



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανολόγων Μηχανικών
Πολυτεχνική Σχολή

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Πολυκριτηριακή βελτιστοποίηση σχεδιασμού ηλιακού θερμικού συστήματος, υποβοηθούμενου από αντλία θερμότητας, για κάλυψη θερμικών φορτίων»

ΚΑΡΑΝΑΣΟΣ ΑΓΓΕΛΟΣ,

A.E.M.:2086

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΠΑΝΑΡΑΣ ΓΕΩΡΓΙΟΣ

Κοζάνη

2021

Περίληψη

Η συνεχής αύξηση των ενεργειακών μας αναγκών αποτελεί ένα βασικό πρόβλημα στην εποχή μας, το οποίο με σωστή διαχείριση αυτών και χρήση κατάλληλων μέσων παραγωγής της ενέργειας μπορεί να μειωθεί η ζημία του προβλήματος. Η χρήση κατάλληλων μέσων παραγωγής ενέργειας αφορά κυρίως συστήματα που εκμεταλλεύονται Ανανεώσιμες Πηγές Ενέργειας, οι οποίες πρόκειται για πηγές φιλικές προς το περιβάλλον και ανεξάντλητες. Με δεδομένο ότι ο κτιριακός τομέας είναι υπεύθυνος για το 40% περίπου της συνολικής κατανάλωσης ενέργειας, η εξοικονόμηση ενέργειας στα κτίρια θεωρείται απαραίτητη και μπορεί να επιτευχθεί είτε με το βιοκλιματικό σχεδιασμό των κτιρίων είτε με τη χρήση πιο αποδοτικών συστημάτων θέρμανσης και κλιματισμού είτε με σωστή λειτουργία-συμπεριφορά των χρηστών.

Κάποιες βασικές έννοιες θέρμανσης και κλιματισμού όπως η θερμότητα, η θερμοκρασία και η θερμοπερατότητα αναλύονται στη συνέχεια για την περαιτέρω κατανόηση παραγόντων που σχετίζονται με την ενέργεια. Επιπλέον, αυτοί οι παράγοντες αφορούν και τη θερμική συμπεριφορά των κτιρίων που αποτελείται από τρία στάδια τις θερμικές ανάγκες, τα θερμικά κέρδη και τις θερμικές απώλειες των κτιρίων. Οι αντλίες θερμότητας αφορούν συστήματα τα οποία επιτρέπουν την μεταφορά θερμότητας από χώρους χαμηλότερης θερμοκρασίας σε χώρους υψηλότερης θερμοκρασίας βασιζόμενες στον αντίστροφο κύκλο Carnot. Ένα άλλο σύστημα που καταγράφεται στην συνέχεια είναι τα ηλιακά θερμικά συστήματα, τα οποία είναι συστήματα που εκμεταλλεύονται την ηλιακή ενέργεια με στόχο την κάλυψη θερμικών φορτίων ενός κτιρίου.

Επόμενο στάδιο της εργασίας είναι η εισαγωγή στα μονοκριτηριακά και πολυκριτηριακά προβλήματα βελτιστοποίησης. Επιπρόσθετα αναφέρεται ο σκοπός της παρούσας εργασίας ο οποίος ήταν η πολυκριτηριακή βελτιστοποίηση σχεδιασμού ηλιακού θερμικού συστήματος υποβοηθούμενου από αντλία θερμότητας, για κάλυψη θερμικών φορτίων. Το συγκεκριμένο πρόβλημα βελτιστοποίησης διαθέτει δύο κριτήρια απόφασης:

1. την ελαχιστοποίηση της συνεισφοράς ισχύος της αντλίας θερμότητας, $Q_{HP,cons}$ [kWh]
2. τη μεγιστοποίηση του μέσου ποσοστού κάλυψης των φορτίων ZNX για την ικανοποίηση των χρηστών, DHW_{cov} [%].

Αυτά τα κριτήρια βασίστηκαν στην κατάλληλη επιλογή πέντε παραμέτρων σχεδιασμού και λειτουργίας οι οποίες είναι:

- ύψος δεξαμενής ζεστού νερού χρήσης
- επιφάνεια επίπεδου ηλιακού συλλέκτη
- είδος επίπεδου ηλιακού συλλέκτη
- παροχή νερού της αντλίας θερμότητας
- όριο θερμοκρασίας λειτουργίας της αντλίας θερμότητας

Η εξεταζόμενη διάταξη, η οποία πρόκειται για μία θεωρητική διάταξη, περιλαμβάνει έναν επίπεδο ηλιακό συλλέκτη, μία αντλία θερμότητας, δύο δεξαμενές αποθήκευσης, τρεις

εναλλάκτες θερμότητας και δύο καταβόθρες θερμότητας. Οι εξισώσεις που διέπουν αυτά τα συστήματα προήλθαν από το εγχειρίδιο TRNSYS 16.

Η μεθοδολογία εύρεσης βέλτιστων σεναρίων που ακολουθήθηκε αποτελείται από πέντε βήματα τα οποία επιλύθηκαν με την βοήθεια λογισμικών, το Spyder και το ALAMO. Αρχικά έγινε η δημιουργία του μοντέλου προσομοίωσης στο λογισμικό Spyder. Στην συνέχεια έγινε η υλοποίηση του αλγορίθμου ελέγχου ο οποίος αποτελούταν από δύο αλγορίθμους:

- τον αλγόριθμο ρύθμισης της ροής εξόδου από τον συλλέκτη
- τον αλγόριθμο λειτουργίας της αντλίας θερμότητας

Μέσω της ανάλυσης του μοντέλου προσομοίωσης προέκυψαν σεναρία βέλτιστων παραμέτρων σχεδιασμού και λειτουργίας. Με τα αποτελέσματα των σεναρίων αυτών δημιουργήθηκαν οι μηνιαίες καμπύλες του ορίου Pareto (Pareto frontier), τα μηνιαία υποκατάστατα μοντέλα προσομοίωσης (Surrogate Simulation Model), καθώς και τα υποκατάστατα μοντέλα των ορίων Pareto (Surrogate Pareto frontier). Τελευταίο βήμα της μεθοδολογίας ήταν ο προσδιορισμός του επιθυμητού ποσοστού κάλυψης φορτίων ZNX από τον ηλιακό συλλέκτη, ο υπολογισμός των μηνιαίων (βέλτιστων) τιμών της κατανάλωσης ισχύος της αντλίας θερμότητας μέσω των υποκατάστατων μοντέλων των ορίων Pareto. Επιπλέον, πραγματοποιήθηκε η δημιουργία πολλών σεναρίων βελτιστοποίησης μέσω των υποκατάστατων μοντέλων προσομοίωσης και επιλογή του βέλτιστου σεναρίου βάσει του ελάχιστου αθροίσματος των τετραγωνικών σχετικών σφαλμάτων (SSRE) των σεναρίων βελτιστοποίησης. Στη συνέχεια παρουσιάζονται τα αποτελέσματα που εξήλθαν από την επίλυση της συγκεκριμένης μεθοδολογίας, για το πρόβλημα της πολυκριτηριακής βελτιστοποίησης σχεδιασμού ηλιακού θερμικού συστήματος, υποβοηθούμενου από αντλία θερμότητας, για κάλυψη θερμικών φορτίων.

Τέλος, καταγράφονται τα συμπεράσματα και μερικές προτάσεις για μελλοντική έρευνα που προέκυψαν από την εκπόνηση της παρούσας εργασίας.

Abstract

The constant increase in our energy needs is a major problem nowadays, which with proper management of these and the use of appropriate energy production methods can reduce the damage of the problem. The use of these methods mainly concerns systems that exploit Renewable Energy Sources, which are environmentally friendly and inexhaustible. Given that the building sector is responsible for about 40% of total energy consumption, energy saving in buildings is considered essential and can be achieved either through bioclimatic design of buildings, the use of more efficient heating and air-conditioning systems or through proper user behavior.

Some basic heating and air conditioning concepts such as heat, temperature and thermal permeability are discussed below to further understand energy-related factors. In addition, these factors also affect the thermal behavior of buildings, which consists of three stages: thermal needs, thermal gains and thermal losses of buildings. Heat pumps are systems that allow heat to be transferred from lower temperature rooms to higher temperature rooms based on the reverse Carnot cycle. Another system listed below is solar thermal systems, which are systems that harness solar energy to meet the thermal loads of a building.

The next stage of the paper is the introduction to the single-criteria and multi-criteria optimization problems. In addition, an analysis of Pareto optimal solutions in multi-criteria problems is performed, which involves the emergence of a set of optimal solutions. Moreover, the aim of the present work is mentioned, which was the multi-criteria optimization of the design of a solar thermal system assisted by a heat pump, for covering thermal loads. This particular optimization problem has two decision criteria:

1. minimising the power contribution of the heat pump, $Q_{HP,cons}$ [kWh]
2. maximisation of the average percentage of coverage of DHW loads for user satisfaction, DHW_{cov} [%].

These criteria were based on the appropriate selection of five design and operational parameters which are:

- hot water tank height
- flat solar collector surface
- type of flat solar collector
- water supply of the heat pump
- operating temperature limit of the heat pump

The device under consideration, which is a theoretical device, includes a flat solar collector, a heat pump, two storage tanks, three heat exchangers and two heat sinks. The equations governing these systems were derived from the TRNSYS 16 manual.

The best-case scenario methodology followed consists of five steps which were solved with the help of two open source software, Spyder and ALAMO. First, the simulation model was created in Spyder software. Then the control algorithm was recorded which consisted of two algorithms:

- the algorithm for adjusting the outlet flow from the collector
- the heat pump operating algorithm

Through the analysis of the simulation model, scenarios for the optimization of the design and operation parameters were derived. The results of these scenarios were used to create the monthly Pareto frontier curves, the monthly Surrogate Simulation Model, and the Surrogate Pareto frontier models. The last step of the methodology was to determine the desired rate of DHW load coverage by the solar collector, to calculate the monthly (optimal) values of heat pump power consumption through the surrogate Pareto bounds models. In addition, several optimization scenarios were generated through the surrogate simulation models and the optimal scenario was selected based on the minimum sum of squared relative errors (SSRE) of optimal scenarios. In the following, the results obtained from the solution of this methodology are presented for the problem of multi-criteria optimization of the design of a solar thermal system, assisted by a heat pump, to cover thermal loads.

Finally, the conclusions and some suggestions for future research that emerged from the present study are recorded.

Ευχαριστίες

Η παρούσα μελέτη πραγματοποιήθηκε στα πλαίσια της Διπλωματικής μου εργασίας υπό την επίβλεψη του Επίκουρου Καθηγητή Πανάρα Γεώργιου, του Τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Δυτικής Μακεδονίας. Προτού ξεκινήσει η παρουσίαση της μελέτης μου, θα ήθελα να ευχαριστήσω τους ανθρώπους που συνέβαλαν στην ολοκλήρωσή της.

Πρώτα από όλα, θα ήθελα να ευχαριστήσω θερμά το κο Πανάρα για τις πολύτιμες γνώσεις που μου μετέδωσε καθ' όλη τη διάρκεια εκπόνησης της παρούσας εργασίας, τη συνεχή επιστημονική του καθοδήγηση, την εμπιστοσύνη και τον χρόνο που διέθεσε.

Ακόμη τις θερμές μου ευχαριστίες στον Επίκουρο Καθηγητή Νικόλαο Πλόσκα, του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, για τη βοήθεια στο υπολογιστικό κομμάτι της εργασίας.

Ευχαριστώ θερμά τον υποψήφιο διδάκτορα Ζουλούμη Λεωνίδα για την άψογη συνεργασία και όλη τη βοήθεια που μου παρείχε κατά τη διεξαγωγή της διπλωματικής μου.

Τέλος, ένα μεγάλο ευχαριστώ στην οικογένεια μου και τους φίλους μου που βρίσκονται δίπλα μου σε κάθε βήμα της ακαδημαϊκής μου πορείας.

Περιεχόμενα

Περίληψη	ii
Abstract.....	iv
Ευχαριστίες	vi
1. Εισαγωγή.....	1
1.1 Η κατανάλωση ενέργειας στην σημερινή εποχή.....	1
1.2 Ανανεώσιμες πηγές ενέργειας.....	1
1.3 Ενέργεια και κτίρια	2
2. Θέρμανση-Ψύξη-Κλιματισμός κτιρίων.....	4
2.1 Βασικές έννοιες θέρμανσης και κλιματισμού.....	4
2.2 Θερμική συμπεριφορά κτιρίου.....	5
2.3 Αντλίες θερμότητας.....	8
2.3.1 Εισαγωγή	8
2.3.2 Αρχή λειτουργίας της αντλίας θερμότητας	8
2.3.3 Περιγραφή του θερμοδυναμικού κύκλου της αντλίας θερμότητας	9
2.3.4 Κύκλος Carnot και αντλία θερμότητας.....	10
2.3.5 Πλεονεκτήματα και μειονεκτήματα της αντλίας θερμότητας	12
2.3.6 Τύποι αντλίας θερμότητας	13
2.4 Ηλιακά θερμικά συστήματα	15
2.4.1 Γενικές Πληροφορίες.....	15
2.4.2 Περιγραφή λειτουργίας.....	15
2.4.3 Κατηγορίες ηλιακών θερμικών συστημάτων	16
2.4.4 Τεχνολογίες ηλιακών θερμικών συστημάτων	19
2.5 Εφαρμογές αντλιών θερμότητας σε συστήματα θέρμανσης ZNX.....	26
2.5.1 Σχετικά με τις διατάξεις ZNX	26
2.5.2 Διάταξεις με αυτόνομη αντλία θερμότητας	26
2.5.3 Διατάξεις με συνδυασμό αντλίας θερμότητας και ηλιακών συλλεκτών.....	27
3. Παραμετρική βελτιστοποίηση συστημάτων HVAC	28
3.1 Εισαγωγή στη βελτιστοποίηση	28

3.2 Παρουσίαση σχετικών ερευνών	28
3.3 Ανάλυση πολυκριτηριακής βελτιστοποίησης	29
3.3.1 Η έννοια των Pareto βέλτιστων λύσεων	29
3.3.2 Pareto βέλτιστες λύσεις	30
3.3.4 Βελτιστοποίηση υπό μελέτης διάταξης	31
4. Διάταξη θέρμανσης υπό εξέταση	33
4.1 Περιγραφή	33
4.2 Προσομοίωση της υπό μελέτη διάταξης	37
4.2.1 Εισαγωγή – Συνδεσμολογία διάταξης	37
4.2.2 Δεξαμενές αποθήκευσης	37
4.2.3 Επίπεδος ηλιακός συλλέκτης	38
4.2.4 Αντλία νερού-νερού (γεωθερμική)	42
4.2.5 Καταβόθρες Θερμότητας	44
4.2.6 Εναλλάκτες θερμότητας	45
4.2.7 Συνδεσμολογία εξεταζόμενης διάταξης	46
5. Μεθοδολογία βελτιστοποίησης διάταξης	48
5.1 Γενικά	48
5.2 Εισαγωγή στο Spyder–Anaconda	48
5.3 Εισαγωγή στο λογισμικό ALAMO (Automated Learning of Algebraic Models)	49
5.3 Περιγραφή μεθοδολογίας	50
6. Αποτελέσματα	55
6.1 Εξισώσεις υποκατάστατων μοντέλων προσομοίωσης	55
6.2 Καμπύλες Pareto	60
6.3 Υποκατάστατα μοντέλα καμπυλών Pareto	67
6.4 Παρουσίαση βέλτιστων λύσεων	68
7. Συμπεράσματα	70
Βιβλιογραφία	71
Παράρτημα Προγράμματα που χρησιμοποιήθηκαν στο λογισμικό Spyder	75
Π.1 Βιβλιοθήκη LoopTools	75

Π.2 Βιβλιοθήκη Elements	79
Π.3 Κατάστρωση εξισώσεων που διέπουν τη δεξαμενή αποθήκευσης	80
Π.4 Κατάστρωση εξισώσεων που διέπουν τον εναλλάκτη θερμότητας.....	83
Π.5 Κατάστρωση εξισώσεων που διέπουν τον επίπεδο ηλιακό συλλέκτη.....	85
Π.6 Κατάστρωση εξισώσεων που διέπουν τη γεωθερμική αντλία θερμότητας	87
Π.7 Κατάστρωση εξισώσεων που διέπουν την καταβόθρα θερμότητας.....	89
Π.8 Κατάστρωση εξισώσεων που διέπουν την τρίοδη βάνα εξαγωγής	90
Π.9 Κατάστρωση εξισώσεων που διέπουν την τρίοδη βάνα επιστροφής.....	91
Π.10 Κατάστρωση εξισώσεων που διέπουν τον κυκλοφορητή.....	92
Π.11 Εισαγωγή αρχείου με τις τιμές ηλιακής ακτινοβολίας	93
Π.12 Επιλογή ωριαίας τιμής ηλιακής ακτινοβολίας για μία μέρα	93
Π.13 Εισαγωγή αρχείου με τις τιμές της θερμοκρασίας περιβάλλοντος.....	94
Π.14 Επιλογή ωριαίας τιμής εξωτερικής θερμοκρασίας για μία μέρα.....	95
Π.15 Μετατροπή δευτερολέπτων μίας ημέρας σε ώρες.....	96
Π.16 Κατανάλωση ZNX	96
Π.17 Απομάστευση υγρού από την δεξαμενή των 500 lt.....	96
Π.18 Υπολογισμός αριθμού ημερών κάθε μήνα.....	97
Π.19 Επιλογή μηνιαίας θερμοκρασίας εδάφους.....	97
Π.20 Διατήρηση δύο σημαντικών ψηφίων.....	98
Π.21 Κατάστρωση του μοντέλου προσομοίωσης.....	98
Π.22 Δημιουργία σεναρίων βελτιστοποίησης για όλους τους μήνες.....	105
Π.23 Υπολογισμός συνολικής μηνιαίας κατανάλωσης ισχύος της αντλίας θερμότητας και του μέσου όρου του μηνιαίου ποσοστού κάλυψης φορτίων ZNX από τον ηλιακό συλλέκτη	107
Π.23 Τοποθέτηση των σεναρίων βελτιστοποίησης του κάθε μήνα με τη σειρά ημερών.....	108
Π.24 Δημιουργία αρχείων με τα μηνιαία σενάρια βελτιστοποίησης	109
Π.25 Εύρεση σημείων που σχηματίζουν το μέτωπο Pareto	111
Π.26 Δημιουργία μηνιαίων υποκατάστατων μοντέλων προσομοίωσης (SurrogateSimulationModel)	113
Π.27 Δημιουργία διαγράμματος με το όριο Pareto	116
Π.28 Δημιουργία μηνιαίων υποκατάστατων μοντέλων του ορίου Pareto (SurrogateParetofrontier) ...	117

Π.29 Δημιουργία σεναρίων βελτιστοποίησης	118
Π.30 Υπολογισμός ελάχιστου αθροίσματος σχετικών τετραγωνικών σφαλμάτων	119
Π.31 Εύρεση βέλτιστων λύσεων.....	119

1. Εισαγωγή

1.1 Η κατανάλωση ενέργειας στην σημερινή εποχή

Τα τελευταία χρόνια, η οικονομική ανάπτυξη και η βελτίωση του βιοτικού επιπέδου οδήγησαν στην αύξηση των ενεργειακών αναγκών μας, χωρίς όμως να υπάρχει σωστή διαχείρισή τους. Η χρήση της ενέργειας είναι απαραίτητη για την κάλυψη των βασικών αναγκών του ανθρώπου, όπως για τη θέρμανση και το κλιματισμό των κατοικιών, για τα μέσα μεταφοράς και για τα αυτοκίνητα. Επίσης, μεγάλα ποσά κατανάλωσης ενέργειας εντοπίζονται και σε βιομηχανίες, σε εργοστάσια και σε άλλες μεγάλες εγκαταστάσεις.

Ωστόσο, με την χρήση κατάλληλων μέσων παραγωγής ενέργειας μπορεί να επιτευχθεί σε μεγάλο βαθμό εξοικονόμηση ενέργειας. Η συνεχής αύξηση του κόστους ενέργειας κάνει όλο και πιο ελκυστική την εξοικονόμηση ενέργειας, πράγμα το οποίο επηρεάζει την τεχνολογική και οικονομική ανάπτυξη κάθε χώρας. Εκτός, από τον οικονομικό τομέα, η εξοικονόμηση ενέργειας είναι απαραίτητη για την προστασία του περιβάλλοντος. Η προστασία του περιβάλλοντος είναι ένα θέμα που απασχολεί όλες τις χώρες του κόσμου και αποτελεί την βασικότερη συνιστώσα στις δραστηριότητες του ανθρώπου, που αποσκοπούν στην βελτίωση της ποιότητας ζωής χωρίς να υπάρχει κίνδυνος καταστροφής του οικοσυστήματος. Σύμφωνα με τον Ευρωπαϊκό Οργανισμό Περιβάλλοντος, περίπου το 77% των ενεργειακών αναγκών των Ευρωπαίων πολιτών καλύπτεται από ορυκτά καύσιμα (πετρέλαιο, φυσικό αέριο, άνθρακα). Η πυρηνική ενέργεια καλύπτει το 14%, ενώ το υπόλοιπο 9% προέρχεται από ανανεώσιμες πηγές ενέργειας (ΕΟΠ 2021).

1.2 Ανανεώσιμες πηγές ενέργειας

Οι Ανανεώσιμες Πηγές Ενέργειας (ΑΠΕ) είναι πηγές οι οποίες δεν εξαντλούνται ή μπορούν να αναπληρώνονται με την πάροδο του χρόνου, και αυτή είναι και η βασική τους διαφορά με τα ορυκτά καύσιμα, τα οποία απαιτούν εκατομμύρια χρόνια για τον σχηματισμό τους. Οι μορφές ΑΠΕ που αναπτύσσονται είναι η ηλιακή ενέργεια (ηλιακά θερμικά και φωτοβολταϊκά συστήματα), η αιολική ενέργεια (ανεμογεννήτριες), η υδροηλεκτρική ενέργεια (υδροηλεκτρικοί σταθμοί), η γεωθερμική ενέργεια (γεωθερμικές αντλίες θερμότητας), η ενέργεια από βιομάζα και η ενέργεια των ωκεανών. Οι ΑΠΕ χρησιμοποιούνται κυρίως για την ηλεκτροπαραγωγή, για την παραγωγή μηχανικής ενέργειας, αλλά και για την θέρμανση ή την ψύξη.

Τα κύρια πλεονεκτήματα των Ανανεώσιμων Πηγών Ενέργειας είναι:

- ανεξάντλητες πηγές ενέργειας
- φιλικές προς το περιβάλλον
- έχουν χαμηλό λειτουργικό κόστος που δεν επηρεάζεται από τις διακυμάνσεις της διεθνούς οικονομίας
- η ανάπτυξη τους δημιουργεί σημαντικό αριθμό νέων θέσεων εργασίας

- μπορούν να βοηθήσουν την ενεργειακή αυτάρκεια μικρών και αναπτυσσόμενων χωρών
- οι περισσότερες κυβερνήσεις προωθούν την χρήση τους μέσω επιχορηγήσεων

Βέβαια οι ΑΠΕ διαθέτουν και κάποια μειονεκτήματα, μερικά εκ' των οποίων είναι ότι έχουν μικρό συντελεστή απόδοσης, της τάξης του 30% ή και χαμηλότερο. Επίσης, απαιτείται μεγάλο αρχικό κόστος εγκατάστασης καθώς και διάθεση μεγάλης επιφάνειας γης. Επιπλέον, η παροχή και η απόδοση της αιολικής, ηλιακής και υδροηλεκτρικής ενέργειας εξαρτάται από την εποχή του έτους, την τοποθεσία και το κλίμα της περιοχής στην οποία εγκαθίστανται. Εντούτοις, η Ευρωπαϊκή Ένωση (Ε.Ε.) έχει θέσει ως στόχο έως το 2030 ένα μερίδιο 32% της κατανάλωσης ενέργειας της Ε.Ε. να προέρχεται από ανανεώσιμες πηγές. Για αυτό, για να ενθαρρύνει την χρήση των ΑΠΕ, δίνει επιχορηγήσεις με σκοπό να παροτρύνει τα κράτη μέλη της (Ciucci 2020).

1.3 Ενέργεια και κτίρια

Ο κτιριακός τομέας είναι υπεύθυνος για το 40% περίπου της συνολικής κατανάλωσης ενέργειας σε εθνικό και ευρωπαϊκό επίπεδο. Η κατανάλωση ενέργειας στα κτίρια παρουσιάζει αυξητική τάση, λόγω της αύξησης της χρήσης κλιματιστικών και μικροσυσκευών. Η ενέργεια αυτή που καταναλώνεται στα κτίρια οφείλεται στη θέρμανση, την ψύξη (θερινό κλιματισμό), τον φωτισμό και άλλες μικροσυσκευές (Ε. Λάζαρη 2002).

Η εξοικονόμηση ενέργειας στα κτίρια μπορεί να επιτευχθεί μέσω του βιοκλιματικού σχεδιασμού. Ο βιοκλιματικός σχεδιασμός ενός κτιρίου είναι ο σχεδιασμός ο οποίος λαμβάνει υπόψη το κλίμα κάθε περιοχής και στοχεύει στην εξασφάλιση των απαραίτητων εσωκλιματικών συνθηκών (θερμική και οπτική άνεση, ποιότητα αέρα) με την ελάχιστη δυνατή κατανάλωση ενέργειας. Αυτό εξασφαλίζεται, αξιοποιώντας όλες τις διαθέσιμες πηγές από το περιβάλλον όπως είναι ο άνεμος, ο ήλιος, το νερό και το έδαφος. Κάποιες τεχνικές του βιοκλιματικού σχεδιασμού αποτελούν η θερμική προστασία του κελύφους, τα παθητικά ηλιακά συστήματα, οι τεχνικές και τα συστήματα φυσικού δροσισμού και φυσικού φωτισμού και ορισμένες τεχνικές ορθολογικής χρήσης ενέργειας (θερμικές ζώνες, αποθήκευση θερμότητας στα δομικά στοιχεία του κτιρίου) (Ε. Λάζαρη 2002).

Άλλοι τρόποι μείωσης της κατανάλωσης ενέργειας είναι η χρήση πιο αποδοτικών συστημάτων θέρμανσης και κλιματισμού, η σωστή συντήρηση τους και πάνω από όλα με την σωστή λειτουργία-συμπεριφορά των χρηστών.

Στην συνέχεια αναφέρονται μερικοί λόγοι για τους οποίους είναι εφικτό και χρειάζεται να γίνει μεγάλη εξοικονόμηση ενέργειας στην θέρμανση και στον κλιματισμό των κτιρίων:

- Περισσότερα από 10 εκατομμύρια οικιακοί λέβητες που είναι εγκατεστημένοι σε χώρες της Ε.Ε. είναι παλαιότεροι των 20 ετών (Τζανακάκη 2008).
- Η μεγάλη πλειοψηφία των κτιρίων στην Ελλάδα (περίπου το 80% του συνόλου) είναι κατασκευασμένα πριν το 1980 και δεν είναι καλά θερμομονωμένα. Για αυτό τον λόγο απαιτούν μεγάλα ποσά ενέργειας για να εξασφαλίσουν τις αποδεκτές συνθήκες άνεσης, ιδίως το χειμώνα, και προσφέρουν αρκετές δυνατότητες για εφαρμογή μέτρων εξοικονόμησης ενέργειας.

- Η κατά κανόνα μέτρια κατάσταση των παλαιών συστημάτων που οδηγεί σε μειωμένους βαθμούς απόδοσης και επομένως σε αυξημένη κατανάλωση ενέργειας και περιβαλλοντική επιβάρυνση. Η συντήρηση των συστημάτων αυτών και η εφαρμογή συστημάτων ελέγχου και αυτοματισμών προσφέρουν πολλές δυνατότητες για εξοικονόμηση ενέργειας.
- Η συνεχής αύξηση, τόσο σε αριθμό όσο και σε εγκατεστημένη ισχύ, των συστημάτων και συσκευών που καταναλώνουν ηλεκτρική ενέργεια για θερινό κλιματισμό. Αυτό αφορά τα κτίρια κατοικιών, κυρίως όμως τα κτίρια γραφείων, καταστημάτων και υπηρεσιών. Με την εφαρμογή συστημάτων ανάκτησης θερμότητας, με την συμπαραγωγή ηλεκτρισμού και θερμότητας για εφαρμογές συστημάτων ψύξης με απορρόφηση και με την χρήση αντλιών θερμότητας (κυρίως γεωθερμικών), οι δυνατότητες εξοικονόμησης ενέργειας είναι μεγάλες, αλλά περιορίζεται συγχρόνως και η κατανάλωση ηλεκτρικής ενέργειας (Παπακώστας 2015).

2. Θέρμανση-Ψύξη-Κλιματισμός κτιρίων

Σε αυτό το κεφαλαίο, αρχικά παρουσιάζονται κάποιες βασικές έννοιες θερμότητας και πως αυτές συνδέονται με τη θέρμανση, τη ψύξη και τον κλιματισμό των κτιρίων. Στην συνέχεια, γίνεται περιγραφή των αντλιών θερμότητας και παρουσιάζονται ο τρόπος λειτουργίας τους, οι διάφοροι τύποι τους και εξηγείται γιατί είναι σημαντική η χρήση τους σε σχέση με άλλα συστήματα θέρμανσης και κλιματισμού. Τέλος, περιγράφονται και τα ηλιακά θερμικά συστήματα ως προς τον ορισμό τους αλλά και ως προς τα εξαρτήματα που διαθέτουν και τη λειτουργία τους.

2.1 Βασικές έννοιες θέρμανσης και κλιματισμού

Θερμότητα: είναι η μορφή ενέργειας η οποία μεταφέρεται από ένα σώμα σε ένα άλλο λόγω της διαφοράς θερμοκρασίας τους. Σύμφωνα με το 2^ο Θερμοδυναμικό Αξίωμα, η ροή θερμότητας γίνεται από το θερμότερο σώμα στο ψυχρότερο, έως ότου εξισωθούν οι θερμοκρασίες των δύο σωμάτων. Η μονάδα μέτρησης της θερμότητας στο Διεθνές Σύστημα Μονάδων είναι το Joule (J). Επειδή η μονάδα Joule είναι σχετικά μικρή για να εκφράσει τα τυπικά ποσά θερμότητας, που εμπλέκονται στα προβλήματα καθημερινής πρακτικής, χρησιμοποιούμε συνήθως την μονά kiloJoule (1 kJ = 10³J). Άλλες μονάδες μέτρησης είναι το British thermal unit (Btu) στο αγγλοσαξονικό σύστημα και η θερμίδα (cal) ή χλιοθερμίδα (kcal) (Πάπιστα 2020). Η σχέση των παραπάνω μονάδων συνδέονται ως εξής:

$$1 \text{ kcal} = 1000 \text{ cal} = 4186,8 \text{ Joule} = 3,97 \text{ Btu}$$

Η μετάδοση θερμότητας μπορεί να γίνει με τρεις τρόπους: με αγωγή, συναγωγή και με ακτινοβολία

➤ Μεταφορά θερμότητας με αγωγή

Η μεταφορά θερμότητας με αγωγή είναι η ανταλλαγή ενέργειας, μέσω ενός στερεού σώματος ή διαμέσου σωμάτων τα οποία βρίσκονται σε επαφή, από το σημείο επαφής. Στα στερεά σώματα γίνεται είτε με μετανάστευση ελεύθερων ηλεκτρονίων είτε με ταλαντώσεις του κρυσταλλικού πλέγματος (δονήσεις των ατόμων και των μορίων). Στα ρευστά και στα αέρια η αγωγή οφείλεται στην άτακτη κίνηση των σωματιδίων (Πάπιστα 2020). Η ροή θερμότητας q [W/m²] με αγωγή δίνεται από την σχέση:

$$q = k \cdot \frac{T_1 - T_2}{L} \quad (2.1)$$

όπου T_1, T_2 : οι θερμοκρασίες σε δύο διαφορετικά σημεία του σώματος [K]

k : συντελεστής θερμικής αγωγιμότητας [W/(m·K)], ο οποίος εξαρτάται από τις ιδιότητες του σώματος

L : το πάχος στρώματος μεταξύ των δύο σημείων [m]

➤ Μεταφορά θερμότητας με συναγωγή

Η μεταφορά θερμότητας με συναγωγή είναι η μεταφορά ενέργειας μέσω της κίνησης ενός ρευστού ή αερίου όταν έρχεται σε επαφή με την επιφάνεια κάποιου στερεού. Οι μορφές συναγωγής είναι πέντε και είναι η εξαναγκασμένη, η φυσική, η ελεύθερη, η μικτή (φυσική και εξαναγκασμένη) και ο βρασμός και η συμπύκνωση (Πάπιστα 2020). Η ροή θερμότητας q [W/m²] με συναγωγή δίνεται από την σχέση:

$$q = h \cdot (T_s - T_\infty) \quad (2.2)$$

όπου h : συντελεστής μεταφοράς θερμότητας με συναγωγή [$W/(m \cdot K)$], ο οποίος εξαρτάται από την γεωμετρία της επιφάνειας, το είδος του ρευστού και της ροής του

T_s : η θερμοκρασία στην επιφάνεια του στερεού σώματος [K]

T_∞ : η θερμοκρασία του ρευστού ή αερίου σε σημείο που δεν επηρεάζεται πλέον από την θερμοκρασία του στερεού [K]

➤ Μεταφορά θερμότητας με ακτινοβολία

Η μεταφορά θερμότητας με ακτινοβολία είναι η ανταλλαγή ενέργειας μέσω ηλεκτρομαγνητικής ακτινοβολίας και απορρόφησης. Όλα τα σώματα με θερμοκρασία πάνω από το απόλυτο μηδέν εκπέμπουν θερμική ενέργεια με την μορφή ηλεκτρομαγνητικής ακτινοβολίας (δεν απαιτείται η ύπαρξη υλικού μέσου για την μεταφορά). Η μεταφορά θερμότητας λόγω ακτινοβολίας μπορεί επίσης να περιλαμβάνει την απορρόφηση ακτινοβολίας από τον περιβάλλοντα χώρο (ακτινοβολήση (irradiation), G) καθώς και συναγωγή (αν $T_s \neq T_\infty$) (Πάπιστα 2020). Η ροή θερμότητας q [W/m^2] με ακτινοβολία δίνεται από την σχέση:

$$q = \varepsilon \cdot \sigma \cdot T_s^4 \quad (2.3)$$

όπου ε : ικανότητα εκπομπής του σώματος ($0 < \varepsilon < 1$)

σ : σταθερά Stefan-Boltzmann = $5,67 \cdot 10^{-8}$ [$W/(m^2 \cdot K^4)$]

T_s : η θερμοκρασία του σώματος [K]

Θερμοκρασία: Η θερμοκρασία είναι το φυσικό μέγεθος που μετρά την μέση κινητική ενέργεια από μεταφορά, ταλάντωση ή περιστροφή των δομικών λίθων (άτομα, μόρια, ιόντα) της ύλης. Μονάδα μέτρησης της θερμοκρασίας στο Διεθνές Σύστημα Μονάδων είναι το Kelvin (K). Άλλες μονάδες μέτρησης είναι οι βαθμοί Celsius ($^{\circ}C$) και οι βαθμοί Fahrenheit ($^{\circ}F$). Η σχέση που συνδέει τις μονάδες των θερμοκρασιών είναι η εξής: $0 \text{ K} = -273.15 \text{ }^{\circ}C = -459.67 \text{ }^{\circ}F$ (Wikipedia 2020).

Θερμοπερατότητα: Η θερμοπερατότητα είναι η ποσότητα θερμότητας ανά μονάδα χρόνου που περνά μέσα από 1 m^2 στοιχείου κατασκευής με πάχος d (m) όταν η διαφορά θερμοκρασίας μεταξύ των επιφανειών αυτών είναι ίση με 1 K . Συμβολίζεται με U και η μονάδα μέτρησης της είναι [$W/(m^2 \cdot K)$] (Wikipedia 2020).

2.2 Θερμική συμπεριφορά κτιρίου

Σε αυτή την υποενότητα παρουσιάζονται οι παράγοντες που επηρεάζουν την θερμοκρασία εντός του κτιρίου και που προκαλούν ροή θερμότητας από και προς το εσωτερικό του:

➤ **Θερμικές ανάγκες κτιρίου**

Οι θερμικές ανάγκες ενός χώρου και γενικότερα ενός κτιρίου είναι το ποσό θερμότητας που πρέπει να ληφθεί ως βάση για τον σχεδιασμό της εγκατάστασης του συστήματος θέρμανσης και ψύξης. Οι θερμικές ανάγκες είναι ιδιότητα του κτιρίου και είναι ανεξάρτητες από το σύστημα θέρμανσης ή ψύξης που θα εγκατασταθεί. Οι παράγοντες που επηρεάζουν τις θερμικές ανάγκες ενός κτιρίου είναι (Κουρουπέτογλου 2019):

- το μέγεθος του χώρου
- ο τρόπος κατασκευής των τοίχων
- το μέγεθος και το υλικό των ανοιγμάτων
- ο αερισμός του χώρου.

➤ **Θερμικά κέρδη κτιρίου**

Τα θερμικά κέρδη είναι η θερμότητα που παράγεται ή εισέρχεται μέσα στο χώρο (McQuiston and Parker 2003).

Οι κατηγορίες των θερμικών κερδών είναι:

- Ηλιακά θερμικά κέρδη (θερμότητα που παρέχεται από την ηλιακή ακτινοβολία η οποία εισέρχεται από τους υαλοπίνακες ή αδιαφανείς τοίχους και οροφές)
- Εσωτερικά θερμικά κέρδη (θερμότητα που προσδίδεται από τους χρήστες και από συσκευές).

➤ **Θερμικές απώλειες κτιρίου**

Οι θερμικές απώλειες αναφέρονται στην ισχύ/ενέργεια που ο χώρος ανταλλάσσει με το περιβάλλον, και πρέπει να αναπληρώσουμε ή να αφαιρέσουμε ώστε να διατηρεί σταθερές τις συνθήκες ανέσεως (κυρίως θερμοκρασία αλλά και υγρασία αν υπάρχει δυνατότητα ρύθμισής της και καθαρότητα του αέρα) στο εσωτερικό του (Παντελίδης 2015).

Οι θερμικές απώλειες αποτελούνται κυρίως από δύο κατηγορίες:

- Θερμικές απώλειες από το περίβλημα (τοίχοι, δάπεδα, οροφές, ανοίγματα)
- Θερμικές απώλειες εισόδου αέρα από χαραμάδες (πόρτες, παράθυρα, τζάκια, εξαεριστήρες κλπ.)



Εικόνα 2.1 Θερμικές απώλειες και θερμικά κέρδη του κτιρίου. Πηγή: (Πανάρας 2020)

Οι απώλειες θερμότητας δια μέσου των δομικών στοιχείων ενός κτιρίου υπολογίζονται από την σχέση (Παντελίδης 2015):

$$Q = U \cdot A \cdot (T_{in} - T_{out}) \quad (2.4)$$

όπου U: ο συντελεστής θερμοπερατότητας του δομικού στοιχείου [W/(m²·K)]

A: η επιφάνεια του δομικού στοιχείου [m²]

T_{in}: η θερμοκρασία του εσωτερικού χώρου [K]

T_{out}: η θερμοκρασία του περιβάλλοντος [K]

Όπως φαίνεται από την παραπάνω σχέση, οι απώλειες θερμότητας εξαρτώνται από τον συντελεστή θερμοπερατότητας και για αυτό τον λόγο πρέπει να υπολογίζονται για κάθε δομικό στοιχείο ξεχωριστά. Για ένα δομικό στοιχείο που αποτελείται από N στρώματα πάχους L_i το καθένα με αγωγιμότητα k_i, ο συντελεστής U δίνεται από την σχέση (Παντελίδης 2015):

$$U = \frac{1}{\frac{1}{h_{in}} + \sum_{i=1}^N \frac{L_i}{k_i} + \frac{1}{h_{out}}} \quad (2.5)$$

Ο συντελεστής θερμοπερατότητας εξαρτάται από το πάχος και το συντελεστή θερμικής αγωγιμότητας του δομικού στοιχείου αλλά και από την επιφάνειά του. Οι θερμικές απώλειες είναι αδύνατο να εμποδιστούν τελείως. Μπορούν να περιοριστούν ως προς την ένταση και τη διάρκεια τους. Αυτό επιτυγχάνεται με τοποθέτηση κατάλληλης θερμομόνωσης στα εξωτερικά δομικά στοιχεία του κτιρίου, η οποία μειώνει τον ρυθμό ροής θερμότητας από και προς το κτίριο. Τα κτίρια με καλή θερμομόνωση έχουν μικρό συντελεστή θερμοπερατότητας (δηλαδή μικρές θερμικές απώλειες), ενώ κτίρια με κακή θερμομόνωση έχουν μεγάλο συντελεστή θερμοπερατότητας (δηλαδή υψηλές θερμικές απώλειες).

2.3 Αντλίες θερμότητας

2.3.1 Εισαγωγή

Σε αυτή την ενότητα περιγράφεται λεπτομερώς η λειτουργία και ο θερμοδυναμικός κύκλος της αντλίας θερμότητας. Επιπλέον, παρουσιάζονται τα πλεονεκτήματα της, οι διάφοροι τύποι της, και αναφέρονται διάφορες εφαρμογές της και συστήματα με τα οποία είναι εφικτό να συνδυαστεί.

2.3.2 Αρχή λειτουργίας της αντλίας θερμότητας

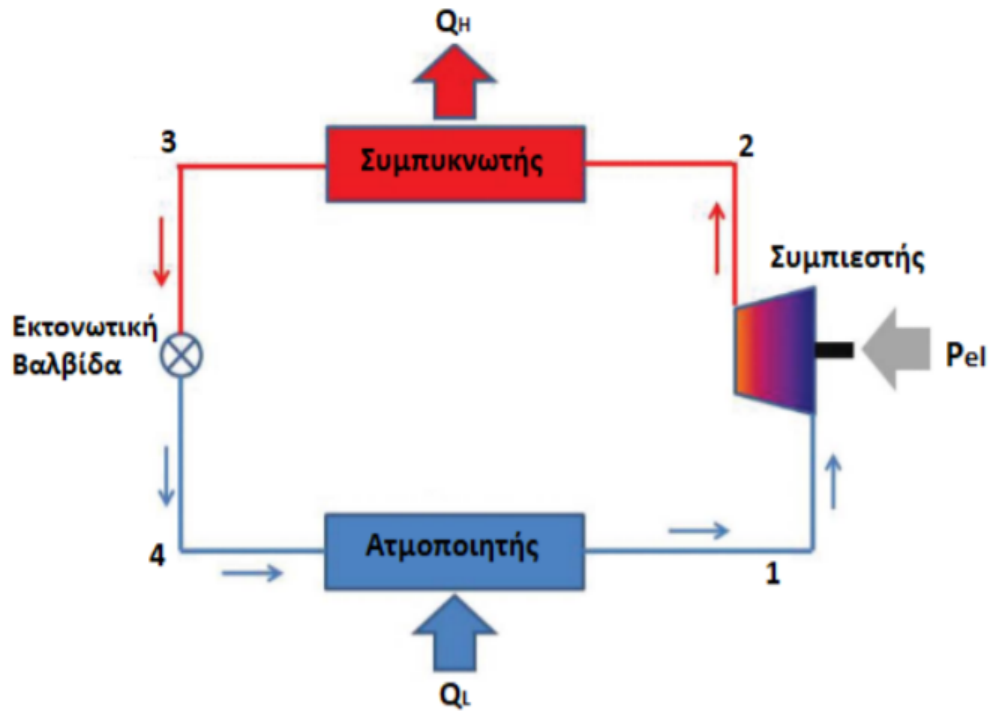
Σύμφωνα με τον 2^ο Θερμοδυναμικό Αξίωμα και την εξίσωση (2.6), σε όλες τις αυθόρμητες (μη αντιστρεπτές) διεργασίες η εντροπία αυξάνεται, δηλαδή σε ένα σύστημα η θερμότητα μεταφέρεται φυσικά από ένα σώμα υψηλότερης θερμοκρασίας σε ένα σώμα χαμηλότερης θερμοκρασίας (McQuiston and Parker 2003).

Για την αντίστροφη διεργασία απαιτείται η κατανάλωση κάποιας μορφής έργου (ενέργειας). Επομένως, δίνοντας κάποιο ποσό ενέργειας (μηχανικό έργο) στην αντλία θερμότητας μπορούμε να έχουμε την αντίστροφη διεργασία, δηλαδή η αντλία θερμότητας μας επιτρέπει να έχουμε μεταφορά ενέργειας από ένα χώρο χαμηλότερης θερμοκρασίας σε ένα χώρο υψηλότερης θερμοκρασίας. Έτσι για παράδειγμα, κατά την χειμερινή περίοδο η αντλία θερμότητας αφαιρεί (αντλεί) ενέργεια από το περιβάλλον (χαμηλή θερμοκρασία) και την διοχετεύει στο εσωτερικό του κτιρίου (υψηλότερη θερμοκρασία) προκειμένου να αυξήσει την ενέργεια του εσωτερικού χώρου, ενώ κατά την θερινή περίοδο συμβαίνει το αντίθετο. Βάσει της θερμοδυναμικής τα πιο πάνω επιβεβαιώνονται από την εξίσωση της εντροπίας (McQuiston and Parker 2003):

$$\Delta S = S_A - S_B = \int_A^B \frac{dq}{T} \quad (2.6)$$

Τα βασικά μέρη της αντλίας θερμότητας είναι τέσσερα (εικόνα 2.2):

- ατμοποιητής
- συμπιεστής
- συμπυκνωτής
- βαλβίδα στραγγαλισμού

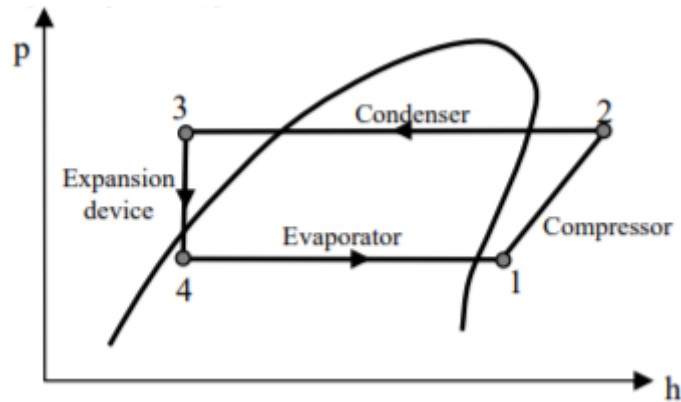


Εικόνα 2.2 Διάταξη αντλίας θερμότητας. Πηγή: (Crall 2015)

Η λειτουργία μιας αντλίας θερμότητας βασίζεται στη φυσική ιδιότητα, όπου το σημείο βρασμού ενός υγρού αυξάνεται με τη πίεση. Με τη μείωση της πίεσης, ένα μέσο μπορεί να ατμοποιηθεί σε χαμηλές θερμοκρασίες (κατά λειτουργία ατμοποιητή) ενώ με τη αύξηση της πίεσης θα οδηγήσει σε υψηλό σημείο βρασμού (κατά τη λειτουργία του συμπυκνωτή). Αυτό, γίνεται πιο κατανοητό στη περιγραφή του κύκλου της αντλίας θερμότητας (McQuiston and Parker 2003).

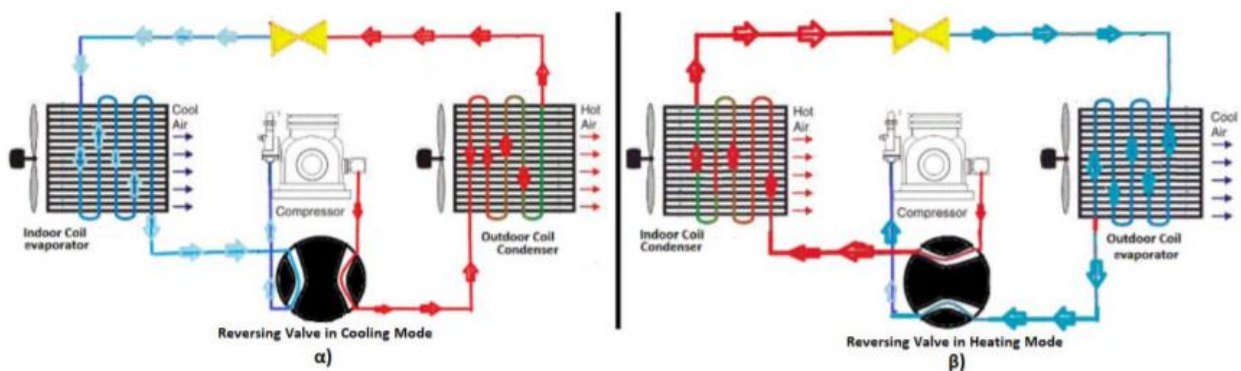
2.3.3 Περιγραφή του θερμοδυναμικού κύκλου της αντλίας θερμότητας

Αρχικά, περιγράφεται ο θερμοδυναμικός κύκλος της αντλίας θερμότητας για θέρμανση. Το εργαζόμενο μέσο που βρίσκεται σε χαμηλή πίεση και θερμοκρασία (κατάσταση 4), ατμοποιείται στον ατμοποιητή (εξατμιστή) και γίνεται ατμός (κατάσταση 1). Στη διεργασία 4→1 απαιτείται πρόσδοση θερμότητας (Q_L). Στην συνέχεια, ο συμπιεστής συμπιέζει το εργαζόμενο μέσο αυξάνοντας την θερμοκρασία του και την πίεση του και το οδηγεί μέσα στον συμπυκνωτή (κατάσταση 2) για να συμπυκνωθεί. Στη διεργασία 1→2 απαιτείται και πρόσδοση ηλεκτρικής ενέργειας. Κατά την συμπύκνωση του εργαζόμενου μέσου σε υψηλή πίεση και θερμοκρασία απελευθερώνεται θερμότητα. Τέλος, το εργαζόμενο μέσο οδηγείται στην εκτονωτική βαλβίδα που μειώνεται η πίεση του και η θερμοκρασία του και μετατρέπεται σε ένα μείγμα από υγρό και ατμό (κατάσταση 4). Έτσι ολοκληρώνεται ο θερμοδυναμικός κύκλος (Crall 2015).



Διάγραμμα 2.1 Θερμοδυναμικός κύκλος αντλίας θερμότητας. Πηγή: (Bolaji, Akintunde and Falade 2011)

Ένα βασικό πλεονέκτημα της αντλίας θερμότητας είναι ότι μπορεί να παρέχει το χειμώνα θέρμανση και παράλληλα η ίδια μονάδα το καλοκαίρι μπορεί να αναστρέφει την λειτουργία των συστημάτων της (συμπυκνωτής, ατμοποιητής) και να παρέχει ψύξη. Επειδή όμως, στον συμπιεστή η αναρρόφηση και η κατάθλιψη είναι συγκεκριμένες, δηλαδή το ρευστό δεν μπορεί να περάσει ανάποδα από μέσα του η αναστροφή επιτυγχάνεται με την βοήθεια μιας ειδικής τετράοδης βαλβίδας που είναι ενσωματωμένη στην αντλία θερμότητας. Έτσι, κατά την λειτουργία της ψύξης (π.χ. το καλοκαίρι), η θερμότητα αναρροφάται από τον χώρο που απαιτείται να κλιματιστεί και αποδίδεται σε μια εξωτερική πηγή θερμότητας (νερό, αέρας, γη), ενώ κατά την λειτουργία της θέρμανσης (π.χ. το χειμώνα) η θερμότητα αναρροφάται από μία εξωτερική πηγή θερμότητας (αέρας, νερό, γη) και αποδίδεται στον χώρο που απαιτείται να θερμανθεί (Evatt 2015).

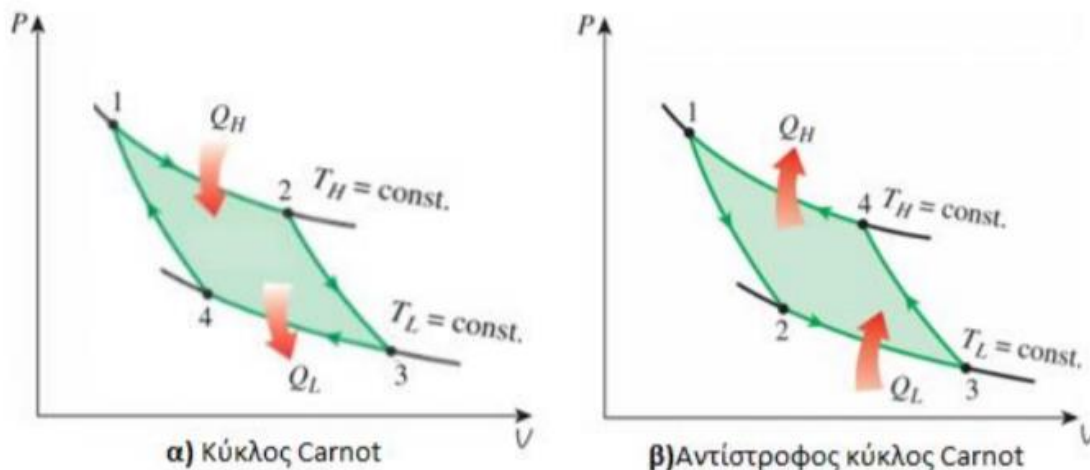


Εικόνα 2.3 α) Λειτουργία της μονάδας σε Ψύξη **β)** Λειτουργία της μονάδας σε Θέρμανση. Πηγή: (Evatt 2015)

2.3.4 Κύκλος Carnot και αντλία θερμότητας

Η αντλία θερμότητας είναι μία θερμική μηχανή που λειτουργεί όπως ο κύκλος Carnot αλλά με την αντίθετη φορά. Η λειτουργία του αντίστροφου κύκλου Carnot αποτελείται από τέσσερις μεταβολές,

δύο ισόθερμες και δύο αδιαβατικές (ισεντροπικές), που εκτελούνται μεταξύ μιας θερμής δεξαμενής με θερμοκρασία T_H και μιας ψυχρής δεξαμενής με θερμοκρασία T_L (Μανωλάκος 2020).



Διάγραμμα 2.2 α) Κύκλος Carnot **β)** Αντίστροφος κύκλος Carnot. Πηγή: (Μανωλάκος 2020)

Οι διεργασίες του αντίστροφου κύκλου Carnot έχουν ως εξής:

Διεργασία **1→2**: το εργαζόμενο μέσο υψηλής θερμοκρασίας T_H εκτονώνεται ισεντροπικά και οδηγείται σε χαμηλή θερμοκρασία T_L .

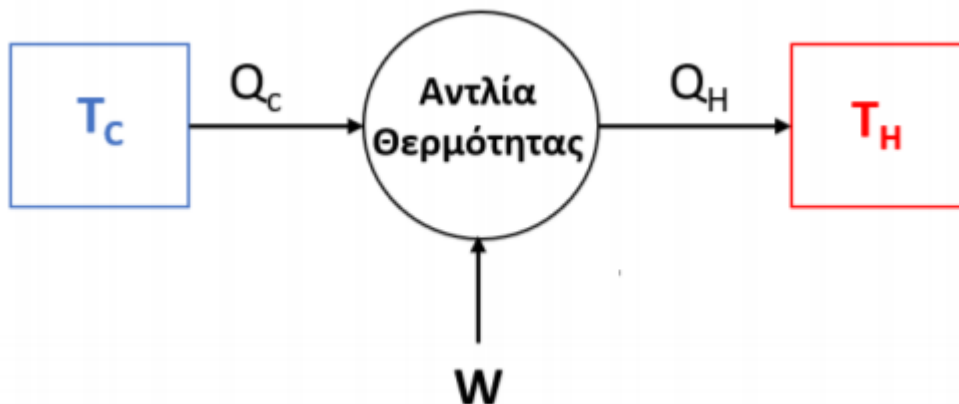
Διεργασία **2→3**: ισοθερμική πρόσδοση ποσού θερμότητας Q_L στο εργαζόμενο μέσο

Διεργασία **3→4**: ισεντροπική συμπίεση μέχρι την θερμοκρασία T_H

Διεργασία **4→1**: ισοθερμική απόρριψη ποσού θερμότητας Q_C .

Σε ιδανικές συνθήκες ο βαθμός απόδοσης του κύκλου Carnot δίνεται από την σχέση:

$$\eta_{\text{Carnot}} = 1 - \frac{T_H}{T_C} \tag{2.7}$$



Εικόνα 2.4 Λειτουργία αντλίας θερμότητας. Πηγή: (Μανωλάκος 2020)

Εφαρμόζοντας το 1^ο Θερμοδυναμικό Αξίωμα (αρχή διατήρησης της ενέργειας) γύρω από μία αντλία θερμότητας έχουμε:

$$Q_C + W = Q_H \quad (2.8)$$

Έτσι ο αντίστοιχος θεωρητικός βαθμός απόδοσης της αντλίας θερμότητας ορίζεται ως ο συντελεστής συμπεριφοράς COP (Coefficient to Performance) και δίνεται από την σχέση:

$$\text{COP}_{th} = \frac{Q_H}{W} = \frac{Q_H}{Q_H - Q_C} = \frac{T_H}{T_H - T_C} = 1 - \frac{T_H}{T_C} \quad (2.9)$$

Στην περίπτωση της αντλίας θερμότητας ο συντελεστής συμπεριφοράς COP ορίζεται ως ο λόγος της θερμικής ισχύος (Q_H) που αποδίδει η αντλία θερμότητας προς την ηλεκτρική ισχύ (Q_{el}) που καταναλώνει:

$$\text{COP} = \frac{Q_H}{Q_{el}} \quad (2.10)$$

Από την πιο πάνω σχέση γίνεται κατανοητό ότι όσο μεγαλύτερος είναι ο συντελεστής COP, τόσο πιο αποδοτική είναι η αντλία θερμότητας. Αξίζει να σημειωθεί ότι ο συντελεστής COP κατά τη λειτουργία της θέρμανσης, μειώνεται όσο μειώνεται η εξωτερική θερμοκρασία περιβάλλοντος.

Τα αντίστοιχα ισχύουν και κατά την λειτουργία της ψύξης. Το αντίστοιχο μέγεθος του συντελεστή COP είναι ο συντελεστής EER (Energy Efficiency Ratio), ο οποίος δίνεται από την σχέση:

$$\text{EER} = \frac{Q_C}{Q_{el}} \quad (2.11)$$

όπου Q_C : η ψυκτική ισχύς της αντλίας θερμότητας [W]

2.3.5 Πλεονεκτήματα και μειονεκτήματα της αντλίας θερμότητας

Η αντλία θερμότητας χαρακτηρίζεται από πολλά πλεονεκτήματα μερικά εκ' των οποίων είναι (EN.E.EΠΠ.Θ.Ε. 2020), (TheRenewableEnergyHub 2020):

- **Χαμηλό κόστος λειτουργίας**

Όπως αναφέρθηκε και πιο πάνω, μία αντλία θερμότητας χρησιμοποιεί ως κύρια πηγή το περιβάλλον για να θερμάνει ή να ψύξει έναν χώρο. Η χρήση ηλεκτρικής ενέργειας απαιτείται μόνο στο στάδιο της άντλησης θερμότητας και ο χρήστης καλείται να πληρώσει μόνο το κόστος της ηλεκτρικής ενέργειας που απαιτείται κατά την συμπίεση του εργαζόμενου μέσου.

- **Χαμηλό κόστος συντήρησης και υψηλή ασφάλεια λειτουργίας**

Είναι συστήματα που δεν χρησιμοποιούν ορυκτά καύσιμα και δεν βασίζονται στην καύση αυτών (π.χ. λέβητες).

- **Μεγάλη διάρκεια ζωής**

Η διάρκεια ζωής μίας αντλίας θερμότητας υπολογίζεται έως 50 χρόνια. Αυτό έχει ως αποτέλεσμα ότι θεωρείται εξαιρετικά αξιόπιστη και μόνιμη πηγή θερμότητας.

- **Εύκολη εγκατάσταση και μικρό μέγεθος**

Όλες οι αντλίες θερμότητας εκτός από τις γεωθερμικές μπορούν να τοποθετηθούν σε οποιοδήποτε λεβητοστάσιο και έχουν επιπλέον κέρδος χώρου εξαιτίας της έλλειψης δεξαμενών αποθήκευσης καυσίμου και καμινάδας απαγωγής των καυσαερίων όπως τα συμβατικά συστήματα θέρμανσης.

- **Συνδυασμός με αποδοτικές τερματικές μονάδες**

Για παράδειγμα, μπορεί να συνδυαστεί με τερματικά σώματα τύπου Fan Coils ή με ενδοδαπέδιο ή επιτοίχιο σύστημα. Επιπλέον, μπορεί να παράγει ζεστό νερό χρήσης καθ' όλη την διάρκεια του έτους.

Εκτός από τα πλεονεκτήματα, η αντλία θερμότητας εμφανίζει και κάποια μειονεκτήματα (TheRenewableEnergyHub 2020):

- **Υψηλό αρχικό κόστος**

Συγκριτικά με τα συμβατικά συστήματα θέρμανσης διαθέτουν υψηλότερο αρχικό κόστος, το οποίο αποσβένεται μετά από λίγα χρόνια λειτουργίας.

- **Απαιτείται έρευνα πριν την εγκατάσταση**

Πριν από την εγκατάσταση μιας αντλίας θερμότητας απαιτείται έρευνα για την κατανόηση της κίνησης της θερμότητας, της τοπικής γεωλογίας καθώς και τις απαιτήσεις θέρμανσης και ψύξης του κτιρίου.

- **Οριοθέτηση θερμοκρασίας λειτουργίας**

Οι αντλίες θερμότητας που χρησιμοποιούν ως πηγή το περιβάλλον δεν μπορούν να δουλέψουν σε χαμηλές θερμοκρασίες.

- **Απαιτείται ηλεκτρική ενέργεια για την λειτουργία**

Όπως έχει προαναφερθεί, για την λειτουργία μιας αντλίας θερμότητας απαιτείται η χρήση και ενός ποσού ηλεκτρικής ενέργειας πράγμα που σημαίνει ότι αυτό το σύστημα δεν είναι εντελώς ουδέτερο ως προς τον άνθρακα.

2.3.6 Τύποι αντλίας θερμότητας

➤ **Διάκριση αντλίας θερμότητας ανάλογα με το μέσο συναλλαγής θερμότητας**

Οι αντλίες θερμότητες ανάλογα με το μέσο που συναλλάσσεται στον εξωτερικό και στον εσωτερικό εναλλάκτη, κατηγοριοποιούνται ως εξής:

1. **Αντλίες θερμότητας αέρα – αέρα (αερόψυκτες):** Ο αέρας αποτελεί την πηγή θερμότητας καθώς και το μέσο μεταφοράς για να διοχετευτεί η ψύξη και η θέρμανση. Με τον τρόπο αυτό λειτουργούν τα κλιματιστικά τοίχου (air-condition). Σε μερικές περιπτώσεις, αποτελούνται από δύο ή περισσότερες εσωτερικές μονάδες, οι οποίες τοποθετούνται ξεχωριστά σε πολλούς χώρους του κτιρίου.

2. Αντλίες θερμότητας αέρα – νερού (αερόψυκτες): Ο αέρας αποτελεί την πηγή θερμότητας και το νερό χρησιμοποιείται ως μέσο μεταφοράς. Συνήθως, αυτό το νερό κυκλοφορεί στο υδραυλικό δίκτυο του κτιρίου και τελικά καταλήγει σε τερματικές μονάδες τύπου FanCoils, είτε συστήματα ενδοδαπέδιας θέρμανσης ή δροσισμού.
3. Αντλίες θερμότητας νερού – αέρα (υδρόψυκτες): Το νερό αποτελεί την πηγή θερμότητας και ο αέρας χρησιμοποιείται ως μέσο μεταφοράς. Αυτού του είδους αντλίες θερμότητας μπορούν να συνδυαστούν και με συστήματα εκμετάλλευσης ηλιακής ενέργειας με στόχο την θέρμανση του νερού.
4. Αντλίες θερμότητας νερού – νερού (γεωθερμικές): Το νερό αποτελεί την πηγή θερμότητας (είτε μέσω γεώτρησης, είτε από την θάλασσα ή την λίμνη) και λειτουργεί ως μέσο μεταφοράς. Αντλίες θερμότητας τέτοιου είδους έχουν πολύ καλύτερη απόδοση διότι, η θερμοκρασία του εδάφους είναι σταθερή σε σχέση με την θερμοκρασία του εξωτερικού αέρα που έχει πολλές διακυμάνσεις. Το μειονέκτημα τους είναι ότι απαιτείται μεγάλος χώρος για την κατασκευή όλης της εγκατάστασης.

➤ **Τύποι αντλιών θερμότητας ανάλογα με την θέση των μηχανισμών της**

1. Ενιαίες αντλίες θερμότητας (τύπου MonoBlock ή compact)

Όλα τα μέρη της συσκευής βρίσκονται εγκατεστημένα σε ένα ενιαίο κέλυφος. Είναι κατασκευασμένες για εγκατάσταση σε εξωτερικό χώρο ώστε να αντλούν άμεσα ενέργεια από το περιβάλλοντα ατμοσφαιρικό αέρα και βρίσκονται κοντά στο σημείο που καταλήγουν οι κεντρικές σωληνώσεις του κτιρίου, οι οποίες κυκλοφορούν το νερό στα θερμαντικά σώματα.

2. Διαιρούμενες αντλίες θερμότητας (τύπου Split)

Τα μέρη της συσκευής είναι διαχωρισμένα. Το ψυκτικό συγκρότημα το οποίο αντλεί ενέργεια από το περιβάλλοντα ατμοσφαιρικό αέρα βρίσκεται στην εξωτερική μονάδα, ενώ στην εσωτερική μονάδα βρίσκεται το υδραυλικό συγκρότημα όπου είναι και το κομμάτι που συνδέονται οι σωλήνες με τα θερμαντικά σώματα.

➤ **Τύποι αντλιών θερμότητας ανάλογα με τον τρόπο που λειτουργούν**

1. Τύπου inverter

Διαθέτουν έναν μηχανισμό που τους επιτρέπει να αυξομειώνουν τις στροφές του συμπιεστή τους και έτσι κάθε στιγμή προσαρμόζουν την ισχύ τους ανάλογα με τις ανάγκες θέρμανσης ή ψύξης του χώρου, με αποτέλεσμα να έχουν χαμηλή κατανάλωση κατά την λειτουργία τους.

2. Τύπου on/off

Τέτοιου είδους αντλίες θερμότητας λειτουργούν σε πλήρη ισχύ μέχρι να επιτευχθεί ο στόχος (θέρμανση ή ψύξη του χώρου) και μετά σταματούν (off). Επανεκιννούν (on) όταν η θερμοκρασία του χώρου αποκλίνει και πάλι από την επιθυμητή (McQuiston and Parker 2003).

2.4 Ηλιακά θερμικά συστήματα

2.4.1 Γενικές Πληροφορίες

Τα ηλιακά θερμικά συστήματα συνδυάζουν διάφορες τεχνολογίες και προσφέρουν ισχύ για ένα μεγάλο εύρος εφαρμογών, καθιστώντας τα αποδοτικά, αξιόπιστα και ανταγωνιστικά. Η θερμότητα από ηλιακή ενέργεια αξιοποιείται σε ένα ευρύ φάσμα εφαρμογών που περιλαμβάνει μεταξύ άλλων θέρμανση νερού χρήσης, θέρμανση χώρου, θερμότητα για βιομηχανικές διεργασίες, κλιματισμό, παραγωγή ατμού για βιομηχανικές διεργασίες, ξήρανση.

Υπάρχουν διάφοροι τύποι ηλιακών θερμικών συστημάτων που μπορούν να χρησιμοποιηθούν ανάλογα με την εφαρμογή για την οποία προορίζονται, την τεχνολογία που χρησιμοποιείται, το μέγεθός τους, τις κλιματικές συνθήκες της περιοχής κ.τ.λ..

Στην Ελλάδα, η οποία αποτελεί παγκοσμίως μία από τις πρωτοπόρες χώρες στην χρήση ηλιακών θερμικών, η συντριπτική πλειοψηφία αυτών των συστημάτων είναι κυρίως τα θερμοσιφωνικά (για θέρμανση νερού χρήσης). Ιδιαίτερα τα τελευταία χρόνια έχει αναπτυχθεί σημαντική εξαγωγική δραστηριότητα. Ενδεικτικά αναφέρεται ότι το 2017 η εγκατεστημένη ισχύς ηλιακών θερμικών συστημάτων σε λειτουργία στην Ελλάδα ήταν ίση με 3.216 MW_{th} που αντιστοιχούν σε 4.59 εκατ. m² συλλεκτών (SolarHeatEurope 2017).

Επιπρόσθετα με την χρήση της ηλιακής ενέργειας αποφεύγεται η χρήση ορυκτών καυσίμων. Το περιβαλλοντικό όφελος από την χρήση ηλιακών θερμικών συστημάτων προκύπτει από την εξοικονόμηση ενέργειας και κατ' επέκταση την μείωση εκπομπών αερίων του θερμοκηπίου. Μία μικρή αναφορά στα περιβαλλοντικά οφέλη που απορρέουν από την χρήση των ενεργητικών ηλιακών συστημάτων δείχνει ότι επιτυγχάνονται τα εξής (Κ.Α.Π.Ε. 2000):

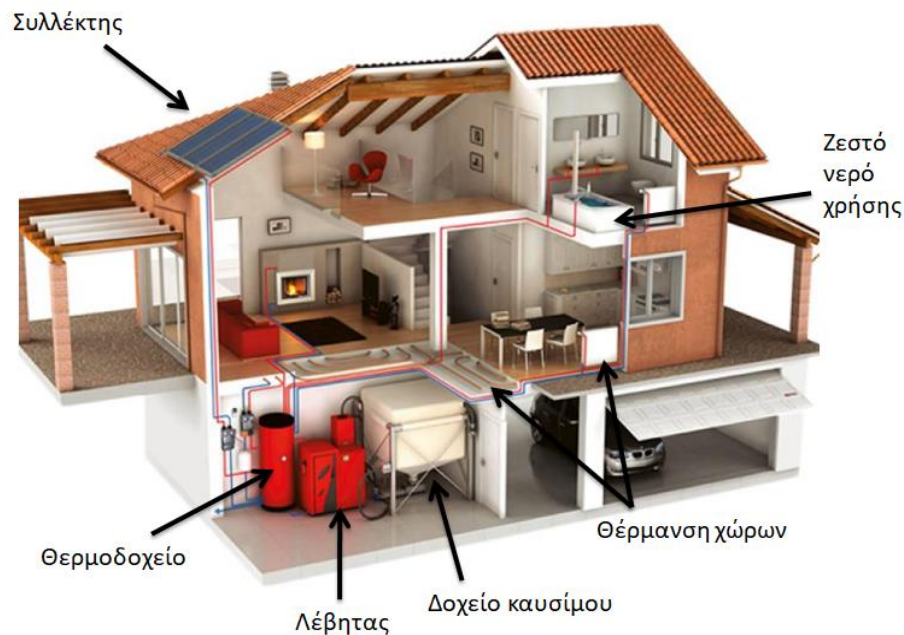
- Εξοικονόμηση καυσίμων που ισοδυναμεί με 50-70 kg πετρελαίου/m² συλλέκτη ετησίως.
- Μείωση εκπομπών CO₂ άνω των 750 kg/m² συλλέκτη ετησίως σε περίπτωση υποκατάστασης ηλεκτρικού ρεύματος.
- Μείωση εκπομπών CO₂ άνω των 250 kg/m² συλλέκτη ετησίως σε περίπτωση υποκατάστασης πετρελαίου.

2.4.2 Περιγραφή λειτουργίας

Η αρχή λειτουργίας των ηλιακών θερμικών συστημάτων είναι απλή: η ηλιακή ακτινοβολία μετατρέπεται σε θερμότητα που συλλέγεται και μεταφέρεται σε ένα ρευστό. Η θερμότητα χρησιμοποιείται απευθείας ή μέσω ενός εναλλάκτη μεταφέρεται στην τελική χρήση.

Ένα τυπικό σύστημα παραγωγής ζεστού νερού αποτελείται από τους ηλιακούς συλλέκτες, μία δεξαμενή αποθήκευσης του ζεστού νερού, τις απαραίτητες σωληνώσεις και το σύστημα ελέγχου. Η ηλιακή ακτινοβολία απορροφάται από τους συλλέκτη και η συλλεγόμενη θερμότητα αντλείται, με φυσικό ή τεχνητό τρόπο στη δεξαμενή. Το παραγόμενο ζεστό νερό χρήσης από θερμικά ηλιακά συστήματα

αποθηκεύεται σε ειδικές δεξαμενές προκειμένου στη συνέχεια να καταναλωθεί σε διάφορα σημεία του κτιρίου στο οποίο βρίσκεται η εγκατάσταση (Sonnenenergie 2005).



Εικόνα 2.5 Εγκατάσταση ηλιακού θερμικού συστήματος. Πηγή:(DianaSolar 2020)

2.4.3 Κατηγορίες ηλιακών θερμικών συστημάτων

2.4.3.1 Συστήματα φυσικής κυκλοφορίας

Τα συστήματα φυσικής κυκλοφορίας χωρίζονται σε δύο κατηγορίες:

- τους συμπαγείς θερμαντήρες νερού
- τα θερμοσιφωνικά συστήματα

2.4.3.1.1 Συμπαγείς θερμαντήρες νερού

Οι συμπαγείς θερμαντήρες νερού ή τα ολοκληρωμένα συστήματα συλλέκτη – αποθήκευσης ICS (Integrated Collector Storage) συνδυάζουν ηλιακούς συλλέκτες και δεξαμενές αποθήκευσης νερού σε μία μονάδα. Τα συστήματα αυτά πρόκειται για συστήματα που είναι απλά στην κατασκευή τους, στην λειτουργία τους και στην εγκατάστασή τους. Επίσης είναι συστήματα που μπορούν να καλύψουν ημερήσιες οικιακές ανάγκες έως 200 lt. Ένα βασικό τους μειονέκτημα είναι οι αυξημένες θερμικές απώλειες που εμφανίζουν κατά τη διάρκεια της νύχτας (Agnoutakis, etal. 2019).



Εικόνα 2.6 Τομή συμπαγούς θερμαντήρα νερού. Πηγή: (Κ.Α.Π.Ε. 2000)

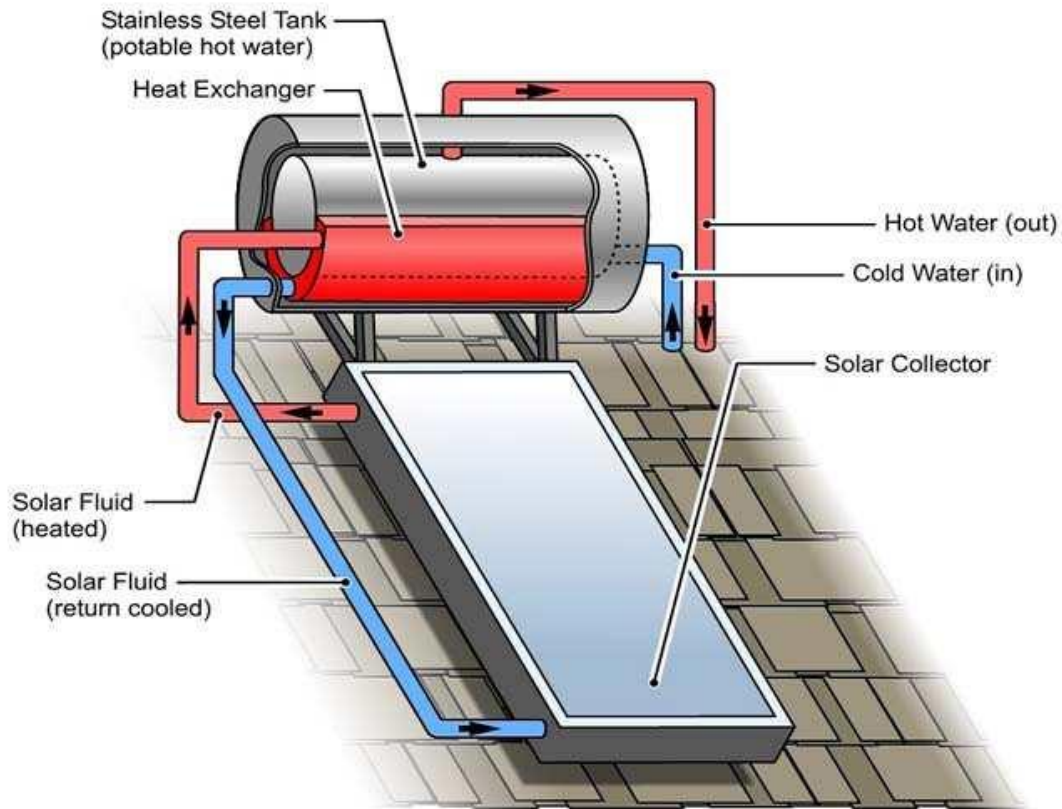
2.4.3.1.2 Θερμοσιφωνικό σύστημα

Τα θερμοσιφωνικά συστήματα αποτελούνται από ένα ή δύο συλλέκτες και μία δεξαμενή αποθήκευσης, η οποία τοποθετείται ακριβώς πάνω από το ή τους συλλέκτες. Τα θερμοσιφωνικά συστήματα θερμαίνουν νερό ή ένα θερμαντικό υγρό και χρησιμοποιούν φυσική συναγωγή για να το μεταφέρουν από το συλλέκτη στο θερμοδοχείο. Το θερμοσιφωνικό φαινόμενο προκύπτει καθώς η πυκνότητα του νερού ή του θερμαντικού υγρού μειώνεται με την αύξηση της θερμοκρασίας.

Η απορρόφηση της ηλιακής ακτινοβολίας έχει ως αποτέλεσμα τη θέρμανση του υγρού. Έτσι το υγρό διαστέλλεται, γίνεται λιγότερο πυκνό και ανυψώνεται μέσω του συλλέκτη στο θερμοδοχείο που βρίσκεται από πάνω. Εκεί αντικαθιστά το ψυχρότερο υγρό που έχει κατακαθίσει στον πάτο του δοχείου αποθήκευσης και το οποίο αρχίζει να ρέει προς τους συλλέκτες. Αυτή η ανακυκλοφορία συνεχίζεται για όση ώρα προσπίπτει ηλιακή ακτινοβολία στο συλλέκτη. Είναι ένα σύστημα που εφαρμόζεται κατά κόρον σε χώρες με υψηλό ηλιακό δυναμικό.

Σε εγκαταστάσεις που τοποθετούνται σε περιοχές με ψυχρότερο κλίμα, χρησιμοποιούνται αντιψυκτικά υγρά διαλύματα, όπως η προπυλενογλυκόλη σε κλειστό κύκλωμα και σωληνώσεις ανθεκτικές στον παγετό κατασκευασμένες συνήθως από υψηλής πυκνότητας πολυαιθυλένιο σταυροειδών δεσμών.

Κάποια από τα βασικά πλεονεκτήματα των θερμοσιφωνικών συστημάτων είναι ότι δεν βασίζονται σε αντλίες και ελεγκτές, είναι πιο αξιόπιστα, έχουν μεγάλη διάρκεια ζωής, καθώς δεν διαθέτουν κινούμενα μέρη και δεν απαιτείται ηλεκτρική τροφοδοσία για την λειτουργία τους (Κ.Α.Π.Ε. 2000).



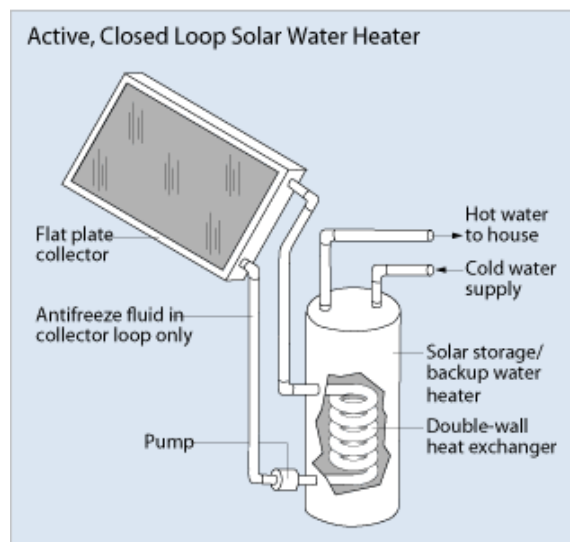
Εικόνα 2.7 Τυπική διάταξη ηλιακού θερμοσίφωνα. Πηγή: (Βαρβούνης 2013)

2.4.3.2 Συστήματα εξαναγκασμένης κυκλοφορίας

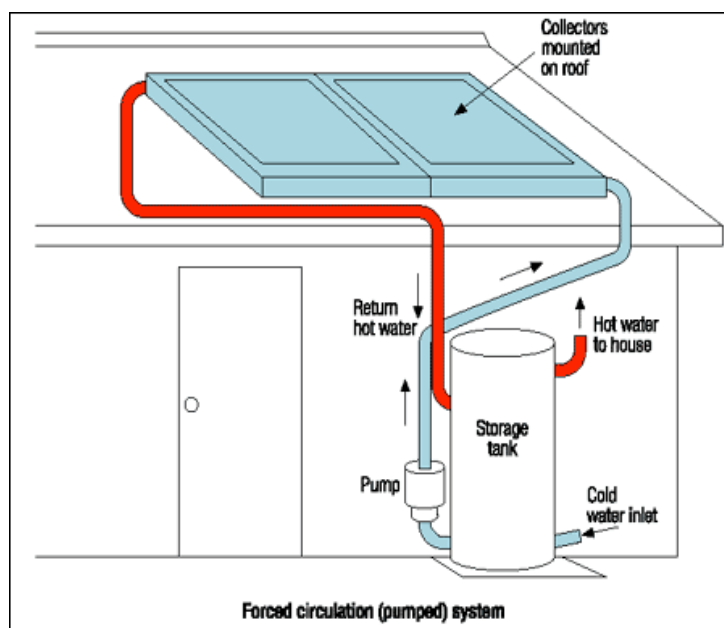
Τα συστήματα εξαναγκασμένης κυκλοφορίας χρησιμοποιούν ηλεκτρικές αντλίες, βαλβίδες και συστήματα ελέγχου για την εξασφάλιση της κυκλοφορίας του νερού ή των άλλων ρευστών μεταφοράς της θερμότητας μέσα στους συλλέκτες. Τα συστήματα αυτά διακρίνονται σε δύο τύπους:

- τα συστήματα ανοιχτού βρόχου
- τα συστήματα κλειστού βρόχου.

Η διαφορά των δύο τύπων έγκειται στην αρχή λειτουργίας τους, όπου τα συστήματα ανοιχτού βρόχου χρησιμοποιούν αντλίες για την κυκλοφορία του νερού του δικτύου στους συλλέκτες εν αντιθέσει με του κλειστού βρόχου, τα οποία αντλούν το ρευστό μεταφοράς θερμότητας συνήθως από ένα μίγμα γλυκόλης και νερού μέσα στους συλλέκτες. Η θερμότητα στη δεύτερη περίπτωση μεταφέρεται μέσω εναλλακτών θερμότητας από το ρευστό στο νερό που αποθηκεύεται στις δεξαμενές αποθήκευσης. Γενικά τα συστήματα ανοιχτού και κλειστού βρόχου μπορεί να είναι και συστήματα φυσικής κυκλοφορίας (Κ.Α.Π.Ε. 2000).



Εικόνα 2.8 Σύστημα κλειστού βρόχου. Πηγή: (wikimediacommons 2009)



Εικόνα 2.9 Σύστημα ανοιχτού βρόχου. Πηγή: (wikimedia commons 2009)

2.4.4 Τεχνολογίες ηλιακών θερμικών συστημάτων

Τα ηλιακά θερμικά συστήματα χρησιμοποιούν κυρίως ηλιακούς συλλέκτες και δεξαμενές αποθήκευσης (για δυνατότητα χρήσης σε περιπτώσεις όπου δεν υπάρχει ηλιοφάνεια, όπως την νύχτα ή όταν έχει συννεφιά) ως χωριστά υποσυστήματα, ενώ η μεταφορά της ενέργειας μέσω ενός θερμαινόμενου ρευστού (νερό, αέρας ή λάδι) πραγματοποιείται είτε με φυσική είτε με εξαναγκασμένη κυκλοφορία. Το βασικό πρόβλημα των ηλιακών θερμικών συστημάτων είναι οι χαμηλές και μεταβαλλόμενες τιμές της ροής της προσπίπτουσας ηλιακής ακτινοβολίας.

Οι ηλιακοί συλλέκτες διακρίνονται αρχικά σε δύο κατηγορίες:

- Μη συγκεντρωτικοί
 - Επίπεδοι συλλέκτες
 - Συλλέκτες με σωλήνες κενού αέρος
- Συγκεντρωτικοί
 - Σύνθετοι παραβολικοί συγκεντρωτές (CPC – Compound Parabolic Concentrators)
 - Με επιμήκεις παραβολικούς συγκεντρωτές (PTC–Parabolic Trough Concentrators)
 - Με παραβολοειδείς συγκεντρωτές
 - Γραμμικός ανακλαστής Fresnel (τύπου LFR– Linear Fresnel Reflector)
 - Με κεντρικό δέκτη

Επίσης οι ηλιακοί συλλέκτες διακρίνονται και βάσει της θερμοκρασίας λειτουργίας σε:

- Χαμηλής θερμοκρασίας (<80 °C) → Επίπεδοι συλλέκτες
- Μέσης θερμοκρασίας (80 – 250 °C) → Συλλέκτες με σωλήνες κενού
- Υψηλής θερμοκρασίας (>250 °C) → Συγκεντρωτικοί συλλέκτες

Οι μη συγκεντρωτικοί συλλέκτες χρησιμοποιούν την ίδια επιφάνεια για τη συλλογή και την απορρόφηση της ηλιακής ακτινοβολίας. Αντιθέτως, οι μη συγκεντρωτικοί συγκεντρώνουν την ηλιακή ακτινοβολία σε έναν δέκτη που αποτελεί μία μικρότερη επιφάνεια απορρόφησης χρησιμοποιώντας ανακλαστικές επιφάνειες ή διαθλαστικούς φακούς, αυξάνοντας έτσι την ροή της απορροφούμενης ακτινοβολίας (Duffie and Beckman 2013).

2.4.4.1 Επίπεδος ηλιακός συλλέκτης

2.4.4.1.1 Κύρια χαρακτηριστικά

Ο επίπεδος ηλιακός συλλέκτης είναι ο πιο διαδεδομένος τύπος συλλέκτη που χρησιμοποιείται σε οικιακές χρήσεις κυρίως για την παροχή ζεστού νερού χρήσης και για θέρμανση κτιρίων. Επίσης, μπορεί να χρησιμοποιηθεί για εφαρμογές κλιματισμού και παραγωγή θερμότητας στην βιομηχανία.

Χρησιμοποιείται κυρίως για εφαρμογές που απαιτούν απόδοση ενέργειας σε μικρές και μέσες θερμοκρασίες. Σε αυτό το είδος συλλέκτη γίνεται εκμετάλλευση τόσο της άμεσης όσο και της διάχυτης ακτινοβολίας.

Επειδή συνήθως οι συλλέκτες αυτοί είναι σταθεροί, δεν απαιτούν μηχανισμό κίνησης για την παρακολούθηση των ακτινών του ήλιου και συνεπώς είναι μηχανικά απλοί και δεν απαιτούν μεγάλη συντήρηση.

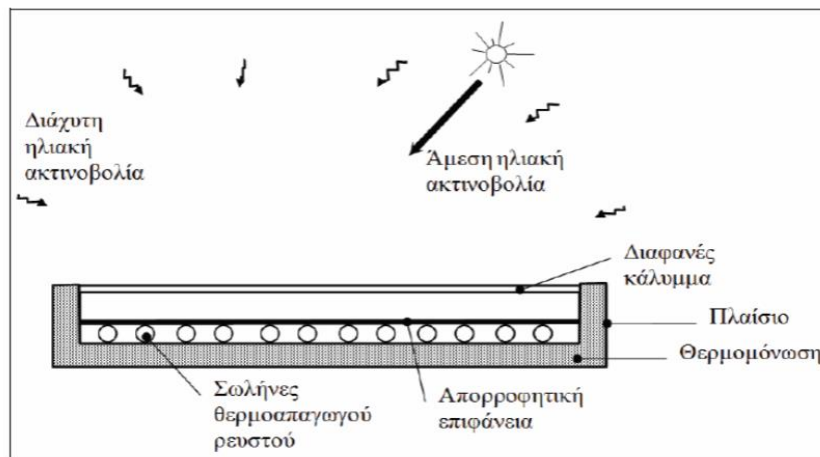
Η τοποθέτηση τους είναι στραμμένη προς τον ισημερινό και η κλίση τους ως προς την οριζόντια επιφάνεια είναι τέτοια ώστε να μεγιστοποιείται η συλλογή ηλιακής ακτινοβολίας ανάλογα με την εποχή που θέλουμε.

2.4.4.1.2 Κύρια μέρη

Τα κύρια μέρη που απαρτίζουν ένα επίπεδο ηλιακό συλλέκτη είναι:

- **Διαφανές κάλυμμα** που αποτελείται από ένα ή περισσότερα καλύμματα (transparent cover)
- **Απορροφητής ή πλάκα απορρόφησης** (absorber plate) που περιλαμβάνει την απορροφητική επιφάνεια και τους σωλήνες (αν πρόκειται για συλλέκτη θέρμανσης υγρού) ή τους αεραγωγούς (αν πρόκειται για θερμαντήρες αέρος) (Fluid conduit)
- **Μόνωση** της πίσω και των παράπλευρων επιφανειών (side and back insulation)
- **Πλαίσιο** που περιλαμβάνει τον συλλέκτη (frame)

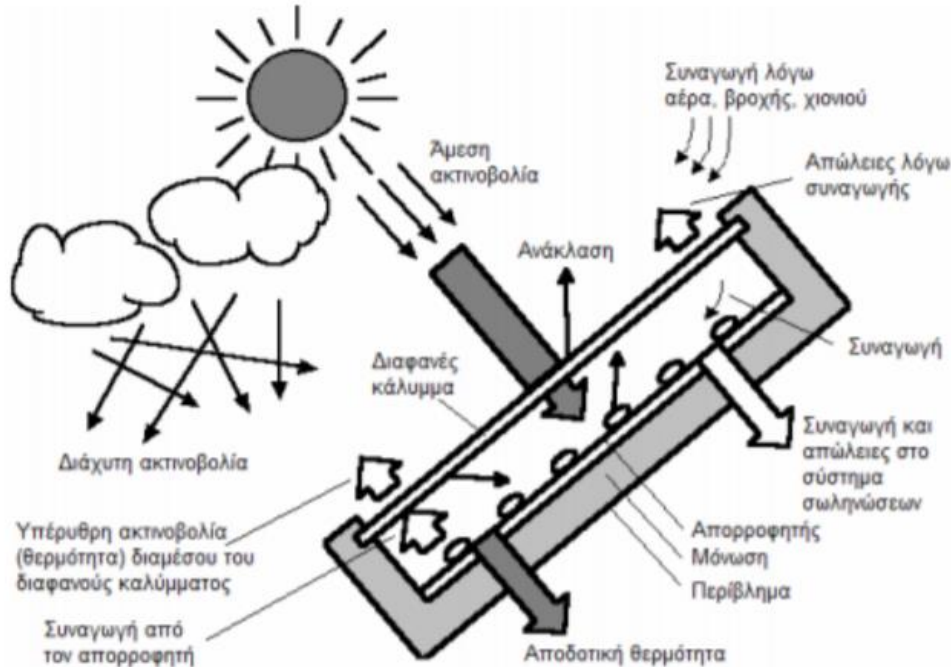
Στην εικόνα 2.10 που ακολουθεί φαίνεται η τομή ενός επίπεδου ηλιακού συλλέκτη με τα κύρια μέρη του.



Εικόνα 2.10 Κύρια μέρη επίπεδου ηλιακού συλλέκτη. Πηγή: (Κατσιγιαννης 2015)

2.4.4.1.3 Αρχή λειτουργίας

Ένα μεγάλο μέρος της προσπίπτουσας ηλιακής ακτινοβολίας διαπερνά το κάλυμμα και προσκρούει στην πλάκα απορρόφησης. Όπως αναπαρίσταται και στην εικόνα 2.11, το μεγαλύτερο ποσοστό της προσπίπτουσας ακτινοβολίας απορροφάται από τον απορροφητή και τις σωληνώσεις, ενώ το υπόλοιπο ανακλάται προς τυχαίες κατευθύνσεις και προσκρούει με την σειρά του στην εσωτερική πλευρά του καλύμματος. Το μεγαλύτερο μέρος αυτής της ανακλώμενης ακτινοβολίας ανακλάται πάλι πίσω από την εσωτερική επιφάνεια του καλύμματος προς την πλάκα, με αποτέλεσμα να παγιδεύεται εντός του διακένου (φαινόμενο του θερμοκηπίου) και τελικά να απορροφάται κατά ένα μεγάλο ποσοστό από την πλάκα και τους αγωγούς. Η θερμότητα που δεσμεύεται από τον απορροφητή μεταφέρεται μέσω αγωγής στα τοιχώματα των αγωγών και στην συνέχεια μέσω συναγωγής στο εργαζόμενο μέσο. Το κάτω μέρος του απορροφητή καθώς και η εσωτερική πλευρά του πλαισίου είναι καλά μονωμένα για την μείωση των θερμικών απωλειών λόγω αγωγής και ακτινοβολίας (Kalogirou 2004).



Εικόνα 2.11 Μετάδοση θερμότητας σε επίπεδο ηλιακό συλλέκτη. Πηγή: (Κατσίγιαννης 2015)

Το εργαζόμενο μέσο εντός ενός τυπικού επίπεδου ηλιακού συλλέκτη φυσικής κυκλοφορίας ρέει βάσει της αρχής του θερμοσιφώνου, η οποία εξηγήθηκε στην υπό ενότητα 2.4.3.1.2.

- Το **διαφανές κάλυμμα** έχει την ιδιότητα να παρουσιάζει μεγάλη διαπερατότητα σε υψηλής συχνότητας ακτινοβολία και χαμηλή διαπερατότητα σε χαμηλής συχνότητας ακτινοβολία. Έτσι επιτρέπει στην προσπίπτουσα ηλιακή ακτινοβολία να το διαπεράσει και να την εγκλωβίσει στο διάκενο που δημιουργείται μεταξύ της πλάκας και του απορροφητή, για την εκμετάλλευση του μεγαλύτερου μέρους της. Επίσης, ταυτόχρονα βοηθά ώστε να μειώνονται οι θερμικές απώλειες δια συναγωγής και ακτινοβολίας προς το περιβάλλον, αφού το μεγαλύτερο μέρος της ηλιακής ακτινοβολίας που ανακλάται από τον απορροφητή επιστρέφει πάλι πίσω και έτσι μειώνονται οι διαφυγές της ηλιακής ακτινοβολίας. Συνήθως τα υλικά που χρησιμοποιούνται για το κάλυμμα είναι το απλό γυαλί (Kalogirou 2004).
- Ο **απορροφητής ή πλάκα απορρόφησης** είναι το πλέον σημαντικό κομμάτι ενός ηλιακού συλλέκτη δεδομένου ότι το μεγαλύτερο μέρος της θερμότητας που θα πάρει το εργαζόμενο μέσο εξαρτάται από την ηλιακή ακτινοβολία που απορροφά. Είναι ένα σύστημα, το οποίο περιλαμβάνει την απορροφητική επιφάνεια και τους αγωγούς κυκλοφορίας του ρευστού. Η απορροφητική επιφάνεια είναι μία επίπεδη μεταλλική επιφάνεια επικαλυμμένη με μαύρο χρώμα ή με στρώμα επιλεκτικής βαφής, ώστε να μεγιστοποιεί την συλλογή θερμικής ισχύος. Επειδή γενικά η απορροφητικότητα των υλικών που χρησιμοποιούνται για την κατασκευή της απορροφητικής επιφάνειας δεν είναι μεγάλη, στην επιφάνεια τους τοποθετείται κατάλληλη επικάλυψη υψηλής απορροφητικότητας. Οι συλλέκτες με επιλεκτική βαφή μπορεί να έχουν μέχρι και 30% μεγαλύτερη απόδοση, σε σχέση με τους συλλέκτες με απλή μαύρη βαφή, λόγω της

μείωσης των απωλειών ανάκλασης. Για χαμηλές θερμοκρασίες (όπως θέρμανση πισίνας) χρησιμοποιούνται πλαστικές απορροφητικές πλάκες. Οι μεταλλικές πλάκες κατασκευάζονται συνήθως από υλικά όπως χαλκό, αλουμίνιο ή χάλυβα εξαιτίας του σχετικά χαμηλού κόστους, της υψηλής αντοχής τους και των καλών θερμομηχανικών ιδιοτήτων τους (Kalogirou 2004).

- Η **μόνωση** των πλάγιων και πίσω πλευρών του συλλέκτη χρησιμοποιείται ώστε να μειώνονται οι θερμικές απώλειες προς το περιβάλλον. Ο υαλοβάμβακας υψηλής ποιότητας και ο πετροβάμβακας εμφανίζουν καλές μονωτικές ιδιότητες και αντοχή στις θερμοκρασίες που αναπτύσσονται μέσα στο συλλέκτη. Επίσης, και ο αφρός πολυουρεθάνης είναι ένα συχνά χρησιμοποιούμενο υλικό εξαιτίας και του πολύ χαμηλού συντελεστή θερμικής αγωγιμότητας, αλλά και της μεγάλης αντοχής σε υγρασία (Kalogirou 2004).
- Το **πλαίσιο** που περιβάλλει τον συλλέκτη έχει ως βασικό στόχο να τον προστατεύει από τις περιβαλλοντικές συνθήκες όπως την σκόνη και την υγρασία ενώ προσφέρει ακαμψία στο σύστημα. Συνήθως είναι μεταλλικό, αλλά σε ορισμένες περιπτώσεις συναντάται και πλαστικό (Kalogirou 2004).

2.4.4.2 Συγκεντρωτικός ηλιακός συλλέκτης

Οι συγκεντρωτικοί ηλιακοί συλλέκτες είναι συσκευές που συγκεντρώνουν την ηλιακή ενέργεια με οπτικά μέσα και στην συνέχεια την απορροφούν και την μετατρέπουν σε χρήσιμη θερμική ενέργεια. Η συγκέντρωση της ηλιακής ενέργειας επιτρέπει τη συλλογή θερμότητας σε πολύ υψηλότερες θερμοκρασίες σε σχέση με τους επίπεδους συλλέκτες. Συνήθως αυτού του τύπου συλλέκτες χρησιμοποιούνται στις βιομηχανίες ή στην παραγωγή ηλεκτρισμού.

Ένας συγκεντρωτικός ηλιακός συλλέκτης αποτελείται από το οπτικό σύστημα ή συγκεντρωτήρα και τον δέκτη. Ο συγκεντρωτήρας δέχεται την ηλιακή ακτινοβολία, τη συγκεντρώνει και την οδηγεί στο δέκτη, ο οποίος την απορροφά. Η πολλαπλασιασμένη ένταση της προσπίπτουσας στον δέκτη ηλιακής ακτινοβολίας συνεπάγεται μικρότερη επιφάνεια για ένα δεδομένο ποσό ενέργειας, άρα και μειωμένες απώλειες προς το περιβάλλον. Ο συγκεντρωτήρας μπορεί να είναι φακός και να διαθλά την ηλιακή ακτινοβολία ή να είναι καθρέπτης και να την ανακλά. Ακόμα μπορεί να είναι επίπεδος, παραβολικός, ή να αποτελείται από μία σειρά κινούμενων επίπεδων κατόπτρων, τα οποία με κατάλληλους μηχανισμούς παρακολουθούν την πορεία του ήλιου (Hanania, et al. 2018).

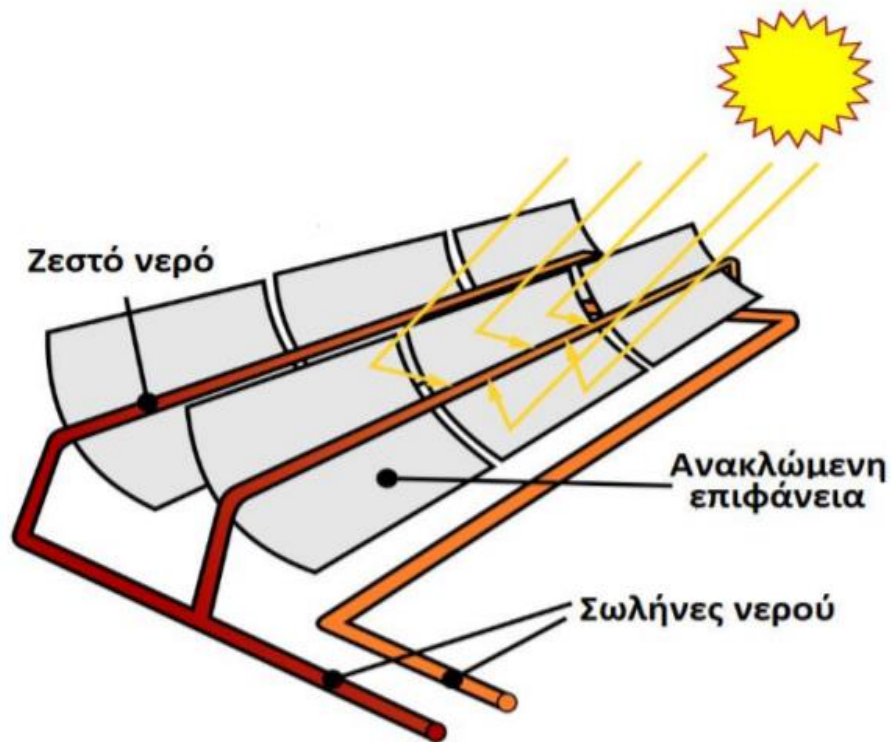
Οι συγκεντρωτικοί συλλέκτες χωρίζονται σε συλλέκτες γραμμικής εστίασης και σημειακής εστίασης.

- **Συλλέκτες γραμμικής εστίασης**
Αυτοί οι συλλέκτες αποτελούνται από παραβολικά διαμορφωμένα ανακλαστικά τμήματα (κάτοπτρα). Κατά μήκος αυτών, στο σημείο εστίασης τοποθετούνται σωλήνες που κυκλοφορεί το θερμοαπωγόνιο ρευστό και στους οποίους γίνεται η συγκέντρωση της ηλιακής ακτινοβολίας (εικόνα 2.12). Χρησιμοποιούν υλικά με μεγάλη ανακλαστικότητα για τη συλλογή και τη συγκέντρωση μεγάλης θερμικής ενέργειας από την ηλιακή ακτινοβολία. Δηλαδή, είναι σύστημα πολύ υψηλής ισχύος και ως εκ

τούτου χρησιμοποιούνται γενικά για την παραγωγή ατμού για ηλιακούς θερμοηλεκτρικούς σταθμούς και δεν χρησιμοποιούνται σε οικιακές εφαρμογές (Hanania, et al. 2018).

➤ Συλλέκτες σημειακής εστίασης

Οι συγκεντρωτικοί συλλέκτες σημειακής εστίασης, συγκεντρώνουν την ηλιακή ακτινοβολία σε ένα σημείο (εικόνα 2.13). Η παρακολούθηση της τροχιάς του ήλιου γίνεται σε δύο διαστάσεις. Συνήθως, αυτοί οι συλλέκτες χρησιμοποιούνται για τη παροχή θερμότητας σε μηχανές (όπως μηχανές Stirling). Επίσης, μπορούν να συνδυαστούν με συγκεντρωτικά φωτοβολταϊκά. Σε αυτήν την περίπτωση, η ενέργεια του ήλιου μετατρέπεται απευθείας σε ηλεκτρική ενέργεια (Hanania, et al. 2018).



Εικόνα 2.12 Συγκεντρωτικός συλλέκτης γραμμικής εστίασης. Πηγή: (Hanania, et al. 2018)



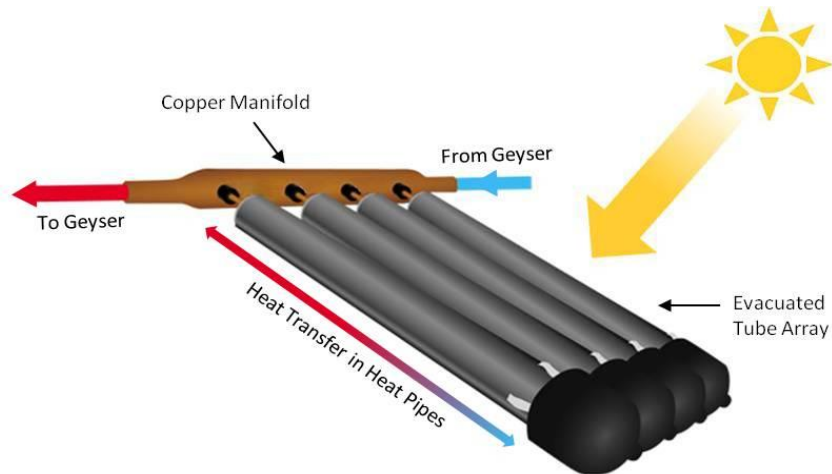
Εικόνα 2.13 Συγκεντρωτικός συλλέκτης σημειακής εστίασης. Πηγή: (Hanania, etal. 2018)

2.4.4.3 Ηλιακός συλλέκτης με σωλήνες κενού αέρος

Οι ηλιακοί συλλέκτες κενού αποτελούνται από μία σειρά σωλήνων κενού σε παράλληλη διάταξη. Ο κάθε σωλήνας αποτελείται από δύο γυάλινους σωλήνες που ο ένας βρίσκεται μέσα στον άλλο με αποτέλεσμα ενδιάμεσα τους να υπάρχει κενό. Η ηλιακή ακτινοβολία διαπερνά το γυάλινο σωλήνα και απορροφάται από τη μαύρη επιφάνεια (απορροφητής) που βρίσκεται εντός του εσωτερικού σωλήνα. Αυτός μετατρέπει την ηλιακή ενέργεια σε θερμότητα. Το θερμοαπαγωγό ρευστό που περνάει έρχεται σε επαφή με αυτήν την επιφάνεια και παίρνει τη θερμότητα που συλλέγεται. Είναι αξιοσημείωτο, ότι το ποσοστό απορρόφησης της ηλιακής ακτινοβολίας μπορεί να φτάσει μέχρι 96% αφού το ποσοστό αντανάκλασης είναι πολύ μικρό. Επειδή το κενό αέρος, που υπάρχει μεταξύ των δύο σωλήνων, είναι πολύ καλό μονωτικό υλικό, οι απώλειες θερμότητας προς το περιβάλλον είναι πολύ μικρές. Λόγω της βελτιωμένης απόδοσής τους σε σχέση με τους επίπεδους ηλιακούς συλλέκτες, χρησιμοποιούνται όταν απαιτούνται μέσες έως υψηλές τιμές θερμοκρασιών ή υψηλότεροι όγκοι ζεστού νερού, καθώς και για συστήματα θέρμανσης και ηλιακού κλιματισμού (GreenSolarVacuum 2020).

Επιπλέον χαρακτηριστικά των συλλεκτών κενού είναι:

- Εκμεταλλεύονται και την άμεση και τη διάχυτη ακτινοβολία
- Έχουν μικρό βάρος και όγκο
- Έχουν υψηλή αντοχή σε θερμοκρασίες (έως μείον 40 °C)
- Επιτυγχάνουν πολύ μεγάλη απόδοση σε σχέση με τους επίπεδους συλλέκτες
- Επιτυγχάνουν υψηλές θερμοκρασίες λειτουργίας και είναι ιδανικοί για ψυχρά κλίματα.



Εικόνα 2.14 Ηλιακός συλλέκτης κενού. Πηγή: (Solar Sence 2013)

2.5 Εφαρμογές αντλιών θερμότητας σε συστήματα θέρμανσης ZNX

2.5.1 Σχετικά με τις διατάξεις ZNX

Οι Duffie and Beckman (Duffie and Beckman 2013) επισημαίνουν πως η πιο σύνηθης θερμοκρασιακή στάθμη των δοχείων ZNX αποτελεί το εύρος των 45-50 °C. Σύμφωνα με την εργασία των Pezzuto S. et al. (Pezzutto S. 2015), η ετήσια κατανάλωση ενέργειας για την κάλυψη φορτίων ZNX ανά επιφάνεια κατοικίας στην Ελλάδα ανέρχεται στις 21 kWh/m². Συγκριτικά αναφέρεται επίσης πως οι αντίστοιχες καταναλώσεις σε θέρμανσης περίπου 90 kWh/m². Ανάλογες καταναλώσεις ZNX μπορούν να παρατηρηθούν όχι μόνο σε χώρες της Ευρώπης, αλλά και στην Αμερική (Behidj N. 2006). Συνεπώς, αξίζει να δοθεί έμφαση στην κατανάλωση φορτίων ZNX, καθώς και στον τρόπο πρόσδοσης αυτών.

2.5.2 Διάταξεις με αυτόνομη αντλία θερμότητας

Χρήση αντλίας θερμότητας για τη προετοιμασία ZNX αποτελεί μια αποτελεσματική προσέγγιση για την κάλυψη φορτίων. Στην ανάλυση των Rossano Scoccia et al. (Rossano Scoccia 2017), παρατηρήθηκε πως οι αντλίες θερμότητας τείνουν να αποδίδουν καλύτερα από συστήματα θερμικής πρόσδοσης, κυρίως σε χώρες της Ευρώπης με κρύο κλίμα και χαμηλό ενεργειακό δυναμικό. Οι εγκαταστάσεις αντλιών θερμότητας μπορούν να αποσβεστούν σε χρονικά διαστήματα μικρότερης της πενταετίας (William M. Duarte 2020). Στην εργασία των Μουζεβίρης και Παπακόστας (Mouzeviris and Parakostas 2020), μπορεί να παρατηρηθεί πως οι βαθμοί απόδοσης των αντλιών τείνουν να ξεπερνούν την τιμή του 3, για θερμοκρασίες πρόσδοσης νερού 35 - 45 °C. Επομένως, οι αντλίες θερμότητας μπορούν να λειτουργήσουν αποδοτικά για τη θέρμανση του ZNX.

2.5.3 Διατάξεις με συνδυασμό αντλίας θερμότητας και ηλιακών συλλεκτών

Η συνδυαστική διάταξη ηλιακού θερμικού συστήματος, υποβοηθούμενου από αντλία θερμότητας μπορεί να αξιοποιήσει την εκμετάλλευση του ηλιακού δυναμικού της περιοχής εγκατάστασης. Ταυτόχρονα, εξασφαλίζει τη συνεχή κάλυψη των θερμικών φορτίων σε περιόδους χαμηλής έντασης ηλιακής ακτινοβολίας, ή κατά τις νυχτερινές ώρες. Σε προσομοιώσεις λειτουργίας των G. Panaras et al. (G. Panaras 2013), σημαντική παράμετρος φαίνεται να αποτελεί η θερμοκρασία-στόχος λειτουργίας της αντλίας θερμότητας, καθώς επηρεάζει την απόδοση του συστήματος. Στην ανάλυση των S. H. Razavi et al. (Seyed Houman Razavi 2018), αναφέρεται πως η εφαρμογή μιας τέτοιας εγκατάστασης μπορεί να μειώσει την ενεργειακή κατανάλωση ως και 8.7%.

Η ιδιαιτερότητα του συγκεκριμένου συστήματος είναι η λειτουργεί χρησιμοποιώντας ηλεκτρικά και ηλιακά φορτία. Παρόλα αυτά, τα ηλεκτρικά φορτία που απαιτούνται από την αντλία θερμότητας μπορούν να καλυφθούν από διατάξεις φωτοβολταϊκών.

Επιπλέον, εξετάζεται ο συνδυασμός φωτοβολταϊκών, ηλιακών συλλεκτών και αντλίας θερμότητας. Από μερικές αναλύσεις (Francesca Martorana 2021), (Puneet Saini 2021) δεν έχει παρατηρηθεί κάποια σημαντική βελτίωση στην απόδοση των θερμικών συστημάτων.

3. Παραμετρική βελτιστοποίηση συστημάτων HVAC

3.1 Εισαγωγή στη βελτιστοποίηση

Σε εφαρμογές του πραγματικού κόσμου, τα περισσότερα προβλήματα βελτιστοποίησης απαιτούν την επίτευξη περισσότερων του ενός στόχου όπως για τον κλάδο των μηχανικών μερικοί στόχοι που τίθενται προς βελτιστοποίηση είναι να μεγιστοποιηθεί η απόδοση ή η αξιοπιστία κάποιου συστήματος, να ελαχιστοποιηθεί το κόστος κ.α.. Οι τύποι βελτιστοποίησης χωρίζονται σε δύο κατηγορίες (Savic 2002):

- μονοκριτηριακή βελτιστοποίηση (Single Objective Optimization, SOO)
- πολυκριτηριακή βελτιστοποίηση (Multi Objective Optimization, MOO)

Ο κύριος στόχος της μονοκριτηριακής βελτιστοποίησης είναι να βρεθεί η καλύτερη λύση, η οποία αντιστοιχεί στην ελάχιστη ή στη μέγιστη τιμή μίας μοναδικής αντικειμενικής συνάρτησης. Αυτός ο τύπος βελτιστοποίησης είναι χρήσιμος ως εργαλείο το οποίο θα παρέχει πληροφορίες σχετικά με την φύση ενός προβλήματος, αλλά συνήθως δεν μπορεί να παρέχει ένα σύνολο εναλλακτικών βέλτιστων λύσεων (Savic 2002).

Ο κύριος στόχος της πολυκριτηριακής βελτιστοποίησης είναι η μεγιστοποίηση ή ελαχιστοποίηση ενός συνόλου διαφορετικών στόχων, οι οποίοι μπορεί να αλληλεπιδρούν ο ένας με τον άλλο. Η αλληλεπίδραση αυτή μεταξύ των στόχων οδηγεί στην εύρεση ενός συνόλου από βέλτιστες λύσεις και όχι μίας και μοναδικής (Savic 2002).

Τα προβλήματα μεγιστοποίησης και ελαχιστοποίησης ορίζονται ως εξής:

- Μεγιστοποίηση: για κάθε κριτήριο f ενός συνόλου λύσεων υπάρχει ένα $x \in S$ (S αποτελεί την περιοχή των εφικτών λύσεων), για το οποίο ισχύει $f_{\max} = \max (f(x^*)), x^* \in S$.
- Ελαχιστοποίηση: για κάθε κριτήριο f ενός συνόλου λύσεων υπάρχει ένα $x \in S$ (S αποτελεί την περιοχή των εφικτών λύσεων), για το οποίο ισχύει $f_{\min} = \min (f(x^*)), x^* \in S$.

Στις δύο υπό ενότητες που ακολουθούν παρουσιάζονται αρχικά κάποιες σχετικές έρευνες πολυκριτηριακής βελτιστοποίησης και στη συνέχεια γίνεται η ανάλυση της πολύκριτηριακής βελτιστοποίησης με τη μέθοδο βελτιστοποίησης Pareto, που χρησιμοποιήθηκε και στην επίλυση της συγκεκριμένης εργασίας.

3.2 Παρουσίαση σχετικών ερευνών

Στη βιβλιογραφία υπάρχει πληθώρα δημοσιεύσεων που ερευνά την πολυκριτηριακή βελτιστοποίηση συστημάτων θέρμανσης, εξαερισμού και κλιματισμού (Heating, Ventilation, and Air Conditioning, HVAC). Παρακάτω παρουσιάζεται μια σύντομη αναφορά τέτοιου είδους εργασιών.

Στην εργασία των Andrew Kusiak et al. (Kusiak, TangandXu 2011): μελετήθηκε η βελτιστοποίηση του συστήματος HVAC ως προς την ελαχιστοποίηση της χρήσης ενέργειας διατηρώντας

την ποιότητα αέρα εσωτερικού χώρου (indoor air quality, IAQ) εντός εύρους που καθορίζεται από τον χρήστη. Η εξοικονόμηση ενέργειας επιτυγχάνεται με τον υπολογισμό των καθορισμένων σημείων για την θερμοκρασία αέρα τροφοδοσίας και τη στατική πίεση του αεραγωγού.

Στην εργασία των Αλλαν Ρ. Σταρκε et al. (Starke, Cardemil and Collea 2018): παρουσιάστηκε η πολυκριτηριακή βελτιστοποίηση έμμεσων ηλιακών συστημάτων αντλίας θερμότητας για την θέρμανση εξωτερικής πισίνας. Οι δύο στόχοι βελτιστοποίησης ήταν η ελαχιστοποίηση του ετήσιου κύκλου ζωής (Annualized Life Cycle Cost, ALCC) και η μεγιστοποίηση του επιπέδου άνεσης που προσφέρει η κολυμβητική δεξαμενή. Τα τέσσερα συστήματα που μελετήθηκαν ήταν μια αντλία θερμότητας αέρα-νερού και τρεις αντλίες θερμότητας με ηλιακή βοήθεια (αέρα-νερού, νερού-νερού και διπλής πηγής). Η αξιολόγηση των συστημάτων έγινε σε έξι διαφορετικές περιοχές.

Οι Ν. Νασσιφ et al. (Nassif, Kajl and Sabourin 2004): μελέτησαν την βελτιστοποίηση ενός συστήματος HVAC ως προς την χρήση ενέργειας και την θερμική άνεση. Οι μεταβλητές που εξετάστηκαν ήταν ο καθορισμός της θερμοκρασία αέρα τροφοδοσίας, της στατικής πίεσης του αγωγού, της θερμοκρασίας κρύου νερού και τις θερμοκρασίες θερμικής ζώνης.

Αντικείμενο της μελέτης των Μ. Μοσσόλυ et al. (Mossolly, Ghali and Ghaddar 2009): ήταν η βελτιστοποίηση των στρατηγικών ελέγχου ενός συστήματος κλιματισμού μεταβλητού όγκου αέρα. Οι στρατηγικές αυτές περιελάμβαναν μία βασική στρατηγική ελέγχου σταθερού σημείου ρύθμισης θερμοκρασίας και δυο προηγμένες στρατηγικές για τη διασφάλιση της θερμικής άνεσης και της ποιότητας του εσωτερικού αέρα IAQ. Η βελτιστοποίηση για κάθε στρατηγική γίνεται στο κόστος της κατανάλωσης ενέργειας. Ο πρώτος προηγμένος έλεγχος προσαρμόζει το ρυθμό παροχής φρέσκου αέρα και τη θερμοκρασία αέρα τροφοδοσίας για να διατηρεί το σημείο ρύθμισης θερμοκρασίας σε κάθε ζώνη, ενώ η δεύτερη στρατηγική ελέγχει το ρυθμό φρέσκου αέρα και την θερμοκρασία αέρα τροφοδοσίας για να διατηρήσει μία αποδεκτή θερμική άνεση και την ποιότητα εσωτερικού αέρα IAQ σε κάθε ζώνη.

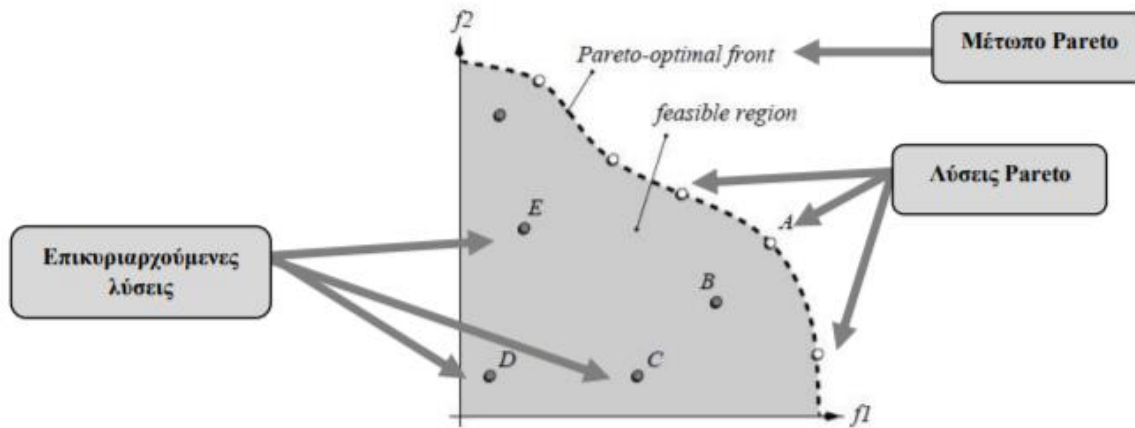
3.3 Ανάλυση πολυκριτηριακής βελτιστοποίησης

3.3.1 Η έννοια των Pareto βέλτιστων λύσεων

Στα προβλήματα βελτιστοποίησης, η βέλτιστη λύση είναι αυτή η οποία έχει όλες τις βέλτιστες τιμές για όλα τα κριτήρια που λαμβάνονται υπόψη, ταυτόχρονα. Επομένως, για ένα πολυκριτηριακό πρόβλημα βελτιστοποίησης ο στόχος είναι να βρεθεί μία λύση x^* όπου $x^* \in S$ και η περιοχή S αποτελεί την περιοχή των εφικτών λύσεων και $f_i(x^*) \geq f_i(X)$ για όλα τα X και τα $x^* \in S$. Η μόνη περίπτωση να υπάρξει μονάχα μία βέλτιστη λύση σε πολυκριτηριακό πρόβλημα είναι όταν τα κριτήρια δεν είναι αντικρουόμενα. Αντίθετα, στην περίπτωση αντικρουόμενων κριτηρίων έχουμε την ύπαρξη ενός συνόλου από βέλτιστες λύσεις οι οποίες αποτελούν τις μη επικυριαρχούμενες λύσεις.

Το φαινόμενο της πολυκριτηριακής βελτιστοποίησης, κατά την οποία δεν εμφανίζεται μια βέλτιστη λύση, αλλά ένα βέλτιστο σύνολο λύσεων παρουσιάζεται στο παρακάτω σχήμα και καλείται όριο

Pareto βέλτιστων λύσεων (Pareto frontier) ή μέτωπο Pareto (Pareto front) (Deb 2011). Στην εικόνα 3.1 παρουσιάζεται ένα χαρακτηριστικό πρόβλημα μεγιστοποίησης πολλαπλών κριτηρίων.



Εικόνα 3.1 Το μέτωπο Pareto σε ένα πρόβλημα μεγιστοποίησης δύο κριτηρίων. Πηγή: (Zitzler 1999)

Για τον σχηματισμό του ορίου Pareto, απαραίτητη είναι η χρήση ενός μοναδικού κριτηρίου, το οποίο να χαρακτηρίζει κάθε λύση. Το κριτήριο αυτό ονομάζεται κριτήριο της κυριαρχίας (dominance). Για ένα πρόβλημα μεγιστοποίησης, αντιστόχου αυτού της εικόνας 3.1., μια λύση x_i χαρακτηρίζεται ως κυριαρχική έναντι μιας άλλης x_j , όταν ικανοποιούνται και οι δύο παρακάτω συνθήκες (Zitzler 1999):

- $f_i(x) \geq f_i(x')$ για κάθε $i \in \{1, 2, \dots, k\}$ (η εξεταζόμενη λύση x να μην είναι χειρότερη από μια άλλη λύση x' σε όλα τα κριτήρια f)
- $f_j(x) > f_j(x')$ για τουλάχιστον ένα $j \in \{1, 2, \dots, k\}$ (η εξεταζόμενη λύση x να μην είναι καλύτερη σε τουλάχιστον ένα κριτήριο f από μια άλλη λύση x')

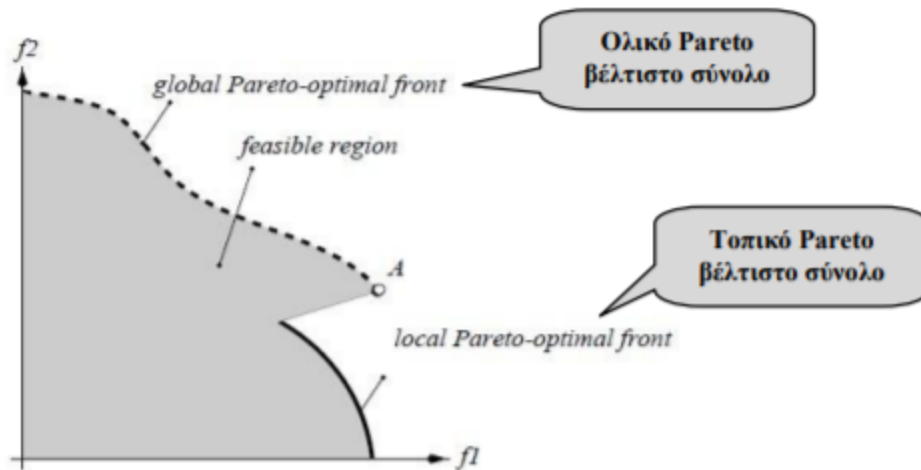
3.3.2 Pareto βέλτιστες λύσεις

Για ένα πεπερασμένο σύνολο λύσεων, μπορούμε να εφαρμόσουμε όλες τις πιθανές συγκρίσεις μεταξύ των λύσεων για την εύρεση των επικυριαρχούμενων και των μη επικυριαρχούμενων λύσεων. Στο τέλος, θα υπάρξει ένα σύνολο λύσεων όπου καμία δεν θα επικυριαρχείται από κάποια άλλη. Το συγκεκριμένο σύνολο έχει μία πρόσθετη ιδιότητα: όλες οι λύσεις που δεν ανήκουν σε αυτό το σύνολο επικυριαρχούνται τουλάχιστον από μία λύση του συγκεκριμένου συνόλου. Με απλά λόγια, οποιαδήποτε λύση του συνόλου αυτού είναι ανώτερη από οποιαδήποτε λύση που δεν είναι μέλος του συνόλου. Το σύνολο αυτό ονομάζεται μη επικυριαρχούμενο σύνολο. Οι παρακάτω ορισμοί θέτουν το μη επικυριαρχούμενο σύνολο λύσεων (Zitzler 1999).

Μη επικυριαρχούμενο σύνολο: Μεταξύ ενός συνόλου λύσεων P , το μη επικυριαρχούμενο σύνολο $p \subseteq P$ είναι αυτό που καμία λύση του p δεν επικυριαρχείται από οποιαδήποτε λύση του P . Όταν το σύνολο P είναι ολόκληρος ο χώρος έρευνας, $P = S$, τότε το δημιουργούμενο μη επικυριαρχούμενο σύνολο καλείται Pareto βέλτιστο σύνολο.

Ολικά Paretoβέλτιστο σύνολο: Το μηεπικυριαρχούμενο σύνολο ολόκληρου του εφικτού χώρου έρευνας S αποτελεί το ολικά Pareto βέλτιστο σύνολο. Σε πολλές περιπτώσεις το ολικά Paretoβέλτιστο σύνολο αναφέρεται ως Pareto βέλτιστο σύνολο, αφού οι λύσεις που περιέχει δεν επικυριαρχούνται από κανένα άλλο μέλος του εφικτού χώρου έρευνας, και αποτελούν τις βέλτιστες λύσεις για το πολυκριτηριακό πρόβλημα βελτιστοποίησης.

Τοπικά Pareto βέλτιστο σύνολο: Αν για κάθε μέλος x στο σύνολο P δεν υπάρχει λύση y (στη γειτονιά του x τέτοια ώστε $\|y - x\|_\infty \leq \epsilon$, όπου ϵ είναι ένας πολύ μικρός θετικός αριθμός) που να επικυριαρχεί οποιοδήποτε μέλος του συνόλου P , τότε οι λύσεις που ανήκουν στο P δημιουργούν ένα τοπικά Pareto βέλτιστο σύνολο. Στο παρακάτω σχήμα παρουσιάζεται ένα παράδειγμα τοπικού αντιδιαστολή με το ολικά Pareto βέλτιστο σύνολο. Τέλος, όπως φαίνεται και από το σχήμα, ένα τοπικό Pareto βέλτιστο δεν περιέχει απαραίτητα όλες τις Pareto βέλτιστες λύσεις και κάθε καθολικό Pareto βέλτιστο σύνολο είναι συγχρόνως και τοπικό βέλτιστο σύνολο.



Εικόνα 3.2 Τοπικό και ολικό Pareto βέλτιστο σύνολο λύσεων. Πηγή: (Zitzler 1999)

3.3.4 Βελτιστοποίηση υπό μελέτης διάταξης

Η βελτιστοποίηση της υπό μελέτης διάταξης που παρουσιάζεται στην ενότητα 4 αφορά πέντε παραμέτρους σχεδιασμού και λειτουργίας:

1. ύψος δεξαμενής ζεστού νερού χρήσης (ZNX), συμβολισμός: H [m]
2. επιφάνεια επίπεδου ηλιακού συλλέκτη, συμβολισμός: A_c [m²]
3. είδος επίπεδου ηλιακού συλλέκτη
4. παροχή νερού της αντλίας θερμότητας, συμβολισμός: \dot{m}_{hp} [kg/sec]
5. όριο θερμοκρασίας λειτουργίας της αντλίας θερμότητας, συμβολισμός: $T_{HP,set}$ [°C]

Τα δύο κριτήρια απόφασης του προβλήματος βελτιστοποίησης είναι:

- A. η ελαχιστοποίηση της συνεισφοράς της αντλίας θερμότητας στη δεξαμενή ZNX, $Q_{HP,cons}$ [kWh]
- B. η μεγιστοποίηση του μέσου ποσοστού κάλυψης των φορτίων ZNX για την ικανοποίηση των χρηστών, DHW_{cov} [%].

Το ποσοστό κάλυψης DHW_{cov} δίνεται από τη σχέση:

$$DHW_{cov} = \frac{\sum_{i=1}^N T_{dp}}{N \cdot T_{znx,set}} \quad (3.1)$$

όπου $\sum_{i=1}^N T_{dp}$: άθροισμα των θερμοκρασιών του νερού που πάει στην καταβάθρα θερμότητας της δεξαμενής [°C]

N: πλήθος των θερμοκρασιών

$T_{znx,set}$: οριακή τιμή επιθυμητής θερμοκρασίας ZNX [°C].

Η βελτιστοποίηση πραγματοποιήθηκε για όλη τη διάρκεια ενός έτους, εκτελώντας 86,400 χρονικές επαναλήψεις ανά ημέρα (χρονικό βήμα = 1 sec).

4. Διάταξη θέρμανσης υπό εξέταση

4.1 Περιγραφή

Η υπό μελέτη διάταξη περιλαμβάνει τα εξής συστήματα:

- έναν επίπεδο ηλιακό συλλέκτη
- μία αντλία νερού-νερού (γεωθερμική)
- δύο δεξαμενές αποθήκευσης ζεστού νερού
- δύο καταβόθρες θερμότητας
- τρεις εναλλάκτες θερμότητας.

Η εξέταση της διάταξης έγινε σε θεωρητικό επίπεδο και για τους υπολογισμούς κάποιων φορτίων της μελέτης και εν συνεχεία της βελτιστοποίησης των παραμέτρων έγιναν δύο υποθέσεις:

- η πόλη στην οποία βρίσκεται το κτίριο εγκατάστασης είναι η Αθήνα και
- το κτίριο πρόκειται για κτίριο το οποίο έχει γραφεία.

Η πρώτη υπόθεση έγινε διότι κάποιες οι εξισώσεις που περιγράφονται στη συνέχεια διαθέτουν όρους (θερμοκρασία εδάφους, ηλιακή ακτινοβολία), οι οποίοι προέρχονται από τα μετεωρολογικά στοιχεία μίας περιοχής.

Η δεύτερη υπόθεση έγινε για τον υπολογισμό των απαιτούμενων φορτίων κατανάλωσης ZNX.

Σε αυτό το σημείο αξίζει να σημειωθεί ότι ο αρχικός σχεδιασμός της διάταξης βασίστηκε σε μία υφιστάμενη διάταξη του Κέντρου Περιβαλλοντικής Εκπαίδευσης Βελβεντού (Κ.Π.Ε. Βελβεντού, Δυτική Μακεδονία, Κοζάνη). Η συγκεκριμένη διάταξη περιλαμβάνει μια γεωθερμική αντλία θερμότητας, τρεις δεξαμενές αποθήκευσης ζεστού νερού, μία σειρά από επίπεδους ηλιακούς συλλέκτες, έναν καυστήρα πετρελαίου και μία δεξαμενή αποθήκευσης πετρελαίου. Το κτίριο του Κ.Π.Ε. Βελβεντού αναπαρίσταται στην εικόνα 4.1 που ακολουθεί. Παράλληλα, στις εικόνες 4.2 – 4.6, παρουσιάζονται στοιχεία της θερμοϋδραυλικής εγκατάστασης.



Εικόνα 4.1 Κέντρο Περιβαλλοντικής Εκπαίδευσης Βελβεντού



Εικόνα 4.2 Σωληνώσεις.



Εικόνα 4.3 Δεξαμενή αποθήκευσης ζεστού νερού προς κατανάλωση χρηστών.



Εικόνα 4.4 Αντλία θερμότητας.



Εικόνα 4.5 Λέβητας πετρελαίου.



Εικόνα 4.6 Δεξαμενές αποθήκευσης ζεστού νερού προς θέρμανση κτιρίου.

4.2 Προσομοίωση της υπό μελέτη διάταξης

4.2.1 Εισαγωγή – Συνδεσμολογία διάταξης

Στην ενότητα αυτή παρουσιάζεται η προσομοίωση της υπό μελέτη διάταξης στο Spyder, το οποίο είναι ένα λογισμικό ανοιχτού πηγαίου κώδικα και η λειτουργία του θα αναλυθεί παρακάτω. Αρχικά γίνεται η περιγραφή όλων των συστημάτων που χρησιμοποιήθηκαν καθώς και οι απαραίτητες εξισώσεις που αφορούν τα εκάστοτε συστήματα. Όλες οι εξισώσεις που χρησιμοποιήθηκαν για τον υπολογισμό των φορτίων θέρμανσης βασίστηκαν στο εγχειρίδιο TRNSYS16 (Transient System Simulation Program) (Klein and Beckman 2007). Στο συγκεκριμένο εγχειρίδιο παρουσιάζονται πληροφορίες σχετικές με τη μαθηματική συμπεριφορά διάφορων εφαρμογών, μερικές εκ' των οποίων είναι:

- συστήματα HVAC
- συστήματα εναλλαγής θερμότητας
- συστήματα ανανεώσιμων πηγών ενέργειας (ηλιακά, γεωθερμικά, φωτοβολταϊκά).

4.2.2 Δεξαμενές αποθήκευσης

Οι δύο δεξαμενές αποθήκευσης που περιλαμβάνει η διάταξη είναι μία δεξαμενή αποθήκευσης του ZNX και μία δεξαμενή αποθήκευσης της περίσσειας θερμικής ισχύος του συλλέκτη. Η δεύτερη δεξαμενή έχει συνολική χωρητικότητα ρευστού 500 lt. Η μάζα του υγρού της δεξαμενής αποθήκευσης του ZNX προκύπτει από τη σχέση:

$$m = d \cdot V \quad [\text{kg}] \quad (4.1)$$

όπου d : πυκνότητα του ρευστού, για το νερό $d = 998 \text{ [kg/m}^3\text{]}$

V : όγκος που καταλαμβάνει το ρευστό $[\text{m}^3]$.

Ο όγκος που καταλαμβάνει το ρευστό στην δεξαμενή είναι ίσος με τον όγκο της δεξαμενής, η οποία έχει κυλινδρικό σχήμα, και ο υπολογισμός του γίνεται βάσει της σχέσης:

$$V = 2 \cdot \pi \cdot R \cdot H + \pi \cdot R^2 \quad (4.2)$$

όπου R : ακτίνα της κυκλικής επιφάνειας της δεξαμενής $[\text{m}]$

H : ύψος της δεξαμενής $[\text{m}]$.

Το ύψος H είναι και η πρώτη παράμετρος η οποία θα βελτιστοποιηθεί στη συνέχεια.

Και οι δύο δεξαμενές πρόκειται για δεξαμενές αποθήκευσης στρωματοποιημένων υγρών, που αποτελούνται από δέκα οριζόντια στρώματα (nodes), και διαθέτουν τριών ειδών φορτία:

- τα φορτία από/προς το περιβάλλον μέσω των τοιχωμάτων της δεξαμενής
- τα φορτία μεταφοράς μεταξύ των στρωμάτων και
- τα φορτία που προσδίδει το θερμαντικό στοιχείο.

Τα φορτία από/προς το περιβάλλον μέσω από τη δεξαμενή συμβολίζονται ως Q_{env} [W] και δίνονται από τη σχέση:

$$Q_{env} = \sum_{i=1}^N U \cdot A_i \cdot (T_i - T_{env}) + \gamma_f \cdot \sum_{i=1}^N (U \cdot A)_{f,i} \cdot (T_i - T_f) \quad (4.3)$$

όπου U: ο συντελεστής θερμοπερατότητας των τοιχωμάτων της δεξαμενής [W/(m²·K)]

A: το εμβαδόν του τμήματος i [m²]

T_i: η θερμοκρασία του τμήματος i της δεξαμενής [°C]

T_{env}: η θερμοκρασία του περιβάλλοντος που περιβάλλει την δεξαμενή [°C]

γ_f: ο συντελεστής ελέγχου λειτουργίας του βοηθητικού θερμαντήρα [0 ή 1]

(U·A)_{f,i}: αγωγιμότητα για απώλεια θερμότητας σε καυσαέριο στον κόμβο i [W/°C]

T_f: μέση θερμοκρασία καυσαερίων όταν ο θερμαντήρας δεν λειτουργεί [°C].

Ο δεύτερος όρος της εξίσωσης 5.1 αφορά τα φορτία που προσδίδει η καπνοδόχος των καυσαερίων στη δεξαμενή. Οι δύο δεξαμενές που διαθέτει η υπό μελέτη διάταξη δεν έχουν καπνοδόχο και έτσι ο όρος αυτός μηδενίζεται.

Τα φορτία μεταφοράς μεταξύ των στρωμάτων της δεξαμενής συμβολίζονται ως $Q_{transfer}$ [W] και δίνονται από τη σχέση:

$$Q_{transfer} = \dot{m}_i \cdot C_{pf} \cdot (T_{i-1} - T_i) \quad (4.4)$$

όπου \dot{m}_i : παροχή μάζας του ρευστού στο τμήμα i [kg/sec]

C_{pf}: ειδική θερμότητα του ρευστού της δεξαμενής [J/(kg·°C)]

T_{i-1}: θερμοκρασία του τμήματος i-1 της δεξαμενής [°C]

T_i: θερμοκρασία του τμήματος i της δεξαμενής [°C].

Τα φορτία που προσδίδει το θερμαντικό στοιχείο στη δεξαμενή συμβολίζονται με Q_i [W]. Στις δύο υπό μελέτη δεξαμενές το θερμαντικό στοιχείο είναι ένας εναλλάκτης θερμότητας στην κάθε μία. Οι σχέσεις υπολογισμού των φορτίων θα δοθούν στη συνέχεια στην υποενότητα 4.2.6.

Σύμφωνα με όλες τις εξισώσεις που διέπουν τις δεξαμενές αποθήκευσης, η συνολική σχέση που περιγράφει την θερμική συμπεριφορά της κάθε δεξαμενής είναι το άθροισμα όλων των προαναφερθέντων φορτίων και η μαθηματική αποτύπωση της είναι:

$$Q_{BT} = Q_{env} + Q_{transfer} + Q_i \text{ [W]} \quad (4.5)$$

4.2.3 Επίπεδος ηλιακός συλλέκτης

Η μελέτη πραγματοποιήθηκε για πέντε διαφορετικούς τύπους επίπεδου ηλιακού συλλέκτη:

- Επίπεδος συλλέκτης με μαύρο χρώμα και έναν υαλοπίνακα

- Επίπεδος συλλέκτης με μαύρο χρώμα και δύο υαλοπίνακες ή με επιλεκτική βαφή και έναν υαλοπίνακα
- Επίπεδος συλλέκτης με σωλήνες κενού-αέρος
- Απλός επίπεδος συλλέκτης με πλαστικούς σωλήνες, χωρίς κάλυμμα και μόνωση

Η ωφέλιμη θερμική ισχύς από έναν επίπεδο ηλιακό συλλέκτη δίνεται από την σχέση:

$$Q_u = A_c \cdot \dot{m} \cdot C_p \cdot (T_{co} - T_{ci}) = [A_c \cdot (\tau \cdot \alpha)_n \cdot I_T \cdot F_R] - [A_c \cdot F_R \cdot U_L \cdot (T_{ci} - T_a)] \quad (4.6)$$

όπου Q_u : ωφέλιμη θερμική ισχύς [W]

\dot{m} : παροχή του εργαζόμενου μέσου σε συνθήκες χρήσης [kg/sec]

C_p : ειδική θερμοχωρητικότητα ρευστού [J/(kg°C)], για το νερό το $C_p = 4180$ J/(kg°C)

A_c : επιφάνεια συλλέκτη [m^2]

T_{co} : θερμοκρασία του εξόδου του εργαζόμενου μέσου από τον συλλέκτη [°C]

T_{ci} : θερμοκρασία εισόδου του εργαζόμενου μέσου στο συλλέκτη [°C]

I_T : συνολική προσπίπτουσα ηλιακή ακτινοβολία ανά επιφάνεια συλλέκτη [W/m^2]

T_a : θερμοκρασία περιβάλλοντος [°C]

F_R : συντελεστής θερμικής απολαβής του συλλέκτη

τ : η διαπερατότητα στην ηλιακή ακτινοβολία του καλύμματος του συλλέκτη

α : η απορροφητικότητα στην ηλιακή ακτινοβολία

U_L : ο συντελεστής ολικών θερμικών απωλειών του συλλέκτη [$W/(m^2 \cdot K)$]

Ο πρώτος όρος της σχέσης (4.6) εκφράζει τα θερμικά κέρδη από την προσπίπτουσα ακτινοβολία λαμβάνοντας υπόψη τις οπτικές απώλειες ενώ ο δεύτερος όρος εκφράζει τις θερμικές απώλειες του συλλέκτη.

Ο βαθμός απόδοσης του συλλέκτη δίνεται από την σχέση:

$$\eta = \frac{Q_u}{A_c \cdot I_T} = F_R \cdot (\tau \cdot \alpha)_n - F_R \cdot U_L \cdot \frac{(T_{ci} - T_a)}{I_T} \quad (4.7)$$

Ο συντελεστής $(\tau \cdot \alpha)_n$ αναφέρεται σε επίπεδο κάθετο στις ακτίνες του ήλιου.

Τυπικές τιμές των χαρακτηριστικών ντου επίπεδου ηλιακού συλλέκτη δίνονται στον πίνακα 4.1 που ακολουθεί.

Πίνακας 4.1 Τυπικές τιμές των χαρακτηριστικών επίπεδου ηλιακού συλλέκτη. Πηγή: (Μπέλλος 2016)

Απορροφητικότητα του απορροφητήρα	0.85-0.95
Διαπερατότητα του καλύμματος	0.88-0.95
Συντελεστή εκπομπής επιλεκτικού απορροφητήρα	0.05-0.15
Συντελεστή εκπομπής μη επιλεκτικού απορροφητήρα	0.90-0.98
Συντελεστής εκπομπής καλύμματος	0.88-0.90
Αγωγιμότητα μονωτικού υλικού (W/m K)	0.04
Αγωγιμότητα απορροφητήρα (W/m K)	210-385
Πάχος μόνωσης (mm)	4.0-5.0
Πάχος πλάκας απορρόφησης (mm)	0.5-1.0
Διάκενο μεταξύ πλάκας και καλύμματος (mm)	3-4
Συλλεκτική επιφάνεια (m ²)	1.5- 2.5
Αριθμός σωλήνων	8-11
Ανηγγμένη παροχή μάζας ανά m ² συλλεκτικής επιφάνειας (kg/m ² s)	0.01-0.03
Εσωτερική διάμετρος αγωγού (mm)	8-12
Εξωτερική διάμετρος αγωγού (mm)	10-15

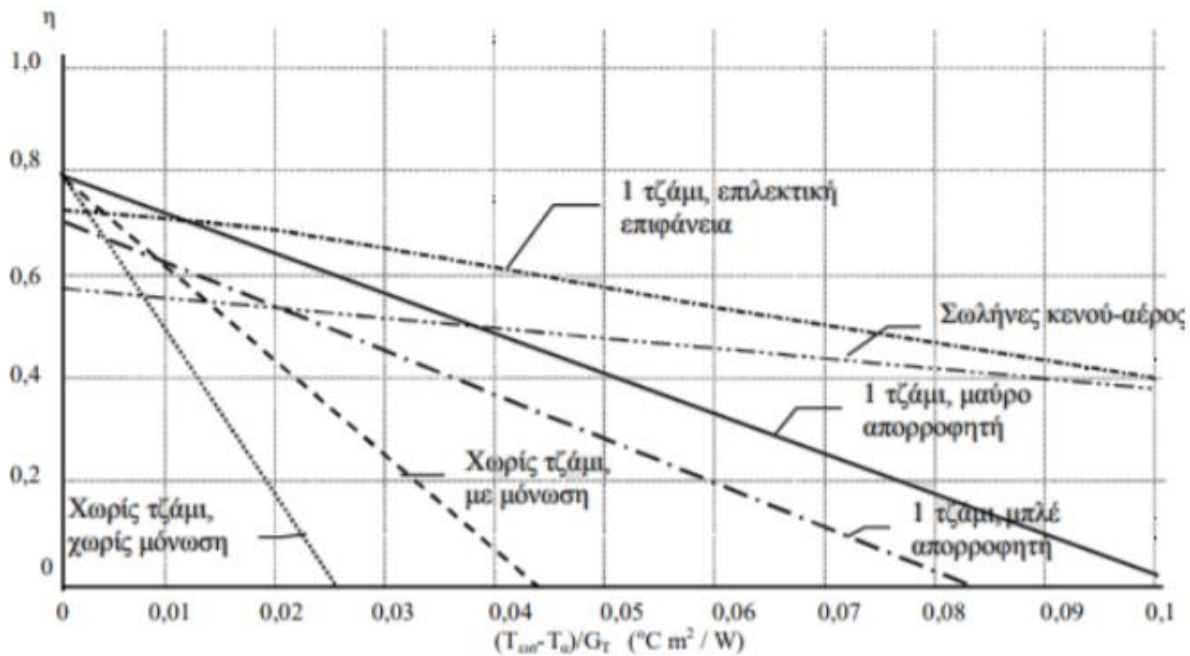
Οι τιμές των γινομένων $F_R \cdot (\tau \cdot \alpha)_n$ και $F_R \cdot U_L$ εξαρτώνται από τα χαρακτηριστικά του συλλέκτη όπως το χρώμα του (αν είναι επιλεκτικός ή όχι), τον αριθμό των υάλινων καλυμμάτων, την απορροφητικότητα του απορροφητήρα του, τη διαπερατότητα του καλύμματος του κ.α.. Για αυτό και αυτά τα χαρακτηριστικά είναι κρίσιμα για την απόδοση του συλλέκτη. Μερικές τυπικές τιμές αυτών των παραμέτρων που προέκυψαν από πειράματα, δοκιμές απόδοσης και πιστοποίησης από αναγνωρισμένο εργαστήριο δοκιμών (Κέντρο Δοκιμών Ηλιακών Συλλεκτών του Δημόκριτου), αναπαρίστανται στον πίνακα που ακολουθεί.

Πίνακας 4.2 Τυπικές τιμές παραμέτρων απόδοσης διάφορων τύπων επίπεδου ηλιακού συλλεκτών. Πηγή: (Βραχόπουλος, et al. 2011)

Περιγραφή ηλιακού συλλέκτη	$F_R \cdot (\tau \cdot \alpha)_n$	$F_R \cdot U_L$ [W/m ² °C]
Μαύρο χρώμα με μονό υαλοπίνακα	0.82	7.5
Μαύρο χρώμα, με διπλό υαλοπίνακα ή επιλεκτική επιφάνεια με μονό υαλοπίνακα	0.75	5
Σωλήνες κενού-αέρος	0.57	1.82
Απλός συλλέκτης (πλαστικοί σωλήνες) χωρίς κάλυμμα και μόνωση (ταχύτητα ανέμου 2.2 m/s)	0.86	21.5

Από τον πιο πάνω πίνακα, παρατηρείται ότι ο απλός τύπου συλλέκτης χωρίς κάλυμμα και μόνωση, έχει τιμή $F_R \cdot (\tau \cdot \alpha)_n$ μεγαλύτερη από ότι άλλοι τύποι και δηλαδή έχει μεγαλύτερα θερμικά κέρδη από την προσπίπτουσα ηλιακή ακτινοβολία, ενώ η τιμή του $F_R \cdot U_L$ που είναι πολύ μεγαλύτερη από ότι στους άλλους τύπους συλλέκτη δείχνει ότι έχει πολύ μεγαλύτερες θερμικές απώλειες.

Σύμφωνα με την σχέση (4.7) ο βαθμός απόδοσης του συλλέκτη είναι γραμμική συνάρτηση του μεγέθους $(T_m - T_a)/G_T$ με κλίση $F_R \cdot U_L$ και έχει τομή με τον άξονα των τεταγμένων στη θέση $F_R \cdot (\tau \cdot \alpha)_n$. Όλα αυτά παρουσιάζονται στο διάγραμμα 4.2 που ακολουθεί.



Διάγραμμα 4.1 Καμπύλες απόδοσης για διάφορους τύπους επίπεδου ηλιακού συλλέκτη. Πηγή: (Βραχόπουλος, et al. 2011)

Οι τιμές των παραμέτρων απόδοσης όλων των συλλεκτών που χρησιμοποιήθηκαν στην υπό μελέτη διάταξη είναι οι τιμές του πίνακα 4.2. Όπως προκύπτει και από τον πίνακα, ο επίπεδος συλλέκτης με μαύρο χρώμα και δύο υαλοπίνακες και ο επίπεδος συλλέκτης με επιλεκτική βαφή και έναν υαλοπίνακα έχουν τις ίδιες τιμές παραμέτρων απόδοσης. Το εργαζόμενο μέσο των ηλιακών συλλεκτών είναι το νερό.

Ο ηλιακός συλλέκτης έχει ως προτεραιότητα την πρόσδοση θερμότητας στο δοχείο του ZNX μέχρις ότου να φτάσει το νερό της δεξαμενής τους 50 °C, που είναι και η απαιτούμενη θερμοκρασία του νερού προς χρήση. Στη συνέχεια την περίσσεια θερμική ισχύ την προσδίδει στη δεξαμενή των 500 lt. Είναι προφανές το γεγονός ότι ο συλλέκτης δεν προσδίδει ενέργεια καθ' όλη τη διάρκεια ενός 24ώρου, διότι η ηλιακή ακτινοβολία είναι διαθέσιμη για συγκεκριμένες ώρες της ημέρας.

Η ενέργεια που προσδίδει ο συλλέκτης δίνεται από τη σχέση 4.6, και ο βαθμός απόδοσης του συλλέκτη από τη σχέση 4.7. Η συνολική προσπίπτουσα ηλιακή ακτινοβολία (I_T) και η θερμοκρασία περιβάλλοντος (T_a) που αναγράφονται στην σχέση αυτή επιλέχθηκαν τιμές για την πόλη της Αθήνας (Argiriou, et al. 1999). Οι τιμές αυτές είναι ωριαίες και δίνονται για όλο το έτος. Για την θερμοκρασία περιβάλλοντος έγινε μία μείωση κατά 5 °C για να είναι πιο αντιπροσωπευτικές οι τιμές.

Η θερμοκρασία του εργαζόμενου μέσου στην έξοδο του συλλέκτη (T_{co}) διαφοροποιείται ανάλογα με το αν ο συλλέκτης προσδίδει ενέργεια στις δεξαμενές ή όχι. Όταν έχουμε πρόσδοση θερμότητας, η θερμοκρασία εξόδου υπολογίζεται από την σχέση:

$$T_{co} = T_{co-1} + \frac{Q_u \cdot dt}{m_c \cdot C_p} \quad (4.8)$$

όπου T_{co-1} : θερμοκρασία εξόδου του ρευστού την προηγούμενη χρονική στιγμή [°C]

Q_u : θερμική ισχύς του συλλέκτη [W], (σχέση 3.1)

dt : χρονικό βήμα [sec]

m_c : συνολική μάζα του ρευστού στον συλλέκτη [kg]

C_p : ειδική θερμοχωρητικότητα ρευστού [J/kg°C], για το νερό το $C_p = 4180$ J/kg°C.

Στην περίπτωση που δεν έχουμε πρόσδοση θερμότητας από τον συλλέκτη, η θερμοκρασία του ρευστού στην έξοδο δίνεται από τη σχέση:

$$T_{co} = T_{co-1} + \frac{Q_{change} \cdot dt}{m_c \cdot C_p} = T_{co-1} + \frac{[\dot{m} \cdot C_p \cdot (T_{ci} - T_{co})] \cdot dt}{m_c \cdot C_p} \quad (4.9)$$

όπου T_{co-1} : θερμοκρασία εξόδου την προηγούμενη χρονική στιγμή [°C]

\dot{m} : παροχή του ρευστού προς τις δεξαμενές [kg/sec]

T_{ci} : θερμοκρασία του ρευστού στην είσοδο του συλλέκτη [°C]

T_{co} : θερμοκρασία του ρευστού στην έξοδο του συλλέκτη [°C].

Η διαφοροποίηση αυτή στην θερμοκρασία του ρευστού στην έξοδο του συλλέκτη γίνεται διότι ο συλλέκτης διαθέτει ένα μηχανισμό θερμικής αδράνειας, ανάλογης μεν, αλλά πολύ μικρότερης τάξης μεγέθους δε από αυτή ενός δοχείου. Πιο συγκεκριμένα, όταν κλείνει η παροχή του ρευστού προς τις δεξαμενές η συνολική μάζα (m_c) που παραμένει στους εσωτερικούς σωλήνες του συλλέκτη θερμαίνεται ή ψύχεται ανάλογα με την θερμοκρασιακή διαφορά ανάμεσα στο ρευστό και στο περιβάλλον. Το μέγεθος της επιφάνειας του συλλέκτη (A_c) είναι και η δεύτερη παράμετρος η οποία θα βελτιστοποιηθεί στη συνέχεια.

Τέλος, αξίζει να σημειωθεί ότι κατά τις περιόδους όπου δεν έχουμε ποσό προσπίπτουσας ηλιακής ακτινοβολίας στο συλλέκτη, ο βαθμός απόδοσής του θεωρείται μηδέν. Αυτό είναι μία παραδοχή που έγινε για την αποφυγή εύρεσης αδύνατης λύσης στην υπό μελέτη διάταξη, διότι η ηλιακή ακτινοβολία, σύμφωνα με την σχέση 4.7, βρίσκεται στον παρονομαστή του κλάσματος υπολογισμού του βαθμού απόδοσης.

4.2.4 Αντλία νερού-νερού (γεωθερμική)

Η αντλία θερμότητας της διάταξης πρόκειται για μία αντλία νερού-νερού (γεωθερμική) με ονομαστική ηλεκτρική ισχύ $Q_{ei} = 1000$ W. Σύμφωνα με την σχέση 2.10 προηγούμενης ενότητας, η ισχύς που αποδίδει η αντλία συνδέεται με το συντελεστή απόδοσης COP με την εξής σχέση:

$$Q_H = COP \cdot Q_{ei} \quad (4.10)$$

Ο συντελεστής απόδοσης είναι άμεσα συνδεδεμένος με την διαφορά ανάμεσα στην θερμοκρασία εξόδου του ρευστού από την αντλία και την θερμοκρασία του εδάφους. Η σχέση που δίνει τον βαθμό απόδοσης COP είναι (Mouzeviris και Parakostas 2020):

$$\text{COP} = 0.0015 \cdot (T_{wo} - T_{gs})^2 - 0.2172 \cdot (T_{wo} - T_{gs}) + 9.9075 \quad (4.11)$$

όπου T_{wo} : θερμοκρασία εξόδου του ρευστού από την αντλία [°C]

T_{gs} : θερμοκρασία εδάφους [°C] (Τ.Ο.Τ.Ε.Ε. 20701-3 2010).

Η θερμοκρασία εξόδου του ρευστού από την αντλία δίνεται από την σχέση:

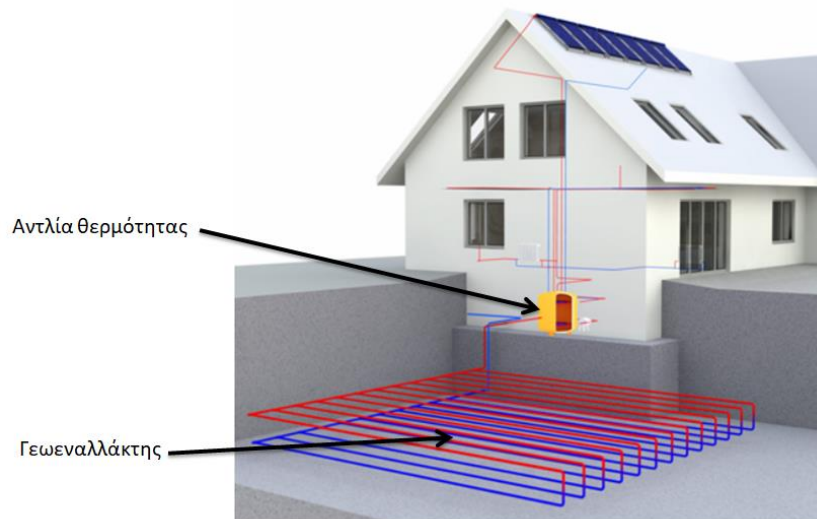
$$T_{wo} = \frac{Q_H}{\dot{m}_{hp} \cdot C_p} + T_{wi} \quad (4.12)$$

όπου Q_H : ισχύς που αποδίδει η αντλία [W]

\dot{m}_{hp} : παροχή μάζας του ρευστού [kg/sec]

C_p : ειδική θερμοχωρητικότητα ρευστού [J/kg°C], για το νερό το $C_p = 4180 \text{ J/kg°C}$

T_{wi} : θερμοκρασία εισόδου του ρευστού στην αντλία [°C].



Εικόνα 4.7 Αντλία θερμότητας. Πηγή: (Rossato 2020)

Η αντλία θερμότητας είναι συνδεδεμένη με τη δεξαμενή αποθήκευσης του ZNX. Αυτό σημαίνει ότι η αντλία δουλεύει ως βοηθητική πηγή όταν η απαιτούμενη θερμοκρασία του ZNX δεν καλύπτεται από τον ηλιακό συλλέκτη. Μία ακόμη παράμετρος της αντλίας είναι το όριο της θερμοκρασίας λειτουργίας της ($T_{HP,set}$), το οποίο είναι ουσιαστικά η θερμοκρασία μέχρι την οποία προσδίδει θερμότητα στη δεξαμενή αποθήκευσης του ZNX η αντλία θερμότητας. Αυτή η θερμοκρασία καθώς και η παροχή μάζας του ρευστού της αντλίας είναι άλλες δύο παράμετροι που θα βελτιστοποιηθούν.

4.2.5 Καταβόθρες Θερμότητας

Οι δύο καταβόθρες της διάταξης αντιπροσωπεύουν την ύπαρξη ανοιχτού κυκλώματος ρευστού το οποίο αλληλεπιδρά με το υπό εξέταση σύστημα. Για ευκολία κατανόησης, αντικαταστάθηκαν οι σωληνώσεις ανοιχτής ροής με τις καταβόθρες θερμότητας, ώστε το σύστημα να θεωρείται πλέον κλειστό. Και οι δύο δεξαμενές έχουν από μία καταβόθρα αντίστοιχα.

Η καταβόθρα του δοχείου του ZNX είναι ουσιαστικά η κατανάλωση ZNX στο κτίριο. Σύμφωνα με μία μελέτη που πραγματοποιήθηκε στην Κορέα (Chunga και Park 2015), επειδή στα κτίρια γραφείων η κατανάλωση ZNX είναι πολύ μικρή σε σχέση με τα φορτία θέρμανσης που απαιτούνται στο κτίριο, συνυπολογίζεται μαζί με αυτά. Για τη διευκόλυνση των υπολογισμών, επειδή η έρευνα που έγινε στην παρούσα εργασία δεν εξέτασε τα συνολικά φορτία θέρμανσης που απαιτούνται σε ένα κτίριο γραφείων, η κατανάλωση ZNX ανά εργαζόμενο αναπαρίσταται στον πίνακα που ακολουθεί.

Πίνακας 4.3 Ειδικές ημερήσιες απαιτήσεις νερού.

Κτίρια γραφείων ανά εργαζόμενο			
Ολική ποσότητα απαιτούμενου νερού (lt/ημέρα)	Απαιτούμενη ποσότητα κρύου νερού (lt/ημέρα)	Απαιτούμενη ποσότητα ZNX(lt/ημέρα)	Θερμοκρασία ZNX [°C]
250 – 30	15 – 20	5 - 10	45 - 50

Για τους υπολογισμούς των φορτίων επιλέχθηκε ως ημερήσια απαιτούμενη ποσότητα ZNX τα 10 lt/ημέρα. Αυτά τα 10 lt διαμοιράστηκαν σε ωριαία ποσά τα οποία έγιναν βάσει των ωρών λειτουργίας του κτιρίου που προκύπτουν από την τεχνική οδηγία του τεχνικού επιμελητηρίου (T.O.T.E.E. 20701-1 2017). Επίσης ως θερμοκρασία του ZNX επιλέχθηκαν οι 50 °C.

Η καταβόθρα που είναι συνδεδεμένη με τη δεξαμενή αποθήκευσης της περίσσειας θερμικής ισχύος του συλλέκτη, είναι ένας χώρος στον οποίο οδηγείται το θερμό νερό που βρίσκεται μέσα στη δεξαμενή. Αυτή η καταβόθρα τοποθετήθηκε στη διάταξη για τη σχεδόν πλήρη απομάστευση του υγρού της δεξαμενής. Αυτό έχει ως αποτέλεσμα την αποφυγή υπερθέρμανσης του νερού του συλλέκτη. Αυτή η απομάστευση του υγρού γίνεται με ωριαίο βήμα και η ποσότητα του νερού που φεύγει προς την καταβόθρα είναι 20 lt/hr.

Η απομάστευση υγρού από τις δεξαμενές στις καταβόθρες ερμηνεύεται και ως ενεργειακές απώλειες των δύο δεξαμενών. Αυτές η απώλειες εκφράζονται ως Q_{loss} [W] και η σχέση του υπολογισμού τους είναι:

$$Q_{\text{loss}} = \dot{m}_w \cdot C_p \cdot (T_{i,d} - T_{o,d}) \quad (4.13)$$

όπου \dot{m}_w : παροχή μάζας του ρευστού [kg/sec]

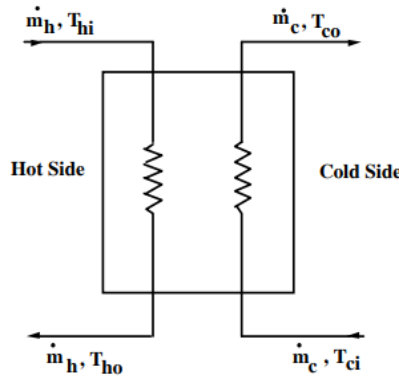
C_p : ειδική θερμοχωρητικότητα ρευστού [J/kg°C], για το νερό το $C_p = 4180$ J/kg°C

$T_{i,d}$: Θερμοκρασία που εξέρχεται το νερό από τη δεξαμενή [$^{\circ}\text{C}$]

$T_{o,d}$: θερμοκρασία που εισέρχεται το νερό στη δεξαμενή [$^{\circ}\text{C}$].

4.2.6 Εναλλάκτες θερμότητας

Το τελευταίο σύστημα που περιλαμβάνει η υπό μελέτη διάταξη είναι οι εναλλάκτες θερμότητας. Ο εναλλάκτης θερμότητας είναι μία συσκευή διεργασιών που επιτρέπει τη μεταφορά θερμικής ενέργειας μεταξύ δύο ρευστών που έχουν διαφορετική θερμοκρασία (Πάπιστα 2020).



Εικόνα 4.8 Εναλλάκτης θερμότητας. Πηγή: (Klein and Beckman 2007)

Η διάταξη περιέχει τρεις εναλλάκτες θερμότητας οι οποίοι είναι τοποθετημένοι μέσα στις δεξαμενές, οι δυο στην δεξαμενή αποθήκευσης του ZNX και ο ένας στη δεξαμενή των 500lt. Το μέγεθος τους είναι ίσο με τα δέκα οριζόντια στρώματα στα οποία επιμερίζονται και οι δεξαμενές. Όπως αναφέρθηκε και προηγουμένως, τα φορτία που προσδίδει ο εναλλάκτης συμβολίζονται με Q_i [W] και δίνονται από τη σχέση:

$$Q_i = \varepsilon \cdot Q_{\max} \quad (4.14)$$

όπου ε : αποτελεσματικότητα του εναλλάκτη θερμότητας

Q_{\max} : μέγιστος ρυθμός μεταφοράς θερμότητας ενός εναλλάκτη [W].

Ο μέγιστος ρυθμός μεταφοράς θερμότητας ενός εναλλάκτη εξαρτάται από το ρυθμό χωρητικότητας του ρευστού με την εξής σχέση:

$$Q_{\max} = \begin{cases} C_h \cdot (T_{hi} - T_{ci}), \text{ αν } C_{\min} = C_h \\ C_c \cdot (T_{hi} - T_{ci}), \text{ αν } C_{\min} = C_c \end{cases} \quad (4.15)$$

όπου C_h : ρυθμός χωρητικότητας του ρευστού στη ζεστή πλευρά του εναλλάκτη [$\text{J/s}^{\circ}\text{C}$]

C_c : ρυθμός χωρητικότητας του ρευστού στην κρύα πλευρά του εναλλάκτη [$\text{J/s}^{\circ}\text{C}$]

T_{hi} : θερμοκρασία εισόδου του ρευστού στη ζεστή πλευρά του εναλλάκτη [$^{\circ}\text{C}$]

T_{ci} : θερμοκρασία εισόδου του ρευστού στην κρύα πλευρά του εναλλάκτη [°C]

C_{min} : ελάχιστος ρυθμός χωρητικότητας [J/s°C].

Ο ρυθμός χωρητικότητας του ρευστού στη ζεστή και στην κρύα πλευρά δίνονται στις σχέσεις 4.16, 4.17 αντίστοιχα, που ακολουθούν:

$$C_h = \dot{m}_h \cdot C_{ph} \quad (4.16)$$

$$C_c = \dot{m}_c \cdot C_{pc} \quad (4.17)$$

όπου \dot{m}_h : παροχή ρευστού στη ζεστή πλευρά του εναλλάκτη [kg/sec]

\dot{m}_c : παροχή ρευστού στην κρύα πλευρά του εναλλάκτη [kg/sec]

C_{ph} : ειδική θερμοχωρητικότητα ρευστού που ρέει στη ζεστή πλευρά [J/kg°C]

C_{pc} : ειδική θερμοχωρητικότητα ρευστού που ρέει στην κρύα πλευρά [J/kg°C].

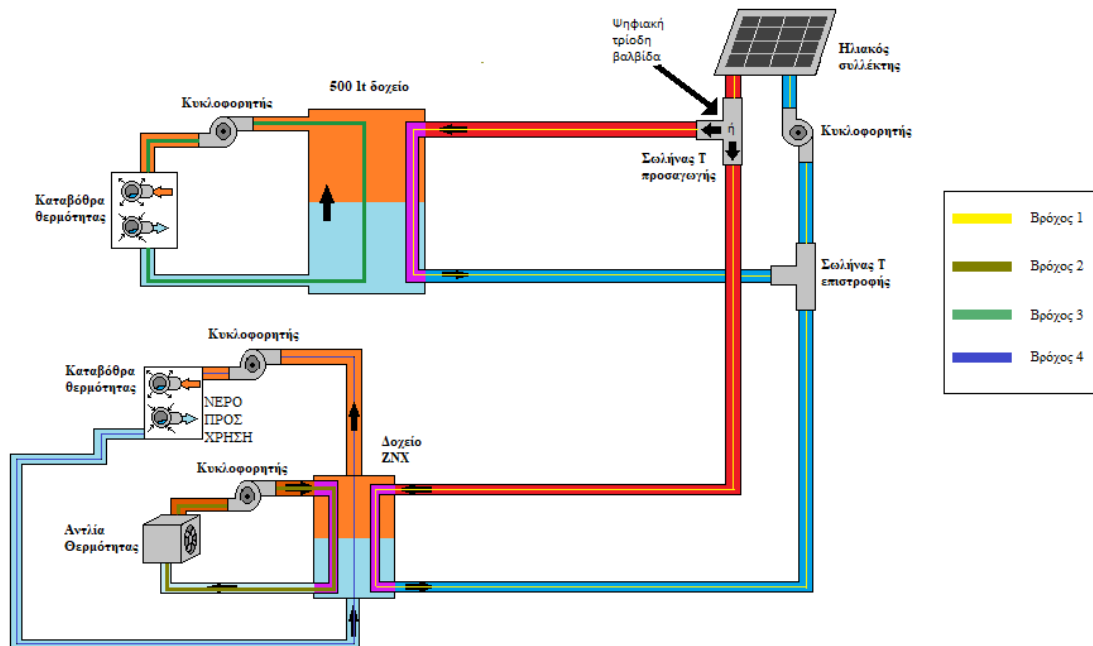
Τελευταίος υπολογισμός που αφορά τους εναλλάκτες θερμότητας, είναι οι θερμοκρασίες εξόδου για τα δύο ρευστά οι οποίες δίνονται από τις σχέσεις:

$$T_{ho} = T_{hi} - \frac{Q_i}{C_h} \quad [^\circ\text{C}] \quad (4.18)$$

$$T_{co} = T_{ci} - \frac{Q_i}{C_c} \quad [^\circ\text{C}] \quad (4.19)$$

4.2.7 Συνδεσμολογία εξεταζόμενης διάταξης

Στην εικόνα 4.3 που ακολουθεί αναπαρίσταται η συνδεσμολογία της εξεταζόμενης διάταξης.



Εικόνα 4.9 Συνδεσμολογία εξεταζόμενης διάταξης

Όπως φαίνεται και στην παραπάνω εικόνα (4.9) η συνδεσμολογία της εξεταζόμενης διάταξης περιέχει τέσσερις βρόχους. Ο κάθε βρόχος διαθέτει έναν κυκλοφορητή που είναι υπεύθυνος για την έναρξη και τη διακοπή της κίνησης του εργαζόμενου μέσου καθώς για την ρύθμιση της παροχής του. Τα λοιπά στοιχεία του κάθε βρόχου έχουν ως εξής:

Βρόχος 1: Ο πρώτος βρόχος περιλαμβάνει τον επίπεδο ηλιακό συλλέκτη, δύο εναλλάκτες θερμότητας (ένας σε κάθε δεξαμενή) και δύο σωλήνες T (έναν προσαγωγής και έναν επιστροφής), οι οποίοι είναι υπεύθυνοι για την ρύθμιση της πορείας του εργαζόμενου μέσου.

Βρόχος 2: Στο δεύτερο βρόχο ανήκουν η δεξαμενή αποθήκευσης του ZNX και η αντλία θερμότητας.

Βρόχος 3: Ο τρίτος βρόχος περιλαμβάνει την δεξαμενή αποθήκευσης των 500 lt και την καταβόθρα θερμότητας της συγκεκριμένης δεξαμενής.

Βρόχος 4: Ο τελευταίος βρόχος της διάταξης περιλαμβάνει έναν εναλλάκτη θερμότητας και την καταβόθρα θερμότητας της δεξαμενής του ZNX.

Εκτός από τα στοιχεία που διαθέτει ο κάθε βρόχος, αξίζει να σημειωθεί και η πορεία που διαγράφει το νερό μέσα στους βρόχους. Η κίνηση του νερού μέσα στους βρόχους ξεκινάει από τον κυκλοφορητή όταν αυτό ενεργοποιείται. Η πορεία του νερού στον κάθε βρόχο έχει ως εξής:

Βρόχος 1: Κυκλοφορητής → επίπεδος ηλιακός συλλέκτης → σωλήνας T προσαγωγής → εναλλάκτες θερμότητας → σωλήνας T επιστροφής → κυκλοφορητής.

Βρόχος 2: Κυκλοφορητής → δεξαμενή αποθήκευσης ZNX → αντλία θερμότητας → κυκλοφορητής.

Βρόχος 3: Κυκλοφορητής → καταβόθρα θερμότητας δεξαμενής 500 lt → δεξαμενή αποθήκευσης των 500 lt → κυκλοφορητής.

Βρόχος 4: Κυκλοφορητής → καταβόθρα θερμότητας της δεξαμενής ZNX → εναλλάκτης θερμότητας → κυκλοφορητής.

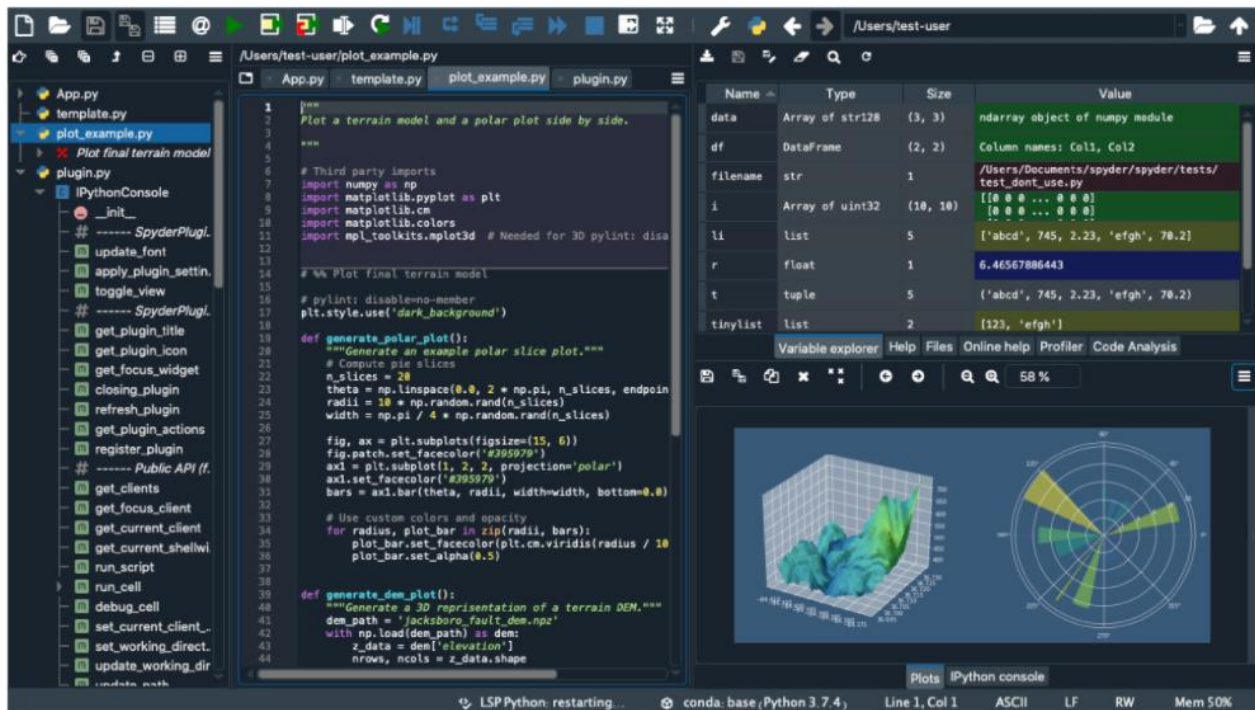
5. Μεθοδολογία βελτιστοποίησης διάταξης

5.1 Γενικά

Σε αυτήν την ενότητα αρχικά παρουσιάζονται κάποιες πληροφορίες για τα δύο λογισμικά που χρησιμοποιήθηκαν στην επίλυση του προβλήματος βελτιστοποίησης, και στη συνέχεια περιγράφονται όλα τα βήματα της μεθοδολογίας επίλυσης του προβλήματος.

5.2 Εισαγωγή στο Spyder-Anaconda

Το Spyder είναι ένα ελεύθερο και ανοιχτού κώδικα επιστημονικό περιβάλλον, που περιλαμβάνεται στην πλατφόρμα Anaconda, το οποίο είναι γραμμένο σε γλώσσα προγραμματισμού Python και χρησιμοποιεί και τη γλώσσα αυτή. Σχεδιάστηκε από και για επιστήμονες, μηχανικούς και αναλυτές δεδομένων. Διαθέτει ένα μοναδικό συνδυασμό της προηγμένης λειτουργικότητας επεξεργασίας, ανάλυσης, εντοπισμού σφαλμάτων και προφίλ ενός ολοκληρωμένου εργαλείου ανάπτυξης με την εξερεύνηση δεδομένων, τη διαδραστική εκτέλεση, τη βαθιά επιθεώρηση και τις όμορφες δυνατότητες οπτικοποίησης ενός επιστημονικού πακέτου. Στην εικόνα που ακολουθεί παριστάνεται το γραφικό περιβάλλον του προγράμματος Spyder (Spyder 2020).

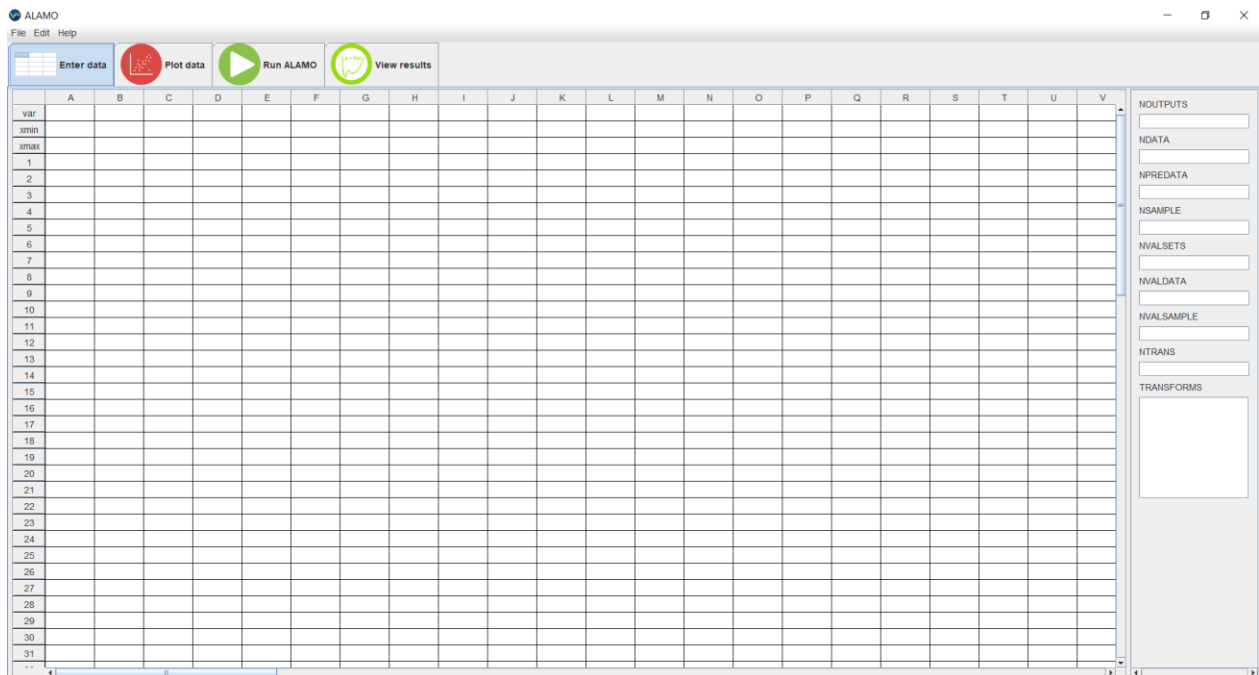


Εικόνα 5.1 Γραφικό περιβάλλον του προγράμματος Spyder.

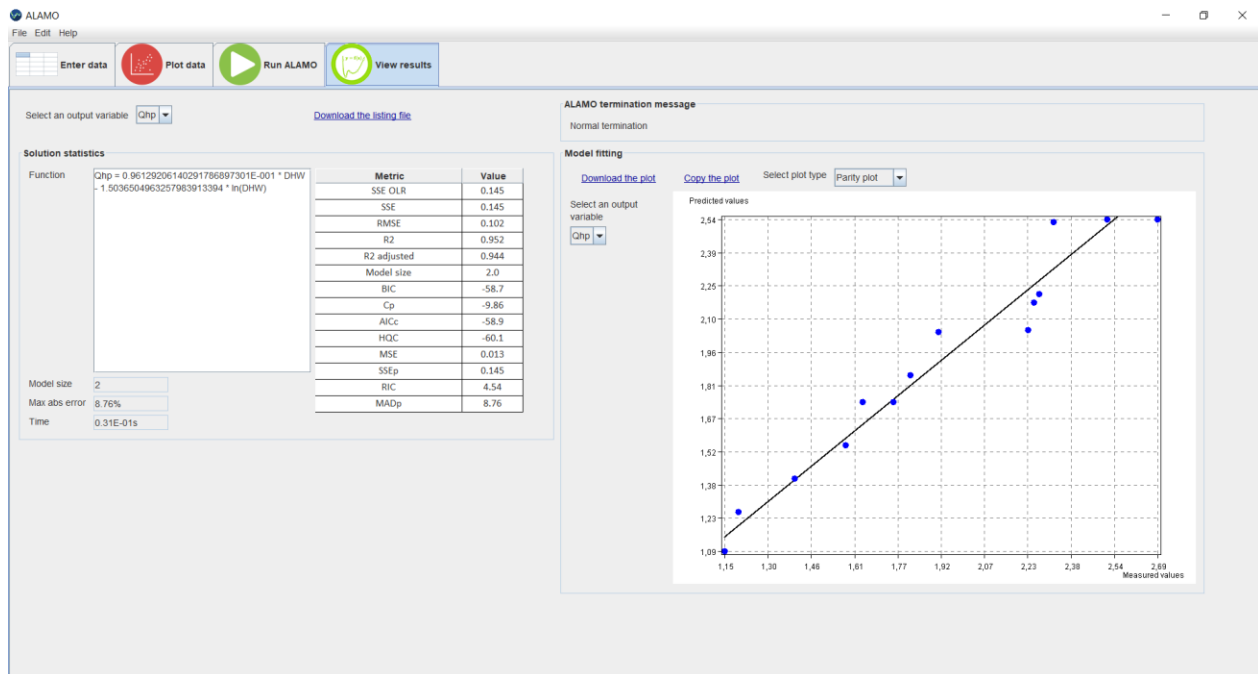
5.3 Εισαγωγήστολογισμικό ALAMO (Automated Learning of Algebraic Models)

Το λογισμικό ALAMO είναι ένα λογισμικό, το οποίο επιτρέπει στους χρήστες να κατασκευάσουν αλγεβρικά μοντέλα από δεδομένα, διασφαλίζοντας την ακρίβεια και την απλότητα αυτών των μοντέλων. Είναι το μοναδικό λογισμικό το οποίο μπορεί να επιβάλει φυσικούς περιορισμούς στη μηχανική μάθηση των μοντέλων. Το ALAMO βρίσκει εφαρμογές σε διάφορους τομείς μερικοί εκ' των οποίων είναι ο τομέας των μηχανικών, των χημικών στον τομέα της φαρμακευτικής καθώς και σε διάφορες βιομηχανίες. Σύμφωνα με μία μελέτη (Cozada, Sahinidisab and Miller 2015), η οποία σύγκρινε το ALAMO με ένα άλλο λογισμικό παρόμοιου τύπου, δείχνει ότι το λογισμικό ALAMO είναι απλούστερο και παράλληλα έχει μεγαλύτερη ακρίβεια στα αποτελέσματά του. Τέλος, αξίζει να σημειωθεί ότι το λογισμικό αυτό επιτρέπει την δημιουργία ενός υποκατάστατου μοντέλου με απεριόριστο μέγεθος προβλήματος και ταυτόχρονα απεριόριστο αριθμό χρήσεων (TheOptimizationFirm 2001).

Στις εικόνες 5.2, 5.3 που ακολουθούν παριστάνονται το γραφικό περιβάλλον εισόδου δεδομένων του λογισμικού και δημιουργίας του αλγεβρικού μοντέλου, αντίστοιχα.



Εικόνα 5.2 Γραφικό περιβάλλον εισόδου δεδομένων του ALAMO.



Εικόνα 5.3 Γραφικό περιβάλλον δημιουργίας αλγεβρικού μοντέλου.

5.3 Περιγραφή μεθοδολογίας

Σε αυτήν την ενότητα παρουσιάζονται όλα τα βήματα που ακολουθήθηκαν για την επίλυση του προβλήματος της πολυκριτηριακής βελτιστοποίησης.

Βήμα 1: Στο πρώτο βήμα έγινε η δημιουργία του μοντέλου προσομοίωσης στο προγραμματιστικό περιβάλλον του Spyder. Αυτή η διαδικασία περιελάμβανε την καταγραφή των φυσικών εξισώσεων που διέπουν τα υποσυστήματα της διάταξης καθώς και την ανάπτυξη της λειτουργίας του κάθε βρόχου.

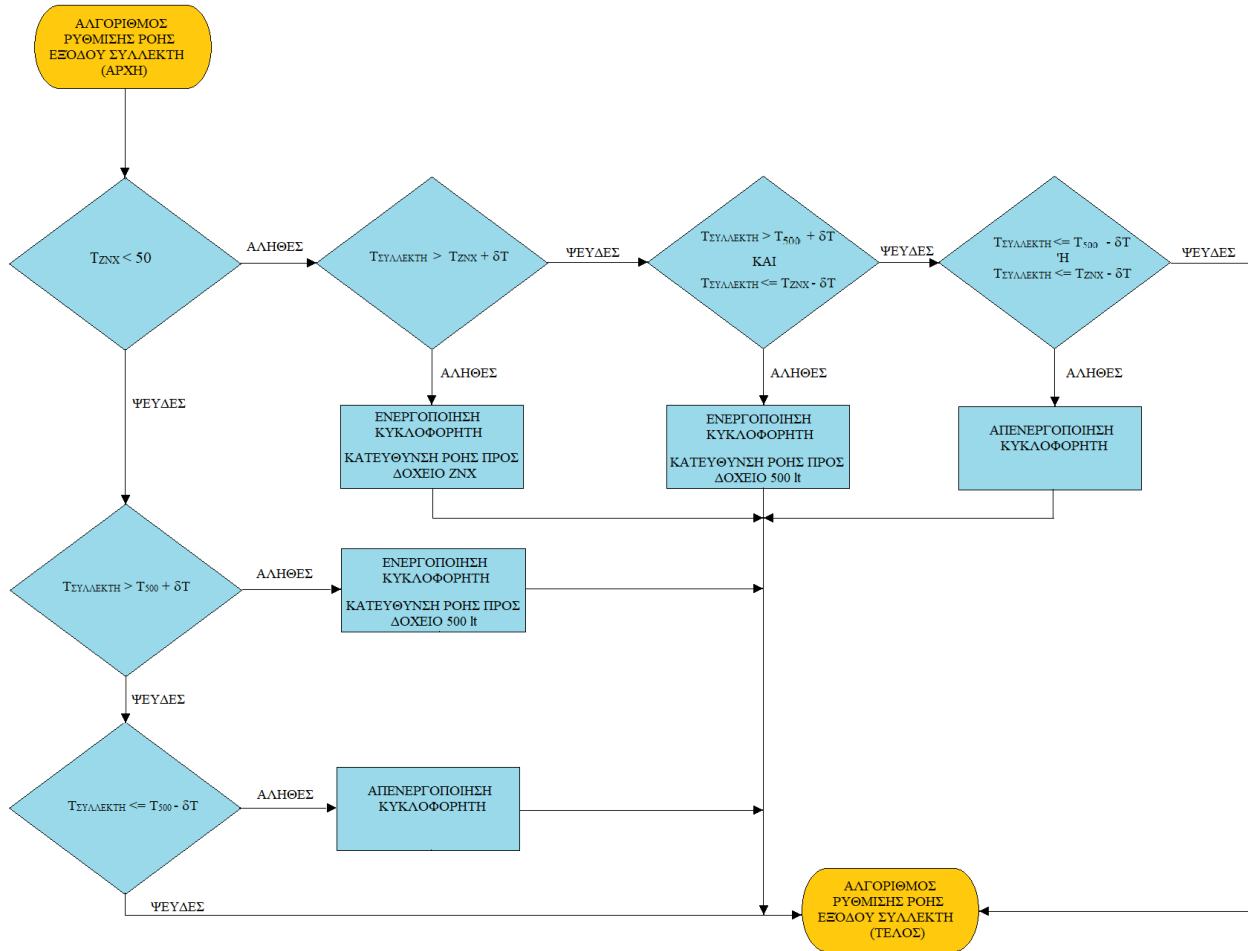
Βήμα 2: Στο δεύτερο βήμα αναπτύχθηκε ο αλγόριθμος ελέγχου του μοντέλου. Ο αλγόριθμος αυτός περιελάμβανε δύο αλγορίθμους, τον αλγόριθμο ρύθμισης της ροής εξόδου του συλλέκτη και τον αλγόριθμο λειτουργίας της αντλίας θερμότητας. Στη συνέχεια παρουσιάζεται ο αλγόριθμος ρύθμισης της ροής εξόδου του συλλέκτη πιο αναλυτικά.

A. Πρόσδοση θερμότητας στις δεξαμενές από τον ηλιακό συλλέκτη.

1. Αν $T_{znx} < 50$ και $T_{co} > T_{znx} + \delta T$ τότε ενεργοποιείται ο κυκλοφορητής του πρώτου βρόχου, ο σωλήνας Προσαγωγής ρυθμίζει την κατεύθυνση της ροής του ρευστού προς την δεξαμενή του ZNX και το νερό ρέει με ρυθμό 0.1 kg/sec
2. Αν $T_{znx} < 50$ και $T_{co} > T_{500} + \delta T$ και $T_{co} \leq T_{znx} - \delta T$ τότε ο σωλήνας T προσαγωγής ρυθμίζει την κατεύθυνση της ροής προς τη δεξαμενή των 500 lt και το νερό ρέει πάλι με ρυθμό 0.1 kg/sec.
3. Αν $T_{znx} < 50$ και $T_{co} \leq T_{500} + \delta T$ και $T_{co} \leq T_{znx} + \delta T$ τότε ο κυκλοφορητής απενεργοποιείται και δεν έχουμε πρόσδοση θερμότητας από τον ηλιακό συλλέκτη.

4. Αν $T_{znx} > 50$ και $T_{co} > T_{500} + \delta T$ τότε ο κυκλοφορητής ενεργοποιείται, ο σωλήνας T προσαγωγής ρυθμίζει την κατεύθυνση του ρευστού προς τη δεξαμενή των 500lt και η παροχή του νερού είναι 0.1 kg/sec.
5. Αν $T_{znx} > 50$ και $T_{co} \leq T_{500} - \delta T$ τότε ο κυκλοφορητής του πρώτου βρόχου απενεργοποιείται διότι δεν μπορεί να προσδώσει θερμότητα ο ηλιακός συλλέκτης.

Στο σχήμα 5.1 που ακολουθεί απεικονίζεται το διάγραμμα ροής του αλγορίθμου ρύθμισης της ροής εξόδου του συλλέκτη.



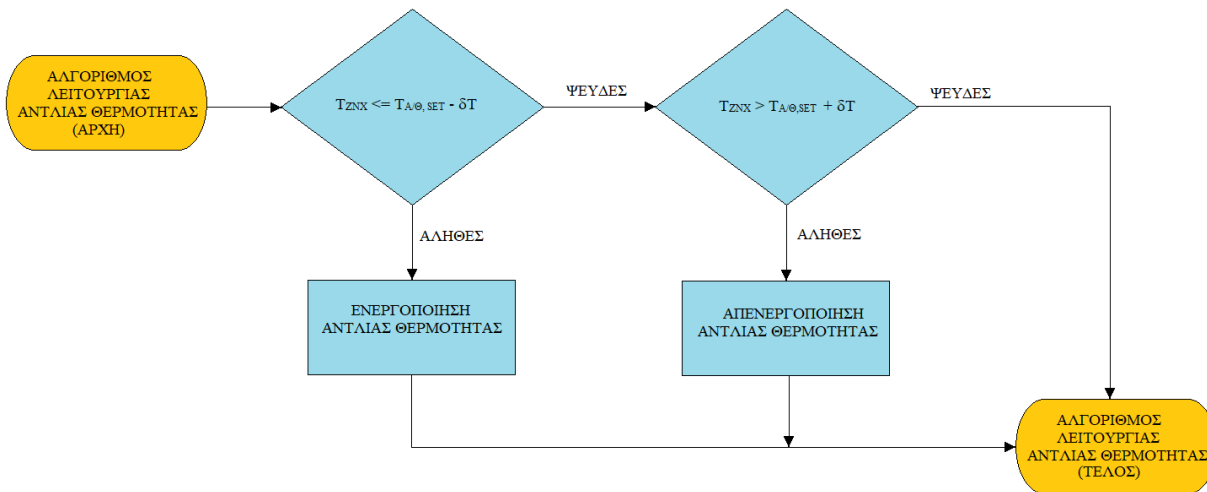
Σχήμα 5.1 Διάγραμμα ροής αλγορίθμου ρύθμισης της ροής εξόδου του συλλέκτη

Στην συνέχεια ακολουθεί η ανάλυση του αλγορίθμου λειτουργίας της αντλίας θερμότητας, καθώς και η σχηματική απεικόνιση αυτού (σχήμα 5.2).

B. Πρόσδοση θερμότητα στις δεξαμενές από την αντλία θερμότητας

1. Αν $T_{znx} \leq T_{HP, set} - \delta T$ τότε ο κυκλοφορητής του δεύτερου βρόχου ενεργοποιείται και το νερό ρέει με ρυθμό \dot{m}_{hp} .
2. Αν $T_{znx} > T_{HP, set} - \delta T$ τότε ο κυκλοφορητής του δεύτερου βρόχου απενεργοποιείται και η αντλία δεν προσδίδει θερμότητα στη δεξαμενή του ZNX.

3. Αν ο κυκλοφορητής του πρώτου βρόχου είναι ενεργοποιημένος και ο σωλήνας Τπροσαγωγής κατευθύνει το νερό προς τη δεξαμενή του ZNX, τότε ο κυκλοφορητής του τρίτου βρόχου απενεργοποιείται και η αντλία δεν προσδίδει θερμότητα στη δεξαμενή του ZNX.



Σχήμα 5.2 Διάγραμμα ροής αλγορίθμου λειτουργίας της αντλίας θερμότητας.

Τα σύμβολα που παρουσιάζονται στους αλγορίθμους ορίζονται ως:

T_{zNX} : θερμοκρασία του νερού στην έξοδο της δεξαμενής του ZNX [°C]

T_{500} : θερμοκρασία του νερού στην έξοδο της δεξαμενής των 500 lt [°C]

Οι άλλες δύο θερμοκρασίες που περιλαμβάνουν οι περιορισμοί υπολογίζονται από τις σχέσεις της ενότητας 4, ενώ ο όρος δT είναι μία διαφορά θερμοκρασίας που εισάγεται στους περιορισμούς για την αποφυγή του συνεχούς ανοίγματος και κλεισίματος των κυκλοφορητών. Η τιμή αυτού του όρου ισούται με 0.5 °C.

Βήμα 3: Στο τρίτο βήμα έγινε η δημιουργία σεναρίων προσομοίωσης. Τα σενάκια αυτά περιελάμβαναν τις τιμές των παραμέτρων βελτιστοποίησης καθώς και τις τιμές των δύο κριτηρίων απόφασης (DHW_{cov} και $Q_{HP,cons}$). Ο αριθμός αυτών των σεναρίων ήταν 400 για κάθε μήνα και έτσι προέκυψαν 4,800 σενάκια για ένα έτος. Τα σύνολα των τιμών των παραμέτρων βελτιστοποίησης αναπαρίστανται στον πίνακα που ακολουθεί.

Πίνακας 5.1 Τιμές παραμέτρων βελτιστοποίησης.

Παράμετροι	Τιμές
H [m]	0.18 - 0.55
A_c [m ²]	2 - 7
\dot{m}_{hp} [kg/sec]	0.1 - 0.5
$T_{HP,set}$ [°C]	40 - 50
Τύπος συλλέκτη	Ενότητα 4.2.3, Πίνακας 4.2

Δύο ακόμη παράμετροι οι οποίες θεωρούνται ως δεδομένα στον υπολογισμό είναι οι θερμοκρασίες που έχει το νερό των δεξαμενών κατά την εκκίνηση της ημέρας. Όπως αναφέρθηκε και

παραπάνω (ενότητα 3.4.3), η βελτιστοποίηση πραγματοποιήθηκε για την διάρκεια ενός έτους και για αυτό το λόγο θα έπρεπε να γίνεται μία ανανέωση της θερμοκρασίας του νερού των δεξαμενών κατά την εκκίνηση της κάθε μέρας. Την πρώτη μέρα κάθε μήνα θεωρήθηκε ότι η θερμοκρασία του εργαζόμενου μέσου μέσα στις δύο δεξαμενές ισούται με την θερμοκρασία νερού δικτύου, όπως αυτή προκύπτει από την Τεχνική Οδηγία 20701-3, του Τεχνικού Επιμελητηρίου Ελλάδας (Τ.Ο.Τ.Ε.Ε. 20701-3 2010). Εφόσον το πρόγραμμα δούλευε με συνεχή ρυθμό, η θερμοκρασία του εργαζόμενου μέσου στις δεξαμενές κατά την εκκίνηση της επόμενης μέρας ισούταν με την θερμοκρασία που είχε το εργαζόμενο μέσο στην τελευταία επανάληψη της προηγούμενης μέρας, δηλαδή στις 00:00. Οι τιμές των θερμοκρασιών αυτών ανανεώνονταν συνεχώς κατά την διάρκεια της κάθε ημέρας, και προέρχονταν από τον υπολογισμό των θερμικών ισχύων που προσέφεραν τα συστήματα (ηλιακός συλλέκτης, αντλία θερμότητας) στις δύο δεξαμενές.

Σε αυτό το σημείο αξίζει να σημειωθεί ότι, μέσω όλης της διεργασίας της βελτιστοποίησης προέκυψαν και δύο ακόμα μεγέθη τα RES_{con} και $Heat_{cov}$ τα οποία εκφράζουν το ποσοστό της συνεισφοράς του ηλιακού συλλέκτη ως προς την κατανάλωση της αντλίας θερμότητας (RES_{con}) και το ποσοστό κάλυψης των φορτίων της δεξαμενής των 500lt από τον ηλιακό συλλέκτη ($Heat_{cov}$). Αυτά τα δύο μεγέθη χρησιμεύουν ως επιπλέον πληροφορία στην συμπεριφορά της διάταξης.

Βήμα 4: Το τέταρτο βήμα περιελάμβανε δύο στάδια:

1. δημιουργία και διατύπωση των μηνιαίων υποκατάστατων μοντέλων προσομοίωσης (Surrogate Simulation Model) των δύο κριτηρίων απόφασης (DHW_{cov} και $Q_{HP,cons}$), μέσω του λογισμικού ALAMO. Αυτά τα υποκατάστατα μοντέλα αφορούν τη συσχέτιση των δύο κριτηρίων απόφασης με τις παραμέτρους βελτιστοποίησης
2. κατάστρωση των μηνιαίων ορίων Pareto (Pareto frontier) (βλέπε ενότητα 3.3). Πιο συγκεκριμένα, αρχικά εφαρμόστηκε το κριτήριο κυριαρχίας (dominance) για όλα τα σενάρια του κάθε μήνα, και στη συνέχεια εντοπίστηκαν τα σημεία που σχηματίζουν το όριο Pareto.

Βήμα 5: Σε αυτό το βήμα έγινε η δημιουργία και η απεικόνιση των μηνιαίων υποκατάστατων μοντέλων του ορίου Pareto (Surrogate Pareto frontier), μέσω του λογισμικού ALAMO. Αυτά τα υποκατάστατα μοντέλα συνδέουν τα δύο κριτήρια απόφασης μεταξύ τους. Πιο ειδικά δίνοντας μία τιμή στο ποσοστό κάλυψης των φορτίων ZNX και επιστρέφει η τιμή της κατανάλωσης ισχύος της αντλίας θερμότητας.

Βήμα 6: Στο έκτο και τελευταίο βήμα περιελάμβανε τέσσερα στάδια:

1. καθορισμός σημείου επιθυμητού ποσοστού κάλυψης των φορτίων ZNX από τον ηλιακό συλλέκτη ($DHW_{cov,set}$) και προσδιορισμός βάσει των υποκατάστατων μοντέλων του ορίου Pareto (Surrogate Pareto frontier) της αντίστοιχης (βέλτιστης) τιμής της κατανάλωσης ισχύος της αντλίας θερμότητας ($Q_{HP,cons,set}$) κάθε μήνα. Η επιθυμητή τιμή του $DHW_{cov,set}$ ήταν 95 %.
2. δημιουργία 30,000 σεναρίων βελτιστοποίησης βάσει των μηνιαίων υποκατάστατων μοντέλων προσομοίωσης (Surrogate Simulation Model). Αυτά τα σενάρια περιείχαν έναν συνδυασμό τιμών από όλες τις παραμέτρους που αναφέρθηκαν στο πίνακα 5.1, τις τιμές των παραμέτρων απόδοσης

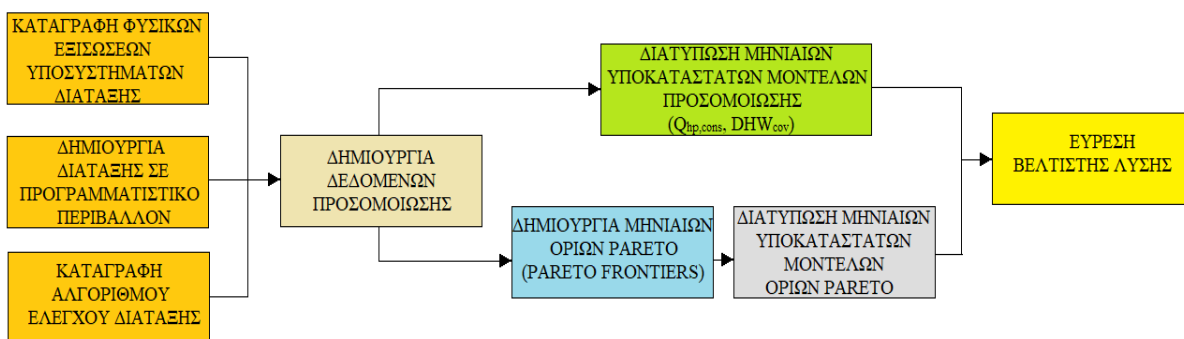
των τεσσάρων ειδών ηλιακών συλλεκτών που αναγράφηκαν στον πίνακα 4.2, καθώς και τις τιμές των κριτηρίων απόφασης ($DHW_{cov,i}$ και $Q_{HP,cons,i}$)

- υπολογισμός του αθροίσματος των τετραγωνικών σχετικών σφαλμάτων (Sum of Square Relative Errors, SSRE) των σεναρίων βελτιστοποίησης. Το SSRE υπολογίζεται ως:

$$SSRE = \sum_{i=1}^{30,000} \left[\left(\frac{Q_{HP,cons,i} - Q_{HP,cons,set,i}}{Q_{HP,cons,set,i}} \right)^2 + \left(\frac{DHW_{cov,i} - DHW_{cov,set}}{DHW_{cov,set}} \right)^2 \right] \quad (5.1)$$

- εύρεση βέλτιστης λύσης βάσει του ελάχιστου αθροίσματος των τετραγωνικών σχετικών σφαλμάτων. Η λύση αυτή περιελάμβανε τις βέλτιστες τιμές των παραμέτρων σχεδιασμού και λειτουργίας.

Το σχήμα 5.3 που ακολουθεί απεικονίζει το διάγραμμα ροής όλου του προβλήματος βελτιστοποίησης.



Σχήμα 5.3 Διάγραμμα ροής προβλήματος βελτιστοποίησης.

6. Αποτελέσματα

6.1 Εξισώσεις υποκατάστατων μοντέλων προσομοίωσης

Σε αυτήν την υπό ενότητα παρουσιάζονται οι εξισώσεις των υποκατάστατων μοντέλων προσομοίωσης (Surrogate Simulation Model), που προέκυψαν από το ALAMO. Η γενική μορφή των εξισώσεων είναι:

$$Y_i = x_{1,i} \cdot H + x_{2,i} \cdot A_c + x_{3,i} \cdot T_{HP,set} + x_{4,i} \cdot \dot{m}_{hp} + x_{5,i} \cdot (\tau \cdot \alpha)_n + x_{6,i} \cdot U_L + x_{7,i} \cdot \ln(H) + x_{8,i} \cdot \ln(A_c) + x_{9,i} \cdot \ln(T_{HP,set}) + x_{10,i} \cdot \ln(\dot{m}_{hp}) + x_{11,i} \cdot \ln((\tau \cdot \alpha)_n) + x_{12,i} \cdot \ln(U_L) + c \quad (6.1)$$

όπου Y: είναι ο συμβολισμός των δύο κριτηρίων $Q_{HP,cons}$ και DHW_{cov}

$x_{j,i}$: συντελεστές των παραμέτρων των εξισώσεων, με $j = 1$ έως 12

i : δείκτης των μηνών, $i = 1$ έως 12, όπου 1 = Ιανουάριος, 2 = Φεβρουάριος κ.ο.κ.

c : σταθερός όρος

Τα κριτήρια αξιολόγησης των μηνιαίων παλινδρομήσεων που διεξήχθησαν στο ALAMO αποτελούν οι δείκτες $R^2_{training}$ (συντελεστής συσχέτισης δεδομένων εκπαίδευσης, εξίσωση 6.2) και $RMSE_{training}$ (τετραγωνική ρίζα του μέσου τετραγωνικού σφάλματος δεδομένων εκπαίδευσης, εξίσωση 6.3).

$$R^2 = 1 - \frac{\sum(y_i - f_i)^2}{\sum(y_i - \bar{y})^2} \quad (6.2)$$

όπου y_i : δεδομένα μετρήσεων

\bar{y} : μέσος όρος των δεδομένων μετρήσεων

f_i : δεδομένα προβλέψεων

$$RMSE = \sqrt{\frac{\sum_{n=1}^N (f_i - y_n)^2}{N}} \quad (6.3)$$

όπου f_i :δεδομένα προβλέψεων

y_i : δεδομένα μετρήσεων

N : πλήθος δεδομένων

Οι τιμές των συντελεστών για το πρώτο κριτήριο απόφασης ($Q_{HP,cons}$), όπως προέκυψαν από το ALAMO, καταγράφονται στον πίνακα 6.1 που ακολουθεί.

Πίνακας 6.1 Τιμές συντελεστών των παραμέτρων βελτιστοποίησης για το πρώτο κριτήριο ($Q_{HP,cons}$).

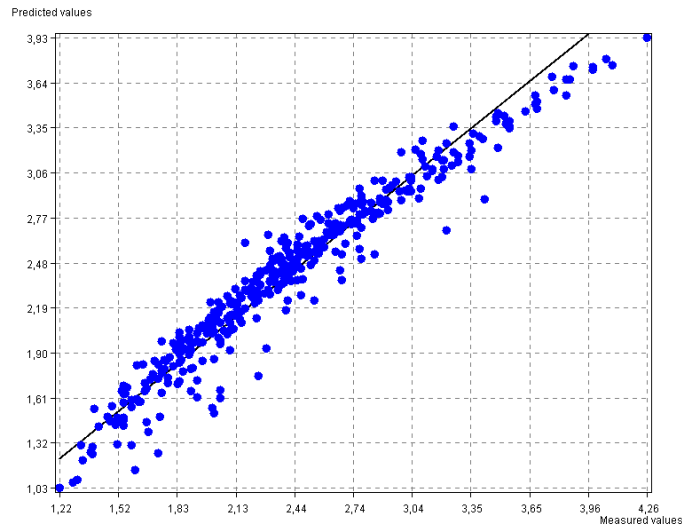
Μήνας	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	c
1	3.09	-0.15	0.12	0.00	-12.56	-0.20	0.00	0.76	0.00	-0.52	0.00	3.31	0.00
2	3.12	0.00	0.12	0.00	-4.76	0.00	0.00	0.20	0.00	-0.52	4.63	0.00	0.00
3	3.46	0.00	0.18	0.00	0.00	0.00	0.00	0.00	-1.92	-0.54	-0.33	0.00	0.00
4	2.41	0.00	0.00	0.00	0.00	0.70	0.00	0.00	6.10	-0.51	29.51	-10.71	0.00
5	2.68	-0.21	0.12	0.00	-10.21	-0.15	0.00	0.82	0.00	-0.40	0.00	2.34	0.00
6	4.13	0.00	0.12	0.00	-13.54	-0.20	-0.75	0.00	0.00	-0.38	0.00	3.19	0.00
7	1.68	0.00	0.18	0.00	0.00	0.00	0.00	0.00	-2.01	-0.33	0.00	0.25	0.00
8	1.85	0.00	0.12	0.00	-4.97	0.00	0.00	0.00	0.00	-0.31	3.51	0.00	0.00
9	3.76	0.00	0.16	0.00	0.00	0.00	-0.59	0.00	-2.06	-0.31	0.30	0.00	0.00
10	2.51	0.00	0.12	0.00	-4.99	0.00	0.00	0.00	0.00	-0.38	3.53	0.00	0.00
11	3.19	0.00	0.34	0.00	0.00	0.00	0.00	0.00	-9.59	-0.45	0.00	0.00	21.68
12	3.46	0.00	0.17	0.00	0.00	0.00	0.00	0.00	-1.90	-0.49	-0.12	0.00	0.00

Στον πίνακα 6.2 που ακολουθεί αναπαριστούνται τα αποτελέσματα παλινδρόμησης, που προέκυψαν από το ALAMO, για τις μηνιαίες καταναλώσεις ισχύος της αντλίας θερμότητας ($Q_{HP,cons,i}$).

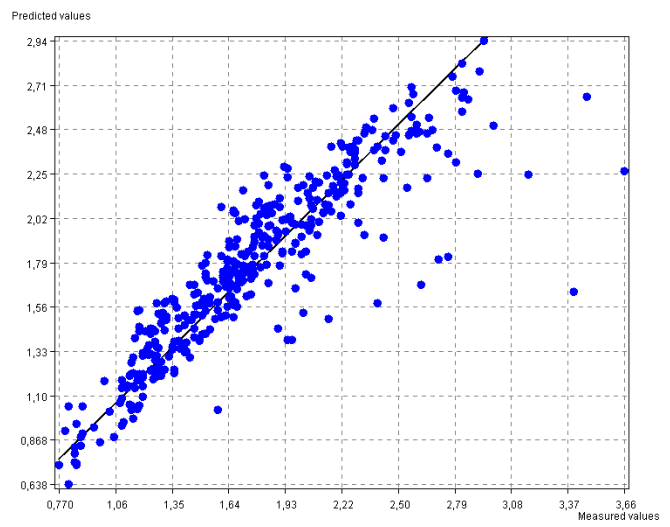
Πίνακας 6.2 Αποτελέσματα παλινδρόμησης για το πρώτο κριτήριο ($Q_{HP,cons,i}$).

Μήνας	R ² (training) [-]	RMSE (training) [kWh]
1	0.941	0.144
2	0.932	0.150
3	0.930	0.152
4	0.804	0.293
5	0.826	0.224
6	0.795	0.221
7	0.793	0.234
8	0.869	0.166
9	0.864	0.166
10	0.909	0.151
11	0.973	0.089
12	0.972	0.096

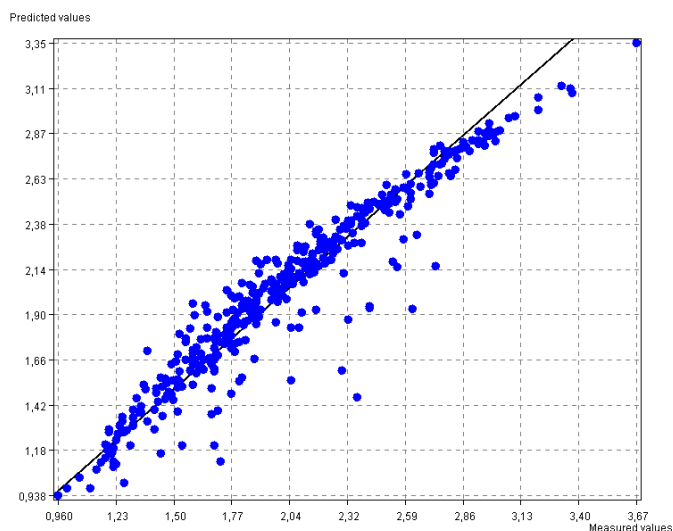
Στη συνέχεια ακολουθούν κάποια διαγράμματα (διάγραμμα 6.1 – 6.3), των μηνιαίων παλινδρομήσεων των τιμών του κριτηρίου απόφασης $Q_{HP,cons}$ που διεξήχθησαν στο λογισμικό ALAMO. Τα μπλε σημεία είναι τα σημεία δεδομένων του κριτηρίου απόφασης και η μαύρη καμπύλη είναι η παλινδρόμηση.



Διάγραμμα 6.1 Παλινδρόμηση κριτηρίου $Q_{HP,cons}$ για τον Ιανουάριο.



Διάγραμμα 6.2 Παλινδρόμηση κριτηρίου $Q_{HP,cons}$ για τον Ιούλιο.



Διάγραμμα 6.3 Παλινδρόμηση κριτηρίου $Q_{HP,cons}$ για τον Οκτώβριο.

Στον πίνακα 6.3 που ακολουθεί καταγράφονται οι τιμές των συντελεστών των παραμέτρων βελτιστοποίησης, που προέκυψαν από το ALAMO, για τις μηνιαίες τιμές του δεύτερου κριτηρίου (DHW_{cov}).

Πίνακας 6.3 Τιμές συντελεστών των παραμέτρων βελτιστοποίησης για το δεύτερο κριτήριο (DHW_{cov}).

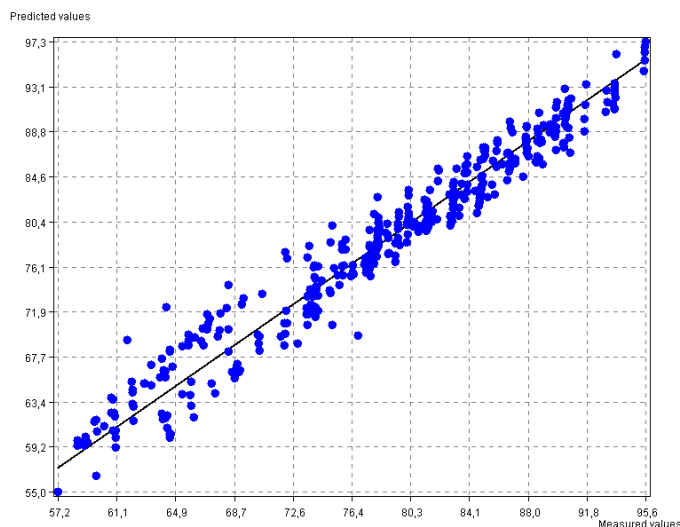
Μήνας	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	c
1	0.00	-2.19	1.73	-2.23	-68.41	-1.76	-3.13	14.69	0.00	0.00	0.00	29.94	0.00
2	-13.00	0.00	1.99	0.00	0.00	0.00	0.00	9.00	0.00	0.00	48.96	0.00	0.00
3	1.61	0.00	2.05	0.00	5.16	0.00	-0.38	-0.10	-2.51	0.00	-4.03	0.36	0.00
4	1.63	0.00	2.00	0.00	0.00	0.00	0.00	-0.10	-0.36	0.00	0.00	0.00	0.00
5	0.00	0.00	2.31	0.00	0.00	0.00	0.74	-0.10	-14.80	0.00	0.00	-0.10	42.50
6	-2.71	0.00	1.94	0.00	0.76	0.00	1.66	0.00	1.41	0.00	0.00	0.00	0.00
7	-3.53	0.00	1.95	0.00	0.00	0.00	1.84	-0.13	1.25	0.00	0.00	0.00	0.00
8	-4.40	0.00	1.55	0.00	0.00	0.00	2.07	0.00	18.88	0.00	0.00	-0.11	-48.46
9	-5.15	0.00	1.61	0.00	0.00	0.00	2.61	0.00	15.42	0.00	0.00	0.00	-37.30
10	0.00	0.00	1.96	0.00	0.00	0.00	0.73	0.00		0.00	0.00	0.00	1.60
11	2.48	0.00	2.02	0.00	0.00	0.00	-0.69	0.00	0.86	0.00	0.00	0.27	0.00
12	0.68	0.00	1.85	0.00	1.11	0.00	-0.32	-0.06	7.24	0.00	0.00	0.00	-22.41

Στον τελευταίο πίνακα 6.4 της συγκεκριμένης υπό ενότητας καταγράφονται τα αποτελέσματα παλινδρόμησης, που προέκυψαν από το ALAMO, για το δεύτερο κριτήριο (DHW_{cov}).

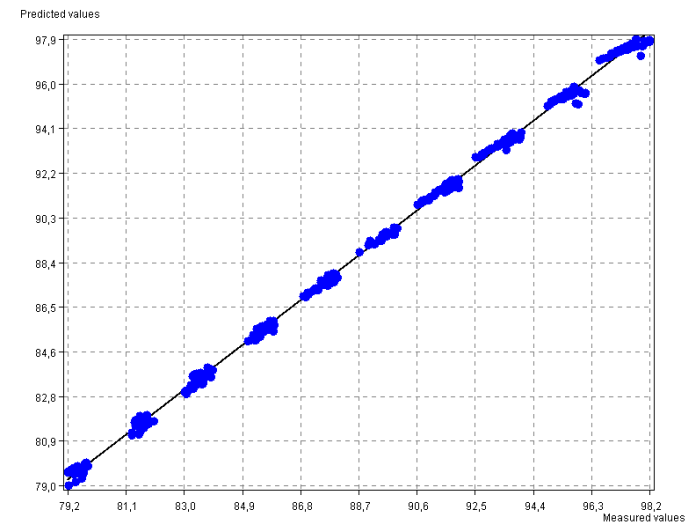
Πίνακας 6.4 Αποτελέσματα παλινδρόμησης για το δεύτερο κριτήριο (DHW_{cov}).

Μήνας	R^2 (training) [-]	RMSE (training) [%]
1	0.952	2.070
2	0.871	3.780
3	1.000	0.097
4	0.999	0.176
5	0.999	0.171
6	0.999	0.192
7	0.999	0.186
8	0.999	0.173
9	0.999	0.174
10	0.999	0.174
11	1.000	0.110
12	1.000	0.076

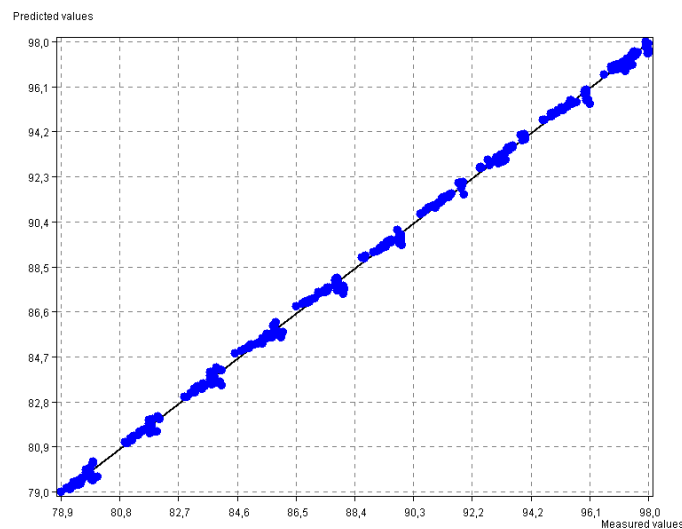
Στη συνέχεια ακολουθούν κάποια διαγράμματα (διάγραμμα 6.4 – 6.6), που απεικονίζουν την παλινδρόμηση των τιμών του κριτηρίου απόφασης DHW_{cov} που διεξήχθησαν στο λογισμικό ALAMO. Τα μπλε σημεία είναι τα σημεία δεδομένων του κριτηρίου απόφασης και η μαύρη καμπύλη είναι η παλινδρόμηση.



Διάγραμμα 6.4 Παλινδρόμηση κριτηρίου DHW_{cov} για τον Ιανουάριο.



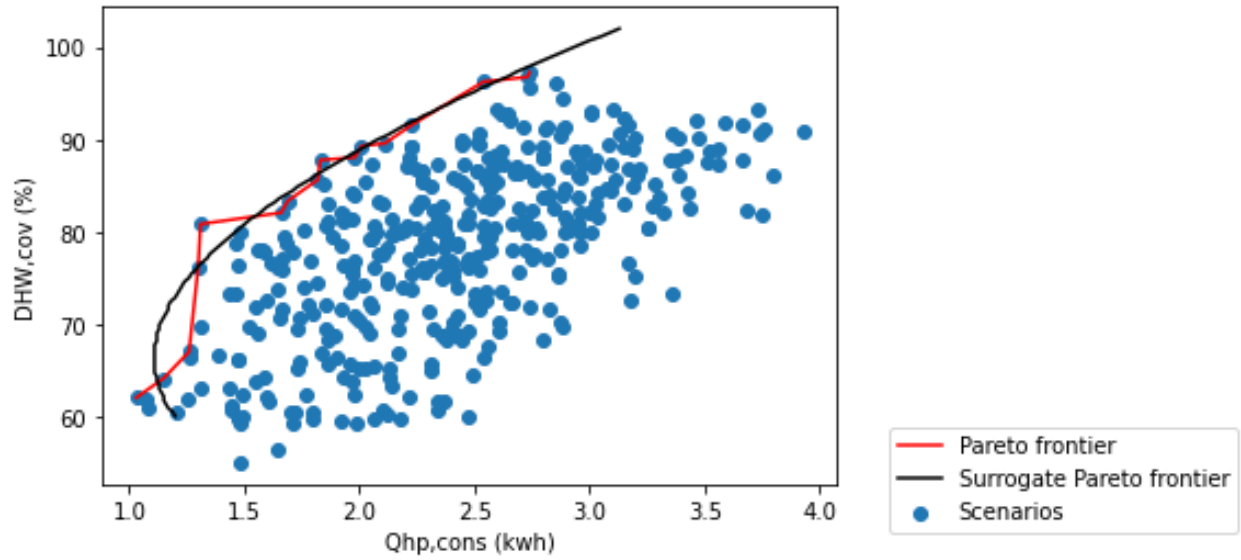
Διάγραμμα 6.5 Παλινδρόμηση κριτηρίου DHW_{cov} για τον Ιούλιο.



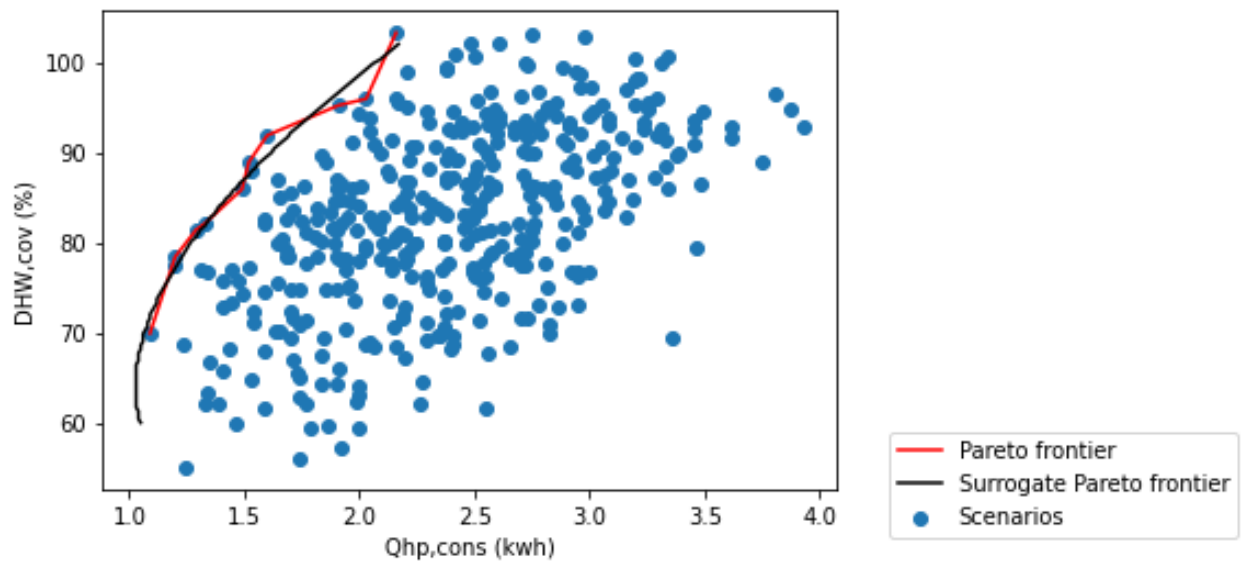
Διάγραμμα 6.6 Παλινδρόμηση κριτηρίου DHW_{cov} για τον Οκτώβριο.

6.2 Καμπύλες Pareto

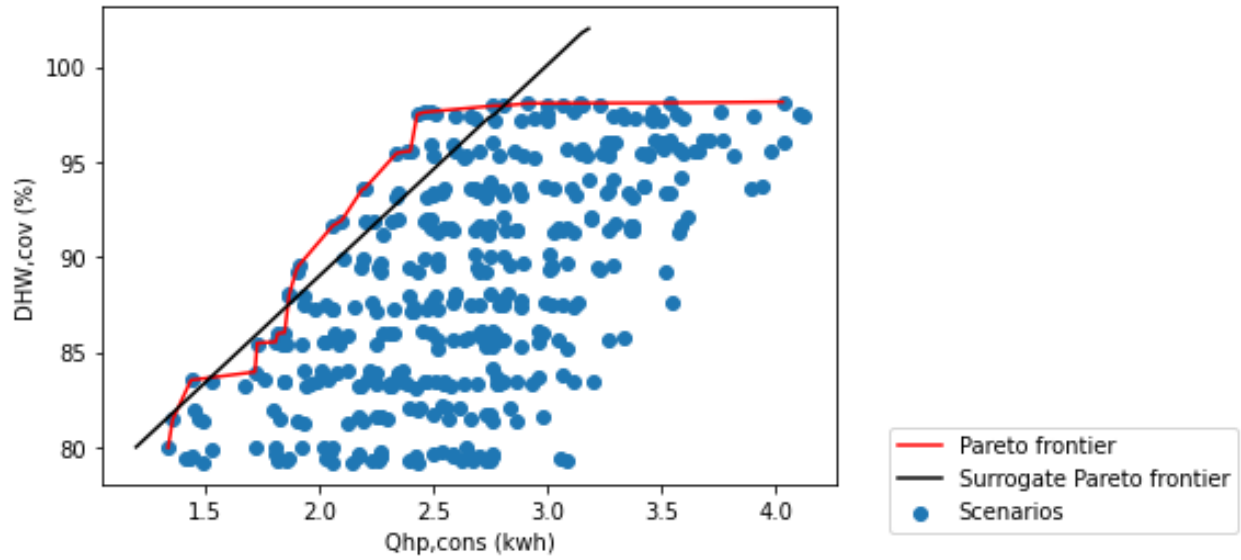
Σε αυτήν την υπό ενότητα παριστάνονται τα διαγράμματα (διάγραμμα 6.7 – 6.18), με τις μηνιαίες καμπύλες του ορίου Pareto (Pareto frontier) και με τις υποκατάστατες καμπύλες του ορίου Pareto (Surrogate Pareto frontier), για τα σενάρια βελτιστοποίησης που προέκυψαν από το βήμα 3 της μεθοδολογίας. Ειδικότερα τα μπλε σημεία αναφέρονται στα αποτελέσματα του μοντέλου προσομοίωσης, η κόκκινη καμπύλη είναι τα βέλτιστα σημεία (όριο Pareto) από τα αποτελέσματα και η μαύρη καμπύλη είναι το υποκατάστατο μοντέλο του ορίου Pareto.



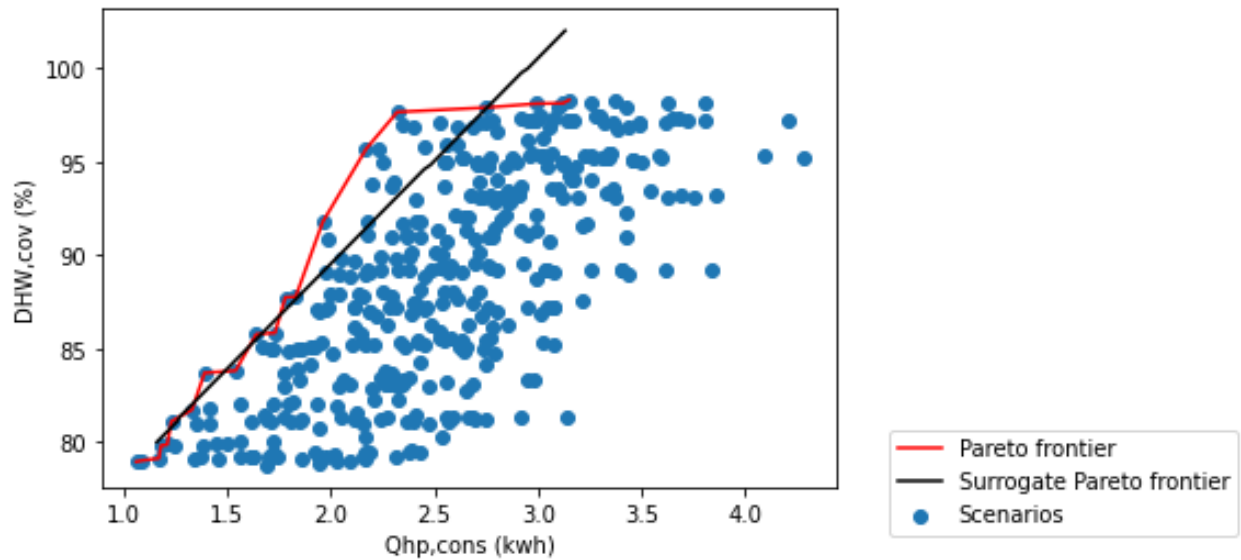
Διάγραμμα 6.7 Καμπύλη Pareto για το μήνα Ιανουάριο.



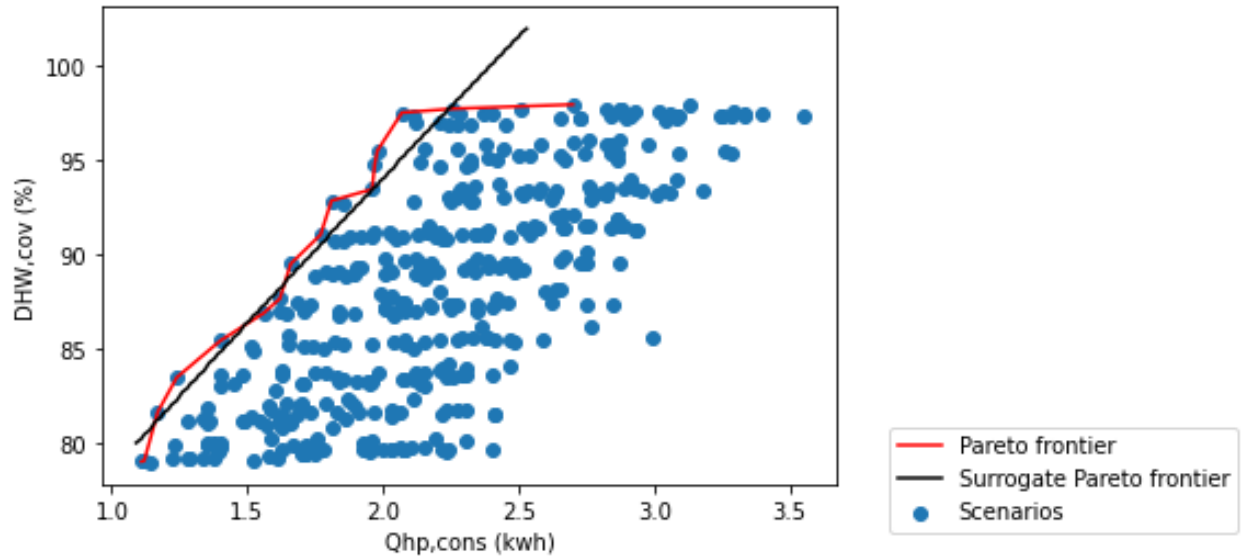
Διάγραμμα 6.8 Καμπύλη Pareto για το μήνα Φεβρουάριο.



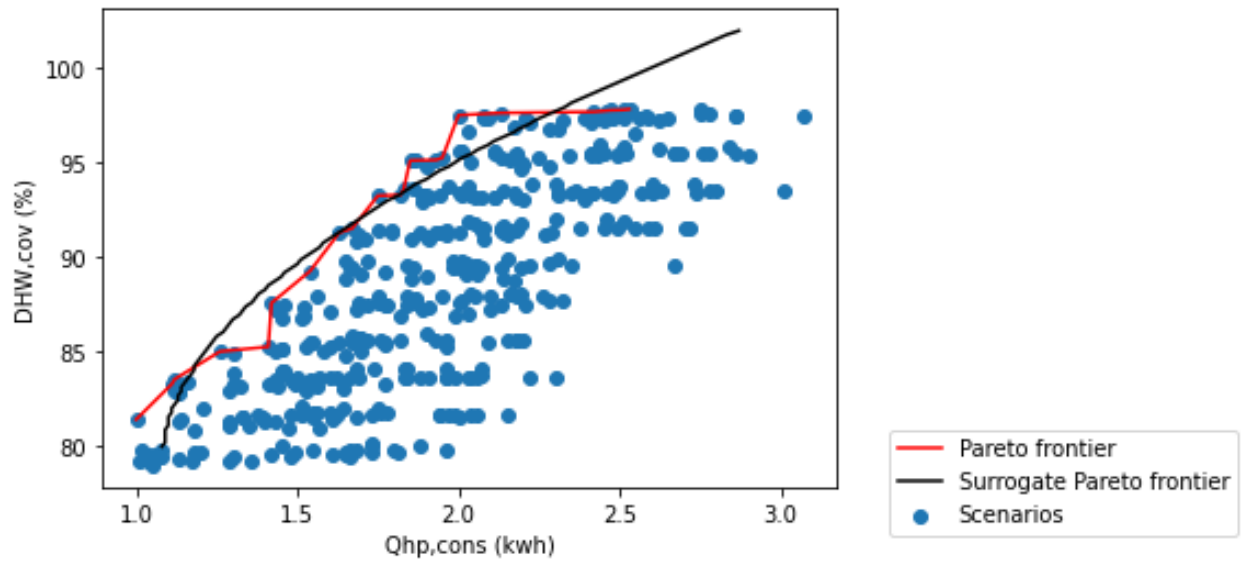
Διάγραμμα 6.9 Καμπύλη Pareto για το μήνα Μάρτιο.



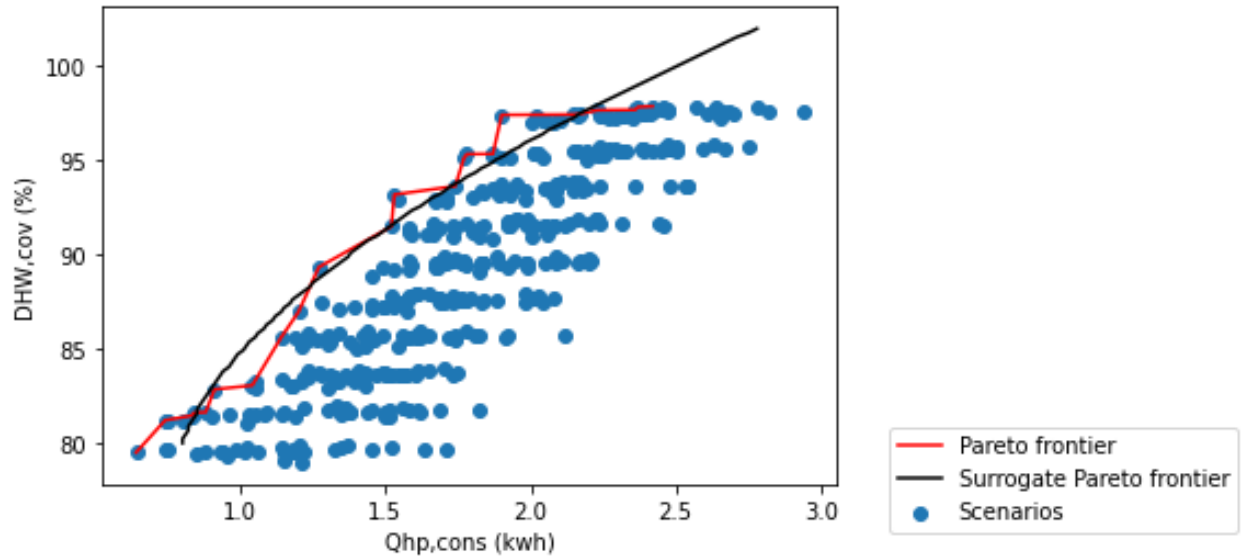
Διάγραμμα 6.10 Καμπύλη Pareto για το μήνα Απρίλιο.



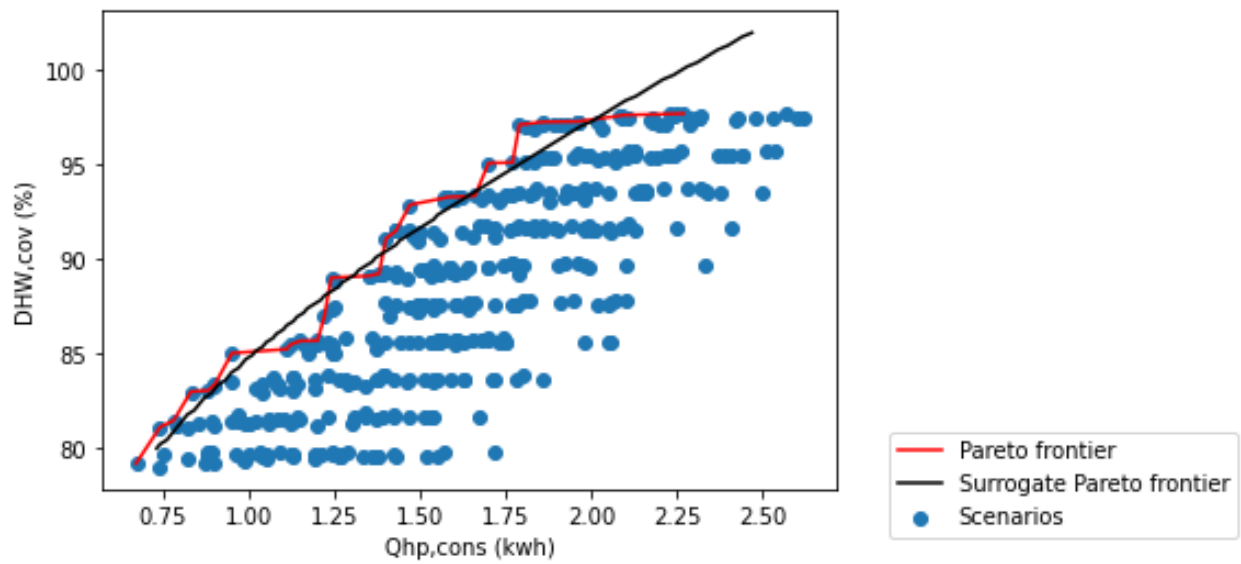
Διάγραμμα 6.11 Καμπύλη Pareto για το μήνα Μάιο.



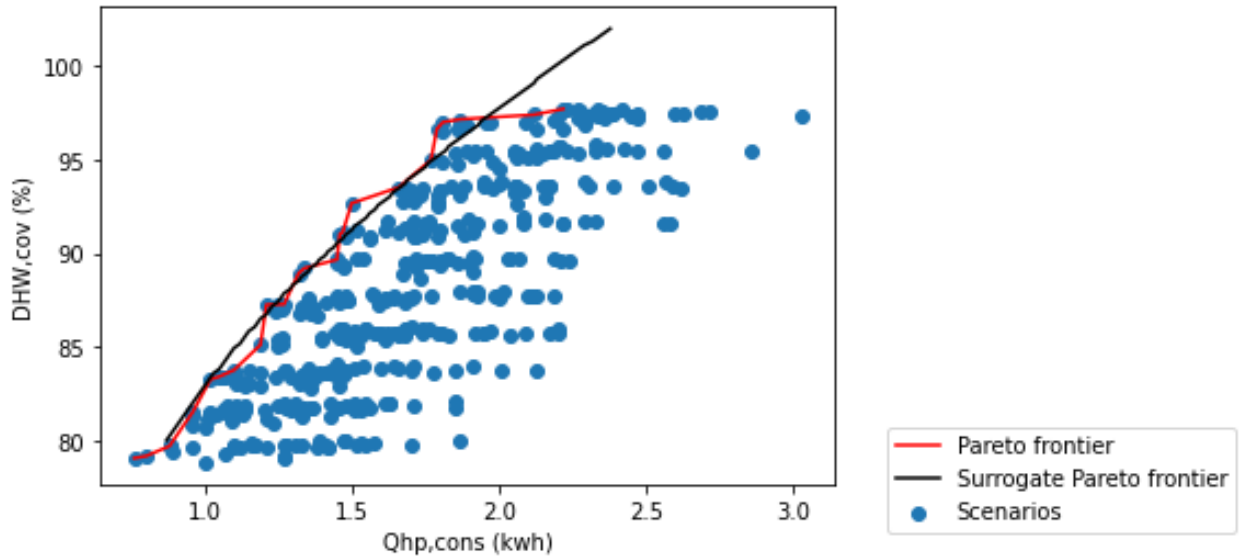
Διάγραμμα 6.12 Καμπύλη Pareto για το μήνα Ιούνιο.



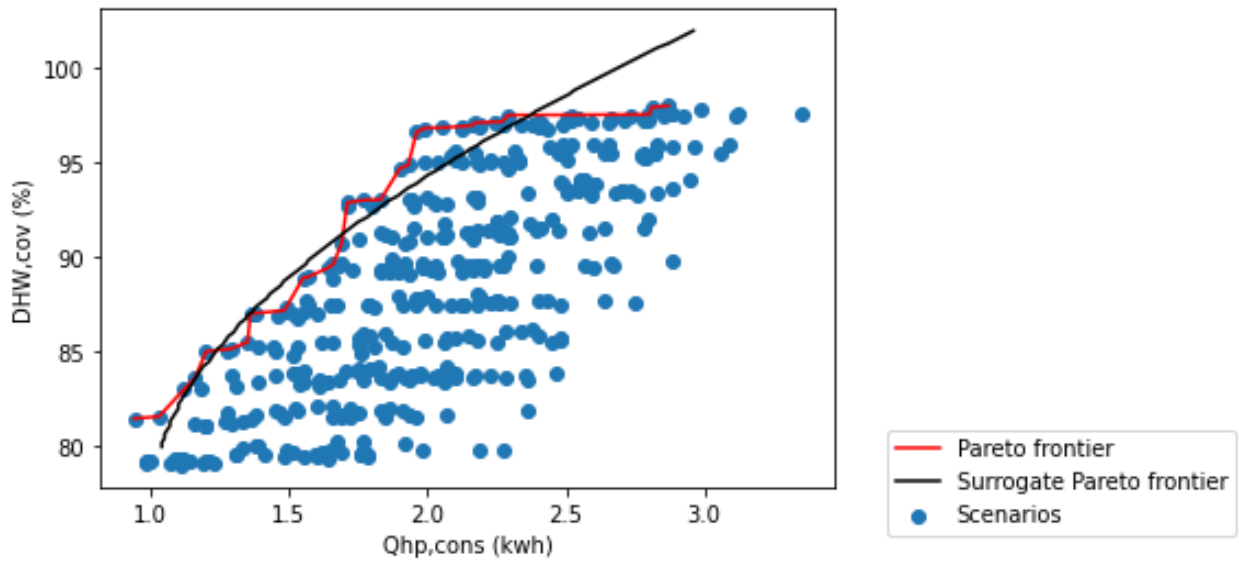
Διάγραμμα 6.13 Καμπύλη Pareto για το μήνα Ιούλιο.



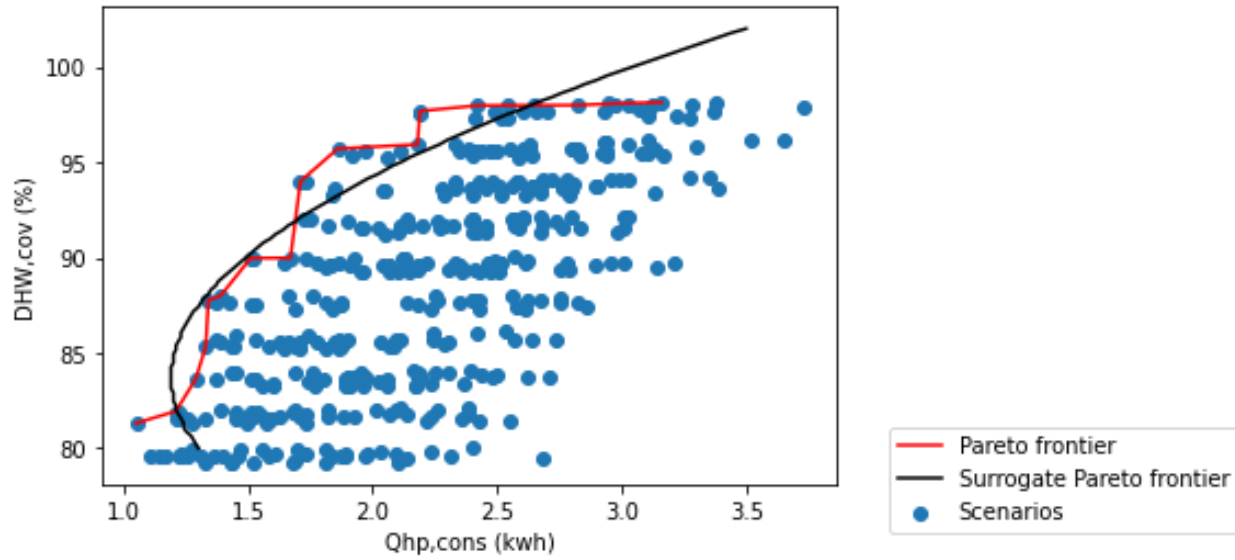
Διάγραμμα 6.14 Καμπύλη Pareto για το μήνα Αύγουστο.



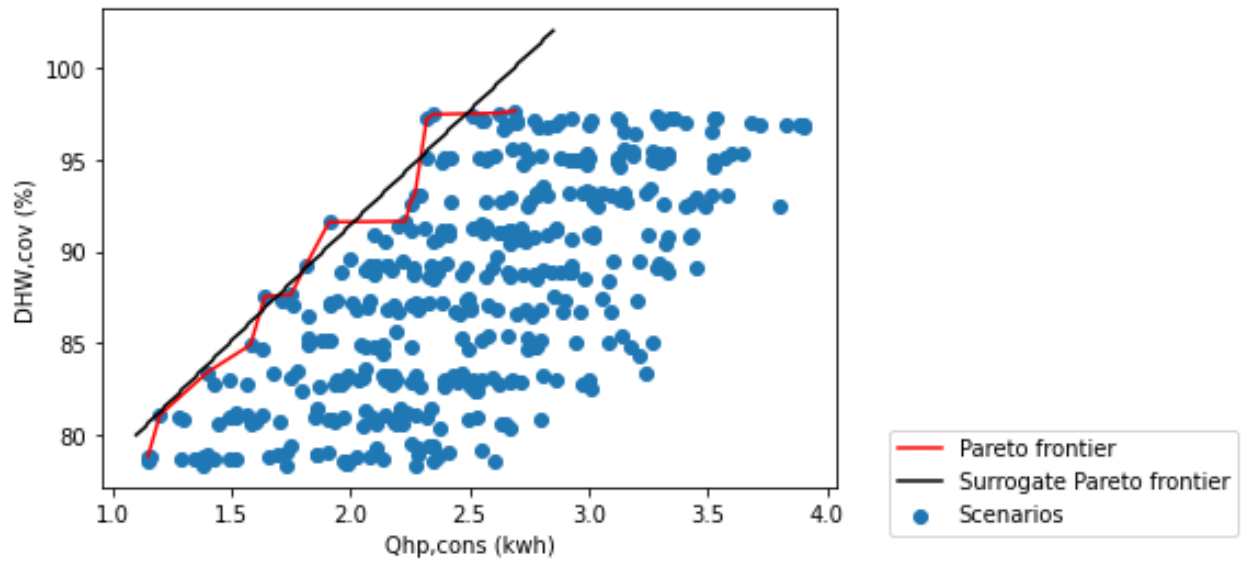
Διάγραμμα 6.15 Καμπύλη Pareto για το μήνα Σεπτέμβριο.



Διάγραμμα 6.16 Καμπύλη Pareto για το μήνα Οκτώβριο.



Διάγραμμα 6.17 Καμπύλη Pareto για το μήνα Νοέμβριο.



Διάγραμμα 6.18 Καμπύλη Pareto για το μήνα Δεκέμβριο.

6.3 Υποκατάστατα μοντέλα καμπυλών Pareto

Οι εξισώσεις που περιγράφουν τα υποκατάστατα μοντέλα των ορίων Pareto (Surrogate Pareto frontier) του κάθε μήνα προέκυψαν από το λογισμικό ALAMO. Η γενική μορφή των εξισώσεων αυτών είναι:

$$Q_{HP,cons,set,i} = x_{1,i} \cdot DHW_{cov,set,i} + x_{2,i} \cdot \ln(DHW_{cov,set,i}) + c \quad (6.2)$$

όπου $Q_{HP,cons,set,i}$: η κατανάλωση ισχύος της αντλίας θερμότητας για κάθε μήνα, με $i = 1$ έως 12 όπου 1 = Ιανουάριος, 2 = Φεβρουάριος κ.ο.κ.

$x_{j,i}$: συντελεστές των παραμέτρων των εξισώσεων, με $j = 1$ έως 2

c : σταθερός όρος

Στον πίνακα 6.5 που ακολουθεί καταγράφονται οι τιμές των συντελεστών όπως αυτές προέκυψαν από το ALAMO

Πίνακας 6.5 Τιμές συντελεστών των εξισώσεων Pareto

Μήνας	x_1	x_2	c
1	0.28	-18.73	60.93
2	0.14	-8.91	29.18
3	0.11	-1.70	0.00
4	0.11	-1.72	0.00
5	0.08	-1.17	0.00
6	0.64	-50.63	171.74
7	0.57	-43.65	146.31
8	0.31	-21.18	68.54
9	0.28	-19.24	62.70
10	0.57	-44.38	149.30
11	1.32	-110.42	379.66
12	0.96	-1.50	0.00

6.4 Παρουσίαση βέλτιστων λύσεων

Με την εφαρμογή του τελευταίου βήματος της μεθοδολογίας που περιγράφηκε παραπάνω (ενότητα 5.3, βήμα 6), προκύπτει η εύρεση βέλτιστων λύσεων τόσο για τις παραμέτρους βελτιστοποίησης όσο και για τα κριτήρια απόφασης. Ειδικότερα, για κάθε μήνα, και για τη δεδομένη τιμή κάλυψης δορτίου $DHW_{cov,set,i}$ (στην περίπτωση εδώ ίση με 95%), από τις εξισώσεις 6.2 προκύπτει η τιμή του $Q_{HP,cons,set,i}$ που παρουσιάζεται στον πίνακα 6.6. Στη συνέχεια για τις τιμές αυτές, βάσει των μηνιαίων υποκατάστατων μοντέλων προσομοίωσης έγινε δημιουργία 30,000 σεναρίων βελτιστοποίησης και επιλογή από αυτά τα σενάρια εκείνων τα οποία αντιστοιχούν στις βέλτιστες τιμές των κριτηρίων απόφασης ($DHW_{cov,i}$ και $Q_{HP,cons,i}$). Για την επιλογή χρησιμοποιείται το κριτήριο του ελάχιστου αθροίσματος τετραγωνικών σχετικών σφαλμάτων (SSRE) κάθε σεναρίου. Στον πίνακα 6.6 καταγράφονται οι μηνιαίες καταναλώσεις που χαρακτηρίζουν τις βέλτιστες λύσεις, βάσει της υποκατάστατης καμπύλης Pareto (Surrogate Pareto frontier).

Πίνακας 6.6 Καταναλώσεις βέλτιστων λύσεων για μέση επιθυμητή κάλυψη φορτίων ZNX 95%

Μήνας	$Q_{HP,cons,set}$ [kWh]
Ιανουάριος	2.48
Φεβρουάριος	1.83
Μάρτιος	2.54
Απρίλιος	2.49
Μάιος	2.06
Ιούνιος	1.99
Ιούλιος	1.88
Αύγουστος	1.78
Σεπτέμβριος	1.78
Οκτώβριος	2.07
Νοέμβριος	2.11
Δεκέμβριος	2.28

Στον πίνακα 6.7 παρουσιάζονται οι βέλτιστες λύσεις που προέκυψαν ύστερα από δέκα εκτελέσεις της μεθοδολογίας εύρεσης βέλτιστου σεναρίου. Αυτές οι λύσεις περιέχουν τις τιμές των βέλτιστων παραμέτρων σχεδιασμού και λειτουργίας καθώς και το ελάχιστο άθροισμα του τετραγωνικού σφάλματος.

Πίνακας 6.7 Βέλτιστες λύσεις για μέση επιθυμητή κάλυψη φορτίων ZNX 95 %

	Ύψος δεξαμενής ZNX [m]	Επιφάνεια ηλιακού συλλέκτη [m ²]	Παροχή νερού αντλίας θερμότητας [kg/sec]	Όριο θερμοκρασίας λειτουργίας της αντλίας θερμότητας [°C]	Τύπος συλλέκτης	SSRE [%]
Λύση 1	0.34	5.8	0.41	42.0	Μαύρο χρώμα με διπλό υαλοπίνακα ή επιλεκτική επιφάνεια με μονό υαλοπίνακα	0.083
Λύση 2	0.42	3.2	0.35	45.0	Σωλήνες κενού-αέρος	0.083
Λύση 3	0.33	2.5	0.48	46.0	Μαύρο χρώμα με μονό υαλοπίνακα	0.066
Λύση 4	0.37	5.6	0.21	49.0	Μαύρο χρώμα με διπλό υαλοπίνακα ή επιλεκτική επιφάνεια με μονό υαλοπίνακα	0.088
Λύση 5	0.37	5.2	0.25	47.0	Μαύρο χρώμα με μονό υαλοπίνακα	0.071
Λύση 6	0.43	4.3	0.31	46.0	Σωλήνες κενού-αέρος	0.066
Λύση 7	0.52	3.1	0.44	43.0	Μαύρο χρώμα με διπλό υαλοπίνακα ή επιλεκτική επιφάνεια με μονό υαλοπίνακα	0.063
Λύση 8	0.48	5.7	0.47	49.0	Μαύρο χρώμα με μονό υαλοπίνακα	0.072
Λύση 9	0.36	5.2	0.23	40.0	Μαύρο χρώμα με διπλό υαλοπίνακα ή επιλεκτική επιφάνεια με μονό υαλοπίνακα	0.069
Λύση 10	0.33	4.6	0.24	40.0	Μαύρο χρώμα με μονό υαλοπίνακα	0.072

7. Συμπεράσματα

Στην παρούσα διπλωματική εργασία μελετήθηκε η πολυκριτηριακή βελτιστοποίηση σχεδιασμού ηλιακού θερμικού συστήματος, υποβοηθούμενου από αντλία θερμότητας, για κάλυψη θερμικών φορτίων. Η υπό μελέτη διάταξη περιελάμβανε τον επίπεδο ηλιακό συλλέκτη, την γεωθερμική αντλία θερμότητας, τους εναλλάκτες θερμότητας, τις δεξαμενές αποθήκευσης και τις καταβόθρες θερμότητας. Για τη δημιουργία δεδομένων χρησιμοποιήθηκε φυσικό μοντέλο δυναμικής προσομοίωσης. Η προτεινόμενη μεθοδολογία επιχείρησε να λάβει υπόψη της την επίδραση των επιλεγόμενων παραμέτρων σχεδιασμού και λειτουργίας στις τιμές των κριτηρίων απόφασης (ποσοστό κάλυψης των φορτίων ZNX για την ικανοποίηση των χρηστών, κατανάλωση ισχύος της αντλίας θερμότητας) για το σύνολο συνθηκών λειτουργίας που θα αντιμετωπίσει το σύστημα σε ετήσια βάση.

Από την μελέτη της πολυκριτηριακής βελτιστοποίησης προέκυψαν οι εξής παρατηρήσεις:

- Η μεθοδολογία εύρεσης βέλτιστου σεναρίου απαιτεί ένα σημαντικό όγκο δεδομένων – μετρήσεων για τη δημιουργία αξιόπιστων υποκατάστατων μοντέλων.
- Τα υποκατάστατα μοντέλα διευκολύνουν την εύρεση βέλτιστων λύσεων, από άποψη ταχύτητας και απαιτούμενης υπολογιστικής ισχύος.

Από την εφαρμογή της μεθοδολογίας για εύρεση βέλτιστων λύσεων συμπεραίνεται ότι:

- Ο απλός συλλέκτης με πλαστικούς σωλήνες χωρίς κάλυμμα και μόνωση δεν χρησιμοποιείται σε κάποια λύση. Μία αναλυτικότερη προσομοίωση της εξεταζόμενης διάταξης με προσθήκη νέων παραμέτρων/δεδομένων μπορεί να προσφέρει επιπλέον πληροφορία σχετικά με αυτό το φαινόμενο.
- Οι υπόλοιπες παράμετροι βελτιστοποίησης εμφανίζουν μία διακύμανση στις τιμές τους, πράγμα το οποίο επιβεβαιώνει την μεγάλη ευελιξία που διαθέτει η συγκεκριμένη μεθοδολογία.
- Παρά το γεγονός ότι η προτεινόμενη μεθοδολογία εξάγει διαφορετικές μεταξύ τους καμπύλες Paretoσε μηνιαίο επίπεδο, οι εξαγόμενες λύσεις είναι αντιπροσωπευτικές καθ' όλη τη διάρκεια ενός έτους λειτουργίας. Πιο συγκεκριμένα, η απόκλιση των κριτηρίων των προτεινόμενων λύσεων ($Q_{HP,cons}$, DHW_{con}) από τα επιθυμητά βέλτιστα κριτήρια δεν υπερβαίνει το 1%.
- Από τα παραπάνω, επιβεβαιώνεται η ικανότητα της μεθοδολογίας να βρίσκει μία γενική αλλά ταυτόχρονα και αξιόπιστη λύση.

Σύμφωνα με την εμπειρία που αποκτήθηκε με την παρούσα διερεύνηση αλλά και πιθανούς περιορισμούς που συζητήθηκαν παραπάνω παρουσιάζονται οι ακόλουθες προτάσεις για μελλοντική διερεύνηση που προέκυψαν από την εκπόνηση της παρούσας εργασίας, οι οποίες αναφέρονται παρακάτω:

1. Χρήση δεδομένων από αντίστοιχες πραγματικές διατάξεις
2. Επέκταση και εφαρμογή μεθοδολογίας και σε διατάξεις HVAC διαφορετικής διαμόρφωσης
3. Διερεύνηση περισσότερων παραμέτρων για τη βελτίωση των υποκατάστατων μοντέλων
4. Μελέτη και εφαρμογή διαφορετικών μεθόδων βελτιστοποίησης

5. Παράλληλη βελτιστοποίηση σχεδιασμού και λειτουργίας συστήματος.

Βιβλιογραφία

Argiriou, A., et al. "Comparison of methodologies for TMY generation using 20 years data for Athens, Greece." Vol. 66. Μάιος 1999. 33-45.

Arnaoutakis, Nektarios, Maria Milousi, Spiros Papaefthimiou, Paris A. Fokaides, Yannis G. Caouris, and Manolis Souliotis. "Life cycle assessment as a methodological tool for the optimum design of integrated collector storage solar water heaters." *Energy*, Σεπτέμβριος 1, 2019: 1084-1099.

Behidj N., Bernier J., Blais S., Demers D., Genest S., Ramsum C., Sassi K. *Energy use data handbook : 1990 and 1998 to 2004*. Canada, 2006.

Bolaji, B.O., M.A. Akintunde, and T.O. Falade. *Comparative Analysis of Performance of Three Ozone-Friends HFC Refrigerants in a Vapour Compression Refrigerator*. Journal of Sustainable Energy and Environment , 2011.

Chunga, Mo, και Hwa-Choon Park. «Comparison of building energy demand for hotels, hospitals, and offices in Korea.» Τόμ. 92. 1 Δεκέμβριος 2015. 383-393.

Ciucci, Matteo. «Ανανεώσιμες Πηγές Ενέργειας.» Ευρωπαϊκό Κοινοβούλιο, 11 2020.

Cozada, Alison, Nikolaos V. Sahinidisab, and David C. Miller. "A combined first-principles and data-driven approach to model building." Vol. 73. Φεβρουάριος 2, 2015. 116-127.

Crall, Christopher P. «Understanding Thermal Systems: Industrial Refrigeration Systems.» *Insulation Outlook*, 1 Ιούλιος 2015.

Deb, Kalyan. «Multi-Objective Optimization Using Evolutionary Algorithms:An Introduction.» Φεβρουάριος 2011.

Diana Solar. 2020. <https://www.diana-solar.gr/>.

Duffie, John A., and William A. Beckman. *Solar Engineering of Thermal Processes*. 4. John Wiley & Sons, 2013.

Evatt, David. «THE POWER OF CHOICE.» Elephant Ears Service Company, 22 Απρίλιος 2015.

Francesca Martorana, Marina Bonomolo, Giuliana Leone, Francesco Monteleone, Gaetano Zizzo, Marco Beccali. «Solar-assisted heat pumps systems for domestic hot water production in small energy communities.» *Solar Energy*, 2021: 113-133.

G. Panaras, E. Mathioulakis, V. Belessiotis. «Investigation of the performance of a combined solar thermal heat pump hot water system.» *Solar Energy*, 2013: 169-182.

Green Solar Vacuum. 2020. https://www.green-solar.net/uploads/6/1/1/7/6117098/solines_kenou_green_solar_vacuum.pdf.

Hanania, Jordan, Kailyn Stenhouse, Brodie Yyelland, and Jason Donev. "Solar collector." *ENERGY EDUCATION*, Μάιος 11, 2018.

Kalogirou, Soteris. «Solar thermal collectors and applications.» *Progress in Energy an Combustion Science*, 2004, 3 εκδ.: 231-295.

Klein, S.A., and W.A. Beckman. "TRNSYS 16: A transient system simulation program." *Mathematical Reference*. Ιανουάριος 2007.

Kusiak, Andrew, Fan Tang, και Guanglin Xu. «Multi-objective optimization of HVAC system with an evolutionary computation algorithm.» *Energy*, Μάιος 2011, 5 εκδ.: 2440-2449.

McQuiston, Faye C., and Jerald D. Parker. *ΘΕΡΜΑΝΣΗ ΑΕΡΙΣΜΟΣ ΚΑΙ ΚΑΙΜΑΤΙΣΜΟΣ*. Ίων, 2003.

Mossolly, M., K. Ghali, and N. Ghaddar. "Optimal control strategy for a multi-zone air conditioning system using a genetic algorithm." *Energy*, Ιανουάριος 2009, 1 ed.: 58-66.

Mouzeviris, G., και K. Papakostas. «Comparative Analysis of Air-to-Water and Ground Source Heat Pumps Performances.» 1 Ιούνιος 2020.

Nassif, N., S. Kajl, and R. Sabourin. "Evolutionary algorithms for multi-objective optimization in HVAC system control strategy." Institute of Electrical and Electronics Engineers, IEEE, Ιούνιος 2004.

Pezzutto S., Toleikyte A., De Felice M., «Assessment of the Space Heating and Cooling Market in the EU28: A Comparison between EU15 and EU13 Member States.» *International Journal of Contemporary Energy*, 2015.

Puneet Saini, Bonato Paolo, Frank Fiedler, Joakim Widén, Xingxing Zhang. «Techno-economic analysis of an exhaust air heat pump system assisted by unglazed transpired solar collectors in a Swedish residential cluster.» *Solar Energy*, 2021: 966-983.

Rossano Scoccia, Tommaso Toppi, Marcello Aprile, Mario Motta. «Absorption and compression heat pump systems for space heating and DHW in European buildings: Energy, environmental and economic analysis.» *Journal of Building Engineering*, 2017: 94-105.

Rossato . 2020. <https://el.rossatogroup.com>.

Savic, Dragan. «Single-objective vs. Multiobjective Optimisation for Integrated Decision Support.» Ιούνιος 2002.

Seyed Housman Razavi, Rouhollah Ahmadi, Alireza Zahedi. «Modeling, simulation and dynamic control of solar assisted ground source heat pump to provide heating load and DHW.» *Applied Thermal Engineering*, 2018: 127-144.

Solar Heat Europe. «Solar Heat.» 2017.

Solar Sence. 2013. <http://www.solarsense.co.za/solar-water-heating-explained.php> (πρόσβαση Ιούνιος 2020).

Sonnenenergie, Deutsche Gesellschaft für. *Planning and Installing Solar Thermal Systems: A Guide for Installers, Architects, and Engineers*. Earthscan, 2005.

Spyder. 2020. <https://www.spyder-ide.org/>.

Starke, Allan R., José M. Cardemil, and Sergio Collea. "Multi-objective optimization of a solar-assisted heat pump for swimming pool heating using genetic algorithm." *Applied Thermal Engineering*, Σεπτέμβριος 2018: 118-126.

The Optimization Firm. 2001. <https://minlp.com/alamo-modeling-tool> (πρόσβαση 2020).

The Renewable Energy Hub. 2020. <https://www.renewableenergyhub.co.uk/main/heat-pumps-information/> (accessed UK).

wikimedia commons. 2009.
https://commons.wikimedia.org/wiki/File:Active_open_loop_solar_HW_system.png (πρόσβαση Ιούλιος 2020).

wikimedia commons. 2009.
https://commons.wikimedia.org/wiki/File:Active_closed_loop_solar_HW_system.png (πρόσβαση Ιούνιος 2020).

Wikipedia. 2020. <https://el.wikipedia.org/wiki/>.

Wikipedia. 2020. <https://el.wikipedia.org/wiki/>.

William M. Duarte, Tiago F. Paulino, Sinthya G. Tavares, Antônio A.T. Maia, Luiz Machado. «Feasibility of solar-geothermal hybrid source heat pump for producing domestic hot water in hot climates.» *International Journal of Refrigeration*, 2020, 124 εκδ.: 184-196.

Zitzler, Eckart. «Evolutionary Algorithms for Multiobjective Optimization:..» 11 Νοέμβριος 1999.

Βαρβούνης, Πέτρος. *VK Μηχανολόγοι Μηχανικοί*. Μάιος 2013. <https://vkme.gr> (πρόσβαση Ιούνιος 2020).

Βραχόπουλος, Μ., Α. Γαγλία, Κ. Καββαδίας, Ι. Καλδέλης, and Κ. Μπαλαράς. "ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΝΕΩΣΙΜΩΝ ΠΗΓΩΝ ΕΝΕΡΓΕΙΑΣ." Αθήνα: Τεχνικό Επιμελητήριο Ελλάδας, ΤΕΕ, Ιούνιος 2011.

ΕΝ.Ε.ΕΠΙ.Θ.Ε. «Αντλίες Θερμότητας: Μέγιστη Εξοικονόμηση Ενέργειας και Ιδανικό Περιβάλλον Χειμώνα-Καλοκαίρι.» Ένωση Ελληνικών Επιχειρήσεων Θέρμανσης και Ενέργειας, 2020.

ΕΟΠ, Ευρωπαϊκός Οργανισμός Περιβάλλοντος. «Ενέργεια.» 2021.

Κ.Α.Π.Ε., Κέντρο Ανανεώσιμων Πηγών Ενέργειας. «Θερμικά Ηλιακά Συστήματα.» 2000.

Κατσίγιαννης, Γ. *Ηλιακοί Συλλέκτες*. 2015.

Κουρουπέτογλου, Α. «Σωστός υπολογισμός θερμικών και ψυκτικών φορτίων.» ΠΗΓΗ Υδραυλικών Α.Ε., 29 Μάιος 2019.

- Λάζαρη, Ε. «Βιοκλιματικός Σχεδιασμός στην Ελλάδα.» Αθήνα: Κέντρο Ανανεώσιμων Πηγών Ενέργειας, ΚΑΠΕ, 2002.
- Λάζαρη, Ευγενία. «Βιοκλιματικός Σχεδιασμός στην Ελλάδα.» Αθήνα: Κέντρο Ανανεώσιμων Πηγών Ενέργειας, Κ.Α.Π.Ε., 2002.
- Μανωλάκος, Δ. «Σημειώσεις μαθήματος 'Εφαρμοσμένη Θερμοδυναμική'.» Αθήνα: Γεωπονικό Πανεπιστήμιο Αθηνών, Τμήμα ΑΦΠ&ΓΜ, 2020.
- Μπέλλος, Ε. Α. «Αξιοποίηση και βελτιστοποίηση της χρήσης ηλιακών συστημάτων στα κτίρια.» Αθήνα, Οκτώβριος 2016.
- Πανάρας, Γ. "Σημειώσεις μαθήματος "Ενεργειακός Σχεδιασμός Κτιρίων Ι"." Κοζάνη: Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανολόγων Μηχανικών, 2020.
- Παντελίδης, Γ. Ν. *Οδηγός Ενεργειακής Επιθεώρησης Κτηρίων*. ΔΕΔΕΜΑΔΗ, 2015.
- Παπακώστας, Κ.Τ. «Εξοικονόμηση Ενέργειας σε Συστήματα Θέρμανσης Και Κλιματισμού.» 2015.
- Πάπιστα, Ε. «Σημειώσεις μαθήματος 'Συσκευές Θερμικών Διεργασιών'.» Κοζάνη: Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανολόγων Μηχανικών, 2020.
- «Τ.Ο.Τ.Ε.Ε. 20701-1.» *Αναλυτικές Εθνικές Προδιαγραφές παραμέτρων για τον Υπολογισμό της Ενεργειακής Απόδοσης Κτιρίων και την έκδοση ΠΕΑ*. Τεχνικό Επιμελητήριο Ελλάδας, ΤΕΕ, 2017α, 2017.
- "Τ.Ο.Τ.Ε.Ε. 20701-3." *"Κλιματικά δεδομένα Ελληνικών περιοχών"*. Τεχνικό Επιμελητήριο Ελλάδας, ΤΕΕ, 2010γ, 2010.
- Τζανακάκη, Ε. «Ενεργειακή Αποδοτικότητα στο Δομημένο Περιβάλλον.» Κέντρο Ανανεώσιμων Πηγών Ενέργειας, ΚΑΠΕ, 2008.

Παράρτημα Προγράμματα που χρησιμοποιήθηκαν στο λογισμικό Spyder

Π.1 Βιβλιοθήκη LoopTools

```
"""  
Created on Thu Dec 10 14:29:35 2020  
@author: Leonidas Zouloumis  
Functions create the tools of the simulation model's classes  
"""  
  
class Loop():  
  
    def __init__(self, msup, mtemp):  
self.Pump = Pump(msup, mtemp)  
self.ComponentsList = [self.Pump]  
self.event = None  
  
    #end def  
  
#end class  
  
  
class LoopManager():  
  
    def __init__(self):  
self.Loops = []  
self.PID_signs = list()  
  
    def AddLoop(self, msup, mtemp):  
self.Loops.append(Loop(msup, mtemp))
```

```

def AddElementToLoop(self, element, loop_indx):
self.Loops[loop_indx].ComponentsList.append(element)

def AddExchangerToTank(self, BfLoop, BfTankPlace, ExLoop, ExPlace, NodeStart, NodeEnd):

self.Loops[ExLoop].ComponentsList[ExPlace].SetInteractor(self.Loops[BfLoop].ComponentsList[BfTan
kPlace], NodeStart, NodeEnd)

def LinkExchangerSides(self, SourceLoop, SourcePlace, LoadLoop, LoadPlace):

self.Loops[SourceLoop].ComponentsList[SourcePlace].SetInteractor(self.Loops[LoadLoop].Components
List[LoadPlace])

def LinkElements(self, LoopIndx, ObjIndexSender, ObjIndexReceiver):
#----
#Set two variables representing the elements to be linked,
#for simplicity
#----
Send = self.Loops[LoopIndx].ComponentsList[ObjIndexSender]
Rec = self.Loops[LoopIndx].ComponentsList[ObjIndexReceiver]
#----
#Link the two elements. The outlet pipe of the first links
#to the inlet pipe of the second
#----
Rec.Inlet = Send.Outlet
if not isinstance(Rec, Pump):
Rec.Outlet.MassSupply = Rec.Inlet.MassSupply
#---- Specific rules
if isinstance(Rec, SourceHeatExchanger):
for i in range(len(Rec.TemperatureAreaOut)):

```

```
Rec.TemperatureAreaOut[i][0] = Rec.Inlet.Temperature[0]
```

```
elifisinstance(Rec, Tpipe2out):
```

```
Rec.Outlet.MassSupply = [0]
```

```
Rec.Outlet.Temperature = Rec.Inlet.Temperature
```

```
    Rec.Outlet2.Temperature = Rec.Inlet.Temperature
```

```
Rec.Outlet.MassSupply[0] = Rec.Inlet.MassSupply[0] * Rec.MassRatio
```

```
    Rec.Outlet2.MassSupply[0] = Rec.Inlet.MassSupply[0] - Rec.Outlet.MassSupply[0]
```

```
def DoubleLinkIn(self, LoopIndx, ObjIndexSender1, ObjIndexSender2, ObjIndexReceiver):
```

```
    #---
```

```
    #Set three variables representing the elements to be linked,
```

```
    #for simplicity
```

```
    #---
```

```
    Send = self.Loops[LoopIndx].ComponentsList[ObjIndexSender1]
```

```
    Send2 = self.Loops[LoopIndx].ComponentsList[ObjIndexSender2]
```

```
    Rec = self.Loops[LoopIndx].ComponentsList[ObjIndexReceiver]
```

```
    #---
```

```
    #Link the three elements. The outlet pipes of the first and the second link
```

```
    #to the inlet pipes of the third
```

```
    #---
```

```
Rec.Inlet = Send.Outlet
```

```
    Rec.Inlet2 = Send2.Outlet
```

```
Rec.Outlet.MassSupply[0] = Rec.Inlet.MassSupply[0] + Rec.Inlet2.MassSupply[0]
```

```
Rec.Outlet.Temperature[0] = (Rec.Inlet.MassSupply[0] * Rec.Inlet.Temperature[0] +  
Rec.Inlet2.MassSupply[0] * Rec.Inlet2.Temperature[0]) / Rec.Outlet.MassSupply[0]
```

```
def DoubleLinkOut(self, LoopIndx, ObjIndexSender, ObjIndexReceiver1, ObjIndexReceiver2):
```

```
    #Set three variables representing the elements to be linked,
```

```
    #for simplicity
```

```

#---
Send = self.Loops[LoopIndx].ComponentsList[ObjIndexSender]
Rec1 = self.Loops[LoopIndx].ComponentsList[ObjIndexReceiver1]
Rec2 = self.Loops[LoopIndx].ComponentsList[ObjIndexReceiver2]
#--- In order to achieve a fully functional PID, using any component and bypass combination
if isinstance(Rec1, DummyPipe):
    p = 1 * (not isinstance(Rec2, LoadHeatExchanger)) + (-1) * (isinstance(Rec2,
LoadHeatExchanger))
self.PID_signs.append(-p)
elif isinstance(Rec2, DummyPipe):
    p = 1 * (not isinstance(Rec1, LoadHeatExchanger)) + (-1) * (isinstance(Rec1,
LoadHeatExchanger))
self.PID_signs.append(p)
#---
#---
#Link the three elements. The outlet pipes of the first link
#to the inlet pipes of the second and the third
#---
Rec1.Inlet = Send.Outlet
Rec1.Outlet.MassSupply = Rec1.Inlet.MassSupply
Rec2.Inlet = Send.Outlet2
Rec2.Outlet.MassSupply = Rec2.Inlet.MassSupply

def SetMassSupply(self, LoopIndx, msup):
self.Loops[LoopIndx].Pump.InputMassSupply[0] = msup

def UpdateComps(self, dt, Tenv):
    for i in range(len(self.Loops)):
        for j in range(1, len(self.Loops[i].ComponentsList)):

```



```

        if isinstance(self.Loops[i].ComponentsList[j], BufferTank) or\
isinstance(self.Loops[i].ComponentsList[j], SolarCollector) or\
isinstance(self.Loops[i].ComponentsList[j], AirHeatPump):
self.Loops[i].ComponentsList[j].CompUpdate(dt, Tenv)
elif isinstance(self.Loops[i].ComponentsList[j], GroundHeatPump) or\
isinstance(self.Loops[i].ComponentsList[j], SourceHeatExchanger):
self.Loops[i].ComponentsList[j].CompUpdate(dt)
        else:
self.Loops[i].ComponentsList[j].CompUpdate()
self.Loops[i].ComponentsList[0].CompUpdate()

def UpdateGraphs(self):
    for i in range(len(self.Loops)):
        for j in range(1,len(self.Loops[i].ComponentsList)):
self.event = self.Loops[i].ComponentsList[j].GraphUpdate()
            if self.event == sg.WIN_CLOSED:
                return self.event
self.event = self.Loops[i].ComponentsList[0].GraphUpdate()
return self.event

```

Π.2 Βιβλιοθήκη Elements

"""

Created on Thu Dec 10 14:29:35 2020

@author: Leonidas Zouloumis

Funcios create the elements of the simulation model and the conditions of mass flow

"""

```
class WaterFlow():
```

```
    def __init__(self, *args):
```

```
        if len(args) == 2:
```

```

self.MassSupply = [args[0]]
self.Temperature = [args[1]]
    else:
self.MassSupply = [0]
self.Temperature = [0]

```

```

class Element():

    def __init__(self, *args):
self.Inlet = WaterFlow()
self.Outlet = WaterFlow()
self.Outlet.MassSupply = self.Inlet.MassSupply

```

Π.3 Κατάστροση εξισώσεων που διέπουν τηδεξαμενή αποθήκευσης

"""

Created on Thu Dec 10 14:29:35 2020

@author: Leonidas Zouloumis

"""

```

class BufferTank(Element):

```

"""

Functions construct the governing equations of Buffer Tank

"""

```

    def __init__(self, *args):
        super().__init__()
        d = 998

        if len(args) == 1 and isinstance(args[0],dict) and len(args[0]) == 5:
self.Height = args[0]['Height']
self.Radius = args[0]['Radius']
self.AreaNodes = args[0]['Number of nodes']

```

```

self.WaterStoredTemp = args[0]['Water temperature']
self.HeatTransmissionCoefficient = args[0]['U']
    else:
self.Height = 0
self.Radius = 0
self.AreaNodes = 3
self.WaterStoredTemp = 0
self.HeatTransmissionCoefficient = 0
self.SideSurfaceArea = 2 * pi * self.Radius * self.Height
self.TopOrBottomAreas = pi * pow(self.Radius, 2)
self.Volume = self.TopOrBottomAreas * self.Height
self.WaterStoredMass = d * self.Volume
    temp = 1 / self.AreaNodes * linspace(0, 1 ,self.AreaNodes)
    temp = temp - mean(temp)
self.TemperatureArea = self.WaterStoredTemp * ones(self.AreaNodes) + temp
self.Inlet.Temperature[0] = self.TemperatureArea[0]
self.Outlet.Temperature[0] = self.TemperatureArea[-1]
    #---
self.ExtGains = [0] * self.AreaNodes
self.NodesOccupied = [0] * self.AreaNodes
self.NodeGrid = self.Height * linspace(0, 1, self.AreaNodes)
    #--- for figures
    # self.Fig = mpl.figure.Figure(figsize=(5, 4), dpi=100)
    # self.cnv = sg.Canvas(key='canvas')
    # sg.theme('Reddit')
    # Layout = [[self.cnv]]
    # self.Win = sg.Window('BufferTank', Layout, finalize=True)
    # self.fcagg = None

```

```

def CompUpdate(self, dt, Tenv):
    Cp = 4180
    IntQTransfer = 0
    for i in range(self.AreaNodes):
        area = self.SideSurfaceArea / self.AreaNodes + self.TopOrBottomAreas * (i == 0 or i ==
self.AreaNodes)
        EnvLosses = self.HeatTransmissionCoefficient * area * (Tenv - self.TemperatureArea[i])
        if i > 0:
            IntQTransfer = self.Inlet.MassSupply[0] /dt * Cp * (self.TemperatureArea[i - 1] -
self.TemperatureArea[i])
        else:
            IntQTransfer = self.Inlet.MassSupply[0] /dt * Cp * (self.Inlet.Temperature[0] - self.TemperatureArea[i])
        #end if
    TotEnChangeRate = EnvLosses + IntQTransfer + self.ExtGains[i]
    self.TemperatureArea[i] = TotEnChangeRate * dt / (self.WaterStoredMass / len(self.TemperatureArea) *
Cp) + self.TemperatureArea[i]
    if i > 0 and self.TemperatureArea[i] <self.TemperatureArea[i - 1]:
        temp = (self.TemperatureArea[i] + self.TemperatureArea[i - 1]) / 2
    self.TemperatureArea[i] = temp
    self.TemperatureArea[i - 1] = temp
    self.Outlet.Temperature[0] = self.TemperatureArea[-1]
    self.WaterStoredTemp = mean(self.TemperatureArea)

def OccupyNodes(self, interstart, interend):
    for i in range(interstart, interend + 1):
        self.NodesOccupied[i] = 1

def GraphUpdate(self):
return 1

```

Π.4 Κατάστρωση εξισώσεων που διέπουν τον εναλλάκτη θερμότητας

"""

Created on Thu Dec 10 14:29:35 2020

@author: Leonidas Zouloumis

"""

```
class SourceHeatExchanger(Element):
```

"""

Functions construct the governing equations of Heat Exchanger

"""

```
def __init__(self, *args):
```

```
    super().__init__()
```

```
    if len(args) == 1:
```

```
self.Effectiveness = args[0]
```

```
    else:
```

```
self.Effectiveness = 0.85
```

```
self.InteractionRange = [0, 0]
```

```
self.Interactor = [0]
```

```
self.TemperatureAreaOut = [[0], [0]]
```

```
self.ExtLoad = [0]
```

```
self.ExtGains = [0]
```

```
self.Cp = 1937
```

```
    def SetInteractor(self, interactor, *args):
```

```
        #NOTE: YOU MUST SET A HEAT EXCHANGER INTO A BUFFER TANK, BEFORE LINKING  
ELEMENTS TOGETHER
```

```
self.Interactor = interactor
```

```
    if isinstance(self.Interactor, BufferTank):
```

```
        if len(args) == 2:
```

```

self.InteractionRange[0] = args[0]
self.InteractionRange[1] = args[1]
self.TemperatureAreaOut = [[0]] * (self.InteractionRange[1] - self.InteractionRange[0] + 1)
    for i in range(len(self.TemperatureAreaOut)):
self.TemperatureAreaOut[i] = [self.Inlet.Temperature[0]]
self.Outlet.Temperature = self.TemperatureAreaOut[-1]
self.ExtLoad = [0] * self.Interactor.AreaNodes
self.ExtGains = [0] * self.Interactor.AreaNodes
    #---
    #will be used to avoid adding two heat exchanger in the same nodes
self.Interactor.OccupyNodes(self.InteractionRange[0], self.InteractionRange[1])
    #---
self.Interactor.ExtGains = self.ExtLoad
elifisinstance(self.Interactor, LoadHeatExchanger):
self.Interactor.SourceInteractor = 0

def CompUpdate(self, dt):
    Cp = 1937
    if isinstance(self.Interactor, BufferTank):
self.TemperatureAreaOut[0][0] = (1 - self.Effectiveness) * self.Inlet.Temperature[0] + self.Effectiveness *
self.Interactor.TemperatureArea[self.InteractionRange[1]]
self.ExtGains[self.InteractionRange[1]] = (self.TemperatureAreaOut[0][0] - self.Inlet.Temperature[0]) *
Cp * self.Inlet.MassSupply[0]
self.ExtLoad[self.InteractionRange[1]] = - self.ExtGains[self.InteractionRange[1]]
        for i in range(1, self.InteractionRange[1] - self.InteractionRange[0] + 1):
self.TemperatureAreaOut[i][0] = (1 - self.Effectiveness) * self.TemperatureAreaOut[i - 1][0] +
self.Effectiveness * self.Interactor.TemperatureArea[self.InteractionRange[1] - i]
self.ExtGains[self.InteractionRange[1] - i] = (self.TemperatureAreaOut[i][0] - self.TemperatureAreaOut[i
- 1][0]) * Cp * self.Inlet.MassSupply[0]/dt

```

```

self.ExtLoad[self.InteractionRange[1] - i] = - self.ExtGains[self.InteractionRange[1] - i]

elif isinstance(self.Interactor, LoadHeatExchanger):
    Cs = self.Inlet.MassSupply[0] * self.Cp
    Cl = self.Interactor.Inlet.MassSupply[0] * self.Interactor.Cp
    if self.Inlet.MassSupply[0] != 0 and self.Interactor.Inlet.MassSupply[0] != 0:
        Cmin = Cl * (Cs > Cl) + Cs * (not(Cs > Cl))
        self.ExtLoad[0] = self.Effectiveness * Cmin * (self.Inlet.Temperature[0] -
self.Interactor.Inlet.Temperature[0])
        self.Outlet.Temperature[0] = self.Inlet.Temperature[0] - (self.ExtLoad[0] / Cs)
        self.Interactor.Outlet.Temperature[0] = self.Interactor.Inlet.Temperature[0] + (self.ExtLoad[0] / Cl)
    else:
        self.ExtLoad[0] = 0

```

```

def GraphUpdate(self):

```

```

    return 1

```

Π.5 Κατάστροση εξισώσεων που διέπουν τον επίπεδο ηλιακό συλλέκτη

```

"""

```

Created on Thu Dec 10 14:29:35 2020

@author: Leonidas Zouloumis

```

"""

```

```

class SolarCollector(Element):

```

```

"""

```

Functions construct the governing equations of Solar Collector

```

"""

```

```

def __init__(self, *args):

```

```

    super().__init__()

```

```

    #---- Collector characteristics

```

```

    if len(args) == 5:

```

```

self.Area = args[0]
    month = args[1]
    day = args[2]
self.ta_n = args[3]
self.Ul = args[4]
    else:
self.Area = 1
    month = 1
    day = 1
self.ta_n = 0.8
self.Ul = 7.5
    self.Fr = 1
self.WaterMassToArea = 1.5
self.WaterStoredMass = self.Area * self.WaterMassToArea
self.WaterStoredTemp = self.Inlet.Temperature[0]
    #--- Collector day clock
self.clock = 0
self.Glist = wt.GetIsolDay(month, day)
self.Isol = 0
self.Efficiency = 0
self.Qsol = 0
self.Qsol_sum = 0

    def GetIsol(self, dt):
self.Isol = self.Glist[wt.GetIndex(self.clock)]
self.clock += dt

    def CompUpdate(self, dt, Tenv):
    Cp = 4180

```



```

self.WaterStoredTemp = self.Inlet.Temperature[0] * (self.clock == 0) + self.WaterStoredTemp *
(self.clock != 0)
self.GetIsol(dt)

#---- Calculate collected energy
if self.Inlet.MassSupply[0] == 0:
self.Qsol = (self.Fr * self.ta_n * self.Isol - self.Fr * self.Ul * (self.WaterStoredTemp - Tenv)) * self.Area
else:
self.Qsol = (self.Fr * self.ta_n * self.Isol - self.Fr * self.Ul * (self.Inlet.Temperature[0] - Tenv)) *
self.Area
self.Qsol_sum += self.Qsol

#---- Update water Temperature
self.WaterStoredTemp = self.WaterStoredTemp + self.Qsol * dt / (self.WaterStoredMass * Cp)
self.Outlet.Temperature[0] = self.WaterStoredTemp

#---- Calculate collector efficiency
if self.Isol != 0:
self.Efficiency = self.Qsol / (self.Isol * self.Area)
else:
self.Efficiency = 0

#---- Update water temperature in collector after fluid movement
self.Qchange = self.Inlet.MassSupply[0] * Cp * self.Inlet.Temperature[0] - self.Outlet.MassSupply[0] *
Cp * self.Outlet.Temperature[0]
self.WaterStoredTemp = self.WaterStoredTemp + self.Qchange / (self.WaterStoredMass * Cp)

defGraphUpdate(self):
return 1

```

Π.6 Κατάστροση εξισώσεων που διέπουν τη γεωθερμική αντλία θερμότητας

""

Created on Thu Dec 10 14:29:35 2020

@author: Leonidas Zouloumis

```

"""
class GroundHeatPump(Element):
"""
Functions construct the governing equations of Ground Heat Pump
"""

def __init__(self):
    super().__init__()
self.COP = 0
self.PowerCons = 1000
self.GroundTemp = 5
self.Qhp_sum = 0
    #----
self.cnt = 0
self.COP_sum = 0

def CompUpdate(self, dt):
    Cp = 4180
    #---- Set COP correlation to Temperature Difference
    a1 = 0.0015
    a2 = - 0.2172
    b = 9.9075
TempDiff = self.Outlet.Temperature[0] - self.GroundTemp
    #---- Calculate COP and Heating Power
self.COP = a1 * pow(TempDiff, 2) + a2 * TempDiff + b
self.Qhp = self.COP * self.PowerCons
    #---- Calculate Water Outlet Temperature
    if self.Inlet.MassSupply[0] != 0:
self.Outlet.Temperature[0] = self.Qhp / (self.Inlet.MassSupply[0] * Cp) + self.Inlet.Temperature[0]
self.Qhp_sum += self.Qhp / self.COP

```

```

        #---
self.COP_sum += self.COP
self.cnt +=1
        else:
self.Outlet.Temperature[0] = self.Inlet.Temperature[0]
self.COP = 0

defGraphUpdate(self):
return 1

```

Π.7 Κατάστρωση εξισώσεων που διέπουν την καταβόθρα θερμότητας

```

"""

```

```

Created on Thu Dec 10 14:29:35 2020

```

```

@author: Leonidas Zouloumis

```

```

"""

```

```

class DrainPipe(Element):

```

```

"""

```

```

Functions construct the governing equations of Drain Pipe

```

```

"""

```

```

def __init__(self, Tw, V):

```

```

    super().__init__()

```

```

self.Qcons = 0

```

```

self.Outlet.Temperature[0] = Tw

```

```

    if V == 0:

```

```

self.Vlist = wt.GetHWconsday()

```

```

        else:

```

```

self.Vlist = wt.GetHWconsday500()

```

```

self.clock = 0

```

```

self.Tcw = Tw

```

```

def GetVwater(self, dt):
Vwater = self.Vlist[wt.GetIndex(self.clock)]
self.clock += dt
    return Vwater

def CompUpdate(self):
    Cp = 4180
self.Outlet.Temperature[0] = self.Tcw
self.Qcons = self.Inlet.MassSupply[0] * Cp * (self.Inlet.Temperature[0] - self.Outlet.Temperature[0])

defGraphUpdate(self):
return 1

```

Π.8 Κατάστροση εξισώσεων που διέπουν την τρίοδη βάννα εξαγωγής

```

"""
Created on Thu Dec 10 14:29:35 2020
@author: Leonidas Zouloumis
"""
class Tpipe2out(Element):
"""
Functions construct the governing equations of Triple Exhaust Valve
"""
def __init__(self):
    super().__init__()
    self.Outlet2 = WaterFlow()
self.Outlet.Temperature = self.Inlet.Temperature
    self.Outlet2.Temperature = self.Inlet.Temperature
self.Outlet.MassSupply = [0]
    self.Outlet2.MassSupply = [0]
self.MassRatio = 0

```

```
def SetMR1(self, value):
```

```
self.MassRatio = value
```

```
self.CompUpdate()
```

```
def CompUpdate(self):
```

```
self.Outlet.MassSupply[0] = self.MassRatio * self.Inlet.MassSupply[0]
```

```
self.Outlet2.MassSupply[0] = self.Inlet.MassSupply[0] - self.Outlet.MassSupply[0]
```

```
defGraphUpdate(self):
```

```
return 1
```

Π.9 Κατάστρωση εξισώσεων που διέπουν την τρίοδη βάννα επιστροφής

```
"""
```

```
Created on Thu Dec 10 14:29:35 2020
```

```
@author: Leonidas Zouloumis
```

```
"""
```

```
class Tpipe2in(Element):
```

```
"""
```

```
Functions construct the governing equations of Triple Return Valve
```

```
"""
```

```
def __init__(self):
```

```
    super().__init__()
```

```
    self.Inlet2 = WaterFlow()
```

```
self.Outlet.MassSupply = [0]
```

```
def CompUpdate(self):
```

```
    if self.Inlet.MassSupply[0] != 0 or self.Inlet2.MassSupply[0] != 0:
```

```
self.Outlet.MassSupply[0] = self.Inlet.MassSupply[0] + self.Inlet2.MassSupply[0]
```

```
self.Outlet.Temperature[0] = (self.Inlet.MassSupply[0] * self.Inlet.Temperature[0] +
self.Inlet2.MassSupply[0] * self.Inlet2.Temperature[0]) / self.Outlet.MassSupply[0]
```

```
else:
```

```
self.Outlet.MassSupply[0] = 0
```

```
self.Outlet.MassSupply[0] = 0
```

```
defGraphUpdate(self):
```

```
return 1
```

Π.10 Κατάστρωση εξισώσεων που διέπουν τον κυκλοφορητή

```
"""
```

```
Created on Thu Dec 10 14:29:35 2020
```

```
@author: Leonidas Zouloumis
```

```
"""
```

```
class Pump(Element):
```

```
"""
```

```
Functions construct the governing equations of Pump
```

```
"""
```

```
def __init__(self, msup, mtemp):
```

```
    super().__init__()
```

```
self.InputMassSupply = [msup]
```

```
self.Outlet.MassSupply = self.InputMassSupply
```

```
self.Outlet.Temperature[0] = mtemp
```

```
def CompUpdate(self):
```

```
self.Outlet.Temperature[0] = self.Inlet.Temperature[0]
```

```
defGraphUpdate(self):
```

```
return 1
```

Π.11 Εισαγωγή αρχείου με τις τιμές ηλιακής ακτινοβολίας

"""

Created on Mon Feb 22 10:00:00 2021

@author: Aggelos Karanasos

Function introduces the file with the solar radiation hourly values of the installation area

"""

```
def GetIsolFile():
```

```
    df = pd.read_excel (r'ATHIhour45.xls')
```

```
    c = df.values
```

```
    a = np.transpose(c)
```

```
Gtot = a[5][:]
```

```
    month = a[0][:]
```

```
    day = a[1][:]
```

```
    return Gtot, month, day
```

Π.12 Επιλογή ωριαίας τιμής ηλιακής ακτινοβολίας για μία μέρα

"""

Created on Mon Feb 22 10:00:00 2021

@author: Aggelos Karanasos

Function selects the hourly value of the solar radiation for one day

"""

```
def GetIsolDay(m, d):
```

```
    G, Month, Day = GetIsolFile()
```

```
cnt = 0
```

```
    first = 0
```

```
    last = 0
```

```
found_first = 0
```

```

found_last = 0
while found_last == 0 and cnt < 8760:
    if Month[cnt] == m and Day[cnt] == d:
        if found_first == 0:
            first = cnt
found_first = 1
    if cnt == len(G)-1:
        last = cnt
found_last = 1
elif cnt > first and found_first == 1:
    last = cnt - 1
found_last = 1
cnt += 1
    G = G[first : last+1]
return G

```

Π.13 Εισαγωγή αρχείου με τις τιμές της θερμοκρασίας περιβάλλοντος

"""

Created on Mon Feb 22 10:00:00 2021

@author: Aggelos Karanasos

Function introduces the file with the ambient temperature values of the installation area

"""

```

def GetTempOut():
    df = pd.read_excel (r'ATHIhour45.xls')
    c = df.values
    a = np.transpose(c)
    Tout = a[8][:]
    month = a[0][:]

```



```
day = a[1][:]
```

```
returnTout, month, day
```

Π.14 Επιλογή ωριαίας τιμής εξωτερικής θερμοκρασίας για μία μέρα

```
"""
```

```
Created on Mon Feb 22 10:00:00 2021
```

```
@author: Aggelos Karanasos
```

```
Function selects the hourly value of the ambient temperature for one day
```

```
"""
```

```
def GetTempOutDay(m, d):
```

```
    Tout, Month, Day = GetTempOut()
```

```
    cnt = 0
```

```
        first = 0
```

```
        last = 0
```

```
    found_first = 0
```

```
    found_last = 0
```

```
        while found_last == 0 and cnt < 8760:
```

```
            if Month[cnt] == m and Day[cnt] == d:
```

```
                if found_first == 0:
```

```
                    first = cnt
```

```
    found_first = 1
```

```
        if cnt == len(Tout)-1:
```

```
            last = cnt
```

```
    found_last = 1
```

```
    elif cnt > first and found_first == 1:
```

```
        last = cnt - 1
```

```
    found_last = 1
```

```
    cnt += 1
```

```
    Tout = Tout[first : last+1]
```

```
returnTout
```

Π.15 Μετατροπή δευτερολέπτων μίας ημέρας σε ώρες

```
"""
```

```
Created on Mon Feb 22 10:00:00 2021
```

```
@author: Aggelos Karanasos
```

```
Function converts the seconds of a day into hours
```

```
"""
```

```
def GetIndex(cnt):
```

```
    h = cnt // 3600
```

```
    if h > 0 and h < 24:
```

```
        index = h - 1
```

```
else:
```

```
    index = 23
```

```
    return index
```

Π.16 Κατανάλωση ZNX

```
"""
```

```
Created on Mon Feb 22 10:00:00 2021
```

```
@author: Aggelos Karanasos
```

```
Function gives the hourly hot water consumption
```

```
"""
```

```
defGetHWconscopy():
```

```
V = [0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 0.5, 0.5, 0.5, 1.5, 1.5, 1, 0.5, 0, 0, 0, 0, 0, 0]
```

```
returnV
```

Π.17 Απομάστευση υγρού από την δεξαμενή των 500 lt

```
"""
```

```
Created on Mon Feb 22 10:00:00 2021
```

@author: Aggelos Karanasos

Function gives the liquid drain from the from the 500 lt tank

"""

```
defGetHWconscopy500():
```

```
V = [20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20]
```

```
returnV
```

Π.18 Υπολογισμός αριθμού ημερών κάθε μήνα

"""

Created on Mon Feb 22 10:00:00 2021

@author: Aggelos Karanasos

Function returns the number of days each month

"""

```
def GetDayList(month):
```

```
    if month == 1 or month == 3 or month == 5 or month == 7 or\  
        month == 8 or month == 10 or month == 12:
```

```
        day_len = 31
```

```
    elif month == 2:
```

```
        day_len = 28
```

```
    else:
```

```
        day_len = 30
```

```
    day_list = 1 * np.linspace(1, day_len, day_len)
```

```
    day_list = list(day_list)
```

```
    returnday_list
```

Π.19 Επιλογή μηνιαίας θερμοκρασίας εδάφους

"""

Created on Mon Feb 22 10:00:00 2021

@author: Aggelos Karanasos

Function selects the month value of ground temperature

"""

```
def GetTw(month):
```

```
    Tw = [11.3, 10.9, 11.8, 14.3, 17.7, 21.6, 24.7, 25.7, 24.2, 21.1, 16.9, 13.5]
```

```
    return Tw[month - 1]
```

Π.20 Διατήρηση δύο σημαντικών ψηφίων

"""

Created on Mon Feb 22 10:00:00 2021

@author: Aggelos Karanasos

Function returns one value with two significant digits

"""

```
def MyRound(*args):
```

```
    rnd_list = list()
```

```
    for i in range(len(args)):
```

```
        rnd_list.append(round(args[i], 2))
```

```
    return rnd_list
```

Π.21 Κατάστρωση του μοντέλου προσομοίωσης

"""

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function calculates the energy consumption of the ground heat pump and the coverage rate by the solar collector for each month and records the result in a list

"""

```
def MultiParamMonth(Huge_list):
```

```
    m = Huge_list[0]
```

```

Thp = Huge_list[1]
Asol = Huge_list[2]
    Tw500 = Huge_list[3]
TwZNX = Huge_list[4]
    month = Huge_list[5]
ZNX_height = Huge_list[6]
day_list = Huge_list[7]
ta_n = Huge_list[8]
    Ul = Huge_list[9]
OutStringList = list()
    for day in day_list:
        dict_500 = {'Height' : 1, 'Radius' : 0.4, 'U' : 0.07, 'Number of nodes' : 10, 'Water temperature' :
Tw500}
dict_ZNX = {'Height' : ZNX_height, 'Radius' : 0.3, 'U' : 0.07, 'Number of nodes' : 10, 'Water temperature' :
TwZNX}
#----
A = LoopManager()
a1 = BufferTank(dict_500)
a2 = BufferTank(dict_ZNX)
b1 = SourceHeatExchanger()
b2 = SourceHeatExchanger()
b3 = SourceHeatExchanger()
c = SolarCollector(Asol, month, day, ta_n, Ul)
d = Tpipe2out()
e = Tpipe2in()
hp = GroundHeatPump()
dr1 = DrainPipe(wt.GetTw(month), 0)
dr2 = DrainPipe(wt.GetTw(month), 1)
#---- Create Loops and place nodes

```

```

A.AddLoop(0.1, 12)
A.AddLoop(0, 12)
A.AddLoop(0, 12)
A.AddLoop(0, 12)
    #---- first loop (Solar, Tpipe [in-out], 2 Exchangers)
A.AddElementToLoop(c, 0)
A.AddElementToLoop(d, 0)
A.AddElementToLoop(b1, 0)
A.AddElementToLoop(b2, 0)
A.AddElementToLoop(e, 0)
    #---- second loop (Buffer)
A.AddElementToLoop(a1, 1)
A.AddElementToLoop(dr2, 1)
    #---- third loop (Buffer, HeatPump)
A.AddElementToLoop(a2, 2)
A.AddElementToLoop(hp, 2)
    #---- four loop (Heatexchanger, DrainPipe)
A.AddElementToLoop(dr1, 3)
A.AddElementToLoop(b3, 3)
    #---- Link Heat Exchanger with Buffer Tank
A.AddExchangerToTank(1, 1, 0, 3, 0, 9)
A.AddExchangerToTank(2, 1, 0, 4, 0, 9)
A.AddExchangerToTank(2, 1, 3, 2, 0, 9)
    #-----
    #---- Link Elements
    #---- first loop
A.LinkElements(0, 0, 1)
A.LinkElements(0, 1, 2)
A.DoubleLinkOut(0, 2, 3, 4)

```

```

A.DoubleLinkIn(0, 3, 4, 5)
A.LinkElements(0, 5, 0)
    #---- second loop
A.LinkElements(1, 0, 1)
A.LinkElements(1, 1, 2)
A.LinkElements(1, 2, 0)
    #---- third loop
A.LinkElements(2, 0, 1)
A.LinkElements(2, 1, 2)
A.LinkElements(2, 2, 0)
    #---- four loop
A.LinkElements(3, 0, 1)
A.LinkElements(3, 1, 2)
A.LinkElements(3, 2, 0)
    #---- control flow to either of tanks
A.SetMassSupply(0, 0.1)
    #---- Temperature data
    Tout = wt.GetTempOutDay(month, day)
    #--- run
TznxList = list()
    T500List = list()
TscList = list()
TdpList = list()
ToutList = list()
    GList = list()
efcList = list()
COPList = list()
ToutHPList = list()
cntList = np.arange(0, 24*3600)

```

```

cnt = 0
    dt = 1
    dT = 0.5
Thp_set = Thp
Tznx_set = 50
    for cnt in range(0, 3600 * 24, dt):
        #---- Solar Collector schedule
        #---- Set names on temperature variables that want to be checked
Tznx = A.Loops[2].ComponentsList[1].Outlet.Temperature[0]
        T500 = A.Loops[1].ComponentsList[1].Outlet.Temperature[0]
Tsc = A.Loops[0].ComponentsList[1].WaterStoredTemp
Tdp = A.Loops[3].ComponentsList[0].Inlet.Temperature[0]
        if Tznx < Tznx_set:
            if Tsc > Tznx + dT:
A.Loops[0].ComponentsList[2].SetMR1(0)
A.SetMassSupply(0, 0.1)
            elif Tsc > T500 + dT and Tsc <= Tznx - dT:
A.Loops[0].ComponentsList[2].SetMR1(1)
A.SetMassSupply(0, 0.1)
            elif Tsc <= T500 - dT or Tsc <= Tznx - dT:
A.SetMassSupply(0, 0)
            elif Tsc > T500 + dT:
A.Loops[0].ComponentsList[2].SetMR1(1)
A.SetMassSupply(0, 0.1)
            elif Tsc <= T500 - dT:
A.SetMassSupply(0, 0)
        #---- Heat Pump schedule
        if (A.Loops[0].ComponentsList[2].MassRatio == 0 and
A.Loops[0].ComponentsList[0].InputMassSupply[0] != 0) or Tznx > Thp_set + dT:

```



```

A.SetMassSupply(2, 0)
elifTznx<= Thp_set - dT:
A.SetMassSupply(2, m)
    #---- DHW schedule
A.SetMassSupply(3, A.Loops[3].ComponentsList[1].GetVwater(dt) * dt/3600)
    #---- Gather Data from Solar Collector
    T = Tout[wt.GetIndex(cnt)]
    G = c.Glist[wt.GetIndex(c.clock)]
TznxList.append(Tznx)
    T500List.append(T500)
TscList.append(Tsc)
    if dr1.Inlet.MassSupply[0] !=0:
TdpList.append(Tdp)
ToutList.append(T)
GList.append(G)
efcList.append(c.Efficiency)
COPLList.append(hp.COP)
ToutHPLList.append(hp.Outlet.Temperature[0])
A.UpdateComps(dt, T)
    print('-----')
    print(cnt)#Hours:: Minutes:: Seconds ---->', cnt//3600,'::', cnt//60,'::', cnt%60)
    print('500lt:', A.Loops[1].ComponentsList[1].WaterStoredTemp)
    print('DHW:', A.Loops[2].ComponentsList[1].WaterStoredTemp)
    print('Collector temperature:', A.Loops[0].ComponentsList[1].Outlet.MassSupply[0])
    print('Solar power collected [W]:', A.Loops[0].ComponentsList[1].Qsol)
    print('Heat Pump Power Output [W]:', A.Loops[2].ComponentsList[2].Qhp *
(A.Loops[2].ComponentsList[2].Inlet.MassSupply[0] != 0))
    print('Flow', A.Loops[0].Pump.InputMassSupply[0])
    print('Qcons', A.Loops[3].ComponentsList[1].Qcons)

```

```

    if cnt%(60*dt) == 0:
        check = A.UpdateGraphs()
        if check == sg.WIN_CLOSED:
            break
#--- COP average
if hp.cnt != 0:
COP_av = hp.COP_sum / hp.cnt
else:
COP_av = 0
#--- Calculate mean coverage percentage
T500_set = 60
Bznx_cover = sum(TdpList)/Tznx_set/len(TdpList)
B500_cover = sum(T500List)/T500_set/len(T500List)
# #--- Print criteria
print("Total Consumption [kWh]:", A.Loops[2].ComponentsList[2].Qhp_sum / (1000 * 3600))
print('Contribution          of          RES          [%]:',
A.Loops[0].ComponentsList[1].Qsol_sum/(A.Loops[0].ComponentsList[1].Qsol_sum +
A.Loops[2].ComponentsList[2].Qhp_sum) * 100)
print('DHW Demand Coverage [%]:', Bznx_cover * 100)
print('Heating Demand Coverage [%]:', B500_cover * 100)
print('Average COP [-]:', COP_av)

Qhptot = A.Loops[2].ComponentsList[2].Qhp_sum / (1000 * 3600)
REScon  =  A.Loops[0].ComponentsList[1].Qsol_sum/(A.Loops[0].ComponentsList[1].Qsol_sum +
A.Loops[2].ComponentsList[2].Qhp_sum) * 100
DHWcover = Bznx_cover * 100
Heatcover = B500_cover * 100
OutString = [str(m), "\t", str(Thp), "\t", str(Asol), "\t", str(Tw500), "\t",\
str(TwZNX), "\t", str(month), "\t", str(ZNX_height), "\t",str(ta_n), "\t",str(UI), "\t",str(Qhptot),
"\t",\

```

```

        str(REScon), "\t", str(DHWcover), "\t", str(Heatcover), "\t", str(int(day)), "\n"]
TwZNX = A.Loops[2].ComponentsList[1].WaterStoredTemp
    Tw500 = A.Loops[1].ComponentsList[1].WaterStoredTemp
OutStringList.append(OutString)
    return OutStringList

```

Π.22 Δημιουργία σεναρίων βελτιστοποίησης για όλους τους μήνες

''''

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function creates optimization scenarios for all months of a year

''''

```

# m = [0.1, 0.5]
# Thp_set = [40, 50]
# A_sol = [2, 7]
# ZNX_height = [0.18, 0.55]
month = range(12)
#----
# day = 5
# f = open("Multiparam_file.txt", 'w')
#
f.write('m\tThp_set\tA_sol\tTw_500\tTw_znx\tmonth\tZNX_height\tQhptot\tREScon\tDHWcover\tHeatc
over\n')
# f.close()
#----
f = open('MonthlyMultiParam.txt','w')
f.write('m\tThp_set\tA_sol\tTw_500\tTw_znx\tmonth\tZNX_height\tta_n\tUI\tQhptot\tDHWcover\tHeatc
over\n')
f.close()

```

```

HUGE_list = list()

# for cnt1 in m:
    # for cnt2 in Thp_set:
        # for cnt3 in A_sol:
            # for cnt6 in month:
                # Tw_znx = GetTw(month)
                # Tw_500 = GetTw(month)
                # for cnt7 in ZNX_height:
                    # HUGE_list.append((cnt1, cnt2, cnt3, cnt4, cnt5, cnt6, cnt7, 5))
                    # MonthGather(cnt1, cnt2, cnt3, Tw_500, Tw_znx, cnt7, cnt6)

# if __name__ == '__main__':
#     mp.Pool(processes = 12).starmap(MultiParam, HUGE_list)

number = 1e1

Collector_ch = [[0.82, 7.5], [0.75, 5], [0.57, 1.82], [0.86, 21.5]]

for cnt1 in month:
    day_list = wt.GetDayList(cnt1 + 1)
    for cnt2 in range(1):
        for cnt3 in Collector_ch:
            Tw_znx = wt.GetTw(cnt1 + 1)
            Tw_500 = wt.GetTw(cnt1 + 1)
            mhp = 0.1 / number * rd.randrange(1 * number, 5 * number)
            Thpset = 10 / number * rd.randrange(4 * number, 5 * number)
            ZNXheight = 0.01 / number * rd.randrange(18 * number, 55 * number)
            Acol = 1 / number * rd.randrange(2 * number, 7 * number)
            HUGE_list.append((mhp, Thpset, Acol, Tw_500, Tw_znx, cnt1 + 1, ZNXheight, day_list, cnt3[0],
            cnt3[1]))

# if __name__ == '__main__':
#     with open('MultiparamFile.txt', 'a') as f:
#         for Result in mp.Pool(processes = 4).imap_unordered(MultiParamMonth, HUGE_list):

```

```
#         for i in Result:
#         f.writelines(i)
#a = MultiParamMonth(HUGE_list[0])
```

Π.23 Υπολογισμός συνολικής μηνιαίας κατανάλωσης ισχύος της αντλίας θερμότητας και του μέσου όρου του μηνιαίου ποσοστού κάλυψης φορτίων ΖΝΧ από τον ηλιακό συλλέκτη

```
"""
```

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function calculates the total monthly power consumption of the ground heat pump and the average of the monthly coverage of hot water loads by the solar collector

```
"""
```

```
def MonthGather(mhp, Thp, Asol, Tw500 ,TwZNX , ZNXHeight, month):
```

```
    ml = np.loadtxt('MultiparamFile.txt', skiprows=1)
```

```
DHW_n = wt.GetDayList(month)
```

```
Heat_n = wt.GetDayList(month)
```

```
Qcons_n = wt.GetDayList(month)
```

```
RES_n = wt.GetDayList(month)
```

```
    #---
```

```
DHWcover = np.average(DHW_n)
```

```
Qhptot = np.sum(Qcons_n)
```

```
REScon = np.average(RES_n)
```

```
Heatcover = np.average(Heat_n)
```

```
    #---
```

```
    for i in range(len(ml)):
```

```
        if ml[i][0] == mhp and ml[i][1] == Thp and ml[i][2] == Asol and\
```

```
            ml[i][6] == ZNXHeight and ml[i][5] == month:
```

```

ind = int(ml[i][11]) - 1
DHW_n[ind] = ml[i][9]
Qcons_n[ind] = ml[i][7]
Heat_n[ind] = ml[i][10]
RES_n[ind] = ml[i][8]
    if RES_n[ind] < 0 or RES_n[ind] > 100:
        print(i)
DHWcover = np.average(DHW_n)
Qhptot = np.sum(Qcons_n)
REScon = np.average(RES_n)
Heatcover = np.average(Heat_n)
    f = open('MonthlyMultiParam.txt','a')
f.writelines([str(mhp), "\t", str(Thp), "\t", str(Asol), "\t", str(Tw500), "\t",\
    str(TwZNX), "\t", str(month), "\t", str(ZNXHeight), "\t",str(Qhptot), "\t",\
    str(REScon), "\t", str(DHWcover), "\t", str(Heatcover), "\n"\
])
f.close()

```

Π.23 Τοποθέτηση των σεναρίων βελτιστοποίησης του κάθε μήνα με τη σειρά ημερών

```

"""

```

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function sets the values of the optimization scenarios of each month in the order of days

```

"""

```

```

f = open('MonthlyMultiParam.txt','w')
f.write('m\tThp_set\tA_sol\tTw_500\tTw_znx\tmonth\tZNX_height\tta_n\tUI\tQhptot\tDHWcover\tHeatc
over\n')

```

```

f.close()

kek = GatherDataCol('MultiparamFile.txt')
HUGE_list = list()

for i in kek:
    Tw_znx = wt.GetTw(int(i[5]))
    Tw_500 = wt.GetTw(int(i[5]))
    HUGE_list.append([i[0], i[1], i[2], Tw_500, Tw_znx, i[3], i[4], i[5], i[6]])

if __name__ == '__main__':
    with open('MonthlyMultiParam.txt', 'a') as f:
        for Result in mp.Pool(processes = 12).imap_unordered(MonthGather, HUGE_list):
            for i in Result:
                f.writelines(i)

```

Π.24 Δημιουργία αρχείων με τα μηνιαία σενάρια βελτιστοποίησης

"""

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function categorizes optimization scenarios by months and records the monthly scenarios in files

"""

```

def GetScenFiles():
    a = np.loadtxt("MonthlyMultiParam.txt", skiprows=1)
    b1 = list()
    b2 = list()
    b3 = list()
    b4 = list()
    b5 = list()
    b6 = list()
    b7 = list()
    b8 = list()

```

```

b9 = list()
b10 = list()
b11 = list()
b12 = list()
for i in range(4799):
    if a[i][5] == 1:
        b1.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 2:
        b2.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 3:
        b3.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 4:
        b4.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 5:
        b5.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 6:
        b6.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 7:
        b7.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 8:
        b8.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 9:
        b9.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 10:
        b10.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 11:
        b11.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])
    if a[i][5] == 12:
        b12.append([a[i][0], a[i][1], a[i][2], a[i][6], a[i][7], a[i][8], a[i][9], a[i][11] ])

```



```

b = [b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12]
cnt = 0
for i in b:
cnt += 1
strcnt = str(cnt)
mystr = "senario" + strcnt + '.csv'
klist = list()
with open(mystr, 'w', newline=") as s:
writer = csv.writer(s)
writer.writerow(["mhp", "Thpset", "Acol", "ZNXheight", "ta_n", "UI", "Qhp", "DHWcov"])
for k in range(len(i[0])):
jlist = list()
for j in range(len(i)):
jlist.append(i[j][k])
xmin = min(jlist)
xmax = max(jlist)
klist.append([xmin, xmax])
writer.writerow([str(klist[0][0]), str(klist[1][0]), str(klist[2][0]), str(klist[3][0]), str(klist[4][0]),
str(klist[5][0]), str(klist[6][0]), str(klist[7][0])])
writer.writerow([str(klist[0][1]), str(klist[1][1]), str(klist[2][1]), str(klist[3][1]), str(klist[4][1]),
str(klist[5][1]), str(klist[6][1]), str(klist[7][1])])
for k in i:
writer.writerow([str(k[0]),str(k[1]), str(k[2]), str(k[3]), str(k[4]), str(k[5]), str(k[6]), str(k[7])])

```

Π.25 Εύρεση σημείων που σχηματίζουν το μέτωπο Pareto

""

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function finds the point that form the Pareto front

```

"""
def Pareto_f(Qhplist, DHWlist):
    Par_Q = list()
    Par_DHW = list()
    cnt_list = list()
    for i in range(len(Qhplist)):
        cnt = 0
        for j in range(len(Qhplist)):
            if i!=j and ((Qhplist[i]>Qhplist[j] and DHWlist[i]<=DHWlist[j]) or (Qhplist[i]>=Qhplist[j] and
DHWlist[i]<DHWlist[j]]):
                cnt += 1
        cnt_list.append(cnt)
        y = min(cnt_list)
        for i in range(len(cnt_list)):
            if cnt_list[i] == y:
                Par_Q.append(Qhplist[i])
                Par_DHW.append(DHWlist[i])
                temp = 0
                for i in range(len(Par_Q)-1):
                    for j in range(i+1,len(Par_Q)):
                        if Par_Q[i] >Par_Q[j]:
                            temp = Par_Q[i]
                Par_Q[i] = Par_Q[j]
                Par_Q[j] = temp
                temp = Par_DHW[i]
                Par_DHW[i] = Par_DHW[j]
                Par_DHW[j] = temp
    return Par_Q, Par_DHW, cnt_list

```

Π.26 Δημιουργία μηνιαίων υποκατάστατων μοντέλων προσομοίωσης (SurrogateSimulationModel)

''''

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function create the surrogate simulation model of each month

''''

def GetAlamoFun(mhp, Thpset, Acol, ZNXheight, ta_n, UI, c):

 if c == 1:

 Qhp = 0.12170566675145376434042 * Thpset - 0.15179151178208735717412 * Acol +
 3.0876852462007495780938 * ZNXheight - 12.564489067489327922544 * ta_n -
 0.20444742880697452935834 * UI - 0.51570230523294480207142 * ln(mhp) +
 0.75622222147874496034348 * ln(Acol) + 3.3129885756632351245798 * ln(UI)

 DHWcov = - 2.2291457771047626934546 * mhp + 1.7298085008064307821485 * Thpset -
 2.1919426260158298980230 * Acol - 68.416439251349956407466 * ta_n - 1.7669409347392224240991
 * UI + 14.696124783411390168908 * ln(Acol) - 3.1294458251143364435620 * ln(ZNXheight) +
 29.943877674330661164959 * ln(UI)

 if c == 2:

 Qhp = 0.11699822670956391834629 * Thpset + 3.1195306960247815020182 * ZNXheight -
 4.7601474050216889821741 * ta_n - 0.51956547212273651581427 * ln(mhp) +
 0.20040684121578680731979 * ln(Acol) + 4.6131802921071445311441 * ln(ta_n)

 DHWcov = 1.9974602378048673667621 * Thpset - 13.000548748915676355864 * ZNXheight +
 8.9999066029610350625489 * ln(Acol) + 48.958834039681029537405 * ln(ta_n)

 if c == 3:

 Qhp = 0.17703503364057204816362 * Thpset + 3.4643769640658508279785 * ZNXheight -
 0.54057073846471281886039 * ln(mhp) - 1.9174998320009470909042 * ln(Thpset) -
 0.32827936803117185871059 * ln(ta_n)

 DHWcov = 2.0573235822742983458511 * Thpset + 1.6144484591272132423256 * ZNXheight +
 5.1611850771351681643750 * ta_n - 2.5170780342895846537488 * ln(Thpset) -
 0.10635569543422912497377 * ln(Acol) - 0.38610345334105833359217 * ln(ZNXheight) -
 4.0296044282907770295310 * ln(ta_n) + 0.36766911476555369509001 * ln(UI)

 if c == 4:

$$\text{Qhp} = 2.4051066908785472264753 * \text{ZNXheight} + 0.70183897528849348290692 * \text{UI} - 0.50803339201023267612811 * \ln(\text{mhp}) + 6.0785636242829479414240 * \ln(\text{Thpset}) + 29.505196879534693721325 * \ln(\text{ta}_n) - 10.704919585164567763513 * \ln(\text{UI})$$

$$\text{DHWcov} = 1.9968018776265772373790 * \text{Thpset} + 1.6327964984103526813186 * \text{ZNXheight} + 0.49569655680527534291624\text{E-}001 * \text{UI} - 0.36122652564986940104674 * \ln(\text{Thpset}) - 0.10776965544069658231585 * \ln(\text{Acol})$$

if c == 5:

$$\text{Qhp} = 0.11619376783507641415216 * \text{Thpset} - 0.20862181783418876079850 * \text{Acol} + 2.6847416625484670227308 * \text{ZNXheight} - 10.205075392104292930640 * \text{ta}_n - 0.14704053285026205677433 * \text{UI} - 0.40390115783991908937622 * \ln(\text{mhp}) + 0.82185837186101406981464 * \ln(\text{Acol}) + 2.3447292900922351854831 * \ln(\text{UI})$$

$$\text{DHWcov} = 2.3108859475533911620460 * \text{Thpset} + 0.45502785065750783743965\text{E-}001 * \text{UI} + 0.52232060427629956045337\text{E-}001 * \ln(\text{mhp}) - 14.802124402073479458863 * \ln(\text{Thpset}) - 0.96956401197023187044088\text{E-}001 * \ln(\text{Acol}) + 0.74494288287970467088428 * \ln(\text{ZNXheight}) - 0.10336190618076457337082 * \ln(\text{UI}) + 42.505484196557929976734$$

if c == 6:

$$\text{Qhp} = 0.11865868895704069496677 * \text{Thpset} + 4.1298037581018913755315 * \text{ZNXheight} - 13.547535958459045346558 * \text{ta}_n - 0.20291051764677284219296 * \text{UI} - 0.37905699424259975405249 * \ln(\text{mhp}) - 0.74857889329912707943038 * \ln(\text{ZNXheight}) + 3.1932384875488946107680 * \ln(\text{UI})$$

$$\text{DHWcov} = 1.9392189143333622425303 * \text{Thpset} - 2.7172448405908369295503 * \text{ZNXheight} - 0.76583262815737018769369 * \text{ta}_n + 0.30924924211970881288147\text{E-}001 * \text{UI} + 1.4123732199153546940096 * \ln(\text{Thpset}) - 0.90124538562568079491655\text{E-}001 * \ln(\text{Acol}) + 1.6676338743685852339382 * \ln(\text{ZNXheight})$$

if c == 7:

$$\text{Qhp} = 0.18271448450140170827183 * \text{Thpset} + 1.6874689013361305889305 * \text{ZNXheight} - 0.30809402542124317281313\text{E-}001 * \text{UI} - 0.33189465779085253194225 * \ln(\text{mhp}) - 2.0109247421800398036851 * \ln(\text{Thpset}) + 0.25032084667352022533393 * \ln(\text{UI})$$

$$\text{DHWcov} = 1.9563723739228500164700 * \text{Thpset} - 3.5326042709865750346410 * \text{ZNXheight} + 0.13996901435458120044331\text{E-}001 * \text{UI} + 1.2542756031875268440956 * \ln(\text{Thpset}) - 0.13346131799069368972788 * \ln(\text{Acol}) + 1.8407412861717966645614 * \ln(\text{ZNXheight})$$

if c == 8:

$$\text{Qhp} = 0.11982059605541853231081 * \text{Thpset} + 1.8578745908488152327465 * \text{ZNXheight} - 4.9746698166442309130275 * \text{ta}_n - 0.31737901771235454972953 * \ln(\text{mhp}) + 3.5178387068173821994321 * \ln(\text{ta}_n)$$

$$\begin{aligned} \text{DHWcov} = & 1.5558614383169995232237 * \text{Thpset} - 4.4044732651009477208959 * \text{ZNXheight} + \\ & 0.23513173103994894352908\text{E-}001 * \text{UI} + 18.882017991765057018938 * \ln(\text{Thpset}) - \\ & 0.62504269543561391531128\text{E-}001 * \ln(\text{Acol}) + 2.0778312355010797318755 * \ln(\text{ZNXheight}) - \\ & 0.11418234120975179202162 * \ln(\text{UI}) - 48.467217806419448322686 \end{aligned}$$

if c == 9:

$$\begin{aligned} \text{Qhp} = & 0.15840469251405578376968 * \text{Thpset} + 3.7620223282926597008213 * \text{ZNXheight} - \\ & 0.31488701392572154835747 * \ln(\text{mhp}) - 2.0612037961719540390959 * \ln(\text{Thpset}) - \\ & 0.59345714160348672638889 * \ln(\text{ZNXheight}) - 0.30589516535665306484759 * \ln(\text{ta}_n) \end{aligned}$$

$$\begin{aligned} \text{DHWcov} = & 1.6163597625829237980355 * \text{Thpset} - 5.1488335944602621196964 * \text{ZNXheight} + \\ & 0.14152629394658342368296\text{E-}001 * \text{UI} + 15.418555259702060311611 * \ln(\text{Thpset}) - \\ & 0.83925357958071142294720\text{E-}001 * \ln(\text{Acol}) + 2.6100746859431729340884 * \ln(\text{ZNXheight}) - \\ & 37.307060665598271498311 \end{aligned}$$

if c == 10:

$$\begin{aligned} \text{Qhp} = & 0.12170416394951480421049 * \text{Thpset} + 2.5107031990527755027642 * \text{ZNXheight} - \\ & 4.9894455375333528479587 * \text{ta}_n - 0.37907686176726157478711 * \ln(\text{mhp}) + \\ & 3.5323351601608434791046 * \ln(\text{ta}_n) \end{aligned}$$

$$\begin{aligned} \text{DHWcov} = & 1.9669232206858398015470 * \text{Thpset} + 0.28134841573376322826361\text{E-}001 * \text{UI} - \\ & 0.82586404310918959392751\text{E-}001 * \ln(\text{Acol}) + 0.73150535745768641149311 * \ln(\text{ZNXheight}) + \\ & 1.6048481596103714164059 \end{aligned}$$

if c == 11:

$$\begin{aligned} \text{Qhp} = & 0.34184039940355875897282 * \text{Thpset} + 3.1885136732257888247943 * \text{ZNXheight} - \\ & 0.44992835795375585306033 * \ln(\text{mhp}) - 9.5951730628869800199254 * \ln(\text{Thpset}) + \\ & 21.678678669772107667768 \end{aligned}$$

$$\begin{aligned} \text{DHWcov} = & 2.0209011303886086352577 * \text{Thpset} + 2.4785704576958091926997 * \text{ZNXheight} - \\ & 0.86634686247953596183180 * \ln(\text{Thpset}) - 0.66524632630166502522684\text{E-}001 * \ln(\text{Acol}) - \\ & 0.68827488325338570529510 * \ln(\text{ZNXheight}) + 0.27209560963143247791507 * \ln(\text{UI}) \end{aligned}$$

if c == 12:

$$\begin{aligned} \text{Qhp} = & 0.17490962427846828974864 * \text{Thpset} + 3.4612712245796100063444 * \text{ZNXheight} - \\ & 0.49139809712500964744564 * \ln(\text{mhp}) - 1.9091554509779531390024 * \ln(\text{Thpset}) + \\ & 0.36367500290916711380707\text{E-}001 * \ln(\text{Acol}) - 0.12334806986190909039447 * \ln(\text{ta}_n) \end{aligned}$$

$$\begin{aligned} \text{DHWcov} = & 1.8490386942712284668033 * \text{Thpset} + 0.68190337305977144488622 * \text{ZNXheight} + \\ & 1.1159995163373603865864 * \text{ta}_n + 0.97111178180224345252158\text{E-}002 * \text{UI} + \\ & 7.2444504181250692909089 * \ln(\text{Thpset}) - 0.62274908721194600913496 * \ln(\text{Acol}) - \\ & 0.32212714244604367941349 * \ln(\text{ZNXheight}) - 22.408176068116329560098 \end{aligned}$$

$$\text{Qhp} = \text{round}(\text{Qhp}, 2)$$

```
DHWcov = round(DHWcov,2)
```

```
return Qhp, DHWcov
```

Π.27 Δημιουργία διαγράμματος με το όριο Pareto

```
"""
```

```
Created on Wed Mar 24 11:49:32 2021
```

```
@author: Aggelos Karanasos
```

```
Function creates diagram with Pareto front
```

```
"""
```

```
def Scatter_data(file,c):
```

```
    a = pd.read_csv(file, skiprows=0)
```

```
    a = a[2:]
```

```
    Qhplist = list()
```

```
    DHWlist = list()
```

```
    for i in range(2,len(a)+2):
```

```
        mhp = a['mhp'][i]
```

```
    Thpset = a['Thpset'][i]
```

```
        Acol = a['Acol'][i]
```

```
    ZNXheight = a['ZNXheight'][i]
```

```
    ta_n = a['ta_n'][i]
```

```
        Ul = a['Ul'][i]
```

```
    Qhp, DHWcov = GetAlamoFun(mhp, Thpset, Acol, ZNXheight, ta_n, Ul, c)
```

```
    Qhplist.append(Qhp)
```

```
    DHWlist.append(DHWcov)
```

```
    DHWlin = np.linspace(80, 102, 99)
```

```
    Qlin_list = GetOptlin(DHWlin)
```

```
    Qlin = Qlin_list[c - 1]
```

```
    [Par_Q, Par_DHW, y] = Pareto_f(Qhplist, DHWlist)
```

```
    plt.scatter(Qhplist,DHWlist)
```

```

plt.plot(Par_Q,Par_DHW, 'r')
plt.plot(Qlin, DHWlin, 'k')
plt.xlabel("Qhp,cons (kwh)")
plt.ylabel("DHW,cov (%)")
plt.legend(labels= ['Pareto frontier','Surrogate Pareto frontier', 'Scenarios', ], loc = (1.07, -0.1))
plt.show()
plt.close()

return Qhplist, DHWlist, Par_Q, Par_DHW, y

```

Π.28 Δημιουργία μηνιαίων υποκατάστατων μοντέλων του ορίου Pareto (SurrogateParetofrontier)

"""

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function createthe surrogate Pareto frontier of each month

"""

def GetOpt(DHWcov):

Qhp1 = 0.28267450230141560885144 * DHWcov - 18.730996345059573116032 * ln(DHWcov) + 60.928272588345450344605

Qhp2 = 0.13951769804067645774914 * DHWcov - 8.9163315149332387221648 * ln(DHWcov) + 29.181023264089507307517

Qhp3 = 0.10845122985158292538976 * DHWcov - 1.7052338674220159919770 * ln(DHWcov)

Qhp4 = 0.10866918452005734641030 * DHWcov - 1.7198023641937716377015 * ln(DHWcov)

Qhp5 = 0.78148260159357996190721E-001 * DHWcov - 1.1774141819636918526726 * ln(DHWcov)

Qhp6 = 0.64019469672884299349391 * DHWcov - 50.632296408142124732876 * ln(DHWcov) + 171.74004512199564942421

Qhp7 = 0.57217529368742658935076 * DHWcov - 43.652806550018965481286 * ln(DHWcov) + 146.30934495541026763021

Qhp8 = 0.31280851927289937686183 * DHWcov - 21.185353760844428450127 * ln(DHWcov) + 68.543245240622368896766

$Q_{hp9} = 0.28096652255923254548620 * DHW_{cov} - 19.238997420781021929770 * \ln(DHW_{cov}) + 62.698184994351656484923$

$Q_{hp10} = 0.57760230246089372396767 * DHW_{cov} - 44.379963190914942572363 * \ln(DHW_{cov}) + 149.30160427343980700243$

$Q_{hp11} = 1.3191441853615055723736 * DHW_{cov} - 110.42651650237981186820 * \ln(DHW_{cov}) + 379.66285076409332077674$

$Q_{hp12} = 0.96129206140291786897301E-001 * DHW_{cov} - 1.5036504963257983913394 * \ln(DHW_{cov})$

$Q_{hp_list} = wt.MyRound(Q_{hp1}, Q_{hp2}, Q_{hp3}, Q_{hp4}, Q_{hp5}, Q_{hp6}, Q_{hp7}, Q_{hp8}, Q_{hp9}, Q_{hp10}, Q_{hp11}, Q_{hp12})$

return Qhp_list

Π.29 Δημιουργία σεναρίων βελτιστοποίησης

"""

Created on Wed Mar 24 11:49:32 2021

@author: Aggelos Karanasos

Function creates optimization scenarios

"""

def RandGenScen(DataSize, Qhp_set, DHW_set):

scenario_list = list()

number = 1e1

Collector_ch = [[0.82, 7.5], [0.75, 5], [0.57, 1.82], [0.86, 21.5]]

for i in range(DataSize):

mhp = 0.1 / number * rd.randrange(1 * number, 5 * number)

Thpset = 10 / number * rd.randrange(4 * number, 5 * number)

ZNXheight = 0.01 / number * rd.randrange(18 * number, 55 * number)

Acol = 1 / number * rd.randrange(2 * number, 7 * number)

Coll = rd.randrange(0,3)

Cc = Collector_ch[Coll]


```

scenario_list.append([round(mhp, 2), round(Thpset, 2), round(ZNXheight, 2), round(Acol, 2),Cc[0],
Cc[1], Qhp_set, DHW_set])

returnscenario_list

```

Π.30 Υπολογισμός ελάχιστου αθροίσματος σχετικών τετραγωνικών σφαλμάτων

"""

CreatedonWedMar 24 11:49:32 2021

@author: Aggelos Karanasos

Function calculates the minimum sum of square relative error

"""

```
def CheckOpt(scenario_list):
```

```
    mhp = scenario_list[0] # mhp
```

```
    Thpset = scenario_list[1] # Thpset
```

```
    ZNXheight = scenario_list[2] # ZNXheight
```

```
    Acol = scenario_list[3] # Acol
```

```
    ta_n = scenario_list[4] # ta_n
```

```
    UI = scenario_list[5] # UI
```

```
    Qhp_set = scenario_list[6] # Set Qhp Value
```

```
    DHW_set = scenario_list[7] # Set DHWcov Value
```

```
    SSE_tot = 0
```

```
    for i in range(len(Qhp_set)):
```

```
        Qhp, DHWcov = GetAlamoFun(mhp, Thpset, Acol, ZNXheight, ta_n, UI, i + 1)
```

```
        SSE = pow((Qhp - Qhp_set[i])/ Qhp_set[i],2) + pow((DHWcov - DHW_set)/DHW_set,2)
```

```
    SSE_tot += pow(SSE,2)
```

```
    return scenario_list, SSE_tot
```

Π.31 Εύρεση βέλτιστων λύσεων

"""

CreatedonWedMar 24 11:49:32 2021

@author: Aggelos Karanasos

This script finds the best solution from 30,000 optimization scenarios

```
"""
```

```
if __name__ == '__main__':
```

```
    DHWcov = 95
```

```
    Qhp = GetOpt(DHWcov)
```

```
        Solution = list()
```

```
scen = list()
```

```
SSE_keys = list()
```

```
    for Scenarios in mp.Pool(processes = 4).map(CheckOpt, RandGenScen(30000, Qhp, DHWcov)):
```

```
        Solution.append(Scenarios)
```

```
            for i in range(len(Solution)):
```

```
                scen.append(Solution[i][0])
```

```
                SSE_keys.append(Solution[i][1])
```

```
kek = np.linspace(0, 29999, 30000)
```

```
    di = dict(zip(kek, SSE_keys))
```

```
    best = min(di, key= lambda k: di[k])
```

```
best_scen = [scen[int(di[best])], di[best]]
```