σχόλια.....	129
Κεφάλαιο 8ο: Αναφορές.....		131
Κεφάλαιο 9ο: ΠΑΡΑΡΤΗΜΑ		133
9.1	ΠΑΡΑΡΤΗΜΑ Α: Αλγόριθμος P&O.....	133
9.2	ΠΑΡΑΡΤΗΜΑ Β: Αλγόριθμος PSO	134
9.3	ΠΑΡΑΡΤΗΜΑ Γ: Αλγόριθμος PSO Advanced	138
9.4	ΠΑΡΑΡΤΗΜΑ Δ: Αλγόριθμος NBDE	142
9.5	ΠΑΡΑΡΤΗΜΑ Ε: Αλγόριθμος ΑΙΑ	148
9.6	ΠΑΡΑΡΤΗΜΑ ΣΤ: GUI	154

Κατάλογος Εικόνων

Εικόνα 1.1. Μεταβολή ρεύματος εξόδου I_o και της ισχύος εξόδου P_o ως προς την τάση εξόδου V_o ...	1
Εικόνα 1.2. Επίδραση ακτινοβολίας και θερμοκρασίας στις χαρακτηριστικές των φωτοβολταϊκών πάνελ, [1]	2
Εικόνα 1.3. Λειτουργία φωτοβολταϊκού πάνελ παρουσία (a) μηδενικής σκίασης, (b) μερικής σκίασης, (c) χαρακτηριστικές καμπύλες PV και IV για (a) και (b).....	3
Εικόνα 1.4. Μοντελοποίηση συστήματος [2].....	5
Εικόνα 1.5. Επίπτωση της θερμοκρασίας στην χαρακτηριστική $P - V$ ενός πάνελ [2].....	6
Εικόνα 1.6. Επίπτωση της θερμοκρασίας στην χαρακτηριστική $I - V$ ενός πάνελ [2].....	6
Εικόνα 1.7. Επίπτωση της ακτινοβολίας στην χαρακτηριστική $P - V$ ενός πάνελ [2]	6
Εικόνα 1.8. Επίπτωση της θερμοκρασίας στην χαρακτηριστική $I - V$ ενός πάνελ [2].....	6
Εικόνα 1.9. Σημείο MPP σύμφωνα με θερμοκρασία και ακτινοβολία [2].....	7
Εικόνα 1.10. Μπλοκ διάγραμμα ελεγκτή PID για χρήση MPPT [2].....	7
Εικόνα 1.11. Συναρτήσεις συμμετοχής μεταβλητής 'E' [2].....	8
Εικόνα 1.12. Συναρτήσεις συμμετοχής μεταβλητής 'dE' [2].....	9
Εικόνα 1.13. Ασαφείς κανόνες, πηγή [2]	9
Εικόνα 1.14. Γλωσσικές μεταβλητές που χρησιμοποιήθηκαν [2].....	9
Εικόνα 1.15. Δομή τεχνητού νευρωνικού δικτύου [2]	10
Εικόνα 1.16. Σετ δεδομένων που χρησιμοποιήθηκε για εκπαίδευση του νευρωνικού δικτύου [2]	10
Εικόνα 1.17. Τιμές (a) ακτινοβολίας, (b) θερμοκρασίας, (c) Τάσης, (d) ισχύς, που χρησιμοποιήθηκαν για την παραγωγή του σετ δεδομένων για εκπαίδευση του νευρωνικού δικτύου [2]	11
Εικόνα 1.18. Μέσο σφάλμα κατά τη διάρκεια εκπαίδευσης [2]	11
Εικόνα 1.19. Δειγματοληψία με FSTS [2]	13
Εικόνα 1.20. Δειγματοληψία με VSTS [2].....	13
Εικόνα 1.21. Διάγραμμα ροής γενετικού αλγορίθμου [2].....	14
Εικόνα 1.22. Αποτέλεσμα αλγορίθμου PSO [2].....	15
Εικόνα 1.23. Περιβαλλοντικό σενάριο προσομοίωσης [2]	15
Εικόνα 1.24. Ταχύτητα σύγκλισης αλγορίθμων [2]	16
Εικόνα 1.25. Συγκριτικά αποτελέσματα προσομοίωσης [2]	16
Εικόνα 1.26. Απόδοση μεθόδων έπειτα από γήρανση των πάνελ [2]	17
Εικόνα 1.27. Σύγκριση κόστους υλοποίησης τεχνικών MPPT [2].....	17
Εικόνα 2.1. Βήμα αλγορίθμου P&O και ταλαντώσεις γύρω από το σημείο MPP [1]	19
Εικόνα 2.2. Απόκλιση από το σημείο MPP έπειτα από σταδιακή αύξηση της ακτινοβολίας [1]	20
Εικόνα 2.3. Καμπύλη ισχύος – τάσης ($P - V$) και απεικόνιση σημείων MPP [3]	21
Εικόνα 2.4. Προτεινόμενο διάγραμμα ροής αλγορίθμου P&O [1]	22
Εικόνα 2.5. Υλοποιημένο διάγραμμα ροής αλγορίθμου P&O	23
Εικόνα 2.6. Ψευδό - κώδικας για αλγόριθμο P&O	24
Εικόνα 3.1: Σωματίδια στο χώρο λύσης του προβλήματος [6]	25
Εικόνα 3.2: Επιλογή επόμενης θέσης σωματιδίων [7]	26
Εικόνα 3.3: Διάγραμμα ροής για αλγόριθμο PSO [7].....	28
Εικόνα 3.4: Περιπτώσεις μερικής σκίασης σε πάνελ [9]	29
Εικόνα 3.5: (a) Χαρακτηριστική P-V, (b) Χαρακτηριστική I-V [9]	29
Εικόνα 3.6: Φωτοβολταϊκό σύστημα για έλεγχο του αλγορίθμου PSO [9]	30
Εικόνα 3.7. Διάγραμμα ροής αλγορίθμου PSO & PSO_advanced	32
Εικόνα 3.8. Τιμές συντελεστών βάση διάστασης προβλήματος και πληθυσμού [11]	34

Εικόνα 3.9. Ψευδό-κώδικας αλγορίθμων PSO και PSO_advanced	37
Εικόνα 4.1: Δημιουργία donor διανύσματος [13]	39
Εικόνα 4.2: Ψευδοκώδικας υπολογισμού σταθεράς L	39
Εικόνα 4.3: Μπλοκ διάγραμμα DE ελεγκτή [14]	40
Εικόνα 4.4. Διάγραμμα ροής αλγορίθμου DE [14]	41
Εικόνα 4.5. Καμπύλη ρεύματος – ισχύος (P-I) [14].....	42
Εικόνα 4.6. Χρόνος σύγκλισης αλγορίθμων DE και P&O [14]	42
Εικόνα 4.7. Πίνακας τιμών ηλιακής ακτινοβολίας για 2ο πείραμα [14]	43
Εικόνα 4.8. Χαρακτηριστικές καμπύλες ισχύος – ρεύματος (P-I) [14]	43
Εικόνα 4.9. Απόκριση αλγορίθμων DE και P&O σε δυναμικές περιβαλλοντικές συνθήκες [14]	43
Εικόνα 4.10: Τοπολογία δακτυλίου στο πληθυσμό των χρωμοσωμάτων [15]	45
Εικόνα 4.11. Διάγραμμα ροής αλγορίθμου DE.....	49
Εικόνα 4.12: Ψευδοκώδικας αλγορίθμου DE	50
Εικόνα 5.1. Αρχικοποίηση πληθυσμού 3 ατόμων στο διάστημα [0, 9.5].....	53
Εικόνα 5.2. Αρχικοποίηση πληθυσμού 3 ατόμων στο διάστημα [0, 9.5] με χρήση βήματος	54
Εικόνα 5.3. Σημεία δειγματοληψίας και περιοχές μετάβασης	54
Εικόνα 5.4. Αρχικοποίηση πληθυσμού με χρήση 10 ατόμων στο διάστημα [0, 9.5].....	55
Εικόνα 5.5. Σημεία δειγματοληψίας και περιοχές μετάβασης	55
Εικόνα 5.6. Πτήση Levy σε δυσδιάστατο επίπεδο [17]	57
Εικόνα 5.7. Διάγραμμα ροής αλγορίθμου AIA	60
Εικόνα 5.8. Ψευδό - κώδικας αλγορίθμου AIA	61
Εικόνα 6.1. Καμπύλη PV (κόκκινο χρώμα), καμπύλη IV (μπλε χρώμα).....	63
Εικόνα 6.2. Καμπύλη PV (κόκκινο χρώμα), καμπύλη IV (μπλε χρώμα), μερική σκίαση	64
Εικόνα 6.3. Καμπύλη PV (κόκκινο χρώμα), καμπύλη IV (μπλε χρώμα), μερική σκίαση	64
Εικόνα 6.4. Μοντέλο προσομοίωσης στη Simulink.....	65
Εικόνα 6.5. Γραφικός περιβάλλον πειραματισμών	66
Εικόνα 6.6. Επιλογή δεδομένων για τα πειράματα 1 – 9	67
Εικόνα 6.7. Πλήκτρο εκκίνησης της προσομοίωσης	68
Εικόνα 6.8. Ένδειξη ότι η προσομοίωση βρίσκεται σε λειτουργία.....	68
Εικόνα 6.9. Εξαγωγή χαρακτηριστικών καμπυλών PV και IV	69
Εικόνα 6.10. Διαθέσιμες επιλογές αλγορίθμων.....	70
Εικόνα 6.11. Απεικόνιση πειράματος 1 στο γραφικό περιβάλλον	70
Εικόνα 6.12. Γράφημα PV, πείραμα 1	72
Εικόνα 6.13. Γράφημα IV, πείραμα 1	72
Εικόνα 6.14. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 1	73
Εικόνα 6.15. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 1	75
Εικόνα 6.16. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 1	76
Εικόνα 6.17. Γράφημα PV, πείραμα 2	78
Εικόνα 6.18. Γράφημα PV, πείραμα 2	78
Εικόνα 6.19. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 2.....	79
Εικόνα 6.20. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 2.....	81
Εικόνα 6.21. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 2.....	82
Εικόνα 6.22. Γράφημα PV, πείραμα 3	84
Εικόνα 6.23. Γράφημα IV, πείραμα 3	84

Εικόνα 6.24. Καμπύλες P,V,I (άξονας Y) σε συνάρτηση με το χρόνο της προσομοίωσης (άξονας x), πείραμα 3.....	85
Εικόνα 6.25. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 3.....	86
Εικόνα 6.26. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 3.....	88
Εικόνα 6.27. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 3.....	89
Εικόνα 6.28. Γράφημα PV, πείραμα 4.....	91
Εικόνα 6.29. Γράφημα IV, πείραμα 4.....	91
Εικόνα 6.30. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 4.....	92
Εικόνα 6.31. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 4.....	93
Εικόνα 6.32. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 4.....	94
Εικόνα 6.33. Γράφημα PV, πείραμα 5.....	96
Εικόνα 6.34. Γράφημα IV, πείραμα 5.....	96
Εικόνα 6.35. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 5.....	97
Εικόνα 6.36. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 5.....	98
Εικόνα 6.37. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 5.....	99
Εικόνα 6.38. Γράφημα PV, πείραμα 6.....	101
Εικόνα 6.39. Γράφημα IV, πείραμα 6.....	101
Εικόνα 6.40. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 6.....	102
Εικόνα 6.41. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 6.....	104
Εικόνα 6.42. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 6.....	105
Εικόνα 6.43. Γράφημα PV, πείραμα 7.....	107
Εικόνα 6.44. Γράφημα IV, πείραμα 7.....	107
Εικόνα 6.45. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 7.....	108
Εικόνα 6.46. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 7.....	110
Εικόνα 6.47. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 7.....	111
Εικόνα 6.48. Γράφημα PV, πείραμα 8.....	113
Εικόνα 6.49. Γράφημα IV, πείραμα 8.....	113
Εικόνα 6.50. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 8.....	114
Εικόνα 6.51. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 8.....	116
Εικόνα 6.52. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 8.....	117
Εικόνα 6.53. Γράφημα PV, πείραμα 9.....	119
Εικόνα 6.54. Γράφημα IV, πείραμα 9.....	119
Εικόνα 6.55. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 9.....	120
Εικόνα 6.56. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 9.....	122
Εικόνα 6.57. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 9.....	123
Εικόνα 7.1. Σύγκριση αλγορίθμων (απόδοση).....	126
Εικόνα 7.2. Χρόνοι εκτέλεσης αλγορίθμων.....	128

Κατάλογος Πινάκων

Πίνακας 2.1. Πίνακας τιμών για αλγόριθμο P&O.....	22
Πίνακας 3.1. Παράμετροι που χρησιμοποιήθηκαν για τον PSO	30
Πίνακας 3.2. Συγκεντρωτικός πίνακας παραμέτρων αλγορίθμου PSO.....	32
Πίνακας 3.3. Τιμές υπολογισμού ποσοστού σύγκλισης.....	34
Πίνακας 3.4. Παράμετροι που χρησιμοποιήθηκαν για τον PSO_advanced.....	35
Πίνακας 3.5. Επεξήγηση υλοποιημένων συναρτήσεων για PSO και PSO_advanced.....	37
Πίνακας 4.1. Επεξήγηση υλοποιημένων συναρτήσεων για PSO και PSO_advanced.....	50
Πίνακας 5.1. Πίνακας τιμών για μεταβλητές και επεξήγηση βοηθητικών συναρτήσεων αλγορίθμου AIA	62
Πίνακας 6.1. Πίνακας τιμών για πειραματικό σενάριο 1	70
Πίνακας 6.2. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 1	71
Πίνακας 6.3. Αποτελέσματα προσομοίωσης, πείραμα 1	73
Πίνακας 6.4. Πίνακας τιμών για πειραματικό σενάριο 2	77
Πίνακας 6.5. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 2.....	77
Πίνακας 6.6. Αποτελέσματα προσομοίωσης, πείραμα 2.....	79
Πίνακας 6.7. Πίνακας τιμών για πειραματικό σενάριο 3	83
Πίνακας 6.8. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 3.....	83
Πίνακας 6.9. Αποτελέσματα προσομοίωσης, πείραμα 3.....	86
Πίνακας 6.10. Πίνακας τιμών για πειραματικό σενάριο 4	90
Πίνακας 6.11. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 4.....	90
Πίνακας 6.12. Αποτελέσματα προσομοίωσης, πείραμα 4.....	91
Πίνακας 6.13. Πίνακας τιμών για πειραματικό σενάριο 5	95
Πίνακας 6.14. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 5.....	95
Πίνακας 6.15. Αποτελέσματα προσομοίωσης, πείραμα 5.....	96
Πίνακας 6.16. Πίνακας τιμών για πειραματικό σενάριο 6	100
Πίνακας 6.17. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 6.....	100
Πίνακας 6.18. Αποτελέσματα προσομοίωσης, πείραμα 6.....	102
Πίνακας 6.19. Πίνακας τιμών για πειραματικό σενάριο 7	106
Πίνακας 6.20. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 7.....	107
Πίνακας 6.21. Αποτελέσματα προσομοίωσης, πείραμα 7.....	108
Πίνακας 6.22. Πίνακας τιμών για πειραματικό σενάριο 8	112
Πίνακας 6.23. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 8.....	113
Πίνακας 6.24. Αποτελέσματα προσομοίωσης, πείραμα 8.....	114
Πίνακας 6.25. Πίνακας τιμών για πειραματικό σενάριο 9	118
Πίνακας 6.26. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 9.....	119
Πίνακας 6.27. Αποτελέσματα προσομοίωσης, πείραμα 9.....	120
Πίνακας 7.1. Μέση τιμή απόδοσης [η] αλγορίθμων, μονάδες [%].....	125
Πίνακας 7.2. Τυπική απόκλιση [σ] αλγορίθμων για τιμή απόδοσης, 10 επαναλήψεις	126
Πίνακας 7.3. Μέση τιμή χρόνου σύγκλισης [t] αλγορίθμων, μονάδες [sec]	127
Πίνακας 7.4. Τυπική απόκλιση [σ] αλγορίθμων για χρόνο σύγκλισης, 10 επαναλήψεις.....	127
Πίνακας 7.5. Ποσοστό επιτυχών συγκλίσεων για 10 επαναλήψεις	128
Πίνακας 7.6. Συγκεντρωτικός πίνακας αποτελεσμάτων	130

Συντομογραφίες

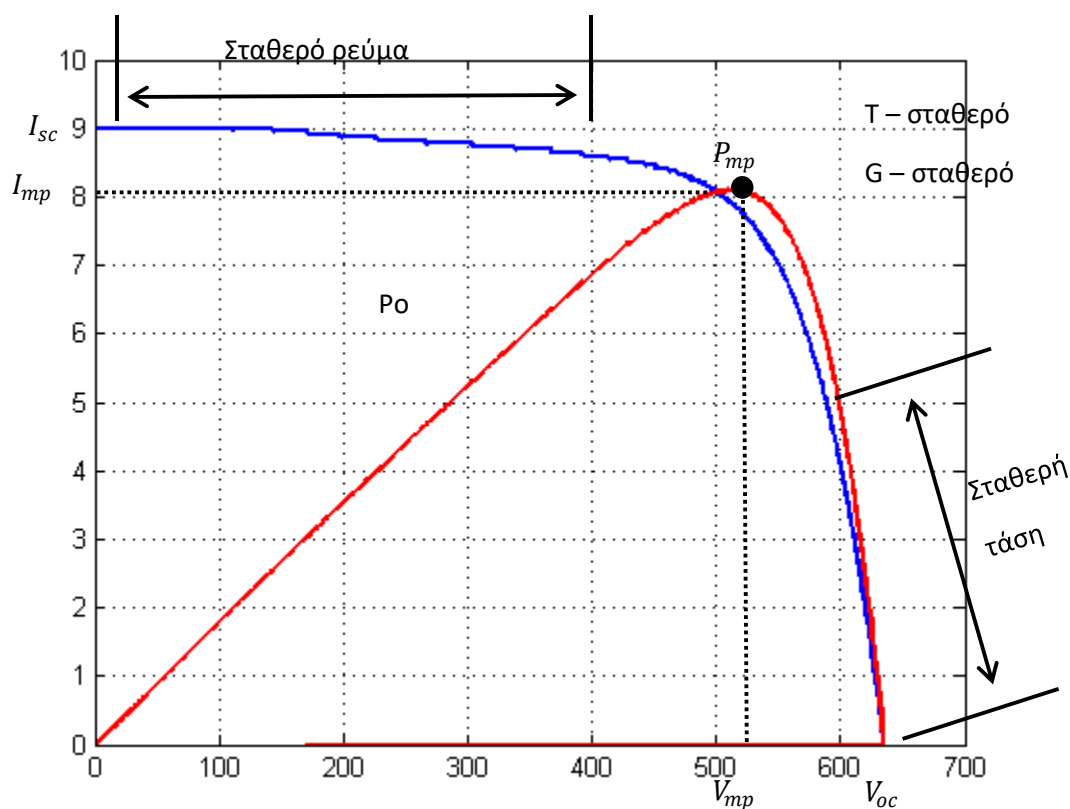
Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
MPPT	Maximum Power Point Tracking
MPP	Maximum Power Point
AIA	Ants Inspired Algorithm
DC	Direct Current
PID	Proportional Integral Derivative
FL	Fuzzy Logic
GA	Genetic Algorithm
ANN	Artificial Neural Network
PSO	Particle Swarm Optimization
P&O	Perturb & Observe
IC	Incremental Conductance
HL	Hill Climbing
ACO	Ant Colony Optimization
CS	Cuckoo Search
ABC	Artificial Bee Colony
FA	Colony of Flashing Firefly
FSTS	Fixed Sampling Time Strategy
VSTS	Variable Sampling Time Strategy
PV	Power - Voltage
IV	Current – Voltage
P	Power in Watts
V	Voltage in Volts
I	Current in Amperes
Wp	Watt peak
STC	Standard Test Conditions
GUI	Graphical User Interface

Κεφάλαιο 1ο: Εισαγωγή

1.1 Πρόλογος

Οι φωτοβολταϊκές μονάδες χρησιμοποιούνται για την άμεση μετατροπή της ηλιακής ενέργειας σε ηλεκτρική χρησιμοποιώντας το φωτοηλεκτρικό φαινόμενο, το οποίο διατυπώθηκε το 1839 από το Γάλλο φυσικό Edmund Bequerel. Μία φωτοβολταϊκή μονάδα αποτελείται από κατάλληλο αριθμό κυψελών, οι οποίες συνδέονται κατάλληλα για το σχηματισμό της απαιτούμενης τάσης και ισχύος για κάθε εφαρμογή και σχηματίζουν μια φωτοβολταϊκή διάταξη.

Η μέγιστη παραγόμενη ισχύς μιας φωτοβολταϊκής διάταξης εκφράζεται σε W_p και ορίζεται από τον εκάστοτε κατασκευαστή σε πρότυπες συνθήκες ακτινοβολίας και θερμοκρασίας (STC). Οι πρότυπες συνθήκες STC, αναφέρονται σε πυκνότητα ισχύος ηλιακής ακτινοβολίας ίση με 1 kW/m^2 και σε θερμοκρασία κυψελών 25°C . Πρακτικά σε εφαρμογές η μέγιστη ένταση της ηλιακής ακτινοβολίας θεωρείται 1000 W/m^2 και η θερμοκρασία των κυψελών κυμαίνεται από 40°C ως 60°C .

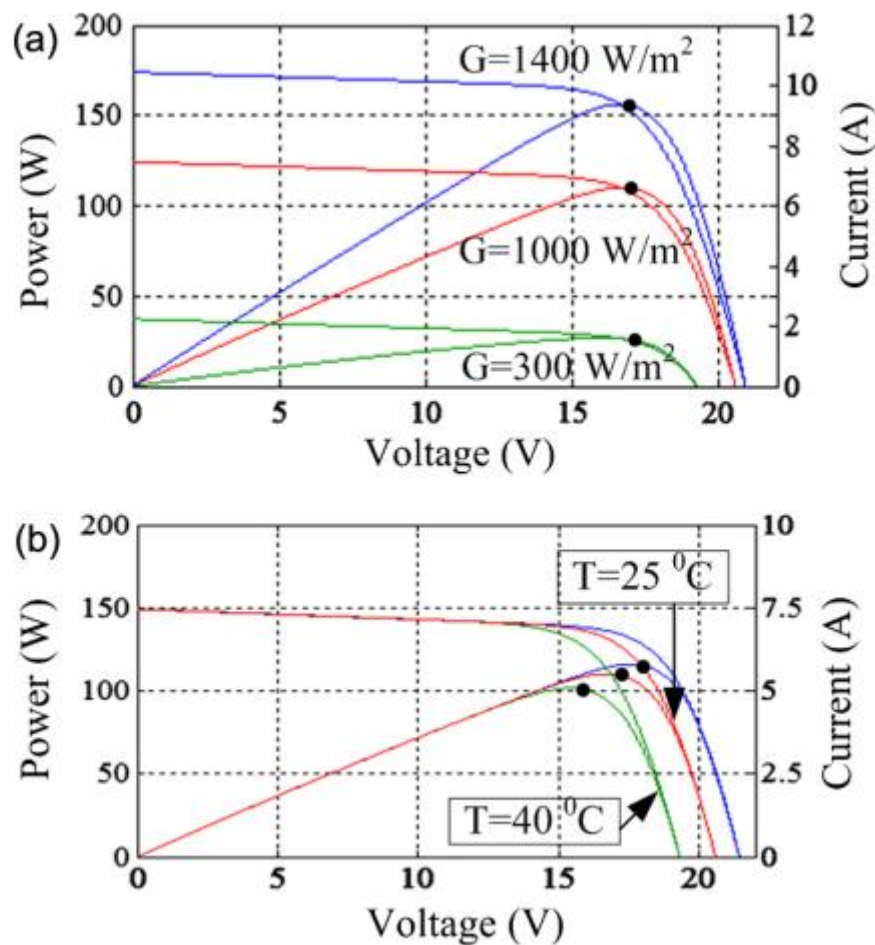


Εικόνα 1.1. Μεταβολή ρεύματος εξόδου I_o και της ισχύος εξόδου P_o ως προς την τάση εξόδου V_o

Στην εικόνα 1.1 με μπλε χρώμα απεικονίζεται η μεταβολή του ρεύματος εξόδου I_o μιας φωτοβολταϊκής μονάδας. Με κόκκινο χρώμα, απεικονίζεται η ισχύς P_o της ως προς την τάση εξόδου V_o . Από την παραπάνω χαρακτηριστική ορίζονται η τάση ανοικτού κυκλώματος V_{oc} – μέγιστη τάση εξόδου φωτοβολταϊκής διάταξης – καθώς και το ρεύμα βραχυκύκλωσης I_{sc} – μέγιστο ρεύμα εξόδου της φωτοβολταϊκής διάταξης. Η λειτουργία της φωτοβολταϊκής μονάδας, είναι δυνατή σε όλα τα σημεία

της χαρακτηριστικής και εξαρτάται από το ωμικό φορτίο R που συνδέεται στα άκρα της. Συνεπώς, για μια τιμή της αντίστασης R , υπάρχει ένα σημείο στο οποίο η ισχύς της φωτοβολταϊκής μονάδας γίνεται μέγιστη. Στο σημείο μέγιστης ισχύος P_o η τιμή του ρεύματος είναι $I_{mp} = I_{sc}$ και η τιμή της τάσης είναι $V_{mp} = V_{oc}$. Η τιμή της αντίστασης R είναι διαφορετική για διαφορετική ένταση της ηλιακής ακτινοβολίας.

Η αύξηση της θερμοκρασίας αυξάνει ελαφρά γραμμικά το ρεύμα βραχυκύκλωσης I_{sc} , ενώ ταυτόχρονα μειώνει σημαντικά την τάση ανοικτού κυκλώματος V_{oc} . Συμπερασματικά, η αύξηση της θερμοκρασίας προκαλεί μείωση της ισχύος P_{mp} των φωτοβολταϊκών μονάδων.

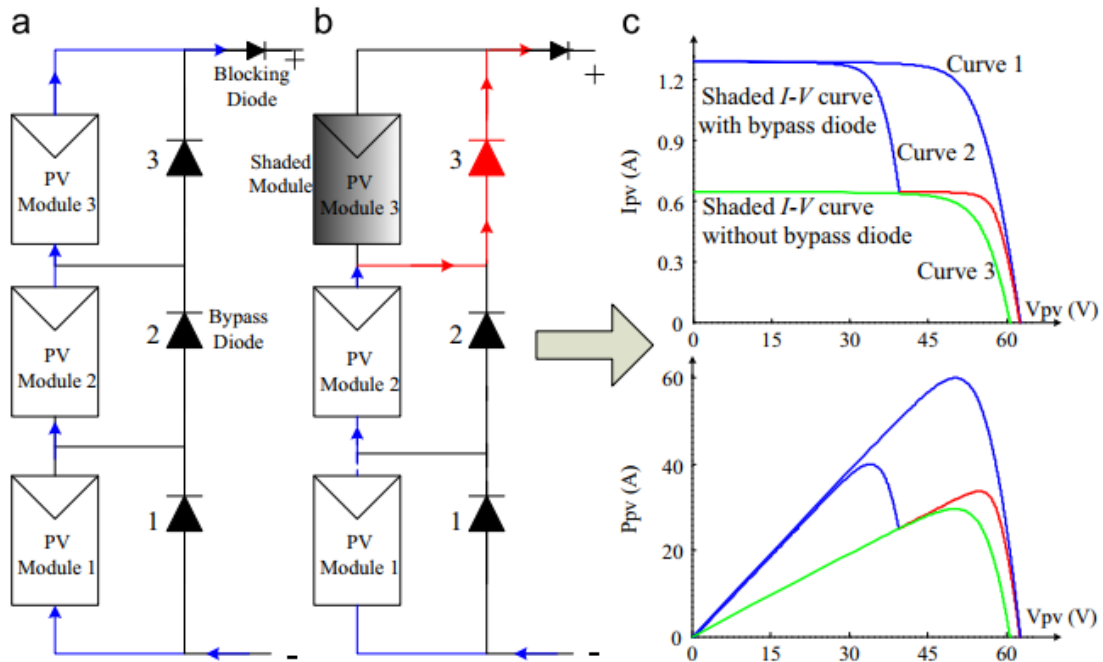


Εικόνα 1.2.Επίδραση ακτινοβολίας και θερμοκρασίας στις χαρακτηριστικές των φωτοβολταϊκών πάνελ, [1]

Η ισχύς που παράγει μια φωτοβολταϊκή μονάδα, εξαρτάται από την ένταση της ηλιακής ακτινοβολίας που προσπίπτει στο πάνελ. Συνεπώς, παρουσία μερικής σκίασης, το ρεύμα ολόκληρης της συστοιχίας θα εξαρτάται από το ρεύμα της μονάδας που σκιάζεται. Το φαινόμενο αυτό παρατηρείται, διότι οι κυψέλες εντός της μονάδας συνδέονται συνήθως σε σειρά. Το αποτέλεσμα της μερικής σκίασης είναι η σημαντικά μειωμένη απόδοση του συστήματος σε σχέση με τις περιπτώσεις ομοιόμορφης κατανομής της ηλιακής ακτινοβολίας.

Οι επιπτώσεις της σκίασης αντιμετωπίζονται πλήρως με την τοποθέτηση αντιπαράλληλης διόδου σε κάθε μονάδα – ισχύει για τις φωτοβολταϊκές κυψέλες που συνδέονται στη σειρά. Τα συγκεκριμένα

υλικά τοποθετούνται από τους κατασκευαστές στο κιβώτιο των ακροδεκτών των πάνελ. Η χρήση των διόδων επιτρέπει στις φωτοβολταϊκές μονάδες να λειτουργούν και να αποδίδουν ακόμα και αν κάποιο τμήμα των πάνελ σκιάζεται χωρίς να υπάρχει η πιθανότητα υπερθέρμανσης ή καταστροφής του. Εντούτοις, η χρήση των διόδων παράκαμψης, σε συνθήκες σκίασης, δυσκολεύει τον εντοπισμό του σημείου μέγιστης ισχύος, διότι εμφανίζονται επιμέρους τοπικά σημεία κορυφής και πρέπει να προσδιοριστεί ποιο από τα τοπικά αυτά σημεία επιφέρει τη μέγιστη ισχύ.



Εικόνα 1.3. Λειτουργία φωτοβολταϊκού πάνελ παρουσία (a) μηδενικής σκίασης, (b) μερικής σκίασης, (c) χαρακτηριστικές καμπύλες PV και IV για (a) και (b).

Η παρούσα εργασία πραγματεύεται την υλοποίηση αλγορίθμων που χρησιμοποιούνται για τον εντοπισμό του σημείου MPP (Max Power Point) και συνεπώς για την εξαγωγή της μέγιστης δυνατής ισχύος από μια φωτοβολταϊκή μονάδα κάτω από συνθήκες μερικής σκίασης. Σκοπός της εργασίας είναι η σύγκριση των πιο διαδεδομένων τεχνικών, ο εντοπισμός των πλεονεκτημάτων και των μειονεκτημάτων τους, καθώς και η επιβεβαίωση της λειτουργίας τους σε περιβάλλον προσομοίωσης. Επίσης, ένας από τους στόχους της εργασίας, ήταν η πρόταση ενός νέου αλγορίθμου και η σύγκριση του με τις υπάρχουσες μεθοδολογίες. Οι στόχοι της εργασίας επιτεύχθηκαν με επιτυχία και στα επόμενα κεφάλαια παρουσιάζεται ένας νέος αλγόριθμος για τον εντοπισμό του MPP σημείου με το όνομα AIA (Ants Inspired Algorithm). Ο νέος αλγόριθμος βασίστηκε στα πλεονεκτήματα άλλων διαδεδομένων μεθόδων. Τέλος, να αναφερθεί ότι για την υλοποίηση των αλγορίθμων, καθώς και για τους πειραματισμούς για τη διακρίβωση της λειτουργίας τους, χρησιμοποιήθηκε το περιβάλλον προσομοίωσης της Simulink.

Σκοπός της εργασίας είναι η σύγκριση των αλγορίθμων και η εξαγωγή συμπερασμάτων που αφορούν την απόδοση, την αξιοπιστία, την πολυπλοκότητα, το χρόνο εκτέλεσης και την ευκολία εφαρμογής τους σε πραγματικό σύστημα. Στα επόμενα κεφάλαια ακολουθούν λεπτομέρειες για την υλοποίηση των αλγορίθμων, το μοντέλο προσομοίωσης και για τις παραμέτρους του φωτοβολταϊκού συστήματος που χρησιμοποιήθηκε.

1.2 Σύγκριση μεθόδων για εντοπισμό σημείο MPP

Υπάρχουν εκατοντάδες άρθρα και αναφορές στο διαδίκτυο σχετικά με τον τρόπο αντιμετώπισης του φαινομένου της μερικής σκίασης στα φωτοβολταϊκά συστήματα. Το συγκεκριμένο φαινόμενο είναι ανεπιθύμητο, διότι επηρεάζει άμεσα την απόδοση του συστήματος. Εντούτοις, ο τομέας της βελτιστοποίησης της απόδοσης των φωτοβολταϊκών συστημάτων προσφέρει ένα γόνιμο έδαφος για έρευνα και ανάπτυξη μεθόδων, προσελκύοντας μεγάλο επιστημονικό ενδιαφέρον. Οδηγός για την επιλογή των υπολογιστικών μεθόδων που χρησιμοποιήθηκαν στην παρούσα εργασία, ήταν το άρθρο [2]. Το συγκεκριμένο άρθρο αποτελεί μια συγκριτική μελέτη των υπολογιστικών μεθόδων που χρησιμοποιούνται για την εύρεση του σημείου MPP σε ένα φωτοβολταϊκό σύστημα. Επίσης, παρέχει σημαντικές πληροφορίες για την απόδοση των τεχνικών που χρησιμοποιούνται σε πραγματικά συστήματα και όπως αναφέρθηκε παραπάνω, αποτέλεσε σημαντική πηγή πληροφορίας για την επιλογή και την αξιολόγηση των μεθόδων που πραγματεύεται η συγκεκριμένη εργασία. Στο επόμενο κεφάλαιο παρουσιάζεται μια σύντομη περίληψη του άρθρου, με σκοπό να παρέχει στον αναγνώστη μια σύντομη και περιεκτική επισκόπηση των διαθέσιμων τεχνικών με τα αντίστοιχα πλεονεκτήματα τους.

1.3 Συγκριτικό τεχνικών εύρεσης σημείου MPP

Στο άρθρο [2] πραγματοποιείται μια εκτενής σύγκριση των τεχνικών εύρεσης του σημείου MPP οι οποίες χρησιμοποιούν τεχνητή νοημοσύνη. Οι τεχνικές που αναλύονται και συγκρίνονται βασίζονται σε μεθόδους που χρησιμοποιούν α) PID ελεγκτή, β) ασαφή λογική (FL), γ) νευρωνικά δίκτυα (ANN), δ) γενετικούς αλγορίθμους (GA) και στ) αλγορίθμους νοημοσύνης σμήνους (PSO). Οι πειραματισμοί υλοποιήθηκαν στο ίδιο φωτοβολταϊκό σύστημα κάτω από τις ίδιες καιρικές συνθήκες. Η προσομοίωση πραγματοποιήθηκε στο περιβάλλον της Simulink και το φωτοβολταϊκό σύστημα που χρησιμοποιήθηκε περιλαμβάνει ένα DC – DC buck – boost μετατροπέα. Η αξιολόγηση των παραπάνω τεχνικών έγινε με βάση την συμπεριφορά και το κόστος της υλοποίησης των αλγορίθμων. Η συμπεριφορά περιλαμβάνει την ταχύτητα εντοπισμού του σημείου MPP, το μέσο σφάλμα εντοπισμού, τη διακύμανση γύρω από το σημείο MPP και τέλος, την απόδοση του αλγορίθμου. Η εκτίμηση του κόστους υλοποίησης έγινε βάσει των αισθητήρων και της πολυπλοκότητας του κυκλώματος που απαιτείται, καθώς και από την πολυπλοκότητα του λογισμικού.

Οι συμβατικές τεχνικές εύρεσης του MPP σημείου είναι ευρέως γνωστές και με εκτενή βιβλιογραφία. Τεχνικές όπως αυτή της διαταραχής και παρατήρησης (P&O), της αυξητικής αγωγιμότητας (IC) και της αναρρίχησης κορυφής (HL) έχουν απλή υλοποίηση, αλλά ταυτόχρονα εμφανίζουν ταλαντώσεις στο σημείο MPP. Αυτό το χαρακτηριστικό είναι ανεπιθύμητο, διότι οι ταλαντώσεις οδηγούν σε μειωμένη απόδοση του συστήματος. Έχει διεξαχθεί μεγάλη έρευνα για την ανάπτυξη μεθόδων που βελτιώνουν την απόδοση των συγκεκριμένων τεχνικών. Σύγχρονες υπολογιστικές μέθοδοι έχουν χρησιμοποιηθεί για να βελτιώσουν την απόδοση των κλασσικών αυτών μεθόδων. Νευρωνικά δίκτυα (ANN) καθώς επίσης και ασαφής λογική (FL) έχουν χρησιμοποιηθεί τόσο με τον αλγόριθμο της διαταραχής και παρατήρησης (P&O) όσο και με την αυξητική αγωγιμότητα (IC). Στο σημείο αυτό να αναφερθεί ότι ο PID, είναι ένας ελεγκτής που χρησιμοποιείται ευρέως στα συστήματα ελέγχου, λόγω της απλότητας και της αποδοτικότητας του και ως εκ τούτου, έχουν προταθεί και για το συγκεκριμένο μέθοδοι για αύξηση της απόδοσης του. Για παράδειγμα, έχουν προταθεί μέθοδοι που χρησιμοποιούν ασαφή λογική (FL) σε συνδυασμό με τον ελεγκτή, δημιουργώντας με αυτόν τον τρόπο ένα υβριδικό ελεγκτή που συνδυάζει τα πλεονεκτήματα και των δύο μεθόδων.

Το γεγονός ότι η σχέση μεταξύ της παραγόμενης ενέργειας και της τάσης εξόδου των πάνελ είναι μη γραμμική, οδηγεί στο συμπέρασμα ότι η ασαφής λογική (FL) και τα νευρωνικά δίκτυα (ANN) είναι

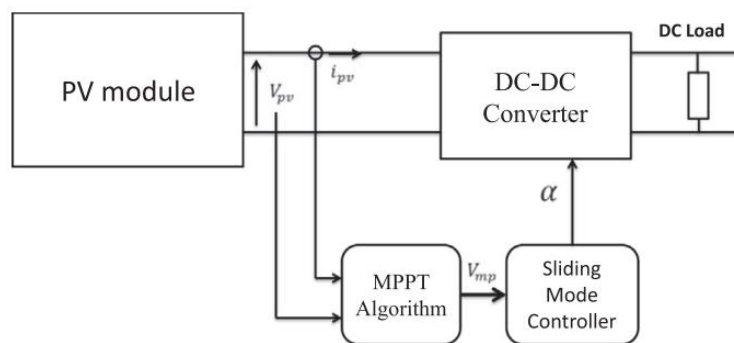
οι πιο κατάλληλες λύσεις για την εύρεση του σημείου MPP. Η χρήση της ασαφής λογικής, έχει συνδυαστεί και με γενετικούς αλγορίθμους (GA) με στόχο την αύξηση της απόδοσης του συστήματος. Από την άλλη μεριά, η χρήση των τεχνητών νευρωνικών δικτύων (ANN), απαιτεί μεγάλο σετ δεδομένων για την εκπαίδευσή τους και συχνά αυτό αποτελεί ένα περιοριστικό παράγοντα για τη χρήση τους. Ωστόσο, έχει χρησιμοποιηθεί ο συνδυασμός FL με ANN καθώς και ο συνδυασμός με GA. Οι συνδυασμοί αυτοί στοχεύουν είτε στην παραγωγή δεδομένων εκπαίδευσης για τα ANN, είτε στην απευθείας εκπαίδευσή τους.

Στις περιπτώσεις μερικής σκίασης, δημιουργούνται πολλά MPP με αποτέλεσμα ο εντοπισμός του σημείου βέλτιστου σημείου MPP να ανάγεται σε πρόβλημα βελτιστοποίησης. Τέτοιου είδους προβλήματα χρησιμοποιούν μετά-ευρετικούς ή γενετικούς αλγορίθμους. Για την επίλυση του συγκεκριμένου προβλήματος έχουν χρησιμοποιηθεί τόσο γενετικοί αλγόριθμοι, όπως η διαφορική εξέλιξη (DE), όσο και μετά-ευρετικοί όπως ο PSO. Ο PSO για παράδειγμα, ανήκει στην ομάδα αλγορίθμων βελτιστοποίησης σωματιδίων και χρησιμοποιείται συνήθως λόγω της απλότητας που παρουσιάζει η υλοποίηση του και της ταχύτητας σύγκλισης του. Κατά καιρούς έχουν χρησιμοποιηθεί και άλλοι αλγόριθμοι της ίδιας οικογενείας όπως η ACO, CS, ABC και FA.

Το συγκεκριμένο άρθρο, στοχεύει στην ανάλυση των πιο διαδομένων τεχνικών τεχνητής νοημοσύνης για την εύρεση του σημείου MPP. Αυτές οι τεχνικές παρουσιάζονται και εφαρμόζονται κάτω από τις ίδιες περιβαλλοντικές συνθήκες στο ίδιο φωτοβολταϊκό σύστημα. Στο άρθρο επίσης υλοποιήθηκε ένας έλεγχος ευρωστίας της μεθόδου MPPT σε σχέση με τα χαρακτηριστικά των πάνελ, όταν τα τελευταία υφίστανται γήρανση. Σύμφωνα με τα αποτελέσματα της προσομοίωσης και λαμβάνοντας υπόψη τις απαιτήσεις υλισμικού και λογισμικού, ακολουθεί σύγκριση όλων των μεθόδων, τονίζοντας τα πλεονεκτήματα και τα μειονεκτήματα της κάθε μιας. Στο τέλος του άρθρου ακολουθούν προτάσεις για κάθε μια μεθοδολογία από τους συγγραφείς.

1.4 Προσομοίωση και μοντελοποίηση συστήματος

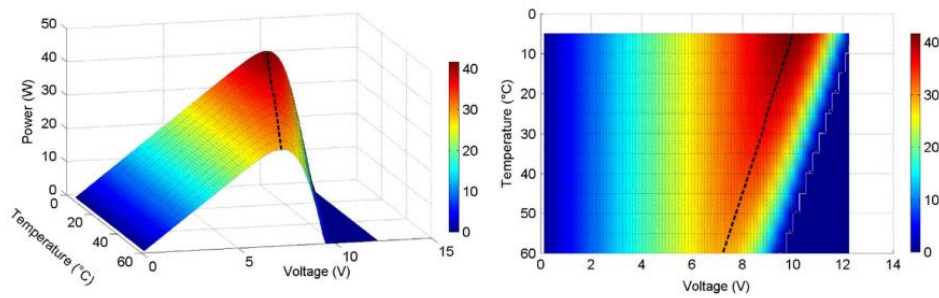
Το σύστημα που χρησιμοποιήθηκε για τις ανάγκες της προσομοίωσης του άρθρου απεικονίζεται στην παρακάτω εικόνα.



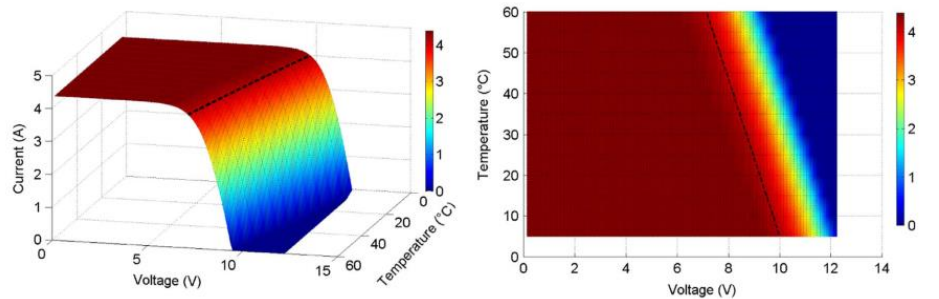
Εικόνα 1.4. Μοντελοποίηση συστήματος [2]

Αρχικά έγινε η μελέτη της απόδοσης του συστήματος κάτω από αλλαγές στη θερμοκρασία και την ηλιακή ακτινοβολία. Από τα αποτελέσματα προέκυψε ότι η τάση του σημείου MPP επηρεάζεται περισσότερο από αλλαγές στη θερμοκρασία, παρά από αλλαγές στην ηλιακή ακτινοβολία. Στο παρακάτω σχήμα, απεικονίζεται το θεωρητικό MPP για διάφορες τιμές της θερμοκρασίας και της ηλιακής ακτινοβολίας. Παρατηρήθηκε ότι η μέγιστη ισχύς είναι ανάλογη της ηλιακής ακτινοβολίας και

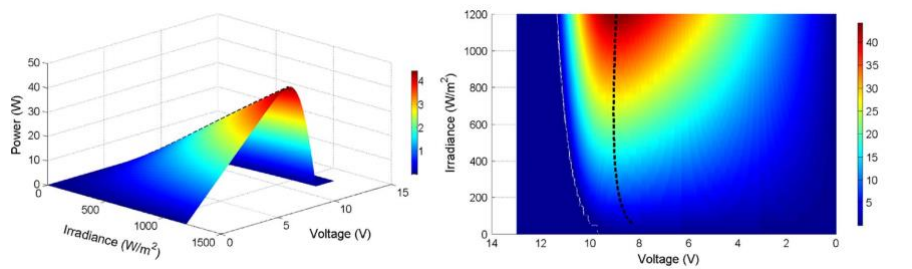
αντιστρόφως ανάλογη της θερμοκρασίας. Επίσης, παρατηρήθηκε ότι η ένταση της ηλιακής ακτινοβολίας επηρεάζει περισσότερο την ισχύ του πάνελ σε σχέση με τη θερμοκρασία.



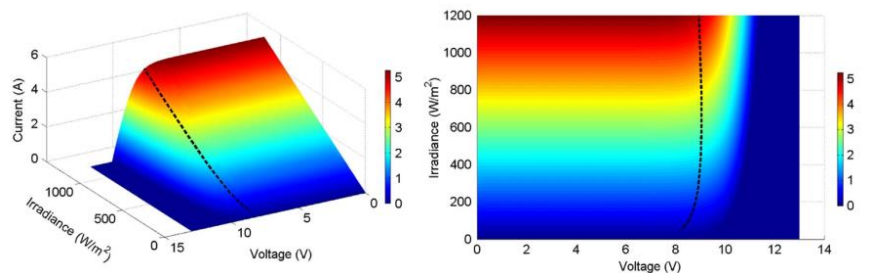
Εικόνα 1.5. Επίπτωση της θερμοκρασίας στην χαρακτηριστική $P - V$ ενός πάνελ [2]



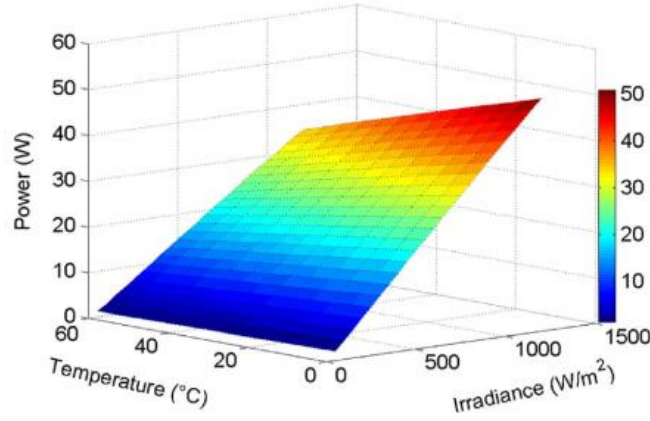
Εικόνα 1.6. Επίπτωση της θερμοκρασίας στην χαρακτηριστική $I - V$ ενός πάνελ [2]



Εικόνα 1.7. Επίπτωση της ακτινοβολίας στην χαρακτηριστική $P - V$ ενός πάνελ [2]



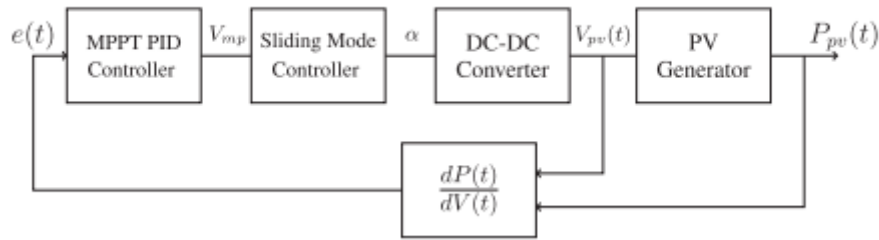
Εικόνα 1.8. Επίπτωση της θερμοκρασίας στην χαρακτηριστική $I - V$ ενός πάνελ [2]



Εικόνα 1.9. Σημείο MPP σύμφωνα με θερμοκρασία και ακτινοβολία [2]

Παρακάτω παρουσιάζονται σύντομα οι αλγόριθμοι MPPT που χρησιμοποιήθηκαν στο συγκεκριμένο paper.

1.5 PID-based MPPT ελεγκτής



Εικόνα 1.10. Μπλοκ διάγραμμα ελεγκτή PID για χρήση MPPT [2]

Η τεχνική ελέγχου διαμέσου του ελεγκτή PID είναι γνωστή για την ευελιξία, την απόδοση και τη σταθερότητα που προσφέρει. Στο συγκεκριμένο ελεγκτή χρησιμοποιείται η μεταβλητή $e(t)$, ως μεταβλητή ελέγχου.

$$e(t) = \frac{dP(t)}{dV(t)} \quad (1.1)$$

$$V_{mp}(t) = K_p(t) + K_i \int e(t)dt + K_D \frac{de(t)}{dt} \quad (1.2)$$

Με τον ελεγκτή PID, χρησιμοποιήθηκε και ένας ελεγκτής ολίσθησης ο οποίος βελτιώνει τη συμπεριφορά του κυκλώματος. Ο ελεγκτής ολίσθησης ουσιαστικά επιτρέπει στον DC – DC μετατροπέα να λειτουργεί στο σημείο MPP ανεξάρτητα από τις διακυμάνσεις του φορτίου. Από την προσομοίωση προέκυψε ότι οι τιμές του PID είναι: $K_p = 1$, $K_i = 200$, $K_D = 0$.

1.6 FL-based MPPT ελεγκτής

Η ασαφής λογική θεωρείται μία από τις πιο κατάλληλες τεχνικές για MPPT. Μερικά από τα πλεονεκτήματα που παρουσιάζει είναι το γεγονός ότι δεν απαιτεί ακριβή μαθηματική μοντελοποίηση, μπορεί να χειριστεί εισόδους με ανακρίβεια και μπορεί να εφαρμοστεί σε μη γραμμικά προβλήματα. Εντούτοις, για το σχεδιασμό του FL ελεγκτή απαιτείται πολύ καλή γνώση του αντικειμένου ελέγχου.

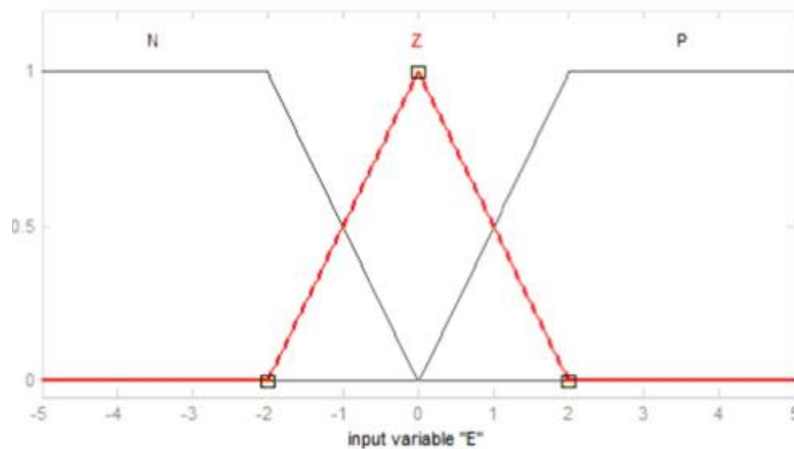
Ο ελεγκτής FL, αποτελείται από τέσσερα επιμέρους στοιχεία. Το πρώτο στάδιο αφορά την ασαφοποίηση (fuzzification), το δεύτερο αφορά τους ασαφείς κανόνες (rule base), το τρίτο αφορά το μηχανισμό συμπερασμού (inference engine) ενώ το τέταρτο στάδιο αφορά την ασαφή συνάθροιση των κανόνων (aggregation). Ο ελεγκτής FL που χρησιμοποιήθηκε χρησιμοποιεί δύο εισόδους και μια έξοδο. Οι είσοδοι του ελεγκτή είναι το σφάλμα $E(k)$ και η μεταβολή του σφάλματος $\Delta E(k)$, τα οποία υπολογίζονται από τους παρακάτω τύπους:

$$E(k) = \frac{P(k) - P(k-1)}{V(k) - V(k-1)} \quad (1.3)$$

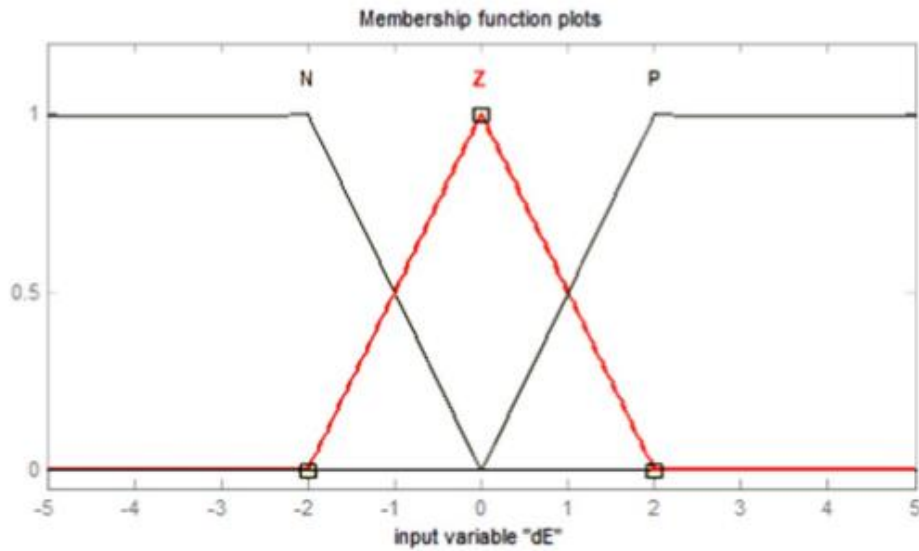
$$\Delta E(k) = E(k) - E(k-1) \quad (1.4)$$

Στο στάδιο της ασαφοποίησης, οι αριθμητικές τιμές μετατρέπονται σε γλωσσικές μεταβλητές, πχ αρνητική τιμή (N), μηδενική τιμή (Z) και θετική τιμή (P). Ο αριθμός, ο τύπος και το εύρος των συναρτήσεων συμμετοχής επιλέχθηκαν από προηγούμενη γνώση ή από πειραματικά δεδομένα. Η ακρίβεια της εξόδου, εξαρτάται από τον αριθμό των γλωσσικών μεταβλητών που χρησιμοποιήθηκαν. Ωστόσο, αυτό αυξάνει την πολυπλοκότητα του ελεγκτή καθώς και το χρόνο εκτέλεσης του αλγορίθμου. Στη συγκεκριμένη περίπτωση έχουν χρησιμοποιηθεί συνολικά εννέα συναρτήσεις συμμετοχής. Μετά το βήμα της ασαφοποίησης, υπάρχουν οι ασαφείς κανόνες, οι οποίοι εφαρμόζονται μέσω του μηχανισμού συμπερασμού (μέθοδος Mamdani), ενώ ακολουθεί το βήμα της συνάθροισης των κανόνων (Sugeno – Type), όπου εξάγεται η τάση αναφοράς του μετατροπέα ισχύος. Η τιμή της τάσης αναφοράς υπολογίζεται από τον μέσο όρο των εννέα συναρτήσεων και δίνεται από τον παρακάτω τύπο:

$$V_{ref} = \frac{\sum_{i=1}^9 \mu_i(s) s_i}{\sum_{i=1}^9 \mu_i(s)} \quad (1.5)$$



Εικόνα 1.11. Συναρτήσεις συμμετοχής μεταβλητής 'E' [2]

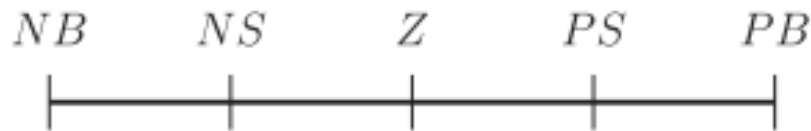


Εικόνα 1.12. Συναρτήσεις συμμετοχής μεταβλητής 'dE' [2]

Rules base table of the FL-based MPPT controller.

$\Delta E/E$	N	Z	P
N	NB	NS	Z
Z	NB	Z	PB
P	Z	PS	PB

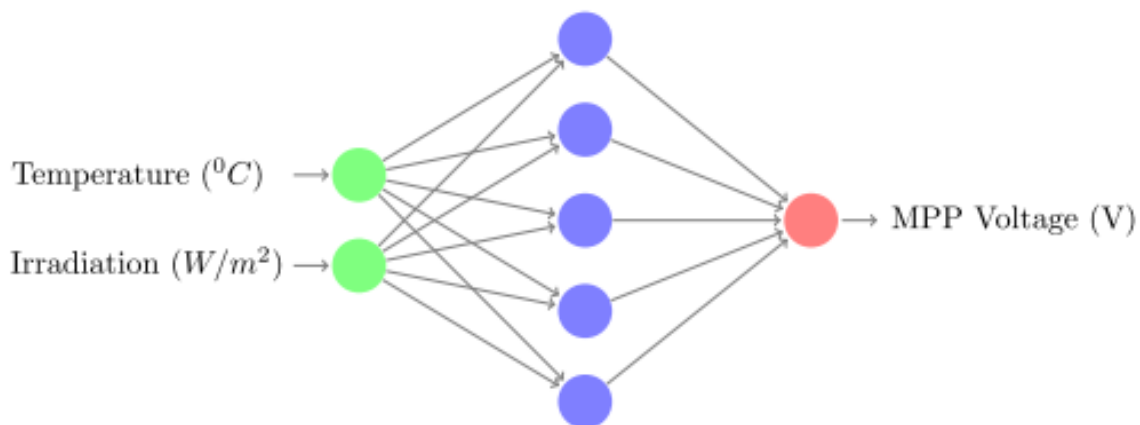
Εικόνα 1.13. Ασαφείς κανόνες, πηγή [2]



Εικόνα 1.14. Γλωσσικές μεταβλητές που χρησιμοποιήθηκαν [2]

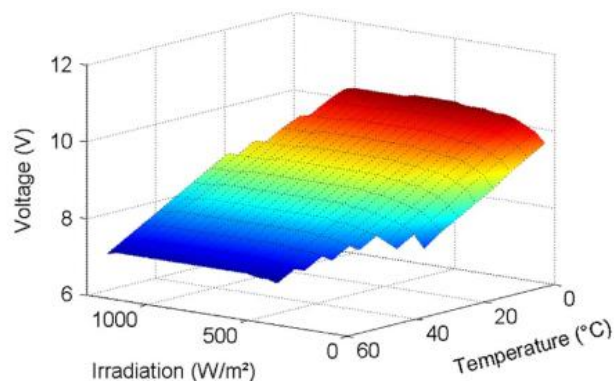
1.7 ANN-based MPPT ελεγκτής

Τα τεχνητά νευρωνικά δίκτυα αποτελούν μια εναλλακτική λύση για την επίλυση μη γραμμικών προβλημάτων. Για την εύρεση του σημείου MPP, πρέπει να δημιουργηθεί κατάλληλο σετ δεδομένων, έτσι ώστε να χρησιμοποιηθεί για την εκπαίδευση του δικτύου. Για κάθε τιμή θερμοκρασίας και ηλιακής ακτινοβολίας, το πάνελ έχει ένα μοναδικό σημείο MPP και για τον εντοπισμού του απαιτείται η εφαρμογή μιας τάσης V_{mp} στην έξοδο του φωτοβολταϊκού πάνελ. Το νευρωνικό δίκτυο που προτείνεται για τη συγκεκριμένη εφαρμογή, αποτελείται από δύο στρώματα και είναι πρόσω-τροφοδότησης.

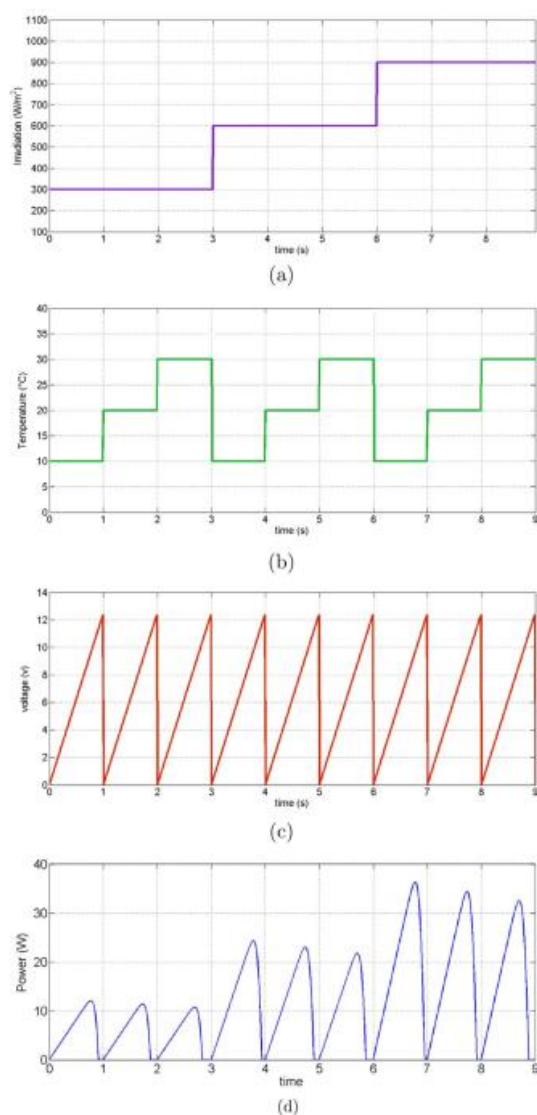


Εικόνα 1.15. Δομή τεχνητού νευρωνικού δικτύου [2]

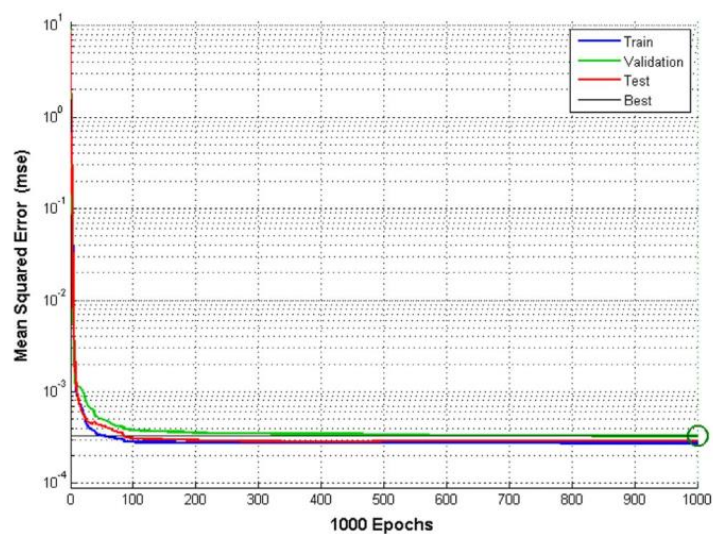
Η παραπάνω δομή του δικτύου επιλέχθηκε έπειτα από πολλές δοκιμές με σκοπό την αύξηση της απόδοσης. Το σετ δεδομένων που χρησιμοποιείται για εκπαίδευση πρέπει να περιλαμβάνει ένα μεγάλο εύρος μετρήσεων, έτσι ώστε να επιτευχθεί η απαιτούμενη ακρίβεια. Στη συγκεκριμένη περίπτωση, το σετ δεδομένων δημιουργήθηκε από προσομοίωση. Η εικόνα 1.13 απεικονίζει τη διαδικασία προσομοίωσης κατά την οποία δημιουργήθηκαν 312 δεδομένα εκπαίδευσης (τιμές θερμοκρασίας – ηλιακής ακτινοβολίας). Το 80 % των δεδομένων χρησιμοποιήθηκε για εκπαίδευση, ενώ το υπόλοιπο 20% χρησιμοποιήθηκε για επικύρωση και έλεγχο της ποιότητας εκπαίδευσης. Ο αλγόριθμος που χρησιμοποιήθηκε για εκπαίδευση είναι ο Levenberg – Marquardt back propagation. Το σφάλμα εκπαίδευσης που επιτεύχθηκε είναι 3.29×10^{-4} . Το σφάλμα κατά την επικύρωση των δεδομένων ανέρχεται στα 0.056 V, ενώ το σχετικό σφάλμα δεν είναι μεγαλύτερο από 0.78 %. Μετά την εκπαίδευση του δικτύου, το σύστημα θα μπορεί να εντοπίζει το MPP σημείο κάτω από οποιεσδήποτε καιρικές συνθήκες. Το πλεονέκτημα της συγκεκριμένης τεχνικής είναι ότι το MPP σημείο εντοπίζεται χωρίς πολλές επαναλήψεις, το οποίο ισοδυναμεί με ελάχιστες ταλαντώσεις γύρω από αυτό και αυξημένη απόδοση του ελεγκτή. Για την υλοποίηση της συγκεκριμένης τεχνικής, απαιτούνται και οι κατάλληλοι αισθητήρες, το οποίο μεταφράζεται σε αύξηση του κόστους.



Εικόνα 1.16. Σετ δεδομένων που χρησιμοποιήθηκε για εκπαίδευση του νευρωνικού δικτύου [2]



Εικόνα 1.17. Τιμές (α) ακτινοβολίας, (β) θερμοκρασίας, (γ) Τάσης, (δ) ισχύς, που χρησιμοποιήθηκαν για την παραγωγή του σετ δεδομένων για εκπαίδευση του νευρωνικού δικτύου [2]



Εικόνα 1.18. Μέσο σφάλμα κατά τη διάρκεια εκπαίδευσης [2]

1.8 GA-based MPPT ελεγκτής

1.8.1 Γενικές πληροφορίες για τους γενετικούς αλγορίθμους

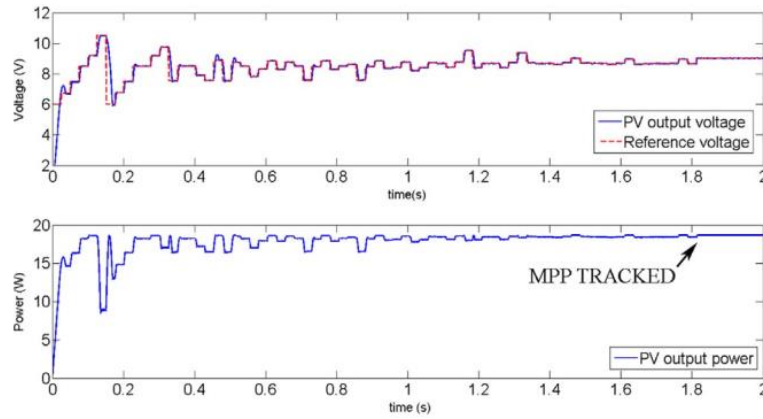
Οι γενετικοί αλγόριθμοι (GA) είναι μια τεχνική βελτιστοποίησης, η οποία βασίζεται στην εξέλιξη των βιολογικών συστημάτων. Η εκτέλεση των συγκεκριμένων αλγορίθμων αποτελείται από τα παρακάτω βήματα:

1. Κατασκευή συνάρτησης κόστους για το κάθε ένα πρόβλημα.
2. Δημιουργία αρχικού πληθυσμού ατόμων.
3. Αξιολόγηση των ατόμων του πληθυσμού με βάση τη συνάρτηση κόστους.
4. Δημιουργία νέου πληθυσμού χρησιμοποιώντας μεθόδους επιλογής, διασταύρωσης, μετάλλαξης.
 - a. Επιλογή ατόμων που θα συμμετέχουν στη δημιουργία νέου πληθυσμού.
 - b. Εφαρμογή τεχνικής διασταύρωσης για δημιουργία νέων ατόμων.
 - c. Εφαρμογή τεχνικής μετάλλαξης κατά την οποία γίνεται μια τυχαία αλλαγή στο νέο άτομο με πολύ μικρή πιθανότητα.
5. Τερματισμός αλγορίθμου σε περίπτωση ικανοποίησης των κριτηρίων τερματισμού.

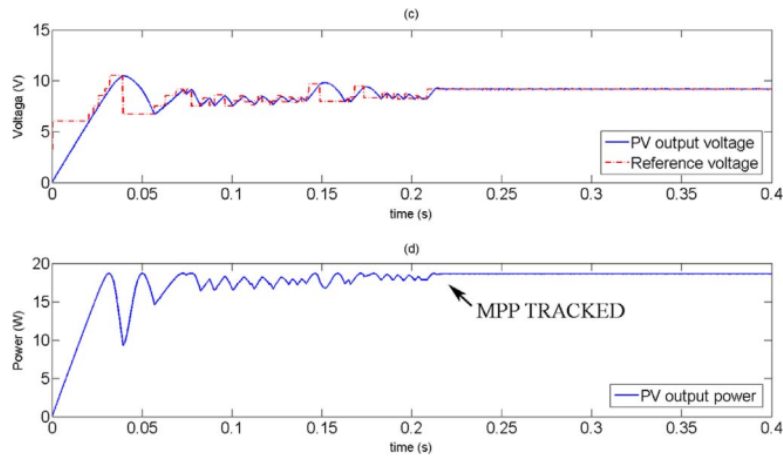
1.8.2 Εφαρμογή γενετικών αλγορίθμων στο πρόβλημα MPPT

Στόχος της μεθόδου είναι η βελτιστοποίηση της εξαγόμενης ισχύος από τα φωτοβολταϊκά πάνελ. Στο συγκεκριμένο πρόβλημα θεωρήθηκε ένας αρχικός πληθυσμός με συνολικά έξι άτομα [$V_1, V_2, V_3, V_4, V_5, V_6$]. Κάθε άτομο του πληθυσμού αντιπροσωπεύει και μια τιμή τάσης αναφοράς για τον DC – DC μετατροπέα. Αρχικά, κάθε ένα άτομο του πληθυσμού «εκπέμπει» την τάση του στον μετατροπέα και το σύστημα αξιολογεί την ισχύ για κάθε μια τάση αναφοράς.

Στη βιβλιογραφία αναφέρονται δύο τρόποι δειγματοληψίας. Ο πρώτος τρόπος είναι ο FSTS και ο VSTS. Ο πιο διαδεδομένος είναι ο FSTS, όπου ο χρόνος δειγματοληψίας είναι σταθερός. Η τεχνική αυτή παρέχει μια καλή ενημέρωση για την ισχύ του συστήματος, ωστόσο το μειονέκτημα της είναι ότι η σταθερή κατάσταση του συστήματος (σημείο MPP) μπορεί να επέλθει σε λιγότερο χρόνο από το χρόνο δειγματοληψίας. Αυτό έχει ως αποτέλεσμα, ο ανεκμετάλλετος χρόνος να δράσει αρνητικά ως προς την ταχύτητα εύρεσης του σημείου MPP. Για την αποφυγή του συγκεκριμένου προβλήματος, προτείνεται η χρήση της μεθόδου VSTS. Σε αυτήν την μέθοδο, ο χρόνος δειγματοληψίας αλλάζει σύμφωνα με το εύρος ζώνης (duty cycle) της διαταραχής που επιβάλλει ο μετατροπέας. Στο συγκεκριμένο paper η μεθοδολογία που χρησιμοποιείται είναι η VSTS.



Εικόνα 1.19. Δειγματοληψία με FSTS [2]



Εικόνα 1.20. Δειγματοληψία με VSTS [2]

Από τις εικόνες 1.16 και 1.17 προκύπτει ότι με την μέθοδο FSTS ο χρόνος εντοπισμού του σημείου MPP είναι περίπου 1.83 sec, ενώ με την μέθοδο ASTS ο χρόνος μειώθηκε σε 0.23 sec.

Στη συνέχεια, αφού έχει ολοκληρωθεί η διαδικασία της αξιολόγησης όλου του αρχικού πληθυσμού, λαμβάνει μέρος η τεχνική της διασταύρωσης. Δύο άτομα του πληθυσμού, διασταυρώνονται έτσι ώστε να δημιουργήσουν ένα νέο απόγονο. Η διασταύρωση γίνεται σύμφωνα με τον παρακάτω τύπο:

$$V_i(k) = rand(1)V_i(k-1) + (1 - rand(1))V_j(k-1) \quad (1.6)$$

Η τεχνική της μετάλλαξης λαμβάνει μέρος στη συνέχεια με πολύ μικρή πιθανότητα (5%).

$$V_i(k) = V_{min} + rand(1)(V_{max} - V_{min}) \quad (1.7)$$

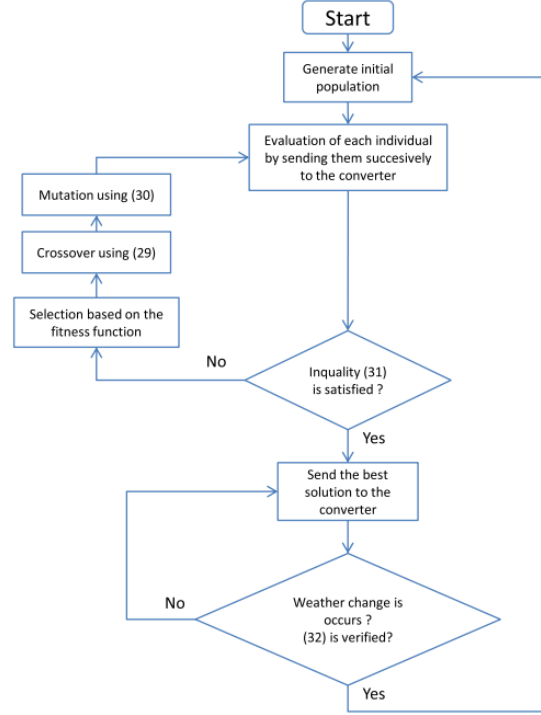
Έπειτα γίνεται έλεγχος εύρεσης του σημείου MPP, όπου ο πληθυσμός πρέπει να έχει συγκλίνει σε μια λύση. Στο σημείο MPP και για να μειωθούν οι ταλαντώσεις λόγω της μετάλλαξης, εφαρμόζεται η παρακάτω σχέση:

$$\begin{cases} |\Delta P| \leq \Delta P_{min} = 0.1 \text{ W} \\ |\Delta V| \leq \Delta V_{min} = 0.05 \text{ V} \end{cases} \quad (1.8)$$

Στην περίπτωση που υπάρχουν περιβαλλοντικές αλλαγές, θα υπάρξουν αλλαγές και στην ισχύ των φωτοβολταϊκών πάνελ. Στο σημείο αυτό λαμβάνεται η απόφαση επανεκκίνησης του αλγορίθμου.

$$\begin{cases} \frac{|P_{i+1} - P_i|}{P_i} \geq \Delta P_{set} \\ |V_{i+1} - V_i| \leq \Delta V_{set} \end{cases} \quad (1.9)$$

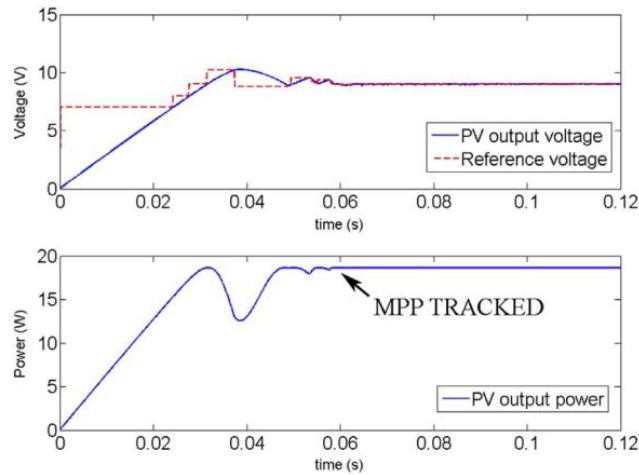
Στο παρακάτω διάγραμμα ροής, απεικονίζονται τα βήματα εκτέλεσης του αλγορίθμου.



Εικόνα 1.21. Διάγραμμα ροής γενετικού αλγορίθμου [2]

1.9 PSO-based MPPT ελεγκτής

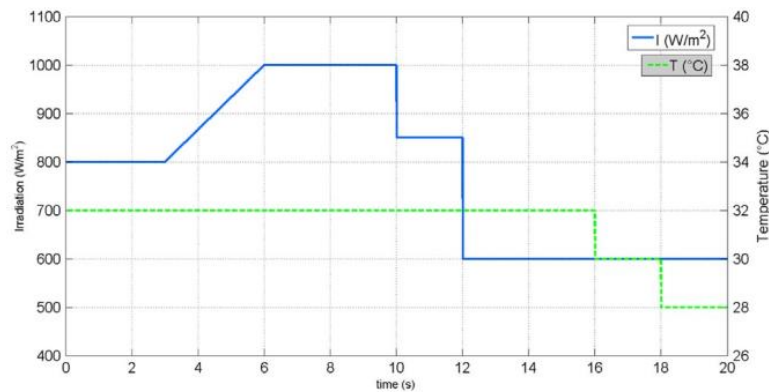
Στο συγκεκριμένο τμήμα του άρθρου γίνεται ανάλυση του αλγορίθμου του PSO. Ο συγκεκριμένος αλγόριθμος αναλύεται στα παρακάτω κεφάλαια και δεν θα αναλυθεί περισσότερο για οικονομία χώρου και χρόνου. Εντούτοις, αναφέρονται μόνο οι διαφορές στις μεταβλητές και στις ρυθμίσεις που χρησιμοποιήθηκαν. Αρχικά, η τεχνική δειγματοληψίας που χρησιμοποιήθηκε είναι η ASTS και αυτό έχει ως αποτέλεσμα την πολύ γρήγορη σύγκλιση του αλγορίθμου. Συνολικά χρησιμοποιήθηκαν τέσσερα σωματίδια και οι παράμετροι του αλγορίθμου που αφορούν την εξερεύνηση, την εκμετάλλευση και την ταχύτητα των σωματιδίων είναι, $w = 0.4$, $c_1 = 0.2$, $c_2 = 0.9$. Από την προσομοίωση προέκυψε το συμπέρασμα ότι το MPP σε φυσιολογικές συνθήκες εντοπίστηκε σε λιγότερο από 0.06 sec.



Εικόνα 1.22. Αποτέλεσμα αλγορίθμου PSO [2]

1.10 Αποτελέσματα προσομοιώσεων

Το περιβαλλοντικό σενάριο που χρησιμοποιήθηκε για την αξιολόγηση των μεθόδων απεικονίζεται στην παρακάτω εικόνα. Η διάρκεια της προσομοίωσης ήταν 20 sec με μεταβλητή ηλιακή ακτινοβολία και θερμοκρασία.



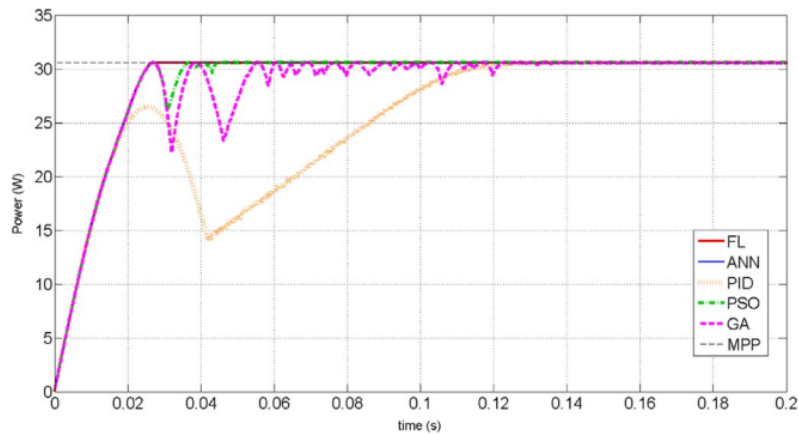
Εικόνα 1.23. Περιβαλλοντικό σενάριο προσομοίωσης [2]

Από τα αποτελέσματα προσομοίωσης προέκυψε το συμπέρασμα ότι όλες οι μέθοδοι που χρησιμοποιήθηκαν εντοπίζουν με επιτυχία το σημείο MPP. Συνολικά έγιναν τρεις διαφορετικές συγκρίσεις, με βάση την απόδοση των μεθόδων, με βάση την απόκριση λόγω περιβαλλοντικών αλλαγών και με βάση τις απαιτήσεις και το ολικό κόστος του συστήματος.

1.10.1 Σύγκριση βάση απόδοσης

Σύμφωνα με το εικόνα 1.21 προκύπτει το συμπέρασμα ότι όλες οι μέθοδοι έχουν γρήγορη απόκριση και εντοπίζουν το MPP σε λιγότερο από 140 ms. Επίσης, κατά την αξιολόγηση υπολογίστηκε το σφάλμα εντοπισμού $E_m(t)$, η διακύμανση V_m στο σημείο MPP και η απόδοση η του MPPT αλγορίθμου. Ο παρακάτω πίνακας περιέχει όλες τις παραπάνω μετρικές. Από τον πίνακα προκύπτει ότι η μέθοδος FL και ANN εμφανίζουν τα καλύτερα αποτελέσματα, εντοπίζοντας το MPP γρήγορα με ακρίβεια και πολύ

μικρές ταλαντώσεις. Για τους GA και PSO απαιτείται αρχικοποίηση του αλγορίθμου παρουσία περιβαλλοντικών αλλαγών, το οποίο μεταφράζεται σε αυξημένο χρόνο υπολογισμού. Για αυτό το λόγο εμφανίζουν μικρότερη απόδοση από τους FL και ANN. Μεταξύ των GA και PSO, ο PSO εμφανίζει μεγαλύτερη ακρίβεια και ταχύτητα εντοπισμού. Ο PID, εμφανίζει μεγάλη ακρίβεια και απόδοση, αλλά έχει αργή απόκριση.



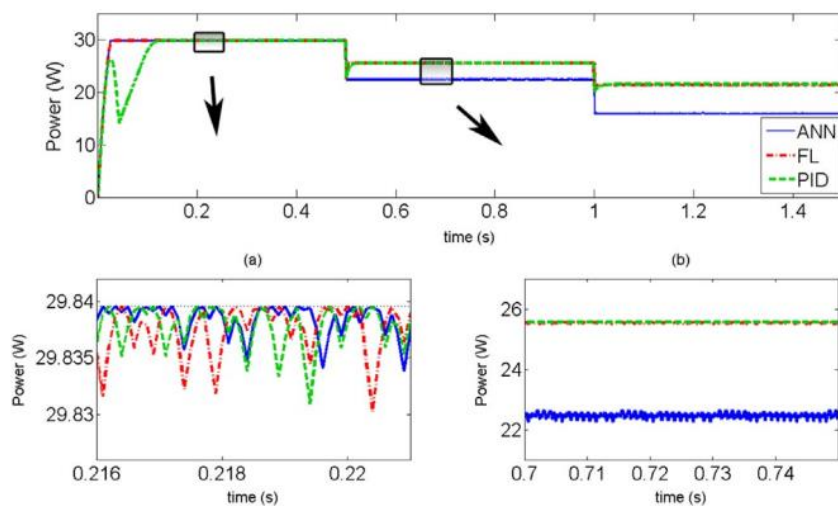
Εικόνα 1.24. Ταχύτητα σύγκλισης αλγορίθμων [2]

MPPT Algorithm	t_{rep} (ms)	E_m	V_m	η (%)
PID	131.5	0.0518	0.6922	99.83
FL	27	0.0188	0.369	99.94
ANN	27	0.0160	0.3168	99.95
GA	139.2	0.0945	0.5214	99.68
PSO	47.2	0.0352	0.3763	99.88

Εικόνα 1.25. Συγκριτικά αποτελέσματα προσομοίωσης [2]

1.10.2 Απόκριση λόγω περιβαλλοντικών αλλαγών

Στο συγκεκριμένο βήμα, προσομοιώθηκε η γήρανση των πάνελ και σε αυτό το στάδιο ελέγχεται η απόκριση των μεθόδων στο χρόνο. Υπάρχουν μέθοδοι που για την υλοποίησή τους δεν απαιτείται κάποια γνώση των χαρακτηριστικών των πάνελ και ονομάζονται άμεσες (Direct). Αντιθέτως, μέθοδοι που απαιτούν γνώσεις των χαρακτηριστικών των πάνελ ή των καιρικών συνθηκών ονομάζονται έμμεσες (Indirect). Συνεπώς, αν αλλάξουν τα χαρακτηριστικά των πάνελ για μια έμμεση μέθοδο, τότε η απόδοση της μεθόδου θα αλλάξει και αυτή. Από τον πίνακα παρακάτω, προκύπτει ότι η γήρανση επηρεάζει σημαντικά την απόδοση ενός φωτοβολταϊκού συστήματος. Η ANN μέθοδος, η οποία είναι έμμεση, αστοχεί να εντοπίσει το νέο MPP σημείο, το οποίο μεταφράζεται σε σταδιακή μείωση της απόδοσης του συστήματος MPPT. Με το πέρασμα του χρόνου, το σύστημα που χρησιμοποιεί νευρωνικό δίκτυο θα χρειαστεί ξανά εκπαίδευση. Οι υπόλοιπες μέθοδοι ανταποκρίνονται με επιτυχία στις αλλαγές των παραμέτρων και υποδεικνύουν το πλεονέκτημα που εμφανίζει η άμεση μεθοδολογία.



Εικόνα 1.26. Απόδοση μεθόδων έπειτα από γήρανση των πάνελ [2]

1.10.3 Απαιτήσεις υλισμικού – λογισμικού

Στην εικόνα 1.24, παρουσιάζεται ένας πίνακας στον οποίο απεικονίζεται το κόστος υλοποίησης ενός MPPT ελεγκτή που χρησιμοποιεί έναν από τους αλγόριθμους που παρουσιάστηκαν. Ο ελεγκτής PID είναι εύκολος, απλός και απαιτεί μικρό αριθμό αισθητήρων. Η υλοποίηση του μπορεί να γίνει είναι με αναλογικά είτε με ψηφιακά κυκλώματα. Ωστόσο, οι μέθοδοι FL, PSO, GA και ANN είναι πιο σύνθετες και απαιτούν χρήση επεξεργαστή. Η μέθοδος ANN απαιτεί επιπλέον αισθητήρες ηλιακής ακτινοβολίας και αισθητήρες θερμοκρασίας, το οποίο αυξάνει το κόστος της υλοποίησης.

MPPT technique	PID	FL	ANN	GA	PSO
Required sensors	Current, Voltage	Current, Voltage	Temperature, Irradiation	Current, Voltage	Current, Voltage
Hardware implementation	Analog or Digital	Digital	Digital	Digital	Digital
Software complexity	Low	High	Medium	Very high	High
Total Cost	Low	Medium	High	Medium	Medium

Εικόνα 1.27. Σύγκριση κόστους υλοποίησης τεχνικών MPPT [2]

1.11 Συμπεράσματα

Η μέθοδος με τον PID αποτελεί μια οικονομική, εύκολη και απλή λύση. Όταν απαιτείται απόδοση με μικρό κόστος, ο PID είναι μια λύση συμβιβασμού. Η χρήση του συγκεκριμένου αλγορίθμου συνίσταται σε εγκαταστάσεις με μικρό κόστος, όπως πχ ποτιστικά συστήματα ή συστήματα φωτισμού οικιακών περιοχών.

Τα νευρωνικά δίκτυα (ANN) είναι η ιδανική επιλογή όταν απαιτείται υψηλή απόδοση στο σύστημα. Η απόδοση τους ξεπερνά όλες τις υπόλοιπες τεχνικές, αλλά έχουν ένα σοβαρό μειονέκτημα. Απαιτούνται αισθητήρες για την ηλιακή ακτινοβολία και αισθητήρες θερμοκρασίας. Αυτό αυξάνει αρκετά το κόστος υλοποίησης της συγκεκριμένης μεθόδου. Επίσης, η απόδοση της συγκεκριμένης μεθόδου μειώνεται με την γήρανση των πάνελ. Συνεπώς, δεν είναι κατάλληλη για εφαρμογές MPPT.

Η τεχνική της ασαφούς λογικής (FL), έχει υψηλή απόδοση με πολύ καλά αποτελέσματα. Επίσης, δεν απαιτείται γνώση των χαρακτηριστικών των πάνελ και μπορεί να υλοποιηθεί χρησιμοποιώντας έναν αισθητήρα ρεύματος και έναν αισθητήρα τάσης. Η συγκεκριμένη μέθοδος είναι κατάλληλη σε ακριβές εφαρμογές για τις οποίες απαιτείται μεγάλη ακρίβεια, όπως πχ για δορυφορικές εφαρμογές.

Ο PSO και οι γενετικοί αλγόριθμοι GA έχουν υψηλή απόδοση. Εντούτοις, αυτές οι μεθοδολογίες απαιτούν αρκετές επαναλήψεις, έτσι ώστε να εντοπίσουν το σημείο MPP, το οποίο μεταφράζεται σε αυξημένο χρόνο εκτέλεσης. Επίσης, χρειάζονται αρχικοποίηση και επανεκτέλεση στην περίπτωση που αλλάξουν οι περιβαλλοντικές συνθήκες. Οι παραπάνω παράγοντες υποβιβάζουν την απόδοση τους. Δεν προτείνεται σε μικρές εγκαταστάσεις με ομοιόμορφη ηλιοφάνεια. Προτείνεται ως μεθοδολογία σε φωτοβολταϊκά πάρκα, όπου υπάρχει έντονο το φαινόμενο της μερικής σκίασης.

Κεφάλαιο 2ο: Αλγόριθμος P&O

2.1 Εισαγωγή

Ένας αρκετά διαδεδομένος αλγόριθμος εύρεσης του σημείου MPP είναι ο P&O. Χρησιμοποιείται ευρέως στον τομέα των φωτοβολταϊκών και δύναται να υλοποιηθεί από χαμηλού κόστους μικροεπεξεργαστές. Είναι αρκετά εύκολος στην υλοποίηση του, συγκλίνει γρήγορα σε ένα σημείο (βέλτιστο ή μη) και δεν απαιτεί μεγάλο αριθμό αισθητήρων για τη λειτουργία του. Εντούτοις, η κλασσική υλοποίηση του αλγορίθμου P&O αντιμετωπίζει προβλήματα, τα οποία επηρεάζουν τόσο την απόδοση όσο και την ταχύτητα του.

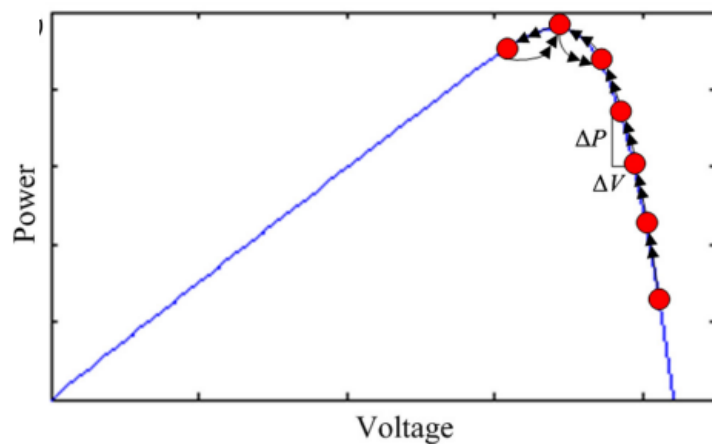
2.2 Ταλαντώσεις σε σταθερή κατάσταση (σημείο MPP)

Κατά την υλοποίηση της συμβατικής μεθόδου του P&O, ο υπολογισμός της ισχύος (P) της φωτοβολταϊκής συστοιχίας υπολογίζεται από τις μετρούμενες τιμές της τάσης (V) και του ρεύματος (I). Με βάση τις τιμές αυτές, ο αλγόριθμος προκαλεί μια διαταραχή (ΔV) στην τάση λειτουργίας της φωτοβολταϊκής συστοιχίας, η οποία βασίζεται στην αλλαγή της τιμής της ισχύος (P). Η νέα τιμή της τάσης (V_{new}) υπολογίζεται σύμφωνα με τις παρακάτω εξισώσεις:

$$V_{new} = V_{old} + \Delta V \times slope \text{ (if } P > P_{old} \text{)} \quad (2.1)$$

$$V_{new} = V_{old} - \Delta V \times slope \text{ (if } P < P_{old} \text{)} \quad (2.2)$$

όπου $slope$ θεωρείται η κατεύθυνση της διαταραχής (ανοδική – καθοδική).

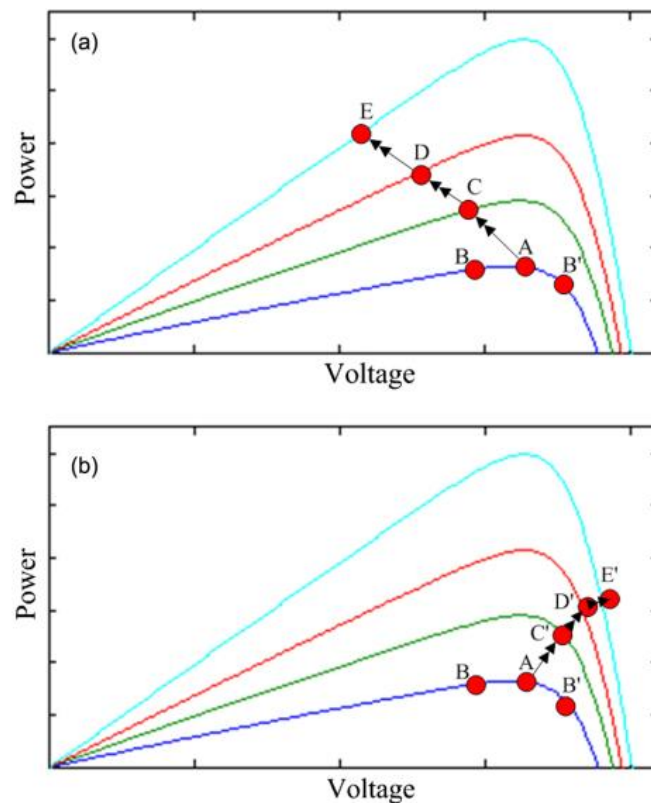


Εικόνα 2.1. Βήμα αλγορίθμου P&O και ταλαντώσεις γύρω από το σημείο MPP [1]

2.3 Απόκλιση από το σημείο MPP

Εκτός από τις απώλειες σταθερής κατάστασης, ο αλγόριθμος P&O υστερεί στην ικανότητα παρακολούθησης του σημείου MPP. Απόκλιση του σημείου MPP πραγματοποιείται κατά τη σταδιακή αλλαγή της ακτινοβολίας. Θεωρώντας τις καμπύλες ισχύος της εικόνας 2.2, το MPP σημείο βρίσκεται στη θέση A. Συνεπώς, υπάρχει μια ταλάντωση του σημείου λειτουργίας, γύρω από το σημείο A (σημεία B και B'). Έστω ότι ο αλγόριθμος βρίσκεται στο σημείο B' και η ακτινοβολία αρχίζει να αυξάνεται σταδιακά. Το αποτέλεσμα της σταδιακής αύξησης της ακτινοβολίας θα είναι η μετατόπιση του σημείου λειτουργίας από το A στο C. Είναι προφανές ότι ο αλγόριθμος δεν καταφέρνει να εντοπίσει το νέο σημείο λειτουργίας και με την προϋπόθεση ότι η αύξηση της ακτινοβολίας συνεχίζεται, τότε το σημείο

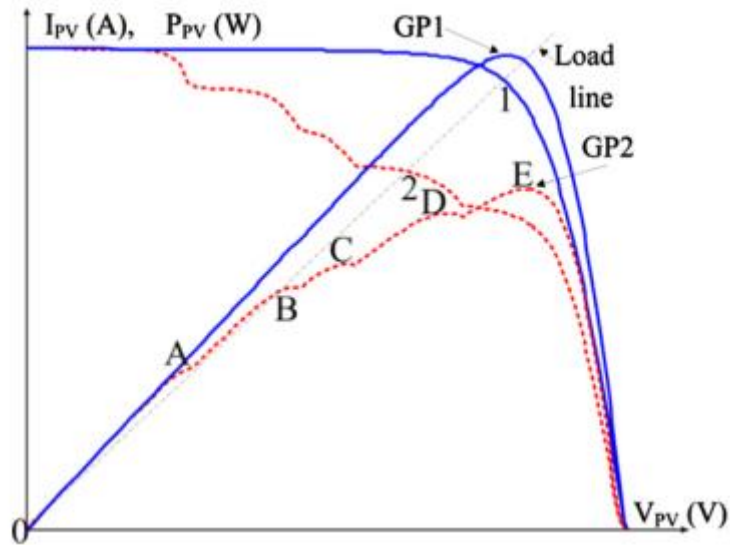
λειτουργίας του συστήματος θα ακολουθήσει το μονοπάτι A-C-D-E. Το αντίστοιχο φαινόμενο πραγματοποιείται και κατά τη σταδιακή μείωση της ακτινοβολίας. Εντούτοις, το φαινόμενο αυτό λαμβάνει μέρος μόνο κατά τη σταδιακή αλλαγή της έντασης της ηλιακής ακτινοβολίας.



Εικόνα 2.2. Απόκλιση από το σημείο MPP έπειτα από σταδιακή αύξηση της ακτινοβολίας [1]

2.4 Αδυναμία εύρεσης βέλτιστου σημείου MPP υπό συνθήκες μερικής σκίασης

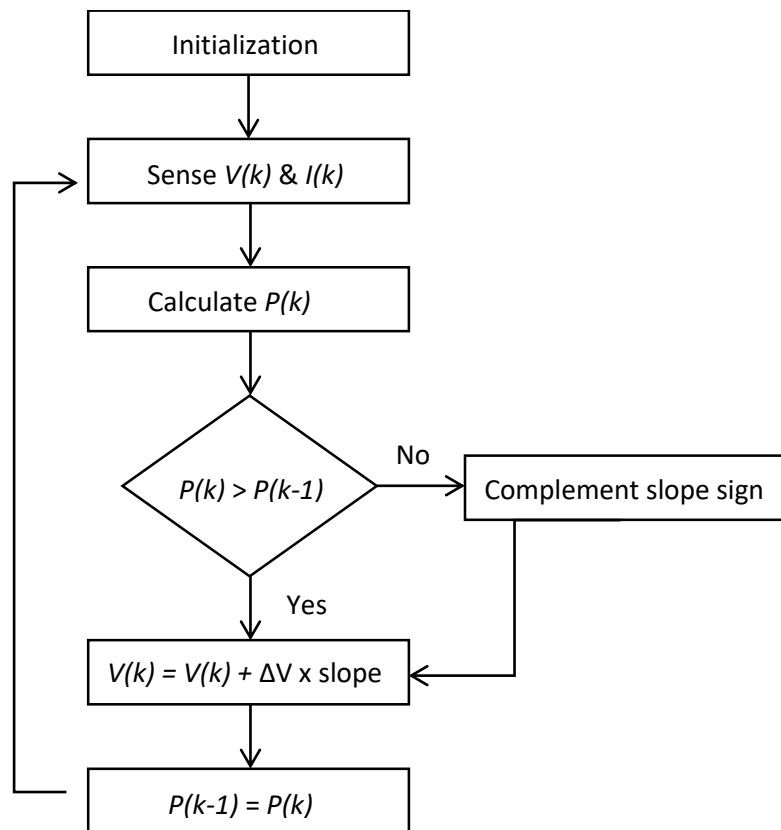
Σε συνθήκες όπου η ακτινοβολία είναι ομοιόμορφα κατανομημένη, στην καμπύλη ισχύος – τάσης ($P-V$) της φωτοβολταϊκής συστοιχίας υπάρχει ένα μοναδικό σημείο MPP (GP1, εικόνα 2.3). Ωστόσο, υπό συνθήκες μερικής σκίασης, είναι δυνατό να προκύψουν πολλά και διαφορετικά σημεία τοπικού μεγίστου. Στην εικόνα 2.3, για παράδειγμα, παρατηρούνται πέντε σημεία A-E. Το σημείο E είναι το ολικό μέγιστο σημείο της καμπύλης και θεωρείται το σημείο MPP. Ο αλγόριθμος πρέπει να είναι ικανός να εντοπίσει με επιτυχία το σημείο E, διότι στο συγκεκριμένο σημείο μεγιστοποιείται η ισχύς που μπορεί να αντληθεί από τη φωτοβολταϊκή συστοιχία.



Εικόνα 2.3. Καμπύλη ισχύος – τάσης (P – V) και απεικόνιση σημείων MPP [3]

Θεωρείται το σενάριο κατά το οποίο υπάρχει ομοιόμορφη κατανομή στην ηλιακή ακτινοβολία. Έπειτα από κάποιο χρονικό διάστημα, έστω ότι επικρατούν συνθήκες μερικής σκίασης. Παρουσία μερικής σκίασης, η καμπύλη ισχύος – τάσης (P-V) αλλάζει μορφή (1 $\rightarrow$ 2). Οι συνθήκες της μερικής σκίασης επηρεάζουν την απόδοση του συστήματος, με αποτέλεσμα την ενεργοποίηση του αλγορίθμου P&O, για να εντοπιστεί το νέο σημείο MPP (E στην παρούσα περίπτωση). Εντούτοις, ο αλγόριθμος P&O εγκλωβίζεται στο σημείο D, με αποτέλεσμα τη μείωση της απόδοσης του συστήματος. Το παραπάνω παράδειγμα υποδεικνύει τον περιορισμό του αλγορίθμου P&O στον εντοπισμό του σημείου MPP. Κατά καιρούς οι ερευνητές έχουν υλοποιήσει πολλαπλές παραλλαγές, έτσι ώστε να αυξήσουν την απόδοση του συγκεκριμένου αλγορίθμου, είτε προσαρμόζοντας κατάλληλα τις παραμέτρους του [3], είτε συνδυάζοντας το συγκεκριμένο αλγόριθμο με άλλους αποδοτικότερους [4]. Για τις ανάγκες της παρούσης εργασίας, θεωρήθηκε η συμβατική μορφή του P&O, διότι ως μέθοδος είναι η δημοφιλέστερη και αυτή που χρησιμοποιείται περισσότερο από κάθε άλλη [5].

2.5 Διάγραμμα ροής αλγορίθμου P&O

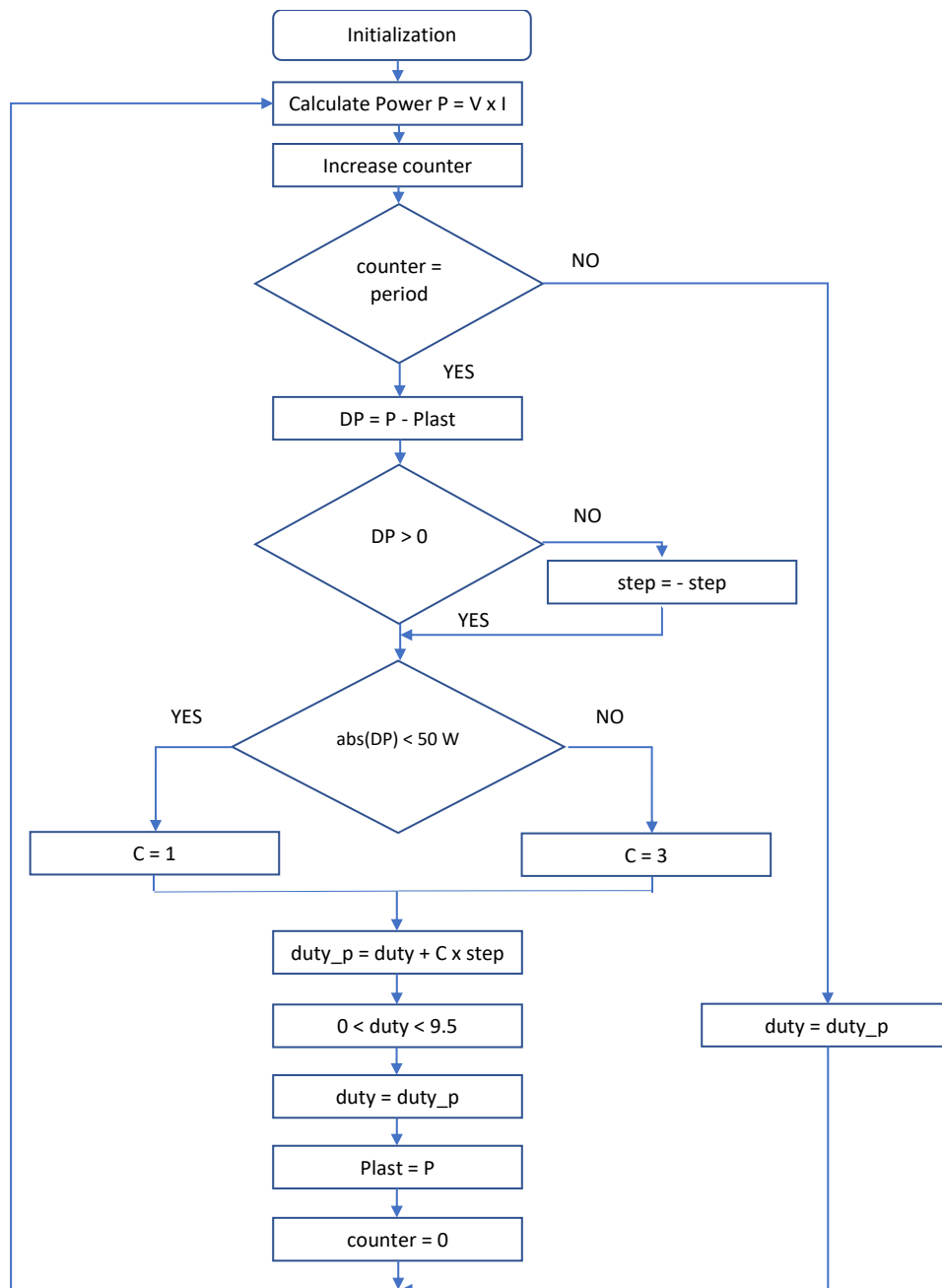


Εικόνα 2.4. Προτεινόμενο διάγραμμα ροής αλγορίθμου P&O [1]

Ο συγκεκριμένος αλγόριθμος χρησιμοποιείται στην παρούσα εργασία ως μέτρο σύγκρισης της απόδοσης των υπόλοιπων αλγορίθμων. Η τιμή της διαταραχής ορίστηκε ως 0.05 ($\Delta V = 0.05$) για τις ανάγκες της εργασίας. Παρακάτω παρουσιάζεται το υλοποιημένο διάγραμμα ροής, ο πίνακας με τις τιμές του αλγορίθμου που χρησιμοποιήθηκαν καθώς και ο ψευδο-κώδικας του αλγορίθμου P&O.

Πίνακας 2.1. Πίνακας τιμών για αλγόριθμο P&O

Παράμετρος	Τιμή	Σχόλια
Ddc	0.05	Βήμα αλγορίθμου
C	1 ή 3	Συντελεστής βήματος



Εικόνα 2.5. Υλοποιημένο διάγραμμα ροής αλγορίθμου P&O

2.6 Ψευδό – κώδικας P&O

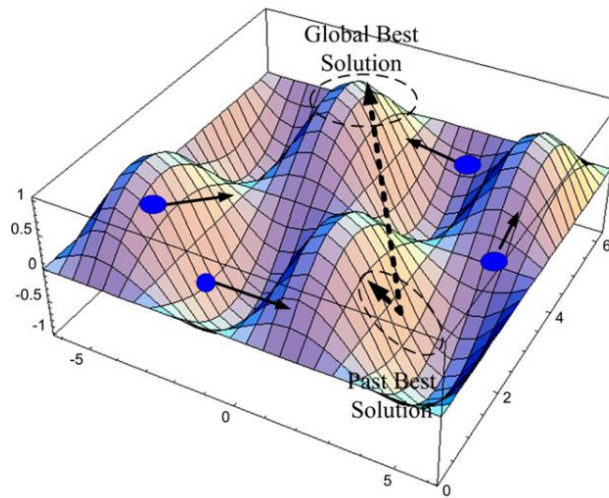
Psedo-code for P&O

```
initialize all required parameters

calculate Power
increase counter
if counter == period
    calculate dP = Power - Previous_Power
    if dP < 0
        step = - step
    if dP > 50 W
        C = 3
    else
        C = 1
    duty_last = duty_last + C * step
    0 < duty_lat < 9.5 A
    Previous_Power = Power
    duty = duty_last
    counter = 0
else
    duty = duty_last
```

Εικόνα 2.6. Ψευδό - κώδικας για αλγόριθμο P&O

Κεφάλαιο 3ο: Αλγόριθμος PSO



Εικόνα 3.1: Σωματίδια στο χώρο λύσης του προβλήματος [6]

Ο αλγόριθμος PSO εμπνέεται από τις διεργασίες συλλογικής συμπεριφοράς των έμβιων οργανισμών. Προτάθηκε το 1995 από τους J.Kennedy & R.Eberhart ως αλγόριθμος βελτιστοποίησης σμήνους (ή πλήθους) σωματιδίων (Particle Swarm Optimization). Η αρχική έμπνευση προέρχεται από τη συλλογική κίνηση σμήνους πτηνών, από όπου προέρχεται και το μεγαλύτερο μέρος της ορολογίας που χρησιμοποιείται για την περιγραφή του αλγορίθμου. Ο συγκεκριμένος αλγόριθμος δεν ανήκει στους εξελικτικούς αλγορίθμους, καθώς τα άτομα του πληθυσμού δεν ανταλλάσσουν άμεσα τιμές, αλλά μόνο έμμεσα. Επίσης, εισάγει την έννοια της δυναμικής κίνησης εντός του χώρου αναζήτησης της λύσης.

3.1 Βασικές αρχές συλλογικής συμπεριφοράς

Η συλλογική κίνηση ενός πληθυσμού εμφανίζεται πολύ συχνά στο ζωικό βασίλειο. Ζώα που κινούνται σε ομάδες ή αγέλες έχουν μεγαλύτερη πιθανότητα να βρουν φαγητό. Αντίστοιχες διεργασίες εντοπίζονται και στις ανθρώπινες δραστηριότητες, όπως για παράδειγμα σε αγώνες ποδηλασίας, όπου οι ποδηλάτες κινούνται σε ομάδες με στόχο τη μικρότερη δυνατή κατανάλωση ενέργειας. Η νοημοσύνη σμήνους βασίζεται σε τρεις βασικές αρχές συλλογικής συμπεριφοράς που τους βοηθούν να εντοπίζουν την επόμενη καλύτερη λύση:

- Αδράνεια (Inertia): Η τάση για διατήρηση παλαιότερων συμπεριφορών που αποδείχθηκαν σωστές.
- Κοινωνική επιρροή.
- Επιρροή από γειτονικά άτομα: Η αλληλεπίδραση με γειτονικά πρόσωπα έχει μεγαλύτερη ισχύ από την επιρροή ολόκληρης της κοινωνίας.

Με βάση τα παραπάνω διαμορφώθηκε ο αλγόριθμος PSO.

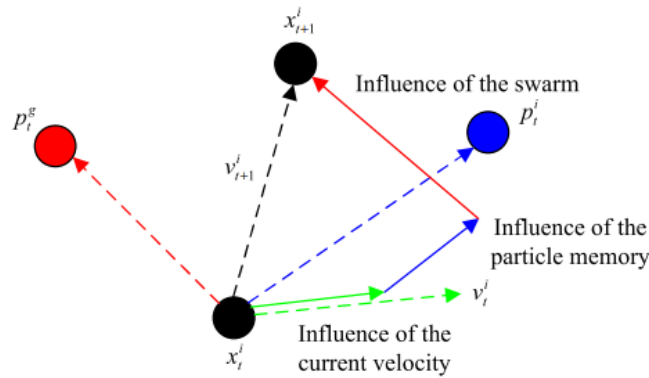
3.2 Ο βασικός αλγόριθμος PSO

Έστω ένα πρόβλημα βελτιστοποίησης (στην παρούσα εργασία εντοπίζεται το πρόβλημα μεγιστοποίησης της ισχύος) που ορίζεται στον πραγματικό χώρο n διαστάσεων ($n = 1$ για την εύρεση MPP σημείου). Αρχικά ορίζεται ένας πληθυσμός N ατόμων (σωματιδίων) $\{x_i\}, i = 1, \dots, N$ τα οποία αντιπροσωπεύουν ένα κινούμενο σμήνος στο χώρο των λύσεων. Στη συνέχεια ορίζεται μια συνάρτηση καταλληλότητας $f(x_i)$, την οποία ο αλγόριθμος καλείται να μεγιστοποιήσει. Οι θέσεις των σωματιδίων στο χώρο είναι οι λύσεις του προβλήματος. Για κάθε άτομο ορίζεται και ένα αντίστοιχο διάνυσμα ταχύτητας n διαστάσεων $\{v_i\}$, που καθορίζει την ταχύτητα κίνησης ως προς κάθε διεύθυνση. Κάθε άτομο τείνει να διατηρήσει την ταχύτητα του (αδράνεια), αλλά αυτή επηρεάζεται από δύο παράγοντες:

- Την καλύτερη θέση του ως την τρέχουσα γενιά.
- Την καλύτερη θέση όλου του πληθυσμού μέχρι την τρέχουσα γενιά.

Ο αλγόριθμος εκτελείται για συγκεκριμένο αριθμό επαναλήψεων (γενιές), όπου σε κάθε γενιά ενημερώνεται η θέση και η ταχύτητα των σωματιδίων. Επίσης, ενημερώνεται η καλύτερη θέση του κάθε σωματιδίου, καθώς και η βέλτιστη θέση όλου του πληθυσμού. Ο αλγόριθμος δύναται να τερματίσει επίσης, αν ικανοποιείται ένα κριτήριο τερματισμού που έχει θέσει ο χρήστης. Η καλύτερη θέση όλου του πληθυσμού είναι και η λύση του προβλήματος βελτιστοποίησης.

Στο παρακάτω σχήμα απεικονίζεται ο τρόπος με τον οποίο επιλέγεται η θέση των σωματιδίων για κάθε επόμενη γενιά.



Εικόνα 3.2: Επιλογή επόμενης θέσης σωματιδίων [7]

Με κόκκινο χρώμα στην παραπάνω εικόνα απεικονίζεται η βέλτιστη θέση του πληθυσμού, ενώ με μπλε χρώμα απεικονίζεται η βέλτιστη θέση του σωματιδίου μέχρι την τρέχουσα γενιά. Τέλος, με πράσινο χρώμα απεικονίζεται η ταχύτητα που έχει το σωματίδιο στην τρέχουσα γενιά. Συνδυάζοντας όλα τα παραπάνω, υπολογίζεται η νέα ταχύτητα του σωματιδίου από την παρακάτω σχέση:

$$v_{t+1}^i = w * v_t^i + c_1 * r_1 * (p_t^i - x_t^i) + c_2 * r_2 * (p_t^g - x_t^i) \quad (3.1)$$

Η παραπάνω εξίσωση μπορεί να μεταφραστεί ως εξής:

$$\text{ταχύτητα}' = \text{αδράνεια} + \text{βέλτιστη θέση σωματιδίου} + \text{βέλτιστη θέση πληθυσμού}$$

Η νέα θέση των σωματιδίων υπολογίζεται από την παρακάτω σχέση:

$$x_{t+1}^i = x_t^i + v_{t+1}^i \quad (3.2)$$

Από τις παραπάνω εξισώσεις προκύπτει, ότι ο αλγόριθμος μπορεί να ωθήσει μια μεταβλητή εκτός του διαστήματος ορισμού της. Συνεπώς, είναι σημαντικό να περιορίσουμε τις μεταβλητές αυτές εντός του πεδίου ορισμού.

$$x_i \leftarrow \min(x_i, x_{\max}) \quad (3.3)$$

$$x_i \leftarrow \max(x_i, x_{\min}) \quad (3.4)$$

$$v_i \leftarrow \min(v_i, v_{\max}) \quad (3.5)$$

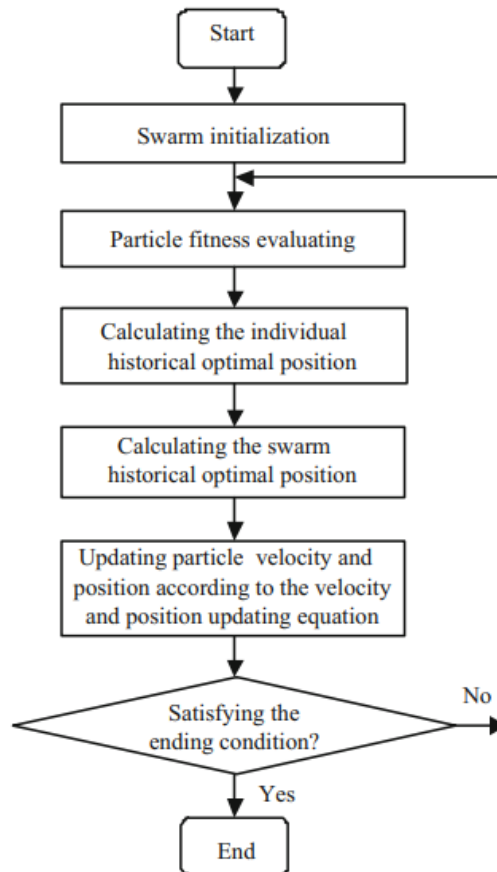
$$v_i \leftarrow \max(v_i, v_{\min}) \quad (3.6)$$

Για την εκτέλεση του αλγορίθμου πρέπει να καθοριστούν οι παρακάτω τιμές:

- Αρχικός αριθμός σωματιδίων N
- Αρχικές θέσεις και ταχύτητες πληθυσμού $\{x_i\}, \{v_i\}$
- Τιμές επιρροής c_1 και c_2
- Όρια ταχυτήτων σωματιδίων
- Βάρος συντελεστή αδράνειας w

Στο παρακάτω σχήμα απεικονίζεται το βασικό διάγραμμα ροής του αλγορίθμου PSO. Διακρίνονται έξι διαφορετικά βήματα:

1. Αρχικοποίηση αλγορίθμου: Συνήθως τα σωματίδια λαμβάνουν τυχαίες τιμές εντός του πεδίου ορισμού του προβλήματος.
2. Αξιολόγηση της συνάρτησης κόστους για κάθε ένα από τα στοιχεία του πληθυσμού.
3. Ενημέρωση καλύτερης θέσης των σωματιδίων (individual best)
4. Εντοπισμός του σωματιδίου που περιέχει τη τρέχουσα βέλτιστη λύση (global best).
5. Ενημέρωση θέσης και ταχύτητας του πληθυσμού σύμφωνα με τις παραπάνω εξισώσεις.
6. Έλεγχος ικανοποίησης κριτηρίου τερματισμού.



Εικόνα 3.3: Διάγραμμα ροής για αλγόριθμο PSO [7]

Λαμβάνοντας υπόψιν όλα τα παραπάνω, ο αλγόριθμος PSO δύναται να τροποποιηθεί και να χρησιμοποιηθεί για τον εντοπισμό του βέλτιστου σημείου λειτουργίας MPP σε ένα φωτοβολταϊκό σύστημα.

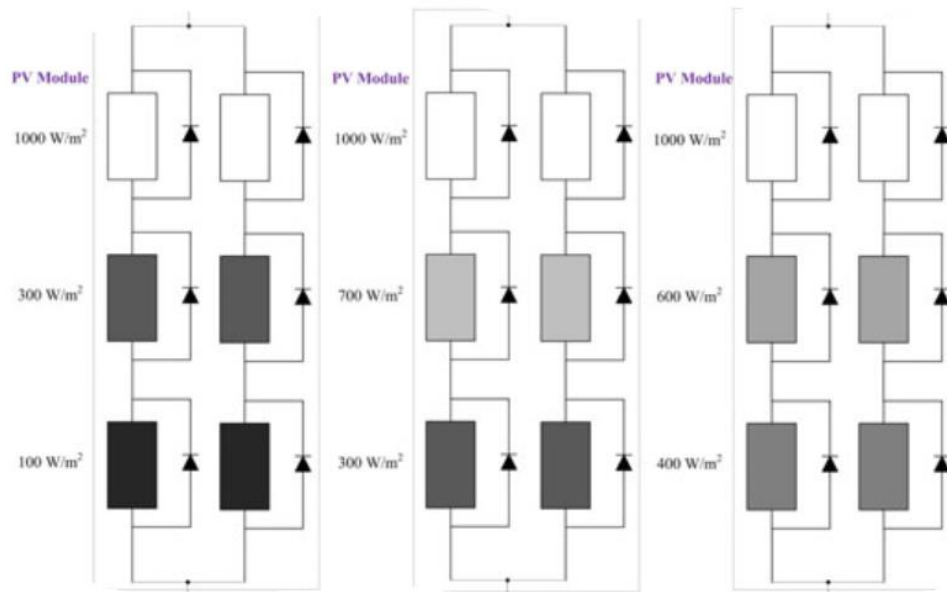
3.3 Αλγόριθμος PSO για MPPT

Ο αλγόριθμος PSO μπορεί να τροποποιηθεί και να χρησιμοποιηθεί για την εξαγωγή του βέλτιστου σημείου MPP. Για τις ανάγκες της εργασίας υλοποιήθηκαν δύο εκδοχές του αλγορίθμου PSO. Η πρώτη εκδοχή του αλγορίθμου είναι βασισμένη κατά ένα μεγάλο ποσοστό στην υλοποίηση που έγινε στο άρθρο [6]. Στο άρθρο [6], παρουσιάστηκε μια νέα μέθοδος ανανέωσης της ταχύτητας και της θέσης του πληθυσμού, η οποία όμως δεν παρουσίασε τα επιθυμητά αποτελέσματα και απορρίφθηκε έπειτα από πειραματισμούς. Συνεπώς, χρησιμοποιήθηκε η μοντελοποίηση του προβλήματος από το άρθρο [6] αλλά η ανανέωση της θέσης των σωματιδίων υλοποιήθηκε βάση του κλασσικού αλγορίθμου [8].

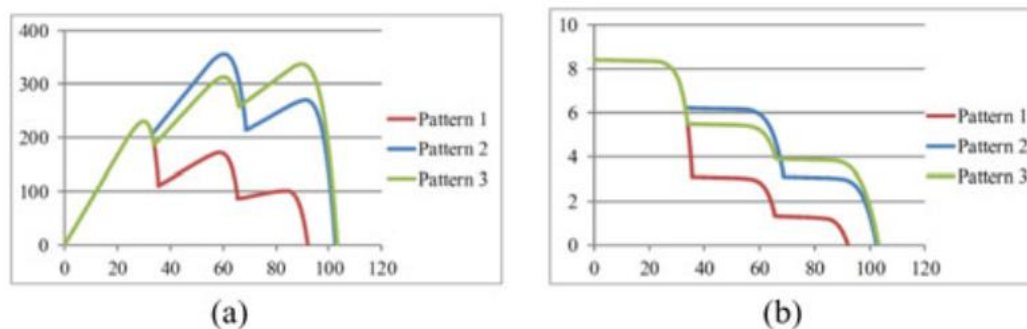
3.3.1 Μοντελοποίηση του προβλήματος

Μια φωτοβολταϊκή συστοιχία αποτελείται από πολλά πάνελ, συνδεδεμένα είτε σε σειρά, είτε παράλληλα. Όταν κάποιο από τα πάνελ της συστοιχίας βρίσκεται υπό σκίαση, συμπεριφέρεται ως φορτίο, αντί για γεννήτρια παραγωγής ρεύματος, με αποτέλεσμα να δημιουργείται ένα προβληματικό σημείο στην εγκατάσταση. Για αυτό το λόγο τοποθετείται η δίοδος bypass. Λόγω της διόδου, η συστοιχία παρουσιάζει πολλά τοπικά σημεία λειτουργίας και ένα από αυτά τα σημεία είναι και το

σημείο MPP, στο οποίο έχουμε τη μέγιστη απόδοση του συστήματος. Στην παρακάτω εικόνα απεικονίζονται συστήματα στα οποία έχουμε μερική σκίαση (διαφορετική ένταση ακτινοβολίας / πάνελ) και στη συνέχεια απεικονίζονται οι καμπύλες P-V και I-V των συστημάτων.

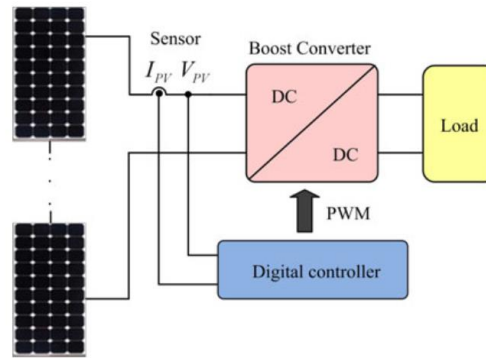


Εικόνα 3.4: Περιπτώσεις μερικής σκίασης σε πάνελ [9]



Εικόνα 3.5: (a) Χαρακτηριστική P-V, (b) Χαρακτηριστική I-V [9]

Παρατηρώντας τις παραπάνω χαρακτηριστικές, διαπιστώνεται ότι το πρόβλημα εύρεσης του βέλτιστου σημείου λειτουργίας MPP είναι ουσιαστικά ένα πρόβλημα μεγιστοποίησης της ισχύος. Η κλασσική εκδοχή της διαταραχής και παρατήρησης (P&O) αντιμετωπίζει πρόβλημα στην εύρεση του μεγίστου σε τέτοιες περιπτώσεις, διότι μπορεί εύκολα να εγκλωβιστεί σε τοπικά μέγιστα. Συνεπώς, υπάρχει η ανάγκη χρήσης αλγορίθμων οι οποίοι έχουν τη δυνατότητα να προσεγγίζουν το ολικό μέγιστο της συνάρτησης καταλληλότητας και όχι κάποιο τοπικό. Ο αλγόριθμος PSO είναι ιδανικός για την επίλυση τέτοιου είδους προβλημάτων. Παρακάτω περιγράφονται τα στάδια υλοποίησης του αλγορίθμου PSO για την εύρεση του μεγίστου σημείου MPP.



Εικόνα 3.6: Φωτοβολταϊκό σύστημα για έλεγχο του αλγορίθμου PSO [9]

3.3.2 Στάδια Αλγορίθμου

Ο αλγόριθμος PSO αποτελείται από τα παρακάτω στάδια:

1. **Επιλογή παραμέτρων:** Ως θέση των σωματιδίων θεωρείται το ρεύμα το οποίο επιβάλλει η πηγή ρεύματος στο σύστημα. Ο αλγόριθμος ρυθμίζει το ρεύμα της πηγής, το οποίο με τη σειρά του επιβάλλεται στο φωτοβολταϊκό σύστημα από το οποίο γίνεται η μέτρηση της ισχύος του. Όσο περισσότερα σωματίδια έχει ο πληθυσμός, τόσο ακριβέστερα και αμεσότερα αποτελέσματα αναμένει ο χρήστης. Το μειονέκτημα εντοπίζεται στο γεγονός ότι για μεγάλο πληθυσμό σωματιδίων απαιτείται μεγάλος υπολογιστικός χρόνος, που είναι ανεπιθύμητο σε τέτοια συστήματα. Σύμφωνα με τη βιβλιογραφία [10], υπάρχουν μέχρι m MPPs σημεία σε μια φωτοβολταϊκή διάταξη με m συνδεδεμένα πάνελ σε σειρά. Στην παρούσα εργασία, 4 πάνελ βρίσκονται σε σειρά συνδεδεμένα μεταξύ τους. Θεωρητικά λοιπόν, σε συνθήκες μερικής σκίασης, θα εντοπιστούν μέχρι και 4 διαφορετικές αιχμές στις χαρακτηριστικές καμπύλες. Μια συνηθισμένη πρακτική στην βιβλιογραφία [9], είναι να επιλέγονται τόσα σωματίδια, όσες και οι διαφορετικές αιχμές που εντοπίζονται στις χαρακτηρίστηκες καμπύλες. Ωστόσο, μέσω πειραματισμών προέκυψε ότι η απόδοση του αλγορίθμου είναι παρόμοια για $N=4$ και $N=3$. Συνεπώς, για να υπάρχει μια ισορροπία ανάμεσα στα αποτελέσματα και το χρόνο εκτέλεσης της προσομοίωσης, ορίστηκε ότι για το συγκεκριμένο πρόβλημα $N=3$ (3 σωματίδια).

Πίνακας 3.1. Παράμετροι που χρησιμοποιήθηκαν για τον PSO

Παράμετρος	Τιμή	Σχόλια
Αριθμός σωματιδίων (N)	3	Πληθυσμός λύσεων
Αριθμός γενιών ($\max_iter$)	50	Μέγιστος αριθμός γενιών
Μεταβολή ρεύματος (current_step)	0.3	Μέγιστη μεταβολή ρεύματος από βήμα σε βήμα
Χρόνος δειγματοληψίας (mean_time)	15	Συλλογή δείγματος μετά από 15 ms από ορισμένη τιμή

2. **Αρχικοποίηση αλγορίθμου:** Το πρόβλημα που καλείται ο αλγόριθμος να λύσει είναι μονοδιάστατο και η λύση βρίσκεται στο διάστημα $[0,9.5]$. Συνεπώς οι αρχικές θέσεις των σωματιδίων μπορούν να λάβουν τυχαίες τιμές στο διάστημα αυτό. Οι αρχικές τιμές των σωματιδίων είναι καθοριστικές για την ταχύτητα σύγκλισης του αλγορίθμου και για αυτό το λόγο τα σωματίδια τοποθετήθηκαν ομοιόμορφα σε ίσες αποστάσεις μεταξύ τους στο διάστημα αυτό, έτσι ώστε να ευνοείται η εξερεύνηση στις αρχικές γενιές.

3. **Αξιολόγηση συνάρτησης καταλληλότητας:** Σκοπός του αλγορίθμου είναι να μεγιστοποιηθεί η παραγόμενη ισχύς. Οι θέσεις των σωματιδίων αντιστοιχούν σε διαφορετικές τιμές ρεύματος, το οποίο μεταφράζεται σε διαφορετικά μεγέθη ισχύος. Για κάθε θέση των σωματιδίων του πληθυσμού γίνεται μέτρηση της τάσης του κυκλώματος και υπολογισμός της παραγόμενης ισχύος.
4. **Ενημέρωση καλύτερης θέσης σωματιδίων και καλύτερης λύσης σε όλο τον πληθυσμό:** Αν κάποιο σωματίδιο πετύχει μεγαλύτερη ισχύ από αυτήν της προηγούμενης θέσης του, τότε η νέα καλύτερη θέση του ενημερώνεται. Υπάρχει, δηλαδή, μνήμη για κάθε σωματίδιο, στην οποία αποθηκεύεται η βέλτιστη τιμή της θέσης του. Επίσης, γίνεται αποθήκευση της καλύτερης λύσης ολόκληρου του πληθυσμού για όλες τις γενιές. Αυτή η λύση, είναι και η λύση του προβλήματος βελτιστοποίησης (λύση που μεγιστοποιεί την ισχύ που αντλείται από το φωτοβολταϊκό σύστημα).
5. **Ενημέρωση θέσης και ταχύτητας σωματιδίων:** Χρησιμοποιώντας τις παρακάτω εξισώσεις υπολογίζεται η νέα θέση και η νέα ταχύτητα των σωματιδίων του πληθυσμού.

$$v_{t+1}^i = w * v_t^i + c_1 * r_1 * (p_{best}^i - x_t^i) + c_2 * r_2 * (g_{best} - x_t^i) \quad (3.7)$$

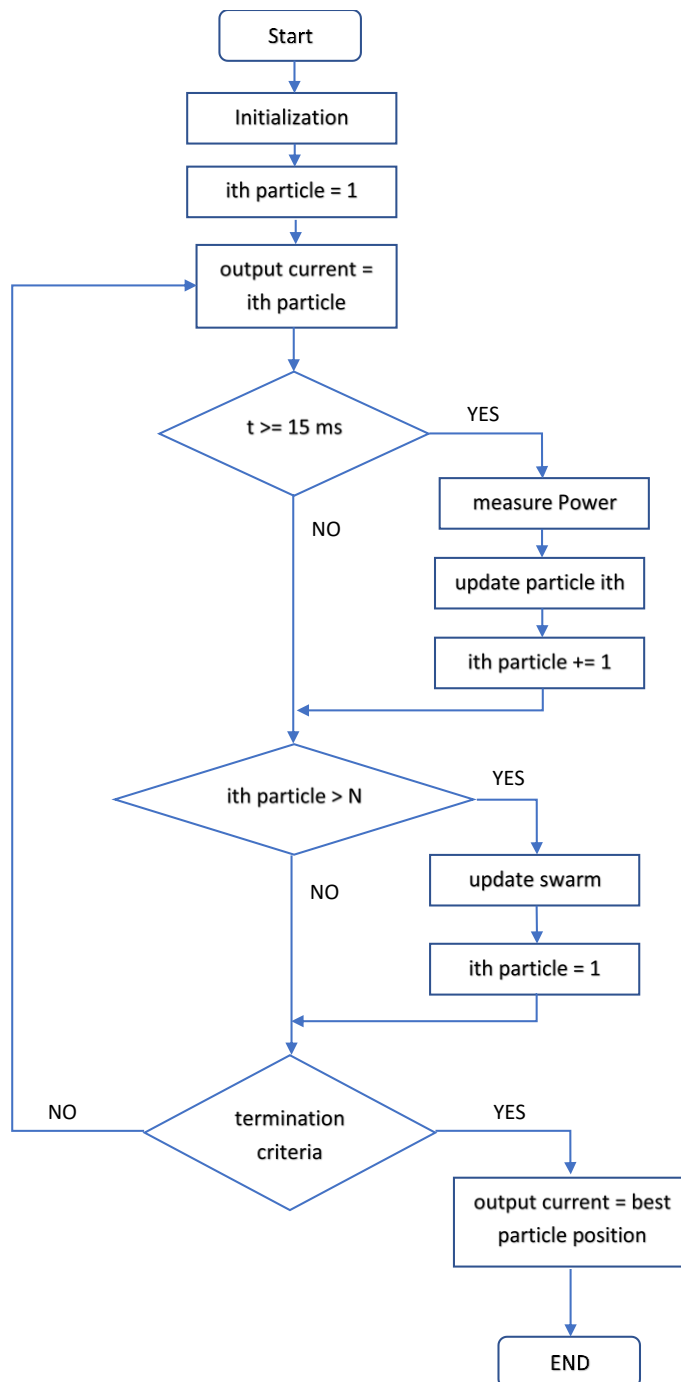
$$x_{t+1}^i = x_t^i + v_{t+1}^i \quad (3.8)$$

Η τιμή w ελέγχει το συντελεστή αδρανείας του αλγορίθμου. Μεγάλες τιμές του w ευνοούν την εξερεύνηση, ενώ μικρές τιμές της τιμής αυτής εύνουν την εκμετάλλευση. Χρειάζεται εντοπισμός της συγκεκριμένης τιμής και είναι διαφορετική για κάθε πρόβλημα. Στην παρούσα εργασία και έπειτα από αρκετούς πειραματισμούς, εξάχθηκε το συμπέρασμα ότι για $w = 0.4$ ο αλγόριθμος παρουσιάζει βέλτιστα αποτελέσματα.

6. **Έλεγχος ικανοποίησης κριτηρίου τερματισμού:** Στο συγκεκριμένο αλγόριθμο ορίζονται δύο κριτήρια τερματισμού. Το πρώτο είναι ο αριθμός των επαναλήψεων του αλγορίθμου. Το δεύτερο είναι οι ταχύτητες των σωματιδίων. Αν όλες οι ταχύτητες του πληθυσμού είναι κάτω από ένα κατώφλι, τότε ο αλγόριθμος θεωρούμε ότι έχει συγκλίνει. Με την ικανοποίηση μιας από των παραπάνω συνθηκών τερματίζεται η εκτέλεση του αλγορίθμου. Ως βέλτιστη λύση, χρησιμοποιείται η καλύτερη θέση που βρέθηκε για όλο τον πληθυσμό.
7. **Επαναρχικοποίηση:** Ανάλογα με τα περιβαλλοντολογικές συνθήκες που επικρατούν σε μια περιοχή, οι συνθήκες σκίασης των πάνελ, άρα και το ολικό μέγιστο του συστήματος, δύναται να αλλάξει μετά από κάποιο χρονικό διάστημα. Συνεπώς, ο αλγόριθμος θα πρέπει να υπολογίσει εκ νέου ποια είναι η μέγιστη ισχύς του συστήματος. Αν η μετρούμενη ισχύς μεταβληθεί ως προς ένα ποσοστό από την προηγούμενη μέτρηση της, τότε ο αλγόριθμος επαναρχικοποιείται και ακολουθείται η διαδικασία από το βήμα 2.

$$\frac{|P_{NEW} - P_{OLD}|}{P_{OLD}} \geq \Delta P \% \quad (3.9)$$

Στην παρακάτω εικόνα απεικονίζεται το διάγραμμα ροής του αλγορίθμου:



Εικόνα 3.7. Διάγραμμα ροής αλγορίθμου PSO & PSO_advanced

Πίνακας 3.2. Συγκεντρωτικός πίνακας παραμέτρων αλγορίθμου PSO

Παράμετρος	Τιμή
Αριθμός σωματιδίων [N]	3
Αριθμός γενιών [max_iter]	50
Μεταβολή ρεύματος [current_step]	0.3
Χρόνος δειγματοληψίας [mean_time]	15

Συντελεστή αδρανείας [w]	0.4
Τιμή επιρροής [c_1]	1
Τιμή επιρροής [c_2]	2

3.3.3 Αλγόριθμος PSO_advanced

Η συγκεκριμένη υλοποίηση του αλγορίθμου χρησιμοποιεί ως βάση την πλατφόρμα του αλγορίθμου PSO, που αναλύθηκε παραπάνω, με την πρώτη διαφορά τους να εντοπίζεται στον τρόπο με τον οποίο ανανεώνεται η ταχύτητα των σωματιδίων. Η τιμή της ταχύτητας των σωματιδίων υπολογίζεται από τον τύπο (4.7) και ως w θεωρείται συνήθως η τιμή 0.9 στην πρώτη γενιά, η οποία καταλήγει γραμμικά στην τιμή 0.4. Μεγάλες τιμές του w ευνοούν την εξερεύνηση, η οποία είναι σημαντική στα πρώτα στάδια του αλγορίθμου. Καθώς όμως αυξάνονται οι γενιές και μειώνοντας την τιμή του w , ευνοείται η εκμετάλλευση των λύσεων που μεγιστοποιούν/ελαχιστοποιούν τη συνάρτηση κόστους. Ο παραπάνω κανόνας μετάβασης της τιμής w , είναι εμπειρικός και δύναται να αλλάζει από πρόβλημα σε πρόβλημα. Η δεύτερη διαφορά της συγκεκριμένης υλοποίησης, εντοπίζεται στην εισαγωγή μιας μεθόδου απόρριψης των χειρότερων λύσεων η οποία θα αναλυθεί παρακάτω.

3.3.4 Βάρος συντελεστή αδρανείας

Ο αλγόριθμος PSO_advanced, χρησιμοποιεί ένα διαφορετικό τρόπο υπολογισμού του βάρους του συντελεστή αδρανείας. Κατά την εκτέλεση του αλγορίθμου PSO, ενδέχεται οι τιμές της ταχύτητας των σωματιδίων να αυξάνονται χωρίς όριο. Για τον περιορισμό αυτής της ανεξέλεγκτης αύξησης των τιμών της ταχύτητας των σωματιδίων χρησιμοποιείται μια μεθοδολογία στάθμισης του συντελεστή αδρανείας. Από την ανάλυση που έγινε στο [11] προέκυψε ο πίνακας της εικόνας 3.8 με τις προτεινόμενες τιμές που πρέπει να έχει ο συντελεστής αδρανείας w καθώς και οι συντελεστές επιρροής c_1 και c_2 .

Από τον πίνακα της εικόνας 3.8 γίνεται εύκολα αντιληπτό ότι οι συντελεστές είναι διαφορετικοί για κάθε πρόβλημα και πρέπει να επιλέγονται με προσοχή, διότι μπορούν να οδηγήσουν τον αλγόριθμο σε αστάθεια – μη σύγκλιση. Σύμφωνα με τη θεωρητική ανάλυση που έγινε στο [11], η χρήση του ισοδύναμου συντελεστή στένωσης (constriction coefficient) εγγυάται την ευστάθεια του αλγορίθμου υπό την προϋπόθεση ότι οι τιμές K και φ_T της εξίσωσης 4.14, ακολουθούν τα παρακάτω όρια:

$$\varphi_T > 4 \quad (3.10)$$

$$K < \frac{2}{\varphi_T - 2 + \sqrt{\varphi_T(\varphi_T - 4)}} \quad (3.11)$$

Problem Dimension	Function Evaluations	N	w	ϕ_1	ϕ_2
2	400	25	0.3925	2.5586	1.3358
		29	-0.4349	-0.6504	2.2073
2	4,000	156	0.4091	2.1304	1.0575
		237	-0.2887	0.4862	2.5067
5	1,000	63	-0.3593	-0.7238	2.0289
		47	-0.1832	0.5287	3.1913
5	10,000	223	-0.3699	-0.1207	3.3657
		203	0.5069	2.5524	1.0056
10	2,000	63	0.6571	1.6319	0.6239
		204	-0.2134	-0.3344	2.3259
10	20,000	53	-0.3488	-0.2746	4.8976
20	40,000	69	-0.4438	-0.2699	3.3950
		149	-0.3236	-0.1136	3.9789
		60	-0.4736	-0.9700	3.7904
		256	-0.3499	-0.0513	4.9087
30	600,000	95	-0.6031	-0.6485	2.6475
50	100,000	106	-0.2256	-0.1564	3.8876
100	200,000	161	-0.2089	-0.0787	3.7637

Εικόνα 3.8. Τιμές συντελεστών βάση διάστασης προβλήματος και πληθυσμού [11]

Η ανανέωση της ταχύτητας των σωματιδίων που υλοποιείται μέσω της μεθοδολογίας του ισοδύναμου συντελεστή στένωσης (constriction coefficient) υπολογίζεται από τους παρακάτω τύπους:

$$v_i = K [v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i)] \quad (3.12)$$

$$v(t+1) = K \left[v_i(t) + (\phi_1 + \phi_2) \left(\frac{\phi_1 b_i(t) + \phi_2 h_i(t)}{\phi_1 + \phi_2} - x_i(t) \right) \right] \quad (3.13)$$

$$v(t+1) = K [v_i(t) + \phi_T(p_i(t) - x_i(t))] \quad (3.14)$$

όπου K ,

$$K = \frac{2a}{\phi_T - 2} \quad (3.15)$$

όπου $\phi_T > 4$ (κατά κανόνα 4.1) και a μεταξύ του διαστήματος (0,1). Έπειτα από μια σειρά πειραματισμών, οι τιμές που επιλέχθηκαν και χρησιμοποιήθηκαν στο πρόβλημα της παρούσας εργασίας είναι οι παρακάτω:

Πίνακας 3.3. Τιμές υπολογισμού ποσοστού σύγκλισης

Παράμετρος	Τιμή
Φ_T	rand [4.1, 8]
a	4.1

3.3.5 Απόρριψη χειρότερης λύσης

Η δεύτερη διαφορά που παρουσιάζει ο PSO_advanced από τον PSO, εντοπίζεται στο γεγονός ότι ο πρώτος υιοθετεί ένα μηχανισμό απόρριψης της χειρότερης λύσης. Ο λόγος που υλοποιήθηκε η συγκεκριμένη λειτουργία είναι για να μειωθεί, όσο το δυνατόν, ο χρόνος εκτέλεσης του αλγορίθμου και να αυξηθεί η απόδοση του. Ένα σύστημα το οποίο αντλεί ενέργεια από τον ήλιο και αλλάζει η απόδοση του σύμφωνα με το ποσοστό της σκίασης που επικρατεί στον περιβάλλοντα χώρο του, πρέπει να αποκρίνεται άμεσα και να συγκλίνει γρήγορα στο βέλτιστο σημείο εξασφαλίζοντας τη μέγιστη δυνατή μεταφορά ενέργειας από το φωτοβολταϊκά πάνελ στο φορτίο. Στη συνέχεια παρουσιάζεται ο τρόπος με τον οποίο υλοποιείται η απόρριψη της χειρότερης λύσης.

3.3.6 Στάδια Αλγορίθμου

Ο αλγόριθμος PSO_advanced αποτελείται από τα παρακάτω στάδια:

1. **Επιλογή παραμέτρων:** Ο αρχικός πληθυσμός που χρησιμοποιείται σε αυτήν την υλοποίηση είναι 5 σωματίδια. Οι αρχικές τιμές των σωματιδίων λειτουργούν καθοριστικά για την απόδοση και τη σύγκλιση του αλγορίθμου. Χρησιμοποιώντας περισσότερα σωματίδια κατά την αρχικοποίηση του αλγορίθμου, αυξάνεται η πιθανότητα κάποιο σωματίδιο, τυχαία, να λάβει τιμή κοντά στο MPP σημείο, κάτι το οποίο μειώνει αρκετά το χρόνο σύγκλισης, καθώς απαιτούνται λιγότερες επαναλήψεις για την εύρεση του ολικού μεγίστου/ελαχίστου. Ο αυξημένος αριθμός του πληθυσμού όμως, οδηγεί σε αύξηση του χρόνου σύγκλισης. Για αυτό το λόγο, στην πρώτη γενιά ο πληθυσμός θα αποτελείται από 5 σωματίδια και σε κάθε επόμενη γενιά θα αφαιρείται το χειρότερο σωματίδιο – αυτό το οποίο παρουσιάζει τη μικρότερη ισχύ. Ο ελάχιστος αριθμός των σωματιδίων του πληθυσμού ορίζεται ως 3 σωματίδια. Σταδιακά ο πληθυσμός θα μειώνεται, αλλά το πλεονέκτημα που προσφέρεται στην πρώτη γενιά – το οποίο είναι η αύξηση της πιθανότητας κάποιο σωματίδιο να λάβει τιμή κοντά στο σημείο MPP – έχει χρησιμοποιηθεί από τον αλγόριθμο. Ο παραπάνω τρόπος αυξάνει την ταχύτητα και την απόδοση του αλγορίθμου και αποτελεί μια σημαντική προσθήκη αναβάθμισης της λειτουργίας του.

Πίνακας 3.4. Παράμετροι που χρησιμοποιήθηκαν για τον PSO_advanced

Παράμετρος	Τιμή	Σχόλια
Αριθμός σωματιδίων (N)	5 → 3	Πληθυσμός λύσεων
Αριθμός γενιών (max_iter)	100	Μέγιστος αριθμός γενιών
Μεταβολή ρεύματος (current_step)	0.3	Μέγιστη μεταβολή ρεύματος από βήμα σε βήμα
Χρόνος δειγματοληψίας (mean_time)	15	Συλλογή δείγματος μετά από 15 ms από ορισμένη τιμή

2. **Αρχικοποίηση αλγορίθμου:** Το πρόβλημα που καλείται ο αλγόριθμος να λύσει είναι μονοδιάστατο και η λύση βρίσκεται στο διάστημα $[0, 9.5]$. Συνεπώς, οι αρχικές θέσεις των σωματιδίων μπορούν να λάβουν τυχαίες τιμές στο διάστημα αυτό. Οι αρχικές τιμές των σωματιδίων είναι καθοριστικές για την ταχύτητα σύγκλισης του αλγορίθμου και για αυτό το λόγο τα σωματίδια τοποθετήθηκαν ομοιόμορφα σε ίσες αποστάσεις μεταξύ τους στο διάστημα αυτό, έτσι ώστε να ευνοείται η εξερεύνηση στις αρχικές γενιές.

3. **Αξιολόγηση συνάρτησης καταλληλότητας:** Σκοπός του αλγορίθμου είναι να μεγιστοποιηθεί η παραγόμενη ισχύς. Οι θέσεις των σωματιδίων αντιστοιχούν σε διαφορετικές τιμές ρεύματος, το οποίο μεταφράζεται σε διαφορετικά μεγέθη ισχύος. Για κάθε θέση των σωματιδίων του πληθυσμού γίνεται μέτρηση της τάσης του κυκλώματος και υπολογισμός της παραγόμενης ισχύος.
4. **Ενημέρωση καλύτερης θέσης σωματιδίων και καλύτερης λύσης σε όλο τον πληθυσμό:** Αν κάποιο σωματίδιο πετύχει μεγαλύτερη ισχύ από αυτήν της προηγούμενης θέσης του, τότε η νέα καλύτερη θέση του ενημερώνεται. Υπάρχει δηλαδή μνήμη για κάθε σωματίδιο, στην οποία αποθηκεύεται η βέλτιστη τιμή της θέσης του. Επίσης, γίνεται αποθήκευση της καλύτερης λύσης ολόκληρου του πληθυσμού για όλες τις γενιές. Αυτή η λύση, είναι και η λύση του προβλήματος βελτιστοποίησης (λύση που μεγιστοποιεί την ισχύ που αντλείται από το φωτοβολταϊκό σύστημα). Επίσης, σε κάθε ενημέρωση του πληθυσμού γίνεται έλεγχος για την ανάγκη αφαίρεσης του σωματιδίου με τη χειρότερη λύση – η συγκεκριμένη διαδικασία υλοποιείται στις πρώτες γενιές του αλγορίθμου.
5. **Ενημέρωση θέσης και ταχύτητας σωματιδίων:** Χρησιμοποιώντας τις παρακάτω εξισώσεις υπολογίζεται η νέα θέση και νέα ταχύτητα των σωματιδίων του πληθυσμού.

$$v(t+1) = K [v_i(t) + \varphi_T(p_i(t) - x_i(t))] \quad (3.16)$$

$$x_{t+1}^i = x_t^i + v_{t+1}^i \quad (3.17)$$

όπου οι συντελεστές K και φ_T σύμφωνα με τον πίνακα 3.3.

6. **Έλεγχος ικανοποίησης κριτηρίου τερματισμού:** Στο συγκεκριμένο αλγόριθμο ορίζονται δύο κριτήρια τερματισμού. Το πρώτο είναι ο αριθμός των επαναλήψεων του αλγορίθμου. Το δεύτερο είναι οι ταχύτητες των σωματιδίων. Αν όλες οι ταχύτητες του πληθυσμού είναι κάτω από ένα κατώφλι, τότε ο αλγόριθμος θεωρούμε ότι έχει συγκλίνει. Με την ικανοποίηση μιας από των παραπάνω συνθηκών τερματίζεται η εκτέλεση του αλγορίθμου. Ως βέλτιστη λύση, χρησιμοποιείται η καλύτερη θέση που βρέθηκε για όλο τον πληθυσμό.
7. **Επαναρχικοποίηση:** Ανάλογα με τα περιβαλλοντολογικές συνθήκες που επικρατούν σε μια περιοχή, οι συνθήκες σκίασης των πάνελ, άρα και το ολικό μέγιστο του συστήματος, δύναται να αλλάξει μετά από κάποιο χρονικό διάστημα. Συνεπώς ο αλγόριθμος θα πρέπει να υπολογίσει εκ νέου ποια είναι η μέγιστη ισχύς του συστήματος. Αν η μετρούμενη ισχύς μεταβληθεί ως προς ένα ποσοστό από την προηγούμενη μέτρηση της, τότε ο αλγόριθμος επαναρχικοποιείται και ακολουθείται η διαδικασία από το βήμα 2.

$$\frac{|P_{NEW} - P_{OLD}|}{P_{OLD}} \geq \Delta P \% \quad (3.18)$$

Το διάγραμμα ροής του αλγορίθμου απεικονίζεται στην εικόνα 3.7.

3.3.7 Ψευδό-κώδικας αλγορίθμων

Παρακάτω παρουσιάζεται ο ψευδό-κώδικας των αλγορίθμων PSO και PSO_advanced. Οι διαφορές στον κώδικα των παραπάνω αλγορίθμων εντοπίζονται στη συνάρτηση η οποία ανανεώνει τις θέσεις των σωματιδίων.

Pseudo-code for PSO & PSO_advanced

initialize all required parameters

```
if (particles.velocity > threshold) && (number_of_iteration <= max_iteration)
    broadcast duty (ith particle solution)
    adjust duty (smooth transition from previous duty value to next value)
    if broadcast duty value == real duty value
        increase a counter
        if counter >= mean_time
            update particle's variables (measure p, save V, save I)
            update particle's position
            choose next particle
        if ith particle > N
            update swarm
            choose particle 1
    else
        broadcast best solution
        adjust duty (smooth transition from previous duty to best solution)
        if power changes
            re-initiate population
```

Εικόνα 3.9. Ψευδό-κώδικας αλγορίθμων PSO και PSO_advanced

Κατά την υλοποίηση του κώδικα των παραπάνω αλγορίθμων δημιουργήθηκαν συναρτήσεις με σκοπό να μειώσουν την πολυπλοκότητα και να αυξήσουν την ευρωστία του κώδικα. Στον παρακάτω πίνακα παρουσιάζονται οι συναρτήσεις που χρησιμοποιήθηκαν.

Πίνακας 3.5. Επεξήγηση υλοποιημένων συναρτήσεων για PSO και PSO_advanced

Συνάρτηση	Σχόλια
create_swarm	Δημιουργία και αρχικοποίηση πληθυσμού
adjust_duty	Χρησιμοποιείται για ομαλή μετάβαση από μια τιμή εξόδου του αλγορίθμου σε άλλη (π.χ. από ρεύμα 0.5A → 8 A)
update_particle	Χρησιμοποιείται για την ανανέωση των τιμών των σωματιδίων
update_swarm	Χρησιμοποιείται για την ανανέωση όλου του πληθυσμού

Κεφάλαιο 4ο: Αλγόριθμος DE

4.1 Εισαγωγή

Η διαφορική εξέλιξη (DE) είναι ένας εξελικτικός αλγόριθμος, ο οποίος δημιουργήθηκε από τους Prince και Strom [12] με σκοπό τη βελτιστοποίηση πραγματικών παραμέτρων και συναρτήσεων. Όπως και στους γενετικούς αλγορίθμους, οι πιθανές λύσεις του προβλήματος αναπαρίστανται από διανύσματα τα οποία ονομάζονται χρωμοσώματα.

Το χρωμόσωμα αποτελείται από γονίδια (genes), κάθε ένα από τα οποία κωδικοποιεί την τιμή μιας παραμέτρου του συστήματος.

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \dots, x_{NP,i,G}] \quad (4.1)$$

Αρχικά θεωρείται ένα πλήθος χρωμοσωμάτων, ένας πλήθος δηλαδή λύσεων του προβλήματος που καλείται ο αλγόριθμος DE να επιλύσει. Ο αλγόριθμος είναι επαναληπτικός και εκτελείται σε γενιές. Σε ένα χώρο πιθανών λύσεων, προσπαθεί να βρει τη βέλτιστη λύση στο χώρο αυτό και για αυτό το λόγο θα πρέπει τα χρωμοσώματα κατά την αρχικοποίηση τους, να καταλάβουν όσο το δυνατόν μεγαλύτερο από το χώρο από το χώρο των λύσεων, έτσι ώστε να αυξηθούν οι πιθανότητες ο αλγόριθμος να εντοπίσει τη λύση του προβλήματος. Το κάθε ένα χρωμόσωμα αρχικοποιείται σύμφωνα με την παρακάτω σχέση:

$$x_{j,i,0} = x_{j,min} + rand_{i,j}[0,1] (x_{j,max} - x_{j,min}) \quad (4.2)$$

όπου $rand$ ένας τυχαίος αριθμός στο διάστημα $[0,1]$ και $x_{j,max}, x_{j,min}$ είναι η μέγιστη και η ελάχιστη τιμή του χώρου των λύσεων.

Στη συνέχεια ακολουθεί η *μετάλλαξη* (mutation), κατά την οποία δημιουργείται ένα νέο διάνυσμα που είναι συνδυασμός τριών ή και περισσότερων άλλων τυχαίων διανυσμάτων από όλο τον πληθυσμό. Τα διανύσματα αυτά συνδυάζονται σύμφωνα με τις παρακάτω σχέσεις και δημιουργούν το διάνυσμα *δότη* (donor), $(\vec{V}_{i,G})$. Παρακάτω παρουσιάζονται οι πέντε πιο συνηθισμένες στρατηγικές δημιουργίας του διανύσματος *δότη* :

$$'DE / rand / 1': \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \quad (4.3)$$

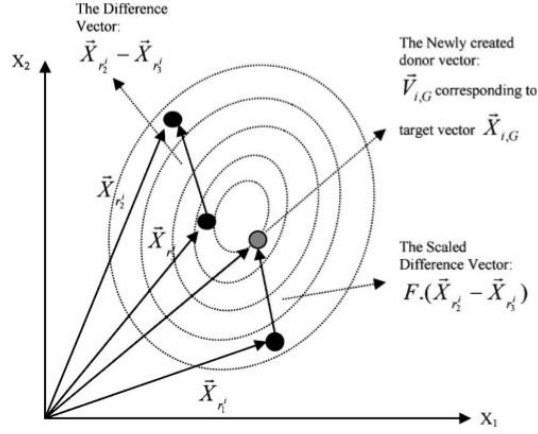
$$'DE / best / 1': \vec{V}_{i,G} = \vec{X}_{best,G} + F (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \quad (4.4)$$

$$'DE / target - to - best / 1': \vec{V}_{i,G} = \vec{X}_{i,G} + F (\vec{X}_{best,G} - \vec{X}_{i,G}) + F (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \quad (4.5)$$

$$'DE / best / 2': \vec{V}_{i,G} = \vec{X}_{best,G} + F (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) + F (\vec{X}_{r_3^i,G} - \vec{X}_{r_4^i,G}) \quad (4.6)$$

$$'DE / rand / 2': \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) + F (\vec{X}_{r_4^i,G} - \vec{X}_{r_5^i,G}) \quad (4.7)$$

όπου F είναι μια παράμετρος η οποία ορίζεται από το χρήστη και $r_1^i, r_2^i, r_3^i, r_4^i, r_5^i$ είναι τυχαίοι αριθμοί στο διάστημα $[1, \text{πλήθος χρωμοσωμάτων (NP)}]$ διαφορετικοί με i . Οι παραπάνω δείκτες παράγονται μια φορά σε κάθε γενιά και είναι διαφορετικοί για κάθε ένα διάνυσμα *δότη* $(\vec{V}_{i,G})$. Τέλος, ο όρος $\vec{X}_{best,G}$ αναφέρεται στο βέλτιστο χρωμόσωμα, το οποίο πετυχαίνει τη μικρότερη ή μεγαλύτερη τιμή της συνάρτησης κόστους σε όλο τον πληθυσμό μέχρι τη προηγούμενη γενιά.



Εικόνα 4.1: Δημιουργία donor διανύσματος [13]

Έχοντας δημιουργήσει το διάνυσμα δότη ($\vec{V}_{i,G}$), στη συνέχεια ακολουθεί η *διασταύρωση* (crossover). Κατά την *διασταύρωση* σχηματίζεται ένα νέο διάνυσμα, το οποίο ονομάζεται *δοκιμαστικό* (trial), ($\vec{U}_{i,G}$) και είναι ο συνδυασμός του διανύσματος δότη και του διανύσματος στόχου (target), ($\vec{X}_{i,G}$). Το διάνυσμα στόχος είναι το διάνυσμα γονέας, το διάνυσμα δηλαδή από το οποίο παράγεται το διάνυσμα του δότη σε κάθε γενιά.

Η διαφορική εξέλιξη (DE) έχει δύο είδη διασταυρώσεων, την *εκθετική* (exponential) και τη *διωνυμική* (binomial). Στην εκθετική αρχικά επιλέγεται ένας ακέραιος n ανάμεσα στο διάστημα $[1,D]$. Ο συγκεκριμένος αριθμός λειτουργεί σαν σημείο έναρξης από το οποίο ξεκινά η *διασταύρωση*. Επίσης, επιλέγεται ένας ακόμα ακέραιος αριθμός L από το διάστημα $[1,D]$. Η τιμή L δείχνει τον αριθμό των στοιχείων των γονιδίων τα οποία θα συμμετέχουν στην *διασταύρωση*. Έχοντας επιλέξει τα n και L το *δοκιμαστικό* διάνυσμα σχηματίζεται ως εξής:

$$\begin{aligned} u_{j,i,G} &= v_{i,j,G} \quad \text{for } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+L-1 \rangle_D \\ x_{j,i,G} & \quad \text{for all other } j \in [1,D] \end{aligned} \quad (4.8)$$

Η σταθερά L υπολογίζεται από τον παρακάτω ψευδοκώδικα:

```

L = 0
do {
    L = L + 1;
}
while (rand ≤ Cr AND L ≤ D)

```

Εικόνα 4.2: Ψευδοκώδικας υπολογισμού σταθεράς L

όπου C_r είναι μια μεταβλητή και ορίζεται από το χρήστη. Για κάθε διάνυσμα δότη ($\vec{V}_{i,G}$), οι τιμές n και L πρέπει να επιλέγονται με τον τρόπο που αναφέρεται παραπάνω.

Τέλος, κατά την *διωνυμική διασταύρωση* υλοποιούνται αλλαγές σε κάθε ένα γονίδιο του χρωμοσώματος. Η *διασταύρωση* πραγματοποιείται σύμφωνα με την παρακάτω σχέση:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } (rand_{i,j}[0,1] \leq C_r \text{ ή } j = j_{rand}) \\ x_{j,i,G}, & \text{else} \end{cases} \quad (4.9)$$

Για να διατηρηθεί ο πληθυσμός των χρωμοσωμάτων σταθερός στις μεταγενέστερες γενιές, το τελευταίο στάδιο που απαιτείται είναι το στάδιο της επιλογής. Μέσα από τη συγκεκριμένη διαδικασία ο αλγόριθμος επιλέγει ποια από τα δοκιμαστικά διανύσματα που έχουν δημιουργηθεί, θα περάσουν στην επόμενη γενιά. Για να περάσει ένα δοκιμαστικό διάνυσμα στην επόμενη γενιά πρέπει να ισχύει:

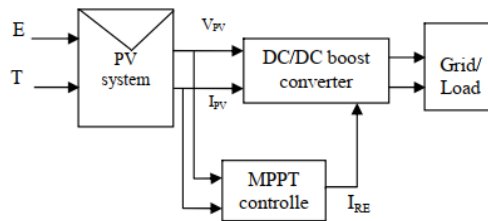
$$\begin{aligned} \vec{X}_{i,G+1} &= \vec{U}_{i,G} & \text{if } f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ \vec{X}_{i,G+1} &= \vec{X}_{i,G} & \text{if } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G}) \end{aligned} \quad (4.10)$$

όπου $f(\vec{X})$ είναι η συνάρτηση κόστους η οποία, σε πρόβλημα ελαχιστοποίησης, πρέπει να ελαχιστοποιηθεί, ενώ αντίστοιχα να μεγιστοποιηθεί για πρόβλημα μεγιστοποίησης. Η παραπάνω εξίσωση μας υποδηλώνει πρόβλημα ελαχιστοποίησης. Έτσι, αν το νέο διάνυσμα που σχηματίστηκε πετύχει μικρότερη τιμή στη συνάρτηση κόστους (στη συγκεκριμένη περίπτωση), τότε αντικαθιστά το αντίστοιχο χρωμόσωμα της παρούσας γενιάς. Οπότε, είτε ο πληθυσμός συγκλίνει σε μια καλύτερη λύση, είτε παραμένει σταθερός χωρίς να μεταβάλλεται.

4.2 Αλγόριθμος DE για MPPT

Από μελέτη σχετικής βιβλιογραφίας προέκυψε το συμπέρασμα ότι ο αλγόριθμος της διαφορικής εξέλιξης έχει χρησιμοποιηθεί για την εύρεση του σημείου MPP, αλλά απαιτείται περεταίρω πειραματισμός έτσι ώστε να εξαχθεί ένα ασφαλές συμπέρασμα για την απόδοσή του [3], [14]. Η μέθοδος που προτείνεται στο [14] υλοποιήθηκε και δοκιμάστηκε πειραματικά και από τα αποτελέσματα εξάχθηκε το συμπέρασμα ότι η απόδοση του αλγορίθμου υπολείπεται των άλλων μεθόδων που χρησιμοποιήθηκαν στην εργασία αυτή. Συνεπώς, δημιουργήθηκε η ανάγκη για βελτίωση της απόδοσης της DE με εφαρμογή στο πεδίο των φωτοβολταϊκών – εύρεση του βέλτιστου σημείου MPP. Οι βελτιώσεις καθώς και ο τρόπος με τον οποίο σχεδιάστηκε και ο υλοποιήθηκε η νέα αυτή έκδοση, παρουσιάζεται στα επόμενα κεφάλαια έπειτα από μια σύντομη αναφορά στην υλοποίηση του άρθρου [14].

Στο [14], υλοποιείται ένας αλγόριθμος για την εύρεση του σημείου MPP, ο οποίος βασίζεται στη διαφορική εξέλιξη (γενετικός αλγόριθμος). Για τον έλεγχο του αλγορίθμου, χρησιμοποιήθηκε προσομοίωση στο περιβάλλον της Matlab.



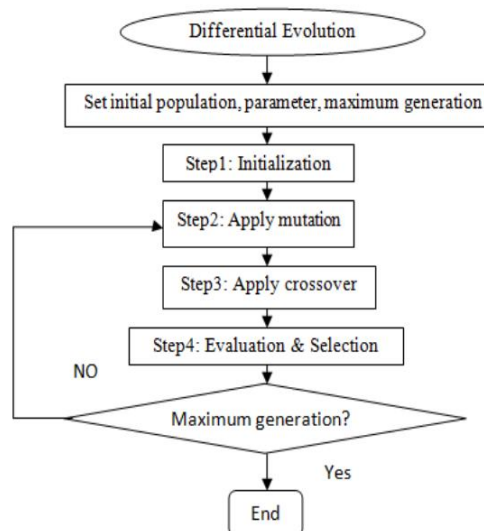
Εικόνα 4.3: Μπλοκ διάγραμμα DE ελεγκτή [14]

4.2.1 Διαφορική εξέλιξη (DE) για MPP

Η διαφορική εξέλιξη ανήκει στους γενετικούς αλγορίθμους και είναι ένας στοχαστικός αλγόριθμος πληθυσμού (stochastic population based optimization algorithm). Ο συγκεκριμένος αλγόριθμος χρησιμοποιήθηκε γιατί παρουσιάζει τρία πλεονεκτήματα:

- 1) Δυνατότητα εντοπισμού, με ακρίβεια, του σημείου MPP ανεξάρτητα από τις αρχικές παραμέτρους του συστήματος (ο PSO για παράδειγμα επηρεάζεται πολύ από τις αρχικές τιμές των παραμέτρων του. Είναι τόσο σημαντικές που μπορεί να αστοχήσει λόγω λανθασμένων αρχικών τιμών).
- 2) Γρήγορη σύγκλιση στο σημείο MPP (απαιτείται μικρός αριθμός επαναλήψεων).
- 3) Μικρός αριθμός των παραμέτρων λειτουργίας του, το οποίο οδηγεί σε εύκολο χειρισμό και ρύθμιση του.

Η διαδικασία εύρεσης του κατάλληλου σημείου MPP, περιλαμβάνει τα στάδια της μετάλλαξης, της διασταύρωσης και της επιλογής. Αρχικά, γίνεται η αρχικοποίηση του πληθυσμού με τυχαίες αρχικές τιμές. Στη συνέχεια υλοποιείται το στάδιο της μετάλλαξης για την εξερεύνηση του χώρου των λύσεων. Μέσω της διαδικασίας της διασταύρωσης, παράγεται μια τροποποιημένη λύση η οποία αν βελτιώσει τη συνάρτηση κόστους, επιλέγεται μέσω της διαδικασίας της επιλογής. Παρακάτω απεικονίζεται το διάγραμμα ροής που υλοποιήθηκε στο άρθρο [14].



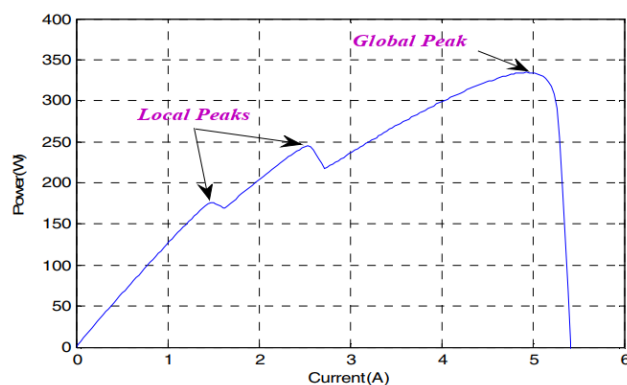
Εικόνα 4.4. Διάγραμμα ροής αλγορίθμου DE [14]

4.2.2 Πειραματισμοί

Για την αξιολόγηση της απόδοσης του αλγορίθμου διενεργήθηκαν δύο σειρές από πειραματισμούς με διαφορετικές συνθήκες σκίασης. Η διασύνδεση των πάνελ με το φορτίο υλοποιήθηκε με τη χρήση ενός DC – DC μετατροπέα. Για τον έλεγχο της λειτουργίας / απόδοσης του αλγορίθμου, χρησιμοποιήθηκε το πάνελ Shell PowerMax Ultra SQ85-P απόδοσης 85 W.

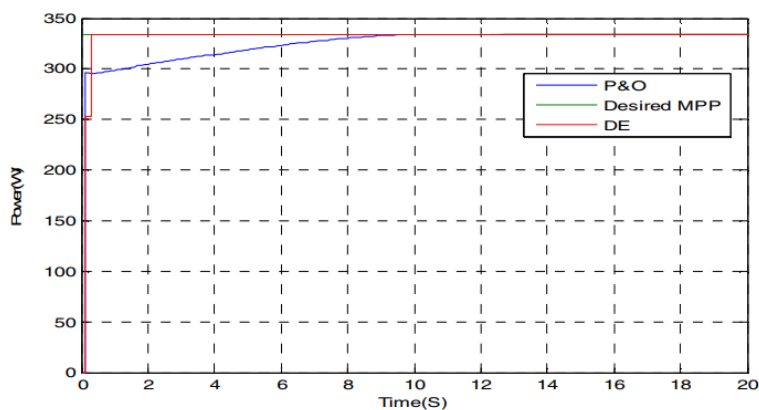
4.2.2.1 1^ο Πείραμα

Στο πείραμα αυτό χρησιμοποιήθηκαν έξι πάνελ συνδεδεμένα σε σειρά. Η ακτινοβολία για τα τέσσερα πρώτα πάνελ ανέρχεται στα 1000 W/m² (πλήρης ηλιοφάνεια), ενώ στα υπόλοιπα δύο πάνελ η ακτινοβολία ανέρχεται στα 500 W/m² και 300 W/m² αντίστοιχα. Στην παρακάτω εικόνα απεικονίζεται η καμπύλη ισχύος – ρεύματος.



Εικόνα 4.5. Καμπύλη ρεύματος – ισχύος (P-I) [14]

Η απόδοση της DE συγκρίθηκε με την απόδοση του P&O, όπου και οι δύο αλγόριθμοι εντόπισαν με επιτυχία το σημείο MPP. Η διαφορά ανάμεσα στις δύο υλοποιήσεις εντοπίζεται στο χρόνο που απαιτείται για τον εντοπισμό του σημείου. Από την εικόνα 4.6 προκύπτει ότι ο P&O εντόπισε το σημείο MPP περίπου στα 10 sec. Αυτό πρακτικά σημαίνει ότι ο συγκεκριμένος αλγόριθμος ενδέχεται να μην εντοπίσει το πραγματικό MPP σημείο σε καταστάσεις με γρήγορες εναλλαγές στην ηλιοφάνεια.



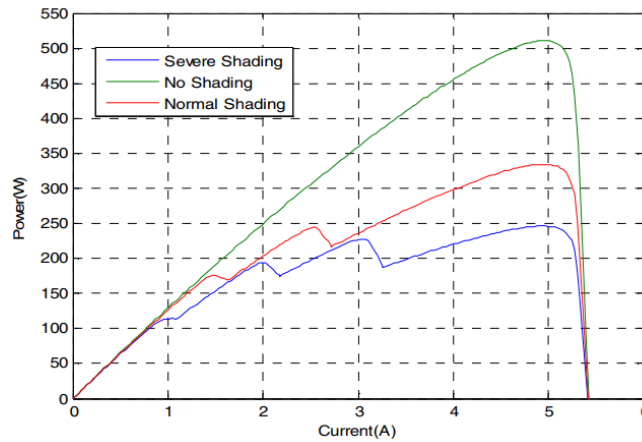
Εικόνα 4.6. Χρόνος σύγκλισης αλγορίθμων DE και P&O [14]

4.2.2.2 2ο Πείραμα

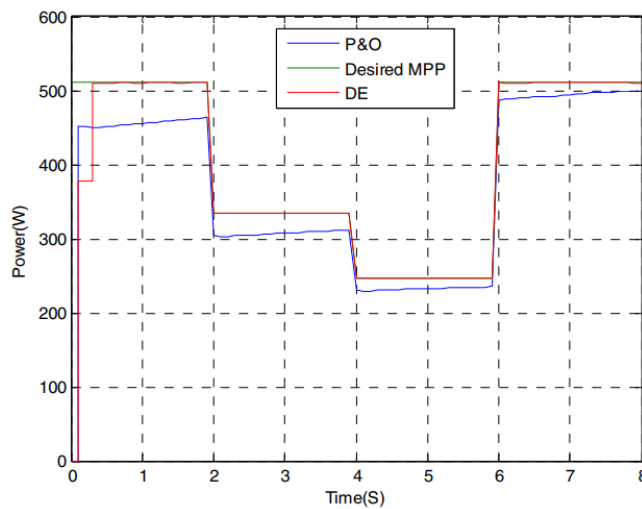
Το συγκεκριμένο πείραμα χρησιμοποιήθηκε για τον έλεγχο της απόκρισης των αλγορίθμων κάτω από δυναμικές συνθήκες σκίασης. Ανάλογα με τις τιμές της ακτινοβολίας στα πάνελ, δημιουργήθηκαν τρεις περιπτώσεις σκίασης. Οι τιμές της ηλιακής ακτινοβολίας που χρησιμοποιήθηκαν για να περιγράψουν τις διαφορετικές περιπτώσεις σκίασης στο συγκεκριμένο πείραμα απεικονίζονται παρακάτω:

Module Number	Irradiance(W/m ²)			
	Step1	Step 2	Step 3	Step4
1 to 3	1000	1000	1000	1000
4	1000	1000	200	1000
5	1000	500	400	1000
6	1000	300	600	1000
Time(Second)	0-2	2-4	4-6	6-8

Εικόνα 4.7. Πίνακας τιμών ηλιακής ακτινοβολίας για 2ο πείραμα [14]



Εικόνα 4.8. Χαρακτηριστικές καμπύλες ισχύος – ρεύματος (P-I) [14]



Εικόνα 4.9. Απόκριση αλγορίθμων DE και P&O σε δυναμικές περιβαλλοντικές συνθήκες [14]

Από τα αποτελέσματα, προέκυψε το συμπέρασμα ότι η DE αποκρίνεται επαρκώς σε δυναμικές συνθήκες σκίασης εντοπίζοντας με επιτυχία το πραγματικό σημείο MPP με μηδενικές ταλαντώσεις γύρω από αυτό. Ο χρόνος που απαιτήθηκε για τον εντοπισμό του MPP σημείου, κατά το πρώτο σετ παραμέτρων για την ηλιακή ακτινοβολία, ήταν 0.3 sec. Αντιθέτως, ο P&O εμφανίζει καθυστέρηση στην απόκριση και αδυναμία εντοπισμού του πραγματικού σημείου MPP. Επίσης, οι συγγραφείς αναφέρουν ότι η απόδοση του αλγορίθμου δεν επηρεάζεται από τις αρχικές παραμέτρους, κάτι το οποίο αποτελεί πλεονέκτημα της συγκεκριμένης μεθόδου έναντι του PSO.

Χρησιμοποιώντας τις πληροφορίες για τη μοντελοποίηση του προβλήματος από το άρθρο [14], παρακάτω παρουσιάζεται μια τροποποιημένη έκδοση της διαφορικής εξέλιξης, η οποία συγκριτικά με τον αλγόριθμο PSO παρουσιάζει υψηλότερη απόδοση. Η νέα έκδοση, ονομάζεται NDE (Neighbor-Based Differential Evolution).

4.3 NBDE

4.3.1 Εισαγωγή

Η υλοποίηση του αλγορίθμου NDE, βασίζεται στο άρθρο των Das και Sil [15] και μια από τις διαφορές του με την κλασική υλοποίηση της DE, εντοπίζεται στη χρήση διαφορετικού τελεστή μετάλλαξης. Χάρης στο διαφορετικό αυτό τελεστή μετάλλαξης, η αρχική λύση ευνοείται στις πρώτες γενιές, αυξάνοντας το ρυθμό σύγκλισης, αλλά σε επόμενες γενιές η συνολική λύση (βέλτιστη λύση) λαμβάνει μεγαλύτερη βαρύτητα, αποφεύγοντας την παγίδευση σε τοπικά ελάχιστα. Έτσι, στα αρχικά στάδια του αλγορίθμου ευνοείται η εξερεύνηση (exploration), ενώ καθώς τα χρωμοσώματα συγκλίνουν σε μια λύση, ευνοείται η εκμετάλλευση (exploitation). Με τον όρο εξερεύνηση (exploration) αναφερόμαστε στην ικανότητα του αλγορίθμου να «εξερευνά» όσο το δυνατόν μεγαλύτερο μέρος του χώρου των λύσεων, ενώ με τον όρο εκμετάλλευση (exploitation) αναφερόμαστε στην ικανότητα του αλγορίθμου να συγκλίνει σε βέλτιστες λύσεις όσο το δυνατόν γρηγορότερα. Η παραλλαγή του DE, DE/target-to-best/1 (εξίσωση 5.5) χρησιμοποιεί το καλύτερο χρωμόσωμα του πληθυσμού για να δημιουργήσει το διάνυσμα δότη ($\vec{V}_{i,G}$). Η συγκεκριμένη παραλλαγή ευνοεί την εκμετάλλευση, αφού όλοι οι απόγονοι «έλκονται» προς τη θέση του καλύτερου χρωμοσώματος στο χώρο των λύσεων, με αποτέλεσμα όλα τα χρωμοσώματα να συγκλίνουν πολύ γρήγορα στη συγκεκριμένη θέση. Αυτό, όμως, έχει ως αποτέλεσμα πολλές φορές ο πληθυσμός να συγκλίνει σε τοπικά ελάχιστα/μέγιστα αδυνατώντας να εξερευνήσει όλο το χώρο των λύσεων. Επίσης, σε συνδυασμό με τη διαφορά των διανυσμάτων ($\vec{X}_{r_1,G} - \vec{X}_{r_2,G}$), όταν ο πληθυσμός έχει συγκλίνει σε μια μικρή περιοχή, δεν επιτρέπει την περαιτέρω εξερεύνηση γειτονικών περιοχών, με αποτέλεσμα ο πληθυσμός να εγκλωβίζεται σε τοπικά ελάχιστα ή μέγιστα, ανάλογα με το πρόβλημα που καλείται να αντιμετωπίσει ο αλγόριθμος. Συνεπώς, πρέπει να υπάρχει μια ισορροπία μεταξύ της εξερεύνησης και της εκμετάλλευσης, έτσι ώστε ο αλγόριθμος να λειτουργεί αποδοτικά.

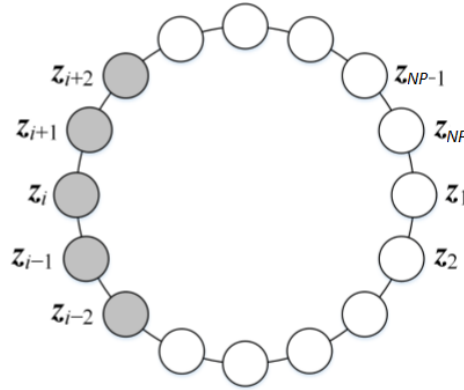
Για να ξεπεραστούν τέτοιου είδους προβλήματα, εισάγεται μια νέα μεθοδολογία θεωρώντας έναν πιο εξελιγμένο τελεστή μετάλλαξης. Σύμφωνα με την νέα αυτή μεθοδολογία, θεωρούνται δύο μοντέλα γειτονιών στις οποίες ανήκουν τα χρωμοσώματα. Η πρώτη γειτονιά ονομάζεται τοπική γειτονιά (local neighborhood), όπου κάθε χρωμόσωμα σε αυτήν μεταλλάσσεται βάσει του καλύτερου χρωμοσώματος στην τοπική αυτή γειτονιά. Η δεύτερη γειτονιά ονομάζεται καθολική γειτονιά (global neighborhood) και αναφέρεται σε όλο τον πληθυσμό, θεωρώντας καλύτερο χρωμόσωμα σε ολόκληρο τον πληθυσμό. Σύμφωνα με τη συγκεκριμένη μεθοδολογία υπάρχει μια καθυστέρηση στη μετάδοση της πληροφορίας από γενιά σε γενιά, όσον αφορά την καλύτερη θέση του πληθυσμού, με αποτέλεσμα να υπάρχει μια ισορροπία μεταξύ των δύο παραπάνω εννοιών, της εκμετάλλευσης και της εξερεύνησης. Παρακάτω παρουσιάζεται αναλυτικά η αναφερόμενη μέθοδος.

4.3.2 Κωδικοποίηση χρωμοσωμάτων

Θεωρείται αρχικά ένας πληθυσμός χρωμοσωμάτων $[\vec{Z}_1, \vec{Z}_2, \dots, \vec{Z}_{NP}]$, όπου $\vec{Z}_i$ ($i = 1, 2, \dots, NP$) ένα μονοδιάστατο διάνυσμα. Ο πληθυσμός, για την παρούσα εργασία αποτελείται από $N = 5$ χρωμοσώματα και αρχικοποιείται με τυχαίες τιμές στο διάστημα $[0, 9.5]$.

4.3.3 Μετάλλαξη, διασταύρωση και συνάρτηση κόστους

Η συγκεκριμένη μέθοδος θεωρεί ένα διαφορετικό τελεστή μετάλλαξης. Για κάθε ένα χρωμόσωμα, από ένα πληθυσμό $\vec{Z}_i$, ορίζεται μια γειτονία ακτίνας k . Η γειτονία αποτελείται από $2k+1$ χρωμοσώματα $N_i = \{\vec{Z}_{i-k}, \dots, \vec{Z}_i, \dots, \vec{Z}_{i+k}\}$. Το σύνολο του πληθυσμού βρίσκεται σε *τοπολογία δακτυλίου* (ring topology) ως προς τους δείκτες, έτσι ώστε τα χρωμοσώματα $\vec{Z}_{NP}$ και $\vec{Z}_2$ να είναι άμεσοι γείτονες του $\vec{Z}_1$. Η συγκεκριμένη τοπολογία φαίνεται στην παρακάτω εικόνα για $k=2$ και $NP=16$. Στην παρούσα εργασία χρησιμοποιήθηκαν 10 χρωμοσώματα σε όλους τους πειραματισμούς που εκτελέστηκαν. Συνεπώς το $k=10$, δηλαδή αποτελεί το 10 % του πληθυσμού όπως προτείνεται και στην αναφορά [15].



Εικόνα 4.10: Τοπολογία δακτυλίου στο πληθυσμό των χρωμοσωμάτων [15]

Η θεωρητική αυτή σειρά των χρωμοσωμάτων καθορίζεται από τους δείκτες τους στον αρχικό πληθυσμό. Η ειδική μετάλλαξη που γίνεται στον NBDE πραγματοποιείται αρχικά για κάθε χρωμόσωμα $\vec{Z}_i$ και παράγει ένα νέο διάνυμα $\vec{L}_i$, λαμβάνοντας υπόψη μόνο τη τοπική γειτονιά του (local neighborhood) :

$$\vec{L}_i = \vec{Z}_i + \alpha(\vec{Z}_{best} - \vec{Z}_i) + \beta(\vec{Z}_p - \vec{Z}_q) \quad (4.11)$$

όπου α και β οι συντελεστές κλιμάκωσης, p και q τυχαίοι δείκτες εντός του διαστήματος $[i-k, i+k]$, για τους οποίους ισχύει $p \neq q \neq i$ και $\vec{Z}_{best}$ το βέλτιστο χρωμόσωμα της γειτονιάς N_i . Ταυτόχρονας, δημιουργείται και η μεταλλαγμένη λύση από ολόκληρο τον πληθυσμό:

$$\vec{g}_i = \vec{Z}_i + \alpha(\vec{G}_{best} - \vec{Z}_i) + \beta(\vec{Z}_{r_1} - \vec{Z}_{r_2}) \quad (4.12)$$

όπου r_1 και r_2 τυχαίοι δείκτες στο διάστημα $[1, NP]$, έτσι ώστε $r_1 \neq r_2 \neq i$ και $\vec{G}_{best}$ το βέλτιστο χρωμόσωμα ολόκληρου του πληθυσμού της προηγούμενης γενιάς. Το συνολικό μεταλλαγμένο διάνυμα *δότη* $\vec{V}_i$ προέρχεται από το βεβαρημένο μέσο όρο του τοπικού και του καθολικού διανύσματος:

$$\vec{V}_i = w\vec{g}_i + (1-w)\vec{L}_i \quad (4.13)$$

όπου w ένα βάρος που αυξάνεται γραμμικά σε κάθε γενιά t , από μια κατώτερη τιμή w_{min} ως μια μέγιστη w_{max} :

$$w = w_{min} + (w_{max} - w_{min}) \frac{t - 1}{I_{max} - 1} \quad (4.14)$$

Οι ακραίες τιμές προτείνεται να τίθενται ως $w_{min} = 0.1$ και $w_{max} = 0.5$, [15]. Με αυτόν τον τρόπο, η τοπική λύση ευνοείται στην αρχή της διαδικασίας αυξάνοντας τον ρυθμό σύγκλισης, αλλά σε επόμενες γενιές η συνολική λύση λαμβάνει μεγαλύτερη βαρύτητα, αποφεύγοντας την παγίδευση σε τοπικά ελάχιστα. Εκτός από τον ειδικό τελεστή μετάλλαξης, η μεθοδολογία NBDE είναι ίδια με την τυπική μεθοδολογία του DE.

Ο NBDE εφαρμόζει διωνυμική (binomial) διασταύρωση, η οποία επιτελείται εφαρμόζοντας σε κάθε γονίδιο $n = 1, \dots, D$ τη σχέση:

$$u_{i,n} = \begin{cases} v_{i,n}, & \text{αν } rand(0,1) < Cr \text{ ή } n = j_{rand} \\ z_{i,n}, & \text{διαφορετικά} \end{cases} \quad (4.15)$$

Στην παραπάνω σχέση, j_{rand} ένας τυχαίος αριθμός στο διάστημα $[1,D]$, ενώ με $rand(0,1)$ συμβολίζεται μια τυχαία δεκαδική τιμή στο διάστημα $[0,1]$. Επιπλέον, $C_r \in [0,1]$ είναι η πιθανότητα διασταύρωσης. Ο j_{rand} επιλέγεται μια φορά κατά τη δημιουργία του δοκιμαστικού χρωμοσώματος $\vec{U}_i$, ενώ η συνάρτηση $rand(0,1)$ καλείται εκ νέου για κάθε ένα γονίδιο του χρωμοσώματος. Ο όρος $n = j_{rand}$ στην πρώτη συνθήκη εξασφαλίζει ότι τουλάχιστον ένα γονίδιο θα προέρχεται από το μεταλλαγμένο δοκιμαστικό χρωμόσωμα. Για την υλοποίηση της διωνυμικής διασταύρωσης απαιτείται μια ειδική διαδικασία, μιας και το πρόβλημα που καλείται να επιλύσει ο αλγόριθμος είναι μονοδιάστατο και ολόκληρο το χρωμόσωμα περιγράφεται από ένα δεκαδικό αριθμό. Η ειδική αυτή διαδικασία περιγράφεται παρακάτω.

Τέλος, ως συνάρτηση κόστους θεωρείται η ισχύς, το γινόμενο δηλαδή της τάσης που παράγει η φωτοβολταϊκή μονάδα με το ρεύμα από την πηγή ρεύματος. Ο αλγόριθμος καλείται να μεγιστοποιήσει την τιμή της ισχύος και συνεπώς το πρόβλημα ανάγεται σε πρόβλημα μεγιστοποίησης της συνάρτησης κόστους.

$$P = V \times I \quad (4.16)$$

4.3.4 Υλοποίηση διωνυμικής διασταύρωσης

Για την υλοποίηση της διωνυμικής διασταύρωσης στο συγκεκριμένο πρόβλημα απαιτείται μια ειδική διαδικασία, που είναι καταλυτική για την απόδοση του αλγορίθμου. Η διαδικασία αυτή περιγράφεται αναλυτικά παρακάτω.

Κατά τη δημιουργία του δοκιμαστικού χρωμοσώματος $\vec{U}_i$, δημιουργείται το δοκιμαστικό διάνυσμα, το οποίο αποτελεί ένα συνδυασμό του διανύσματος δότη $\vec{V}_i$ και του διανύσματος γονέα $\vec{Z}_i$. Στο συγκεκριμένο πρόβλημα τα διανύσματα είναι ένας αριθμός στο διάστημα $[0, 9.5]$. Συνεπώς, είναι αδύνατο να εφαρμοστεί η σχέση (4.15) για διανύσματα με ένα δεκαδικό αριθμό. Για το λόγο αυτό τα διανύσματα μετατρέπονται στη δυαδική τους μορφή, έτσι ώστε να είναι δυνατή η εφαρμογή της διωνυμικής διασταύρωσης. Για την καλύτερη κατανόηση της μεθόδου αυτής παρακάτω παρουσιάζεται ένα παράδειγμα.

Έστω ότι η τιμή του χρωμοσώματος γονέα για μια γενιά i είναι $\vec{Z}_i = 6.325$ και η τιμή του χρωμοσώματος δότη $\vec{V}_i = 3.125$. Η διαδικασία για τη δημιουργία του δοκιμαστικού διανύσματος είναι η παρακάτω:

1. Η τιμή των δύο διανυσμάτων πολλαπλασιάζεται με τον αριθμό 10000 και στρογγυλοποιείται. Αυτό γίνεται για να μην χάσουμε ακρίβεια από τις πιθανές δεκαδικές τιμές που θα έχουν τα χρωμοσώματα. Στη συνέχεια οι τιμές μεταφράζονται στη δυαδική μορφή τους με συνολικό αριθμό 24 ψηφίων.

$$\begin{aligned}\vec{Z}_i = 6.325 &\rightarrow \text{round}(6.325 * 10000) \\ &= 63250 \xrightarrow{\text{bin}} 000000001111011100010010\end{aligned}\quad (4.17)$$

$$\begin{aligned}\vec{V}_i = 3.125 &\rightarrow \text{round}(3.125 * 10000) \\ &= 31250 \xrightarrow{\text{bin}} 000000000111101000010010\end{aligned}\quad (4.18)$$

2. Στη συνέχεια επιλέγεται ένας τυχαίος αριθμός στο διάστημα $[0, \text{μήκος χρωμοσωμάτων}]$. Έστω ότι στη συγκεκριμένη περίπτωση ο αριθμός είναι το 20. Έχοντας επιλέξει μια τιμή για το συντελεστή CR εφαρμόζεται η παρακάτω συνθήκη:

$$\text{mask}_{i,n} = \begin{cases} 1, & \text{αν } \text{rand}(0,1) < Cr \\ 0, & \text{διαφορετικά} \end{cases}\quad (4.19)$$

Στο συγκεκριμένο παράδειγμα, έστω ότι η μάσκα έχει την παρακάτω μορφή:

$$\text{mask}_{i,n} = 001010111100101110011100\quad (4.20)$$

3. Λαμβάνοντας υπόψη το παραπάνω βήμα, η σχέση () δύναται να μετασχηματιστεί στην παρακάτω νέα της μορφή:

$$u_{i,n} = \begin{cases} v_{i,n}, & \text{αν } \text{mask}_{i,n} = 1 \\ z_{i,n}, & \text{διαφορετικά} \end{cases}\quad (4.21)$$

Το δοκιμαστικό χρωμόσωμα θα αποτελείται από τα παρακάτω τμήματα των χρωμοσωμάτων του δότη και του γονέα:

$$u_{i,n} = \begin{cases} 000000001111011100010010 \\ 0000000000111101000010010 \end{cases}\quad (4.22)$$

4. Η δυαδική τιμή του δοκιμαστικού χρωμοσώματος είναι η:

$$u_{i,n} = 000000001111001100010010\quad (4.23)$$

5. Στη συνέχεια η τιμή μετατρέπεται σε δεκαδική και διαιρείται με το 10000, έτσι ώστε να τοποθετηθεί στο διάστημα $[0, 9.5]$

$$u_{i,n} = \frac{\text{bin2dec}('000000001111001100010010')}{10000} = 6.2226\quad (4.24)$$

Σκοπός του παραπάνω παραδείγματος είναι η κατανόηση της διαδικασίας υλοποίησης της διωνυμικής διασταύρωσης, η οποία αναπτύχθηκε και χρησιμοποιήθηκε αποκλειστικά για το συγκεκριμένο πρόβλημα βελτιστοποίησης. Μια διαφορετική επιλογή θα ήταν η κωδικοποίηση των αρχικών τιμών των χρωμοσωμάτων απευθείας στη δυαδική τους μορφή, με σκοπό την παράλειψη της

παραπάνω διαδικασίας. Τα οφέλη και τα πλεονεκτήματα της απευθείας κωδικοποίησης των χρωμοσωμάτων στη δυαδική τους μορφή δεν έχουν αξιολογηθεί.

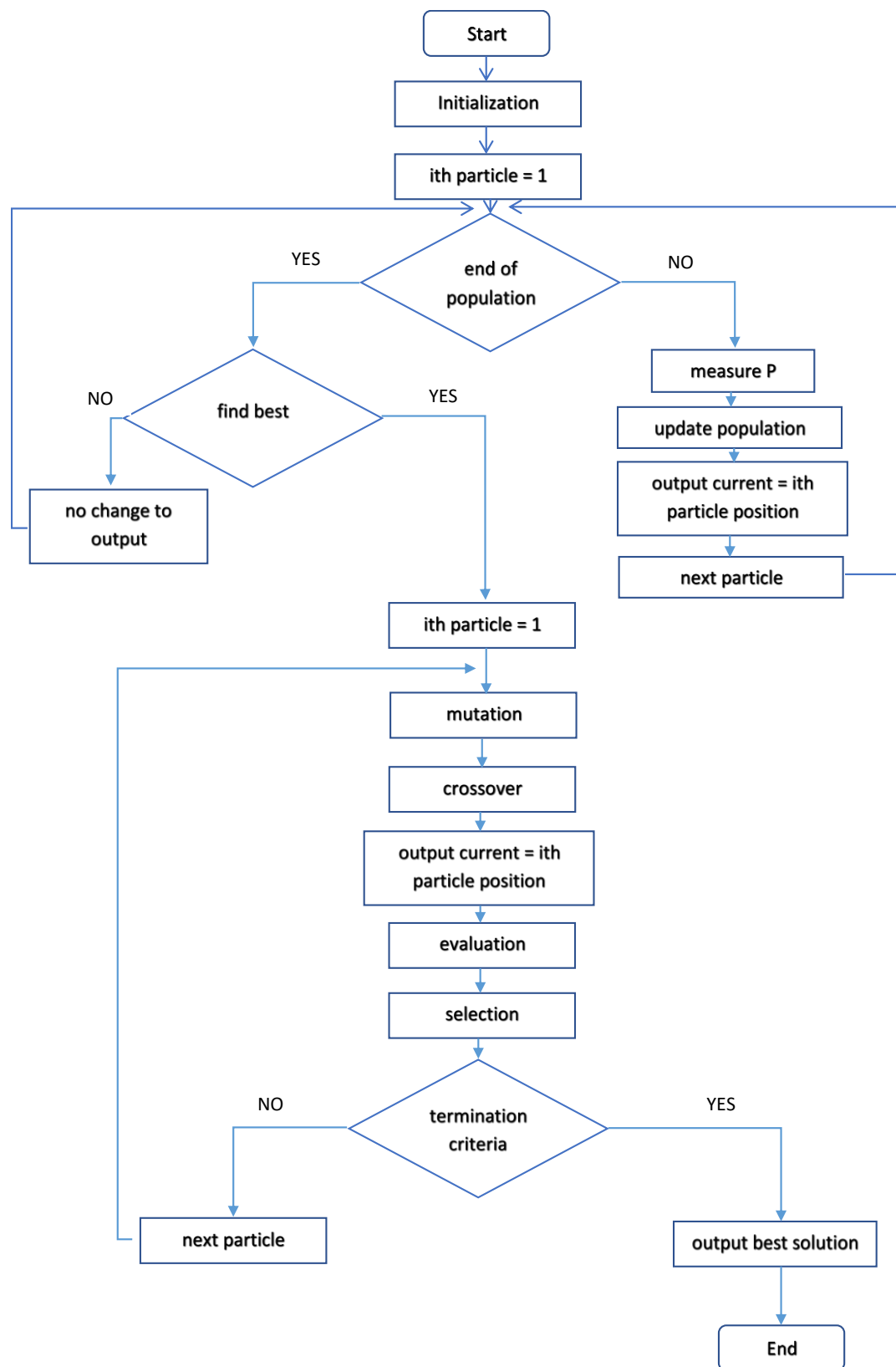
4.3.5 Διαδικασία επιλογής

Το δοκιμαστικό χρωμόσωμα $\vec{U}_i$ αντικαθιστά το αρχικό $\vec{Z}_i$ στον πληθυσμό της επόμενης γενιάς μόνο αν χαρακτηρίζεται από υψηλότερη τιμή της συνάρτησης κόστους (υψηλότερη τιμή ισχύος). Στον NBDE θεωρείται πρόβλημα μεγιστοποίησης και ισχύει η παρακάτω σχέση:

$$\vec{Z}_i \leftarrow \begin{cases} \vec{U}_i, & \text{αν } f(\vec{U}_i) > f(\vec{Z}_i) \\ \vec{Z}_i, & \text{αν } f(\vec{U}_i) \leq f(\vec{Z}_i) \end{cases} \quad (4.25)$$

Η παραπάνω διαδικασία εκτελείται για όλα τα χρωμοσώματα του πληθυσμού και επαναλαμβάνεται για ένα μέγιστο αριθμό επαναλήψεων $\max_iter$.

4.3.6 Διάγραμμα ροής



Εικόνα 4.11. Διάγραμμα ροής αλγορίθμου DE

4.3.7 Ψευδό – κώδικας

Παρακάτω παρουσιάζεται ο ψευδό – κώδικας του αλγορίθμου NBDE

Pseudo-code for NBDE

```
initialize all required parameters

if (not termination criteria) and (not main_routine)

    broadcast duty (ith particle solution)
    adjust duty (smooth transition from previous duty value to next value)
    update population
    choose next particle
    if end of population
        main_routine = 1;
        current_individual = 1;
    end

if main_routine
    create local mutation
    create global mutation
    create donor particle from local and global mutation
    create trial particle
    broadcast duty (trial particle solution)
    adjust duty (smooth transition from previous duty value to next value)
    evaluation of trial particle
    selection trial particle
    check termination criteria
    choose next particle
else
    broadcast best solution
end
```

Εικόνα 4.12: Ψευδοκώδικας αλγορίθμου DE

Κατά την υλοποίηση του κώδικα NBDE δημιουργήθηκαν συναρτήσεις με σκοπό να μειώσουν την πολυπλοκότητα και να αυξήσουν την ευρωστία του κώδικα. Στον παρακάτω πίνακα παρουσιάζονται οι συναρτήσεις και οι τιμές των μεταβλητών που χρησιμοποιήθηκαν.

4.3.8 Πίνακας τιμών που χρησιμοποιήθηκαν

Πίνακας 4.1. Επεξήγηση υλοποιημένων συναρτήσεων για PSO και PSO_advanced

Συνάρτηση	Σχόλια
n_population	Πληθυσμός χρωμοσωμάτων
CR	Συντελεστής CR – τυχαία τιμή που ορίζεται από το χρήστη

a	Συντελεστής κλιμάκωσης
b	Συντελεστής κλιμάκωσης
k	Ακτίνα γειτονιάς
w_min	Κάτω όριο συντελεστή στάθμισης χρωμοσώματος δότη
w_max	Άνω όριο συντελεστή στάθμισης χρωμοσώματος δότη
duty_step	Μέγιστη εναλλαγή ρεύματος (αποτέλεσμα αλγορίθμου) ανά κύκλο εκτέλεσης αλγορίθμου
mean_time	Χρόνος αποκατάστασης ρεύματος – δειγματοληψία μετά το πέρας του συγκεκριμένου χρονικού διαστήματος
max_iter	Αριθμός επαναλήψεων αλγορίθμου
crossover	Υλοποιεί τη διαδικασία διασταύρωσης
adjust_duty	Χρησιμοποιείται για ομαλή μετάβαση από μια τιμή εξόδου του αλγορίθμου σε άλλη (π.χ. από ρεύμα 0.5A → 8 A)
create_population	Χρησιμοποιείται για την αρχικοποίηση του πληθυσμού

Κεφάλαιο 5ο: Ant Inspired Algorithm

5.1 Εισαγωγή

Ο συγκεκριμένος αλγόριθμος θα μπορούσε να θεωρηθεί μια βελτιωμένη έκδοση του αλγορίθμου ACO (Ant Colony Optimization) [16]. Ωστόσο, οι δύο αλγόριθμοι διαφέρουν αρκετά και μοιράζονται ελάχιστα κοινά σημεία. Για τις ανάγκες της εργασίας, αρχικά, δημιουργήθηκε ο αλγόριθμος ACO ο οποίος παρουσίασε πολύ χαμηλή απόδοση. Τροποποιώντας κάποια από τα τμήματα του, δημιουργήθηκε ο αλγόριθμος AIA, ο οποίος παρουσιάζει αξιολογα αποτελέσματα. Για τον παραπάνω λόγο κρίθηκε ότι δεν είναι απαραίτητη η επεξήγηση του αλγορίθμου ACO, αλλά απευθείας του AIA.

5.2 Σύλληψη ιδέας για δημιουργία αλγορίθμου

Μετά από υλοποίηση και προσομοίωση πολλών και διαφορετικών αλγορίθμων MPP, εξάχθηκε το συμπέρασμα ότι η απόδοση του αλγορίθμου εξαρτάται από το σχεδιασμό τριών επιμέρους τμημάτων. Τα τμήματα αυτά είναι:

1. Τμήμα αρχικοποίησης του αλγορίθμου.
2. Τμήμα ανανέωσης λύσεων – ανανέωση θέσης των ατόμων του πληθυσμού
3. Τμήμα διαχείρισης και απόρριψης λύσεων.

Κάθε ένα από τα παραπάνω τμήματα επηρεάζει την ταχύτητα σύγκλισης και την απόδοση του συστήματος. Η αρχικοποίηση, για παράδειγμα, είναι υπεύθυνη για το πόσο κοντά στο σημείο MPP τοποθετούνται οι αρχικές λύσεις. Μεγαλύτερη απόσταση μεταφράζεται σε ανάγκη περισσότερων βημάτων εκτέλεσης, το οποίο με τη σειρά του μεταφράζεται σε αυξημένη χρονική διάρκεια εκτέλεσης του αλγορίθμου. Επίσης, κατά τη διάρκεια λειτουργίας του αλγορίθμου, υπάρχουν λύσεις οι οποίες δεν επιφέρουν αύξηση στην απόδοση του αλγορίθμου και απαιτούν ειδικό χειρισμό. Υπάρχουν αλγόριθμοι όπως ο PSO, που χρησιμοποιούν τη χειρότερη λύση του πληθυσμού, σε συνδυασμό με τη βέλτιστη, για τη δημιουργία νέας λύσης υψηλότερης απόδοσης [11]. Άλλοι αλγόριθμοι, όπως ο ACO [16], δημιουργεί νέες λύσεις τις οποίες προσθέτει στον πληθυσμό και στη συνέχεια επιλέγει ορισμένες από αυτές για την επόμενη γενιά. Συνεπώς, ο τρόπος με τον οποίο γίνεται η διαχείριση των ατόμων – λύσεων επηρεάζει την ταχύτητα του αλγορίθμου. Εντούτοις, το πιο σημαντικό τμήμα του αλγορίθμου είναι ο μηχανισμός με τον οποίο γίνεται η ανανέωση – το βήμα – των ατόμων του πληθυσμού από γενιά σε γενιά. Στόχος των μηχανισμών ανανέωσης, είναι η τοποθέτηση μελλοντικών λύσεων όσο το δυνατόν πιο κοντά στο σημείο MPP. Στα προηγούμενα κεφάλαια έγινε περιγραφή του τρόπου ανανέωσης του πληθυσμού που ακολουθεί κάθε μέθοδος. Γίνεται εύκολα αντιληπτό ότι υπάρχουν πολλοί και διαφορετικοί τρόποι για υλοποίηση του συγκεκριμένου βήματος, γεγονός που υποδηλώνει έναν τομέα με αυξημένο επιστημονικό ενδιαφέρον.

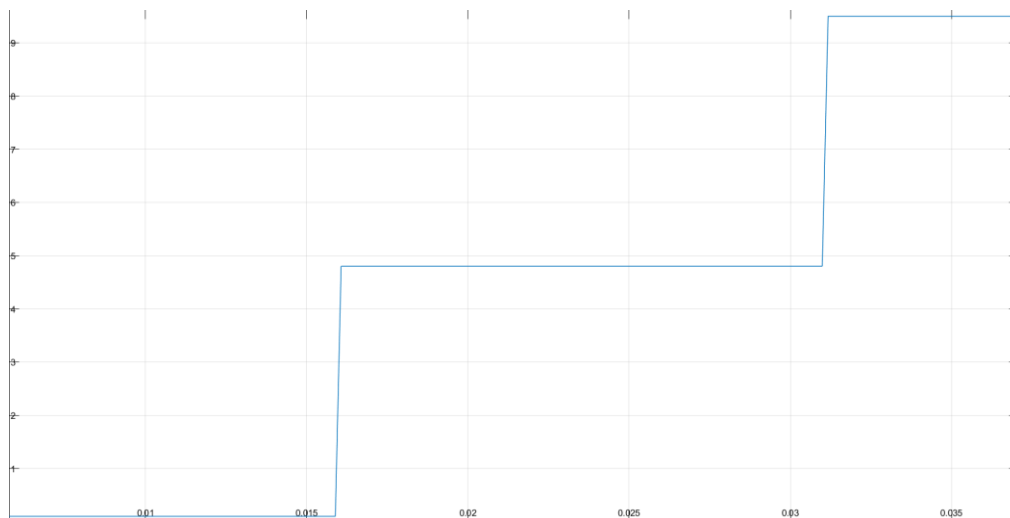
Στα παρακάτω κεφάλαια γίνεται παρουσίαση της μεθοδολογίας η οποία ακολουθήθηκε, έτσι ώστε να υλοποιηθεί ο αλγόριθμος AIA. Στη συνέχεια, ακολουθεί το διάγραμμα ροής του αλγορίθμου, ο ψευδό – κώδικας και τέλος, ο πίνακας με τις τιμές και τις συναρτήσεις που χρησιμοποιήθηκαν.

5.3 Αρχικοποίηση αλγορίθμου

Το πρώτο τμήμα του αλγορίθμου, θεωρείται αυτό της αρχικοποίησης. Στο συγκεκριμένο τμήμα υλοποιείται η τοποθέτηση των σωματιδίων – χρωμοσωμάτων – ατόμων του πληθυσμού, στο χώρο των λύσεων. Στόχος του συγκεκριμένου τμήματος είναι η τοποθέτηση των σωματιδίων όσο το δυνατόν πιο κοντά στη βέλτιστη λύση του συστήματος. Οι πιθανότητες σύγκλισης του αλγορίθμου στη βέλτιστη λύση αυξάνονται με την αύξηση του ατόμων του πληθυσμού. Εντούτοις, με την αύξηση των ατόμων

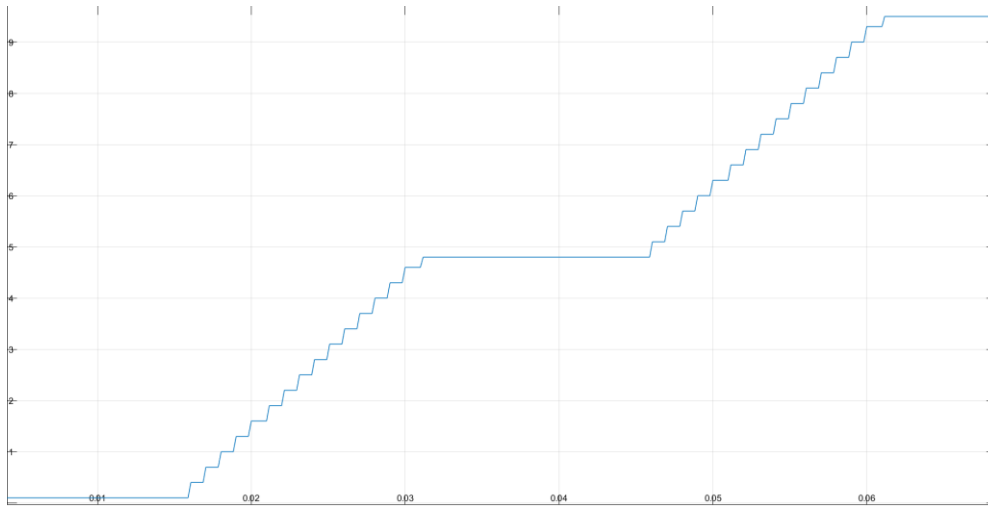
του πληθυσμού αυξάνεται και ο χρόνος σύγκλισης του αλγορίθμου, το οποίο είναι απαγορευτικό σε ένα σύστημα που μεταβάλλονται οι παράμετροι λειτουργίας του σύμφωνα με τις περιβαλλοντικές συνθήκες. Για το λόγο αυτό, είναι απαραίτητη η εφαρμογή μιας στρατηγικής, η οποία αυξάνει ταυτόχρονα τις πιθανότητες τοποθέτησης ενός ατόμου του πληθυσμού κοντά στη βέλτιστη λύση και δεν επιβαρύνει χρονικά την εκτέλεση του αλγορίθμου, ή την επιβαρύνει ελάχιστα. Ο τεχνική που εφαρμόστηκε για την αρχικοποίηση του αλγορίθμου περιγράφεται παρακάτω.

Αρχικά, θεωρείται ένας πληθυσμός με 3 άτομα – χρωμοσώματα – σωματίδια. Χάριν απλότητας και συντομίας θα χρησιμοποιείται ο όρος “άτομα” από εδώ και στο εξής. Ο πληθυσμός πρέπει να καλύψει όσο το δυνατόν περισσότερο χώρο από αυτό των λύσεων, για να αυξηθεί η πιθανότητα να τοποθετηθεί κάποιο από τα άτομα του πληθυσμού κοντά στην πραγματική λύση του προβλήματος. Στο διάστημα $[0, 9.5]$, μια πιθανή αρχικοποίηση θα τοποθετούσε τα άτομα του πληθυσμού στις τιμές $[0, 4.75, 9.5]$. Το διάγραμμα της αρχικοποίησης θα είχε την παρακάτω μορφή:



Εικόνα 5.1. Αρχικοποίηση πληθυσμού 3 ατόμων στο διάστημα $[0, 9.5]$

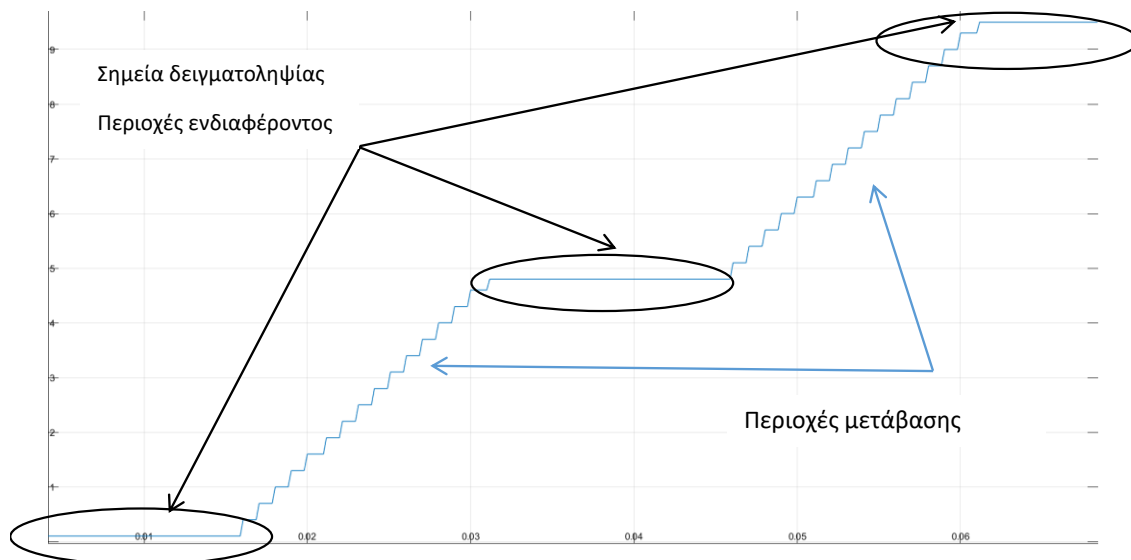
Η παραπάνω εικόνα (εικόνα 5.1) απεικονίζει την μετάβαση μεταξύ των τιμών του πληθυσμού. Ο αλγόριθμος στη συγκεκριμένη φάση εφαρμόζει τις τιμές του ρεύματος των σωματιδίων στην έξοδο του και μετράει την τιμή της ισχύος που αντλείται από το φωτοβολταϊκό σύστημα. Υπενθυμίζεται ότι σκοπός της προσομοίωσης είναι η επαλήθευση της λειτουργίας του αλγορίθμου και ο τελικός στόχος είναι η τοποθέτηση των αλγορίθμων που δημιουργήθηκαν σε πραγματικές συνθήκες λειτουργίας. Στην προσομοίωση που χρησιμοποιήθηκε στην παρούσα εργασία, αφαιρέθηκε για λόγους απλότητας ο DC – DC μετατροπέας. Η έξοδος του συστήματος, είναι το ρεύμα το οποίο επιβάλλεται απευθείας στη φωτοβολταϊκή μονάδα. Ωστόσο, παρουσία DC – DC μετατροπέα, η έξοδος του αλγορίθμου θα είναι το εύρος ζώνης του μετατροπέα. Συνεπώς, μια άμεση εναλλαγή της τιμής – όπως αυτή της εικόνας XX - στο εύρος ζώνης του μετατροπέα ισχύος, θα δημιουργούσε προβλήματα στη λειτουργία του. Με γνώμονα την εφαρμογή των αλγορίθμων της εργασίας αυτής σε πραγματικό σύστημα, αναθεωρήθηκε ο τρόπος με τον οποίο γίνεται η εκπομπή των τιμών μεταξύ των ατόμων του πληθυσμού. Για την καλύτερη προετοιμασία των αλγορίθμων στην εφαρμογή τους σε πραγματικές συνθήκες, έγινε εισαγωγή ενός βήματος, το οποίο εναλλάσσει σταδιακά την έξοδο του αλγορίθμου έως η έξοδος φθάσει στην επιθυμητή τιμή. Εφαρμόζοντας την συγκεκριμένη τεχνική, εισάγεται μια χρονική καθυστέρηση στο σύστημα, που είναι όμως απαραίτητη για την εύρυθμη λειτουργία του συστήματος σε πραγματικές συνθήκες.



Εικόνα 5.2. Αρχικοποίηση πληθυσμού 3 ατόμων στο διάστημα $[0, 9.5]$ με χρήση βήματος

Στο σημείο αυτό κρίνεται απαραίτητο να αναφερθεί, ότι ο λόγος για τον οποίο έγινε η επιλογή της συγκεκριμένης μεθόδου αρχικοποίησης, είναι ότι έπειτα από αρκετούς πειραματισμούς, αποδείχθηκε ότι προσφέρει τα πιο αξιόπιστα αποτελέσματα. Μια διαφορετική τεχνική αρχικοποίησης θα ήταν η τοποθέτηση των ατόμων του πληθυσμού σε τυχαίες θέσεις του χώρου των λύσεων, ωστόσο η τυχειότητα των τιμών του πληθυσμού πολλές φορές οδηγούσε τον αλγόριθμο σε αστοχία, επηρεάζοντας με αυτόν τον τρόπο την ευρωστία του.

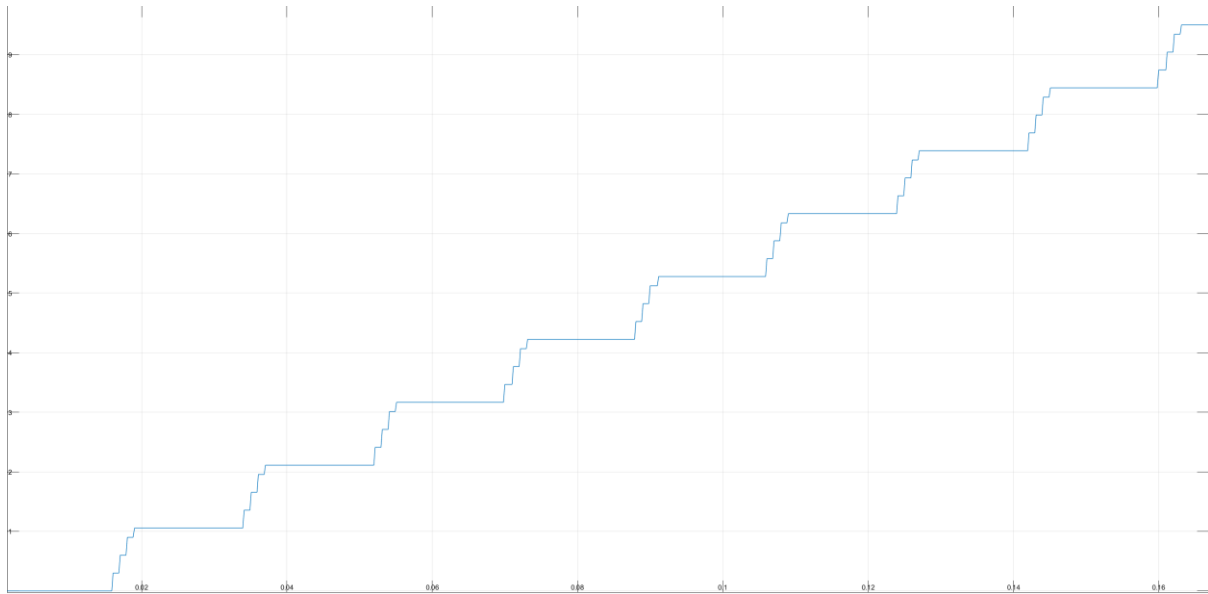
Το βήμα που χρησιμοποιείται σε όλη την εργασία για την εναλλαγή είναι 0.3 A. Ο χρόνος αναμονής του αλγορίθμου σε μια τιμή, έτσι ώστε να υλοποιηθεί με επιτυχία η δειγματοληψία της ισχύος που αντλείται από το φωτοβολταϊκό σύστημα, ορίστηκε στα 15 ms. Από την εικόνα 5.3, γίνεται εύκολα αντιληπτό ότι χρησιμοποιώντας πληθυσμό τριών ατόμων, η πληροφορία που τροφοδοτείται στον αλγόριθμο είναι περιορισμένη.



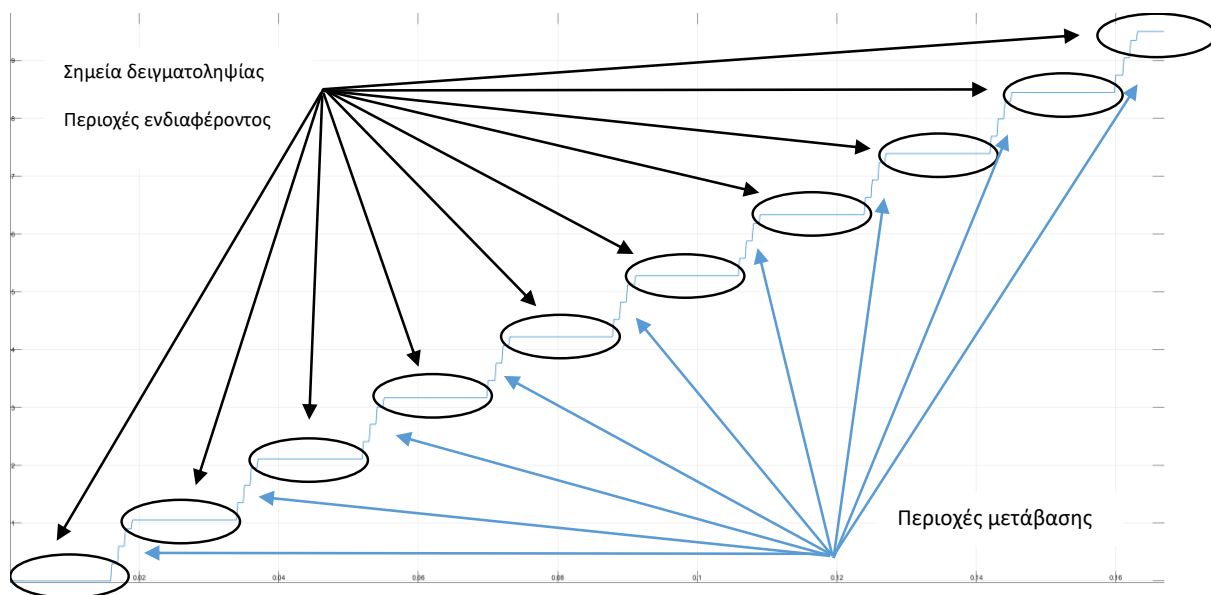
Εικόνα 5.3. Σημεία δειγματοληψίας και περιοχές μετάβασης

Οι περιοχές μετάβασης είναι απαραίτητες για την ομαλή λειτουργία του αλγορίθμου σε πραγματικές συνθήκες. Εντούτοις, δεν προσφέρουν κάποια χρήσιμη πληροφορία στο σύστημα και σε ένα ιδανικό

σενάριο πρέπει να είναι μηδενικές. Ο τρόπος με τον οποίο υλοποιείται η μείωση των περιοχών αυτών είναι η τοποθέτηση περισσότερων ατόμων στον πληθυσμό (εικόνα 5.4).



Εικόνα 5.4. Αρχικοποίηση πληθυσμού με χρήση 10 ατόμων στο διάστημα $[0, 9.5]$



Εικόνα 5.5. Σημεία δειγματοληψίας και περιοχές μετάβασης

Η εικόνα 5.4 απεικονίζει την αρχικοποίηση του πληθυσμού με δέκα άτομα. Είναι φανερό ότι οι περιοχές μετάβασης είναι χρονικά μικρότερες και η πληροφορία που τροφοδοτείται στον αλγόριθμο είναι πολλαπλάσια από την πληροφορία που τροφοδοτείται από τρία άτομα (εικόνα 5.5). Το μειονέκτημα της συγκεκριμένης τεχνικής είναι ότι αυξάνει το χρόνο αρχικοποίησης του αλγορίθμου (περίπου 10 ms). Αντιθέτως, το πλεονέκτημα της είναι ότι αυξάνεται η απόδοση του αλγορίθμου και σε συνδυασμό με τον όγκο της πληροφορίας που προσφέρεται στον αλγόριθμο από την αύξηση του πληθυσμού, το τελικό αποτέλεσμα είναι εξίσου ανταγωνιστικό με άλλα συστήματα μικρότερου πληθυσμού – όσον αφορά την ταχύτητα σύγκλισης.

5.4 Ανανέωση θέσης ατόμων πληθυσμού

Έχοντας ολοκληρώσει την αρχικοποίηση, το επόμενο τμήμα του αλγορίθμου είναι αυτό που διαχειρίζεται τον τρόπο με τον οποίο επανατοποθετούνται τα άτομα του πληθυσμού σε νέες θέσεις. Στα προηγούμενα κεφάλαια παρουσιάστηκε ο τρόπος με τον οποίο υλοποιείται η ανανέωση της θέσης των ατόμων του πληθυσμού. Για παράδειγμα, στον PSO και την NBDE σημαντικό ρόλο έχει η θέση των γειτονικών ατόμων. Οι θέσεις τους συνδυάζονται και μέσω κάποιων σχέσεων υπολογίζεται η νέα θέση του ατόμου σε κάθε γενιά. Ο αλγόριθμος AIA δεν χρησιμοποιεί πληροφορίες από τα γειτονικά άτομα για ανανέωση της θέσης ενός ατόμου στο χώρο των λύσεων. Αντιθέτως, χρησιμοποιεί τη θέση του ατόμου που βελτιστοποιεί τη συνάρτηση κόστους – μέχρι εκείνη τη γενιά – για την παραγωγή νέων ατόμων. Τα άτομα αυτά προστίθενται στον αρχικό πληθυσμό των δέκα ατόμων. Η θεώρηση που γίνεται στη συγκεκριμένη περίπτωση είναι ότι κάποιο από τα δέκα άτομα του αρχικού πληθυσμού – αυτό που δίνει τη βέλτιστη λύση – έχει τοποθετηθεί πολύ κοντά στη πραγματική λύση και χρησιμοποιώντας ως βάση τη θέση του συγκεκριμένου ατόμου, ο αλγόριθμος βρίσκεται μακριά από την πραγματική λύση μερικές γενιές μόνο.

Για την παραγωγή νέων ατόμων στον πληθυσμό, χρησιμοποιείται η παρακάτω εξίσωση:

$$position_{new} = position_{best} + dx \quad (5.1)$$

όπου $position_{best}$ η θέση του βέλτιστου ατόμου στον τρέχοντα πληθυσμό και dx συντελεστής μετακίνησης.

5.5 Συντελεστής μετακίνησης

Ο συντελεστής μετακίνησης είναι υπεύθυνος για τη θέση των νέων ατόμων γύρω από την τρέχουσα βέλτιστη θέση. Αρχικά, έγιναν αρκετοί πειραματισμοί χρησιμοποιώντας τυχαίες τιμές ($dx = rand$) σε ορισμένο διάστημα γύρω από την τρέχουσα βέλτιστη τιμή. Τα αποτελέσματα των πειραματισμών έδειξαν ότι είναι αρκετά δύσκολο να υπολογιστεί η διάσταση του ορισμένου διαστήματος γύρω από την τρέχουσα βέλτιστη τιμή και διέφερε ανάλογα με τα πειραματικά δεδομένα – απαίτηση για διαφορετικό διάστημα ανάλογα με τις συνθήκες μερικής σκίασης. Επίσης, υπήρχε αυξημένος χρόνος σύγκλισης του αλγορίθμου, διότι ο αλγόριθμος παγιδευόταν γύρω από ένα σημείο – πιθανόν λόγω της κανονικής κατανομής που χρησιμοποιείται για τους τυχαίους αριθμούς από τη Simulink – με αποτέλεσμα είτε να συμπληρώνεται ο μέγιστος αριθμός επαναλήψεων του αλγορίθμου, είτε να ολοκληρώνεται ο χρόνος εκτέλεσης της προσομοίωσης. Οι διαφορετικές περιβαλλοντικές συνθήκες που μπορεί να επικρατήσουν, επιφέρουν πολλά και διαφορετικά σενάρια μερικής σκίασης. Συνεπώς, δεν είναι δυνατόν να γίνονται διαφορετικές ρυθμίσεις στον αλγόριθμο για κάθε ένα από αυτά τα σενάρια. Ο αλγόριθμος θα πρέπει να αποδίδει μέγιστα για οποιοδήποτε σενάριο σκίασης και για το λόγο αυτό απορρίφθηκε η παρακάτω σχέση.

$$position_{new} = position_{best} + rand \quad (5.2)$$

5.6 Levy Random Walks

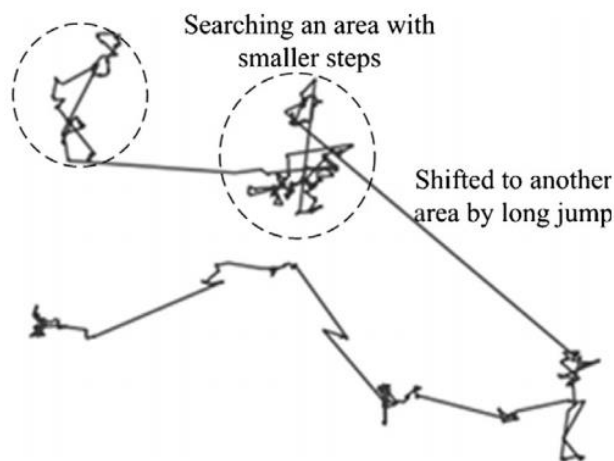
Για την ανανέωση του συντελεστή μετακίνησης dx χρησιμοποιήθηκαν οι πτήσεις Levy. Η ιδέα προήλθε από το άρθρο [17] στο οποίο παρουσιάζεται η υλοποίηση του αλγορίθμου CS (Cuckoo Search), όπου και χρησιμοποιείται για την εύρεση του σημείου MPP παρουσία συνθηκών μερικής σκίασης. Ο συγκεκριμένος αλγόριθμος βασίζεται στην παρασιτική συμπεριφορά που έχουν τα πτηνά της οικογένειας Κοκκυγίδων. Μια ιδιαιτερότητα που εμφανίζουν τα πτηνά αυτά, είναι η τοποθέτηση των αυγών τους σε φωλιές άλλων πτηνών, έτσι ώστε να μεγαλώσουν τα μικρά τους από τους άλλα πτηνά.

Επίσης, υπάρχουν περιπτώσεις που τα πτηνά αυτά καταστρέφουν τα αυγά των ξενιστών, έτσι ώστε να αυξήσουν τις πιθανότητες επιβίωσης των δικών τους απογόνων. Συνεπώς, η ιδέα της διαγραφής – απόρριψης λύσεων των ατόμων με τη χειρότερη απόδοση του πληθυσμού, προκύπτει από τη συγκεκριμένη συμπεριφορά των πτηνών. Οι συγγραφείς του άρθρου [17], υποστηρίζουν ότι ο αλγόριθμος CS είναι αποδοτικότερος από τον αλγόριθμο PSO και P&O, με πολύ καλή απόδοση και σε συνθήκες μερικής σκίασης. Η υλοποίηση του δεν ήταν δυνατή στη συγκεκριμένη εργασία, ωστόσο, ο αλγόριθμος CS μελετήθηκε και υιοθετήθηκαν κάποια από τα χαρακτηριστικά του. Το πρώτο, είναι το στοιχείο της απόρριψης της χειρότερης λύσης που χρησιμοποιήθηκε στους αλγορίθμους PSO_advanced και AIA, ενώ το δεύτερο, είναι ο τρόπος με τον οποίο υλοποιείται η τοποθέτηση των νέων ατόμων του πληθυσμού στο χώρο των λύσεων. Τα νέα σωματίδια του πληθυσμού τοποθετούνται στο χώρο των λύσεων χρησιμοποιώντας πτήσεις Levy.

Μαθηματικά, μια πτήση Levy, είναι μια τυχαία μετακίνηση με μεταβλητό βήμα. Τα βήματα από αυτήν την τυχαία κίνηση (random walk) εξάγονται από την κατανομή Levy σύμφωνα με την παρακάτω σχέση:

$$y = l^{-\lambda} \quad (5.3)$$

όπου l το μήκος της πτήσης και λ η απόκλιση μεταξύ των βημάτων. Στην εικόνα 57 απεικονίζεται ένα παράδειγμα από τα βήματα μιας πτήσης Levy σε δύο διαστάσεις. Για τη θέση των νέων ατόμων στο χώρο των λύσεων, χρησιμοποιείται η κατανομή Levy, όπου υλοποιούνται πολύ μικρά βήματα και περιστασιακά άλματα μεγαλύτερων αποστάσεων. Τα μεγάλα αυτά άλματα, είναι ικανά να αυξήσουν την αποτελεσματικότητα των αλγορίθμων που τα χρησιμοποιούν, ειδικά σε περιπτώσεις που υπάρχει μεγάλη πιθανότητα παγίδευσης του αλγορίθμου σε τοπικά μέγιστα ή ελάχιστα.



Εικόνα 5.6. Πτήση Levy σε δυοδιάστατο επίπεδο [17]

Με τη χρήση των πτήσεων Levy, ο αλγόριθμος οδηγείται γρήγορα στο ολικό μέγιστο και η πιθανότητα εγκλωβισμού του σε τοπικά μέγιστα ελαχιστοποιείται. Η τεχνική των πτήσεων υιοθετήθηκε έπειτα από αρκετούς πειραματισμούς. Για τον αλγόριθμο AIA, οι νέες θέσεις των ατόμων του πληθυσμού ανανεώνονται σύμφωνα με τους παρακάτω τύπους:

$$position_{new} = position_{best} + dx \quad (5.4)$$

$$dx = \left(\frac{u}{(|v|)^{\frac{1}{\beta}}} \right) (v_{best} - v_i) \quad (5.5)$$

όπου β μια τυχαία τιμή στο διάστημα $[0.1, 2]$. Ο συντελεστής β , ονομάζεται συντελεστής πολλαπλασιασμού πτήσης. Η συγκεκριμένη τιμή ορίζεται από το σχεδιαστή του συστήματος και επιλέχθηκε έπειτα από δοκιμές αρκετών τιμών (try and error). Ο συντελεστής v_{best} ορίζεται ως το άτομο με τη βέλτιστη λύση στον πληθυσμό, ενώ ο συντελεστής v_i , ορίζεται ως το άτομο του πληθυσμού από το οποίο πρόκειται να προκύψει ένα νέο άτομο. Οι συντελεστές u και v , ακολουθούν τις καμπύλες της κανονικής κατανομής και επιλέγονται ως εξής:

$$u \approx N(0, \sigma_u^2) \quad (5.6)$$

$$v \approx N(0, \sigma_v^2) \quad (5.7)$$

Τέλος, αν Γ οριστεί η συνάρτηση γάμμα (Gamma Function), τότε οι μεταβλητές σ_u και σ_v ορίζονται ως:

$$\sigma_u = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\beta \times \pi}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) \times \beta \times (2)^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (5.8)$$

5.7 Διαχείριση και απόρριψη λύσεων

Έχοντας περιγράψει τη διαδικασία με την οποία γίνεται η ανανέωση των βημάτων του πληθυσμού, στη συνέχεια θα γίνει περιγραφή της διαδικασίας, με την οποία γίνεται η διαχείριση και η απόρριψη των ατόμων του πληθυσμού.

Αρχικά, ο αλγόριθμος αρχικοποιείται με πληθυσμό δέκα ατόμων. Στη συνέχεια, εντοπίζεται η βέλτιστη λύση και χρησιμοποιείται ως βάση για τη δημιουργία δέκα νέων ατόμων που προστίθενται στον πληθυσμό. Με αυτόν τον τρόπο δημιουργείται ένας πληθυσμός από είκοσι συνολικά άτομα. Έπειτα, απορρίπτονται – διαγράφονται δέκα άτομα με χειρότερη τιμή συνάρτησης κόστους και η διαδικασία επαναλαμβάνεται. Ανανεώνεται, δηλαδή, ο πληθυσμός των δέκα ατόμων και εμπλουτίζεται με νέα άτομα σε κάθε γενιά.

5.8 Αλγόριθμος ΑΙΑ

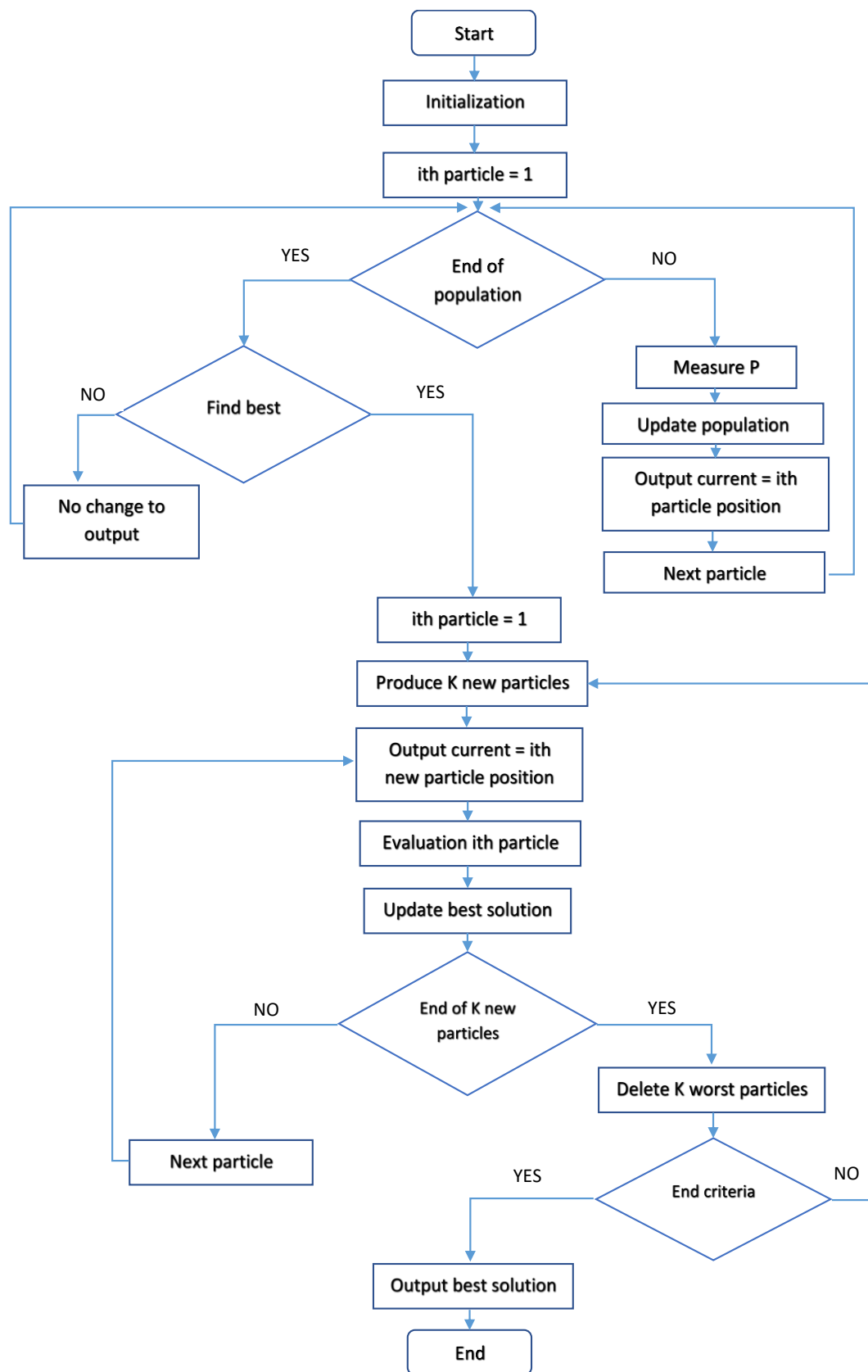
Έχοντας αναλύσει τις ιδιαιτερότητες του αλγορίθμου ΑΙΑ, παρακάτω παρουσιάζονται τα βήματα εκτέλεσης του.

Βήματα:

1. Αρχικοποίηση πληθυσμού δέκα ατόμων στο διάστημα των λύσεων. Στη συγκεκριμένη εργασία το διάστημα είναι $[0, 9.5]$, αλλά ο αλγόριθμος δύναται να υλοποιηθεί σε οποιοδήποτε διάστημα. Αν υπήρχε, για παράδειγμα, ένας μετατροπέας DC – DC στην προσομοίωση και ο αλγόριθμος επενέβαινε στο εύρος ζώνης του, τότε το διάστημα θα ήταν το $[0, 1]$.

2. Αναζήτηση βέλτιστης λύσης στον πληθυσμό. Το βήμα αυτό υλοποιείται επιβάλλοντας το ρεύμα του κάθε ατόμου – της κάθε λύσης – του πληθυσμού στην πηγή ρεύματος του κυκλώματος, μετρώντας στη συνέχεια τη μεταβολή στην ισχύ του συστήματος. Η βέλτιστη τιμή της ισχύος καταγράφεται για να χρησιμοποιηθεί στο παρακάτω βήμα.
3. Έχοντας εντοπίσει το άτομο του πληθυσμού που επιφέρει τη μεγαλύτερη ισχύ, γίνεται η χρήση των εξισώσεων 5.4 – 5.8, για την παραγωγή δέκα νέων ατόμων. Ο πληθυσμός στη συγκεκριμένη φάση αποτελείται από είκοσι άτομα.
4. Στη συνέχεια γίνεται η μέτρηση της ισχύος που επιφέρει το κάθε ένα από τα νέα άτομα του πληθυσμού και γίνεται έλεγχος αν υπάρχει νέα βέλτιστη λύση. Αν υπάρχει, γίνεται ανανέωση της τιμής της.
5. Ακολουθεί απόρριψη των δέκα χειρότερων ατόμων του πληθυσμού. Ο αριθμός των ατόμων του πληθυσμού επαναφέρεται στα δέκα άτομα.
6. Τέλος, γίνεται έλεγχος της θέσης των ατόμων. Αν τα άτομα όλου του πληθυσμού έχουν την ίδια θέση, τότε τερματίζει η λειτουργία του αλγορίθμου θεωρώντας ότι έχει συγκλίνει στη βέλτιστη λύση. Διαφορετικά, εκτελούνται τα βήματα 3 – 6.

5.9 Διάγραμμα ροής



Εικόνα 5.7. Διάγραμμα ροής αλγορίθμου ΑΙΑ

5.10 Ψευδό – κώδικας

Pseudo-code for AIA

```
initialize all required parameters

if (not termination criteria) and (not main_routine)

    broadcast duty (ith particle solution)
    adjust duty (smooth transition from previous duty value to next value)
    measure Power
    choose next particle
    if end of population
        find best particle
        main_routine = 1
    end

elseif main_routine

    create K new particles
    current particle = 1
    while current particle <=K
        broadcast duty (ith from K new particles)
        adjust duty (smooth transition from previous duty value to next value)
        measure Power
        current particle = current particle + 1
    end
    sort population
    delete K worst particles from population
    check termination criteria
    if termination criteria
        main_routine = 0
    else
        broadcast best solution
    end
```

Εικόνα 5.8. Ψευδό - κώδικας αλγορίθμου AIA

Κατά την υλοποίηση του κώδικα AIA δημιουργήθηκαν συναρτήσεις, με σκοπό να μειώσουν την πολυπλοκότητα και να αυξήσουν την ευρωστία του κώδικα. Στον παρακάτω πίνακα παρουσιάζονται οι συναρτήσεις και οι τιμές των μεταβλητών που χρησιμοποιήθηκαν.

5.11 Πίνακας τιμών που χρησιμοποιήθηκαν

Πίνακας 5.1. Πίνακας τιμών για μεταβλητές και επεξήγηση βοηθητικών συναρτήσεων αλγορίθμου ΑΙΑ

Μεταβλητή / Συνάρτηση	Σχόλια
K	Πληθυσμός
number_of_solution	Τρέχων άτομο πληθυσμού
duty_step	Μέγιστη εναλλαγή ρεύματος (αποτέλεσμα αλγορίθμου) ανά κύκλο εκτέλεσης αλγορίθμου
mean_time	Χρόνος αποκατάστασης ρεύματος – δειγματοληψία μετά το πέρας του συγκεκριμένου χρονικού διαστήματος
sort_archive	Ταξινόμηση πληθυσμού
sort_archive_2	Διαγραφή χειρότερων ατόμων από πληθυσμό
adjust_duty	Χρησιμοποιείται για ομαλή μετάβαση από μια τιμή εξόδου του αλγορίθμου σε άλλη (π.χ. από ρεύμα 0.5A $\rightarrow$ 8 A)
create_population	Χρησιμοποιείται για την αρχικοποίηση του πληθυσμού

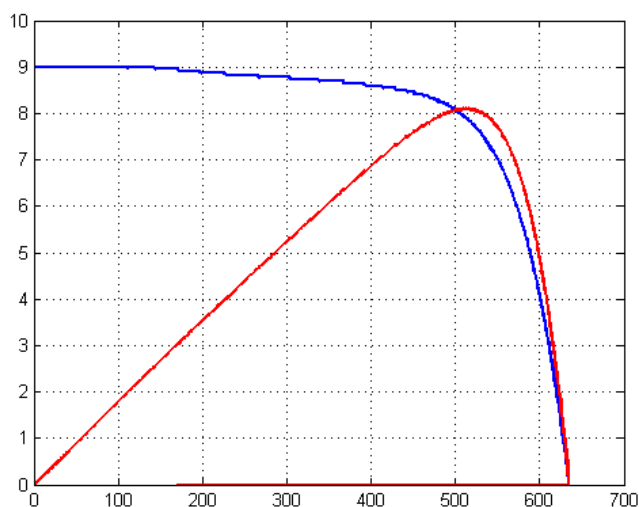
Κεφάλαιο 6ο: Πειραματική διαδικασία

6.1 Εισαγωγή

Το παρόν κεφάλαιο παρουσιάζει τη διαδικασία μέσα από την οποία προκύπτει η επικύρωση των αποτελεσμάτων για τους αλγόριθμους που υλοποιήθηκαν. Όπως αναφέρθηκε και στα προηγούμενα κεφάλαια, ένας από τους σκοπούς της παρούσας εργασίας είναι η μελέτη της απόδοσης των αλγορίθμων παρουσία συνθηκών μερικής σκίασης. Ως μερική σκίαση ορίζεται η μη ομοιόμορφη κατανομή της ακτινοβολίας (ηλιακής ή μη) πάνω σε ένα φωτοβολταϊκό πάνελ.

Για την επικύρωση των αποτελεσμάτων των αλγορίθμων δημιουργήθηκαν εννέα συνολικά πειράματα. Κάθε ένα πείραμα έχει διαφορετικές τιμές για τη θερμοκρασία, την ακτινοβολία και τον αριθμό των φωτοβολταϊκών πάνελ που συμμετέχουν στη συστοιχία. Στο σημείο αυτό να αναφερθεί ότι οι πειραματισμοί που υλοποιήθηκαν αφορούν μια φωτοβολταϊκή συστοιχία, η οποία αποτελείται από συνολικά πέντε φωτοβολταϊκές μονάδες. Κάθε μονάδα αποτελείται και από διαφορετικό αριθμό πάνελ και συνολικά σε όλο το σύστημα θεωρήθηκαν δεκαέξι πάνελ. Κάθε φωτοβολταϊκή μονάδα αποτελείται και από διαφορετικό αριθμό πάνελ – για κάθε πείραμα.

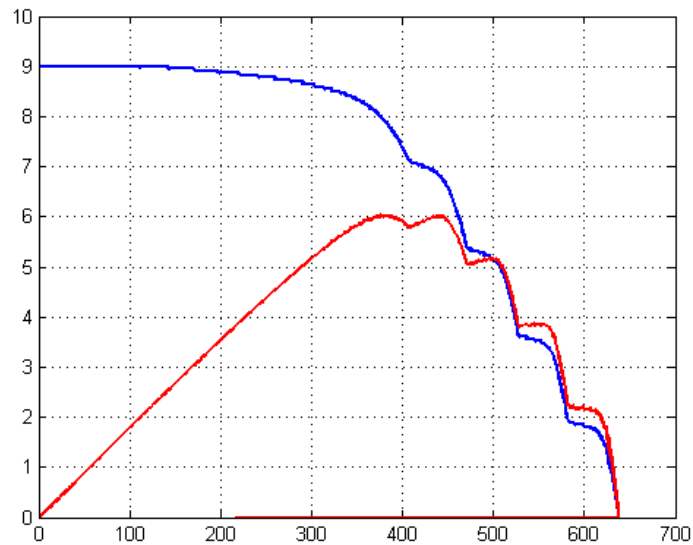
Ένα από τα πρώτα βήματα επικύρωσης της λειτουργίας των αλγορίθμων είναι η δημιουργία μιας σειράς πειραμάτων, που έχουν ως σκοπό να εντοπίσουν και να υποδείξουν τα αδύναμα σημεία τους, καθώς και τα σημεία που χρήζουν βελτίωσης. Είναι προφανές, ότι για την καμπύλη της εικόνας 6.1, υπάρχει ένα ολικό μέγιστο και βιβλιογραφικά αποδεικνύεται ότι υπάρχει μια πληθώρα αλγορίθμων που εντοπίζει εύκολα και με ακρίβεια το σημείο αυτό.



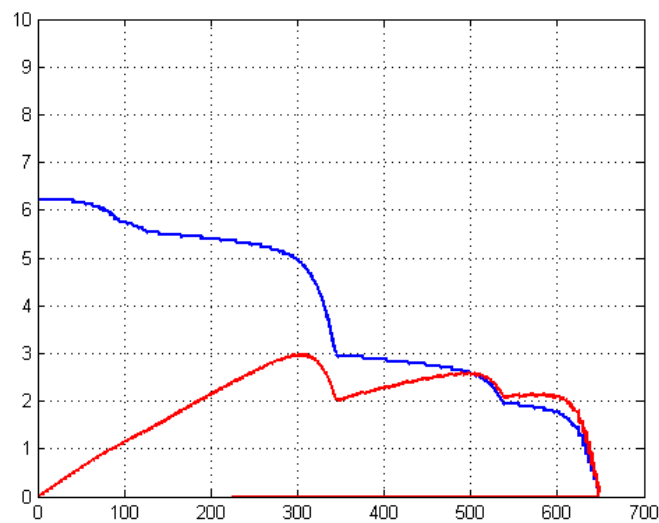
Εικόνα 6.1. Καμπύλη PV (κόκκινο χρώμα), καμπύλη IV (μπλε χρώμα)

Υπάρχουν όμως περιπτώσεις που δεν είναι ξεκάθαρος ο εντοπισμός του ολικού μεγίστου, εικόνα 6.2. Επίσης, υπάρχουν περιπτώσεις όπου το ολικό μέγιστο εντοπίζεται είτε στο αριστερό, είτε στο κεντρικό τμήμα της καμπύλης, εικόνα 6.3. Εντούτοις, δεν είναι δυνατό να υλοποιηθούν και να προσομοιωθούν πειραματικά όλες οι δυνατές παραλλαγές του συστήματος, έτσι ώστε να δοκιμαστούν οι αλγόριθμοι για όλα τα δυνατά σενάρια μερικής σκίασης. Για το λόγο αυτό, επιλέχθηκαν μόνο εννέα αντιπροσωπευτικά σενάρια, τα οποία θεωρήθηκαν ότι είναι ικανά να αναδείξουν τη βέλτιστη τεχνική – βέλτιστο αλγόριθμο – ο οποίος εν συνεχεία θα εφαρμοστεί σε ένα πραγματικό σύστημα, ώστε να συνεχιστούν οι πειραματισμοί. Συνεπώς, η μελέτη και η ανάλυση που έγινε στη συγκεκριμένη εργασία,

αφορά το διαχωρισμό και την επιλογή μιας μεθόδου, από ένα σύνολο, η οποία είναι δυνητικά ικανή να αποδειχθεί ως η πλέον κατάλληλη επιλογή για εφαρμογή σε φωτοβολταϊκά συστήματα.



Εικόνα 6.2. Καμπύλη PV (κόκκινο χρώμα), καμπύλη IV (μπλε χρώμα), μερική σκίαση



Εικόνα 6.3. Καμπύλη PV (κόκκινο χρώμα), καμπύλη IV (μπλε χρώμα), μερική σκίαση

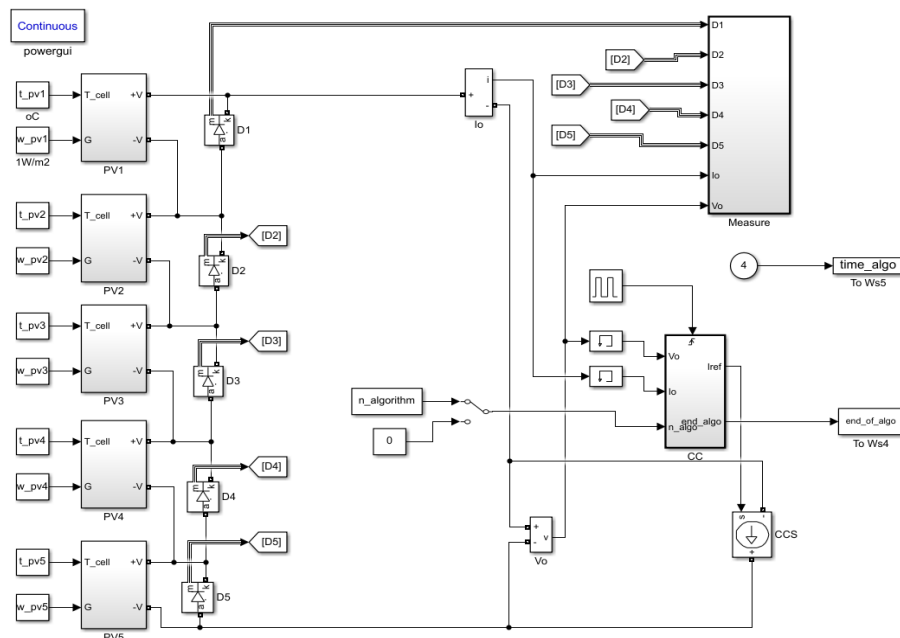
Στα επόμενα υπό-κεφάλαια ακολουθούν οι πειραματισμοί που υλοποιήθηκαν για να επαληθεύσουν την ορθή λειτουργία των αλγορίθμων. Η αξιολόγηση της απόδοσης τους περιορίζεται αρχικά σε εννέα διαφορετικά σενάρια μερικής σκίασης, με εξαίρεση το πρώτο σενάριο, το οποίο δεν παρουσιάζει σκίαση και αποτελεί το βασικό σενάριο στο οποίο αναμένεται υψηλό ποσοστό απόδοσης από όλους τους αλγορίθμους. Η διαδικασία με την οποία υλοποιήθηκαν οι πειραματισμοί είναι η εξής:

1. Εντοπισμός και καταγραφή των βέλτιστων τιμών για ισχύ, τάση και ρεύμα για κάθε σενάριο.
2. Εκτέλεση προσομοίωσης και εξαγωγή των απαραίτητων δεδομένων για αξιολόγηση. Τα παραγόμενα δεδομένα είναι:
 - a. Ισχύς, P_o
 - b. Τάση, V_o

- Με τον όρο *παραγόμενα δεδομένα*, προσδιορίζονται τα δεδομένα που χρησιμοποιήθηκαν από την προσομοίωση για τον υπολογισμό της απόδοσης του αλγορίθμου και τη δημιουργία γραφημάτων.

- $$\text{Απόδοση αλγορίθμου [ACC]} = \frac{\text{Μέγιστη ισχύς αλγορίθμου}}{\text{Πραγματική μέγιστη ισχύς}} \times 100 \quad (6.1)$$

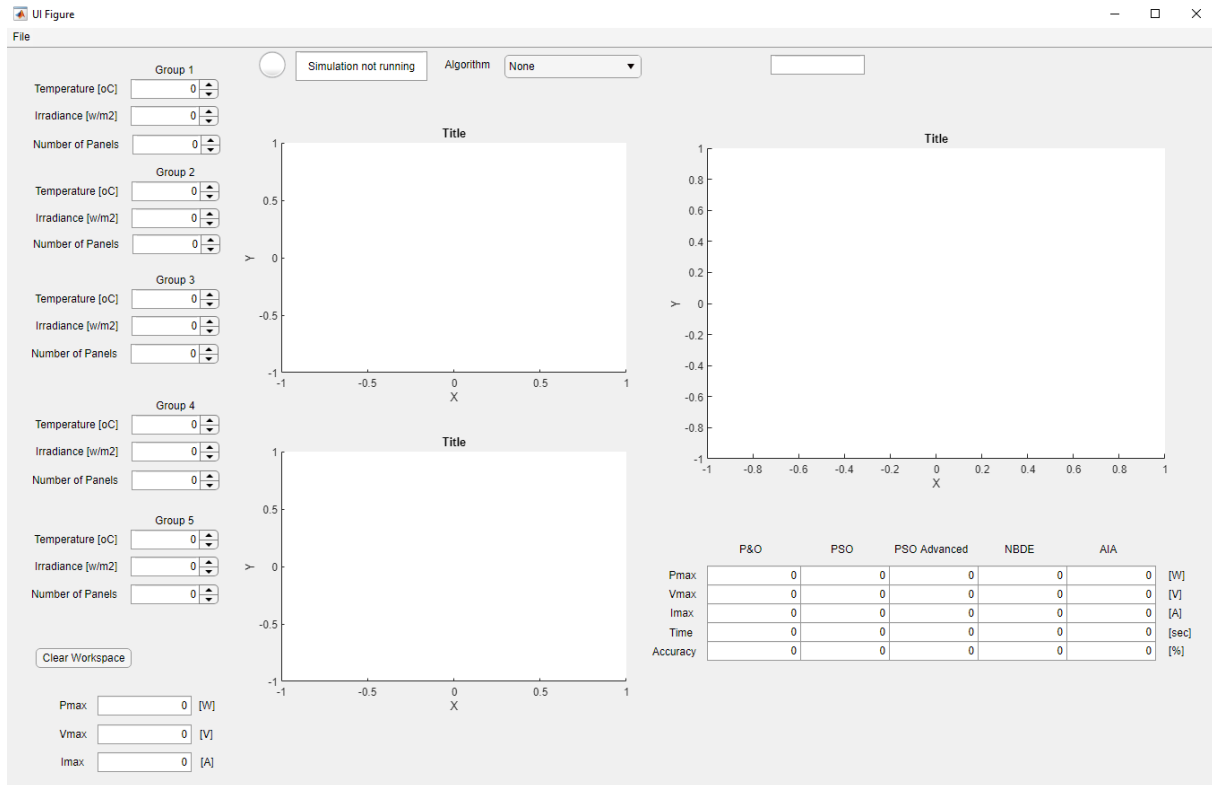
- Τα αποτελέσματα των προσομοιώσεων θα αναδείξουν τον αποδοτικότερο αλγόριθμο, ο οποίος μελλοντικά, θα εφαρμοστεί σε ένα πραγματικό σύστημα για να διακριβωθεί η ορθή λειτουργία του. Εφαρμόζοντας τον αλγόριθμο σε ένα πραγματικό σύστημα, είναι δυνατόν να προκύψουν αλλαγές – διορθώσεις, οι οποίες είναι ικανές να επηρεάσουν την απόδοση και το χρόνο εκτέλεσης του. Για το λόγο αυτό, η φυσική εφαρμογή του σε ένα πραγματικό σύστημα θεωρείται απαραίτητη για την ασφαλή εξαγωγή του τελικού συμπεράσματος.



65

6.2 Γραφικό περιβάλλον (GUI)

Για την ευκολότερη και πληρέστερη υλοποίηση των πειραματισμών, σχεδιάστηκε και υλοποιήθηκε ένα γραφικό περιβάλλον GUI, χρησιμοποιώντας τα ενσωματωμένα εργαλεία που παρέχει η Matlab. Σκοπός της συγκεκριμένης υλοποίησης είναι η οπτικοποίηση των αποτελεσμάτων, έτσι ώστε να είναι αρχικά ευκολότερη η διαδικασία απασφαλμάτωσης του κώδικα των αλγορίθμων και επίσης, η εξαγωγή των συμπερασμάτων. Το περιβάλλον που δημιουργήθηκε απεικονίζεται στην παρακάτω εικόνα:



Εικόνα 6.5. Γραφικός περιβάλλον πειραματισμών

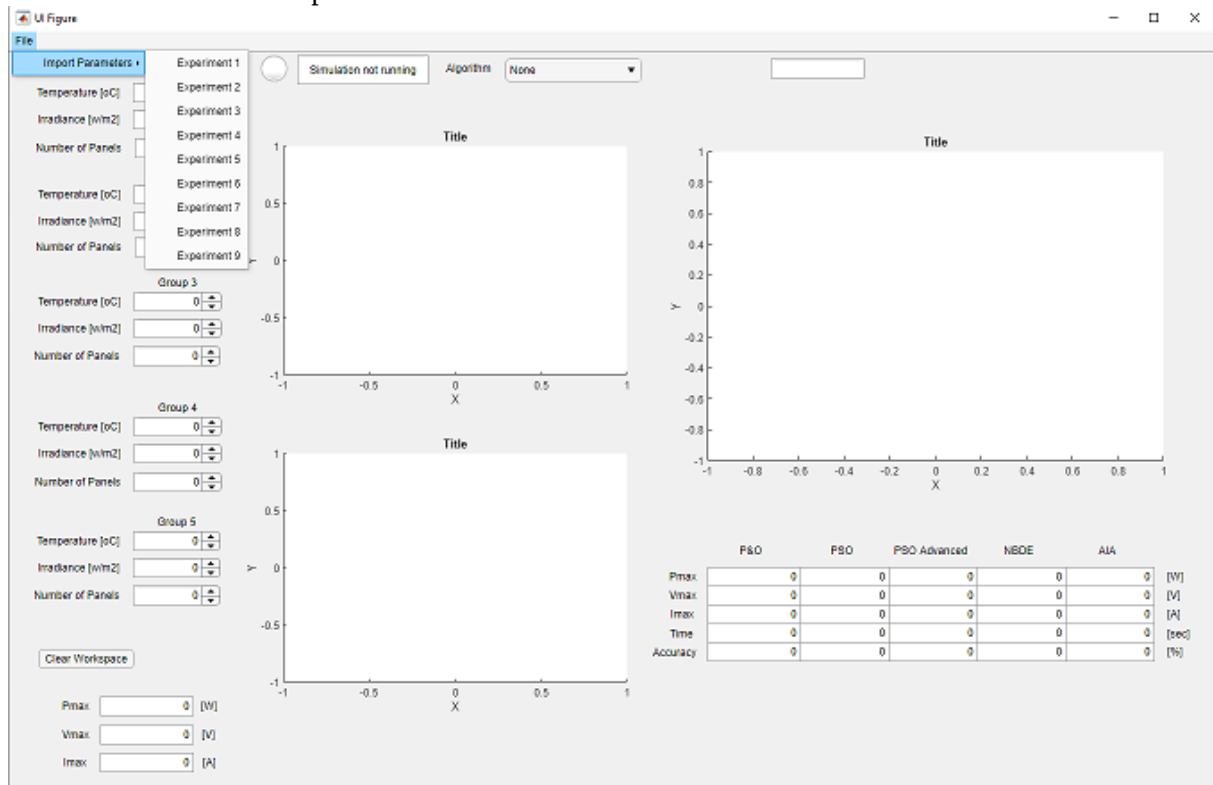
Ο σχεδιασμός είναι λιτός και οι πληροφορίες που παρέχονται στο χρήστη είναι αρκετές για να εξάγει κάποια αρχικά συμπεράσματα. Οι δυνατότητες που παρέχει το περιβάλλον αυτό είναι:

- Απεικόνιση χαρακτηριστικών PV και IV τρέχοντος πειράματος.
- Καμπύλη ισχύος εξόδου φωτοβολταϊκού συστήματος ανά αλγόριθμο.
- Συγκεντρωτικός πίνακας αποτελεσμάτων με πληροφορίες για την ισχύ εξόδου, τάση και ρεύμα εξόδου, χρόνο σύγκλισης και βαθμός απόδοσης των αλγορίθμων.
- Ένδειξη μέγιστης τιμής ισχύος, τάσης και ρεύματος για κάθε ένα πείραμα.
- Δυνατότητα φόρτωσης παραμέτρων για κάθε έναν από τους 9 πειραματισμούς που πραγματοποιείται η συγκεκριμένη εργασία.
- Δυνατότητα δημιουργίας νέων σεναρίων σκίασης και αξιολόγηση των αλγορίθμων στα νέα αυτά σενάρια.
- Δυνατότητα επιλογής συγκεκριμένου αλγορίθμου για εκτέλεση.
- Ένδειξη λειτουργίας προσομοίωσης.
- Ένδειξη επιλογής σεναρίου.
- Δυνατότητα διαγραφής δεδομένων, χρησιμοποιώντας το αντίστοιχο πλήκτρο.

Όπως αναφέρθηκε παραπάνω, το περιβάλλον αυτό δημιουργήθηκε καθαρά για βοηθητικούς σκοπούς. Τα δεδομένα παρέχονται στο περιβάλλον από τη Simulink και οπτικοποιούνται. Τα βήματα

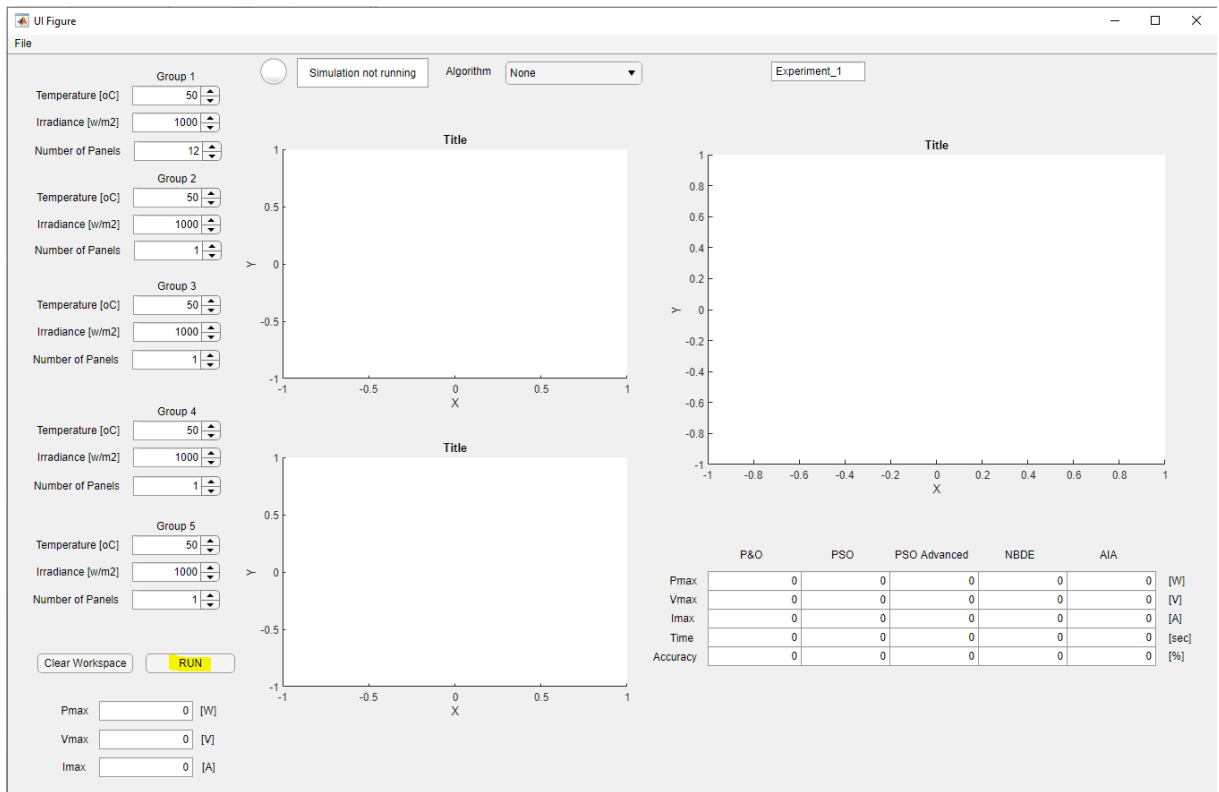
που απαιτούνται, έτσι ώστε να λειτουργήσει η οπτικοποίηση των αποτελεσμάτων, αναφέρονται στη συνέχεια. Το αρχείο της εφαρμογής πρέπει να βρίσκεται στον ίδιο φάκελο με το αρχείο της προσομοίωσης και να είναι ενεργή η Simulink. Χρησιμοποιήθηκε η έκδοση 2019a της Matlab.

1. Αρχικά ο χρήστης επιλέγει κάποιο από τα διαθέσιμα σενάρια από το πλήκτρο File → Import Parameters → Experiment 1 .. 9.

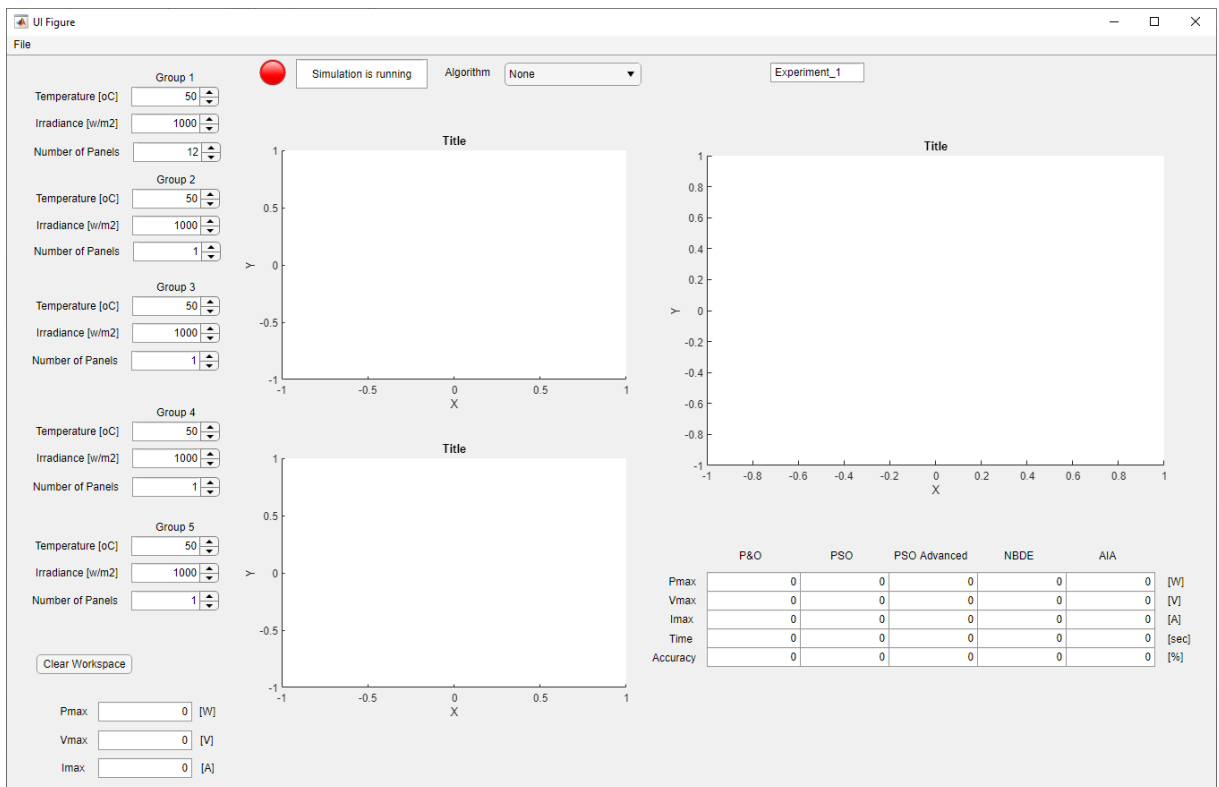


Εικόνα 6.6. Επιλογή δεδομένων για τα πειράματα 1 – 9

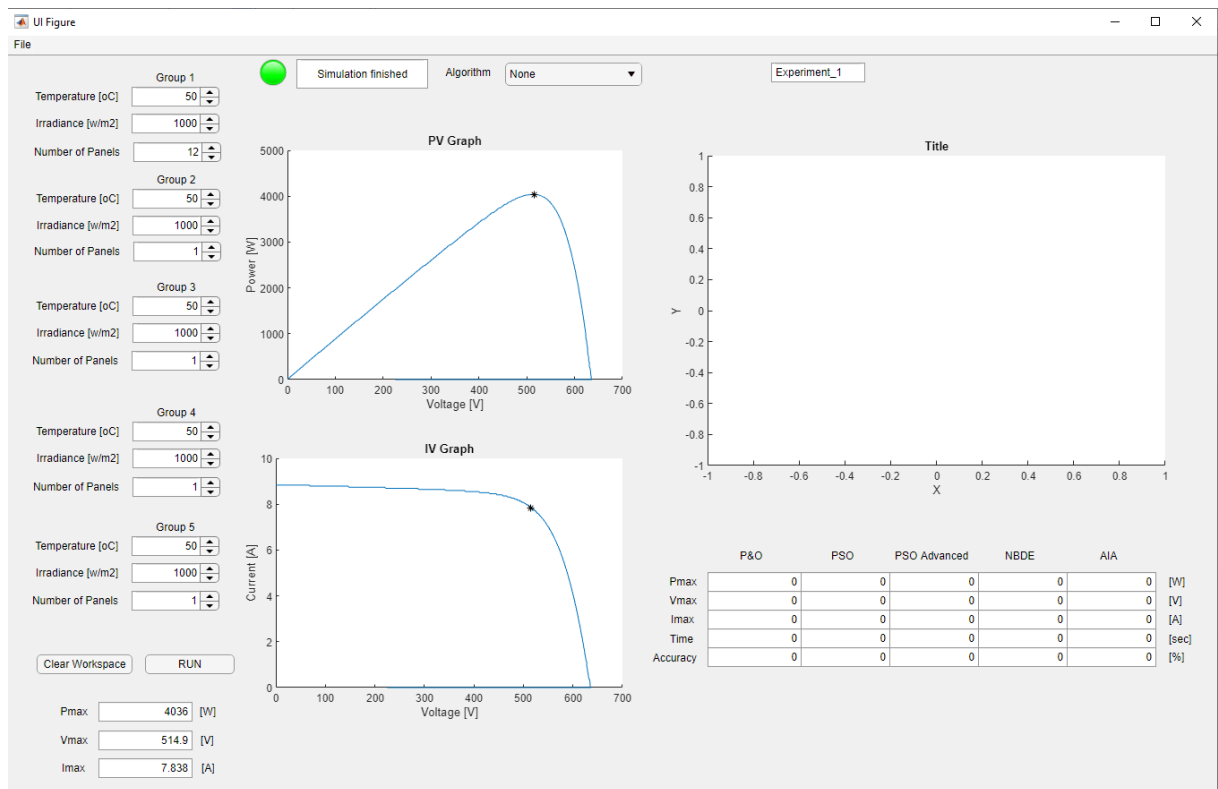
2. Αν ολοκληρωθεί η εισαγωγή δεδομένων με επιτυχία, τότε θα εμφανιστεί το πλήκτρο RUN δίπλα από το πλήκτρο Clear Workspace, το οποίο υποδεικνύει ότι η προσομοίωση είναι έτοιμη να εκτελεστεί. Στην περίπτωση που ο χρήστης δεν επιθυμεί την εκτέλεση κάποιου αλγορίθμου, μπορεί να εκτελέσει την προσομοίωση. Κατά τη διάρκεια εκτέλεσης της προσομοίωσης, η ενδεικτική λυχνία έχει χρώμα κόκκινο. Με το πέρας της εκτέλεσης, η ενδεικτική λυχνία αποκτά πράσινο χρώμα και εμφανίζονται οι γραφικές PV και IV του πειράματος. Επίσης, εμφανίζονται στο κάτω αριστερό τμήμα του παραθύρου της εφαρμογής, η μέγιστη ισχύς, η τάση και το ρεύμα του συστήματος. Αυτές οι τιμές είναι οι μέγιστες δυνατές για το σύστημα με τις υπάρχουσες περιβαλλοντικές συνθήκες.
3. Στη συνέχεια, ο χρήστης επιλέγει τον επιθυμητό αλγόριθμο και ενεργοποιεί την προσομοίωση με το πλήκτρο RUN. Μετά το πέρας της προσομοίωσης εμφανίζονται οι τιμές του αλγορίθμου στον πίνακα που βρίσκεται στο κάτω δεξί τμήμα του παραθύρου, καθώς επίσης και η καμπύλη της ισχύος κατά τη διάρκεια της προσομοίωσης. Τέλος, εμφανίζεται το όνομα του αλγορίθμου με το αντίστοιχο MPP σημείο στις καμπύλες PV και IV.
4. Η διαδικασία μπορεί να επαναλαμβάνεται μέχρι να εκτελεστούν όλοι οι διαθέσιμοι αλγόριθμοι. Στην περίπτωση που ο χρήστης επιλέξει διαφορετικό πείραμα, όλα τα δεδομένα διαγράφονται από το παράθυρο και ο χρήστης πρέπει να επαναλάβει τα παραπάνω βήματα.



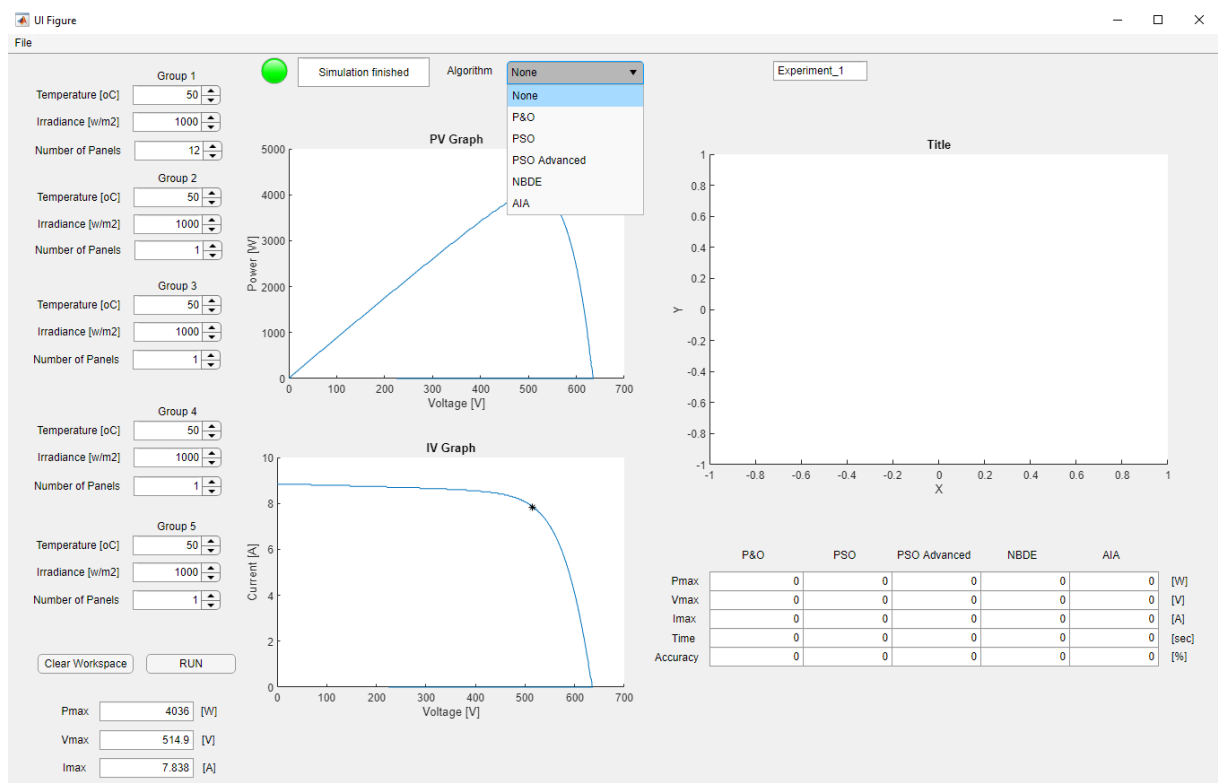
Εικόνα 6.7. Πλήκτρο εκκίνησης της προσομοίωσης



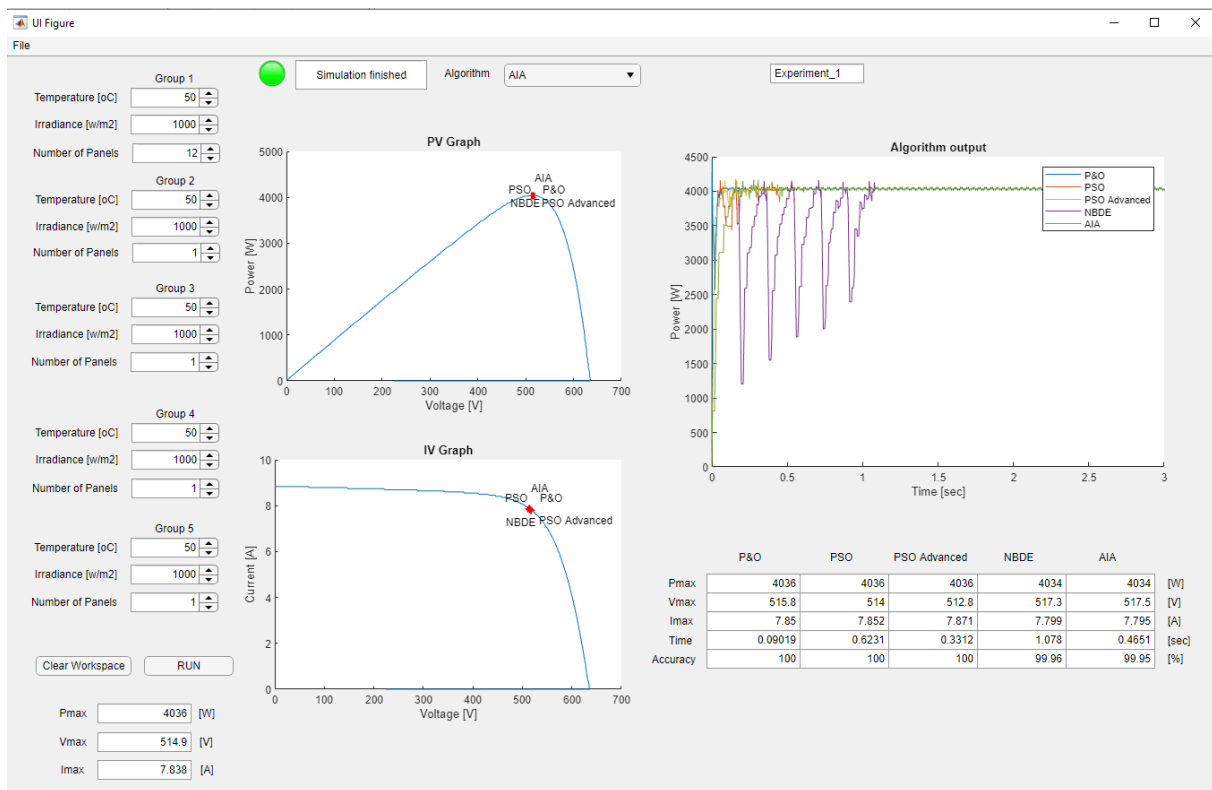
Εικόνα 6.8. Ένδειξη ότι η προσομοίωση βρίσκεται σε λειτουργία



Εικόνα 6.9. Εξαγωγή χαρακτηριστικών καμπυλών PV και IV



Εικόνα 6.10. Διαθέσιμες επιλογές αλγορίθμων



Εικόνα 6.11. Απεικόνιση πειράματος 1 στο γραφικό περιβάλλον

6.3 Πείραμα 1^ο

Το συγκεκριμένο σενάριο χρησιμοποιήθηκε για την διακρίβωση της ορθής λειτουργίας των αλγορίθμων, καθώς και για την απασφαλμάτωση τους. Δεν εμφανίζει σκιάσεις και αποτελεί την βάση για οποιονδήποτε περαιτέρω πειραματισμό. Είναι προφανές, ότι αν κάποιος από τους αλγορίθμους αστοχήσει στον εντοπισμό του μεγίστου σημείου MPP, είτε εμφανίζει κάποιο πρόβλημα στην υλοποίηση του και χρήζει διόρθωσης, είτε είναι ακατάλληλος για τη συγκεκριμένη μορφή προβλήματος.

Όπως αναφέρθηκε παραπάνω, το μοντέλο που χρησιμοποιήθηκε για την προσομοίωση περιέχει μια φωτοβολταϊκή συστοιχία, η οποία αποτελείται από συνολικά πέντε φωτοβολταϊκές μονάδες. Κάθε μονάδα αποτελείται και από διαφορετικό αριθμό πάνελ και συνολικά σε όλο το σύστημα θεωρήθηκαν δεκαέξι πάνελ. Κάθε φωτοβολταϊκή μονάδα αποτελείται και από διαφορετικό αριθμό πάνελ – για κάθε πείραμα. Στο συγκεκριμένο πείραμα ο αριθμός πάνελ, καθώς και οι τιμές της ακτινοβολίας και της θερμοκρασίας που χρησιμοποιήθηκαν απεικονίζονται στον παρακάτω πίνακα.

Πίνακας 6.1. Πίνακας τιμών για πειραματικό σενάριο 1

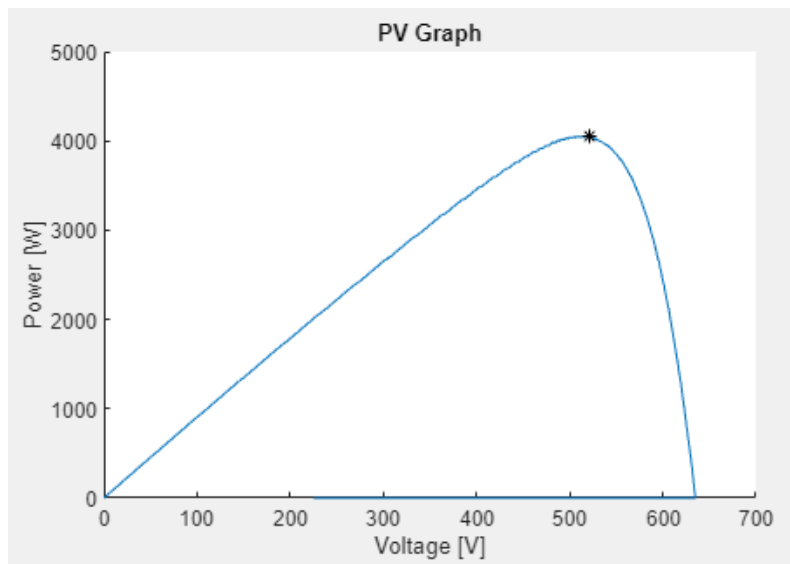
Μεταβλητή	Τιμή	Περιγραφή
Nss_1	12	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	1	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)

Nss_3	1	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	1	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	1	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	50° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	50° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	50° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	50° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	50° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	1000 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	1000 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	1000 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	1000 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	1000 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

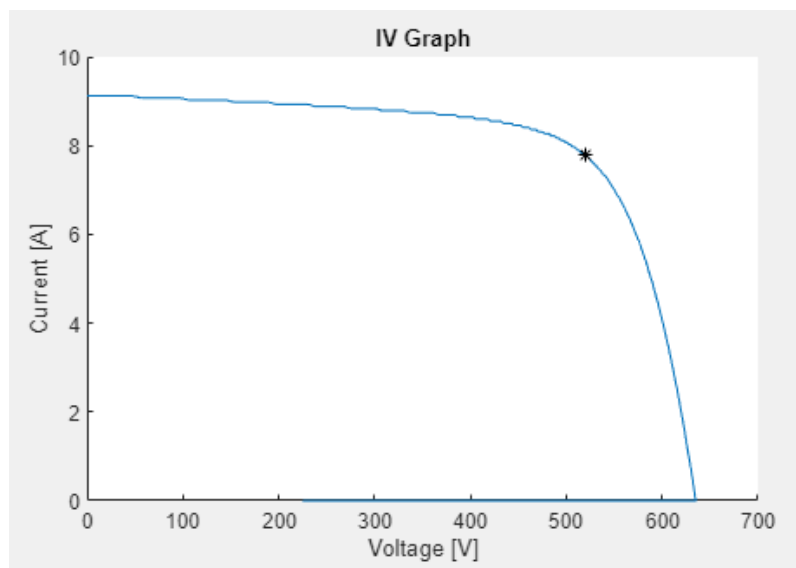
Αρχικά, εξάγονται οι καμπύλες ισχύος, τάσης και ρεύματος και καταγράφονται οι τιμές τους. Για το πείραμα 1 προέκυψαν οι παρακάτω τιμές.

Πίνακας 6.2. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 1

Μεταβλητή	Τιμή	Περιγραφή
Po	4036.27 W	Μέγιστη ισχύς 1 ^{ου} πειράματος
Vo	514.81 V	Μέγιστη τάση 1 ^{ου} πειράματος
Io	7.89 A	Μέγιστο ρεύμα 1 ^{ου} πειράματος



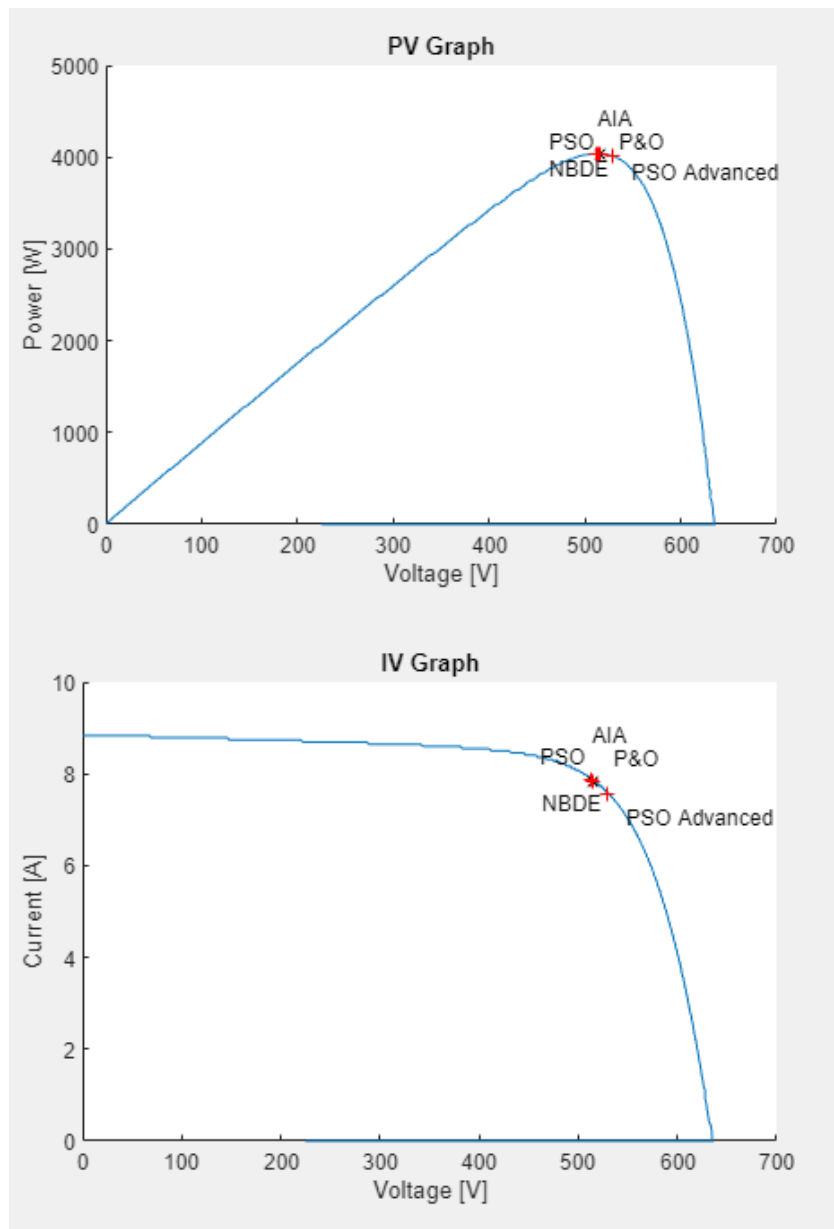
Εικόνα 6.12. Γράφημα PV, πείραμα 1



Εικόνα 6.13. Γράφημα IV, πείραμα 1

Λαμβάνοντας υπόψη τα παραπάνω δεδομένα, για το πείραμα 1, οι αλγόριθμοι αναμένεται να έχουν έξοδο στα 7.89 A, έτσι ώστε η παραγόμενη ισχύς να έχει τιμή 4036 W.

Εκτελώντας την προσομοίωση για κάθε έναν αλγόριθμο, προέκυψαν τα παρακάτω δεδομένα. Αρχικά, απεικονίζονται τα διαγράμματα PV και IV με τα αντίστοιχα σημεία MPP για κάθε αλγόριθμο. Στη συνέχεια, παρουσιάζεται ένας συγκεντρωτικός πίνακας αποτελεσμάτων.



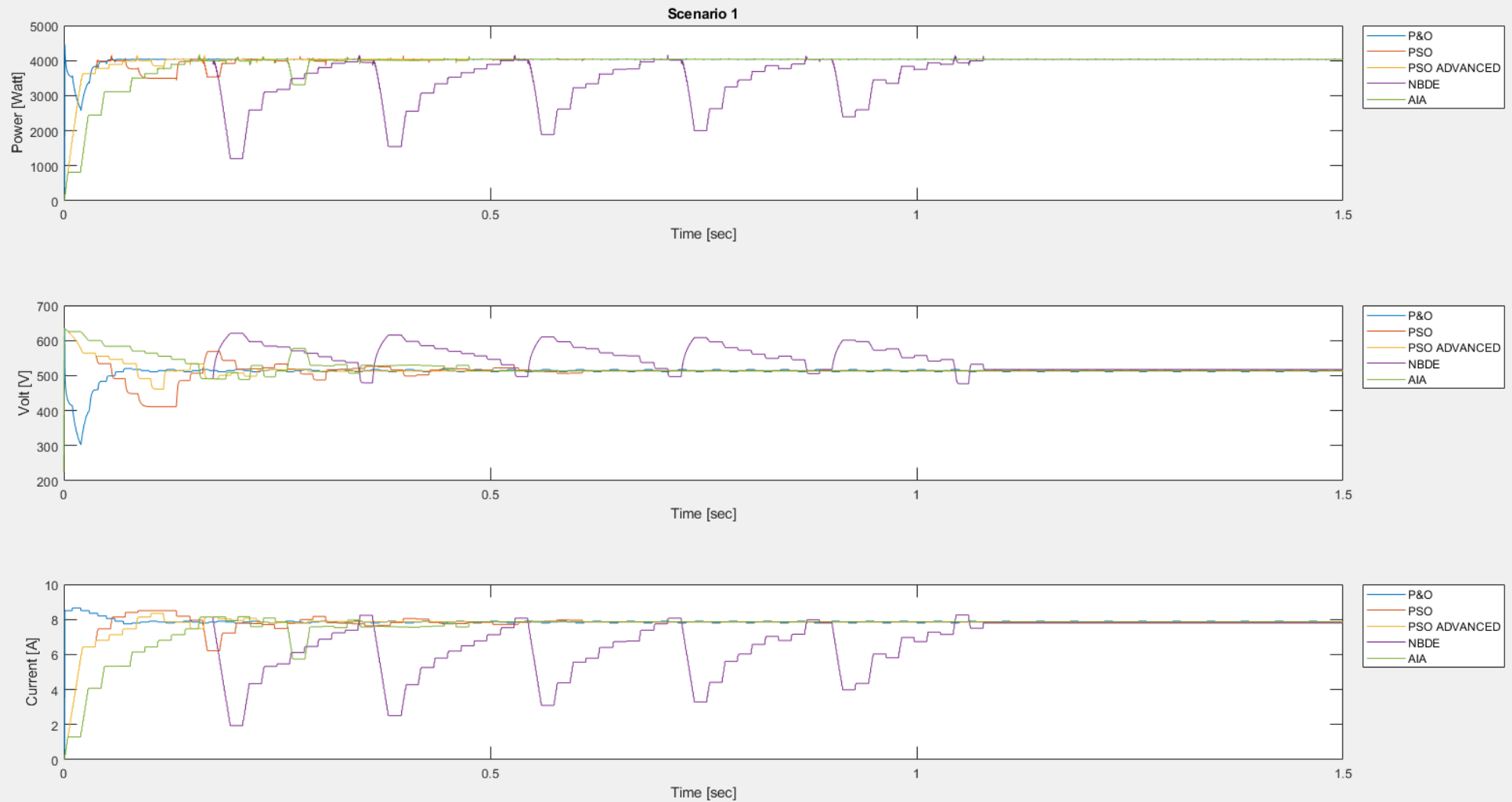
Εικόνα 6.14. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 1

Πίνακας 6.3. Αποτελέσματα προσομοίωσης, πείραμα 1

Αλγόριθμος	P_{max} [W]	V_{max} [V]	I_{max} [A]	Acc [%]	Χρόνος [sec]
-	4036.27	514.81	7.89	100.00	0.00
P&O	4036.12	515.84	7.85	100.00	0.09
PSO	4036.04	513.99	7.85	99.99	0.62
PSO Advanced	4036.13	512.77	7.87	100.00	0.33
NBDE	4034.21	517.27	7.80	99.95	1.08
AIA	4036.13	513.11	7.87	100.00	0.48

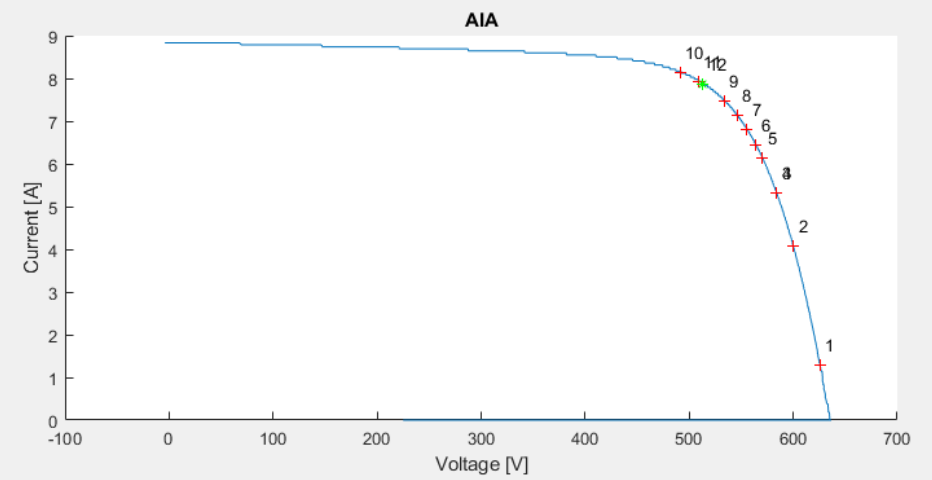
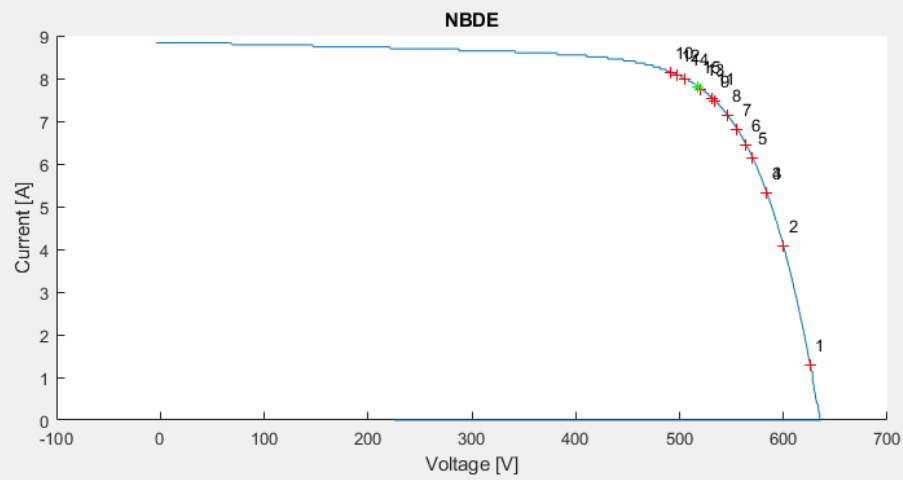
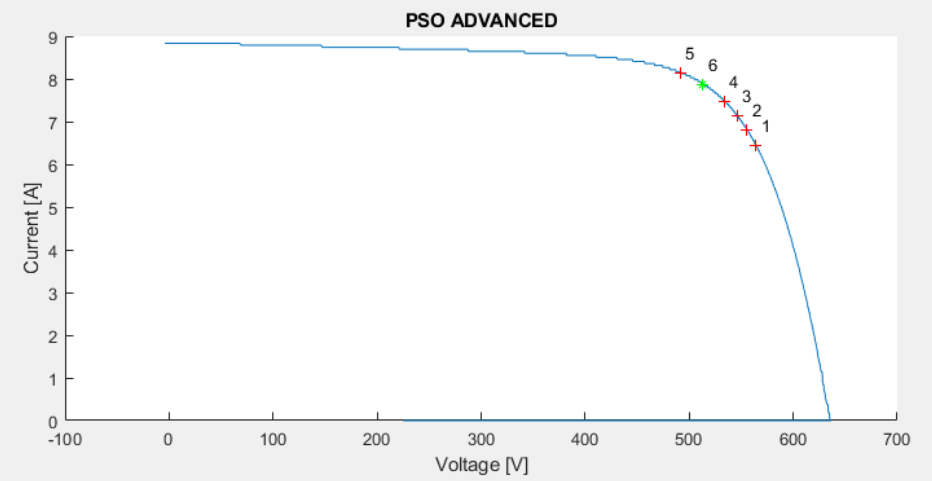
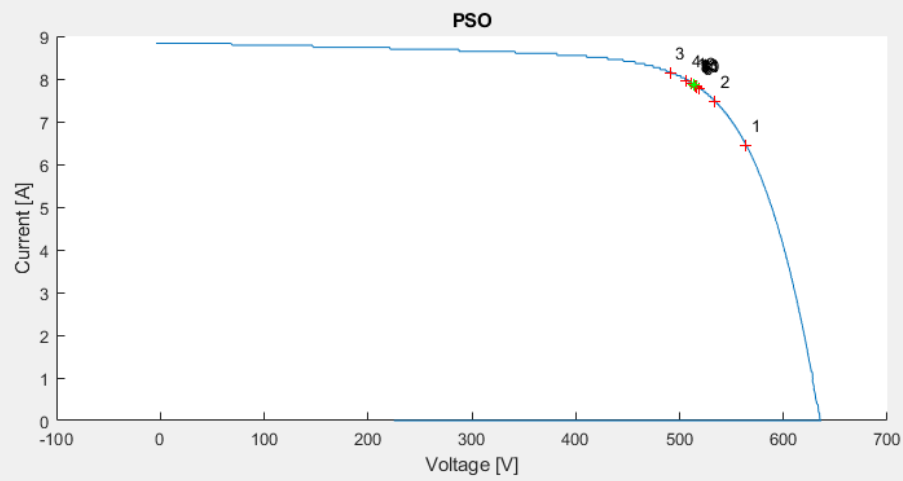
Από τα παραπάνω αποτελέσματα προκύπτει το συμπέρασμα ότι όλοι οι αλγόριθμοι, όπως ήταν αναμενόμενο, εντοπίζουν με επιτυχία το πραγματικό σημείο MPP με μεγάλη ακρίβεια. Συνεπώς, ως προς τη λειτουργία των αλγορίθμων, προκύπτει το συμπέρασμα ότι τουλάχιστον σε συνθήκες έλλειψης σκίασης, οποιοσδήποτε από τους παραπάνω αλγορίθμους είναι ικανός να εντοπίσει το επιθυμητό σημείο. Επίσης, από τα αποτελέσματα προκύπτει ότι ο αλγόριθμος P&O συγκλίνει πιο γρήγορα χρονικά από τους υπολοίπους, καθιστώντας τους μη απαραίτητους σε αυτές τις συνθήκες. Για αυτό το λόγο και ο συγκεκριμένος αλγόριθμος (P&O) θεωρείται ο πιο δημοφιλής, διότι υλοποιείται πολύ εύκολα και συγκλίνει με μεγάλη ταχύτητα στο σημείο MPP. Εντούτοις, παρουσιάζει μειονεκτήματα τα οποία δημιούργησαν την ανάγκη για αναζήτηση νέων μεθόδων.

Τα επόμενα πειράματα εξετάζουν την απόδοση των αλγορίθμων παρουσία συνθηκών μερικής σκίασης και αναμένεται να υπάρξει εμφανής διαφορά στις αποδόσεις τους.



Εικόνα 6.15. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 1

Algorithm steps



Εικόνα 6.16. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 1

6.4 Πείραμα 2^ο

Το δεύτερο πείραμα εμφανίζει ένα επιπλέον τοπικό μέγιστο. Αποτελεί την πρώτη περίπτωση μερικής σκίασης και αναμένεται ότι ο μοναδικός αλγόριθμος που δύναται να αστοχήσει είναι ο P&O – εξαρτάται φυσικά από την τιμή του ρεύματος εκκίνησης του. Στο συγκεκριμένο πείραμα οι μέγιστες τιμές για την ισχύ, την τάση και το ρεύμα απεικονίζονται στον πίνακα 6.5. Αρχικά, απεικονίζονται οι παράμετροι που αφορούν την προσομοίωση καθώς και οι καμπύλες PV και IV.

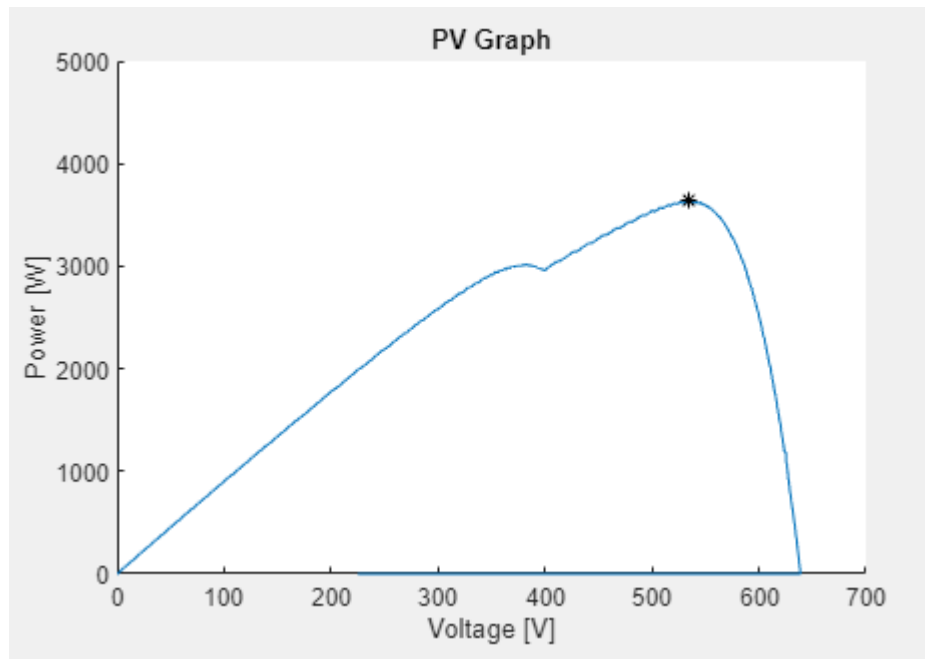
Πίνακας 6.4. Πίνακας τιμών για πειραματικό σενάριο 2

Μεταβλητή	Τιμή	Περιγραφή
Nss_1	12	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	1	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
Nss_3	1	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	1	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	1	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	50° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	40° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	40° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	40° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	40° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	1000 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	800 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	800 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	800 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	800 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

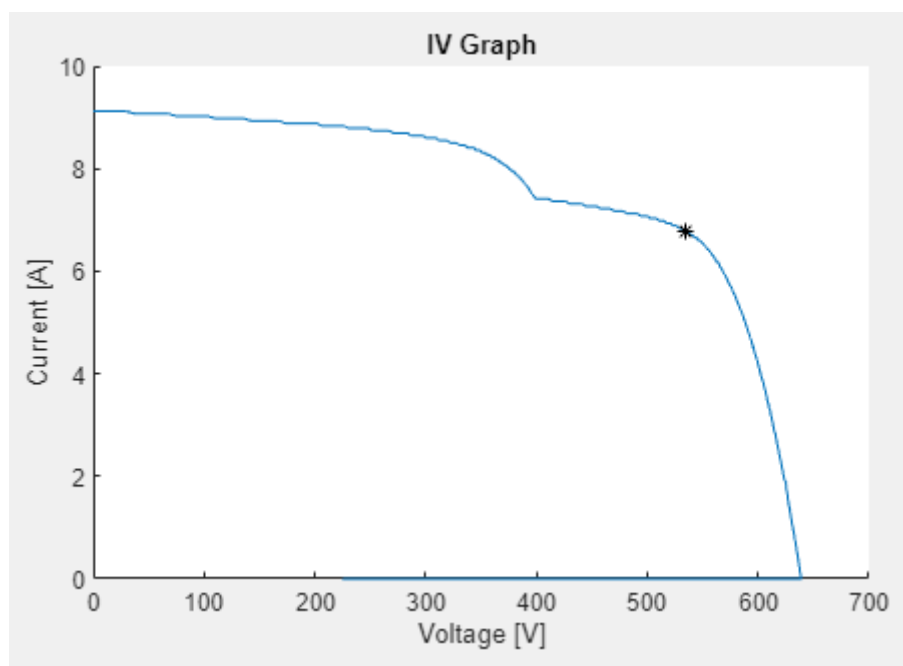
Συγκρίνοντας τα δεδομένα του παραπάνω πίνακα με τα αντίστοιχα του 1^{ου} πειράματος, παρατηρείται μια πτώση τόσο στην ακτινοβολία, όσο και στη θερμοκρασία των φωτοβολταϊκών μονάδων 2-5. Η πτώση της τιμής της ακτινοβολίας θα επιφέρει και αλλαγή στην καμπύλη της PV και IV, εμφανίζοντας ένα επιπλέον τοπικό μέγιστο, εικόνες 6.9, 6.10.

Πίνακας 6.5. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 2

Μεταβλητή	Τιμή	Περιγραφή
Po	3605.17 W	Μέγιστη ισχύς 2 ^{ου} πειράματος
Vo	540.8 V	Μέγιστη τάση 2 ^{ου} πειράματος
Io	6.69 A	Μέγιστο ρεύμα 2 ^{ου} πειράματος

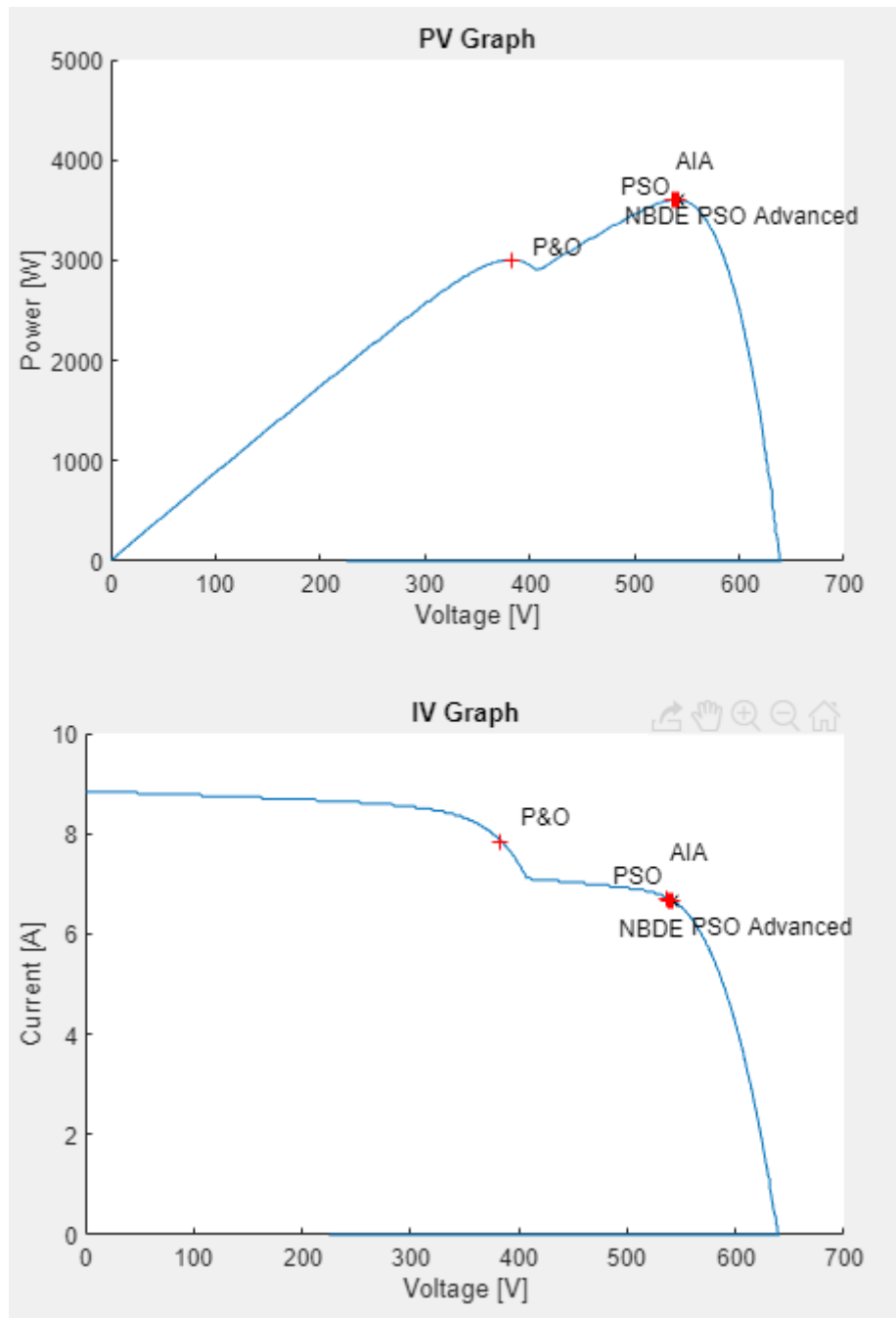


Εικόνα 6.17. Γράφημα PV, πείραμα 2



Εικόνα 6.18. Γράφημα PV, πείραμα 2

Λαμβάνοντας υπόψη τα παραπάνω δεδομένα, για το πείραμα 2, οι αλγόριθμοι που εντοπίζουν με επιτυχία το σημείο MPP, αναμένεται να έχουν έξοδο περίπου στα 6.8 A, έτσι ώστε η παραγόμενη ισχύς να έχει τιμή 3634 W. Εκτελώντας την προσομοίωση για κάθε έναν αλγόριθμο, προέκυψαν τα παρακάτω δεδομένα. Αρχικά, απεικονίζονται τα διαγράμματα PV και IV, με τα αντίστοιχα σημεία MPP για κάθε αλγόριθμο. Στη συνέχεια, παρουσιάζεται ένας συγκεντρωτικός πίνακας αποτελεσμάτων.

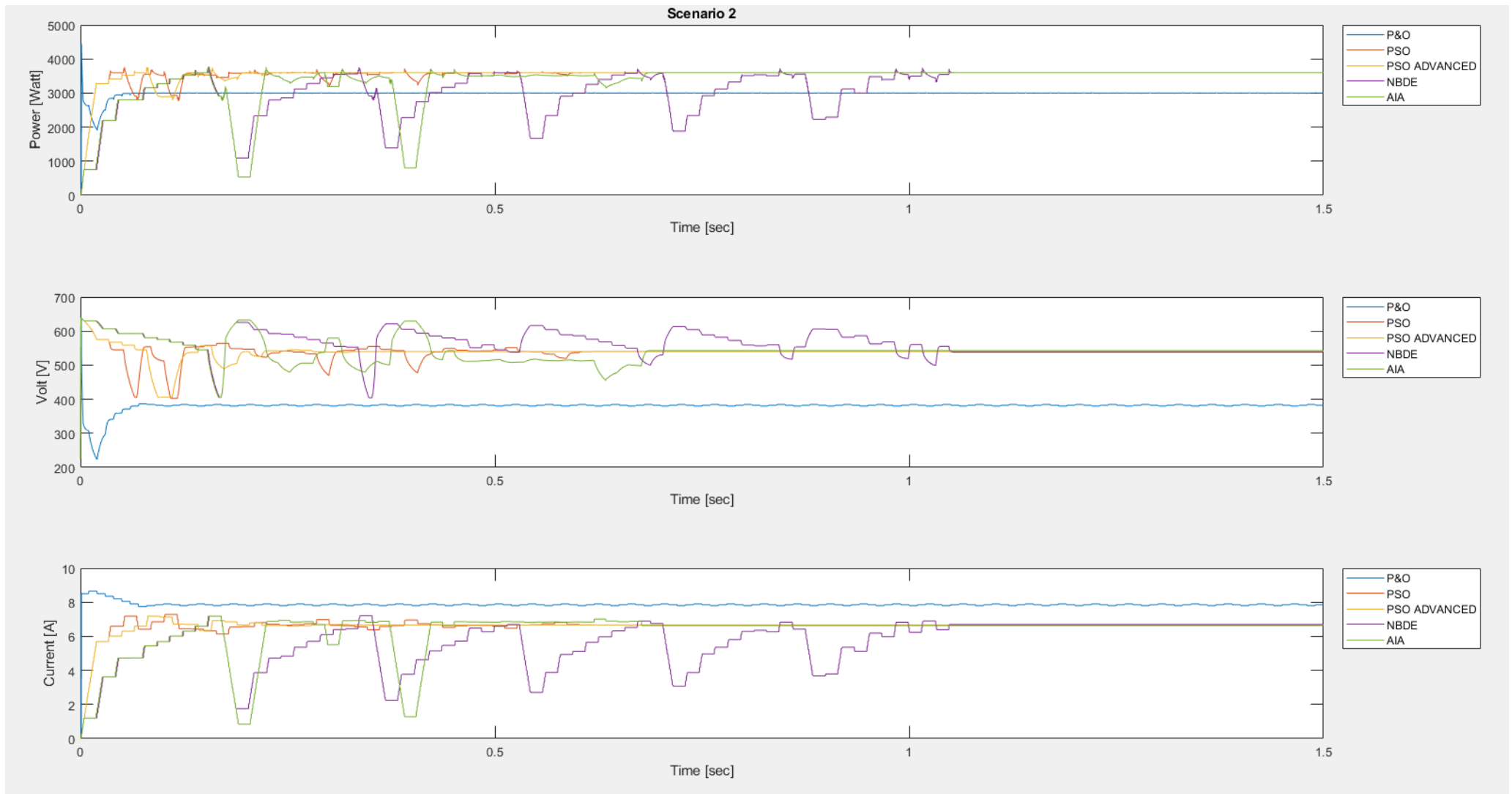


Εικόνα 6.19. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 2

Πίνακας 6.6. Αποτελέσματα προσομοίωσης, πείραμα 2

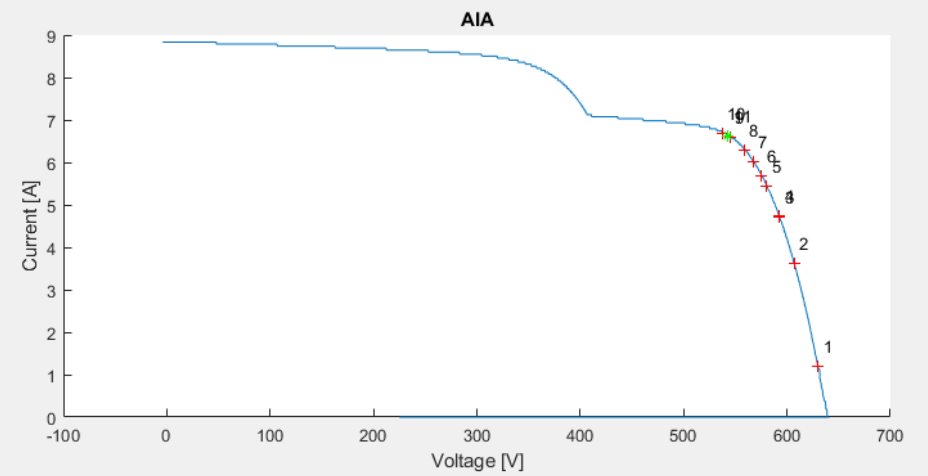
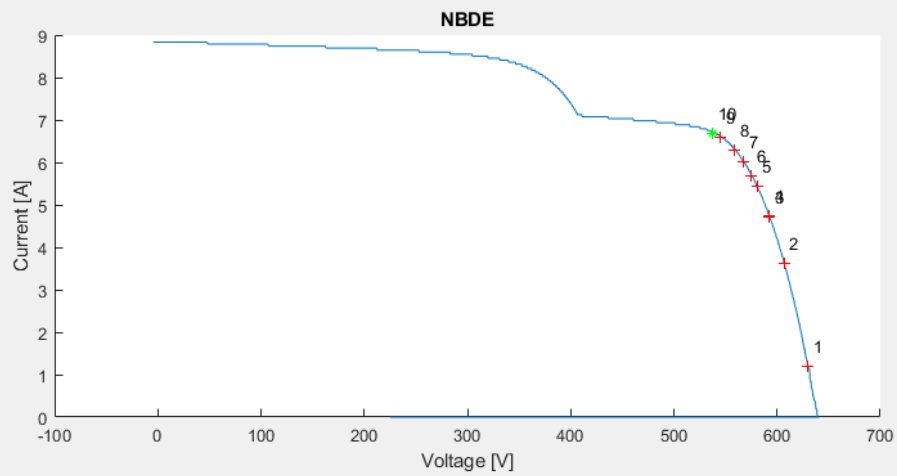
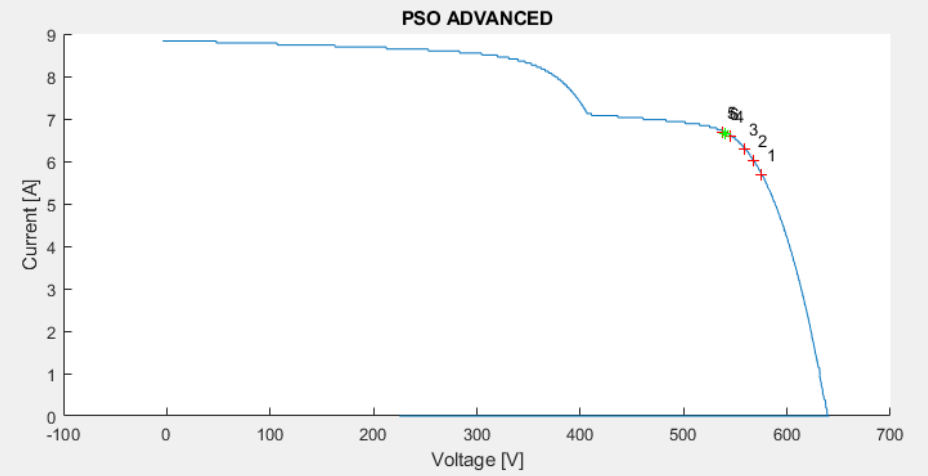
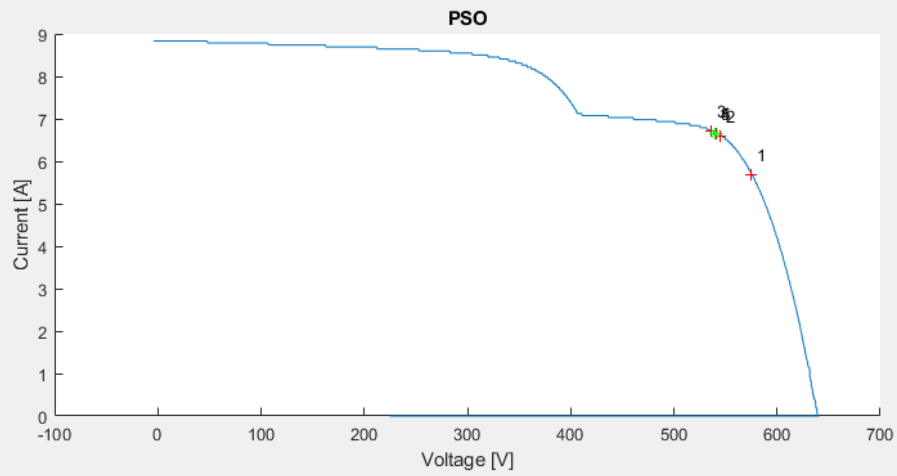
Αλγόριθμος	P_{max} [W]	V_{max} [V]	I_{max} [A]	Acc [%]	Χρόνος [sec]
-	3605.177	540.805	6.698	100.000	0.000
P&O	3001.711	383.576	7.850	83.261	0.090
PSO	3599.022	540.245	6.662	99.829	0.617
PSO Advanced	3599.039	539.916	6.666	99.830	0.324
NBDE	3598.311	537.954	6.689	99.810	1.048
AIA	3597.554	542.699	6.629	99.789	0.677

Μελετώντας τα παραπάνω αποτελέσματα, παρατηρείται ότι ο αλγόριθμος P&O αστοχεί στον εντοπισμό του ολικού μεγίστου. Δεδομένης της αρχικής τιμής του αλγορίθμου, ήταν αναμενόμενο το αποτέλεσμα αυτό. Στη συνέχεια, παρατηρείται ότι οι υπόλοιποι αλγόριθμοι έχουν εντοπίσει με επιτυχία το σημείο MPP. Συγκρίνοντας τα αποτελέσματα με βάση την απόδοση, παρατηρείται ότι όλοι οι αλγόριθμοι εμφανίζουν παρόμοια υψηλή απόδοση. Συγκρίνοντας τα αποτελέσματα με βάση το χρόνο εντοπισμού, παρατηρείται ότι ο P&O συγκλίνει ταχύτατα σε ένα τοπικό μέγιστο. Σε σχέση με τους υπόλοιπους αλγορίθμους, ο PSO εμφανίζει παρόμοιο χρόνο σύγκλισης με τον AIA, ο NBDE εμφανίζει το μεγαλύτερο χρόνο σύγκλισης και ο PSO Advanced παρουσιάζει το συντομότερο χρόνο σύγκλισης στο σημείο MPP. Ο διαφορετικός τρόπος αρχικοποίησης των σωματιδίων, που χρησιμοποιείται από τον PSO Advanced, είναι ο λόγος που του επιτρέπει την ταχύτερη σύγκλισή του από τους υπολοίπους. Επίσης, η εσωτερική δομή του αλγορίθμου NBDE δεν του επιτρέπει να εξάγει αποτέλεσμα νωρίτερα από τον προκαθορισμένο χρόνο εκτέλεσης του συνόλου των ορισμένων επαναλήψεων (100). Συνεπώς, ο χρόνος 1 sec αναμένεται να είναι ίδιος για όλα τα πειράματα. Ο PSO και ο AIA φαίνεται να έχουν παρόμοια συμπεριφορά. Τα επόμενα πειράματα αναμένεται να επιβεβαιώσουν ή όχι την υπόθεση αυτή.



Εικόνα 6.20. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 2

Algorithm steps



Εικόνα 6.21. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 2

6.5 Πείραμα 3^ο

Στο πείραμα 3, τα δεδομένα της ακτινοβολίας και της θερμοκρασίας των πάνελ αλλάζουν με τέτοιο τρόπο, έτσι ώστε να προκύψουν πέντε τοπικά μέγιστα στην καμπύλη PV. Ουσιαστικά κάθε μια φωτοβολταϊκή μονάδα έχει διαφορετική τιμή ακτινοβολίας και θερμοκρασίας από τις άλλες. Η διαφορά αυτή στις πέντε φωτοβολταϊκές μονάδες του συστήματος είναι υπεύθυνη για τις πέντε διαφορετικές κορυφές που δημιουργούνται στην καμπύλη της ισχύος. Το πρόβλημα εντοπισμού του σημείου MPP, που καλούνται να λύσουν οι αλγόριθμοι, γίνεται πιο δύσκολο και πιο περίπλοκο στο συγκεκριμένο πείραμα. Οι τιμές που χρησιμοποιήθηκαν στο συγκεκριμένο πείραμα παρουσιάζονται στον παρακάτω πίνακα.

Πίνακας 6.7. Πίνακας τιμών για πειραματικό σενάριο 3

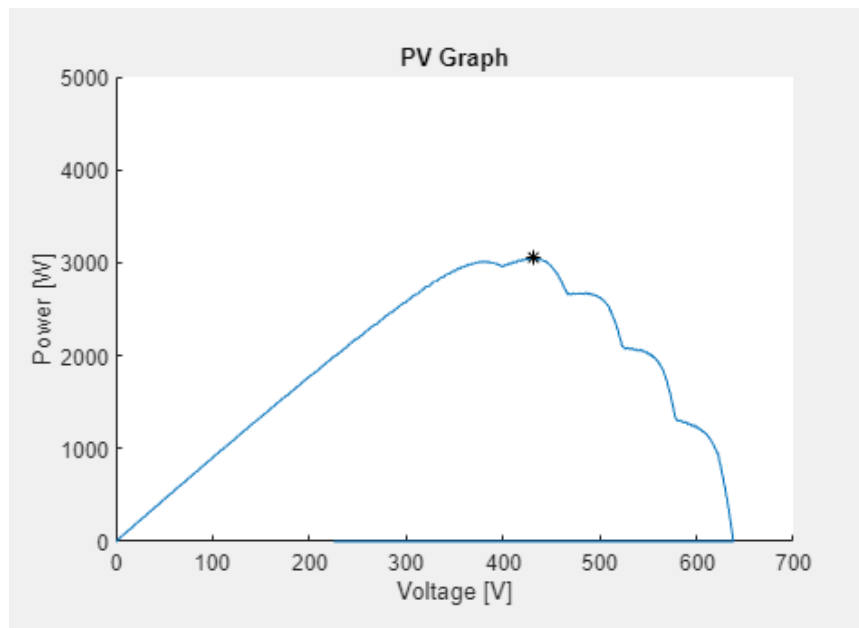
Μεταβλητή	Τιμή	Περιγραφή
Nss_1	12	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	1	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
Nss_3	1	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	1	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	1	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	50° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	40° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	35° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	30° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	20° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	1000 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	800 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	600 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	400 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	200 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

Από τα δεδομένα του πίνακα 6.7 παρατηρείται μια φθίνουσα τάση στις τιμές της θερμοκρασίας και της ακτινοβολίας. Οι διαφορετικές τιμές στις παραμέτρους είναι υπεύθυνες για τα πολλαπλά τοπικά μέγιστα που εμφανίζονται στις καμπύλες PV και IV, εικόνες 6.13 και 6.14.

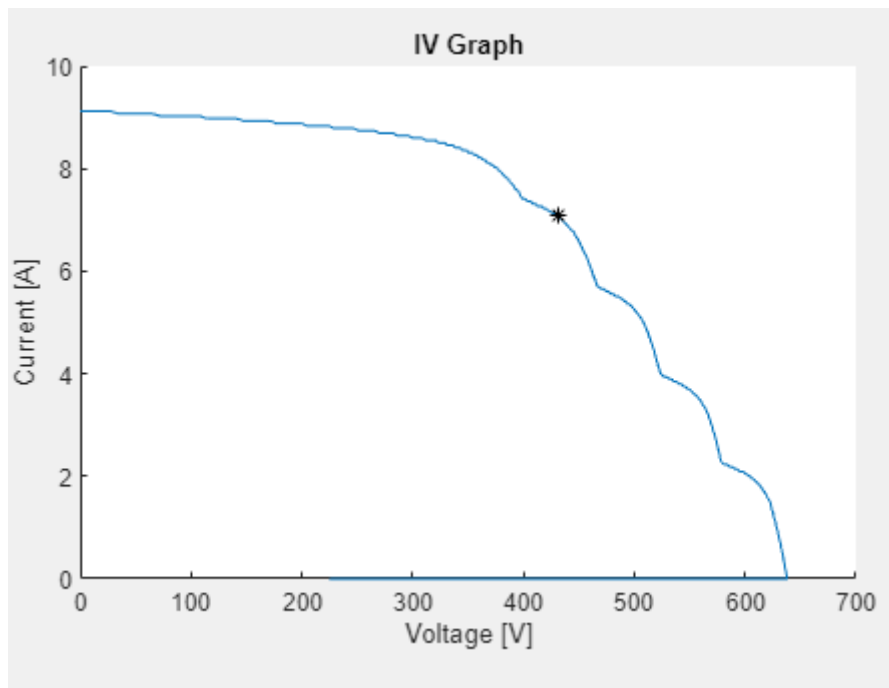
Πίνακας 6.8. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 3

Μεταβλητή	Τιμή	Περιγραφή
Po	3006 W	Μέγιστη ισχύς 3 ^{ου} πειράματος
Vo	441.1 V	Μέγιστη τάση 3 ^{ου} πειράματος

I_o	6.84 A	Μέγιστο ρεύμα 3 ^ο πειράματος
----------------------	--------	---



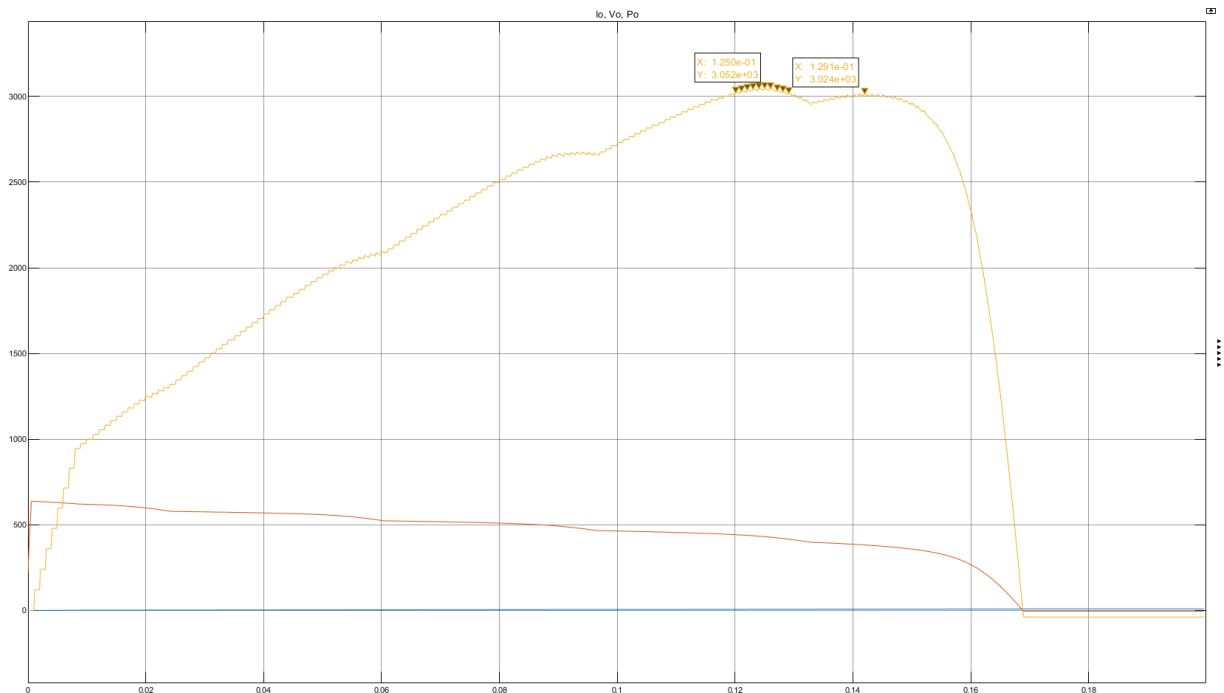
Εικόνα 6.22. Γράφημα PV, πείραμα 3



Εικόνα 6.23. Γράφημα IV, πείραμα 3

Από τα παραπάνω γραφήματα παρατηρείται ότι το ολικό μέγιστο της καμπύλης PV βρίσκεται ανάμεσα σε τοπικά μέγιστα. Ωστόσο, το συγκεκριμένο πρόβλημα απαιτεί περαιτέρω ανάλυση, ώστε να εξαχθεί ένα ασφαλές συμπέρασμα σχετικά με την απόδοση των αλγορίθμων. Παρατηρώντας την εικόνα 6.13, παρατηρείται ότι υπάρχουν δύο μέγιστα, τα οποία έχουν πολύ κοντινή τιμή. Εκτελώντας την προσομοίωση με μικρότερο βήμα, προκύπτουν τα παρακάτω συμπεράσματα και κυματομορφές. Το ολικό μέγιστο της ισχύος εντοπίζεται στα 3052 W. Η αμέσως επόμενη διαθέσιμη μέγιστη τιμή της

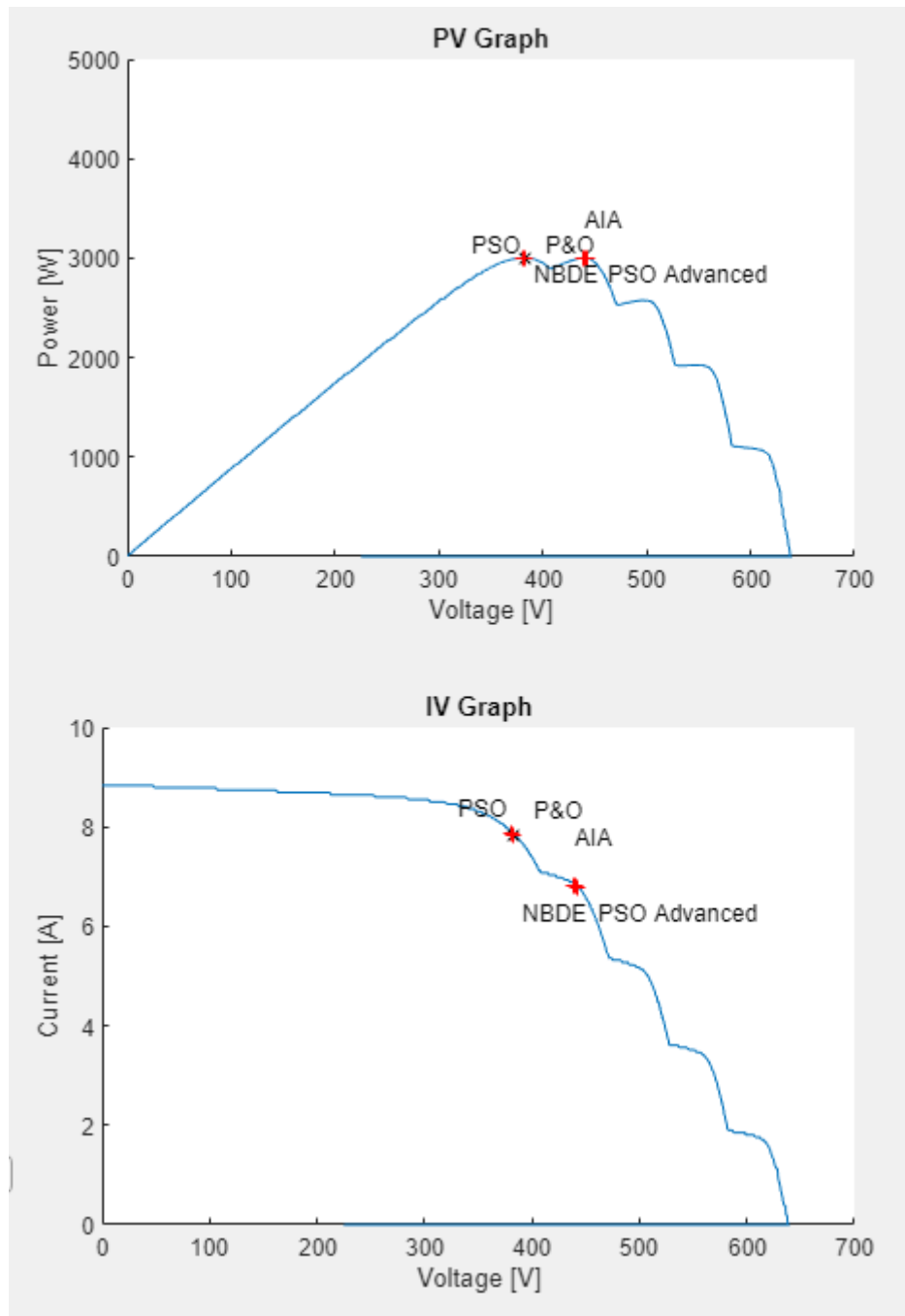
ισχύος εμφανίζεται στα 3024 W – δεύτερο τοπικό μέγιστο. Υπάρχει, δηλαδή, μια διαφορά της τάξης του 1 % μεταξύ ολικού μεγίστου και ενός από τα τοπικά μέγιστα.



Εικόνα 6.24. Καμπύλες P,V,I (άξονας Y) σε συνάρτηση με το χρόνο της προσομοίωσης (άξονας x), πείραμα 3

Λόγω της παραπάνω διαφοράς μεταξύ του ολικού και τοπικού μεγίστου, θεωρήθηκε ότι ο αλγόριθμος θα έχει συγκλίνει επιτυχώς, αν συγκλίνει, σε οποιοδήποτε από τα δύο αυτά σημεία. Η παραπάνω θεώρηση έγινε, διότι τα αποτελέσματα του αλγορίθμου επηρεάζονται άμεσα από την συχνότητα δειγματοληψίας της ισχύος. Για όλα τα πειράματα που έγιναν η συχνότητα δειγματοληψίας ήταν 1 ms. Σε ένα πραγματικό σύστημα η συγκεκριμένη συχνότητα θα είναι αρκετά μεγαλύτερη. Ο λόγος που επιλέχθηκε η συγκεκριμένη τιμή, ήταν για να κρατηθεί ο χρόνος προσομοίωσης σε μια λογική τιμή, διότι πολύ συχνά από το αποτέλεσμα της προσομοίωσης προέκυπταν διορθώσεις στον αλγόριθμο και απαιτούνταν αρκετά αποτελέσματα για να επιβεβαιωθεί η σωστή του λειτουργία. Συνεπώς, κρατώντας υψηλό χρόνο για την εκτέλεση της προσομοίωσης, η διόρθωση και η ρύθμιση του αλγορίθμου θα ήταν μια επώδυνη διαδικασία. Έχοντας ρυθμίσει τους αλγορίθμους για μέγιστη απόδοση με συχνότητα δειγματοληψίας 1 ms, αποφασίστηκε να υιοθετηθεί ο συγκεκριμένος χρόνος και για την υλοποίηση των πειραμάτων. Μπορεί φαινομενικά να υπάρχει διαφορά στην τιμή της ισχύος, όμως το ζητούμενο είναι ο αλγόριθμος να είναι ικανός να υποβοηθήσει το σύστημα στην άντληση της μέγιστης δυνατής ισχύος. Συνεπώς, απώλειες της τάξεως 1% θεωρούνται αμελητέες για το συγκεκριμένο πείραμα και δεδομένου ότι η συχνότητα δειγματοληψίας είναι 1 ms. Το βασικό κομμάτι του πειράματος είναι οι αλγόριθμοι να μπορέσουν να καταλήξουν σύντομα σε κάποιο από τα δύο σημεία, τα οποία δίνουν την ικανότητα στο φωτοβολταϊκό σύστημα να αντλεί τη μέγιστη δυνατή ισχύ.

Τα αποτελέσματα της προσομοίωσης παρουσιάζονται στη συνέχεια.

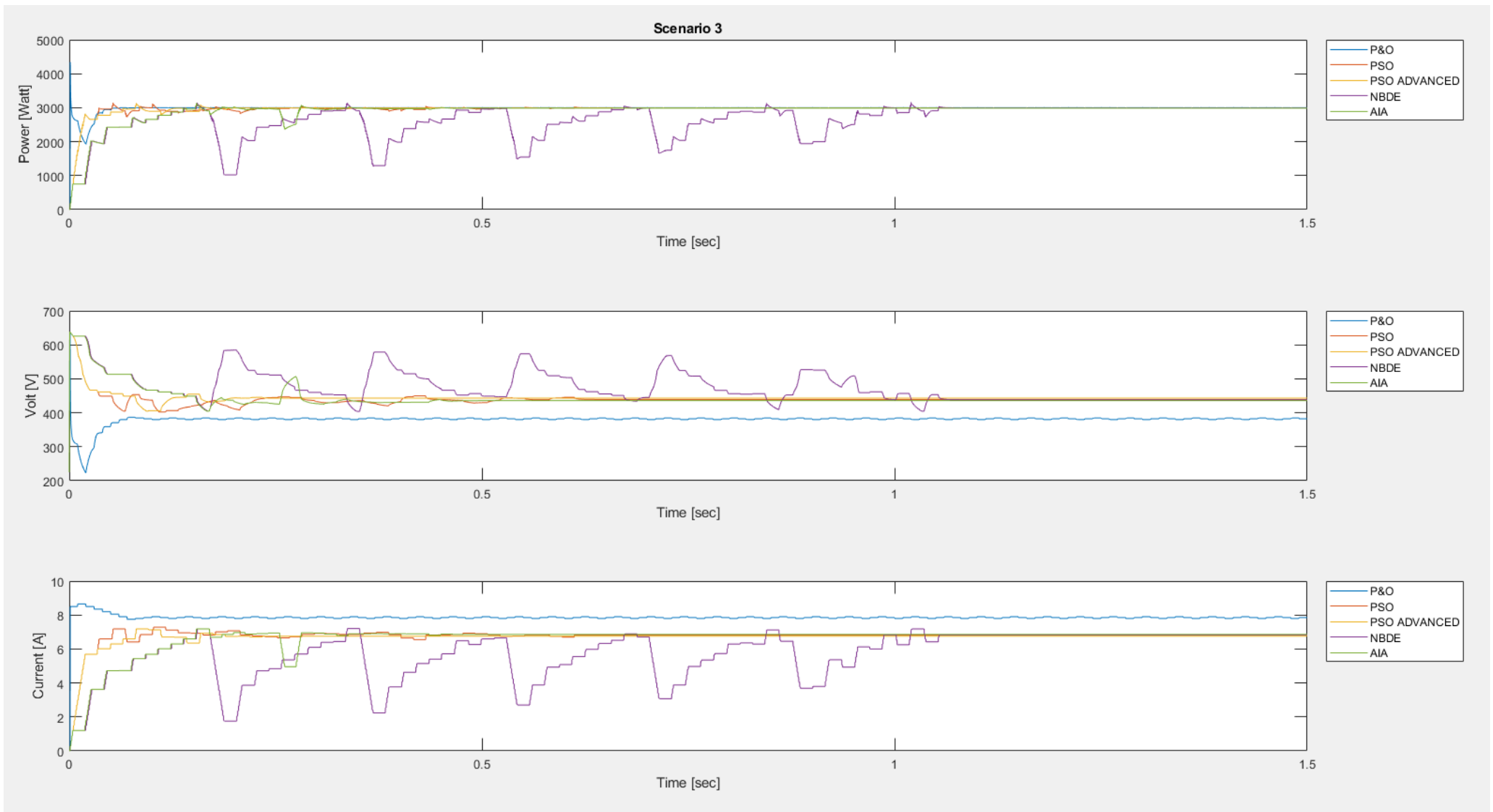


Εικόνα 6.25. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 3

Πίνακας 6.9. Αποτελέσματα προσομοίωσης, πείραμα 3

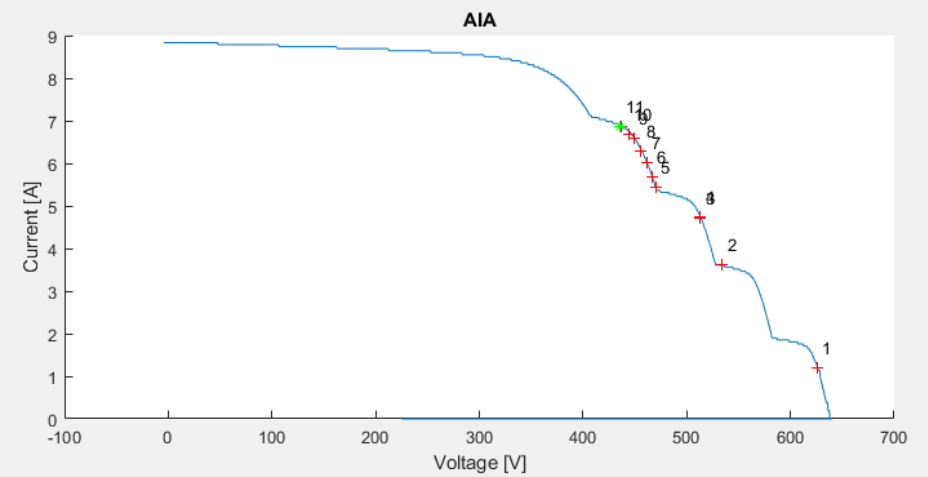
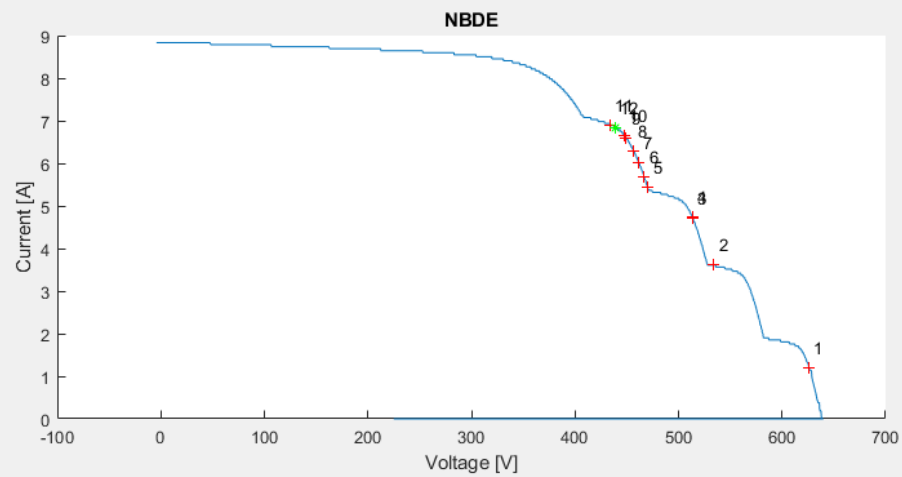
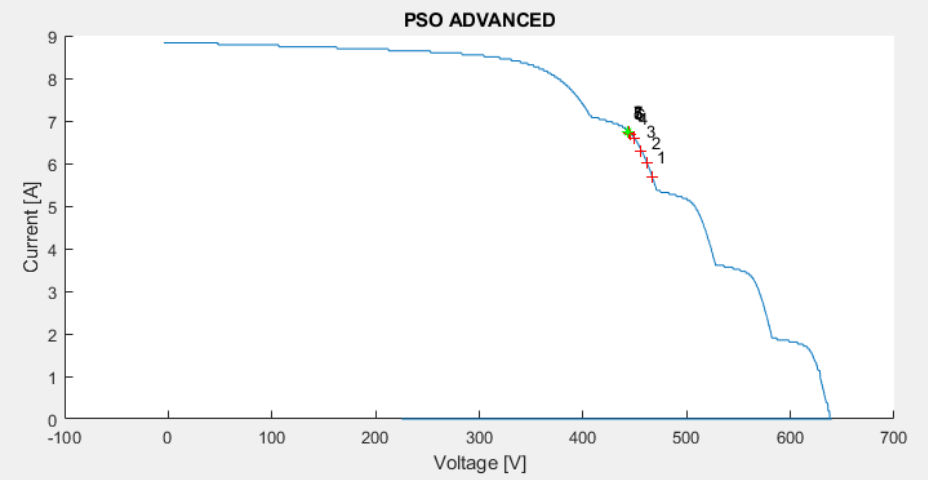
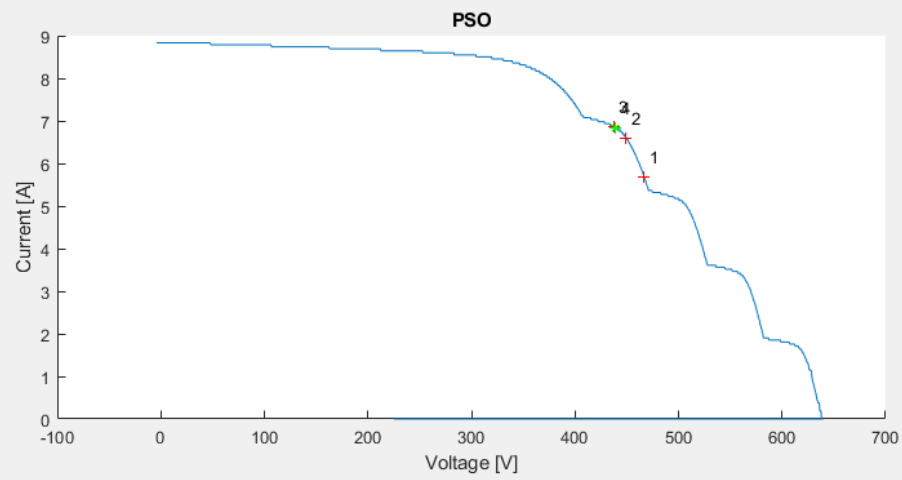
Αλγόριθμος	P_{max} [W]	V_{max} [V]	I_{max} [A]	Acc [%]	Χρόνος [sec]
-	3006.97	441.10	6.84	100.00	0.00
P&O	3000.85	383.02	7.85	99.80	0.09
PSO	2995.45	439.07	6.82	99.62	0.61
PSO Advanced	2993.83	443.14	6.76	99.56	0.23
NBDE	2994.30	438.34	6.83	99.58	1.05
AIA	2987.87	435.72	6.86	99.36	0.47

Από τα παραπάνω αποτελέσματα παρατηρείται ότι όλοι αλγόριθμοι, εκτός του PSO, κατέληξαν στο ίδιο αποτέλεσμα, το οποίο είναι ένα τοπικό μέγιστο με τιμή ισχύος πολύ κοντά στην τιμή του ολικού μεγίστου. Ο PSO κατέληξε στη σωστή περιοχή – περιοχή που εντοπίζεται το ολικό μέγιστο – αλλά σε λίγο χαμηλότερη τιμή ρεύματος από την τιμή που επιφέρει τη μέγιστη ισχύ και για αυτό το λόγο εμφανίζει ελαφρώς μειωμένη ισχύ σε σχέση με τους υπόλοιπους αλγορίθμους. Το γενικό συμπέρασμα που εξάγεται για το συγκεκριμένο πείραμα, είναι ότι όλοι οι αλγόριθμοι εμφανίζουν μεγάλη απόδοση, με τον PSO Advanced να εμφανίζει και τη γρηγορότερη σύγκλιση. Από τη σύγκριση για την ταχύτητα σύγκλισης εξαιρείται ο P&O, διότι οι εσωτερικοί μηχανισμοί λειτουργίας του δεν είναι ίδιοι με των υπολοίπων και επίσης το γεγονός ότι εμφανίζει τη συγκεκριμένη απόδοση, είναι ένα τυχαίο γεγονός που εξαρτάται από την τιμή του ρεύματος εκκίνησης. Ο NBDE εμφανίζει χρόνο σύγκλισης περίπου 1 sec, που είναι ο ίδιος χρόνος με τον χρόνο σύγκλισης του πειράματος 1 και 2. Όσον αφορά την ταχύτητα σύγκλισης, ο PSO εμφανίζει τον ίδιο περίπου χρόνο σύγκλισης με τον αλγόριθμο AIA.



Εικόνα 6.26. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 3

Algorithm steps



Εικόνα 6.27. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 3

6.6 Πείραμα 4^ο

Το συγκεκριμένο πείραμα εμφανίζει ομοιότητες με το πείραμα 3. Η διαφορά τους εντοπίζεται στον αριθμό των φωτοβολταϊκών πάνελ, που τοποθετήθηκαν σε κάθε φωτοβολταϊκή μονάδα. Συγκεκριμένα, η δεύτερη φωτοβολταϊκή μονάδα ενισχύθηκε με τρία επιπλέον πάνελ, τα οποία αφαιρέθηκαν από την πρώτη. Το αποτέλεσμα της μετατόπισης των πάνελ αύξησε τη διακριτότητα μεταξύ των μεγίστων σημείων και αναμένεται ότι όλοι οι αλγόριθμοι – εκτός του P&O – είναι ικανοί να εντοπίσουν με ακρίβεια το βέλτιστο σημείο. Οι τιμές που χρησιμοποιήθηκαν στο πείραμα 4 απεικονίζονται στο παρακάτω πίνακα.

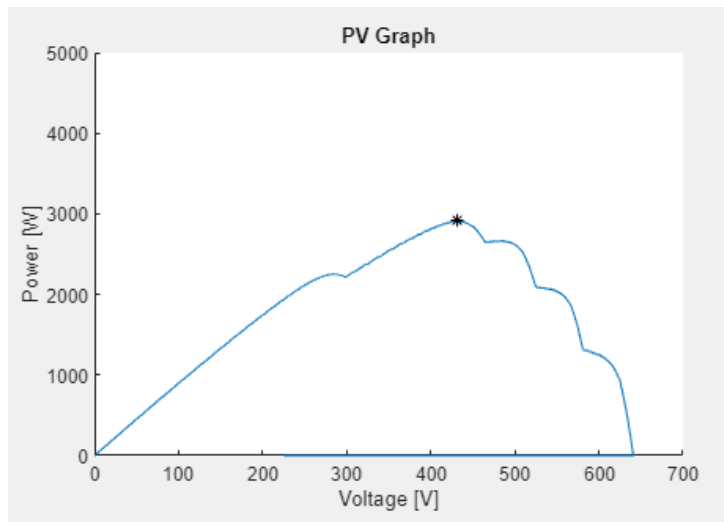
Πίνακας 6.10. Πίνακας τιμών για πειραματικό σενάριο 4

Μεταβλητή	Τιμή	Περιγραφή
Nss_1	9	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	4	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
Nss_3	1	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	1	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	1	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	50° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	40° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	35° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	30° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	25° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	1000 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	800 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	600 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	400 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	200 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

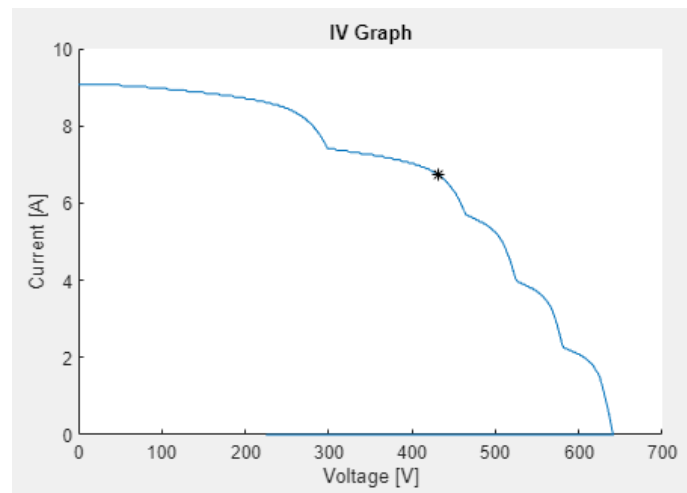
Από τα δεδομένα του πίνακα 6.10, παρατηρείται μια φθίνουσα τάση στις τιμές της θερμοκρασίας και της ακτινοβολίας. Οι διαφορετικές τιμές στις παραμέτρους είναι υπεύθυνες για τα πολλαπλά τοπικά μέγιστα που εμφανίζονται στις καμπύλες PV και IV, εικόνες 6.18 και 6.19.

Πίνακας 6.11. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 4

Μεταβλητή	Τιμή	Περιγραφή
Po	2889.19 W	Μέγιστη ισχύς 4 ^{ου} πειράματος
Vo	436.2 V	Μέγιστη τάση 4 ^{ου} πειράματος
Io	6.65 A	Μέγιστο ρεύμα 4 ^{ου} πειράματος



Εικόνα 6.28. Γράφημα PV, πείραμα 4

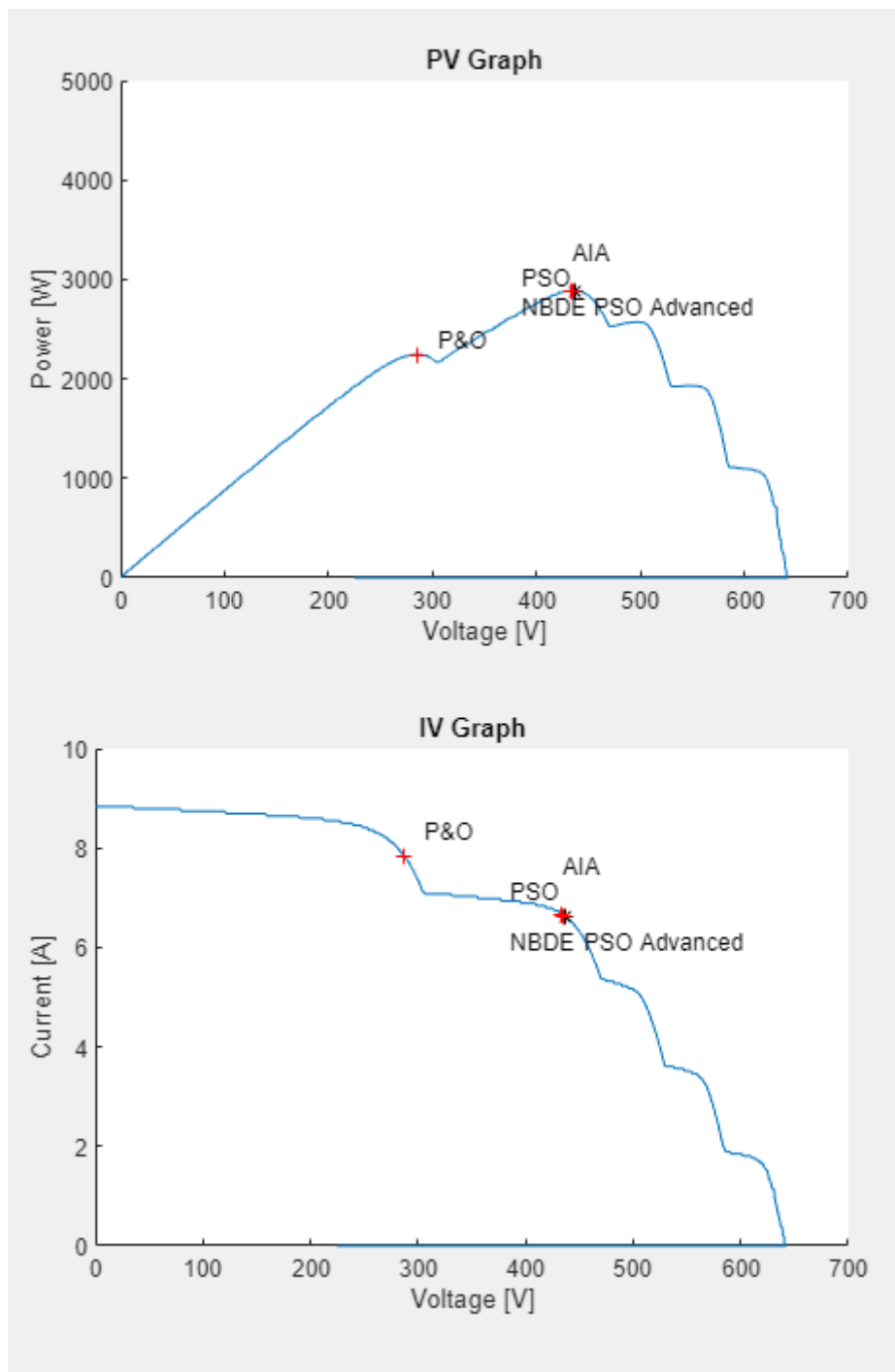


Εικόνα 6.29. Γράφημα IV, πείραμα 4

Με βάση τα παραπάνω δεδομένα παρατηρείται ότι το συγκεκριμένο πείραμα αποτελεί μια απλούστερη εκδοχή του πειράματος 3. Τα αποτελέσματα αναμένεται να είναι παραπλήσια, όσον αφορά τις τιμές των αποτελεσμάτων. Τα αποτελέσματα της προσομοίωσης παρουσιάζονται στη συνέχεια.

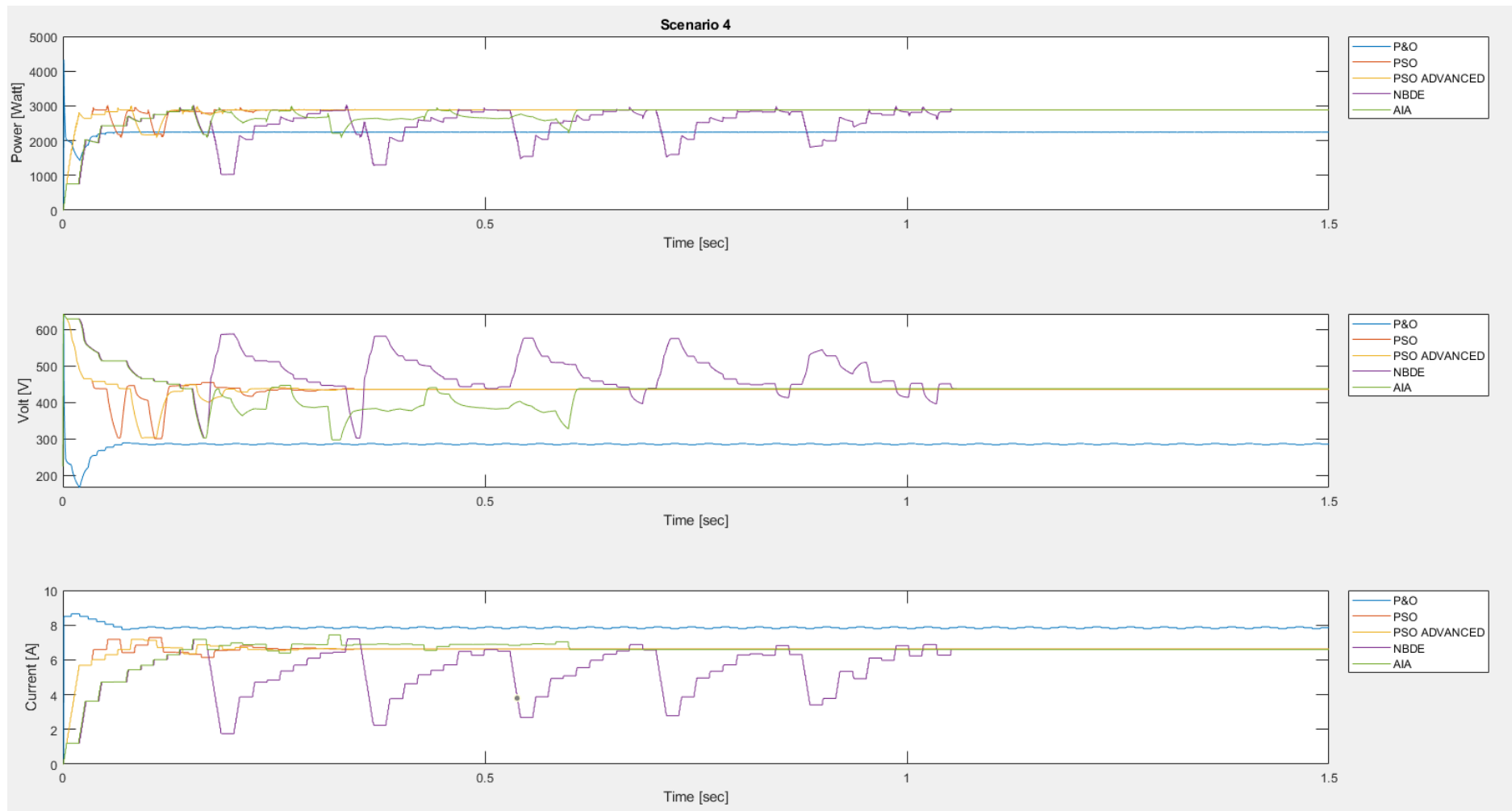
Πίνακας 6.12. Αποτελέσματα προσομοίωσης, πείραμα 4

Αλγόριθμος	$P_{\max}$ [W]	$V_{\max}$ [V]	$I_{\max}$ [A]	Acc [%]	Χρόνος [sec]
-	2889.19	436.20	6.65	100.00	0.00
P&O	2244.10	286.67	7.85	77.67	0.09
PSO	2885.53	435.27	6.63	99.87	0.35
PSO Advanced	2885.52	435.61	6.62	99.87	0.32
NBDE	2884.56	437.47	6.59	99.84	1.05
AIA	2884.56	437.47	6.59	99.84	0.60



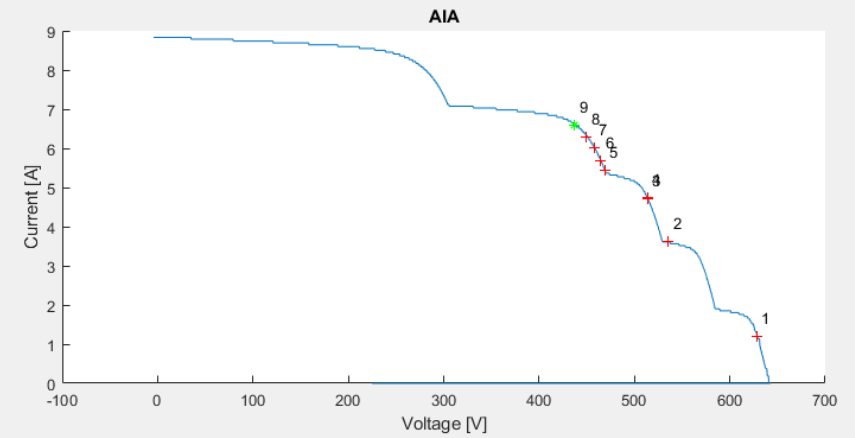
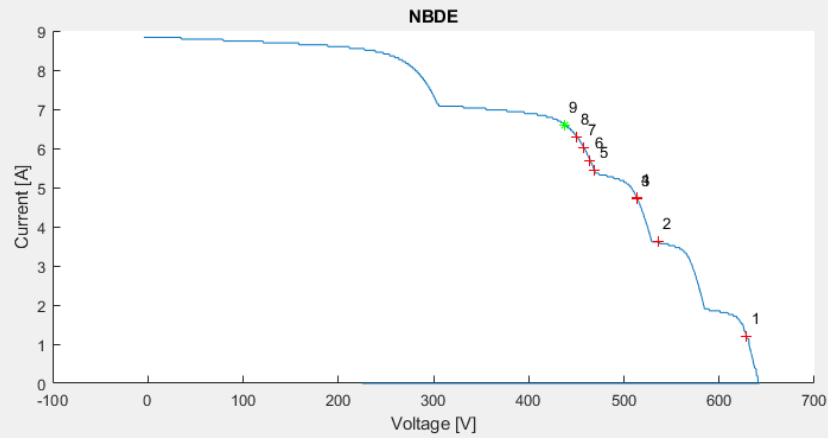
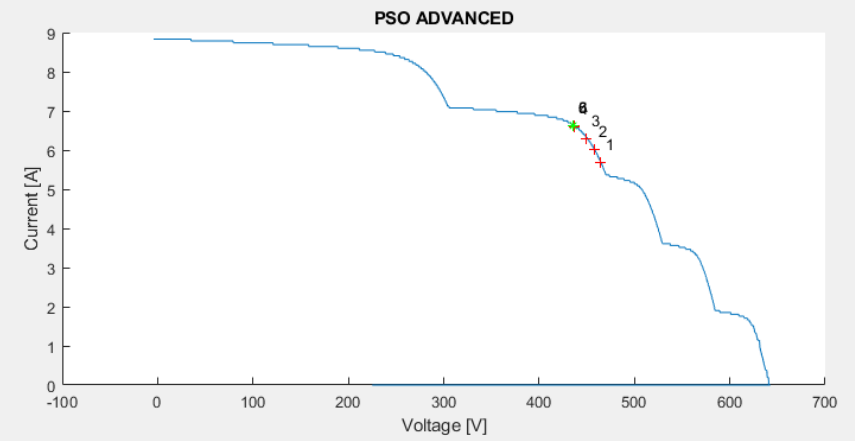
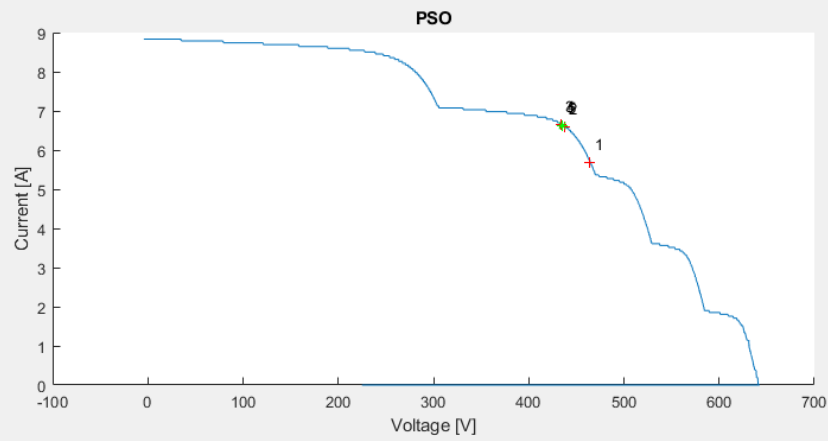
Εικόνα 6.30. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 4

Η αρχική πρόβλεψη που έγινε για την επιτυχία των αλγορίθμων, επιβεβαιώθηκε. Παρατηρώντας τα παραπάνω αποτελέσματα, διακρίνεται το υψηλό ποσοστό ακρίβειας. Ο P&O αστόχησε να εντοπίσει το πραγματικό μέγιστο, καταλήγοντας όμως σε κάποια λύση πιο σύντομα από όλους τους υπόλοιπους αλγορίθμους. Επίσης, ο αλγόριθμος PSO Advanced εμφανίζει το δεύτερο συντομότερο χρόνο σύγκλισης, με επόμενο τον AIA. Ο NBDE παρουσιάζει σταθερή τιμή σύγκλισης, περίπου 1 sec. Το γενικό συμπέρασμα που εξάγεται από το συγκεκριμένο πειραματισμό, είναι ότι οι αλγόριθμοι παρουσιάζουν παρόμοια απόδοση στα πειράματα 3 και 4.



Εικόνα 6.31. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 4

Algorithm steps



Εικόνα 6.32. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 4

6.7 Πείραμα 5^ο

Για το πείραμα 5 χρησιμοποιήθηκαν τιμές παραμέτρων, ώστε το ολικό μέγιστο να εμφανιστεί στην αριστερή πλευρά της καμπύλης PV, εικόνα 6.22. Οι παράμετροι του πειράματος απεικονίζονται στον πίνακα 6.13.

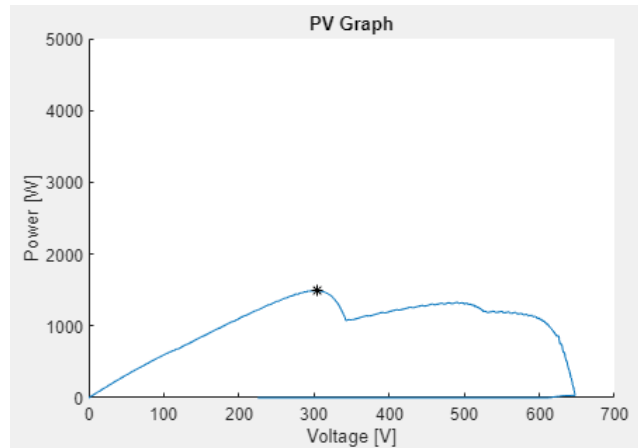
Πίνακας 6.13. Πίνακας τιμών για πειραματικό σενάριο 5

Μεταβλητή	Τιμή	Περιγραφή
Nss_1	5	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	5	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
Nss_3	3	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	2	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	1	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	27° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	35° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	37° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	25° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	36° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	300 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	600 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	700 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	200 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	650 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

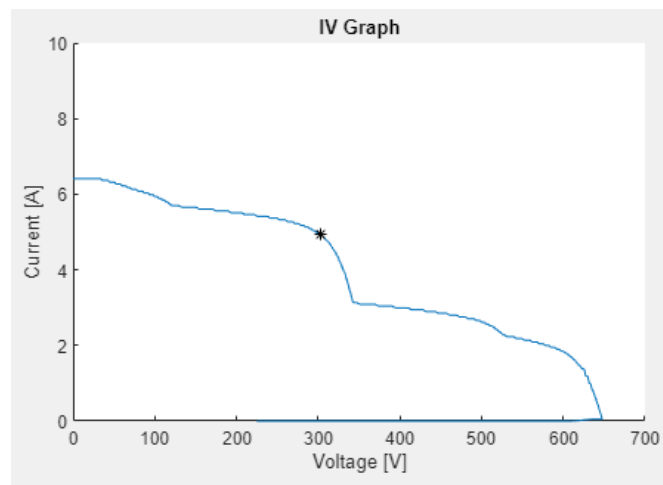
Από τα δεδομένα του πίνακα 6.13, παρατηρούνται τρεις μεγάλες τιμές για την ακτινοβολία, οι οποίες είναι υπεύθυνες για τα τρία μέγιστα που εμφανίζονται στις καμπύλες PV και IV, εικόνες 6.22 και 6.23.

Πίνακας 6.14. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 5

Μεταβλητή	Τιμή	Περιγραφή
Po	1481.1 W	Μέγιστη ισχύς 5 ^{ου} πειράματος
Vo	304.3 V	Μέγιστη τάση 5 ^{ου} πειράματος
Io	4.89 A	Μέγιστο ρεύμα 5 ^{ου} πειράματος



Εικόνα 6.33. Γράφημα PV, πείραμα 5

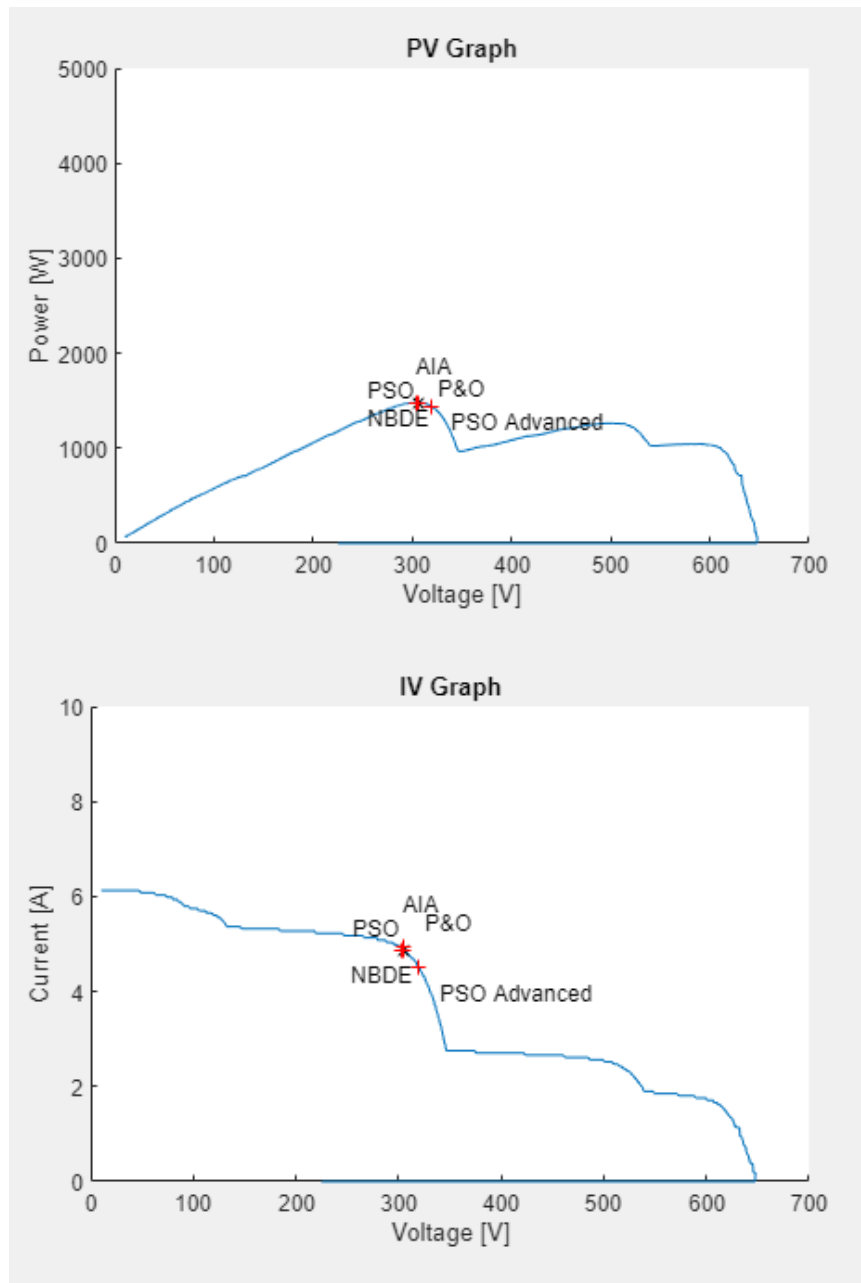


Εικόνα 6.34. Γράφημα IV, πείραμα 5

Το ολικό μέγιστο εντοπίζεται στην αριστερή πλευρά της καμπύλης PV και αναμένεται ο εντοπισμός του σημείου MPP από όλους τους αλγορίθμους.

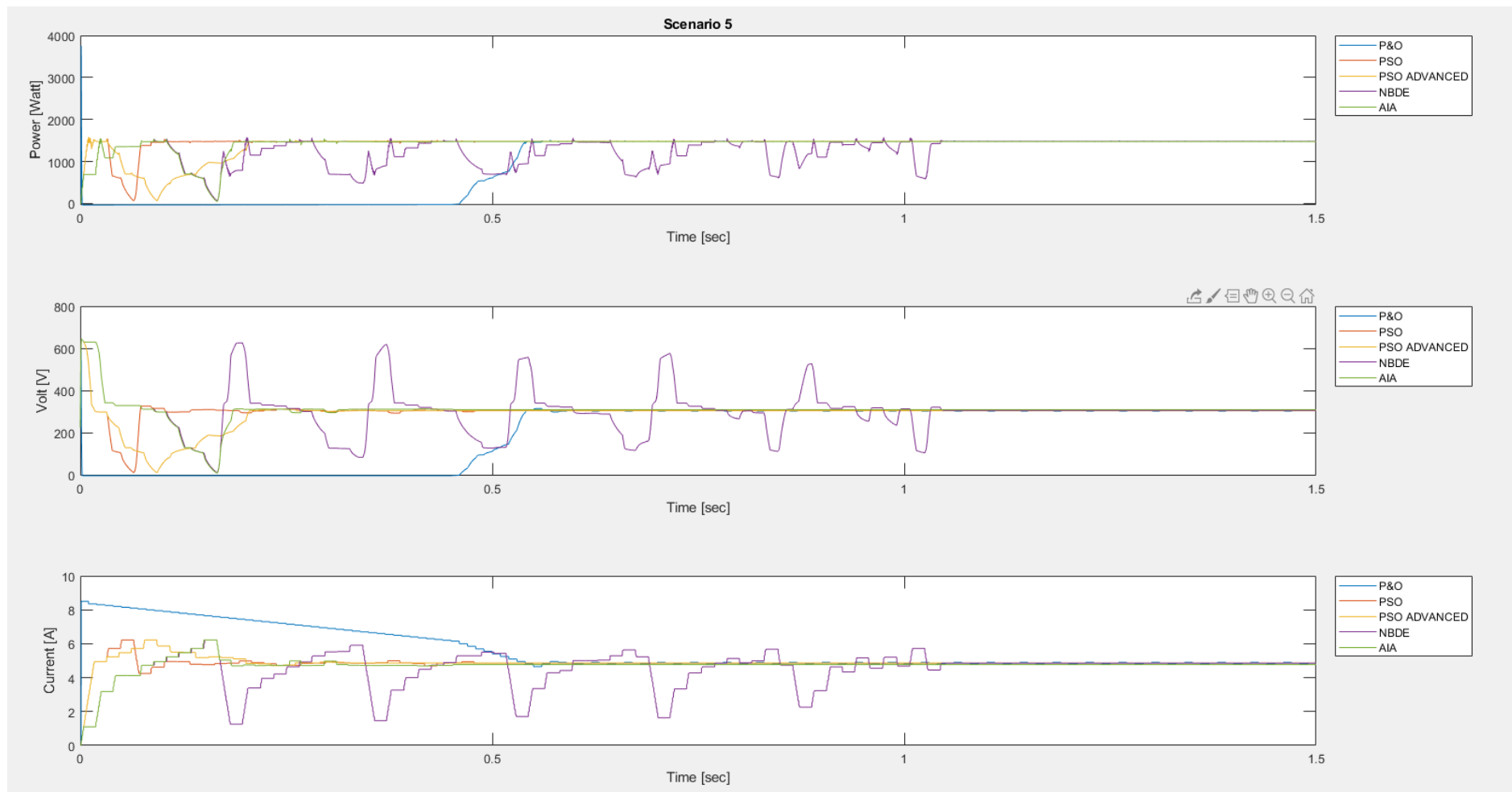
Πίνακας 6.15. Αποτελέσματα προσομοίωσης, πείραμα 5

Αλγόριθμος	$P_{\max}$ [W]	$V_{\max}$ [V]	$I_{\max}$ [A]	Acc [%]	Χρόνος [sec]
-	1481.08	304.29	4.89	100.00	0.00
P&O	1481.98	304.03	4.95	100.06	0.57
PSO	1479.27	304.50	4.86	99.88	0.48
PSO Advanced	1479.28	304.21	4.86	99.88	0.35
NBDE	1479.22	304.87	4.85	99.87	1.04
AIA	1475.36	309.17	4.77	99.61	0.45



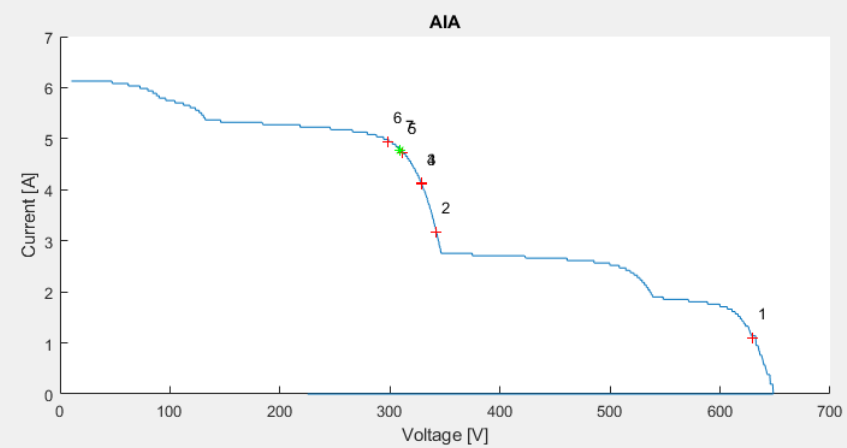
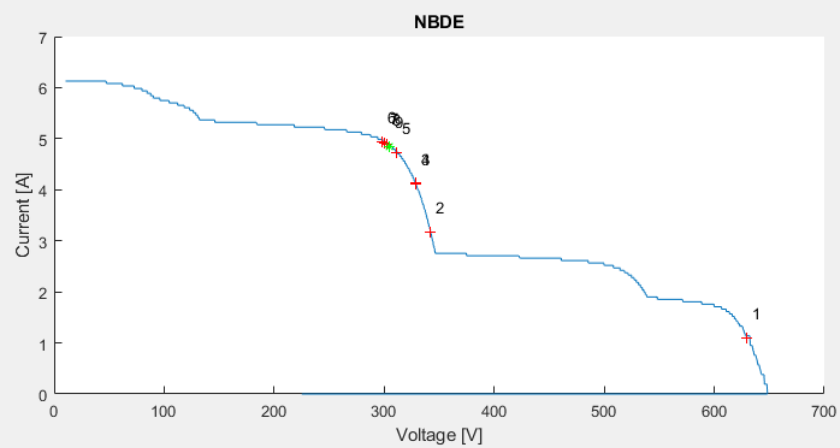
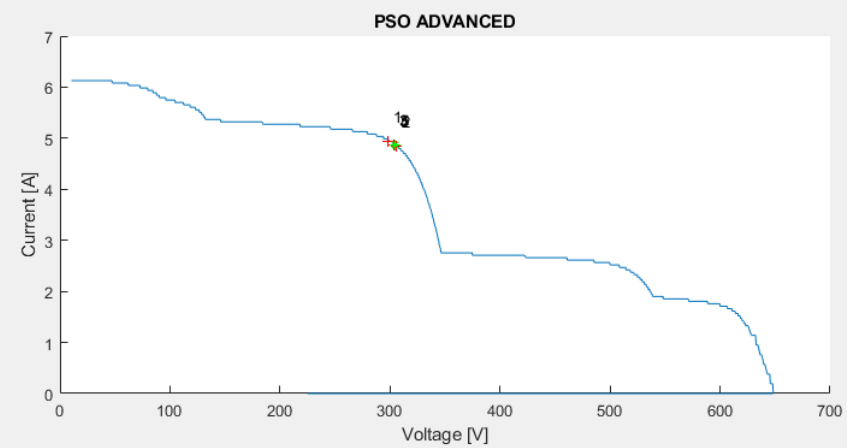
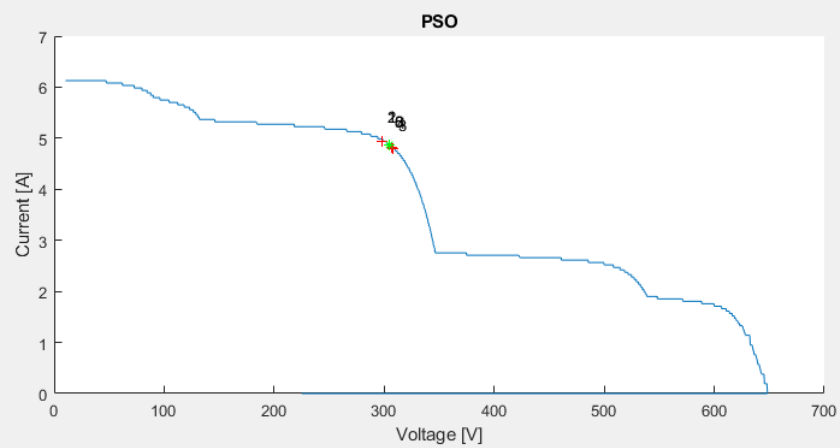
Εικόνα 6.35. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 5

Από τα παραπάνω αποτελέσματα παρατηρείται ότι όλοι οι αλγόριθμοι κατέληξαν στο ίδιο αποτέλεσμα, το οποίο είναι ένα τοπικό μέγιστο με τιμή ισχύος πολύ κοντά στην τιμή του ολικού μεγίστου. Το γενικό συμπέρασμα που εξάγεται για το συγκεκριμένο πείραμα, είναι ότι όλοι οι αλγόριθμοι εμφανίζουν μεγάλη απόδοση, με τον PSO Advanced να εμφανίζει για άλλη μια φορά τη γρηγορότερη σύγκλιση. Από τη σύγκριση για την ταχύτητα σύγκλισης εξαιρείται ο P&O, διότι οι εσωτερικοί μηχανισμοί λειτουργίας του δεν είναι ίδιοι με των υπολοίπων και επίσης, το γεγονός ότι εμφανίζει τη συγκεκριμένη απόδοση, είναι ένα τυχαίο γεγονός που εξαρτάται από την τιμή του ρεύματος εκκίνησης. Ο NBDE εμφανίζει χρόνο σύγκλισης περίπου 1 sec, που είναι ο ίδιος χρόνος με το χρόνο σύγκλισης του πειράματος 1, 2, 3 και 4. Όσον αφορά την ταχύτητα σύγκλισης, ο PSO εμφανίζει τον ίδιο περίπου χρόνο σύγκλισης με τον αλγόριθμο AIA.



Εικόνα 6.36. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 5

Algorithm steps



Εικόνα 6.37. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 5

6.8 Πείραμα 6°

Για το πείραμα 6 χρησιμοποιήθηκαν μειωμένες τιμές ακτινοβολίας και θερμοκρασίας, έτσι ώστε να γίνει έλεγχος της απόδοσης των αλγορίθμων σε συνθήκες χαμηλής ισχύος. Το συγκεκριμένο πείραμα όπως και το επόμενο, δοκιμάζουν τους αλγορίθμους σε συνθήκες μερικής σκίασης και χαμηλής ακτινοβολίας, που αποτελεί επίσης ένα ρεαλιστικό σενάριο, όπου είναι απαραίτητη η ορθή λειτουργία του συστήματος. Οι τιμές που χρησιμοποιήθηκαν για το πείραμα 6, απεικονίζονται στον παρακάτω πίνακα.

Πίνακας 6.16. Πίνακας τιμών για πειραματικό σενάριο 6

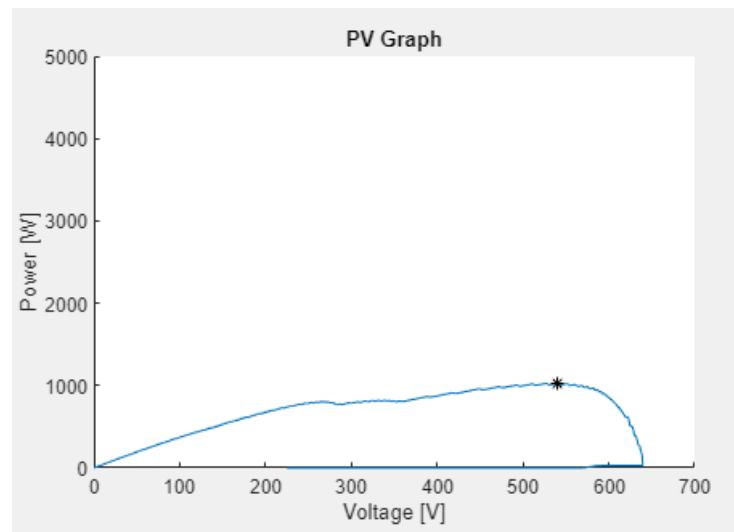
Μεταβλητή	Τιμή	Περιγραφή
Nss_1	5	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	3	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
Nss_3	5	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	1	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	2	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	30° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	28° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	25° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	25° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	26° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	400 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	350 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	200 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	200 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	250 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

Από τον πίνακα 6.16 απεικονίζονται οι τιμές για την ακτινοβολία και τη θερμοκρασία για κάθε φωτοβολταϊκή μονάδα. Οι τιμές υποδεικνύουν ένα ολικό μέγιστο. Η διαφορά του συγκεκριμένου πειράματος εντοπίζεται στις μειωμένες τιμές των παραμέτρων, που υποδεικνύουν ένα σύστημα σε λειτουργία χαμηλής ισχύος. Παρατηρώντας τον πίνακα 6.17, επιβεβαιώνεται η λειτουργία του συστήματος σε χαμηλή ισχύ.

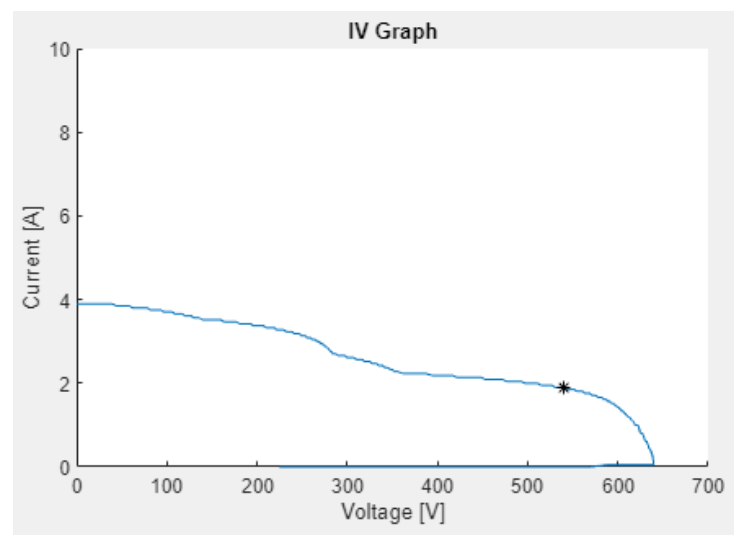
Πίνακας 6.17. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 6

Μεταβλητή	Τιμή	Περιγραφή
Po	958 W	Μέγιστη ισχύς 6 ^{ου} πειράματος
Vo	565.8 V	Μέγιστη τάση 6 ^{ου} πειράματος

I_o	1.71 A	Μέγιστο ρεύμα 6 ^ο πειράματος
----------------------	--------	---

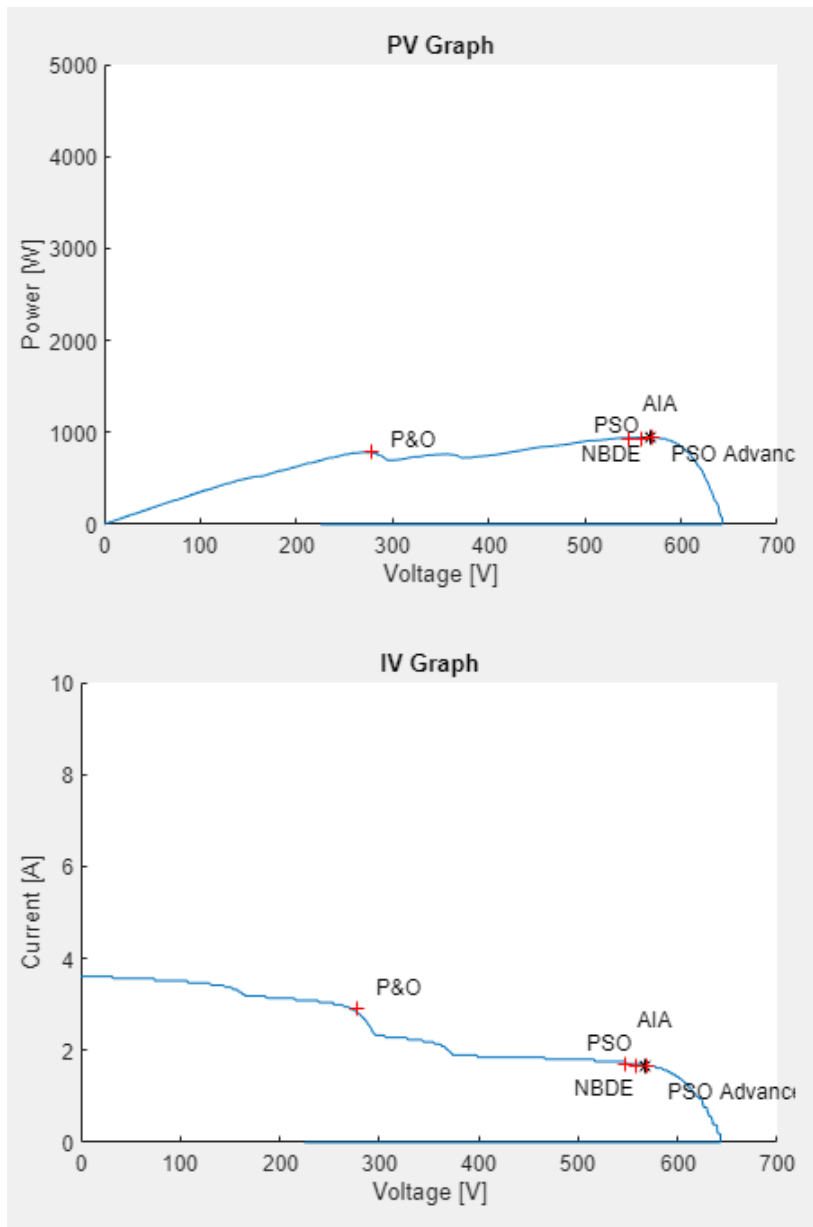


Εικόνα 6.38. Γράφημα PV, πείραμα 6



Εικόνα 6.39. Γράφημα IV, πείραμα 6

Παρατηρώντας τα γραφήματα των εικόνων 6.26 και 6.27, εντοπίζεται ένα ολικό μέγιστο σημείο στο δεξί τμήμα της καμπύλης PV και αναμένεται όλοι οι αλγόριθμοι να εντοπίσουν με επιτυχία το σημείο αυτό.



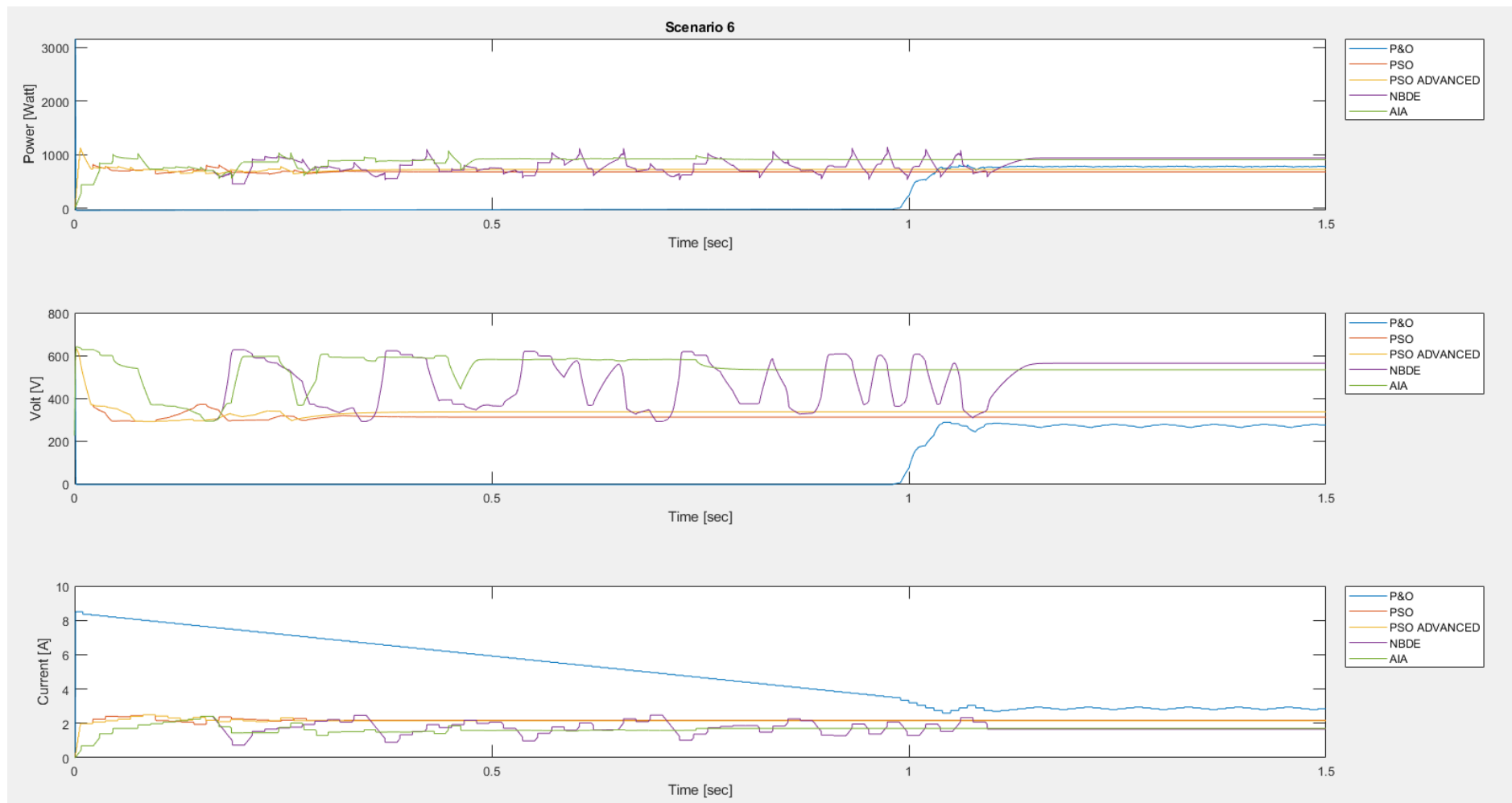
Εικόνα 6.40. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 6

Πίνακας 6.18. Αποτελέσματα προσομοίωσης, πείραμα 6

Αλγόριθμος	$P_{\max}$ [W]	$V_{\max}$ [V]	$I_{\max}$ [A]	Acc [%]	Χρόνος [sec]
-	958.01	565.87	1.71	100.00	0.00
P&O	790.23	277.16	2.90	82.49	1.06
PSO	681.31	311.80	2.19	71.12	0.32
PSO Advanced	719.36	333.63	2.16	75.09	0.28
NBDE	938.60	563.72	1.67	97.97	1.09
AIA	910.58	533.55	1.71	95.05	0.75

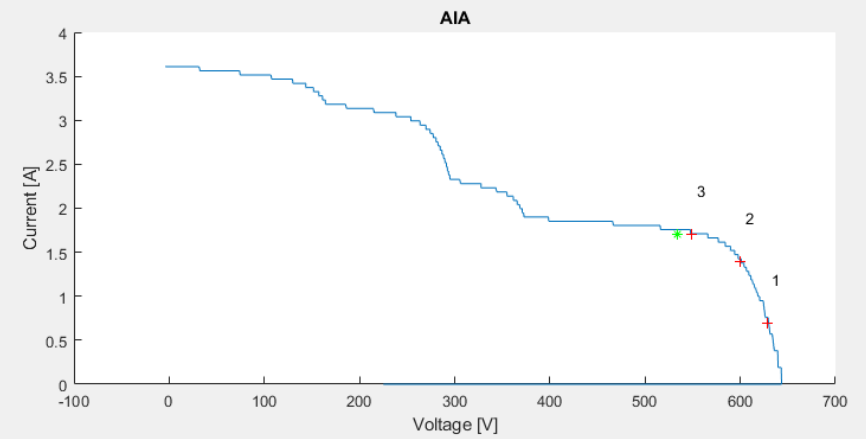
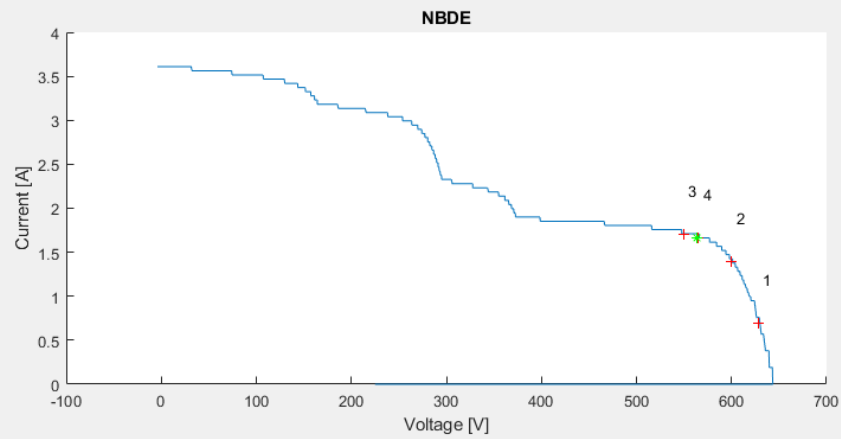
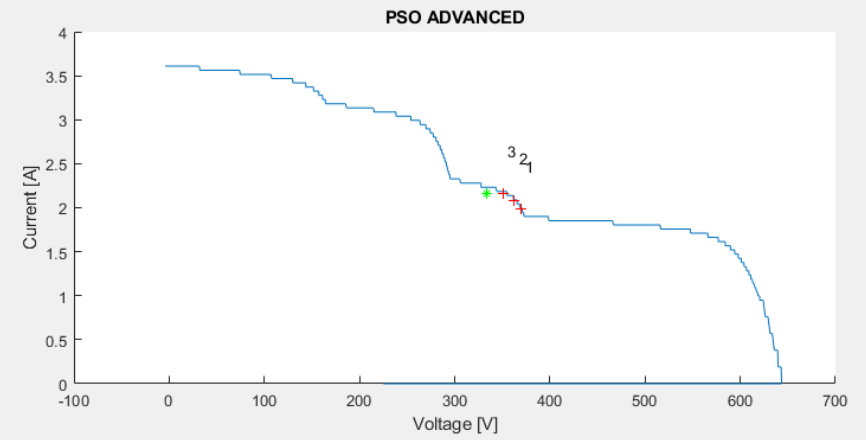
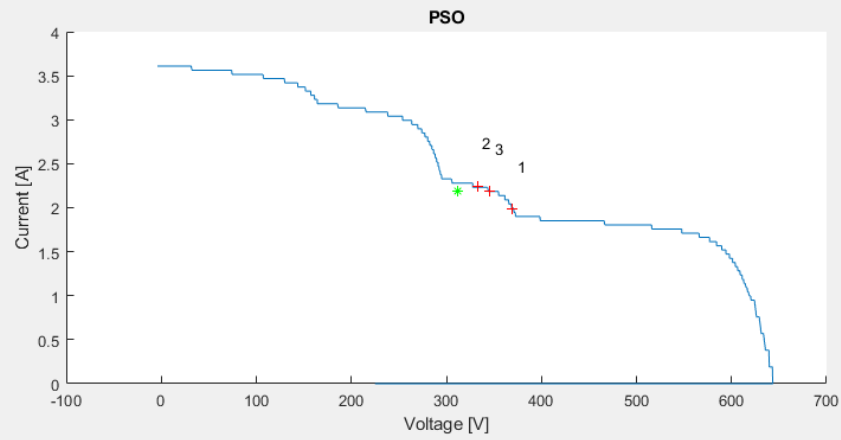
Από τα αποτελέσματα παρατηρείται ότι ο P&O και ο PSO_advanced είναι οι αλγόριθμοι που αστόχησαν στον εντοπισμό του πραγματικού σημείου MPP. Επίσης, οι υπόλοιποι αλγόριθμοι

εμφανίζουν εμφανώς μειωμένη απόδοση, σε σχέση με τα προηγούμενα πειράματα. Ένας λόγος που παρατηρείται το φαινόμενο αυτό, είναι το γεγονός ότι το σύστημα λειτουργεί σε πολύ χαμηλή ισχύ και οι αλγόριθμοι δυσκολεύονται να εντοπίσουν το MPP σημείο. Για παράδειγμα, για έξοδο από 1.65 A μέχρι 2.9 A, η απόδοση του συστήματος κυμαίνεται από 76 % έως 100%. Το μέγιστο σημείο, δηλαδή, πρέπει να εντοπιστεί στη συγκεκριμένη περιοχή. Αυτό έχει ως αποτέλεσμα την καθυστέρηση των αλγορίθμων και η απόδειξη της καθυστέρησης αυτής επιβεβαιώνεται, παρατηρώντας το χρόνο εκτέλεσής τους στον πίνακα 6.18. Ο AIA στα προηγούμενα πειράματα είχε παρόμοια απόδοση με τον PSO και στο συγκεκριμένο πείραμα παρατηρείται ότι καθυστερεί κατά 0.4 sec περίπου. Ο NBDE συγκλίνει και στο πείραμα 6 σε περίπου 1 sec, όπως και στα προηγούμενα πειράματα, ενώ ο PSO παρουσιάζεται σε αυτό το πείραμα ως ο ταχύτερος όλων των επιτυχόντων αλγορίθμων. Το γενικό συμπέρασμα που εξάγεται από το πείραμα 6, είναι ότι οι αλγόριθμοι PSO και AIA φαίνεται να επιτυγχάνουν παρόμοια απόδοση σε παρόμοιο χρόνο, ενώ αντίστοιχα ο NBDE εμφανίζει απόδοση εφάμιλλη των AIA και PSO, με τη διαφορά ότι η ταχύτητα σύγκλισής του εξαρτάται από τις ρυθμίσεις του. Επίσης, υπάρχει δυνατότητα να αυξηθεί η απόδοση των αλγορίθμων PSO, AIA και NBDE αλλάζοντας τις παραμέτρους τους. Εντούτοις, η αλλαγή των παραμέτρων τους εγκυμονεί κινδύνους, διότι υπάρχει περίπτωση να αυξηθεί η απόδοση για το πείραμα 6, αλλά να μειωθεί στα υπόλοιπα πειράματα. Η τελική ρύθμιση των παραμέτρων των αλγορίθμων προτείνεται να πραγματοποιηθεί σε φυσικό σύστημα.



Εικόνα 6.41. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 6

Algorithm steps



Εικόνα 6.42. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 6

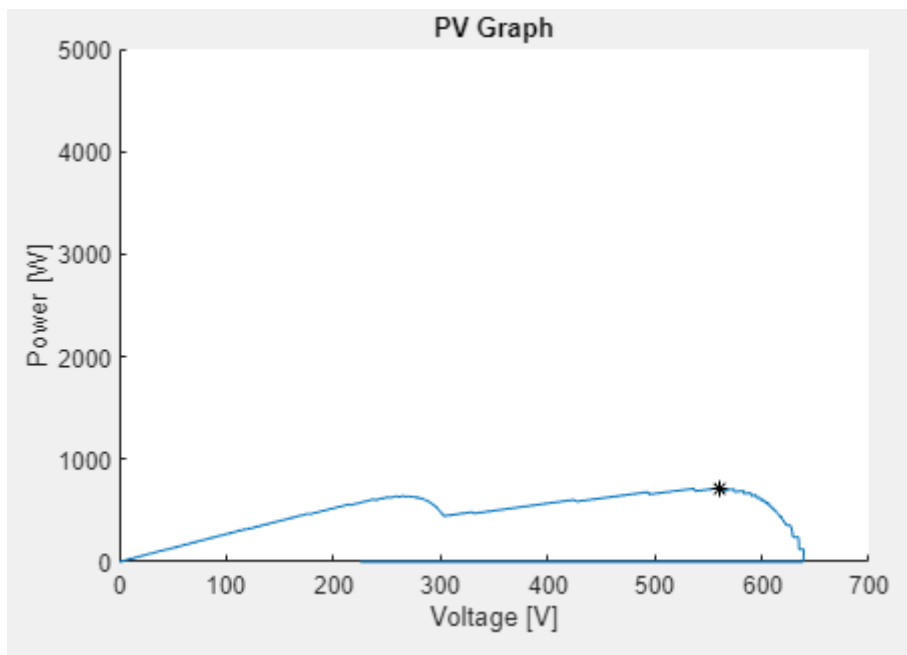
6.9 Πείραμα 7^ο

Όπως και στο πείραμα 6, το πείραμα 7 στοχεύει στον έλεγχο της λειτουργίας των αλγορίθμων σε χαμηλή ισχύ παρουσία μερικής σκίασης. Είναι ένα σενάριο το οποίο είναι πιθανό να εμφανιστεί σε μια πραγματική εγκατάσταση και συνεπώς, η αξιολόγηση των υλοποιημένων μεθόδων σε τέτοια σενάρια είναι απαραίτητη. Παρατηρώντας τις κυματομορφές PV και IV, εντοπίζεται με ευκολία το ολικό μέγιστο. Ωστόσο, η λειτουργία χαμηλής ισχύος αποδεικνύεται ιδιαίτερα απαιτητική για τους αλγορίθμους, όσον αφορά τον εντοπισμό του ολικού μεγίστου. Οι παράμετροι που χρησιμοποιήθηκαν για το συγκεκριμένο πείραμα είναι:

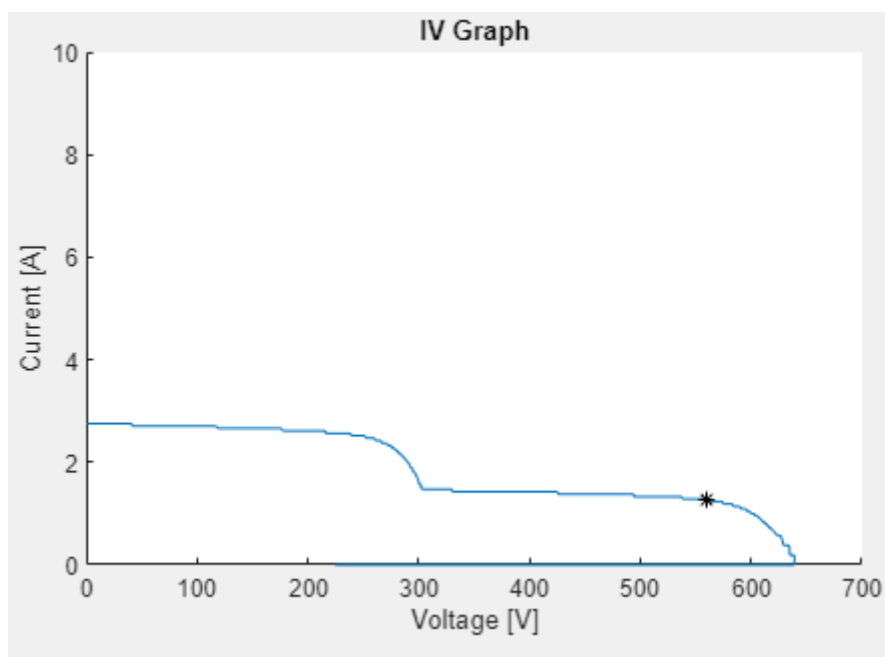
Πίνακας 6.19. Πίνακας τιμών για πειραματικό σενάριο 7

Μεταβλητή	Τιμή	Περιγραφή
Nss_1	4	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	4	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
Nss_3	4	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	3	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	1	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	27° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	27° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	22° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	22° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	26° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	300 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	300 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	150 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	150 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	150 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

Στον πίνακα 6.19 απεικονίζονται οι τιμές για την ακτινοβολία και τη θερμοκρασία για κάθε φωτοβολταϊκή μονάδα. Οι τιμές υποδεικνύουν ένα ολικό μέγιστο. Η διαφορά του συγκεκριμένου πειράματος εντοπίζεται στις μειωμένες τιμές των παραμέτρων, που υποδεικνύουν ένα σύστημα σε λειτουργία χαμηλής ισχύος. Παρατηρώντας τον πίνακα 6.20, επιβεβαιώνεται η λειτουργία του συστήματος σε χαμηλή ισχύ.



Εικόνα 6.43. Γράφημα PV, πείραμα 7

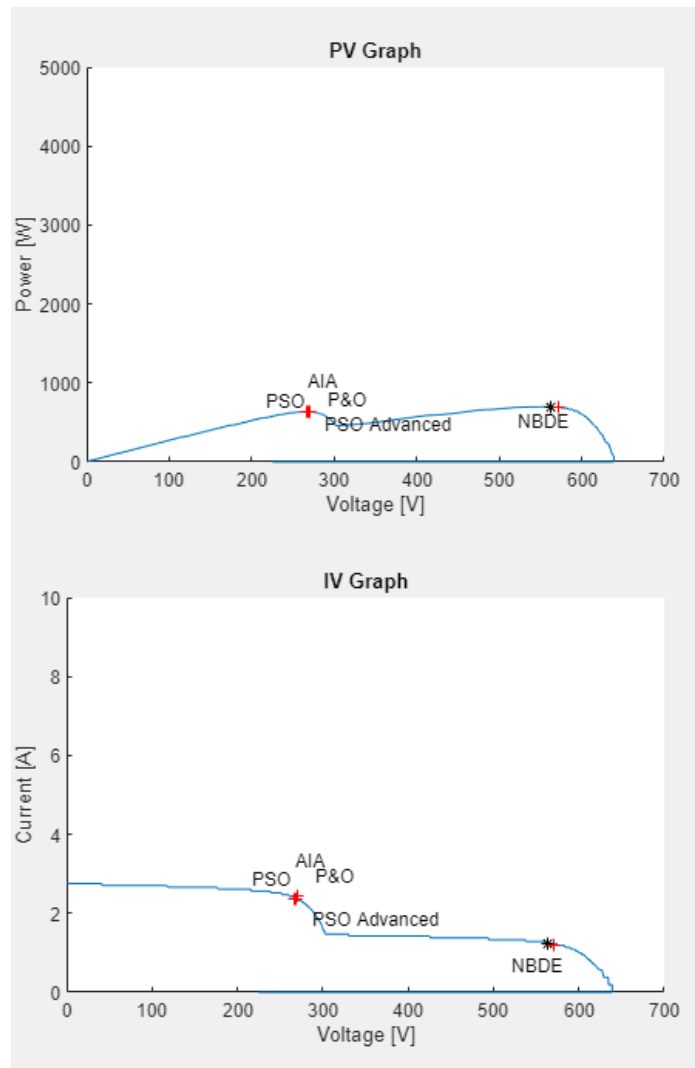


Εικόνα 6.44. Γράφημα IV, πείραμα 7

Πίνακας 6.20. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 7

Μεταβλητή	Τιμή	Περιγραφή
P_o	710 W	Μέγιστη ισχύς 7 ^{ου} πειράματος
V_o	560.2 V	Μέγιστη τάση 7 ^{ου} πειράματος
I_o	1.28 A	Μέγιστο ρεύμα 7 ^{ου} πειράματος

Στη συνέχεια, ακολουθούν τα αποτελέσματα της προσομοίωσης.



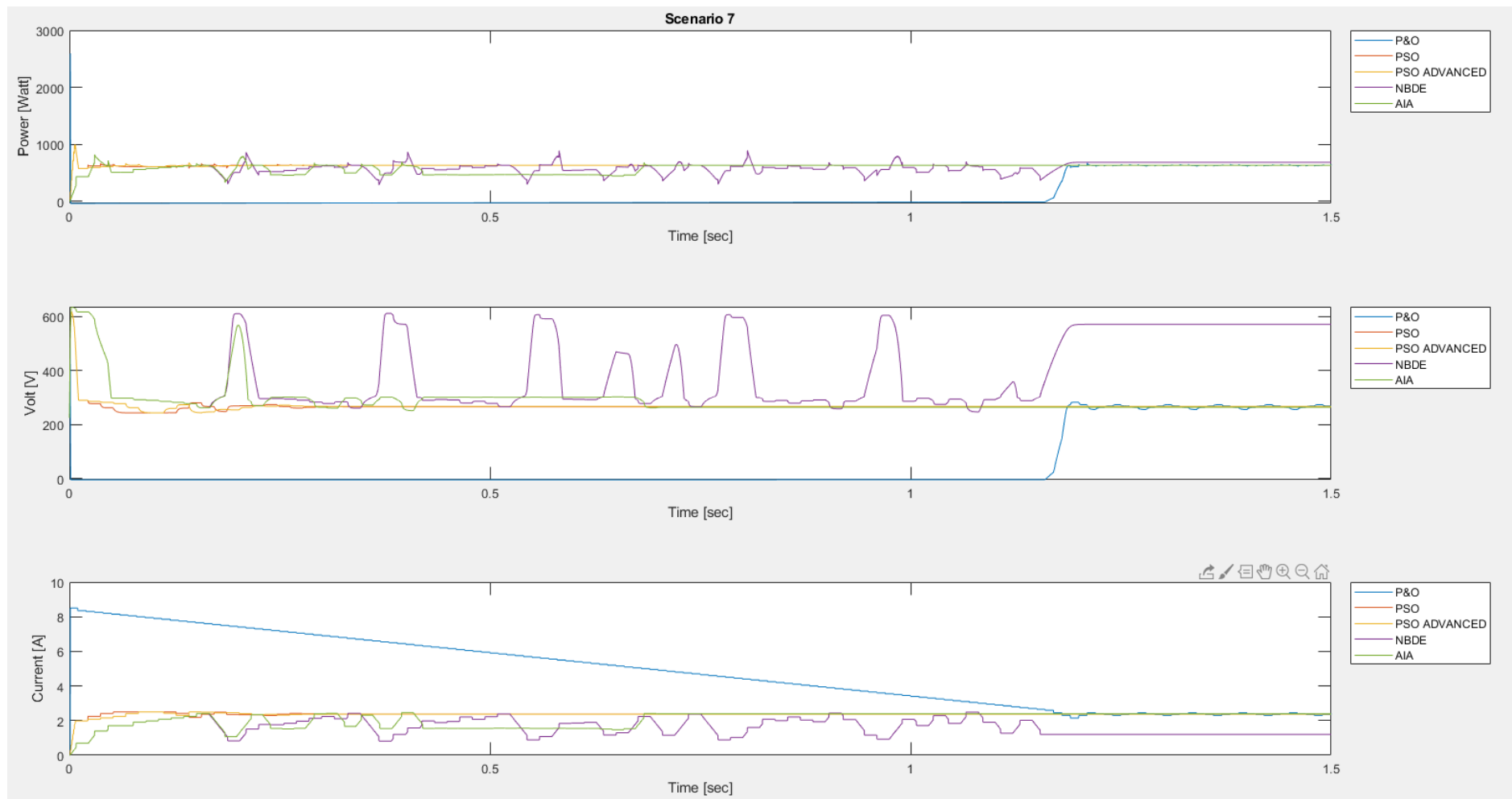
Εικόνα 6.45. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 7

Πίνακας 6.21. Αποτελέσματα προσομοίωσης, πείραμα 7

Αλγόριθμος	P_{max} [W]	V_{max} [V]	I_{max} [A]	Acc [%]	Χρόνος [sec]
-	710.03	560.19	1.28	100.00	0.00
P&O	641.23	271.19	2.45	90.31	1.21
PSO	632.87	266.25	2.38	89.13	0.32
PSO Advanced	632.81	267.40	2.37	89.12	0.31
NBDE	685.00	571.31	1.20	96.48	1.15
AIA	632.56	264.21	2.39	89.09	0.70

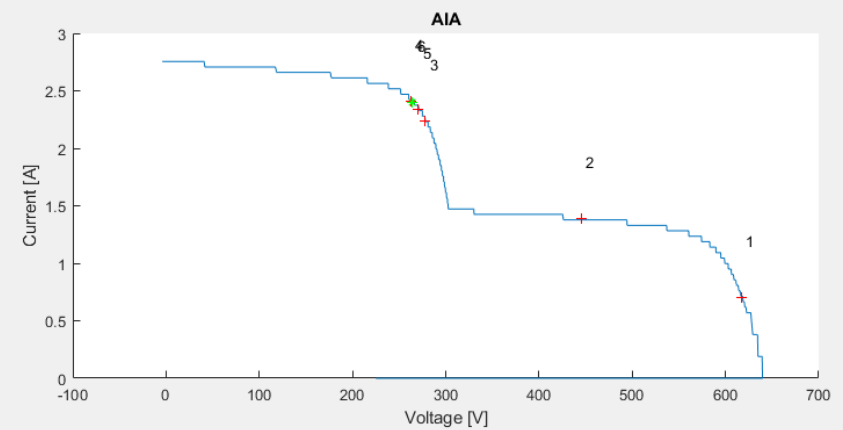
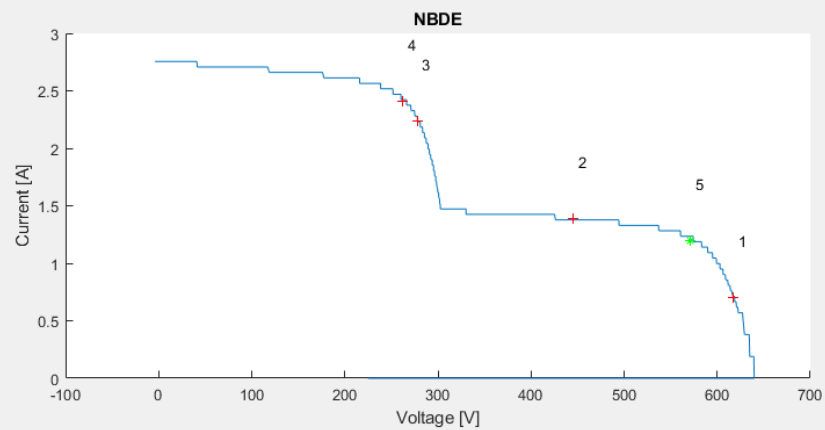
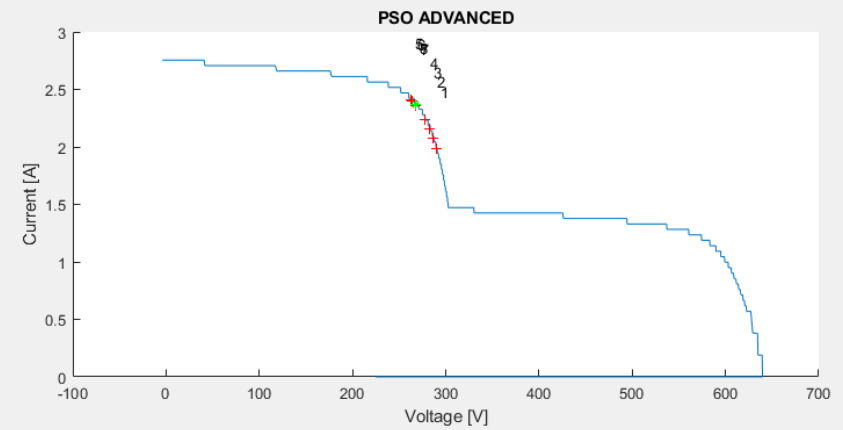
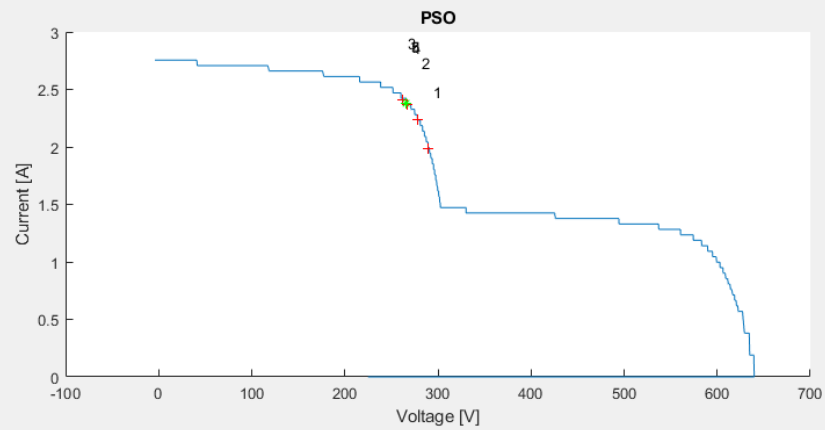
Από τα αποτελέσματα παρατηρείται ότι οι αλγόριθμοι, εκτός του NBDE, αστόχησαν στον εντοπισμό του πραγματικού σημείου MPP. Ένας λόγος που παρατηρείται το φαινόμενο αυτό, είναι το γεγονός ότι το σύστημα λειτουργεί σε πολύ χαμηλή ισχύ και οι αλγόριθμοι δυσκολεύονται να εντοπίσουν το MPP σημείο. Η περιοχή που εντοπίζεται το ολικό μέγιστο είναι μικρή και οι στοχαστικές διεργασίες των αλγορίθμων πρέπει να εντοπίσουν την περιοχή αυτή. Ο NBDE συγκλίνει και στο πείραμα 7 σε περίπου

1 sec, όπως και στα προηγούμενα πειράματα, ενώ ο PSO παρουσιάζεται και σε αυτό το πείραμα ως ο ταχύτερος όλων των επιτυχόντων αλγορίθμων. Το γενικό συμπέρασμα που εξάγεται από το πείραμα 7 είναι ότι απαιτείται περισσότερος χρόνος εκτέλεσης για τους PSO, PSO Advanced και AIA, καθώς επίσης και αλλαγές στις ρυθμίσεις των αλγορίθμων αυτών, ώστε να μπορέσουν να εντοπίσουν με ακρίβεια το ολικό μέγιστο. Ωστόσο, η αλλαγή των παραμέτρων τους εγκυμονεί κινδύνους, διότι υπάρχει περίπτωση να αυξηθεί η απόδοση για το πείραμα 6 και 7, αλλά να μειωθεί στα υπόλοιπα πειράματα. Η τελική ρύθμιση των παραμέτρων των αλγορίθμων προτείνεται να πραγματοποιηθεί σε φυσικό σύστημα. Στο συγκεκριμένο πείραμα ο NBDE εμφανίζεται ως ο αποδοτικότερος όλων των υπολοίπων.



Εικόνα 6.46. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 7

Algorithm steps



Εικόνα 6.47. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 7

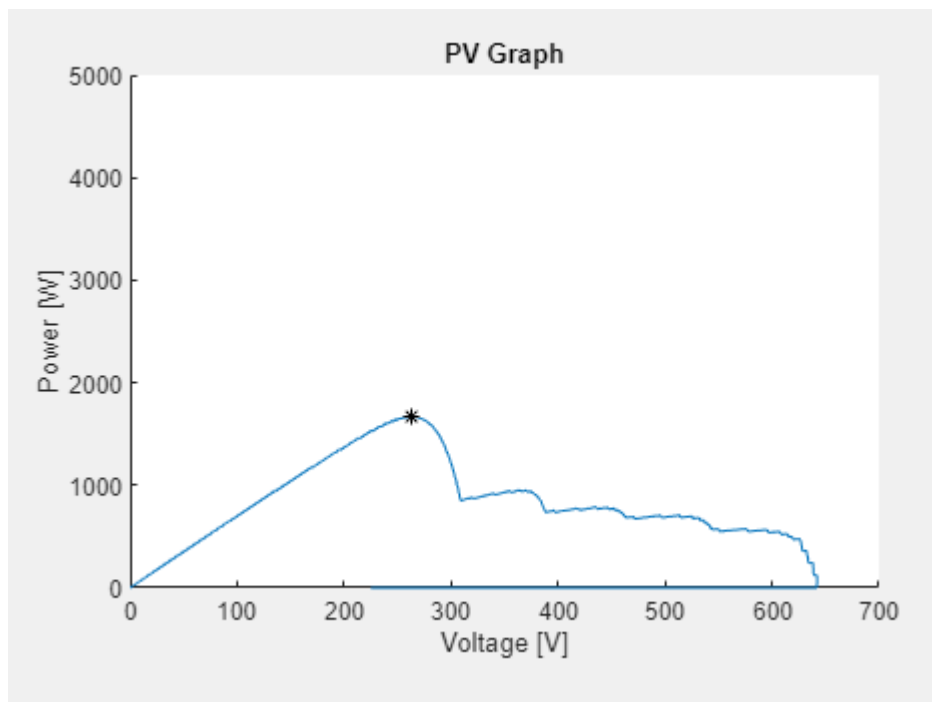
6.10 Πείραμα 8ο

Στο πείραμα 8 οι τιμές για την ακτινοβολία και τη θερμοκρασία των φωτοβολταϊκών μονάδων αυξάνονται. Το ολικό μέγιστο εμφανίζεται στην αριστερή πλευρά της καμπύλης PV. Οι τιμές που χρησιμοποιήθηκαν για την ακτινοβολία και τη θερμοκρασία απεικονίζονται στον πίνακα 6.22.

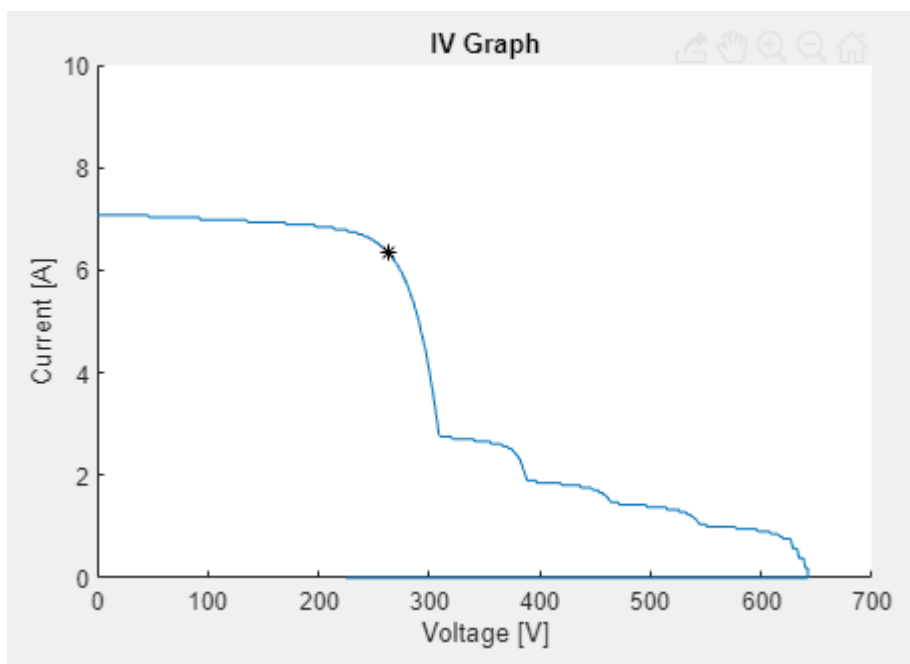
Πίνακας 6.22. Πίνακας τιμών για πειραματικό σενάριο 8

Μεταβλητή	Τιμή	Περιγραφή
Nss_1	8	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	2	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
Nss_3	2	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	2	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	2	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	40° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	27° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	25° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	22° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	20° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	800 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	300 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	200 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	150 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	100 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

Στον πίνακα 6.22 απεικονίζονται οι τιμές για την ακτινοβολία και τη θερμοκρασία για κάθε φωτοβολταϊκή μονάδα. Οι τιμές υποδεικνύουν ένα ολικό μέγιστο. Στη συνέχεια, ακολουθούν τα αποτελέσματα της προσομοίωσης.



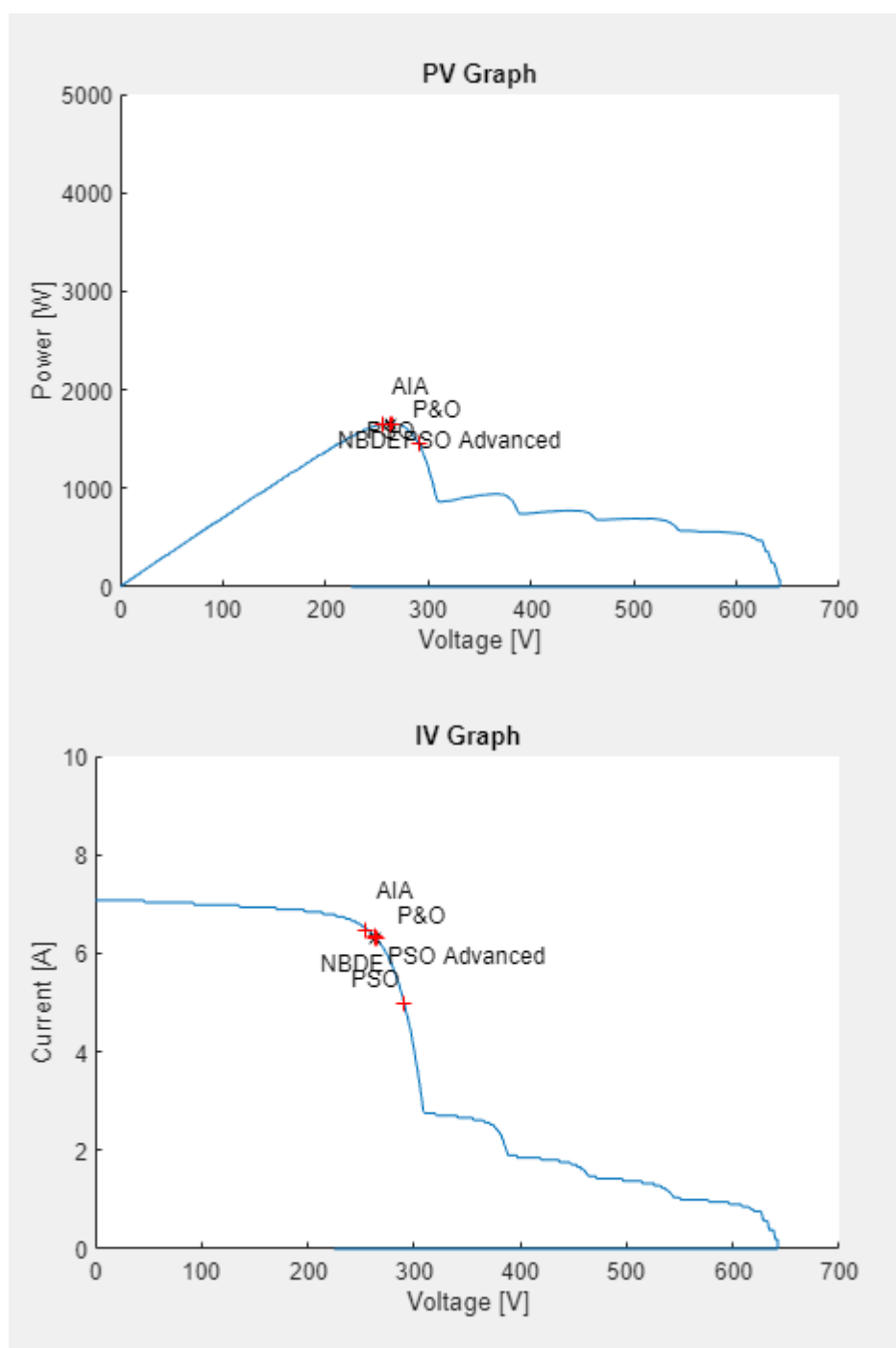
Εικόνα 6.48. Γράφημα PV, πείραμα 8



Εικόνα 6.49. Γράφημα IV, πείραμα 8

Πίνακας 6.23. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 8

Μεταβλητή	Τιμή	Περιγραφή
P_o	1660.16 W	Μέγιστη ισχύς 8 ^{ου} πειράματος
V_o	264.5 V	Μέγιστη τάση 8 ^{ου} πειράματος
I_o	6.32 A	Μέγιστο ρεύμα 8 ^{ου} πειράματος



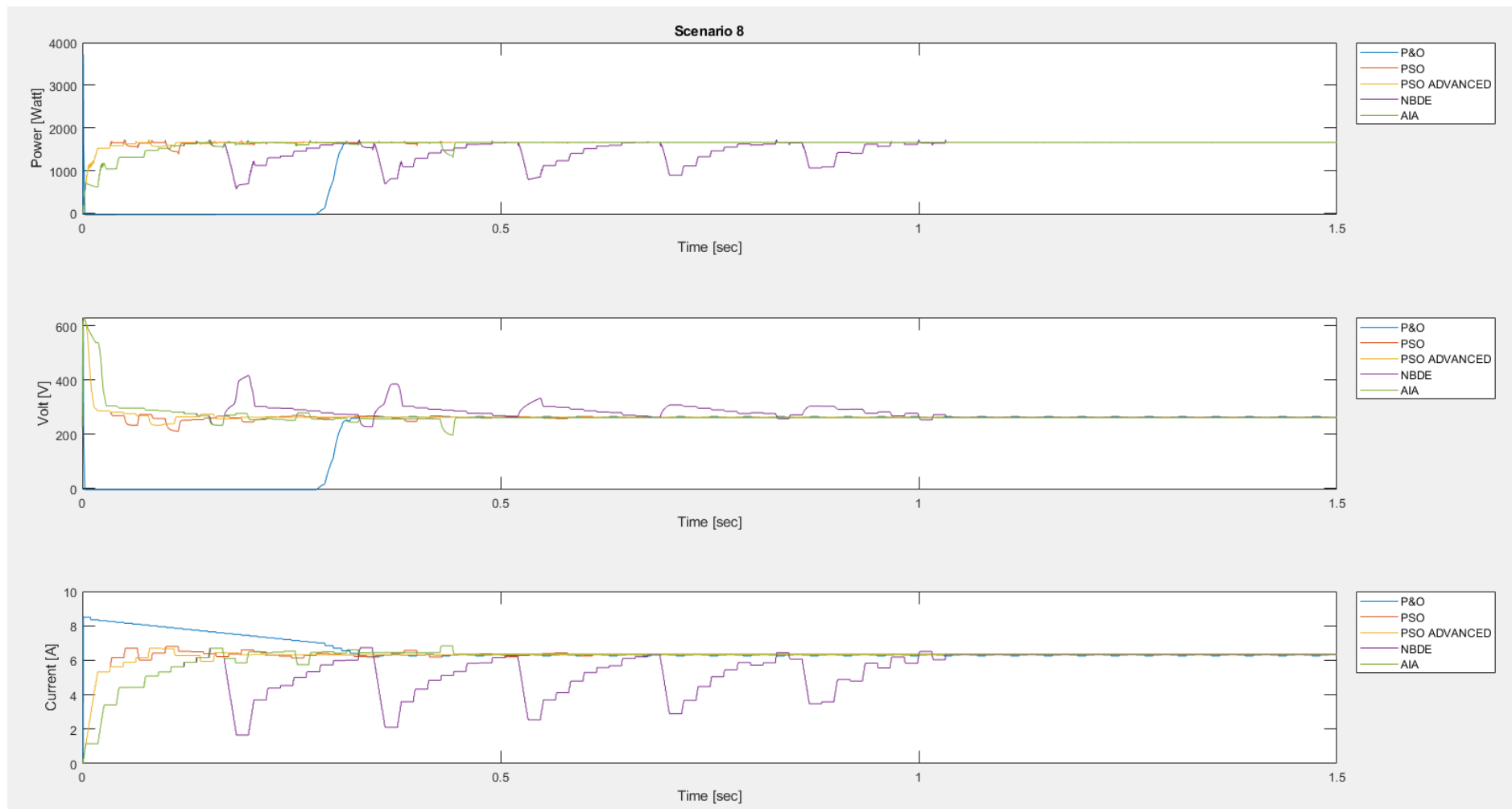
Εικόνα 6.50. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 8

Πίνακας 6.24. Αποτελέσματα προσομοίωσης, πείραμα 8

Αλγόριθμος	P_{max} [W]	V_{max} [V]	I_{max} [A]	Acc [%]	Χρόνος [sec]
-	1660.16	264.47	6.32	100.00	0.00
P&O	1659.96	264.33	6.30	99.99	0.36
PSO	1659.85	263.00	6.31	99.98	0.61
PSO Advanced	1659.85	262.97	6.31	99.98	0.23

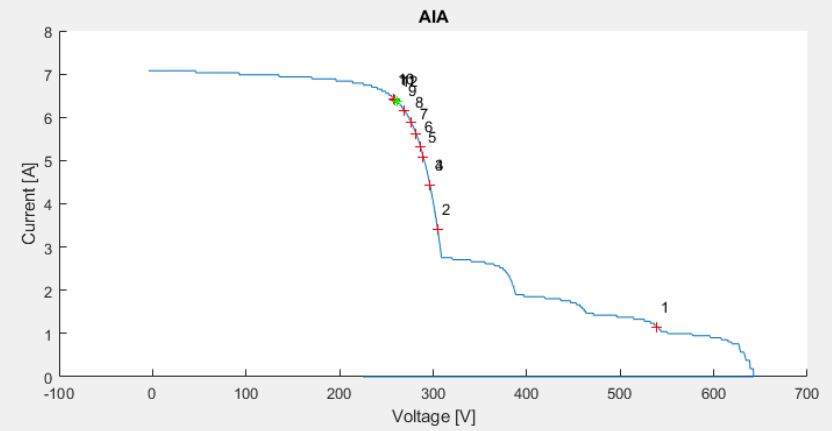
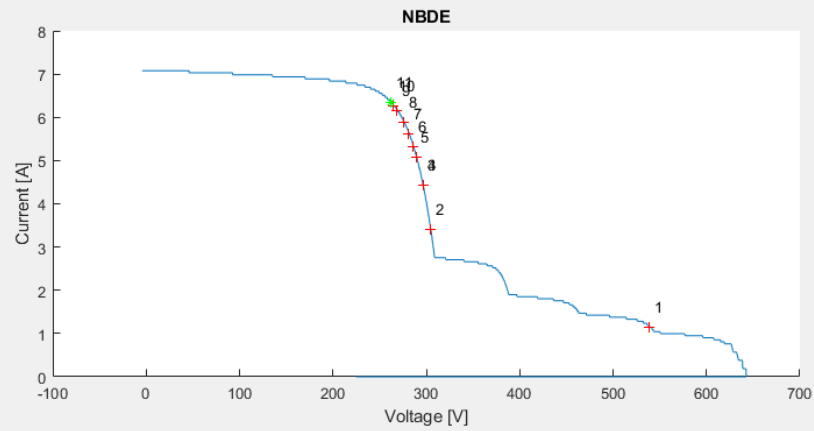
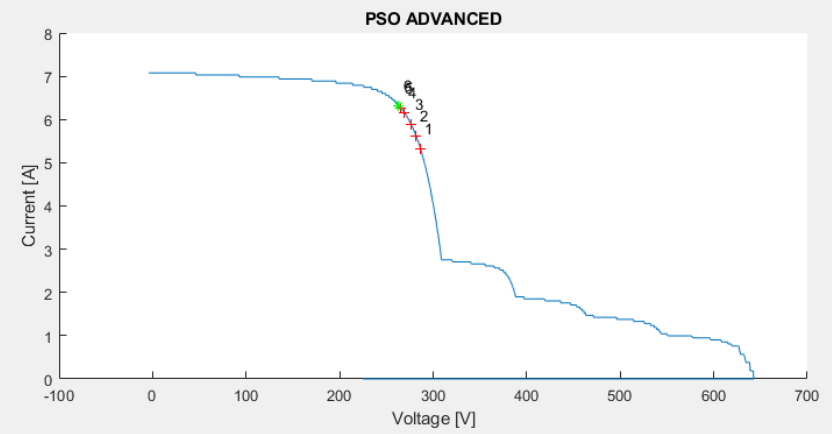
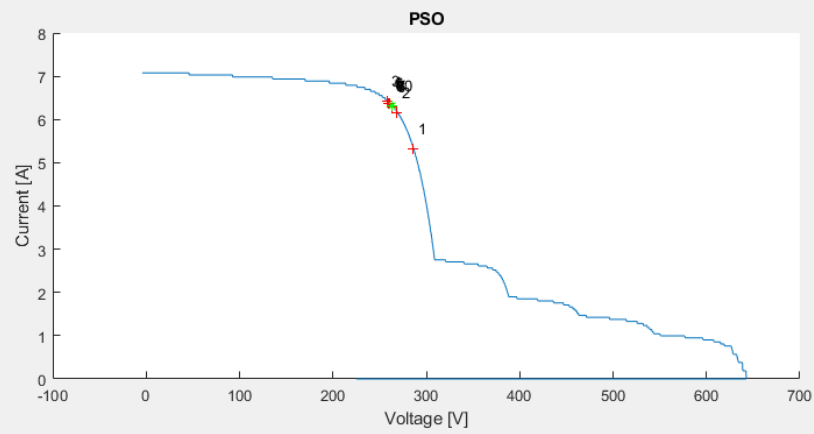
NBDE	1659.74	262.16	6.33	99.97	1.03
AIA	1659.18	260.95	6.36	99.94	0.46

Παρατηρώντας τα παραπάνω αποτελέσματα, εξάγεται το συμπέρασμα ότι όλοι οι αλγόριθμοι εντοπίζουν με επιτυχία το ολικό μέγιστο. Εμφανίζουν παρόμοια απόδοση, με τη διαφορά τους να εντοπίζεται στο χρόνο σύγκλισης. Ο P&O εμφανίζεται ως ο ταχύτερος από όλους, με τη διαφορά ότι η επιτυχία του στον εντοπισμό του ολικού μεγίστου είναι τυχαία. Συνεπώς, ως ο ταχύτερος αλγόριθμος θεωρείται ο PSO advanced. Ο NBDE για μια ακόμα φορά εμφανίζει χρόνο σύγκλισης περίπου 1 sec, λόγω εσωτερικών διεργασιών, ενώ ο PSO εμφανίζει παρόμοιο χρόνο σύγκλισης με τον AIA. Τα αποτελέσματα του συγκεκριμένου πειράματος επαναλαμβάνονται και είναι παρόμοια με τα υπόλοιπα πειραματικά αποτελέσματα. Εξαιρέση αποτελούν τα πειράματα 6 και 7, όπου το σύστημα λειτουργούσε σε πολύ χαμηλή ισχύ (≤ 1 kW).



Εικόνα 6.51. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 8

Algorithm steps



Εικόνα 6.52. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 8

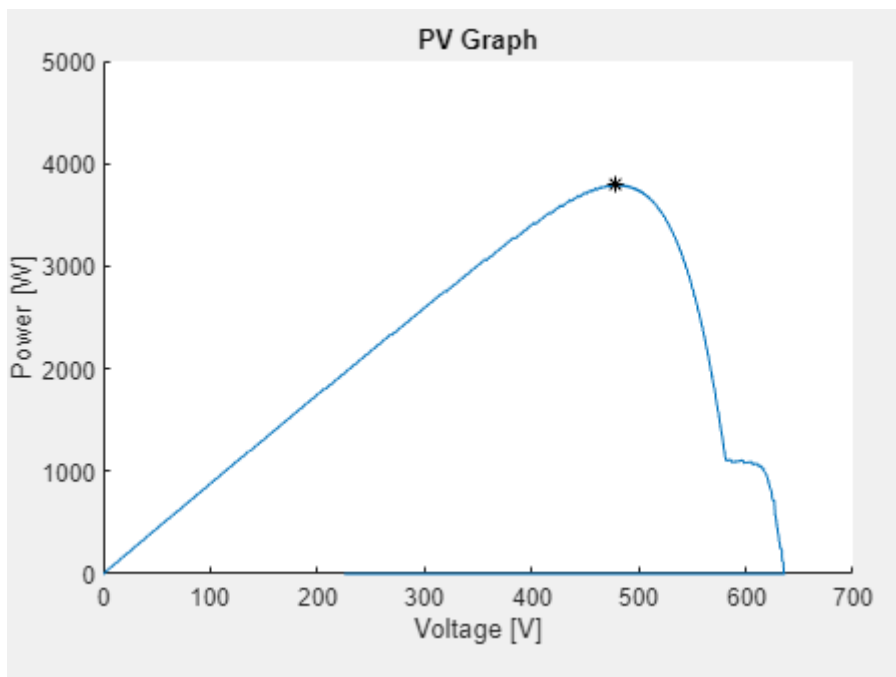
6.11 Πείραμα 9ο

Το τελευταίο πείραμα εμφανίζει παρόμοια χαρακτηριστικά με το πείραμα 1 και συνεπώς αναμένεται να εξαχθούν παρόμοια αποτελέσματα. Μεγάλη ομοιότητα εμφανίζουν επίσης και οι τιμές των παραμέτρων για τις φωτοβολταϊκές μονάδες, με τη διαφορά να εντοπίζεται στην τιμή της θερμοκρασίας και ακτινοβολίας για την 5^η μονάδα. . Οι τιμές που χρησιμοποιήθηκαν για την ακτινοβολία και τη θερμοκρασία απεικονίζονται στον πίνακα 6.25.

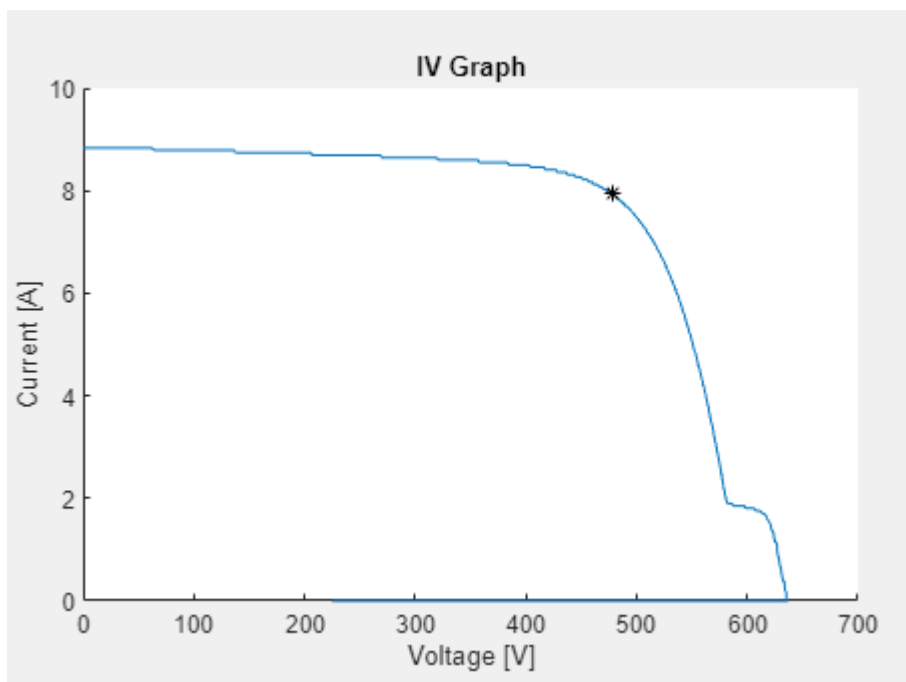
Πίνακας 6.25. Πίνακας τιμών για πειραματικό σενάριο 9

Μεταβλητή	Τιμή	Περιγραφή
Nss_1	12	Αριθμός πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
Nss_2	1	Αριθμός πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
Nss_3	1	Αριθμός πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
Nss_4	1	Αριθμός πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
Nss_5	1	Αριθμός πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
t_pv1	50° C	Θερμοκρασία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
t_pv2	50° C	Θερμοκρασία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
t_pv3	50° C	Θερμοκρασία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
t_pv4	50° C	Θερμοκρασία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
t_pv5	22° C	Θερμοκρασία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)
w_pv1	1000 W/m ²	Ακτινοβολία πάνελ για 1 ^η φωτοβολταϊκή μονάδα (PV1)
w_pv2	1000 W/m ²	Ακτινοβολία πάνελ για 2 ^η φωτοβολταϊκή μονάδα (PV2)
w_pv3	1000 W/m ²	Ακτινοβολία πάνελ για 3 ^η φωτοβολταϊκή μονάδα (PV3)
w_pv4	1000 W/m ²	Ακτινοβολία πάνελ για 4 ^η φωτοβολταϊκή μονάδα (PV4)
w_pv5	200 W/m ²	Ακτινοβολία πάνελ για 5 ^η φωτοβολταϊκή μονάδα (PV5)

Στον πίνακα 6.25 απεικονίζονται οι τιμές για την ακτινοβολία και τη θερμοκρασία για κάθε φωτοβολταϊκή μονάδα. Οι τιμές υποδεικνύουν ένα ολικό μέγιστο. Στη συνέχεια, ακολουθούν τα αποτελέσματα της προσομοίωσης.



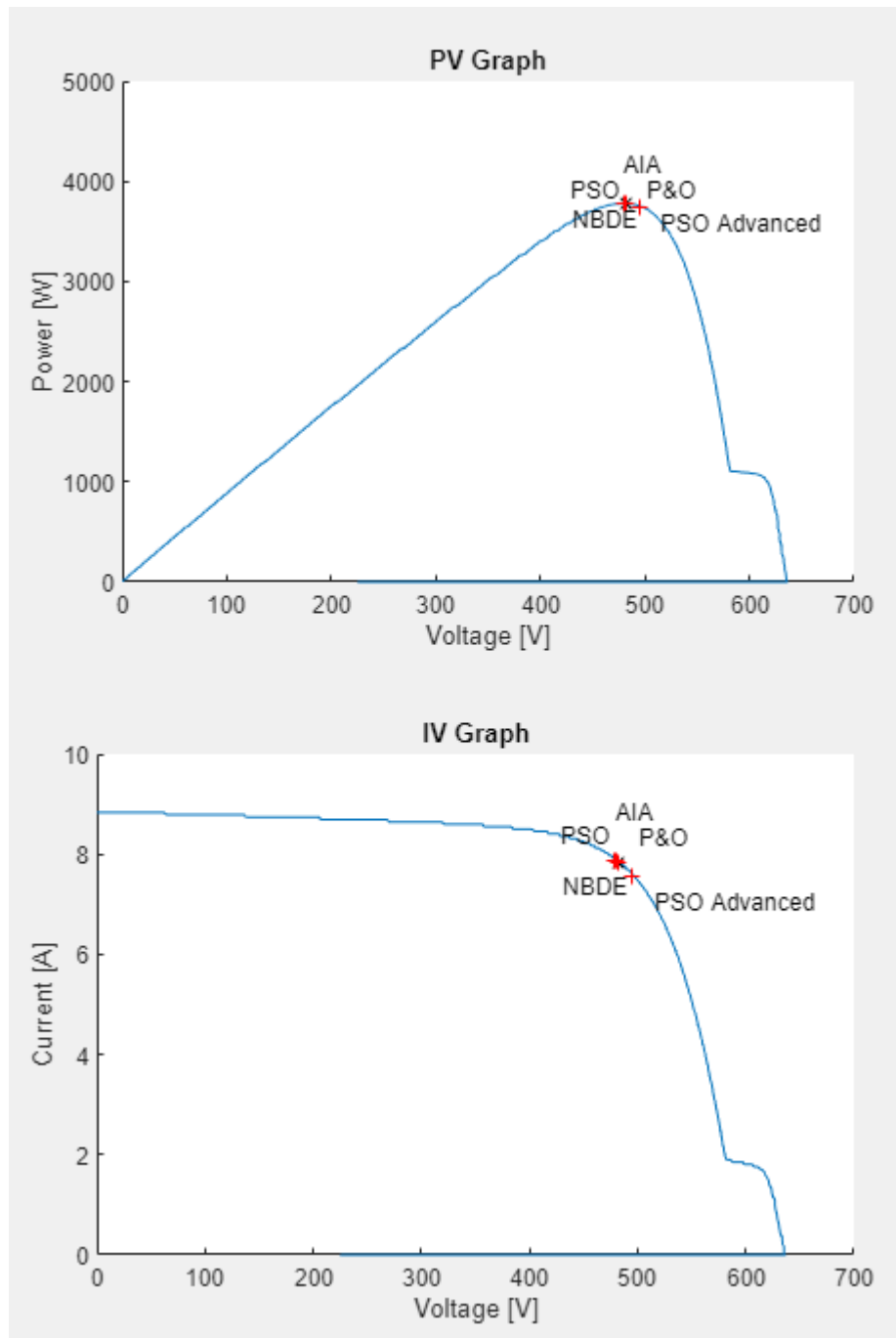
Εικόνα 6.53. Γράφημα PV, πείραμα 9



Εικόνα 6.54. Γράφημα IV, πείραμα 9

Πίνακας 6.26. Μέγιστες τιμές ισχύος, τάσης και ρεύματος, πείραμα 9

Μεταβλητή	Τιμή	Περιγραφή
P_o	3777 W	Μέγιστη ισχύς 9 ^{ου} πειράματος
V_o	481.7 V	Μέγιστη τάση 9 ^{ου} πειράματος
I_o	7.89 A	Μέγιστο ρεύμα 9 ^{ου} πειράματος

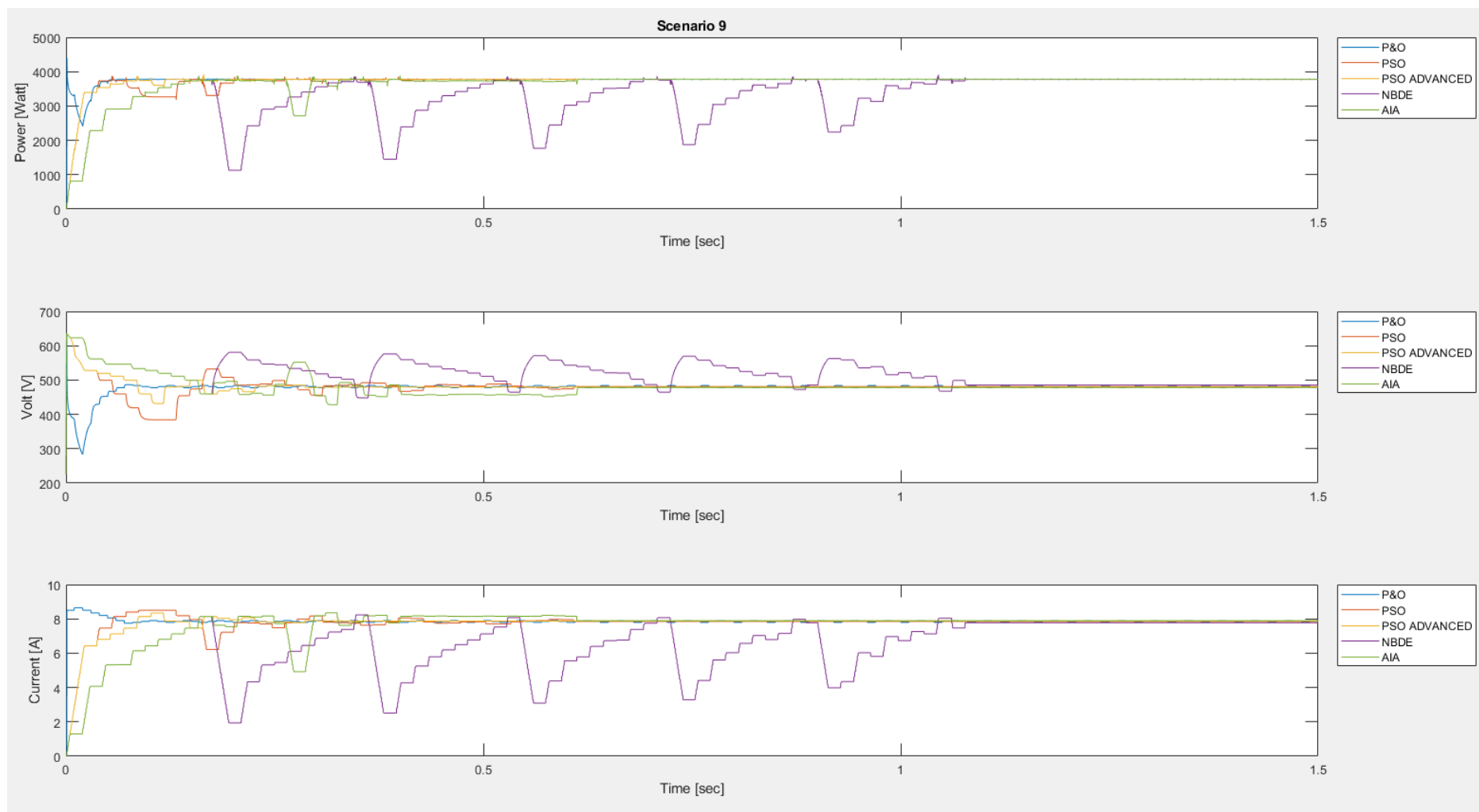


Εικόνα 6.55. Καμπύλη PV και IV, αποτέλεσμα προσομοίωσης για πείραμα 9

Πίνακας 6.27. Αποτελέσματα προσομοίωσης, πείραμα 9

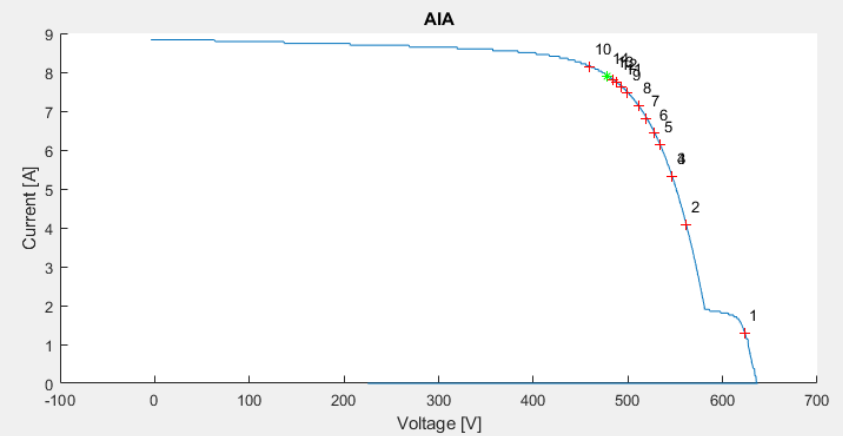
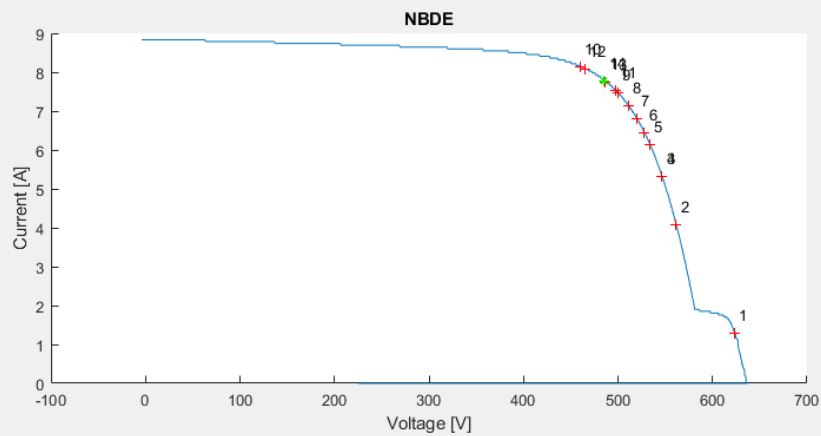
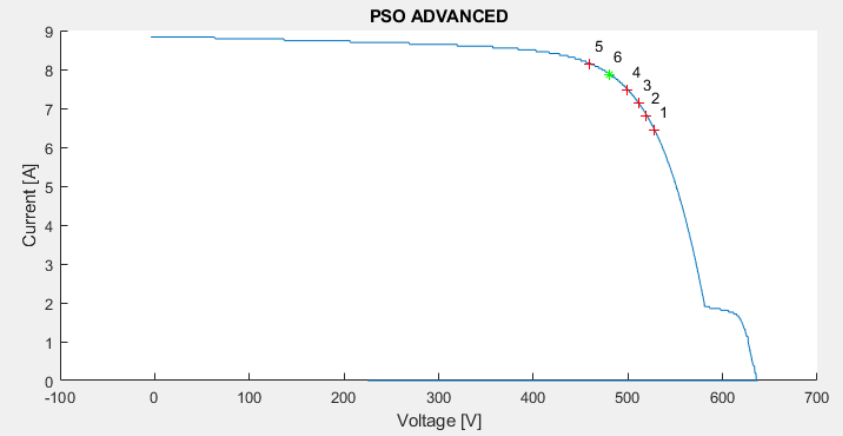
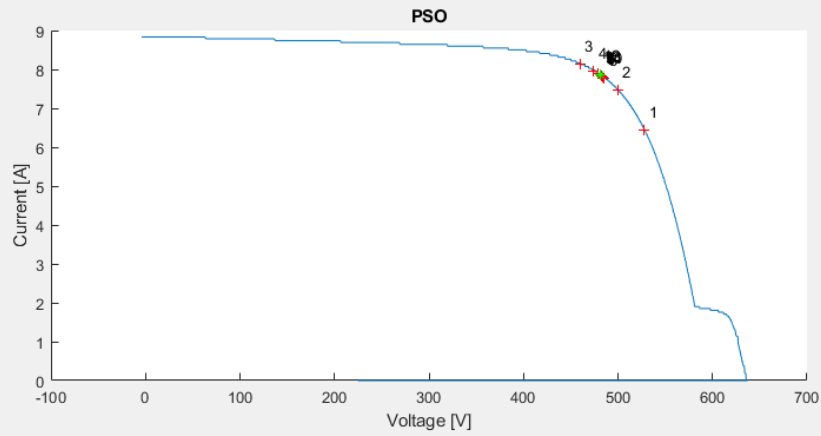
Αλγόριθμος	P_{max} [W]	V_{max} [V]	I_{max} [A]	Acc [%]	Χρόνος [sec]
-	3777.20	481.76	7.89	100.00	0.00
P&O	3777.10	482.65	7.85	100.00	0.09
PSO	3777.02	481.00	7.85	100.00	0.62
PSO Advanced	3777.09	479.86	7.87	100.00	0.33
NBDE	3773.96	485.40	7.78	99.91	1.08
AIA	3776.79	478.47	7.89	99.99	0.63

Παρατηρώντας τα παραπάνω αποτελέσματα, εξάγεται το συμπέρασμα ότι όλοι οι αλγόριθμοι εντοπίζουν με επιτυχία το ολικό μέγιστο. Εμφανίζουν παρόμοια απόδοση, με τη διαφορά τους να εντοπίζεται στο χρόνο σύγκλισης. Ο P&O εμφανίζεται ως ο ταχύτερος από όλους, με τη διαφορά ότι η επιτυχία του στον εντοπισμό του ολικού μεγίστου είναι τυχαία. Συνεπώς, ως ο ταχύτερος αλγόριθμος θεωρείται ο PSO advanced. Ο NBDE για μια ακόμα φορά εμφανίζει χρόνο σύγκλισης περίπου 1 sec, λόγω εσωτερικών διεργασιών, ενώ ο PSO εμφανίζει παρόμοιο χρόνο σύγκλισης με τον AIA. Τα αποτελέσματα του συγκεκριμένου πειράματος επαναλαμβάνονται και είναι παρόμοια με τα υπόλοιπα πειραματικά αποτελέσματα. Εξαιρέση αποτελούν τα πειράματα 6 και 7, όπου το σύστημα λειτουργούσε σε πολύ χαμηλή ισχύ (≤ 1 kW).



Εικόνα 6.56. Καμπύλες ισχύος (P), τάσης (V) και ρεύματος (I) κατά τη διάρκεια προσομοίωσης, πείραμα 9

Algorithm steps



Εικόνα 6.57. Βήματα αλγορίθμων στη χαρακτηριστική IV, πείραμα 9

Κεφάλαιο 7ο: Αποτελέσματα - Συμπεράσματα

Για την αξιολόγηση των αποτελεσμάτων θα χρησιμοποιηθούν οι μετρικές της απόδοσης η , του χρόνου σύγκλισης t με τις αντίστοιχες διακυμάνσεις τους. Επίσης, σημαντικό ρόλο στην ανάδειξη της βέλτιστης τεχνικής διαδραματίζει η πολυπλοκότητα των αλγορίθμων. Εντούτοις, στη συγκεκριμένη εργασία δεν εξετάζεται, ούτε λαμβάνεται υπόψη κατά την αξιολόγηση των αλγορίθμων. Η πολυπλοκότητα θα ληφθεί υπόψη κατά την εφαρμογή της βέλτιστης μεθόδου σε πραγματικό σύστημα. Σε επίπεδο προσομοίωσης ο χρόνος εκτέλεσης των αλγορίθμων είναι ενδεικτικός της πολυπλοκότητας και είναι επαρκής για τη σύγκριση των αποτελεσμάτων.

Ως ακρίβεια απόδοσης ορίζεται το ποσοστό της ισχύος εξόδου του συστήματος προς τη θεωρητική μέγιστη ισχύ. Η ακρίβεια απόδοσης των αλγορίθμων υπολογίζεται από την σχέση:

$$\eta = \frac{P(t)}{P_{MPPT}(t)} \times 100\% \quad (6.1)$$

όπου $P(t)$ η πραγματική ισχύς του συστήματος και $P_{MPPT}(t)$ η θεωρητική μέγιστη ισχύς του κυκλώματος. Η θεωρητική μέγιστη ισχύς υπολογίζεται, αν αντί για κάποιον αλγόριθμο γίνει εισαγωγή μιας συνάρτησης μοναδιαίας κλίσης (ράμπα) με κατάλληλο βήμα εκτέλεσης. Αντίστοιχα, η μέση τιμή της ισχύος, υπολογίζεται ως εξής:

$$\bar{\eta} = \frac{1}{n} \sum_{i=1}^n P_i \quad (6.2)$$

Ως ταχύτητα σύγκλισης θεωρείται ο χρόνος εκτέλεσης – σύγκλισης του αλγορίθμου στο σημείο MPP. Η συγκεκριμένη τιμή παρέχεται από το περιβάλλον της Simulink. Η μέση τιμή για το χρόνο εκτέλεσης υπολογίζεται ως:

$$\bar{t} = \frac{1}{n} \sum_{i=1}^n t_i \quad (6.3)$$

όπου n ο αριθμός του συνόλου των τιμών για την ισχύ και το χρόνο σύγκλισης αντίστοιχα. Τέλος, η τυπική απόκλιση της απόδοσης και του χρόνου εκτέλεσης, δίνονται από τους παρακάτω τύπους:

$$\sigma_{\eta} = \sqrt{\frac{\sum_{i=1}^n (\eta_i - \bar{\eta})^2}{n}} \quad (6.4)$$

$$\sigma_t = \sqrt{\frac{\sum_{i=1}^n (t_i - \bar{t})^2}{n}} \quad (6.5)$$

Ως ποσοστό επιτυχούς σύγκλισης SR ορίζεται ο λόγος μεταξύ των επιτυχημένων εκτελέσεων των αλγορίθμων – εκτελέσεις που εντόπισαν με επιτυχία το σημείο MPP – ως προς το συνολικό αριθμό εκτελέσεων. Η μετρική αυτή είναι απαραίτητη, διότι λόγω στοχαστικών διεργασιών που υλοποιούνται στο εσωτερικό των αλγορίθμων, υπάρχει πιθανότητα κάποιος από τους αλγορίθμους να συγκλίνει σε διαφορετικά σημεία κάθε φορά που εκτελείται. Συνεπώς, η συγκεκριμένη μετρική μετρά την ευρωστία

των αλγορίθμων και είναι καθοριστική για ένα φωτοβολταϊκό σύστημα. Για τη συγκεκριμένη μετρική θα χρησιμοποιηθεί η σχέση:

$$SR = \frac{\text{επιτυχημένες εκτελέσεις}}{\text{συνολικές εκτελέσεις}} \times 100\% \quad (6.2)$$

Ο όρος επιτυχημένες εκτελέσεις περιγράφουν τις εκτελέσεις που απέχουν μέχρι 5 % από το μέγιστο σημείο που εντοπίζει ο εκάστοτε αλγόριθμος. Το ποσοστό των 5 % ορίστηκε αυθαίρετα από το συγγραφέα της διπλωματικής και αποτελεί ένα όριο το οποίο δύναται να αλλάξει. Είναι μια μετρική που καθορίζεται αποκλειστικά από το χρήστη και τις ανάγκες του.

7.1 Απόδοση

Ως προς την απόδοση των αλγορίθμων, παρακάτω παρουσιάζεται ο πίνακας με τα αποτελέσματα για κάθε πείραμα και αλγόριθμο. Από τον πίνακα 7.1 παρατηρείται ότι ο αλγόριθμος P&O παρουσιάζει την μικρότερη απόδοση. Το ποσοστό της απόδοσης του συγκεκριμένου αλγορίθμου καθορίζεται από τη θέση του ολικού μεγίστου στην καμπύλη της ισχύος PV και από την αρχική θέση αναζήτησης. Συνεπώς, το ποσοστό της απόδοσης που απεικονίζεται στον πίνακα 7.1 – ως αριθμητική τιμή – είναι καθαρά ενδεικτικό και το μόνο ασφαλές συμπέρασμα που εξάγεται από τα πειράματα της εργασίας αυτής, είναι ότι ο αλγόριθμος P&O, παρουσιάζει τη μικρότερη απόδοση σε σχέση με τους υπολοίπους.

Αναφορικά με τον αλγόριθμο PSO, παρατηρείται υψηλή απόδοση σε όλα τα πειράματα. Η υψηλή απόδοση του PSO ήταν αναμενόμενη και επιβεβαιώνει τη μεγάλη δημοτικότητα που έχει ανάμεσα στους αλγορίθμους για τις εφαρμογές MPPT. Επίσης, παρατηρείται ότι σε περιπτώσεις που η ακτινοβολία του συστήματος είναι χαμηλή, σενάριο 6 και 7, ο PSO παρουσιάζει πτώση στην απόδοσή του, κυρίως λόγω του μικρού πληθυσμού του. Επίσης, στα πειράματα 6,7 και γενικότερα σε καταστάσεις με χαμηλή ακτινοβολία – το οποίο μεταφράζεται σε περιορισμένο χώρο λύσεων - ο εντοπισμός του σημείου MPP γίνεται δυσκολότερος και συνεπώς πολλοί αλγόριθμοι καταλήγουν σε κάποιο τοπικό μέγιστο.

Ο PSO Advanced, παρουσιάζει την ίδια εικόνα με τον PSO αναφορικά με την απόδοση του. Οι αστοχίες τους εντοπίζονται στα ίδια σενάρια μερικής σκίασης 6 και 7. Ο εσωτερικός μηχανισμός του PSO Advanced είναι ίδιος με αυτόν του PSO, με τη διαφορά τους να εντοπίζεται στην ανανέωση των θέσεων του πληθυσμού τους. Ο διαφορετικός τρόπος ανανέωσης των θέσεων του πληθυσμού επηρεάζει την ταχύτητα σύγκλισης του αλγορίθμου, για την οποία θα γίνει λόγος στην επόμενη ενότητα.

Ο NBDE παρουσιάζει τη μέγιστη απόδοση μαζί με τον AIA. Ουσιαστικά, οι δύο αυτοί αλγόριθμοι εντόπισαν με επιτυχία τα MPP των σεναρίων 1 έως 9. Οι δύο αυτοί αλγόριθμοι είναι διαφορετικοί ως προς τον τρόπο ανανέωσης του πληθυσμού, αλλά και οι δύο εντοπίζουν με επιτυχία το βέλτιστο σημείο. Η διαφορά τους εντοπίζεται στο χρόνο εκτέλεσης και θα γίνει σχολιασμός στην επόμενη ενότητα.

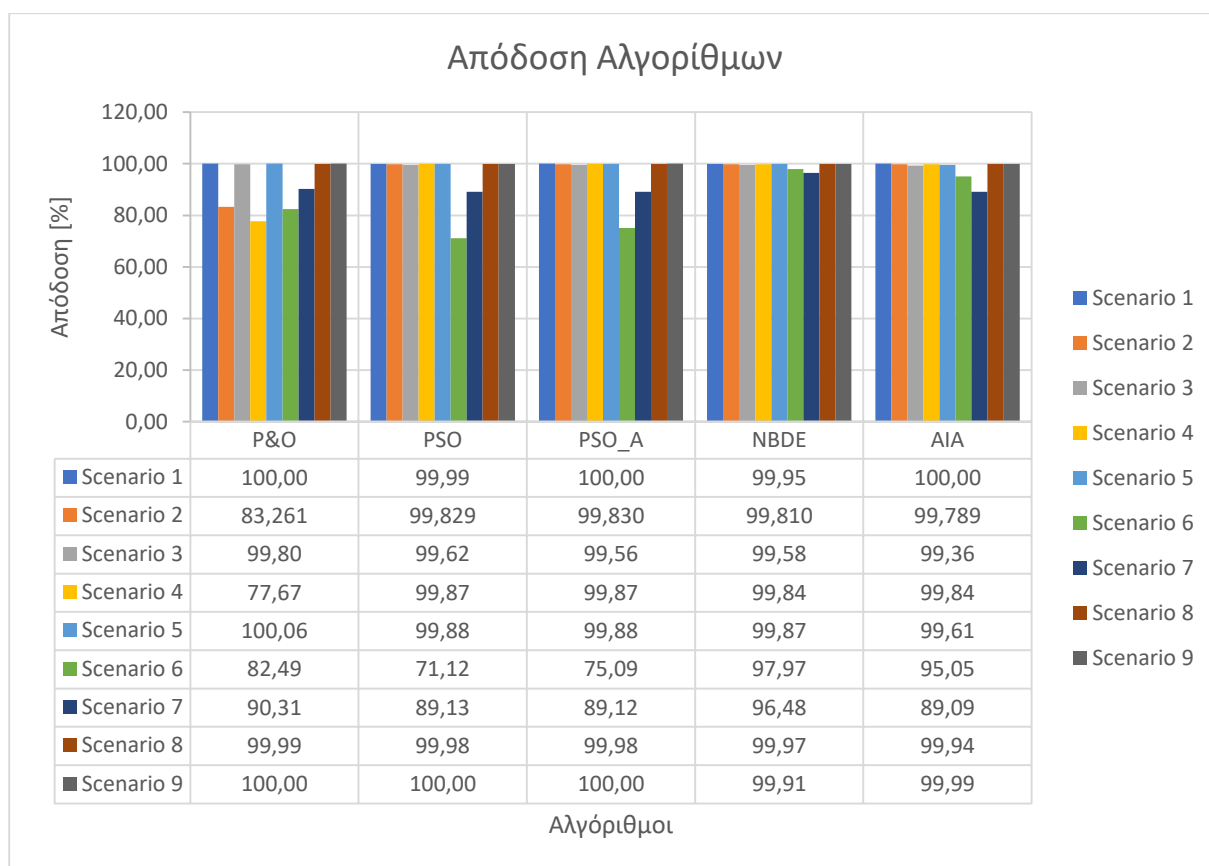
Πίνακας 7.1. Μέση τιμή απόδοσης [η] αλγορίθμων, μονάδες [%]

Αλγόριθμος	Πείραμα 1	Πείραμα 2	Πείραμα 3	Πείραμα 4	Πείραμα 5	Πείραμα 6	Πείραμα 7	Πείραμα 8	Πείραμα 9
P&O	100.00	83.26	99.80	77.67	100.06	82.49	90.31	99.99	100.00
PSO	99.99	99.83	99.62	99.87	99.88	71.12	89.13	99.98	100.00
PSO_A	100.00	99.83	99.56	99.87	99.88	75.09	89.12	99.98	100.00

NBDE	99.95	99.76	99.58	99.85	99.87	97.97	96.47	99.97	99.91
AIA	99.99	99.80	99.31	99.85	99.75	95.89	89.06	99.97	99.92

Πίνακας 7.2. Τυπική απόκλιση [σ] αλγορίθμων για τιμή απόδοσης, 10 επαναλήψεις

Αλγόριθμος	Πείραμα 1	Πείραμα 2	Πείραμα 3	Πείραμα 4	Πείραμα 5	Πείραμα 6	Πείραμα 7	Πείραμα 8	Πείραμα 9
P&O	0	0	0	0	0	0	0	0	0
PSO	0	0	0	0	0	0	0	0	0
PSO_A	0	0	0	0	0	0	0	0	0
NBDE	0	2	0	0	0	0	0	0	0
AIA	1	1	14	0	2	24	0	0	6



Εικόνα 7.1. Σύγκριση αλγορίθμων (απόδοση)

7.2 Ταχύτητα σύγκλισης

Η ταχύτητα σύγκλισης είναι εξίσου σημαντική για την επιλογή της κατάλληλης μεθόδου εύρεσης του σημείου MPP. Είναι επιθυμητό σε ένα σύστημα, του οποίου η απόδοση εξαρτάται από τις περιβαλλοντικές συνθήκες, να αποκρίνεται και να εντοπίζει άμεσα το νέο σημείο λειτουργίας, έτσι ώστε να είναι ικανό να ανταπεξέλθει σε ταχείες αλλαγές των συνθηκών αυτών. Φυσικά, σε εφαρμογές που

δεν απαιτείται άμεση απόκριση του συστήματος, η επιλογή της κατάλληλης μεθόδου είναι στη διάθεση του χρήστη. Από τα δεδομένα των πειραματισμών προκύπτει το συμπέρασμα, ότι το μικρότερο χρόνο σύγκλισης τον παρουσιάζει ο αλγόριθμος PSO Advanced. Ο συγκεκριμένος αλγόριθμος, συγκλίνει σε χρόνο μικρότερο των 0.5 sec για όλα τα πειράματα. Η ταχύτατη σύγκλιση του PSO Advanced είναι υπεύθυνη για το γεγονός ότι σε κάποια πειράματα ο αλγόριθμος εγκλωβίζεται σε τοπικά μέγιστα. Συνεπώς, η απόδοση και η ταχύτητα σύγκλισης είναι δύο όροι που είναι αλληλοεξαρτώμενοι.

Ο αλγόριθμος NBDE παρουσιάζει με τη σειρά του τη μεγαλύτερη απόδοση, αλλά έχει τη μεγαλύτερη τιμή για το χρόνο σύγκλισης. Είναι ο αλγόριθμος που συγκλίνει πιο αργά από όλους και για αυτό το λόγο εκτελείται περισσότερες φορές και εμφανίζει καλύτερα αποτελέσματα από τους υπολοίπους. Ο συγκεκριμένος αλγόριθμος εκτελείται για συγκεκριμένο αριθμό επαναλήψεων και για αυτό το λόγο εμφανίζει σταθερό χρόνο κατά την εκτέλεση του. Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, ο NBDE έχει διαφορετικές εσωτερικές διεργασίες από τους υπολοίπους και αφού ολοκληρώσει το σύνολο των ρυθμισμένων επαναλήψεων, εξάγει το αποτέλεσμα.

Ο AIA και ο PSO εμφανίζουν τον ίδιο χρόνο σύγκλισης, γεγονός που υποδεικνύει την ομοιότητα των δύο αυτών αλγορίθμων ως προς τις εσωτερικές τους διεργασίες. Η διαφορά τους εντοπίζεται στην απόδοση του, με τον AIA να εμφανίζει 3% περίπου μεγαλύτερη ακρίβεια στα πειράματα της εργασίας αυτής.

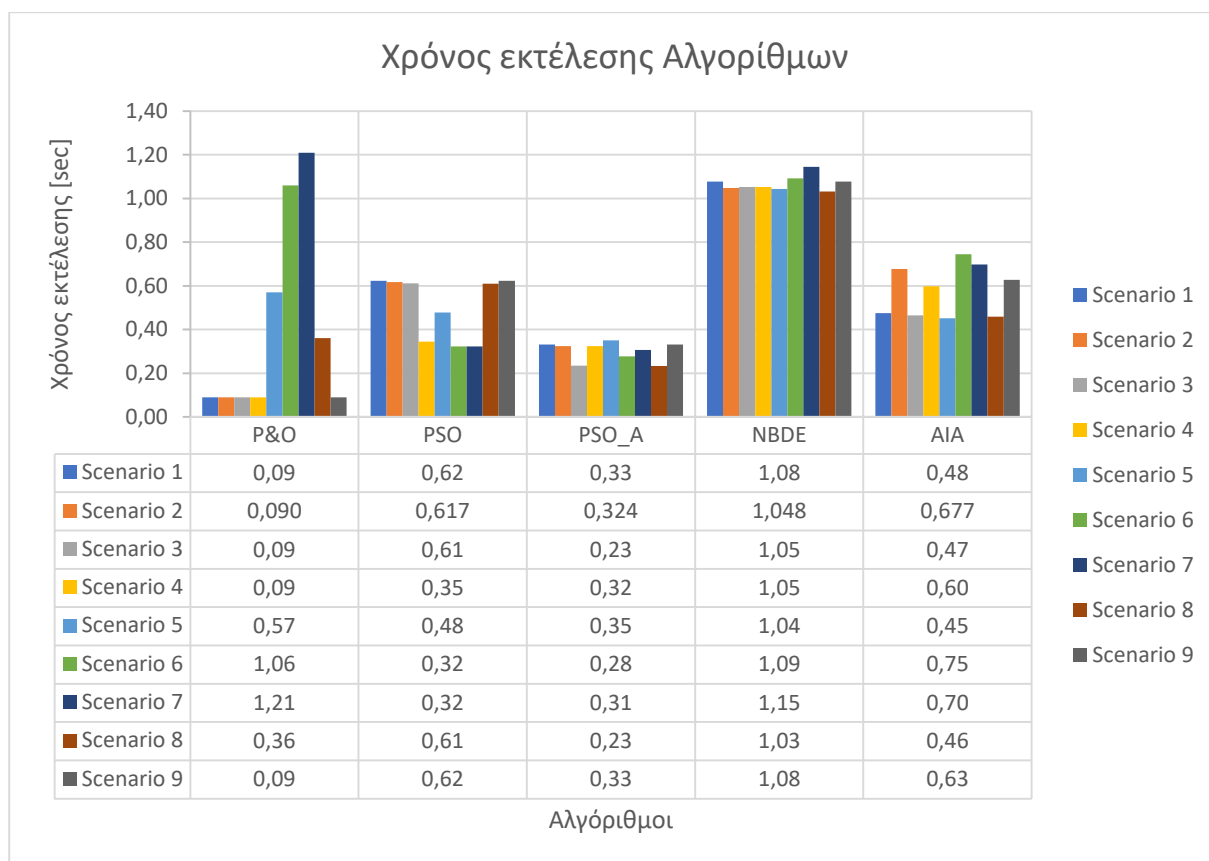
Ο P&O εμφανίζει μέσο χρόνο απόκρισης παρόμοιο με αυτόν των AIA και PSO. Από την εικόνα 7.2 και από τον πίνακα 7.1, προκύπτει το συμπέρασμα ότι ανάλογα με την αρχική θέση εκκίνησης του P&O ως προς το σημείο MPP, θα είναι και η ταχύτητα σύγκλισης του.

Πίνακας 7.3. Μέση τιμή χρόνου σύγκλισης [t] αλγορίθμων, μονάδες [sec]

Αλγόριθμος	Πείραμα 1	Πείραμα 2	Πείραμα 3	Πείραμα 4	Πείραμα 5	Πείραμα 6	Πείραμα 7	Πείραμα 8	Πείραμα 9
P&O	0.09	0.09	0.09	0.09	0.57	1.06	1.21	0.36	0.09
PSO	0.62	0.62	0.61	0.35	0.48	0.32	0.32	0.61	0.62
PSO_A	0.33	0.32	0.23	0.32	0.35	0.28	0.31	0.23	0.33
NBDE	1.08	1.05	1.05	1.05	1.04	1.08	1.15	1.03	1.08
AIA	0.43	0.53	0.47	0.52	0.46	0.68	0.73	0.43	0.37

Πίνακας 7.4. Τυπική απόκλιση [σ] αλγορίθμων για χρόνο σύγκλισης, 10 επαναλήψεις

Αλγόριθμος	Πείραμα 1	Πείραμα 2	Πείραμα 3	Πείραμα 4	Πείραμα 5	Πείραμα 6	Πείραμα 7	Πείραμα 8	Πείραμα 9
P&O	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PSO	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PSO_A	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NBDE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
AIA	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1



Εικόνα 7.2. Χρόνοι εκτέλεσης αλγορίθμων

7.3 Ποσοστό επιτυχούς σύγκλισης

Όπως αναφέρθηκε παραπάνω, η μετρική αυτή είναι απαραίτητη, διότι λόγω στοχαστικών διεργασιών που υλοποιούνται στο εσωτερικό των αλγορίθμων, υπάρχει πιθανότητα κάποιος από τους αλγορίθμους να συγκλίνει σε διαφορετικό σημείο κάθε φορά που εκτελείται. Συνεπώς, η συγκεκριμένη μετρική μετρά την ευρωστία των αλγορίθμων και είναι καθοριστική για ένα φωτοβολταϊκό σύστημα. Για τις ανάγκες τις εργασίας, κάθε πειραματισμός εκτελέστηκε δέκα φορές και εξάχθηκε η μέση τιμή και η διακύμανση της απόδοσης και της ταχύτητας σύγκλισης των αλγορίθμων. Στον παρακάτω πίνακα παρουσιάζεται το ποσοστό επιτυχούς σύγκλισης για κάθε έναν αλγόριθμο.

Πίνακας 7.5. Ποσοστό επιτυχών συγκλίσεων για 10 επαναλήψεις

Αλγόριθμοι	SR [%]
P&O	100
PSO	100
PSO Advanced	100
NBDE	100
AIA	100

Από τα αποτελέσματα, προέκυψε το συμπέρασμα ότι όλοι οι αλγόριθμοι καταλήγουν στο ίδιο αποτέλεσμα και για τις δέκα επαναλήψεις. Συνεπώς, όλοι οι αλγόριθμοι δύνανται να τοποθετηθούν σε ένα πραγματικό σύστημα και για τη συγκεκριμένη μετρική όλοι οι αλγόριθμοι παρουσιάζουν την ίδια απόδοση.

7.4 Σχόλια

Έπειτα από ανάλυση και αξιολόγηση των μεθόδων που χρησιμοποιούνται για τον εντοπισμό του σημείου MPP, υλοποιήθηκαν πέντε αλγόριθμοι με σκοπό να αξιολογηθούν αρχικά θεωρητικά και στη συνέχεια να επιλεγεί η βέλτιστη μέθοδος, ώστε να υλοποιηθεί και να αξιολογηθεί η απόδοση της σε πραγματικές συνθήκες. Στα πλαίσια, λοιπόν, της παρούσης εργασίας, υλοποιήθηκαν οι αλγόριθμοι P&O, PSO, PSO Advanced, NBDE και AIA.

Ο P&O θεωρείται ένας από τους πιο κλασικούς αλγορίθμους, που χρησιμοποιούνται για αυτές τις περιπτώσεις. Εντούτοις, εμφανίζει προβλήματα, όπως ταλαντώσεις σε σταθερή κατάσταση, εμφανίζει αποκλίσεις από το σημείο MPP και αδυνατεί να εντοπίσει με επιτυχία το συγκεκριμένο σημείο. Οι αδυναμίες του αλγορίθμου P&O είναι εμφανείς και στα πειράματα της παρούσης διπλωματικής. Συνεπώς, αποτελεί τον αλγόριθμο που εμφανίζει τη μικρότερη απόδοση σε σχέση με τους υπόλοιπους και συνιστάται για εφαρμογές όπου η απόδοση του συστήματος δεν είναι σημαντική. Επίσης, είναι ο αλγόριθμος που έχει τη μικρότερη πολυπλοκότητα.

Ο PSO θεωρείται και αυτός ένας από τους πιο κλασικούς μετά-ευρετικούς αλγορίθμους, που χρησιμοποιούνται για MPPT. Είναι πολυπλοκότερος από τον P&O, αλλά δεν έχει τα μειονεκτήματα του. Ο PSO εμφανίζει προβλήματα στην αρχικοποίηση του και για αυτό το λόγο συνιστάται τα σωματίδια του να καταλαμβάνουν όσο το δυνατό περισσότερο χώρο από το χώρο των λύσεων. Σε περιπτώσεις όμως με χαμηλή ισχύ, ο χώρος των λύσεων περιορίζεται και για αυτό το λόγο η απόδοση του συγκεκριμένου αλγορίθμου μειώθηκε στα πειράματα με χαμηλή ακτινοβολία. Εντούτοις, παρουσιάζει υψηλή απόδοση και εντοπίζει με επιτυχία το σημείο MPP στις περισσότερες περιπτώσεις. Ο χρόνος εντοπισμού του σημείου MPP για όλα τα πειράματα κυμαίνεται στο 0.5 sec και τον κατατάσσει στην δεύτερη θέση. Τέλος, να αναφερθεί ότι για την ορθή λειτουργία του αλγορίθμου απαιτείται η σωστή παραμετροποίηση του, έτσι ώστε να υπάρχει μια ισορροπία ανάμεσα στην εκμετάλλευση των υπαρχουσών λύσεων και στην εξερεύνηση νέων.

Ο PSO Advanced είναι μια νέα μορφή του αλγορίθμου PSO, που χρησιμοποιήθηκε πρώτη φορά για MPPT. Η διαφορά τους εντοπίζεται στο τρόπο με τον οποίο ανανεώνεται η ταχύτητα των σωματιδίων. Μια δεύτερη διαφορά εντοπίζεται σε ένα μηχανισμό απόρριψης των χειρότερων λύσεων. Η προσαρμογή που έγινε στο συγκεκριμένο αλγόριθμο επηρέασε την ταχύτητα σύγκλισης του. Στους πειραματισμούς που υλοποιήθηκαν, ο PSO Advanced κατατάσσεται στην πρώτη θέση, όσον αφορά την ταχύτητα εντοπισμού του σημείου MPP. Αναφορικά με την απόδοση του, εμφανίζεται ελάχιστα μεγαλύτερη από αυτή του PSO. Συνεπώς, οι αλλαγές που έγιναν στον αλγόριθμο τον οδήγησαν σε γρηγορότερο εντοπισμό του σημείου MPP. Ως αλγόριθμος παρουσιάζει τα ίδια πλεονεκτήματα και μειονεκτήματα με τον κλασικό PSO και συνιστάται για εφαρμογές που η ταχύτητα εκτέλεσης είναι σημαντική. Η πολυπλοκότητα που παρουσιάζει είναι ίδιου βαθμού με αυτήν του PSO.

Ο αλγόριθμος NBDE είναι μια μορφή της DE η οποία χρησιμοποιείται και αυτή για πρώτη φορά στο πεδίο εντοπισμού του σημείου MPP. Ο NBDE χρησιμοποιεί ένα διαφορετικό τελεστή μετάλλαξης σε σχέση με την κλασική DE και χάρις στο διαφορετικό αυτό τελεστή, η αρχική λύση ευνοείται στις πρώτες γενιές, αυξάνοντας το ρυθμό σύγκλισης, αλλά σε επόμενες γενιές η βέλτιστη λύση λαμβάνει μεγαλύτερη βαρύτητα, αποφεύγοντας την παγίδευση σε τοπικά ελάχιστα. Έτσι, στα αρχικά στάδια του αλγορίθμου ευνοείται η εξερεύνηση, ενώ καθώς τα χρωμοσώματα συγκλίνουν σε μια λύση, ευνοείται η εκμετάλλευση. Ο συγκεκριμένος αλγόριθμος εμφανίζει την υψηλότερη ακρίβεια από όλους τους άλλους αλγορίθμους που συμμετείχαν στα πειράματα και κατατάσσεται στην τελευταία θέση αναφορικά με την ταχύτητα σύγκλισης. Όπως και ο PSO, έτσι και ο NBDE απαιτεί σωστή παραμετροποίηση για βέλτιστη

απόδοση. Ο συγκεκριμένος αλγόριθμος έχει μεγαλύτερο βαθμό πολυπλοκότητας από τους υπολοίπους και συνίσταται σε εφαρμογές όπου απαιτείται η μέγιστη δυνατή απόδοση του συστήματος.

Ο αλγόριθμος AIA αποτελεί έναν νέο αλγόριθμο, ο οποίος δημιουργήθηκε από τμήματα άλλων αλγορίθμων και χρησιμοποιείται και αυτός για πρώτη φορά στον τομέα του MPPT. Από τα αποτελέσματα εξάγεται το συμπέρασμα ότι έχει επαναληψιμότητα και εμφανίζει μεγάλη απόδοση. Κατατάσσεται στη δεύτερη θέση αναφορικά με την απόδοση του και εμφανίζει ταχύτητα σύγκλισης περίπου ίση με την ταχύτητα σύγκλισης του κλασικού PSO. Ο βαθμό πολυπλοκότητας του είναι ίδιος με το βαθμό πολυπλοκότητας του NBDE και συνίσταται για χρήση σε εφαρμογές που τόσο η απόδοση, όσο και η ταχύτητα σύγκλισης είναι σημαντικές για το σύστημα.

Παρακάτω παρουσιάζεται ένας συγκεντρωτικός πίνακας των αποτελεσμάτων.

Πίνακας 7.6. Συγκεντρωτικός πίνακας αποτελεσμάτων

Αλγόριθμος	Απόδοση	Ταχύτητα	Πολυπλοκότητα	Βαθμός δυσκολίας για υλοποίηση
P&O	Χαμηλή	Πολύ μεγάλη	Μικρή	Μικρή
PSO	Υψηλή	Μεγάλη	Μικρή	Μικρή
PSO Advanced	Υψηλή	Πολύ μεγάλη	Μικρή	Μικρή
NBDE	Πολύ υψηλή	Μικρή	Μεγάλη	Μεγάλη
AIA	Υψηλή	Μεγάλη	Μεγάλη	Μεγάλη

Λαμβάνοντας υπόψιν τα δεδομένα του παραπάνω πίνακα, προκύπτει το συμπέρασμα ότι αναμφίβολα μεγάλο ενδιαφέρον παρουσιάζουν οι μέθοδοι NBDE και AIA, οι οποίες εφαρμόζονται για πρώτη φορά σε εφαρμογές MPPT και συνεπώς είναι οι μέθοδοι που προτείνεται να εφαρμόζονται σε πραγματικές συνθήκες. Μόνο έπειτα από την αξιολόγησή τους σε πραγματικές συνθήκες, δύναται να ολοκληρωθεί με ακρίβεια και αξιοπιστία η αξιολόγησή τους και να εξαχθεί ένα ασφαλές συμπέρασμα.

Κεφάλαιο 8ο: Αναφορές

- [1] Z. S. Jubaer Ahmed, «An improved perturb and observe (P&O) maximum power point tracking (MPPT) algorithm for higher efficiency,» *Applied Energy* , pp. 97-108, 5 April 2015.
- [2] E. B. M. Kermadi, "Artificial intelligence-based maximum power point tracking controllers for Photovoltaic systems: Comparative study," *Renewable and Sustainable Energy Reviews* 69, pp. 369-386, 9 November 2016.
- [3] Z. S. Kashif Ishaque, «A review of maximum power point tracking techniques of PV system for uniform insolation and partial shading condition,» *Renewable and Sustainable Energy Reviews*, pp. 475-488, 14 December 2012.
- [4] V. V. k. P. S. S. P. S. P. S. R. N. a. S. P. Kinattingal Sundareswaran, "Development of an Improved P&O Algorithm Assisted Through a Colony of Foraging Ants for MPPT in PV System," *IEEE Transactions on Industrial Informatics*, pp. 187-200, 20 November 2015.
- [5] Z. S. Jubaer Ahmed, "A Maximum Power Point Tracking (MPPT) for PV system using Cuckoo Search with partial shading capability," *Applied Energy*, pp. 118-130, 23 December 2013.
- [6] K.-C. Y. S.-C. C. Z.-J. L. Shih-Wei Lin, «Particle swarm optimization for parameter determination and feature selection of support vector machines,» *Expert Systems with Applications* 35, pp. 1817-1824, 17 September 2007.
- [7] D. T. L. L. Dongshu Wang, «Particle swarm optimization algorithm: an overview,» *Soft Computing* 22, pp. 387-408, 17 January 2017.
- [8] R. E. J. Kennedy, «Particle swarm optimization,» *IEEE*, pp. 1942-1948, 06 August 2002.
- [9] S.-C. H. J.-W. H. W.-C. L. Yi-Hwa Liu, «A Particle Swarm Optimization-Based Maximum Power Point Tracking Algorithm for PV Systems Operating Under Partially Shaded Conditions,» *IEEE Transactions on Energy Conversion*, pp. 1027-1035, 02 October 2012.
- [10] V. A. Hiren Patel, «MATLAB-Based Modeling to Study the Effects of Partial Shading on PV Array Characteristics,» *IEEE Transactions on Energy Conversion*, pp. 302-310, 15 February 2008.
- [11] D. Simon, «Evolutionary Optimization Algorithms,» σε *Biologically Inspired and Population-Based Approaches to Computer Intelligence*, New Jersey, John Wiley & Sons, Inc, 2013, pp. 265-292.
- [12] K. P. Rainer Storn, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, pp. 341-359, December 1997.
- [13] P. N. Swagatam Das, «Differential Evolution: A Survey of the State-of-the-Art,» *IEEE Transactions on Evolutionary Computation*, pp. 4-31, 14 October 2010.
- [14] Z. S. K. I. S. Hamed Taheri, «A novel Maximum Power Point tracking control of photovoltaic system under partial and rapidly fluctuating shadow conditions using Differential Evolution,»

2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA), pp. 82-87, 2011 January 2011.

- [15] S. S. Swagatam Das, «Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm,» *Information Sciences*, pp. 1237-1256, 26 November 2009.
- [16] D. L. J. C. Lian Lian Jiang, «A novel ant colony optimization-based maximum power point tracking for photovoltaic systems under partially shaded conditions,» *Energy and Buildings*, pp. 227-336, 20 December 2012.
- [17] Z. S. Jubaer Ahmed, «A Maximum Power Point Tracking (MPPT) for PV system using Cuckoo Search with partial shading capability,» *Applied Energy*, pp. 118-130, 25 January 2014.
- [18] Ι. Ν.Κιοσκερίδης, σε *Ανανεώσιμες Πηγές Ενέργειας και εφαρμογές των Ηλεκτρονικών Ισχύος*, Θεσσαλονίκη, Εκδόσεις Τζιόλα, 2017, pp. 369-422.

Κεφάλαιο 9ο: ΠΑΡΑΡΤΗΜΑ

9.1 ΠΑΡΑΡΤΗΜΑ Α: Αλγόριθμος P&O

```
function [dc, end_of_algo] = fcn(Vo,Io)
%#codegen
persistent SPo_1 SPo Ddc dc1 Cnt1

if isempty(SPo_1)      % Initialization
    SPo_1=0; SPo=0; Ddc = 0.05; dc1= 8.5 ; Cnt1=0;

end
end_of_algo = 0;
Po = Vo * Io;
% Main Program MPPT

Cnt1=Cnt1+1;

if (Cnt1>5)
    SPo=SPo+Po;
end

if (Cnt1==10)      % Period of MPPT 10*1ms

    DPo=SPo-SPo_1;

    if (DPo<0)
        Ddc=-Ddc;
    end

    if (abs(DPo)>10*5)    % 10W
        C = 3;
    else
        C = 1;
    end

    dc1=dc1+C*Ddc;

    if (dc1<0)
        dc1 = 0;
    elseif (dc1>9.5)
        dc1=9.5;
    end

    dc=dc1;
    SPo_1=SPo;
    SPo=0;
    Cnt1=0;
else
    dc=dc1;
end
```

9.2 ΠΑΡΑΡΤΗΜΑ Β: Αλγόριθμος PSO

```
function [duty,end_of_algo,position_history_i,position_history_v] =  
PSO(vout, iout, p, max_iter, current_step,mean_time,n_exp)  
  
persistent swarm n_particle step k kmax duty_old N count  
  
% Η συνάρτηση myfunc είναι απλά μια συνάρτηση που επιστρέφει ένα random  
% αριθμό.  
coder.extrinsic('myfunc')  
  
end_of_algo = 0;  
position_history_i = 0;  
position_history_v = 0;  
  
if isempty(swarm)  
    n_particle = 1; % τρέχων σωματίδιο  
    N = 3; % Πληθυσμός  
    duty_old = 0; % μνήμη για το duty  
    step = current_step; % βήμα για αλλαγή του duty  
    k = 1; % επανάληψη αλγορίθμου  
    kmax = max_iter; % μέγιστος αριθμός επαναλήψεων  
    count = 0;  
    % περιορισμός λύσεων για κάθε ένα πείραμα  
    switch n_exp  
        case 1  
            limit_min = 0.5;  
            limit_max = 8.5;  
        case 2  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 3  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 4  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 5  
            limit_min = 0.5;  
            limit_max = 6.5;  
        case 6  
            limit_min = 0.5;  
            limit_max = 2.5;  
        case 7  
            limit_min = 0.5;  
            limit_max = 2.5;  
        case 8  
            limit_min = 0.5;  
            limit_max = 7;  
        case 9  
            limit_min = 0.5;  
            limit_max = 8.5;  
        otherwise  
            limit_min = 0.5;  
            limit_max = 8.5;  
    end  
    swarm = create_swarm(N,0.1,limit_max,limit_min); % αρχικοποίηση  
    πληθυσμού  
end
```

```

%%

% έλεγχος αν οι ταχύτητες είναι πολύ μικρές ή έλεγχος αριθμού
% επαναλήψεων
if (k <= kmax) && ((sum(abs(swarm.velocity))/length(swarm.velocity)) >
0.02)

    % στο σημείο αυτό ο αλγόριθμος εξάγει την τιμή που έχει το κάθε ένα
    % σωματίδιο και ελέγχει την αντίστοιχη ισχύ του. Η συνάρτηση adjust
    % duty ομαλοποιεί τη μετάβαση από μια λύση στην επόμενη. Πχ η μετάβαση
    από τα
    % 2.5 A στα 8 A δεν γίνεται απευθείας αλλά με κάποιο βήμα.
    duty = swarm.position(swarm.index(n_particle));
    [duty,duty_old] = adjust_duty(duty,duty_old,step);

    % Αν η έξοδος του αλγορίθμου έχει φτάσει στην επιθυμητή τιμή τότε
    % αναμένουμε στην τιμή αυτή για mean_time έτσι ώστε η δειγματοληψία της
    % τιμής να έχει ορθή τιμή
    if(abs(duty - swarm.position(swarm.index(n_particle))) <= 0.0001)
        count = count + 1;
    end

    % Μετά από χρόνο mean_time αποθήκευουμε τις μετρήσεις της ισχύος κλπ.
    if count >= mean_time
        count = 0;
        swarm.p_power(swarm.index(n_particle)) = p;
        swarm.p_voltage(swarm.index(n_particle)) = vout;
        swarm.p_current(swarm.index(n_particle)) = iout;
        [swarm,position_history_i,position_history_v] =
update_particle(swarm,swarm.index(n_particle),position_history_i,position_h
istory_v);
        n_particle = n_particle + 1; % Επιλογή επόμενου σωματιδίου
        duty_old = duty;
    end

    % ανανέωση ολόκληρου του πληθυσμού
    if n_particle > N
        n_particle = 1;
        k = k + 1;
        swarm = update_swarm(swarm,k);
        N = size(swarm.index,2);
    end

else
    %% Global best - Algorithm Output
    % Αν έχει συγκλίνει ο αλγόριθμος τότε κάνουμε βγάζουμε ως έξοδο το
    % καλύτερο σωματίδιο του πληθυσμού
    %         duty = swarm.global_best;
    [duty,duty_old] = adjust_duty(swarm.global_best,duty_old,step);
    count = count + 1;
    end_of_algo =1;

    % Για μελλοντική χρήση. Για αρχικοποίηση αλγορίθμου αν αλλάξουν οι
    % περιβαλλοντικές συνθήκες.
    %         if(count > 20)
    %             if abs(100 - (p_old/p)*100) > 3
    %                 n_particle = 1;
    %                 swarm = create_swarm(N,0.1); % αρχικοποίηση
πληθυσμού

```

```

%             count = 0;
%             k = 1;
%             end_of_algo = 0;
%         end
%     else
%         p_old = p;
%     end

end
end

function [swarm,position_history_i,position_history_v] =
update_particle(swarm,i,position_history_i,position_history_v)
% Συνάρτηση η οποία ενημερώνει τη θέση του σωματιδίου
% Ενημερώνει τη βέλτιστη θέση του σωματιδίου καθώς και ολόκληρου του
% πληθυσμού
if swarm.p_power(i) > swarm.p_power_best(i)
    swarm.position_best(i) = swarm.position(i);
    swarm.p_power_best(i) = swarm.p_power(i);
    if swarm.p_power(i) > swarm.p_power_global_best
        swarm.p_power_global_best = swarm.p_power(i);
        swarm.p_voltage_best = swarm.p_voltage(i);
        swarm.global_best = swarm.position_best(i);
        swarm.p_current_best = swarm.p_current(i);
        position_history_i = swarm.p_current(i);
        position_history_v = swarm.p_voltage(i);
    end
end
end
end

function [duty,duty_old] = adjust_duty(target,duty_old,step)
% Συνάρτηση η οποία αυξάνει ή μειώνει το duty cycle έτσι ώστε να έχουμε
% ομαλή μεταβολή από ένα duty σε άλλο

if abs(duty_old - target) <= step
    duty_old = target;
elseif duty_old > target
    duty_old = duty_old - step;
else
    duty_old = duty_old + step;
end
duty = duty_old;
end

function swarm = create_swarm(n_particles,duty_best,limit_max,limit_min)
% Συνάρτηση η οποία δημιουργεί τον αρχικό πληθυσμό
% Θέτει όλες τις απαραίτητες αρχικές τιμές
field1 = 'position'; value1 = sort((limit_max -
limit_min).*myfunc(n_particles) + limit_min);
field2 = 'velocity'; value2 = myfunc(n_particles);
field3 = 'position_best';
field4 = 'global_best'; value4 = duty_best;
field5 = 'p_voltage'; value5 = zeros(1,n_particles);
field6 = 'p_current'; value6 = zeros(1,n_particles);
field7 = 'p_power'; value7 = zeros(1,n_particles);
field8 = 'p_power_best'; value8 = zeros(1,n_particles);

```

```

field9 = 'p_power_global_best'; value9 = 0;
field10 = 'w';value10 = 0.4;
field11 = 'c1';value11 = 1;
field12 = 'c2';value12 = 2;
field13 = 'r1';value13 = myfunc(1);
field14 = 'r2';value14 = myfunc(1);
field15 = 'p_voltage_best'; value15 = 0;
field16 = 'index'; value16 = linspace(1,n_particles,n_particles);
field17 = 'p_current_best'; value17 = 0;
field19 = 'search_min'; value19 = limit_min;
field20 = 'search_max'; value20 = limit_max;
swarm = struct(field1,value1,field2,value2,field3,value1,field4,value4,...

field5,value5,field6,value6,field7,value7,field8,value8,field9,value9,...
    field10,value10,field11,value11,field12,value12,field13,value13,...
    field14,value14,field15,value15,field16,value16,field17,value17,...
    field19,value19,field20,value20);
end

function swarm = update_swarm(swarm,k)
% Συνάρτηση η οποία ενημερώνει τη θέση και τη ταχύτητα των σωματιδίων
% Τέλος κάνει ταξινόμηση σε μια μεταβλητή έτσι ώστε να γνωρίζει ο
% αλγόριθμος τη σειρά με την οποία θα τα χρησιμοποιήσει για την ανανέωση
% των duty cycle
swarm.r1 = myfunc(1);
swarm.r2 = myfunc(1);
swarm.velocity = (swarm.w .* swarm.velocity) +
((swarm.c1*swarm.r1).*(swarm.position_best - swarm.position)) +
((swarm.c2*swarm.r2).*(swarm.global_best - swarm.position));
swarm.position = swarm.position + swarm.velocity;
swarm.position =
max(min(swarm.search_max,swarm.position),swarm.search_min);
if (mod(k,2) == 0)
    [~,swarm.index] = sort([swarm.position]);
else
    [~,swarm.index] = sort([swarm.position],'descend');
end
end

```

9.3 ΠΑΡΑΡΤΗΜΑ Γ: Αλγόριθμος PSO Advanced

```
function [duty,end_of_algo,position_history_i,position_history_v] =  
PSO_Advanced(vout, iout, p, max_iter,duty_step,mean_time,n_exp)  
persistent swarm n_particle step k kmax duty_old N count  
% Η συνάρτηση myfunc είναι απλά μια συνάρτηση που επιστρέφει ένα random  
% αριθμό.  
coder.extrinsic('myfunc')  
  
end_of_algo = 0;  
position_history_i = 0;  
position_history_v = 0;  
  
if isempty(swarm)  
    n_particle = 1; % τρέχων σωματίδιο  
    N = 5; % Πληθυσμός  
    duty_old = 0; % μνήμη για το duty  
    step = duty_step; % βήμα για αλλαγή του duty  
    k = 1; % επανάληψη αλγορίθμου  
    kmax = max_iter; % μέγιστος αριθμός επαναλήψεων  
    count = 0;  
  
    % περιορισμός λύσεων για κάθε ένα πείραμα  
    switch n_exp  
        case 1  
            limit_min = 0.5;  
            limit_max = 8.5;  
        case 2  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 3  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 4  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 5  
            limit_min = 0.5;  
            limit_max = 6.5;  
        case 6  
            limit_min = 0.5;  
            limit_max = 2.5;  
        case 7  
            limit_min = 0.5;  
            limit_max = 2.5;  
        case 8  
            limit_min = 0.5;  
            limit_max = 7;  
        case 9  
            limit_min = 0.5;  
            limit_max = 8.5;  
        otherwise  
            limit_min = 0.5;  
            limit_max = 8.5;  
    end  
    swarm = create_swarm(N,0.1,max_iter,limit_max,limit_min); %  
    αρχικοποίηση πληθυσμού  
end  
%%
```

```

% έλεγχος αν οι ταχύτητες είναι πολύ μικρές ή έλεγχος αριθμού
% επαναλήψεων
if (k <= kmax) && ((sum(abs(swarm.velocity))/length(swarm.velocity)) >
0.02)

    % στο σημείο αυτό ο αλγόριθμος εξάγει την τιμή που έχει το κάθε ένα
    % σωματίδιο και ελέγχει την αντίστοιχη ισχύ του. Η συνάρτηση adjust
    % duty ομαλοποιεί τη μετάβαση από μια λύση στην επόμενη. Πχ η μετάβαση
    από τα
    % 2.5 A στα 8 A δεν γίνεται απευθείας αλλά με κάποιο βήμα.
    duty = swarm.position(swarm.index(n_particle));

    % Αν η έξοδος του αλγορίθμου έχει φτάσει στην επιθυμητή τιμή τότε
    % αναμένουμε στην τιμή αυτή για mean_time έτσι ώστε η δειγματοληψία της
    % τιμής να έχει ορθή τιμή
    [duty,duty_old] = adjust_duty(duty,duty_old,step);
    if(abs(duty - swarm.position(swarm.index(n_particle))) <= 0.0001)
        count = count + 1;
    end

    %Μετά από χρόνο mean_time αποθήκευουμε τις μετρήσεις κλπ.
    if count >= mean_time
        count = 0;
        swarm.p_power(swarm.index(n_particle)) = p;
        swarm.p_voltage(swarm.index(n_particle)) = vout;
        swarm.p_current(swarm.index(n_particle)) = iout;
        %
        fprintf('Particle %d have power %d [W] with duty
%f\n',int16(n_particle),int16(swarm.p_power(swarm.index(n_particle))),swarm
.position(swarm.index(n_particle)));
        [swarm, position_history_i,position_history_v] =
update_particle(swarm,swarm.index(n_particle),position_history_i,position_h
istory_v);
        duty_old = duty;

        n_particle = n_particle + 1; % Επιλογή επόμενου σωματιδίου
    end

    if n_particle > N
        n_particle = 1;
        k = k + 1;
        % ενημέρωση πληθυσμού
        swarm = update_swarm(swarm,k);
        N = size(swarm.index,2);
    end
else
    %% Global best - Algorithm Output
    % Αν έχει συγκλίνει ο αλγόριθμος τότε κάνουμε βγάζουμε ως έξοδο το
    % καλύτερο σωματίδιο του πληθυσμού
    [duty,duty_old] = adjust_duty(swarm.global_best,duty_old,step);
    end_of_algo =1;
    % FOR LATER
    %
    count = count + 1;
    %
    if(count > 20)
    %
        if abs(100 - (p_old/p)*100) > 3
    %
            n_particle = 1;
    %
            N = 5;

```

```

%           swarm = create_swarm(N,0.1); % αρχικοποίηση πληθυσμού
%           count = 0;
%           k = 1;
%           end_of_algo =0;
%       end
%   else
%       p_old = p;
%   end
end
end

function [swarm,position_history_i,position_history_v] =
update_particle(swarm,i,position_history_i,position_history_v)
% Συνάρτηση η οποία ενημερώνει τη θέση του σωματιδίου
% Ενημερώνει τη βέλτιστη θέση του σωματιδίου καθώς και ολόκληρου του
% πληθυσμού
if swarm.p_power(i) > swarm.p_power_best(i)
    swarm.position_best(i) = swarm.position(i);
    swarm.p_power_best(i) = swarm.p_power(i);
    if swarm.p_power(i) > swarm.p_power_global_best
        swarm.p_power_global_best = swarm.p_power(i);
        swarm.p_voltage_best = swarm.p_voltage(i);
        swarm.global_best = swarm.position_best(i);
        swarm.p_current_best = swarm.p_current(i);
        position_history_i = swarm.p_current(i);
        position_history_v = swarm.p_voltage(i);
    end
end
end
end

function [duty,duty_old] = adjust_duty(target,duty_old,step)
% Συνάρτηση η οποία αυξάνει ή μειώνει το duty cycle έτσι ώστε να έχουμε
% ομαλή μεταβολή από ένα duty σε άλλο

if abs(duty_old - target) <= step
    duty_old = target;
elseif duty_old > target
    duty_old = duty_old - step;
else
    duty_old = duty_old + step;
end
duty = duty_old;
end

function swarm =
create_swarm(n_particles,duty_best,max_iter,limit_max,limit_min)
% Συνάρτηση η οποία δημιουργεί τον αρχικό πληθυσμό
% Θέτει όλες τις απαραίτητες αρχικές τιμές
field1 = 'position'; value1 = sort((limit_max -
limit_min).*myfunc(n_particles) + limit_min);
field2 = 'velocity'; value2 = myfunc(n_particles);
field3 = 'position_best';
field4 = 'global_best'; value4 = duty_best;
field5 = 'p_voltage'; value5 = zeros(1,n_particles);
field6 = 'p_current'; value6 = zeros(1,n_particles);
field7 = 'p_power'; value7 = zeros(1,n_particles);
field8 = 'p_power_best'; value8 = zeros(1,n_particles);
field9 = 'p_power_global_best'; value9 = 0;

```

```

field10 = 'w';value10 = 0.4;
field11 = 'c1';value11 = 1;
field12 = 'c2';value12 = 2;
field13 = 'r1';value13 = myfunc(1);
field14 = 'r2';value14 = myfunc(1);
field15 = 'p_voltage_best'; value15 = 0;
field16 = 'index'; value16 = linspace(1,n_particles,n_particles);
field17 = 'p_current_best'; value17 = 0;
field18 = 'history'; value18 = zeros(max_iter, 1);
field19 = 'search_min'; value19 = limit_min;
field20 = 'search_max'; value20 = limit_max;
swarm = struct(field1,value1,field2,value2,field3,value1,field4,value4,...

field5,value5,field6,value6,field7,value7,field8,value8,field9,value9,...
    field10,value10,field11,value11,field12,value12,field13,value13,...

field14,value14,field15,value15,field16,value16,field17,value17,field18,value18,...
    field19,value19,field20,value20);
end

function swarm = update_swarm(swarm,k)
% Συνάρτηση η οποία ενημερώνει τη θέση και τη ταχύτητα των σωματιδίων
% Τέλος κάνει ταξινόμηση σε μια μεταβλητή έτσι ώστε να γνωρίζει ο
% αλγόριθμος τη σειρά με την οποία θα τα χρησιμοποιήσει για την ανανέωση
% των duty cycle
r1 = 4*myfunc(1);
r2 = 4*myfunc(1);
phi = max(4.1,r1 + r2);
p = ((r1.* swarm.position_best) + (r2 .* swarm.global_best))./(r1 + r2);
a = 0.41;
kappa = (2*a)/(phi - 2);
kappa = min(2/(phi-2+sqrt(phi*(phi-4))),kappa);
swarm.velocity = kappa .* (swarm.velocity + phi .* (p - swarm.position));
swarm.position = swarm.position + swarm.velocity;
swarm.position =
max(min(swarm.search_max,swarm.position),swarm.search_min);

if (size(swarm.index,2) > 2)
    [~,swarm.index] = sort([swarm.position],'descend');
    swarm.position(swarm.index(end)) = [];
    swarm.velocity(swarm.index(end)) = [];
    swarm.position_best(swarm.index(end)) = [];

    swarm.p_voltage(swarm.index(end)) = [];
    swarm.p_current(swarm.index(end)) = [];
    swarm.p_power(swarm.index(end)) = [];
    swarm.p_power_best(swarm.index(end)) = [];
    swarm.index(end) = [];
    [~,swarm.index] = sort([swarm.position],'descend');
end
end

```

9.4 ΠΑΡΑΡΤΗΜΑ Δ: Αλγόριθμος NBDE

```
function [duty,end_of_de,position_history_i,position_history_v] =  
NBDE(vout,iout,p,duty_step,mean_time,max_iter,n_exp)  
% Η συνάρτηση myfunc είναι απλά μια συνάρτηση που επιστρέφει ένα random  
% αριθμό.  
coder.extrinsic('myfunc')  
  
%% Initialization  
persistent donor run_main_routine w...  
    find_best population duty_old count...  
    current_individual n_population generation end_of_algo duty_target...  
    adjust limit_max limit_min power_on_duty possible_global;  
  
end_of_de = 0;  
position_history_i = 0;  
position_history_v = 0;  
  
% αρχικοποιήσεις συντελεστών  
wmin = 0.1;  
wmax = 0.5;  
k = 1;  
a = 0.1;  
b = 0.1;  
CR = 0.9;  
  
if (isempty(population))  
    n_population = 10; % αριθμός χρωμοσωμάτων  
    current_individual = 1;  
    duty_old = 0;  
    count = 0;  
    generation = 1;  
    find_best = 1;  
    run_main_routine = 0;  
    donor = 0;  
    end_of_algo = 0;  
    duty_target = 0;  
    adjust = 1;  
  
    % περιορισμός λύσεων για κάθε ένα πείραμα  
    switch n_exp  
        case 1  
            limit_min = 0.5;  
            limit_max = 8.5;  
        case 2  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 3  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 4  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 5  
            limit_min = 0.5;  
            limit_max = 6.5;  
        case 6  
            limit_min = 0.5;  
            limit_max = 2.5;
```

```

        case 7
            limit_min = 0.5;
            limit_max = 2.5;
        case 8
            limit_min = 0.5;
            limit_max = 7;
        case 9
            limit_min = 0.5;
            limit_max = 8.5;
        otherwise
            limit_min = 0.5;
            limit_max = 8.5;
    end
    population =
create_population(n_population,max_iter,limit_max,limit_min);% αρχικοποίηση
πληθυσμού
    power_on_duty = 0;
    possible_global = 0;

end

if (find_best == 1) && (current_individual <= n_population)

    % στο σημείο αυτό ο αλγόριθμος εξάγει την τιμή που έχει το κάθε ένα
    % σωματίδιο και ελέγχει την αντίστοιχη ισχύ του. Η συνάρτηση adjust
    % duty ομαλοποιεί τη μετάβαση από μια λύση στην επόμενη. Πχ η μετάβαση
από 2.5 A στα 8 A δεν γίνεται απευθείας αλλά με κάποιο βήμα.
    duty = population.position(current_individual);

    % Αν η έξοδος του αλγορίθμου έχει φτάσει στην επιθυμητή τιμή τότε
    % αναμένουμε στην τιμή αυτή για mean_time έτσι ώστε η δειγματοληψία της
    % τιμής να έχει ορθή τιμή
    [duty,duty_old,power_on_duty,possible_global] =
adjust_duty(duty,duty_old,duty_step,limit_max,limit_min,power_on_duty,p,pos
sible_global);
    if(abs(duty - population.position(current_individual)) <= 0.0001)
        count = count + 1;
    end

    %Μετά από χρόνο mean_time αποθήκευουμε τις μετρήσεις κλπ.
    if(count >= mean_time)
        count = 0;
        population.p(current_individual) = vout * iout;
        population.v(current_individual) = vout;
        population.i(current_individual) = iout;

        if(population.p(current_individual) >= population.Pmax)
            population.Pmax = population.p(current_individual);
            population.v_b = vout;
            population.i_b = iout;
            population.global_best = duty;
            position_history_i = iout;
            position_history_v = vout;
        end

        current_individual = current_individual + 1;% Επιλογή επόμενου
σωματιδίου
        duty_old = duty;

```

```

end

elseif (find_best == 1) && (current_individual > n_population)
    find_best = 0;
    run_main_routine = 1;
    current_individual = 1;
    duty = duty_old;
    count = 0;

else
    duty = duty_old;
end

if run_main_routine == 1

    if (count == 0)
        % Προετοιμασία χρωμοσώματος δότη
        % Δημιουργία χρωμοσώματος δότη από τοπική γειτονιά
        z = population.position(current_individual);
        neighborInd = mod((current_individual-k : current_individual+k)-
1,n_population) + 1;
        [~,thesi_max] = max(population.p(neighborInd));
        zbest = population.position(neighborInd(thesi_max));
        neighborInd = mod( [current_individual-k:current_individual-1,
current_individual+1: current_individual+k] - 1, n_population ) + 1; % not
i
        myrandperm = zeros(2*k,2);
        myrandperm = my_randperm(2*k,2);
        pq = myrandperm;
        %       pq = neighborInd(randperm(2*k,2) );
        localChrom = z + a*(zbest - z) + b*(pq(1) - pq(2)); % Χρωμόσωμα
        δότη από τοπική γειτονιά

        % Δημιουργία χρωμοσώματος δότη από πληθυσμό
        global_neighborInd = [1:current_individual-
1,current_individual+1:n_population];
        myrandperm = zeros(n_population-1,2);
        myrandperm = my_randperm(n_population-1,2);
        r12 = global_neighborInd(myrandperm);
        %       r12 = global_neighborInd(randperm(n_population-1,2));
        g_best = population.global_best;
        globalChrom = z + a * (g_best - z) + b * (r12(1) - r12(2));

        w = wmin + (wmax - wmin) * ((generation - 1)/(max_iter -1));

        % Δημιουργία τελικού χρωμοσώματος δότη από τοπική γειτονιά και
        % συνολικό πληθυσμό
        donor =abs(w * globalChrom + (1-w)*localChrom);

        % Χρωμόσωμα γονέας
        parent = population.position(current_individual);

        %       trial = max(1,min(crossover(donor,parent,CR),9.5));
        % Διασταύρωση γονέα και δότη για να βγει το δοκιμαστικό χρωμόσωμα
        trial = crossover(donor,parent,CR);

        if (trial > population.search_max) || (trial <
population.search_min)

```

```

        my_random = 0;
        my_random = myfunc(1);
        trial = population.global_best + ((2*my_random-1)/10);
    end

    % εξαγωγή της λύσης από το δοκιμαστικό χρωμόσωμα
    duty_target = trial;
    count = count + 1;
end

% Αναμονή mean_time μέχρι τη δειγματοληψία της ισχύος από το
% δοκιμαστικό χρωμόσωμα
if(count >= mean_time)
    count = 0;
    adjust = 0;
    if(p >= population.p(current_individual))
        population.p(current_individual) = vout * iout;
        population.v(current_individual) = vout;
        population.i(current_individual) = iout;
        population.position(current_individual) = duty;
        if(population.p(current_individual) >= population.Pmax)
            population.Pmax = population.p(current_individual);
            population.v_b = vout;
            population.i_b = iout;
            population.global_best = duty;
            position_history_i = iout;
            position_history_v = vout;
        end
    end
end

current_individual = current_individual + 1;
generation = generation +1;

if(current_individual > n_population)
    % Έλεγχος συνθηκών για τερματισμό του αλγορίθμου
    current_individual = 1;
    if (generation >= max_iter)%(counter == size(pairs,1)) ||
(generation >= max_iter)
        run_main_routine = 0;
        end_of_algo = 1;
    end
end
end

if (adjust)
    [duty,duty_old,power_on_duty,possible_global] =
adjust_duty(duty_target,duty,duty_step,limit_max,limit_min,power_on_duty,p,
possible_global);
    if(abs(duty_target - duty) <= 0.0001)
        count = count + 1;
    end
else
    duty = duty_old;
    adjust = 1;
end
end

```

```

else
    if end_of_algo
        % τερματισμός αλγορίθμου και εξαγωγή λύσης καλύτερου χρωμοσώματος
        end_of_de = 1;
        duty = population.global_best;
        % duty_old = duty;
        [duty,duty_old,power_on_duty] =
adjust_duty(duty,duty_old,duty_step,limit_max,limit_min,power_on_duty,p,possible_global);
        % FOR LATER
        % count = count + 1;
        % if(count > 20)
        %     if abs(100 - (p_old/p)*100) > 3
        %         find_best = 1;
        %         end_of_algo = 0;
        %         current_individual = 1;
        %         population.position =
linspace(1,9.5,n_population);
        % population.Pmax = 0;
        % count = 0;
        % generation = 1;
        % end
        % else
        %     p_old = p;
        % end
    end

end

end
end

```

```

function trial = crossover(donor,parent,CR)
% συνάρτηση η οποία υλοποιεί τη διασταύρωση δύο χρωμοσωμάτων
parent_new = dec2bin(round(parent * 1000),15);
parent_new = double(parent_new == '1');
donor_new = dec2bin(round(donor * 1000),15);
donor_new = double(donor_new == '1');
% create trial vector u
myrandperm = zeros(15,1);
myrandperm = my_randperm(length(parent_new),1);
jrand = myrandperm;
% jrand = randperm(length(parent_new),1);
mui = myfunc(length(parent_new))' < CR; %all random numbers < Cr are 1, 0 otherwise
mui(jrand) = 1;
mpo = mui < 0.5; % inverse mask to mui
u = parent_new.* mpo' + donor_new.*mui';

% Limit values of the activators
u = min(1,max(0,u));
trial = bin2dec(char(u+'0'))/1000;
end

```

```

function population = create_population(n_individuals,
max_iter,limit_max,limit_min)
% Συνάρτηση η οποία δημιουργεί τον αρχικό πληθυσμό
% Θέτει όλες τις απαραίτητες αρχικές τιμές
field1 = 'position'; value1 = sort((limit_max -
limit_min).*myfunc(n_individuals) + limit_min);
field2 = 'p';value2 = zeros(1,numel(value1));
field3 = 'global_best'; value3 = 0;
field4 = 'Pmax'; value4 = 0;
field5 = 'v';value5 = zeros(1,numel(value1));
field6 = 'i';value6 = zeros(1,numel(value1));
field7 = 'v_b';value7 = zeros(1,numel(value1));
field8 = 'i_b';value8 = zeros(1,numel(value1));
field9 = 'search_min';value9 = limit_min;
field10 = 'search_max';value10 = limit_max;
field11 = 'history'; value11 = zeros(max_iter, 1);
population =
struct(field1,value1,field2,value2,field3,value3,field4,value4,...

field5,value5,field6,value6,field7,value7,field8,value8,field9,value9,field
10,value10,field11,value11);
end

function [duty,duty_old,power_on_duty,possible_global] =
adjust_duty(target,duty_old,step,limit_max,limit_min,power_on_duty,p,possib
le_global)
% Συνάρτηση η οποία αυξάνει ή μειώνει το duty cycle έτσι ώστε να έχουμε
% ομαλή μεταβολή από ένα duty σε άλλο
if abs(duty_old - target) <= step
    duty_old = target;
elseif duty_old > target
    if (abs(duty_old - target) > 0.1) && (abs(limit_max - limit_min)<=5)
        duty_old = duty_old - (step/5);
    else
        duty_old = duty_old - step;
    end
    if p > power_on_duty
        power_on_duty = p;
        possible_global = duty_old;
    end
else
    if abs(duty_old - target) > 0.1 && (abs(limit_max - limit_min)<=5)
        duty_old = duty_old + (step/5);
    else
        duty_old = duty_old + step;
    end
    if p > power_on_duty
        power_on_duty = p;
        possible_global = duty_old;
    end
end
duty = duty_old;
end

```

9.5 ΠΑΡΑΡΤΗΜΑ Ε: Αλγόριθμος ΑΙΑ

```
function [duty,end_of_ants,position_history_i,position_history_v] =  
ANTS(vout,iout,p,duty_step,mean_time,n_exp)  
  
% Η συνάρτηση myfunc είναι απλά μια συνάρτηση που επιστρέφει ένα random  
% αριθμό.  
coder.extrinsic('myfunc')  
  
%% Initialization  
persistent find_best population duty_old count...  
    iter_number generation end_of_algo...  
    K dx power_on_duty possible_global limit_max limit_min  
  
end_of_ants = 0;  
position_history_i = 0;  
position_history_v = 0;  
  
if (isempty(population))  
    iter_number = 1;  
    K = 10; % αριθμός πληθυσμού  
    duty_old = 0;  
    dx = 0;  
    count = 0;  
    generation = 1;  
    find_best = 1;  
    end_of_algo = 0;  
  
    % περιορισμός λύσεων για κάθε ένα πείραμα  
    switch n_exp  
        case 1  
            limit_min = 0.5;  
            limit_max = 8.5;  
        case 2  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 3  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 4  
            limit_min = 0.5;  
            limit_max = 7.5;  
        case 5  
            limit_min = 0.5;  
            limit_max = 6.5;  
        case 6  
            limit_min = 0.5;  
            limit_max = 2.5;  
        case 7  
            limit_min = 0.5;  
            limit_max = 2.5;  
        case 8  
            limit_min = 0.5;  
            limit_max = 7;  
        case 9  
            limit_min = 0.5;  
            limit_max = 8.5;  
        otherwise
```

```

        limit_min = 0.5;
        limit_max = 8.5;
    end
    population = create_population(K,limit_max,limit_min); % αρχικοποίηση
    πληθυσμού
    power_on_duty = 0;
    possible_global = 0;

end

if (find_best == 1) && (iter_number <= K)

    % στο σημείο αυτό ο αλγόριθμος εξάγει την τιμή που έχει το κάθε ένα
    % σωματίδιο και ελέγχει την αντίστοιχη ισχύ του. Η συνάρτηση adjust
    % duty ομαλοποιεί τη μετάβαση από μια λύση στην επόμενη. Πχ η μετάβαση
    από 2.5 A στα 8 A δεν γίνεται απευθείας αλλά με κάποιο βήμα.
    duty = population.position(iter_number);

    % Αν η έξοδος του αλγορίθμου έχει φτάσει στην επιθυμητή τιμή τότε
    % αναμένουμε στην τιμή αυτή για mean_time έτσι ώστε η δειγματοληψία της
    % τιμής να έχει ορθή τιμή
    [duty,duty_old,power_on_duty,possible_global] =
    adjust_duty(duty,duty_old,duty_step,limit_max,limit_min,power_on_duty,p,possible_global);
    if(abs(duty - population.position(iter_number)) <= 0.01)
        count = count + 1;
    end

    % Μετά από χρόνο mean_time αποθήκευουμε τις μετρήσεις κλπ.
    if(count >= mean_time)
        count = 0;
        population.p(iter_number) = p;
        population.v(iter_number) = vout;
        population.i(iter_number) = iout;

        if(p >= population.Pmax)
            population.Pmax = p;
            population.v_b = vout;
            population.i_b = iout;
            population.global_best = duty;
            position_history_i = iout;
            position_history_v = vout;
        end

        iter_number = iter_number + 1;
    end

elseif (find_best == 1) && (iter_number > K)
    % Sorting τον πληθυσμό
    population = sort_archive(population);
    find_best = 0;
    iter_number = 1;
    duty = duty_old;
    count = 0;
end

```

```

else
    if(end_of_algo == 0)
        if(iter_number <= K)
            if (count == 0)

                % Υπολογισμός βήματος dx από Levy Walks
                my_random = 0;
                my_random = myfunc(1);
                beta = 1.9*my_random + 0.1;
                su_num = gamma(1 + beta) * sin(pi * beta / 2);
                su_den = gamma((1 + beta)/2) * beta * (2^((beta - 1)/2));
                su = (su_num / su_den)^(1/beta);
                u = normrnd(0,su^2);

                n = normrnd(0,1);

                dx = (u/(abs(n)^(1/beta))) *
(population.position(iter_number) - population.global_best);

                dx = possible_global + dx/10; % υπολογισμός νέας θέσης

                if (dx > population.search_max) || (dx <
population.search_min)
                    my_random = 0;
                    my_random = myfunc(1);
                    dx = population.global_best + ((2*my_random-1)/10);

                end
                count = count + 1;

            end
            % εξαγωγή λύσης από νέα θέση και δημιουργία νέου ατόμου στον
            % πληθυσμό στη νέα αυτή θέση
            [duty,duty_old,power_on_duty,possible_global] =
adjust_duty(dx,duty_old,duty_step,limit_max,limit_min,power_on_duty,p,possible_global);
            if(abs(duty - dx) <= 0.001)
                count = count + 1;
            end

            if(count >= mean_time)
                count = 0;
                iter_number = iter_number + 1;
                population.position = [population.position, duty];
                population.p = [population.p, p];
                population.v = [population.v, vout];
                population.i = [population.i, iout];
                if(p >= population.Pmax)
                    population.Pmax = p;
                    population.v_b = vout;
                    population.i_b = iout;
                    population.global_best = duty;
                    position_history_i = iout;
                    position_history_v = vout;
                end
            end

        end
    end
end

```

```

else
    iter_number = 1;
    generation = generation + 1;
    % διαγραφή K χειρότερων λύσεων από τον πληθυσμό
    population = sort_archive_2(population,K);
    population = sort_archive(population);

    % κριτήρια τερματισμού αλγορίθμου
    pairs = nchoosek(population.position,2);
    counter = 0;
    for i = 1:size(pairs,1)
        value = abs((pairs(i,1) - pairs(i,2))/((pairs(i,1) +
pairs(i,2))/2)) * 100;
        if(value <= 5)
            counter = counter + 1;
        end
    end

    if (counter == size(pairs,1))
        end_of_algo = 1;

    end
    duty=duty_old;
end
else
    % τερματισμός αλγορίθμου και έξοδος λύσης του καλύτερου ατόμου στον
    % πληθυσμό
    end_of_ants = 1;
    duty = population.global_best;
    [duty,duty_old,power_on_duty,possible_global] =
adjust_duty(duty,duty_old,duty_step,limit_max,limit_min,power_on_duty,p,possible_global);
    count = count + 1;
    %         if(count > 20)
    %             if abs(100 - (p_old/p)*100) > 3
    %                 iter_number = 1;
    %                 end_of_algo = 0;
    %                 count = 0;
    %                 population = create_population(K);
    %                 generation = 1;
    %                 find_best = 1;
    %             end
    %         else
    %             p_old = p;
    %         end
end
end
end

```

end

```

function population = sort_archive(population)
% Συνάρτηση που κάνει απλά sorting τα άτομα του πληθυσμού
[~,population.index] = sort([population.p], 'descend');
end

function population = sort_archive_2(population,K)
% Συνάρτηση που διαγράφει K χειρότερα άτομα από τον πληθυσμό

```

```
[~,population.index] = sort([population.p], 'descend');
index = population.index;
dummy = population.position;
dummy(index(length(dummy) - K-1:length(dummy))) = [];
population.position = dummy;
```

```
dummy = population.p;
dummy(index(length(dummy) - K-1:length(dummy))) = [];
population.p = dummy;
```

```
dummy = population.v;
dummy(index(length(dummy) - K-1:length(dummy))) = [];
population.v = dummy;
```

```
dummy = population.i;
dummy(index(length(dummy) - K-1:length(dummy))) = [];
population.i = dummy;
```

```
dummy = population.index;
dummy(index(length(dummy) - K-1:length(dummy))) = [];
population.index = dummy;
end
```

```
function population = create_population(n_individuals, limit_max, limit_min)
% Συνάρτηση η οποία δημιουργεί τον αρχικό πληθυσμό
% Θέτει όλες τις απαραίτητες αρχικές τιμές
field1 = 'position'; value1 = sort((limit_max -
limit_min).*myfunc(n_individuals) + limit_min);
field2 = 'p'; value2 = zeros(1, numel(value1));
field3 = 'global_best'; value3 = 0;
field4 = 'Pmax'; value4 = 0;
field5 = 'v'; value5 = zeros(1, numel(value1));
field6 = 'i'; value6 = zeros(1, numel(value1));
field7 = 'v_b'; value7 = 0;
field8 = 'i_b'; value8 = 0;
field9 = 'index'; value9 = linspace(1, n_individuals, n_individuals);
field10 = 't'; value10 = linspace(0, 0, n_individuals);
field11 = 't_p'; value11 = linspace(0, 0, n_individuals);
field12 = 'search_min'; value12 = limit_min;
field13 = 'search_max'; value13 = limit_max;
population =
struct(field1, value1, field2, value2, field3, value3, field4, value4, ...
field5, value5, field6, value6, field7, value7, field8, value8, field9, value9, field
10, value10, field11, value11, ...
field12, value12, field13, value13);
end
```

```
function [duty, duty_old, power_on_duty, possible_global] =
adjust_duty(target, duty_old, step, limit_max, limit_min, power_on_duty, p, possib
le_global)
% Συνάρτηση η οποία αυξάνει ή μειώνει το duty cycle έτσι ώστε να έχουμε
% ομαλή μεταβολή από ένα duty σε άλλο
if abs(duty_old - target) <= step
duty_old = target;
elseif duty_old > target
if (abs(duty_old - target) > 0.1) && (abs(limit_max - limit_min) <= 5)
duty_old = duty_old - (step/5);
else
```

```

        duty_old = duty_old - step;
    end
    if p > power_on_duty
        power_on_duty = p;
        possible_global = duty_old;
    end
else
    if abs(duty_old - target) > 0.1 && (abs(limit_max - limit_min)<=5)
        duty_old = duty_old + (step/5);
    else
        duty_old = duty_old + step;
    end
    if p > power_on_duty
        power_on_duty = p;
        possible_global = duty_old;
    end
end

duty = duty_old;
end

```

9.6 ΠΑΡΑΡΤΗΜΑ ΣΤ: GUI

Το μεγαλύτερο μέρος του κώδικα είναι auto generated από το matlab. Παρακάτω παρουσιάζεται ο κώδικας έτσι όπως έχει γίνει exported από το λογισμικό.

```
classdef app1_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        UIAxes_2                matlab.ui.control.UIAxes
        UIAxes                  matlab.ui.control.UIAxes
        UIAxes_All              matlab.ui.control.UIAxes
        FileMenu                matlab.ui.container.Menu
        ImportParametersMenu    matlab.ui.container.Menu
        Experiment1Menu         matlab.ui.container.Menu
        Experiment2Menu         matlab.ui.container.Menu
        Experiment3Menu         matlab.ui.container.Menu
        Experiment4Menu         matlab.ui.container.Menu
        Experiment5Menu         matlab.ui.container.Menu
        Experiment6Menu         matlab.ui.container.Menu
        Experiment7Menu         matlab.ui.container.Menu
        Experiment8Menu         matlab.ui.container.Menu
        Experiment9Menu         matlab.ui.container.Menu
        TemperatureoCSpinnerLabel matlab.ui.control.Label
        TemperatureoCSpinner    matlab.ui.control.Spinner
        Irradiancewm2SpinnerLabel matlab.ui.control.Label
        Irradiancewm2Spinner    matlab.ui.control.Spinner
        TemperatureoCSpinner_2Label matlab.ui.control.Label
        TemperatureoCSpinner_2  matlab.ui.control.Spinner
        Irradiancewm2Spinner_2Label matlab.ui.control.Label
        Irradiancewm2Spinner_2  matlab.ui.control.Spinner
        TemperatureoCSpinner_3Label matlab.ui.control.Label
        TemperatureoCSpinner_3  matlab.ui.control.Spinner
        Irradiancewm2Spinner_3Label matlab.ui.control.Label
        Irradiancewm2Spinner_3  matlab.ui.control.Spinner
        TemperatureoCSpinner_4Label matlab.ui.control.Label
        TemperatureoCSpinner_4  matlab.ui.control.Spinner
        Irradiancewm2Spinner_4Label matlab.ui.control.Label
        Irradiancewm2Spinner_4  matlab.ui.control.Spinner
        TemperatureoCSpinner_5Label matlab.ui.control.Label
        TemperatureoCSpinner_5  matlab.ui.control.Spinner
        Irradiancewm2Spinner_5Label matlab.ui.control.Label
        Irradiancewm2Spinner_5  matlab.ui.control.Spinner
        Group1Label             matlab.ui.control.Label
        Group2Label             matlab.ui.control.Label
        Group3Label             matlab.ui.control.Label
        Group4Label             matlab.ui.control.Label
        Group5Label             matlab.ui.control.Label
        NumberofPanelsSpinnerLabel matlab.ui.control.Label
        NumberofPanelsSpinner    matlab.ui.control.Spinner
        NumberofPanelsSpinner_2Label matlab.ui.control.Label
        NumberofPanelsSpinner_2  matlab.ui.control.Spinner
        NumberofPanelsSpinner_3Label matlab.ui.control.Label
        NumberofPanelsSpinner_3  matlab.ui.control.Spinner
        NumberofPanelsSpinner_4Label matlab.ui.control.Label
        NumberofPanelsSpinner_4  matlab.ui.control.Spinner
        NumberofPanelsSpinner_5Label matlab.ui.control.Label
        NumberofPanelsSpinner_5  matlab.ui.control.Spinner
        ClearWorkspaceButton     matlab.ui.control.Button
        Simulation_Lamp          matlab.ui.control.Lamp
    end
end
```



```

end

methods (Access = private)

function add_experiments(app,value)

    hold (app.UIAxes,"off")
    hold (app.UIAxes_2,"off")
    hold (app.UIAxes_All, "off")

    evalin('base','clear variables')
    delete(timerfindall);
    clc

    [app.TemperatureoCSpinner.Value,
app.Irradiancewm2Spinner.Value, ...
    app.TemperatureoCSpinner_2.Value,
app.Irradiancewm2Spinner_2.Value, ...
    app.TemperatureoCSpinner_3.Value,
app.Irradiancewm2Spinner_3.Value, ...
    app.TemperatureoCSpinner_4.Value,
app.Irradiancewm2Spinner_4.Value, ...
    app.TemperatureoCSpinner_5.Value,
app.Irradiancewm2Spinner_5.Value, ...
    Nss_1, Nss_2, Nss_3, Nss_4, Nss_5, Nss_6, Npp ]=
select_exp(value);

    app.NumberofPanelsSpinner.Value = Nss_1;
    app.NumberofPanelsSpinner_2.Value = Nss_2;
    app.NumberofPanelsSpinner_3.Value = Nss_3;
    app.NumberofPanelsSpinner_4.Value = Nss_4;
    app.NumberofPanelsSpinner_5.Value = Nss_5;
    assignin('base','Npp',Npp);
    assignin('base','Nss_6',Nss_6);
    app.update_parameter_for_simulation();
    app.EditField2.Value = 'Experiment_' + string(value);
    app.RunSimButton.Visible = 'on';
    app.Simulation_Lamp.Color = [1,1,1];
    delete(app.plot_1);
    delete(app.plot_2);
    delete(app.plot_1_max);
    delete(app.plot_2_max);
    delete(app.plot_algo_max_1);
    delete(app.plot_algo_max_2);
    delete(app.text_label_1);
    delete(app.text_label_2);
    delete(app.plot_all);
    delete(app.leg_all);

    plot(app.UIAxes,0,0);

    plot(app.UIAxes_2,0,0);

    plot(app.UIAxes_All,0,0);

    app.PmaxEditField.Value = 0;
    app.ImaxEditField.Value = 0;

```

```

app.VmaxEditField.Value = 0;
app.enable_plot = 0;

app.PmaxEditField_2.Value = 0;
app.ImaxEditField_2.Value = 0;
app.VmaxEditField_2.Value = 0;
app.TimeEditField_2.Value = 0;

app.PmaxEditField_3.Value = 0;
app.ImaxEditField_3.Value = 0;
app.VmaxEditField_3.Value = 0;
app.TimeEditField_3.Value = 0;

app.PmaxEditField_4.Value = 0;
app.ImaxEditField_4.Value = 0;
app.VmaxEditField_4.Value = 0;
app.TimeEditField_4.Value = 0;

app.PmaxEditField_5.Value = 0;
app.ImaxEditField_5.Value = 0;
app.VmaxEditField_5.Value = 0;
app.TimeEditField_5.Value = 0;

app.PmaxEditField_6.Value = 0;
app.ImaxEditField_6.Value = 0;
app.VmaxEditField_6.Value = 0;
app.TimeEditField_6.Value = 0;

app.AccEditField_2.Value = 0;
app.AccEditField_3.Value = 0;
app.AccEditField_4.Value = 0;
app.AccEditField_5.Value = 0;
app.AccEditField_6.Value = 0;

assignin('base','n_algorithm',map_algo(app,app.AlgorithmDropDown.Value));
assignin('base','n_experiment',app.n_exp);

end

function update_parameter_for_simulation(app)
    assignin('base','t_pv1',app.TemperatureoCSpinner.Value);
    assignin('base','w_pv1',app.Irradiancewm2Spinner.Value);

    assignin('base','t_pv2',app.TemperatureoCSpinner_2.Value);
    assignin('base','w_pv2',app.Irradiancewm2Spinner_2.Value);

    assignin('base','t_pv3',app.TemperatureoCSpinner_3.Value);
    assignin('base','w_pv3',app.Irradiancewm2Spinner_3.Value);

    assignin('base','t_pv4',app.TemperatureoCSpinner_4.Value);
    assignin('base','w_pv4',app.Irradiancewm2Spinner_4.Value);

    assignin('base','t_pv5',app.TemperatureoCSpinner_5.Value);

```

```

assignin('base','w_pv5',app.Irradiancewm2Spinner_5.Value);

assignin('base','Nss_1',app.NumberofPanelsSpinner.Value);
assignin('base','Nss_2',app.NumberofPanelsSpinner_2.Value);
assignin('base','Nss_3',app.NumberofPanelsSpinner_3.Value);
assignin('base','Nss_4',app.NumberofPanelsSpinner_4.Value);
assignin('base','Nss_5',app.NumberofPanelsSpinner_5.Value);
assignin('base','time_algo',0);

end

function delete_timer(~,mTimer,~)
    stop(mTimer);
    delete(mTimer);
end

function number_algo = map_algo(~,value)

    if strcmp(value,'None')
        number_algo = 0;
    elseif strcmp(value,'P&O')
        number_algo = 1;
    elseif strcmp(value,'PSO')
        number_algo = 2;
    elseif strcmp(value,'PSO Advanced')
        number_algo = 3;
    elseif strcmp(value,'NBDE')
        number_algo = 4;
    elseif strcmp(value,'AIA')
        number_algo = 5;
    end
end

function plot_basic(app)

    app.Simulation_Lamp.Color = [0,1,0];
    yaxis_1 = evalin('base','Po');
    xaxis_1 = evalin('base','Vo');
    yaxis_2 = evalin('base','Io');
    xaxis_2 = evalin('base','Vo');

    app.RunSimButton.Visible = 'on';
    app.UIAxes.Visible = 'on';
    app.UIAxes_2.Visible = 'on';
    P_filtered = medfilt1(yaxis_1,100);
    [Pmax,index] = max(P_filtered);
    Vmax = xaxis_1(index);
    assignin("base",'Pmax',Pmax);
    assignin("base",'Imax',Vmax);
    app.UIAxes.YLim = [0,5000];
    app.UIAxes.XLim = [0,700];
    app.UIAxes.Title.String = 'PV Graph';
    app.UIAxes.XLabel.String = 'Voltage [V]';
    app.UIAxes.YLabel.String = 'Power [W]';

```

```

app.plot_1 = plot(app.UIAxes,xaxis_1,P_filtered);
hold(app.UIAxes,"on")
app.plot_1_max =
plot(app.UIAxes,Vmax,Pmax,"*", 'color', 'black');

app.UIAxes_2.YLim = [0,10];
app.UIAxes_2.XLim = [0,700];
Imax = yaxis_2(index);
app.UIAxes_2.Title.String = 'IV Graph';
app.UIAxes_2.XLabel.String = 'Voltage [V]';
app.UIAxes_2.YLabel.String = 'Current [A]';
app.plot_2 = plot(app.UIAxes_2,xaxis_2,yaxis_2);
hold(app.UIAxes_2,"on")
app.plot_2_max =
plot(app.UIAxes_2,Vmax,Imax,"*", 'color', 'black');

app.EditField.Value = 'Simulation finished';
assignin('base','plot_1',app.plot_1);
assignin('base','plot_2',app.plot_2);

Po = evalin('base','Po');
Vo = evalin('base','Vo');
Io = evalin('base','Io');

[Pmax,index] = max(medfilt1(Po,100));
app.PmaxEditField.Value = Pmax;
app.ImaxEditField.Value = Io(index);
app.VmaxEditField.Value = Vo(index);
app.AccEditField_2.Value = (app.PmaxEditField_2.Value / Pmax) *
100;
app.AccEditField_3.Value = (app.PmaxEditField_3.Value / Pmax) *
100;
app.AccEditField_4.Value = (app.PmaxEditField_4.Value / Pmax) *
100;
app.AccEditField_5.Value = (app.PmaxEditField_5.Value / Pmax) *
100;
app.AccEditField_6.Value = (app.PmaxEditField_6.Value / Pmax) *
100;

end

function plot_algo(app)

app.Simulation_Lamp.Color = [0,1,0];
yaxis_1 = evalin('base','Po');
xaxis_1 = evalin('base','Vo');
yaxis_2 = evalin('base','Io');
time_algo = evalin('base',"time_algo");
end_of_algo = evalin("base","end_of_algo");

app.RunSimButton.Visible = 'on';
app.UIAxes.Visible = 'on';
app.UIAxes_2.Visible = 'on';

```

```

        if strcmp(app.AlgorithmDropDown.Value, "P&O")
            Pmax = medfilt1(yaxis_1,20);
            value = max(Pmax(200:length(Pmax)));
            indexOfFirstMax = find((abs(Pmax - value) <= 0.1),
1, 'first');

            run_time = time_algo(indexOfFirstMax);
            vmax = xaxis_1(indexOfFirstMax);
            imax = yaxis_2(indexOfFirstMax);

            app.plot_all =
plot(app.UIAxes_All,time_algo,yaxis_1, 'DisplayName', app.AlgorithmDropDown.V
alue);

            app.leg_all = legend(app.UIAxes_All, "show");

        else
            [~,b] = max(end_of_algo);
            run_time = time_algo(b);
            value = yaxis_1(b+500);
            vmax = xaxis_1(b+500);
            imax = yaxis_2(b+500);

            app.plot_all =
plot(app.UIAxes_All,time_algo,yaxis_1, 'DisplayName', app.AlgorithmDropDown.V
alue);

            app.leg_all = legend(app.UIAxes_All, "show");
        end

        hold(app.UIAxes, "on")
        hold(app.UIAxes_2, "on")
        hold(app.UIAxes_All, "on");

        assignin("base", 'Pmax_algo', value);
        assignin("base", 'Imax_algo', vmax);
        assignin("base", "Time_algo", run_time);

        if strcmp(app.AlgorithmDropDown.Value, "P&O")
            value1 = 20;
            value2 = 150;
            value3 = 20;
            value4 = 0.5;
        elseif strcmp(app.AlgorithmDropDown.Value, "PSO")
            value1 = -50;
            value2 = 150;
            value3 = -50;
            value4 = 0.5;
        elseif strcmp(app.AlgorithmDropDown.Value, "PSO Advanced")
            value1 = 20;
            value2 = -150;
            value3 = 20;
            value4 = -0.5;
        elseif strcmp(app.AlgorithmDropDown.Value, "NBDE")
            value1 = -50;
            value2 = -150;
            value3 = -50;

```

```

        value4 = -0.5;
    elseif strcmp(app.AlgorithmDropDown.Value, "AIA")
        value1 = 0;
        value2 = 400;
        value3 = 0;
        value4 = 1;
    end

    app.plot_algo_max_1 =
plot(app.UIAxes, vmax, value, "+", 'color', 'red');
    app.text_label_1 =
text(app.UIAxes, vmax+value1, value+value2, app.AlgorithmDropDown.Value);
    app.plot_algo_max_2 =
plot(app.UIAxes_2, vmax, imax, "+", 'color', 'red');
    app.text_label_2 =
text(app.UIAxes_2, vmax+value3, imax+value4, app.AlgorithmDropDown.Value);

    app.UIAxes_All.Title.String = 'Algorithm output';
    app.UIAxes_All.XLabel.String = 'Time [sec]';
    app.UIAxes_All.YLabel.String = 'Power [W]';

    app.EditField.Value = 'Simulation finished';
    display_algo_results(app, value, imax, vmax, run_time);

end

function display_algo_results(app, pmax, imax, vmax, run_time)
    dummy_value = map_algo(app, app.AlgorithmDropDown.Value);
    switch dummy_value
        case 1
            app.PmaxEditField_2.Value = pmax;
            app.ImaxEditField_2.Value = imax;
            app.VmaxEditField_2.Value = vmax;
            app.TimeEditField_2.Value = run_time;
            app.AccEditField_2.Value = (app.PmaxEditField_2.Value /
app.PmaxEditField.Value) * 100;

            case 2
                app.PmaxEditField_3.Value = pmax;
                app.ImaxEditField_3.Value = imax;
                app.VmaxEditField_3.Value = vmax;
                app.TimeEditField_3.Value = run_time;
                app.AccEditField_3.Value = (app.PmaxEditField_3.Value /
app.PmaxEditField.Value) * 100;

                case 3
                    app.PmaxEditField_4.Value = pmax;
                    app.ImaxEditField_4.Value = imax;
                    app.VmaxEditField_4.Value = vmax;
                    app.TimeEditField_4.Value = run_time;
                    app.AccEditField_4.Value = (app.PmaxEditField_4.Value /
app.PmaxEditField.Value) * 100;

                    case 4

```

```

        app.PmaxEditField_5.Value = pmax;
        app.ImaxEditField_5.Value = imax;
        app.VmaxEditField_5.Value = vmax;
        app.TimeEditField_5.Value = run_time;
        app.AccEditField_5.Value = (app.PmaxEditField_5.Value /
app.PmaxEditField.Value) * 100;
        case 5
            app.PmaxEditField_6.Value = pmax;
            app.ImaxEditField_6.Value = imax;
            app.VmaxEditField_6.Value = vmax;
            app.TimeEditField_6.Value = run_time;
            app.AccEditField_6.Value = (app.PmaxEditField_6.Value /
app.PmaxEditField.Value) * 100;

        end

    end

end

methods (Access = public)

    function update_event(app,~,~,~,~)
        try
            app.EditField.Value = 'Simulation is running';
            if ~isempty(evalin('base','Po'))
                stop(app.t);
                if evalin("base","n_algorithm") == 0
                    app.plot_basic();
                    stop(app.t);
                else
                    app.plot_algo();
                end
                delete(app.t);
            end

            catch ME

            end

        end

    end

end

% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        app.Simulation_Lamp.Color = [1,1,1];
        app.EditField.Value = 'Simulation not running';
        app.RunSimButton.Visible = 'off';
        app.ClearWorkspaceButtonPushed();
        app.PmaxEditField.Value = 0;
        app.enable_plot = 0;
        assignin('base','n_algorithm',0);
        assignin('base','n_experiment',1);
    end
end

```

```

end

% Callback function
function PlotDataButtonPushed(app, event)

end

% Menu selected function: Experiment1Menu
function Experiment1MenuSelected(app, event)
    app.add_experiments(1); app.n_exp = 1;
assignin('base', 'n_experiment', app.n_exp);
end

% Menu selected function: Experiment2Menu
function Experiment2MenuSelected(app, event)
    app.add_experiments(2); app.n_exp = 2;
assignin('base', 'n_experiment', app.n_exp);
end

% Menu selected function: Experiment3Menu
function Experiment3MenuSelected(app, event)
    app.add_experiments(3); app.n_exp = 3;
assignin('base', 'n_experiment', app.n_exp);
end

% Menu selected function: Experiment4Menu
function Experiment4MenuSelected(app, event)
    app.add_experiments(4); app.n_exp = 4;
assignin('base', 'n_experiment', app.n_exp);
end

% Menu selected function: Experiment5Menu
function Experiment5MenuSelected(app, event)
    app.add_experiments(5); app.n_exp = 5;
assignin('base', 'n_experiment', app.n_exp);
end

% Menu selected function: Experiment6Menu
function Experiment6MenuSelected(app, event)
    app.add_experiments(6); app.n_exp = 6;
assignin('base', 'n_experiment', app.n_exp);
end

% Menu selected function: Experiment7Menu
function Experiment7MenuSelected(app, event)
    app.add_experiments(7); app.n_exp = 7;
assignin('base', 'n_experiment', app.n_exp);
end

% Menu selected function: Experiment8Menu
function Experiment8MenuSelected(app, event)
    app.add_experiments(8); app.n_exp = 8;
assignin('base', 'n_experiment', app.n_exp);
end

% Menu selected function: Experiment9Menu
function Experiment9MenuSelected(app, event)
    app.add_experiments(9); app.n_exp = 9;
assignin('base', 'n_experiment', app.n_exp);

```

```

end

% Button pushed function: ClearWorkspaceButton
function ClearWorkspaceButtonPushed(app, event)

    hold (app.UIAxes, "off")
    hold (app.UIAxes_2, "off")
    hold (app.UIAxes_All, "off")

    app.Simulation_Lamp.Color = [1,1,1];
    app.RunSimButton.Visible = 'off';
    app.EditField.Value = 'Simulation not running';

    app.TemperatureoCSpinner.Value = 0;
    app.Irradiancewm2Spinner.Value = 0;
    app.TemperatureoCSpinner_2.Value = 0;
    app.Irradiancewm2Spinner_2.Value = 0;
    app.TemperatureoCSpinner_3.Value = 0;
    app.Irradiancewm2Spinner_3.Value = 0;
    app.TemperatureoCSpinner_4.Value = 0;
    app.Irradiancewm2Spinner_4.Value = 0;
    app.TemperatureoCSpinner_5.Value = 0;
    app.Irradiancewm2Spinner_5.Value = 0;
    app.NumberofPanelsSpinner.Value = 0;
    app.NumberofPanelsSpinner_2.Value = 0;
    app.NumberofPanelsSpinner_3.Value = 0;
    app.NumberofPanelsSpinner_4.Value = 0;
    app.NumberofPanelsSpinner_5.Value = 0;
    app.EditField2.Value = '';

    delete(app.plot_1);
    delete(app.plot_2);
    delete(app.plot_1_max);
    delete(app.plot_2_max);
    delete(app.plot_algo_max_1);
    delete(app.plot_algo_max_2);
    delete(app.text_label_1);
    delete(app.text_label_2);
    delete(app.plot_all);
    delete(app.leg_all);

    plot(app.UIAxes, 0, 0);

    plot(app.UIAxes_2, 0, 0);

    plot(app.UIAxes_All, 0, 0);

    app.PmaxEditField.Value = 0;
    app.ImaxEditField.Value = 0;
    app.VmaxEditField.Value = 0;
    app.enable_plot = 0;

    app.PmaxEditField_2.Value = 0;

```

```

app.ImaxEditField_2.Value = 0;
app.VmaxEditField_2.Value = 0;
app.TimeEditField_2.Value = 0;

app.PmaxEditField_3.Value = 0;
app.ImaxEditField_3.Value = 0;
app.VmaxEditField_3.Value = 0;
app.TimeEditField_3.Value = 0;

app.PmaxEditField_4.Value = 0;
app.ImaxEditField_4.Value = 0;
app.VmaxEditField_4.Value = 0;
app.TimeEditField_4.Value = 0;

app.PmaxEditField_5.Value = 0;
app.ImaxEditField_5.Value = 0;
app.VmaxEditField_5.Value = 0;
app.TimeEditField_5.Value = 0;

app.PmaxEditField_6.Value = 0;
app.ImaxEditField_6.Value = 0;
app.VmaxEditField_6.Value = 0;
app.TimeEditField_6.Value = 0;

app.AccEditField_2.Value = 0;
app.AccEditField_3.Value = 0;
app.AccEditField_4.Value = 0;
app.AccEditField_5.Value = 0;
app.AccEditField_6.Value = 0;

evalin('base','clear variables')
clc

assignin('base','n_algorithm',map_algo(app,app.AlgorithmDropDown.Value));
assignin('base','n_experiment',1);

end

% Button pushed function: RunSimButton
function RUNSimulationRunSimButtonPushed(app, event)

    try
        evalin("base","clear Po");

        evalin("base","clear Io");

        evalin("base","clear Vo");

    catch ME

    end

app.Simulation_Lamp.Color = [1,0,0];
app.RunSimButton.Visible = 'off';

```

```

app.update_parameter_for_simulation();
app.enable_plot = 0;

try
    if strcmp(app.AlgorithmDropDown.Value, 'None')
        assignin("base", 'period', 0.010);

set_param('Partial_Shadow_5_2019', 'SimulationCommand', 'start', 'StopTime', '2
');
        else
            assignin("base", 'period', 0.001);

set_param('Partial_Shadow_5_2019', 'SimulationCommand', 'start', 'StopTime', '3
');

        end

        %
        sim('Partial_Shadow_5.slx', 2);
    catch ME
    end
    delete(timerfindall);

    app.EditField.Value = 'Simulation running';
    app.t = timer;
    app.t.TimerFcn = @app.update_event;
    app.t.StopFcn = @app.delete_timer;
    app.t.Period = 1;
    %
    app.t.StartDelay = 10;
    app.t.TasksToExecute = 1500;
    app.t.ExecutionMode = 'FixedRate';
    start(app.t);

end

% Value changed function: AlgorithmDropDown
function AlgorithmDropDownValueChanged(app, event)
    value = app.AlgorithmDropDown.Value;
    assignin('base', 'n_algorithm', map_algo(app, value));
    assignin('base', 'n_experiment', app.n_exp);
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app.UIFigure = uifigure('Visible', 'off');
    app.UIFigure.Position = [100 100 1351 825];
    app.UIFigure.Name = 'UI Figure';
    app.UIFigure.Interruptible = 'off';
    app.UIFigure.Scrollable = 'on';

    % Create UIAxes_2
    app.UIAxes_2 = uiaxes(app.UIFigure);
    title(app.UIAxes_2, 'Title')
    xlabel(app.UIAxes_2, 'X')

```

```

ylabel(app.UIAxes_2, 'Y')
app.UIAxes_2.Position = [265 85 433 311];

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'Title')
xlabel(app.UIAxes, 'X')
ylabel(app.UIAxes, 'Y')
app.UIAxes.Position = [265 429 433 311];

% Create UIAxes_All
app.UIAxes_All = uiaxes(app.UIFigure);
title(app.UIAxes_All, 'Title')
xlabel(app.UIAxes_All, 'X')
ylabel(app.UIAxes_All, 'Y')
app.UIAxes_All.Position = [740 333 562 401];

% Create FileMenu
app.FileMenu = uimenu(app.UIFigure);
app.FileMenu.Text = 'File';

% Create ImportParametersMenu
app.ImportParametersMenu = uimenu(app.FileMenu);
app.ImportParametersMenu.Text = 'Import Parameters';

% Create Experiment1Menu
app.Experiment1Menu = uimenu(app.ImportParametersMenu);
app.Experiment1Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment1MenuSelected, true);
app.Experiment1Menu.Text = 'Experiment 1';

% Create Experiment2Menu
app.Experiment2Menu = uimenu(app.ImportParametersMenu);
app.Experiment2Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment2MenuSelected, true);
app.Experiment2Menu.Text = 'Experiment 2';

% Create Experiment3Menu
app.Experiment3Menu = uimenu(app.ImportParametersMenu);
app.Experiment3Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment3MenuSelected, true);
app.Experiment3Menu.Text = 'Experiment 3';

% Create Experiment4Menu
app.Experiment4Menu = uimenu(app.ImportParametersMenu);
app.Experiment4Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment4MenuSelected, true);
app.Experiment4Menu.Text = 'Experiment 4';

% Create Experiment5Menu
app.Experiment5Menu = uimenu(app.ImportParametersMenu);
app.Experiment5Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment5MenuSelected, true);
app.Experiment5Menu.Text = 'Experiment 5';

% Create Experiment6Menu
app.Experiment6Menu = uimenu(app.ImportParametersMenu);
app.Experiment6Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment6MenuSelected, true);

```

```

app.Experiment6Menu.Text = 'Experiment 6';

% Create Experiment7Menu
app.Experiment7Menu = uimenu(app.ImportParametersMenu);
app.Experiment7Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment7MenuSelected, true);
app.Experiment7Menu.Text = 'Experiment 7';

% Create Experiment8Menu
app.Experiment8Menu = uimenu(app.ImportParametersMenu);
app.Experiment8Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment8MenuSelected, true);
app.Experiment8Menu.Text = 'Experiment 8';

% Create Experiment9Menu
app.Experiment9Menu = uimenu(app.ImportParametersMenu);
app.Experiment9Menu.MenuSelectedFcn = createCallbackFcn(app,
@Experiment9MenuSelected, true);
app.Experiment9Menu.Text = 'Experiment 9';

% Create TemperatureoCSpinnerLabel
app.TemperatureoCSpinnerLabel = uilabel(app.UIFigure);
app.TemperatureoCSpinnerLabel.HorizontalAlignment = 'right';
app.TemperatureoCSpinnerLabel.Position = [27 769 98 22];
app.TemperatureoCSpinnerLabel.Text = 'Temperature [oC]';

% Create TemperatureoCSpinner
app.TemperatureoCSpinner = uispinner(app.UIFigure);
app.TemperatureoCSpinner.Position = [140 769 100 22];

% Create Irradiancewm2SpinnerLabel
app.Irradiancewm2SpinnerLabel = uilabel(app.UIFigure);
app.Irradiancewm2SpinnerLabel.HorizontalAlignment = 'right';
app.Irradiancewm2SpinnerLabel.Position = [27 739 98 22];
app.Irradiancewm2SpinnerLabel.Text = 'Irradiance [w/m2]';

% Create Irradiancewm2Spinner
app.Irradiancewm2Spinner = uispinner(app.UIFigure);
app.Irradiancewm2Spinner.Position = [140 739 100 22];

% Create TemperatureoCSpinner_2Label
app.TemperatureoCSpinner_2Label = uilabel(app.UIFigure);
app.TemperatureoCSpinner_2Label.HorizontalAlignment = 'right';
app.TemperatureoCSpinner_2Label.Position = [28 655 98 22];
app.TemperatureoCSpinner_2Label.Text = 'Temperature [oC]';

% Create TemperatureoCSpinner_2
app.TemperatureoCSpinner_2 = uispinner(app.UIFigure);
app.TemperatureoCSpinner_2.Position = [141 655 100 22];

% Create Irradiancewm2Spinner_2Label
app.Irradiancewm2Spinner_2Label = uilabel(app.UIFigure);
app.Irradiancewm2Spinner_2Label.HorizontalAlignment = 'right';
app.Irradiancewm2Spinner_2Label.Position = [28 625 98 22];
app.Irradiancewm2Spinner_2Label.Text = 'Irradiance [w/m2]';

% Create Irradiancewm2Spinner_2
app.Irradiancewm2Spinner_2 = uispinner(app.UIFigure);
app.Irradiancewm2Spinner_2.Position = [141 625 100 22];

```

```

% Create TemperatureoCSpinner_3Label
app.TemperatureoCSpinner_3Label = uilabel(app.UIFigure);
app.TemperatureoCSpinner_3Label.HorizontalAlignment = 'right';
app.TemperatureoCSpinner_3Label.Position = [28 535 98 22];
app.TemperatureoCSpinner_3Label.Text = 'Temperature [oC]';

% Create TemperatureoCSpinner_3
app.TemperatureoCSpinner_3 = uispinner(app.UIFigure);
app.TemperatureoCSpinner_3.Position = [141 535 100 22];

% Create Irradiancewm2Spinner_3Label
app.Irradiancewm2Spinner_3Label = uilabel(app.UIFigure);
app.Irradiancewm2Spinner_3Label.HorizontalAlignment = 'right';
app.Irradiancewm2Spinner_3Label.Position = [28 505 98 22];
app.Irradiancewm2Spinner_3Label.Text = 'Irradiance [w/m2]';

% Create Irradiancewm2Spinner_3
app.Irradiancewm2Spinner_3 = uispinner(app.UIFigure);
app.Irradiancewm2Spinner_3.Position = [141 505 100 22];

% Create TemperatureoCSpinner_4Label
app.TemperatureoCSpinner_4Label = uilabel(app.UIFigure);
app.TemperatureoCSpinner_4Label.HorizontalAlignment = 'right';
app.TemperatureoCSpinner_4Label.Position = [28 395 98 22];
app.TemperatureoCSpinner_4Label.Text = 'Temperature [oC]';

% Create TemperatureoCSpinner_4
app.TemperatureoCSpinner_4 = uispinner(app.UIFigure);
app.TemperatureoCSpinner_4.Position = [141 395 100 22];

% Create Irradiancewm2Spinner_4Label
app.Irradiancewm2Spinner_4Label = uilabel(app.UIFigure);
app.Irradiancewm2Spinner_4Label.HorizontalAlignment = 'right';
app.Irradiancewm2Spinner_4Label.Position = [28 365 98 22];
app.Irradiancewm2Spinner_4Label.Text = 'Irradiance [w/m2]';

% Create Irradiancewm2Spinner_4
app.Irradiancewm2Spinner_4 = uispinner(app.UIFigure);
app.Irradiancewm2Spinner_4.Position = [141 365 100 22];

% Create TemperatureoCSpinner_5Label
app.TemperatureoCSpinner_5Label = uilabel(app.UIFigure);
app.TemperatureoCSpinner_5Label.HorizontalAlignment = 'right';
app.TemperatureoCSpinner_5Label.Position = [27 267 98 22];
app.TemperatureoCSpinner_5Label.Text = 'Temperature [oC]';

% Create TemperatureoCSpinner_5
app.TemperatureoCSpinner_5 = uispinner(app.UIFigure);
app.TemperatureoCSpinner_5.Position = [140 267 100 22];

% Create Irradiancewm2Spinner_5Label
app.Irradiancewm2Spinner_5Label = uilabel(app.UIFigure);
app.Irradiancewm2Spinner_5Label.HorizontalAlignment = 'right';
app.Irradiancewm2Spinner_5Label.Position = [27 237 98 22];
app.Irradiancewm2Spinner_5Label.Text = 'Irradiance [w/m2]';

% Create Irradiancewm2Spinner_5
app.Irradiancewm2Spinner_5 = uispinner(app.UIFigure);

```

```

app.Irradiancewm2Spinner_5.Position = [140 237 100 22];

% Create Group1Label
app.Group1Label = uilabel(app.UIFigure);
app.Group1Label.Position = [167 790 49 22];
app.Group1Label.Text = 'Group 1';

% Create Group2Label
app.Group2Label = uilabel(app.UIFigure);
app.Group2Label.Position = [168 676 49 22];
app.Group2Label.Text = 'Group 2';

% Create Group3Label
app.Group3Label = uilabel(app.UIFigure);
app.Group3Label.Position = [168 556 49 22];
app.Group3Label.Text = 'Group 3';

% Create Group4Label
app.Group4Label = uilabel(app.UIFigure);
app.Group4Label.Position = [168 416 49 22];
app.Group4Label.Text = 'Group 4';

% Create Group5Label
app.Group5Label = uilabel(app.UIFigure);
app.Group5Label.Position = [167 288 49 22];
app.Group5Label.Text = 'Group 5';

% Create NumberofPanelsSpinnerLabel
app.NumberofPanelsSpinnerLabel = uilabel(app.UIFigure);
app.NumberofPanelsSpinnerLabel.HorizontalAlignment = 'right';
app.NumberofPanelsSpinnerLabel.Position = [25 707 102 22];
app.NumberofPanelsSpinnerLabel.Text = 'Number of Panels';

% Create NumberofPanelsSpinner
app.NumberofPanelsSpinner = uispinner(app.UIFigure);
app.NumberofPanelsSpinner.Position = [142 707 100 22];

% Create NumberofPanelsSpinner_2Label
app.NumberofPanelsSpinner_2Label = uilabel(app.UIFigure);
app.NumberofPanelsSpinner_2Label.HorizontalAlignment = 'right';
app.NumberofPanelsSpinner_2Label.Position = [25 596 102 22];
app.NumberofPanelsSpinner_2Label.Text = 'Number of Panels';

% Create NumberofPanelsSpinner_2
app.NumberofPanelsSpinner_2 = uispinner(app.UIFigure);
app.NumberofPanelsSpinner_2.Position = [142 596 100 22];

% Create NumberofPanelsSpinner_3Label
app.NumberofPanelsSpinner_3Label = uilabel(app.UIFigure);
app.NumberofPanelsSpinner_3Label.HorizontalAlignment = 'right';
app.NumberofPanelsSpinner_3Label.Position = [23 474 102 22];
app.NumberofPanelsSpinner_3Label.Text = 'Number of Panels';

% Create NumberofPanelsSpinner_3
app.NumberofPanelsSpinner_3 = uispinner(app.UIFigure);
app.NumberofPanelsSpinner_3.Position = [140 474 100 22];

% Create NumberofPanelsSpinner_4Label
app.NumberofPanelsSpinner_4Label = uilabel(app.UIFigure);

```

```

app.NumberofPanelsSpinner_4Label.HorizontalAlignment = 'right';
app.NumberofPanelsSpinner_4Label.Position = [24 333 102 22];
app.NumberofPanelsSpinner_4Label.Text = 'Number of Panels';

% Create NumberofPanelsSpinner_4
app.NumberofPanelsSpinner_4 = uispinner(app.UIFigure);
app.NumberofPanelsSpinner_4.Position = [141 333 100 22];

% Create NumberofPanelsSpinner_5Label
app.NumberofPanelsSpinner_5Label = uilabel(app.UIFigure);
app.NumberofPanelsSpinner_5Label.HorizontalAlignment = 'right';
app.NumberofPanelsSpinner_5Label.Position = [23 206 102 22];
app.NumberofPanelsSpinner_5Label.Text = 'Number of Panels';

% Create NumberofPanelsSpinner_5
app.NumberofPanelsSpinner_5 = uispinner(app.UIFigure);
app.NumberofPanelsSpinner_5.Position = [140 206 100 22];

% Create ClearWorkspaceButton
app.ClearWorkspaceButton = uibutton(app.UIFigure, 'push');
app.ClearWorkspaceButton.ButtonPushedFcn =
createCallbackFcn(app, @ClearWorkspaceButtonPushed, true);
app.ClearWorkspaceButton.Position = [34 135 107 22];
app.ClearWorkspaceButton.Text = 'Clear Workspace';

% Create Simulation_Lamp
app.Simulation_Lamp = uilamp(app.UIFigure);
app.Simulation_Lamp.Position = [283 792 30 30];
app.Simulation_Lamp.Color = [1 1 1];

% Create RunSimButton
app.RunSimButton = uibutton(app.UIFigure, 'push');
app.RunSimButton.ButtonPushedFcn = createCallbackFcn(app,
@RUNSimulationRunSimButtonPushed, true);
app.RunSimButton.Interruptible = 'off';
app.RunSimButton.Visible = 'off';
app.RunSimButton.Position = [155 135 100 22];
app.RunSimButton.Text = 'RUN';

% Create EditField
app.EditField = uieditfield(app.UIFigure, 'text');
app.EditField.HorizontalAlignment = 'center';
app.EditField.Position = [324 789 147 33];

% Create EditField2
app.EditField2 = uieditfield(app.UIFigure, 'text');
app.EditField2.Position = [854 796 105 22];

% Create PmaxEditFieldLabel
app.PmaxEditFieldLabel = uilabel(app.UIFigure);
app.PmaxEditFieldLabel.HorizontalAlignment = 'right';
app.PmaxEditFieldLabel.Position = [55 82 36 22];
app.PmaxEditFieldLabel.Text = 'Pmax';

% Create PmaxEditField
app.PmaxEditField = uieditfield(app.UIFigure, 'numeric');
app.PmaxEditField.Position = [103 82 105 22];

% Create VmaxEditFieldLabel

```

```

app.VmaxEditFieldLabel = uilabel(app.UIFigure);
app.VmaxEditFieldLabel.HorizontalAlignment = 'right';
app.VmaxEditFieldLabel.Position = [55 50 36 22];
app.VmaxEditFieldLabel.Text = 'Vmax';

% Create VmaxEditField
app.VmaxEditField = uieditfield(app.UIFigure, 'numeric');
app.VmaxEditField.Position = [103 50 105 22];

% Create ImaxEditFieldLabel
app.ImaxEditFieldLabel = uilabel(app.UIFigure);
app.ImaxEditFieldLabel.HorizontalAlignment = 'right';
app.ImaxEditFieldLabel.Position = [56 19 32 22];
app.ImaxEditFieldLabel.Text = 'Imax';

% Create ImaxEditField
app.ImaxEditField = uieditfield(app.UIFigure, 'numeric');
app.ImaxEditField.Position = [103 19 105 22];

% Create WLabel
app.WLabel = uilabel(app.UIFigure);
app.WLabel.Position = [216 82 25 22];
app.WLabel.Text = '[W]';

% Create VLabel
app.VLabel = uilabel(app.UIFigure);
app.VLabel.Position = [217 50 25 22];
app.VLabel.Text = '[V]';

% Create ALabel
app.ALabel = uilabel(app.UIFigure);
app.ALabel.Position = [218 19 25 22];
app.ALabel.Text = '[A]';

% Create AlgorithmDropDownLabel
app.AlgorithmDropDownLabel = uilabel(app.UIFigure);
app.AlgorithmDropDownLabel.HorizontalAlignment = 'right';
app.AlgorithmDropDownLabel.Position = [485 796 56 22];
app.AlgorithmDropDownLabel.Text = 'Algorithm';

% Create AlgorithmDropDown
app.AlgorithmDropDown = uidropdown(app.UIFigure);
app.AlgorithmDropDown.Items = {'None', 'P&O', 'PSO', 'PSO
Advanced', 'NBDE', 'AIA'};
app.AlgorithmDropDown.ValueChangedFcn = createCallbackFcn(app,
@AlgorithmDropDownValueChanged, true);
app.AlgorithmDropDown.Position = [557 792 153 26];
app.AlgorithmDropDown.Value = 'None';

% Create WLabel_2
app.WLabel_2 = uilabel(app.UIFigure);
app.WLabel_2.Position = [1298 227 25 22];
app.WLabel_2.Text = '[W]';

% Create VLabel_2
app.VLabel_2 = uilabel(app.UIFigure);
app.VLabel_2.Position = [1298 206 25 22];
app.VLabel_2.Text = '[V]';

```

```

% Create ALabel_2
app.ALabel_2 = uilabel(app.UIFigure);
app.ALabel_2.Position = [1298 185 25 22];
app.ALabel_2.Text = '[A]';

% Create PmaxEditField_2Label
app.PmaxEditField_2Label = uilabel(app.UIFigure);
app.PmaxEditField_2Label.HorizontalAlignment = 'right';
app.PmaxEditField_2Label.Position = [735 227 36 22];
app.PmaxEditField_2Label.Text = 'Pmax';

% Create PmaxEditField_2
app.PmaxEditField_2 = uieditfield(app.UIFigure, 'numeric');
app.PmaxEditField_2.Position = [783 227 105 22];

% Create VmaxEditField_2Label
app.VmaxEditField_2Label = uilabel(app.UIFigure);
app.VmaxEditField_2Label.HorizontalAlignment = 'right';
app.VmaxEditField_2Label.Position = [735 206 36 22];
app.VmaxEditField_2Label.Text = 'Vmax';

% Create VmaxEditField_2
app.VmaxEditField_2 = uieditfield(app.UIFigure, 'numeric');
app.VmaxEditField_2.Position = [783 206 105 22];

% Create ImaxEditField_2Label
app.ImaxEditField_2Label = uilabel(app.UIFigure);
app.ImaxEditField_2Label.HorizontalAlignment = 'right';
app.ImaxEditField_2Label.Position = [736 185 32 22];
app.ImaxEditField_2Label.Text = 'Imax';

% Create ImaxEditField_2
app.ImaxEditField_2 = uieditfield(app.UIFigure, 'numeric');
app.ImaxEditField_2.Position = [783 185 105 22];

% Create secLabel
app.secLabel = uilabel(app.UIFigure);
app.secLabel.Position = [1298 163 31 22];
app.secLabel.Text = '[sec]';

% Create TimeEditField_2
app.TimeEditField_2 = uieditfield(app.UIFigure, 'numeric');
app.TimeEditField_2.Position = [783 164 105 22];

% Create ImaxEditField_2Label_2
app.ImaxEditField_2Label_2 = uilabel(app.UIFigure);
app.ImaxEditField_2Label_2.HorizontalAlignment = 'right';
app.ImaxEditField_2Label_2.Position = [735 163 32 22];
app.ImaxEditField_2Label_2.Text = 'Time';

% Create PmaxEditField_3
app.PmaxEditField_3 = uieditfield(app.UIFigure, 'numeric');
app.PmaxEditField_3.Position = [887 227 100 22];

% Create VmaxEditField_3
app.VmaxEditField_3 = uieditfield(app.UIFigure, 'numeric');
app.VmaxEditField_3.Position = [887 206 100 22];

% Create ImaxEditField_3

```

```

app.ImaxEditField_3 = uieditfield(app.UIFigure, 'numeric');
app.ImaxEditField_3.Position = [887 185 100 22];

% Create TimeEditField_3
app.TimeEditField_3 = uieditfield(app.UIFigure, 'numeric');
app.TimeEditField_3.Position = [887 164 100 22];

% Create PmaxEditField_4
app.PmaxEditField_4 = uieditfield(app.UIFigure, 'numeric');
app.PmaxEditField_4.Position = [986 227 100 22];

% Create VmaxEditField_4
app.VmaxEditField_4 = uieditfield(app.UIFigure, 'numeric');
app.VmaxEditField_4.Position = [986 206 100 22];

% Create ImaxEditField_4
app.ImaxEditField_4 = uieditfield(app.UIFigure, 'numeric');
app.ImaxEditField_4.Position = [986 185 100 22];

% Create TimeEditField_4
app.TimeEditField_4 = uieditfield(app.UIFigure, 'numeric');
app.TimeEditField_4.Position = [986 164 100 22];

% Create PmaxEditField_5
app.PmaxEditField_5 = uieditfield(app.UIFigure, 'numeric');
app.PmaxEditField_5.Position = [1085 227 100 22];

% Create VmaxEditField_5
app.VmaxEditField_5 = uieditfield(app.UIFigure, 'numeric');
app.VmaxEditField_5.Position = [1085 206 100 22];

% Create ImaxEditField_5
app.ImaxEditField_5 = uieditfield(app.UIFigure, 'numeric');
app.ImaxEditField_5.Position = [1085 185 100 22];

% Create TimeEditField_5
app.TimeEditField_5 = uieditfield(app.UIFigure, 'numeric');
app.TimeEditField_5.Position = [1085 164 100 22];

% Create PmaxEditField_6
app.PmaxEditField_6 = uieditfield(app.UIFigure, 'numeric');
app.PmaxEditField_6.Position = [1184 227 100 22];

% Create VmaxEditField_6
app.VmaxEditField_6 = uieditfield(app.UIFigure, 'numeric');
app.VmaxEditField_6.Position = [1184 206 100 22];

% Create ImaxEditField_6
app.ImaxEditField_6 = uieditfield(app.UIFigure, 'numeric');
app.ImaxEditField_6.Position = [1184 185 100 22];

% Create TimeEditField_6
app.TimeEditField_6 = uieditfield(app.UIFigure, 'numeric');
app.TimeEditField_6.Position = [1184 164 100 22];

% Create POLabel
app.POLabel = uilabel(app.UIFigure);
app.POLabel.Position = [819 256 31 22];
app.POLabel.Text = 'P&O';

```

```

% Create PSOLabel
app.PSOLabel = uilabel(app.UIFigure);
app.PSOLabel.Position = [921 256 31 22];
app.PSOLabel.Text = 'PSO';

% Create PSOAdvancedLabel
app.PSOAdvancedLabel = uilabel(app.UIFigure);
app.PSOAdvancedLabel.Position = [992 256 88 22];
app.PSOAdvancedLabel.Text = 'PSO Advanced';

% Create NBDELabel
app.NBDELabel = uilabel(app.UIFigure);
app.NBDELabel.Position = [1115 256 39 22];
app.NBDELabel.Text = 'NBDE';

% Create AIALabel
app.AIALabel = uilabel(app.UIFigure);
app.AIALabel.Position = [1221 256 25 22];
app.AIALabel.Text = 'AIA';

% Create Label
app.Label = uilabel(app.UIFigure);
app.Label.Position = [1298 143 25 22];
app.Label.Text = ' [%]';

% Create AccEditField_2
app.AccEditField_2 = uieditfield(app.UIFigure, 'numeric');
app.AccEditField_2.Position = [783 143 105 22];

% Create ImaxEditField_2Label_3
app.ImaxEditField_2Label_3 = uilabel(app.UIFigure);
app.ImaxEditField_2Label_3.HorizontalAlignment = 'right';
app.ImaxEditField_2Label_3.Position = [716 142 55 22];
app.ImaxEditField_2Label_3.Text = 'Accuracy';

% Create AccEditField_3
app.AccEditField_3 = uieditfield(app.UIFigure, 'numeric');
app.AccEditField_3.Position = [887 143 100 22];

% Create AccEditField_4
app.AccEditField_4 = uieditfield(app.UIFigure, 'numeric');
app.AccEditField_4.Position = [986 143 100 22];

% Create AccEditField_5
app.AccEditField_5 = uieditfield(app.UIFigure, 'numeric');
app.AccEditField_5.Position = [1085 143 100 22];

% Create AccEditField_6
app.AccEditField_6 = uieditfield(app.UIFigure, 'numeric');
app.AccEditField_6.Position = [1184 143 100 22];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

```

```

% Construct app
function app = app1_exported

    % Create UIFigure and components
    createComponents(app)

    % Register the app with App Designer
    registerApp(app, app.UIFigure)

    % Execute the startup function
    runStartupFcn(app, @startupFcn)

    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end

```