ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Μεταπτυχιακή Εργασία

ΣΥΝΔΥΑΣΤΙΚΗ ΑΠΑΡΙΘΜΗΣΗ ΓΙΑ ΒΕΛΤΙΣΤΗ ΥΠΟΒΟΛΗ ΠΡΟΣΦΟΡΩΝ ΣΕ ΗΜΕΡΗΣΙΕΣ ΑΓΟΡΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ

ΧΡΙΣΤΟΔΟΥΛΟΣ ΜΑΤΣΙΓΚΟΣ

Επιβλέπων: Δρ. ΓΕΩΡΓΙΟΣ ΚΟΖΑΝΙΔΗΣ

Υπεβλήθη για την εκπλήρωση μέρους των

απαιτήσεων για την απόκτηση του

Μεταπτυχιακού Διπλώματος Ειδίκευσης

2016

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής (Επιβλέπων)	Δρ. Κοζανίδης Γεώργιος Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Δεύτερος Εξεταστής	Δρ. Λυμπερόπουλος Γεώργιος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Τρίτος Εξεταστής	Δρ. Σαχαρίδης Γεώργιος Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Πρώτα απ’ όλα θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή της μεταπτυχιακής μου εργασίας, κ. Γεώργιο Κοζανίδη , για την πολύτιμη καθοδήγησή και τη βοήθεια του καθ’ όλη τη διάρκεια της δουλειάς μου. Επίσης είμαι ευγνώμων στα υπόλοιπα μέλη της εξεταστικής επιτροπής Καθηγητές κ. Γεώργιο Λυμπερόπουλο και κ. Γεώργιο Σαχαρίδη για την προσεκτική ανάγνωση της εργασίας μου και για τις πολύτιμες υποδείξεις τους. Τέλος, θέλω να εκφράσω την ευγνωμοσύνη μου στους γονείς μου και στη σύζυγο μου για την διαρκή τους υποστήριξη, που επέτρεψε την επιτυχή διεκπεραίωση των σπουδών μου στους οποίους και αφιερώνω αυτήν την μεταπτυχιακή εργασία.

Χριστόδουλος Ματσίγκος

ΣΥΝΔΥΑΣΤΙΚΗ ΑΠΑΡΙΘΜΗΣΗ ΓΙΑ ΒΕΛΤΙΣΤΗ ΥΠΟΒΟΛΗ ΠΡΟΣΦΟΡΩΝ ΣΕ ΗΜΕΡΗΣΙΕΣ ΑΓΟΡΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ

ΧΡΙΣΤΟΔΟΥΛΟΣ ΜΑΤΣΙΓΚΟΣ

Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών, 2016

Επιβλέπων Καθηγητής: Δρ. Γεώργιος Κοζανίδης, Επίκουρος Καθηγητής Επιχειρησιακής Έρευνας

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται το πρόβλημα της εύρεσης βέλτιστων προσφορών προς υποβολή από έναν παραγωγό που συμμετέχει σε μια αγορά ενέργειας με χρονικό ορίζοντα πολλών περιόδων. Γίνεται η υπόθεση ότι ο παραγωγός έχει πλήρη επίγνωση των παραμέτρων της αγοράς, δηλαδή των υποβολών προσφοράς των υπόλοιπων παραγωγών, καθώς και της ζήτησης για ενέργεια σε κάθε περίοδο. Το πρόβλημα μορφοποιείται ως ένα μοντέλο μεικτού ακέραιου διεπίπεδου (bilevel) προγραμματισμού, με τον μεμονωμένο παραγωγό να μεγιστοποιεί το κέρδος του στην αντικειμενική συνάρτηση του άνω επιπέδου. Στο κάτω επίπεδο, ένας Ανεξάρτητος Διαχειριστής του Συστήματος εκκαθαρίζει την αγορά και καθορίζει την ποσότητα ενέργειας που θα προσφέρει ο κάθε συμμετέχων παραγωγός, έτσι ώστε να ικανοποιείται η ζήτηση για ενέργεια στο ελάχιστο συνολικό κόστος βάσει των προσφορών που υποβάλλονται. Αναπτύσσεται ένας αλγόριθμος συνδυαστικής απαρίθμησης που επιλύει το παραπάνω πρόβλημα και βρίσκει το βέλτιστο συνδυασμό των προσφορών για κάθε χρονική περίοδο που μπορεί να υποβάλλει ο μεμονωμένος παραγωγός έτσι ώστε να μεγιστοποιήσει το κέρδος του. Στη συνέχεια, παρουσιάζεται η εφαρμογή του αλγορίθμου πάνω σε δύο ρεαλιστικά παραδείγματα και μελετάται η απόδοση του. Τέλος, παρουσιάζονται κάποιες εναλλακτικές μέθοδοι εφαρμογής του αλγορίθμου όπου χρησιμοποιούνται ορισμένοι ευρετικοί τρόποι επίλυσης με στόχο την βελτίωση της υπάρχουσας λύσης.

Πίνακας Περιεχομένων

Κατάλογος Πινάκων

Κατάλογος Διαγραμμάτων

Κατάλογος Ακρωνυμίων

ΑΔΣ : Ανεξάρτητος Διαχειριστής Συστήματος

ΟΤΣ : Οριακή τιμή συστήματος

ΑΑΠ: Αποζημίωση Ανάλογη της υποβάλλουσας Προσφοράς

MIBP: Mixed Integer Bilevel Programming

Κεφάλαιο 1 Εισαγωγή

Σε αυτό το κεφάλαιο, παρουσιάζονται πληροφορίες εισαγωγικού χαρακτήρα που αφορούν τις αγορές ενέργειας, τη σημασία τους και τον τρόπο λειτουργίας τους. Οι πληροφορίες αυτές δίνουν το κίνητρο και το υπόβαθρο αυτής της μεταπτυχιακής εργασίας. Επιπλέον, περιγράφονται συνοπτικά οι βασικές ενότητες της εργασίας.

1.1 Επισκόπηση Διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία αποτελείται από έξι (6) κεφάλαια, συγκεκριμένα:

- Κεφάλαιο 1^{ο} : Στο κεφάλαιο αυτό γίνεται μια εισαγωγή στις αγορές ενέργειας και στον τρόπο λειτουργίας της καθώς επίσης παρουσιάζεται και η επισκόπηση της εργασίας.

- Κεφάλαιο 2^{ο} : Στο δεύτερο κεφάλαιο γίνεται μια εισαγωγή στον μεικτό ακέραιο διεπίπεδο γραμμικό προγραμματισμό καθώς αυτός αποτελεί το υπόβαθρο για την ανάπτυξη του μαθηματικού μοντέλου στη συνέχεια. Επίσης, παρατίθεται η βιβλιογραφική ανασκόπηση για το πρόβλημα που μελετάται.

- Κεφάλαιο 3^{ο} : Στο κεφάλαιο αυτό διατυπώνεται το πρόβλημα το οποίο μελετάται και αναπτύσσεται το μαθηματικό μοντέλο το οποίο το περιγράφει. Στη συνέχεια παρουσιάζονται οι εναλλακτικές μέθοδοι αποζημίωσης του μεμονωμένου παραγωγού και μελετάται κατά πόσο επηρεάζουν τη βέλτιστη λύση.

- Κεφάλαιο 4^{ο} : Στο τέταρτο κεφάλαιο και αφού έχει τελειώσει η σύντομη θεωρητική ανασκόπηση, αναπτύσσεται ο αλγόριθμος επίλυσης του ανωτέρω προβλήματος βελτιστοποίησης στην πιο σύνθετη περίπτωση των πολλαπλών περιόδων και αναλύεται λεπτομερώς το κάθε βήμα εκτέλεσής του.

- Κεφάλαιο 5^{ο} : Στο πέμπτο κεφάλαιο παρουσιάζονται και σχολιάζονται τα πειραματικά αποτελέσματα από την εφαρμογή του αλγορίθμου σε δύο ρεαλιστικά παραδείγματα και μελετάται η απόδοσή του. Τέλος, γίνεται σύγκριση των αποτελεσμάτων μεταξύ των δύο εναλλακτικών μεθόδων αποζημίωσης για το μεμονωμένο παραγωγό και πραγματοποιείται η ανάλυση ευαισθησίας του προβλήματος.

- Κεφάλαιο 6^{ο} : Στο τελευταίο κεφάλαιο αναφέρονται τα συμπεράσματα της παρούσας διπλωματικής εργασίας και τα ζητήματα που θα μπορούσαν να αποτελέσουν αντικείμενο μελλοντικής έρευνας.

- Παράρτημα : Στο παράρτημα παρατίθεται ο αλγόριθμος συνδυαστικής απαρίθμησης που χρησιμοποιήθηκε για την επίλυση του παραπάνω προβλήματος βελτιστοποίησης.

1.2 Η σημασία της ενέργειας

Η ενέργεια είναι ένα ουσιαστικό μέρος της καθημερινής μας ζωής. Τίποτα δεν θα γινόταν χωρίς ενέργεια. Εξαρτόμαστε από τις εκατοντάδες των διαφορετικών τρόπων με τους οποίους κάνει αισθητή την παρουσία της, καθώς οι άνθρωποι χρειάζονται ενέργεια για να κινηθούν και οι μηχανές χρησιμοποιούν ενέργεια για να λειτουργήσουν. Η ενέργεια είναι αναμφίβολα μια υπηρεσία ζωτικής σημασίας για τον άνθρωπο αφού αποτελεί σημαντικό παράγοντα για την επιβίωσή του (θέρμανση, ψύξη, φωτισμός, μεταφορές, γεωργία, παραγωγή τροφίμων και άλλων αγαθών πρώτης ανάγκης), την ανάπτυξη της βιομηχανίας και των υπηρεσιών, την ανάπτυξη της οικονομίας και του πολιτισμού. Στη σημερινή εποχή αποτελεί τη βάση της παραγωγής και από την οπτική της κατανάλωσης είναι ένα αγαθό συνεχώς αυξανόμενης ζήτησης χωρίς τη δυνατότητα υποκατάστασης.

1.3 Εισαγωγή στις αγορές ενέργειας

Οι αγορές ενέργειας είναι αγορές στις οποίες το προϊόν προς προμήθεια και ανταλλαγή είναι η ενέργεια. Οι αγορές ενέργειας μπορεί να αναφέρονται εκτός από την εμπορευματοποίηση της ηλεκτρικής ενέργειας και σε άλλες μορφές ενέργειας. Η δημιουργία τέτοιων αγορών είναι αποτέλεσμα της ανάπτυξης μιας αντίστοιχης ενεργειακής πολιτική από την κυβέρνηση μιας χώρας η οποία ενθαρρύνει την σύσταση μιας βιομηχανίας ενέργειας με τέτοιον τρόπο ώστε να διευκολύνει τον ελεύθερο και υγιή ανταγωνισμό προς όφελος του καταναλωτή. Η λογική της λειτουργίας της αγοράς με τον τρόπο αυτό βασίζεται σε σενάρια πλήρους ανταγωνισμού, καθώς οι πολλοί μικροπαραγωγοί θα ανταγωνίζονται για να πετύχουν μεγαλύτερη απορρόφηση της ζήτησης και στα πλαίσια αυτά θα προσφέρουν την ενέργεια σε χαμηλότερη τιμή που θα εξασφαλίζει τη συμμετοχή τους στην αγορά.

Η λειτουργία της αγοράς ενέργειας και η ασκούμενη ενεργειακή πολιτική παρουσιάζουν ιδιαίτερο ενδιαφέρον. Αν και ο συγκεκριμένος σχεδιασμός της αγοράς που έχει υιοθετηθεί διαφέρει από χώρα σε χώρα, πολλές από τις βασικές αρχές παραμένουν λίγο πολύ οι ίδιες. Οι περισσότεροι σχεδιασμοί έχουν θεσπίσει μία χονδρική και μία λιανική αγορά ηλεκτρικής ενέργειας που λειτουργούν σε μακροπρόθεσμους και βραχυπρόθεσμους χρονικούς ορίζοντες. Στο επίπεδο του ημερήσιου προγραμματισμού της χονδρικής αγοράς ηλεκτρικής ενέργειας, οι παραγωγοί υποβάλλουν ελεύθερα προσφορές (οι οποίες συνήθως υπόκεινται σε κάποιο ανώτατο όριο) για την παραγωγή ενέργειας. Ένας ανεξάρτητος διαχειριστής του συστήματος (ΑΔΣ), εκκαθαρίζει την αγορά, κατανέμοντας ποσότητες ενέργειας στους συμμετέχοντες παραγωγούς, έτσι ώστε να ελαχιστοποιηθεί το συνολικό κόστος που απαιτείται για την ικανοποίηση της ζήτησης για ενέργεια, βάσει των προσφορών που υποβάλλονται. Στην Ελλάδα τον ρόλο αυτού του διαχειριστή έχει η Ανώνυμη Εταιρεία Διαχείρισης Ελληνικού Συστήματος Μεταφοράς Ηλεκτρικής Ενέργειας (ΔΕΣΜΗΕ). Ο ΔΕΣΜΗΕ φροντίζει να υπάρχει ανά πάσα στιγμή ισορροπία παραγωγής και κατανάλωσης και η ενέργεια να παρέχεται κατά τρόπο αξιόπιστο, ασφαλή και ποιοτικά αποδεκτό.

1.4 Σημασία των αγορών ενέργειας

Οι κυριότεροι δύο λόγοι για τη δημιουργία μιας αγοράς ενέργειας είναι οι ακόλουθοι: η εξασφάλιση της ασφαλούς λειτουργίας του συστήματος ενέργειας και η διευκόλυνση της οικονομικότερής του λειτουργίας.

Η ασφάλεια αποτελεί την πιο σημαντική πτυχή της λειτουργίας του συστήματος, είτε πρόκειται για μια ελεγχόμενη λειτουργία είτε για μια αναδιαρθρωμένη αγορά ενέργειας. Σε ένα ελεγχόμενο περιβάλλον, η ασφάλεια μπορεί να διευκολυνθεί με τη χρήση των διαφορετικών υπηρεσιών που διατίθενται στην αγορά. Η οικονομική λειτουργία της αγοράς ηλεκτρικής ενέργειας θα μειώσει το κόστος χρησιμοποίησης ηλεκτρικής ενέργειας. Αυτό αποτελεί και ένα πρωταρχικό κίνητρο για την αναδιάρθρωση και την ενίσχυση της ασφάλειας του συστήματος ηλεκτρικής ενέργειας μέσω των οικονομικών του. Για να συμβεί αυτό, πρέπει να σχεδιαστούν κατάλληλες στρατηγικές στις αγορές βασισμένες στις απαιτήσεις του συστήματος ηλεκτρικής ενέργειας. Για παράδειγμα, χρηματοδοτικά μέσα, όπως οι συμβάσεις για τις διαφορές (Contracts For Differences - CDFs), οι συμβάσεις για τη συμφόρηση κατά τη μεταφορά (Transmission Congestion Contracts - TCCs) και τα δικαιώματα μεταφοράς ανά εταιρία (Firm Transmission Rights - FTRs), θα μπορούσαν να χρησιμοποιηθούν για την αντιστάθμιση των κινδύνων μεταβλητότητας. Εκτός αυτού, εργαλεία παρακολούθησης επινοούνται σε αρκετές αγορές ώστε να αποφευχθεί μια πιθανή κυριαρχία των ισχυρών στην αγορά ηλεκτρικής ενέργειας.

Κεφάλαιο 2 Εισαγωγή στον μεικτό ακέραιο διεπίπεδο γραμμικό προγραμματισμό

2.1 Γενικά περί γραμμικού προγραμματισμού

Γραμμικός προγραμματισμός (Linear programming –LP) ή γραμμική βελτιστοποίηση (Linear optimization) είναι μία μαθηματική μέθοδος μέσω της οποίας προσδιορίζεται ένας τρόπος ώστε να επιτευχθεί το βέλτιστο αποτέλεσμα σε ένα μαθηματικό μοντέλο, δεδομένης μιας σειράς περιορισμών που εκφράζονται ως γραμμικές συναρτήσεις. Το βέλτιστο αυτό αποτέλεσμα είναι συνήθως η μεγιστοποίηση του κέρδους ή η ελαχιστοποίηση του κόστους. Ο γραμμικός προγραμματισμός υπάγεται στη γενικότερη κατηγορία του μαθηματικού προγραμματισμού όπως αναφέρθηκε παραπάνω.

Η βελτιστοποίηση της γραμμικής συνάρτησης που ονομάζεται αντικειμενική αποτελεί τον κύριο σκοπό. Οι περιορισμοί έχουν τη μορφή ανισο/ισοτήτων και οι μεταβλητές είναι πραγματικοί θετικοί αριθμοί που ονομάζονται μεταβλητές απόφασης. Οι λύσεις εμφανίζονται στη εφικτή περιοχή η οποία καθορίζεται από τους περιορισμούς. Η βέλτιστη λύση εντοπίζεται εντός της εφικτής περιοχής και είναι η τιμή που μεγιστοποιεί ή ελαχιστοποιεί την αντικειμενική συνάρτηση.

2.2 Το πρόβλημα του διεπίπεδου προγραμματισμού

Τα προβλήματα Διεπίπεδου Προγραμματισμού παρουσιάστηκαν για πρώτη φορά από τον V. Stackelberg (1934). Μπορούν να παρομοιαστούν με ιεραρχικά παίγνια στο οποίο συμμετέχουν δύο φορείς λήψης αποφάσεων, όπου ο πρώτος (που ονομάζεται Leader - ηγέτης) έχει τον πρώτο λόγο ή αλλιώς κάνει την πρώτη επιλογή και ο δεύτερος (που ονομάζεται Follower - ακόλουθος) έχει την βέλτιστη αντίδραση με βάση ήδη υφιστάμενη επιλογή του ηγέτη. Στόχος του ηγέτη είναι να βρει μια επιλογή, η οποία σε συνδυασμό με την βέλτιστη αντίδραση του ακόλουθου θα βελτιστοποιήσει την αντικειμενική συνάρτηση του ηγέτη. Είναι ένα είδος μαθηματικού προγραμματισμού το οποίο έχει ένα ιδιαίτερο χαρακτηριστικό: Περιέχεται εμφωλευμένο πρόβλημα βελτιστοποίησης μέσα στους περιορισμούς. Είναι ένα είδος πολυεπίπεδου προγραμματισμού και πιο συγκεκριμένα δύο επιπέδων.

Τα προβλήματα που επιλύνονται με τη χρήση του συγκεκριμένου προγραμματισμού δημιουργήθηκαν λόγω της ανάγκης εξέλιξης δύο άλλων σχετικών με αυτόν επιστημονικών κλάδων, την Λογική επέκταση των προβλημάτων του μαθηματικού προγραμματισμού και της γενίκευση ενός συγκεκριμένου προβλήματος στην θεωρία παιγνίων.

Βρίσκει εφαρμογή σε πολλούς τομείς, όπως οικονομικά, διάφορους τομείς της μηχανικής γενικότερα, κατανομή των πόρων, σχεδιασμός δικτύων μεταφοράς, διαχείριση συμφόρησης κ.α. και κυρίως σε πολυεπίπεδα συστήματα (multilevel systems) αυτών των τομέων. Για παράδειγμα, η αντικειμενική συνάρτηση που έχει να κάνει με ένα τμήμα μιας εταιρείας είναι μερικώς ορισμένη και επηρεασμένη από μεταβλητές που ελέγχονται από άλλα τμήματα της (S.Franke et.al 1991).

2.2.1 Γενικό Μαθηματικό Μοντέλο Προβλημάτων Διεπίπεδου Προγραμματισμού

Κάθε πρόβλημα βελτιστοποίησης διέπεται από το αντίστοιχο μαθηματικό του μοντέλο το οποίο περιγράφει με μαθηματικούς όρους το πρόβλημα και επιλύεται με συγκεκριμένες γνωστές μεθόδους, όπως η μέθοδος Simplex που αφορά στην επίλυση προβλημάτων γραμμικού προγραμματισμού. Το μαθηματικό μοντέλο το οποίο περιγράφει τα προβλήματα μεικτού διεπίπεδου γραμμικού προγραμματισμού στη γενική τους μορφή είναι το ακόλουθο:

Αντικειμενική συνάρτηση:

$$\underset{x \in X}{Min} \; F(x, y) \tag{2.1}$$

Περιορισμοί:

$$\text{s.t. } G(x, y) \leq 0 \tag{2.2}$$

$$\underset{y \in Y}{Min} \; f(x, y) \tag{2.3}$$

$$\text{s.t } g(x, y) \leq 0 \tag{2.4}$$

$$x, y \geq 0 \tag{2.5}$$

Οι μεταβλητές του προβλήματος διαιρούνται σε δύο κατηγορίες, οι μεταβλητές υψηλότερου επιπέδου x και οι μεταβλητές χαμηλότερου επιπέδου y.

Ομοίως, οι συναρτήσεις F και f είναι οι αντικειμενικές συναρτήσεις του ανώτερου και του κατώτερου επιπέδου , αντίστοιχα, ενώ οι συναρτήσεις G και g ονομάζονται ανώτερου επιπέδου και κατώτερου επιπέδου περιορισμοί αντίστοιχα. Περιορισμούς ανώτερου επιπέδου περιλαμβάνουν οι μεταβλητές και των δύο επιπέδων και διαδραματίζουν έναν πολύ συγκεκριμένο ρόλο.

2.3 Βιβλιογραφική ανασκόπηση

Ένα σημαντικό κομμάτι της παρούσας εργασίας έχει αντιμετωπίσει το πρόβλημα της ανάπτυξης βέλτιστων στρατηγικών υποβολής προσφορών για παραγωγούς ενέργειας οι οποίοι συμμετέχουν σε μια αγορά ημερήσιου προγραμματισμού ηλεκτρικής ενέργειας. Πολλές από τις δημοσιευμένα άρθρα (π.χ., Garcia-Martos et al., 2007) προτείνουν μεθόδους πρόβλεψης για την πρόγνωση της τιμής εκκαθάρισης της αγοράς, δεδομένου ότι το κέρδος του κάθε παραγωγού εξαρτάται σε σημαντικό βαθμό από την τιμή αυτή. Άλλοι συγγραφείς (π.χ., Ragupathi and Das, 2004) έχουν αναπτύξει στοχαστικά μοντέλα προγραμματισμού, προκειμένου να αντιμετωπίσουν τις αβεβαιότητες που παρουσιάζουν ορισμένες από τις παραμέτρους του προβλήματος. Η παρούσα ανασκόπηση εστιάζεται κυρίως σε διεπίπεδα μοντέλα γραμμικού προγραμματισμού που έχουν αναπτυχθεί για βέλτιστη στρατηγική υποβολής προσφορών σε αγορές ηλεκτρικής ενέργειας. Οι περισσότεροι συγγραφείς που έχουν αναπτύξει διεπίπεδα μοντέλα προγραμματισμού για βέλτιστη στρατηγική υποβολής προσφορών σε αγορές ηλεκτρικής ενέργειας χρησιμοποιούν είτε μια κατάλληλη αναδιατύπωση του προβλήματος, είτε μια ευρετική διαδικασία, για την επίλυσή του. Οι Weber and Overbye (2002) ανέπτυξαν ένα διεπίπεδο μοντέλο βελτιστοποίησης που μεγιστοποιεί την ευημερία του μεμονωμένου παραγωγού και πρότειναν έναν επαναληπτικό αλγόριθμο αναζήτησης για την επίλυσή του. Χρησιμοποίησαν επίσης τον αλγόριθμο αυτόν για τον προσδιορισμό σημείων ισορροπίας Nash. Οι Gountis and Bakirtzis (2004) και Fampa et al. (2008) ανέπτυξαν διεπίπεδα στοχαστικά μοντέλα βελτιστοποίησης για το ίδιο πρόβλημα. Στο πρώτο άρθρο, οι συγγραφείς χρησιμοποίησαν προσομοίωση Monte-Carlo και γενετικούς αλγόριθμους για την εύρεση της βέλτιστης λύσης, ενώ στη δεύτερη, οι συγγραφείς χρησιμοποίησαν μια ευρετική διαδικασία και μια μεικτή ακέραια αναδιατύπωση του προβλήματος. Μεταξύ άλλων, οι Pereira et al. (2005), Barroso et al. (2006), Bakirtzis et al. (2007) και Ruiz and Conejo (2009) ανέπτυξαν διεπίπεδα μοντέλα βελτιστοποίησης και χρησιμοποίησαν τις συνθήκες βελτιστότητας πρώτης τάξης του κάτω προβλήματος προκειμένου να μετατρέψουν αυτά τα μοντέλα σε μαθηματικά προβλήματα με περιορισμούς ισορροπίας (mathematical problems with equilibrium constraints). Τα προκύπτοντα μη γραμμικά προβλήματα μετατράπηκαν στη συνέχεια σε μεικτά ακέραια γραμμικά προβλήματα μέσω κατάλληλων μορφοποιήσεων και επιλύθηκαν μέσω γενικών λογισμικών βελτιστοποίησης. Οι Li et al. (2004) ανέπτυξαν επίσης ένα διεπίπεδο μοντέλο βελτιστοποίησης και το χρησιμοποίησαν για την αναζήτηση σημείων ισορροπίας Nash. Οι Hu and Ralph (2007) εξέτασαν ένα διεπίπεδο μοντέλο βελτιστοποίησης, το οποίο μετέτρεψαν σε ένα μαθηματικό πρόγραμμα με περιορισμούς ισορροπίας. Στη συνέχεια, απέδειξαν ικανές συνθήκες για αμιγούς-στρατηγικής (pure-strategy) σημεία ισορροπίας Nash. Οι Hobbs et al. (2000) ανέπτυξαν ένα διεπίπεδο μοντέλο βελτιστοποίησης και χρησιμοποίησαν έναν αλγόριθμο ποινής εσωτερικών σημείων (penalty interior point algorithm) για την επίλυσή του. Οι Li and Shahidehpour (2005) ανέπτυξαν ένα διεπίπεδο μοντέλο βελτιστοποίησης και χρησιμοποίησαν συναρτήσεις ευαισθησίας και μια πρωτεύουσα-δυϊκή μεθοδολογία εσωτερικών σημείων (primal-dual interior point method) προκειμένου να το επιλύσουν. Στην παρούσα εργασία χρησιμοποιούνται δυαδικές μεταβλητές για τη μοντελοποίηση της εκκίνησης των μονάδων παραγωγής ενέργειας και επιβάλλονται αυστηρώς θετικά κάτω όρια για τις ποσότητες ενέργειας που οι μονάδες αυτές θα προσφέρουν αν συμμετάσχουν στην αγορά. Τελικά, αναπτύσσεται μια αλγοριθμική μεθοδολογία είναι σε θέση να βρει την ολικά βέλτιστη λύση του προβλήματος. Η μεθοδολογία αυτή δεν είναι ιδιαίτερα ευφυής, καθώς βασίζεται σε μία εξαντλητική αναζήτηση του εφικτού χώρου για την εύρεση της βέλτιστης λύσης. Ως εκ τούτου, απαιτεί υπολογιστικούς πόρους οι οποίοι για προβλήματα ρεαλιστικού μεγέθους είναι απαγορευτικοί. Παρόλα αυτά, και με δεδομένη την παντελή έλλειψη γενικών αλγορίθμων και λογισμικών πακέτων για την επίλυση προβλημάτων διεπίπεδου προγραμματισμού, ο συγκεκριμένος αλγόριθμος μπορεί να αποτελέσει ένα πολύτιμο ερευνητικό εργαλείο για την επιβεβαίωση της βελτιστότητας λύσεων που λαμβάνονται από εξειδικευμένους αλγόριθμους επίλυσης. Παράλληλα, με κατάλληλη παραμετροποίηση μπορεί να χρησιμοποιηθεί και για την εύρεση ευρετικών λύσεων σε προβλήματα μεγάλου μεγέθους.

Κεφάλαιο 3 Το πρόβλημα της βέλτιστης υποβολής προσφορών σε αγορές ενέργειας ημερησίου προγραμματισμού

3.1 Εισαγωγή

Στα προηγούμενα κεφάλαια δόθηκε ένας ορισμός για την έννοια του μεικτού ακέραιου γραμμικού διεπίπεδου προγραμματισμού και παρουσιάστηκε εν συντομία το πρόβλημα της υποβολής βέλτιστων προσφορών από έναν παραγωγό ενέργειας που συμμετέχει σε μια αγορά ημερήσιου προγραμματισμού ενέργειας.

Στο κεφάλαιο αυτό θα γίνει παρουσίαση του μαθηματικού μοντέλου μεικτού ακέραιου γραμμικού διεπίπεδου προγραμματισμού που αναπτύχθηκε από τους Kozanidis et al. (2011) προκειμένου να λυθεί το παραπάνω πρόβλημα.

3.2 Διατύπωση του προβλήματος

Στην παρούσα εργασία θεωρείται ένα σύνολο μονάδων παραγωγής που συμμετέχουν σε μια αγορά ημερήσιου προγραμματισμού ηλεκτρικής ενέργειας. Οι παραγωγοί υποβάλλουν προσφορές ενέργειας σε έναν ΑΔΣ (Ανεξάρτητο Διαχειριστή Συστήματος), ο οποίος εκκαθαρίζει την αγορά και καθορίζει την εκκίνηση και την ποσότητα ενέργειας κάθε μονάδας παραγωγής, εξασφαλίζοντας ότι η συνολική ζήτηση ενέργειας ικανοποιείται στο ελάχιστο συνολικό κόστος για το σύστημα, βάσει των προσφορών που υποβάλλονται. Τα τεχνικά χαρακτηριστικά κάθε μονάδας παραγωγής (τεχνικό ελάχιστο και μέγιστο) καθώς και το σταθερό κόστος εκκίνησης είναι σταθερά και γνωστά στον ΑΔΣ. Μετά τον καθορισμό της ποσότητας ενέργειας κάθε παραγωγού, υιοθετείται ένα σχήμα εκκαθάρισης-πληρωμών, το οποίο αποζημιώνει κάθε συμμετέχουσα μονάδα καταβάλλοντάς της το πλήρες κόστος εκκίνησης και μια ενιαία τιμή εκκαθάρισης της αγοράς (οριακή τιμή του συστήματος) για κάθε MWh που θα παραγάγει.

Κάθε παραγωγός αντιμετωπίζει το πρόβλημα της επιλογής της βέλτιστης τιμής προσφοράς, της προσφοράς δηλαδή που θα πρέπει να υποβάλλει έτσι ώστε μετά την εκκαθάριση της αγοράς και τον προσδιορισμό των ποσοτήτων ενέργειας όλων των συμμετεχουσών μονάδων, το κέρδος του να μεγιστοποιηθεί. Στην εργασία αυτή, υιοθετείται η πλευρά ενός μοναδικού παραγωγού χρησιμοποιώντας ένα καθοριστικό μοντέλο για αυτό το πρόβλημα. Υποθέτοντας ότι έχει πλήρη γνώση των παραμέτρων της αγοράς (των τεχνικών χαρακτηριστικών και των προσφορών/στοιχείων-κόστους των άλλων μονάδων παραγωγής, καθώς και της ζήτησης για ενέργεια), θέλουμε να βρούμε την τιμή προσφοράς για ενέργεια που θα πρέπει να υποβάλλει, έτσι ώστε μετά την εκκαθάριση της αγοράς να μεγιστοποιήσει το μεμονωμένο του κέρδος. Ο παραγωγός αυτός ονομάζεται και *μεμονωμένος παραγωγός*.

3.3 Ανάπτυξη μαθηματικού μοντέλου χρονικού ορίζοντα μίας περιόδου

Στη συνέχεια, παρατίθεται το μαθηματικό μοντέλο που αναπτύχθηκε από τους Kozanidis et.al (2011) για το συγκεκριμένο πρόβλημα μίας χρονικής περιόδου. Για τη μαθηματική μορφοποίηση, χρησιμοποιούνται οι παρακάτω μαθηματικοί συμβολισμοί:

Σύνολα:

U Μονάδες παραγωγής, με δείκτη u

Μεταβλητές απόφασης:

P_1 Τιμή προσφοράς ενέργειας του μεμονωμένου παραγωγού (1^{η} μονάδα)

Q_u Ποσότητα ενέργειας του παραγωγού u

ST_u Δυαδική μεταβλητή που παίρνει την τιμή 1 αν η μονάδα u παράγει θετική ποσότητα ενέργειας, και 0 διαφορετικά

p Σκιώδης τιμή του περιορισμού εκκαθάρισης της αγοράς που εξασφαλίζει την ικανοποίηση της ζήτησης (οριακή τιμή συστήματος)

Παράμετροι:

P_u Τιμή προσφοράς ενέργειας του παραγωγού u

$Q_u^{\max}$ Τεχνικό μέγιστο του παραγωγού u

$Q_u^{\min}$ Τεχνικό ελάχιστο του παραγωγού u

P^{max} Άνω όριο για την προσφορά ενέργειας του παραγωγού 1

c_1 Μοναδιαίο κόστος παραγωγής του μεμονωμένου παραγωγού (μονάδα 1)

SUC_u Κόστος εκκίνησης του παραγωγού u

D Ζήτηση ενέργειας

Όπως προαναφέρθηκε, ο μεμονωμένος παραγωγός αντιμετωπίζει το πρόβλημα διεπίπεδης βελτιστοποίησης που εμφανίζεται παρακάτω, το οποίο περιλαμβάνει στους περιορισμούς του το πρόβλημα βελτιστοποίησης που καλείται να λύσει ο ΑΔΣ. Στα πλαίσια του διεπίπεδου προγραμματισμού, αυτά τα δύο προβλήματα ονομάζονται *ανώτερου* και *κατώτερου επιπέδου* αντίστοιχα. Στο σημείο αυτό πρέπει να τονιστεί ότι το πρόβλημα κατώτερου επιπέδου είναι πάντα μέρος των περιορισμών του προβλήματος ανώτερου επιπέδου, συνεπώς, το πρόβλημα ανώτερου επιπέδου δεν μπορεί να αντιμετωπιστεί μεμονωμένα.

Επομένως, το υπό εξέταση πρόβλημα του μεμονωμένου παραγωγού μορφοποιείται ως εξής:

$$\underset{F_1}{Max}\ F = (p - c_1)Q_1 \tag{3.1}$$

$$\text{s.t. } c_1 \le P_1 \le P^{max} \tag{3.2}$$

$$\underset{ST_u, Q_u}{Min}\ f = \sum_{u \in U} (P_u Q_u + SUC_u ST_u) \tag{3.3}$$

$$\text{s.t. } \sum_{u \in U} Q_u = D \tag{3.4}$$

$$ST_u Q_u^{min} \le Q_u \le ST_u Q_u^{max},\ \forall u \in U \tag{3.5}$$

$$ST_u \text{ δυαδική},\ \forall u \in U \tag{3.6}$$

$$Q_u \ge 0,\ \forall u \in U \tag{3.7}$$

Η αντικειμενική συνάρτηση ανώτερου επιπέδου (3.1) μεγιστοποιεί το κέρδος του μεμονωμένου παραγωγού. Το κέρδος αυτό εξαρτάται από την τιμή εκκαθάρισης της αγοράς, p, που είναι η σκιώδης τιμή του περιορισμού κατώτερου επιπέδου εκκαθάρισης της αγοράς (3.4), ο οποίος εξασφαλίζει την ικανοποίηση της ζήτησης. Παρόλο που το πρόβλημα του κατώτερου επιπέδου (3.3)-(3.7) είναι ένα μεικτό ακέραιο πρόγραμμα και δεν έχει σκιώδεις τιμές σύμφωνα με τον παραδοσιακό ορισμό, η τιμή του p μπορεί να υπολογιστεί με τη διαδικασία που έχει προταθεί από τους O' Neill et al. (2005). Η διαδικασία αυτή περιλαμβάνει πρώτα τη λήψη της βέλτιστης λύσης του αρχικού μεικτού ακέραιου μοντέλου, και στη συνέχεια, την επίλυση του συνεχούς μοντέλου που προκύπτει αφού οι αρχικές ακέραιες μεταβλητές τεθούν ίσες με τις βέλτιστες τιμές τους σε αυτήν τη λύση. Με βάση τη θεωρία της οριακής τιμολόγησης, οι μονάδες ενέργειας αποζημιώνονται με τη σκιώδη τιμή του περιορισμού εκκαθάρισης της αγοράς που υπολογίζεται με αυτόν τον τρόπο. Το κόστος εκκίνησης του πρώτου παραγωγού ενέργειας δεν περιλαμβάνεται στην αντικειμενική συνάρτηση (3.1), δεδομένου ότι οι κανόνες της αγοράς καθορίζουν ότι κάθε παραγωγός αποζημιώνεται πλήρως για το συγκεκριμένο κόστος.

Ο περιορισμός (3.2) επιβάλλει ένα κάτω κι ένα άνω όριο στην τιμή προσφοράς του μεμονωμένου παραγωγού. Αυτά τα όρια επιβάλλονται από κανόνες της αγοράς, σύμφωνα με τους οποίους η τιμή προσφοράς ενός παραγωγού δεν μπορεί να είναι χαμηλότερη από το μοναδιαίο κόστος παραγωγής του και υψηλότερη από μία δεδομένη τιμή που είναι γνωστή και ως price-cap. Το πρόβλημα του κάτω επίπεδου ορίζεται από τη μορφοποίηση (3.3)-(3.7). Η αντικειμενική συνάρτηση (3.3) ελαχιστοποιεί το συνολικό κόστος της παρεχόμενης ενέργειας. Ο περιορισμός (3.4) εκκαθαρίζει την αγορά, εξασφαλίζοντας ότι η ζήτηση για ενέργεια θα ικανοποιηθεί. Ο περιορισμός (3.5) εγγυάται ότι δεν θα παραβιαστεί το τεχνικό ελάχιστο και το τεχνικό μέγιστο κάθε μονάδας παραγωγής. Τέλος, οι περιορισμοί (3.6) και (3.7) επιβάλλουν ακεραιότητα και μη-αρνητικότητα, αντίστοιχα, στις μεταβλητές απόφασης του κάτω επιπέδου.

Το πρόβλημα (3.1)-(3.7) είναι ένα μεικτό ακέραιο διεπίπεδο μοντέλο βελτιστοποίησης. Η ύπαρξη των αδιαιρετοτήτων που προκύπτουν από τη μοντελοποίηση της εκκίνησης των μονάδων παραγωγής με δυαδικές μεταβλητές και οι περιορισμοί για το τεχνικό ελάχιστό τους διαφοροποιούν τη μορφοποίηση αυτή από άλλες παρόμοιες που έχουν αναπτυχθεί στο παρελθόν. Στη συνέχεια, αναπτύσσεται μια μεθοδολογία επίλυσης που βασίζεται σε συνδυαστική απαρίθμηση του συνόλου των εφικτών λύσεων για την εύρεση της βέλτιστης λύσης.

Μια βασική ιδιότητα του προβλήματος (3.1)-(3.7) είναι ότι μπορεί να μην έχει βέλτιστη λύση, ακόμη κι όταν υπάρχει βέλτιστη λύση για το πρόβλημα του κατώτερου επιπέδου. Αυτό μπορεί να συμβεί όταν η βέλτιστη λύση του προβλήματος του κάτω επιπέδου δεν είναι μοναδική. Ας εξετάσουμε, για παράδειγμα, την περίπτωση κατά την οποία το κέρδος του μεμονωμένου παραγωγού μεγιστοποιείται για μια συγκεκριμένη τιμή του P_1. Ας υποτεθεί τώρα ότι, γι' αυτή τη συγκεκριμένη τιμή, η βέλτιστη λύση του κάτω προβλήματος δεν είναι μοναδική, αλλά αντί αυτού, υπάρχουν δύο βέλτιστες λύσεις τέτοιες ώστε ο μεμονωμένος παραγωγός συμμετέχει στην αγορά ($ST_1 = 1$) και μάλιστα μεγιστοποιεί το κέρδος του στην πρώτη, ενώ ο ίδιος δεν συμμετέχει στην αγορά ($ST_1 = 0$) και αντιλαμβάνεται μηδενικό κέρδος στη δεύτερη.

Η βασική θεωρία της διεπίπεδης βελτιστοποίησης (Dempe, 2002) δεν επιτρέπει τη συνεργασία οποιασδήποτε μορφής μεταξύ του άνω και του κάτω επιπέδου λήψης απόφασης. Ο παίκτης του άνω επιπέδου πάντοτε επιλέγει πρώτος τις τιμές των μεταβλητών απόφασης που θεωρεί πιο συμφέρουσες. Στη συνέχεια, με τις τιμές αυτές δεδομένες, ο παίκτης του κάτω επιπέδου έπεται και βρίσκει τις βέλτιστες τιμές των μεταβλητών απόφασης που βρίσκονται υπό τον έλεγχό του. Γίνεται φανερό από το γεγονός αυτό, ότι ο πρώτος παίκτης δεν έχει κανέναν τρόπο να εξαναγκάσει το δεύτερο στην επιλογή μιας συγκεκριμένης βέλτιστης λύσης για το κάτω επίπεδο, όταν υπάρχουν εναλλακτικές τέτοιες λύσεις. Συνεπώς, στο σύντομο υποθετικό σενάριο που περιγράφεται παραπάνω, δεν υπάρχει καμία εγγύηση ότι ο μεμονωμένος παραγωγός θα επιτύχει όντως το μέγιστο κέρδος του.

Αρκετές προσεγγίσεις έχουν προταθεί για την αντιμετώπιση αυτού του ζητήματος, οι πιο δημοφιλείς από τις οποίες καταφεύγουν σε μια μικρή τροποποίηση του ορισμού του προβλήματος και της αντίστοιχης μορφοποίησής του. Στο πλαίσιο των αγορών ηλεκτρικής ενέργειας έχουν προταθεί αρκετοί ειδικοί κανόνες, ωστόσο κανένας από αυτούς δεν έχει υιοθετηθεί καθολικά. Ένας από τους κανόνες αυτούς προτείνει να ευνοείται πάντα η μονάδα με το χαμηλότερο μεταβλητό κόστος παραγωγής. Στην παρούσα εργασία, υιοθετείται η λεγόμενη αισιόδοξη προσέγγιση (Dempe, 2002), σύμφωνα με την οποία κάθε φορά που υφίστανται πολλαπλές βέλτιστες λύσεις για το πρόβλημα του κάτω επιπέδου, επιλέγεται αυτή που είναι πιο συμφέρουσα για το άνω επίπεδο. Η προσέγγιση αυτή προϋποθέτει ότι ο άνω παίκτης έχει πάντα κάποιο τρόπο να εξαναγκάσει τον κάτω παίκτη να επιλέξει μια συγκεκριμένη βέλτιστη λύση για το κάτω πρόβλημα. Έτσι, η θεωρία που αναπτύσσεται στο υπόλοιπο της παρούσας εργασίας αναφέρεται στην αισιόδοξη προσέγγιση του προβλήματος, όπου εξασφαλίζεται η ύπαρξη βέλτιστης λύσης λαμβάνοντας υπ’ όψη κάποιους λογικούς κανόνες, και δεν ισχύει απαραίτητα σε διαφορετικές προσεγγίσεις. Ένας από αυτούς τους κανόνες για να επιλεγεί μία από τις διάφορες βέλτιστες λύσεις του κατώτερου επιπέδου είναι ότι η αγορά ενέργειας προτιμάει την λύση του παραγωγού με το μικρότερο μεταβλητό κόστος.

3.4 Ανάπτυξη μαθηματικού μοντέλου χρονικού ορίζοντα πολλαπλών περιόδων

Το μοντέλο το οποίο μελετήθηκε στην προηγούμενη υποενότητα της εργασίας μπορεί να επεκταθεί ώστε να συμπεριλάβει πρόσθετες πτυχές του υπό εξέταση προβλήματος. Μια πιο ρεαλιστική επέκταση του προβλήματος προκύπτει όταν εξετάζεται ένας χρονικός ορίζοντας H περιόδων (συνήθως $H = 24$ και ο αντίστοιχος δείκτης h συμβολίζει τις ώρες) και ο μεμονωμένος παραγωγός υποβάλλει H τιμές-προσφοράς, έτσι ώστε να μεγιστοποιηθεί το συνολικό αθροιστικό του κέρδος μετά την εκκαθάριση της αγοράς για το σύνολο του χρονικού ορίζοντα. Μια σημαντική διαφοροποίηση σε αυτήν την περίπτωση είναι ότι εκκίνηση μιας μονάδας θεωρείται ότι λαμβάνει χώρα μόνο όταν αυτή η μονάδα παράγει θετική ποσότητα ενέργειας μετά από μία ή περισσότερες περιόδους κατά τις οποίες βρισκόταν εκτός λειτουργίας. Έτσι, το κόστος εκκίνησης της μονάδας συμπεριλαμβάνεται στην αντικειμενική συνάρτηση μόνο μια φορά για κάθε ομάδα διαδοχικών χρονικών περιόδων κατά τις οποίες αυτή η μονάδα συμμετέχει στην αγορά. Επαυξάνοντας το συμβολισμό της υποενότητας 3.3 με έναν επιπλέον δείκτη που αντιπροσωπεύει τη χρονική περίοδο, το πρόβλημα μορφοποιείται ως ακολούθως σε αυτήν την περίπτωση:

Σύνολα:

U Μονάδες παραγωγής, με δείκτη u

H Χρονικές περίοδοι, με δείκτη h

Μεταβλητές απόφασης:

$P_{1,h}$ Τιμή προσφοράς ενέργειας του μεμονωμένου παραγωγού (1η μονάδα) για την χρονική περίοδο h

$Q_{u,h}$ Ποσότητα ενέργειας του παραγωγού u την χρονική περίοδο h

$ST_{u,h}$ Δυαδική μεταβλητή που παίρνει την τιμή 1 αν η μονάδα u παράγει θετική ποσότητα ενέργειας την χρονική περίοδο h, και 0 διαφορετικά

$Y_{u,h}$ Δυαδική μεταβλητή που παίρνει την τιμή 1 αν υπήρχε μεταβολή κατάστασης (από παραγωγή θετικής ποσότητας σε 0 και αντίστροφα) για την μονάδα u από την χρονική περίοδο $h-1$ στην h, και 0 διαφορετικά

p_h Σκιώδης τιμή του περιορισμού εκκαθάρισης της αγοράς που εξασφαλίζει την ικανοποίηση της ζήτησης (οριακή τιμή συστήματος) την χρονική περίοδο h

Παράμετροι:

$P_{u,h}$ Τιμή προσφοράς ενέργειας του παραγωγού u την χρονική περίοδο h

$Q_u^{\max}$ Τεχνικό μέγιστο του παραγωγού u

$Q_u^{\min}$ Τεχνικό ελάχιστο του παραγωγού u

$P^{\max}$ Άνω όριο για την προσφοράς ενέργειας του παραγωγού 1

c_1 Μοναδιαίο κόστος παραγωγής του μεμονωμένου παραγωγού (μονάδα 1)

SUC_u Κόστος εκκίνησης του παραγωγού u

D_h Ζήτηση ενέργειας την χρονική περίοδο h

Πρόβλημα πολλαπλών περιόδων:

$$\underset{P_{1,h}}{Max} F = \sum_{h \in H} (p_h - c_1) Q_{1,h} \tag{3.8}$$

$$\text{s.t.} \quad c_1 \le P_{1,h} \le P^{\max}, \ \forall h \in H \tag{3.9}$$

$$\underset{ST_{u,h}, Y_{u,h}, Q_{u,h}}{Min} f = \sum_{h \in H} \sum_{u \in U} (P_{u,h} Q_{u,h} + SUC_u Y_{u,h}) \tag{3.10}$$

$$\text{s.t.} \ \sum_{u \in U} Q_{u,h} = D_h, \ \forall h \in H \tag{3.11}$$

$$ST_{u,h} Q_u^{\min} \le Q_{u,h} \le ST_{u,h} Q_u^{\max}, \ \forall u \in U, \ \forall h \in H \tag{3.12}$$

$$Y_{u,h} \ge ST_{u,h} - ST_{u,h-1}, \ \forall u \in U, \ \forall h \in H \tag{3.13}$$

$$ST_{u,h}, Y_{u,h} \ \text{δυαδική}, \ \forall u \in U, \ \forall h \in H \tag{3.14}$$

$$Q_{u,h} \ge 0, \ \forall u \in U, \ \forall h \in H \tag{3.15}$$

Σημειώνεται ότι στη μορφοποίηση αυτή, το συνολικό κόστος του συστήματος περιλαμβάνει τις δυαδικές μεταβλητές απόφασης $Y_{u,h}$ αντί των μεταβλητών $ST_{u,h}$. Από τον περιορισμό (3.13), η μεταβλητή $Y_{u,h}$ λαμβάνει την τιμή 1 εάν $ST_{u,h-1} = 0$ και $ST_{u,h-1} = 0$. Σε κάθε άλλη περίπτωση, η τιμή αυτής της μεταβλητής ορίζεται ίση με 0, επειδή εμφανίζεται στην αντικειμενική συνάρτηση του προβλήματος του κάτω επιπέδου με θετικό συντελεστή. Έτσι, η μεταβλητή $Y_{u,h}$ χρησιμοποιείται για να υποδηλώσει μια αλλαγή στην κατάσταση της μονάδας u από εκτός λειτουργίας στην περίοδο $h-1$ σε εντός λειτουργίας στην περίοδο h. Σαν αποτέλεσμα, η παραπάνω μορφοποίηση απαιτεί γνώση της κατάστασης κάθε μονάδας παραγωγής κατά την τελευταία περίοδο του προηγούμενου χρονικού ορίζοντα ($ST_{u,0}$).

Τα υπόλοιπα μέρη της παραπάνω μορφοποίησης αποτελούν μια απλή επέκταση της μορφοποίησης της Ενότητας 3.3 σε ένα χρονικό ορίζοντα πολλών περιόδων. Επιπλέον, η μορφοποίηση αυτή μπορεί να επεκταθεί περαιτέρω για να συμπεριλάβει επιπρόσθετες πτυχές του πραγματικού προβλήματος, όπως οι *ελάχιστοι χρόνοι μη λειτουργίας* (minimum downtimes) και οι *ελάχιστοι χρόνοι λειτουργίας* (minimum uptimes) των μονάδων παραγωγής. Ανάλογα με το συγκεκριμένο τύπο της (καύσης λιγνίτη, φυσικού αερίου, πετρελαίου, κ.τ.λ.), μια μονάδα μπορεί να μην είναι σε θέση να αλλάξει την κατάστασή της κατά βούληση σε δύο διαδοχικές χρονικές περιόδους. Σε γενικές γραμμές, μετά την εκκίνηση (σβήσιμο) μιας μονάδας που βρισκόταν προηγουμένως εκτός (εντός) λειτουργίας, η μονάδα πρέπει να παραμείνει εντός (εκτός) λειτουργίας για έναν ελάχιστο χρόνο (μη) λειτουργίας πριν τεθεί εκτός (εντός) λειτουργίας ξανά. Μεταξύ άλλων, οι Andrianesis et al. (2012) παρουσιάζουν μια κατάλληλη μορφοποίηση, η οποία ενσωματώνει αυτούς τους περιορισμούς. Ο αλγόριθμος που δημιουργήθηκε στην παρούσα διπλωματική εργασία βρίσκει τη βέλτιστη λύση του προβλήματος των πολλαπλών περιόδων.

3.5 Εναλλακτικοί μέθοδοι αποζημίωσης παραγωγού

Υπάρχουν δύο εναλλακτικές μέθοδοι με τις οποίες μπορεί να αποζημιωθεί ο μεμονωμένος παραγωγός λόγω της συμμετοχής του στην αγορά ενέργειας. Σύμφωνα και με τις δύο μεθόδους, ο παραγωγός αποζημιώνεται πλήρως για το κόστος εκκίνησης της μονάδας παραγωγής του. Στην πρώτη περίπτωση, η τιμή αποζημίωσης είναι ίση με τη σκιώδη τιμή του περιορισμού εκκαθάρισης της αγοράς που εξασφαλίζει την ικανοποίηση της ζήτησης (p_h), ενώ στη δεύτερη είναι ίση με την τιμή προσφοράς του παραγωγού. Στην παρούσα εργασία λαμβάνονται και οι δύο τρόποι υπ’ όψιν καθώς επίσης δίνεται η δυνατότητα στον χρήστη του αλγορίθμου να επιλέξει μία από τις δύο παραπάνω μεθόδους αποζημίωσης στην έναρξη του προγράμματος.

3.5.1 Αποζημίωση ανάλογη της Υποβληθείσας Προσφοράς

Σε αυτόν τον τρόπο αποζημίωσης ο παραγωγός αποζημιώνεται ανάλογα με την τιμή της προσφορά που υποβάλει. Για να γίνει κατανοητός αυτός ο τρόπος αποζημίωσης, θεωρούμε τις υποβολές προσφορών ενός παραγωγού για h χρονικές περιόδους. Στη συνέχεια, η σκιώδης τιμή p_h του περιορισμού εκκαθάρισης της αγοράς που εξασφαλίζει την ικανοποίηση της ζήτησης την χρονική περίοδο h θέτεται ίση με την τιμή της υποβάλλουσας προσφοράς του μεμονωμένου παραγωγού εκείνη την χρονική περίοδο h, δηλαδή $P_{1,h} = p_h$. Έτσι, η αντικειμενική συνάρτηση σύμφωνα με την οποία αποζημιώνεται ο παραγωγός μετατρέπεται ουσιαστικά στη συνάρτηση $\sum_{h\in H}\left(P_{1,h}-c_1\right)Q_{1,h}$.

Δηλαδή, το συνολικό ποσό σύμφωνα με το οποίο αποζημιώνεται ο παραγωγός είναι ίσο με το άθροισμα των προσφορών για όλες τις περιόδους που θα υποβάλει στην αγορά ενέργειας μείον το μοναδιαίο κόστος παραγωγής του. Η παραπάνω μέθοδος αποζημίωσης ονομάζεται *Αποζημίωση ανάλογη της Υποβληθείσας Προσφοράς (ΑΑΠ)*.

3.5.2 Αποζημίωση ανάλογη της Οριακής Τιμής του Συστήματος

Η μέθοδος αυτή εφαρμόζεται στην Ελληνική αγορά ενέργειας. Η αποζημίωση σε αυτήν την περίπτωση γίνεται ανάλογα με την *Οριακή Τιμή του Συστήματος (ΟΤΣ).* Η Οριακή Τιμή του Συστήματος είναι η τιμή στην οποία εκκαθαρίζεται η αγορά ηλεκτρικής ενέργειας και είναι η τιμή που εισπράττουν όλοι όσοι εγχέουν ενέργεια στο Σύστημα και πληρώνουν όλοι όσοι ζητούν ενέργεια από το Σύστημα. Συγκεκριμένα, η Οριακή Τιμή του Συστήματος διαμορφώνεται από τον συνδυασμό των προσφορών τιμών και ποσοτήτων που υποβάλλουν κάθε μέρα οι διαθέσιμες μονάδες παραγωγής ηλεκτρικής ενέργειας, και του ωριαίου φορτίου ζήτησης ηλεκτρικής ενέργειας, που διαμορφώνεται σε καθημερινή βάση από τους καταναλωτές. Επιχειρώντας μια απλή περιγραφή του τρόπου υπολογισμού της Οριακής Τιμής του Συστήματος, μπορεί να αναφερθεί ότι οι μονάδες παραγωγής κατατάσσονται αναλόγως των προσφορών τους σε αύξουσα σειρά, ξεκινώντας από την χαμηλότερη προσφερόμενη τιμή για ορισμένη ποσότητα ενέργειας και καταλήγοντας στην υψηλότερη προσφερόμενη τιμή. Η οριακή τιμή του συστήματος είναι ξεχωριστή για κάθε περίοδο και ορίζεται ως η προσφορά του παραγωγού ο οποίος θα παραγάγει ενέργεια μεγαλύτερη από το τεχνικό του ελάχιστο $Q_i^{\min}$ και μικρότερη από το τεχνικό του μέγιστο $Q_i^{\max}$. Για την κατανόηση του παραπάνω ορισμού ακολουθεί το ακόλουθο αριθμητικό παράδειγμα:

Έστω ότι έχουμε τέσσερεις παραγωγούς που συμμετέχουν στην αγορά ενέργειας. Για την πρώτη περίοδο ($h=1$) έστω ότι το τεχνικό ελάχιστο του πρώτου παραγωγού ($u=1$) είναι $Q_1^{\min}=240$ και $Q_1^{\max}=400$. Αντίστοιχα, και για τον δεύτερο παραγωγό ($u=2$) έχουμε $Q_2^{\min}=200$ και $Q_2^{\max}=500$, για τον τρίτο ($u=3$) έχουμε $Q_3^{\min}=100$ και $Q_3^{\max}=300$ και για τον τέταρτο ($u=4$) $Q_4^{\min}=150$ και $Q_4^{\max}=350$. Η προσφορά που έχει υποβάλει ο καθένας την πρώτη περίοδο είναι $P_1^1=50$ για τον πρώτο, $P_2^1=58$ για τον δεύτερο, $P_3^1=54$ και $P_4^1=62$ για τον τρίτο και τέταρτο αντίστοιχα. Η ζήτηση για την περίοδο αυτή είναι $D_1=1150$. Επιλύοντας το πρόβλημα ελαχιστοποίησης του Α.Δ.Σ. προκύπτουν οι εξής λύσεις: $q_1^1=400$, $q_2^1=300$, $q_3^1=100$ και $q_4^1=350$. Παρατηρούμε ότι η ποσότητα που παράγει η δεύτερη μονάδα παραγωγής είναι ανάμεσα στο τεχνικό μέγιστο και τεχνικό ελάχιστο ($Q_1^{\min}=200<q_2^1=300<Q_1^{\max}=500$) οπότε και η προσφορά του ορίζεται ως η Οριακή Τιμή του Συστήματος εκείνης της περιόδου ($p_1=P_2^1$). Η παραπάνω λύση απεικονίζεται στον Πίνακα 3.1.

Μονάδα (u), Περίοδος (h)	$Q_u^{\min}$	$Q_u^{\max}$	P_u^h	q_u^h	$p_h\,(OT\Sigma)$
(1 , 1)	240	400	50	400	52
(2 , 1)	200	500	58	300	
(3 , 1)	100	300	54	100	
(4 , 1)	150	350	62	350	

Πίνακας 3.1 Τεχνικά χαρακτηριστικά των μονάδων παραγωγής, τιμές προσφοράς και βέλτιστη λύση του προβλήματος ΑΔΣ.

Στην ουσία, η Οριακή Τιμή του Συστήματος συμπίπτει με την προσφορά του τελευταίου παραγωγού που πρέπει να λειτουργήσει για να καλυφθεί η ζήτηση αν αυτή αυξανόταν κατά μία μονάδα.

Υπάρχουν όμως και περιπτώσεις όπου καμία από τις μονάδες παραγωγής δεν παράγει ποσότητα ενέργειας ανάμεσα στο τεχνικό της ελάχιστο και τεχνικό της μέγιστο. Για την κάλυψη αυτών των περιπτώσεων ορίζονται εμπειρικά οι παρακάτω κανόνες:

- <u>*Περίπτωση Α*</u>: Στην περίπτωση όπου τουλάχιστον μια μονάδα παράγει σε μία περίοδο ποσότητα ίση με το τεχνικό της ελάχιστο, τότε ως *ΟΤΣ* για εκείνη την περίοδο ορίζεται η προσφορά του παραγωγού με την μικρότερη τιμή μεταξύ όλων των παραγωγών που παράγουν στο τεχνικό τους ελάχιστο. Ένα παράδειγμα της *Περίπτωσης Α* απεικονίζεται στον Πίνακα 3.2 όπου η ζήτηση της περιόδου είναι $D_1 = 590$.

Μονάδα (u), Περίοδος (h)	$Q_u^{\min}$	$Q_u^{\max}$	P_u^h	q_u^h	$p_h\,(OT\Sigma)$
(1 , 1)	240	400	50	240	50
(2 , 1)	200	500	58	200	
(3 , 1)	100	300	54	100	
(4 , 1)	150	350	62	150	

Πίνακας 3.2 Τεχνικά χαρακτηριστικά των μονάδων παραγωγής, τιμές προσφοράς και βέλτιστη λύση του προβλήματος ΑΔΣ για την Περίπτωση Α.

- *Περίπτωση Β*: Αντίθετα, στην περίπτωση όπου όλες οι μονάδες που συμμετέχουν παράγουν σε μία περίοδο ποσότητα ίση με το τεχνικό τους μέγιστο τότε ως *ΟΤΣ* για εκείνη την περίοδο ορίζεται η προσφορά του παραγωγού με την μεγαλύτερη τιμή μεταξύ όλων εκείνων που συμμετέχουν. Ένα παράδειγμα της *Περίπτωσης Β* απεικονίζεται στον Πίνακα 3.3 όπου η ζήτηση της περιόδου είναι $D_1 = 1550$.

Μονάδα (u), Περίοδος (h)	$Q_u^{\min}$	$Q_u^{\max}$	P_u^h	q_u^h	$p_h\,(OT\Sigma)$
(1 , 1)	240	400	50	400	62
(2 , 1)	200	500	58	500	
(3 , 1)	100	300	54	300	
(4 , 1)	150	350	62	350	

Πίνακας 3.3 Τεχνικά χαρακτηριστικά των μονάδων παραγωγής, τιμές προσφοράς και βέλτιστη λύση του προβλήματος ΑΔΣ για την Περίπτωση Β.

Κεφάλαιο 4 Αλγόριθμος επίλυσης του μοντέλου πολλαπλών περιόδων

4.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται ο αλγόριθμος που αναπτύχθηκε για την επίλυση του μαθηματικού μοντέλου διεπίπεδης βελτιστοποίησης στην πιο σύνθετη περίπτωση των πολλαπλών περιόδων. Αναπτύχθηκε σε γλώσσα προγραμματισμού C και το λογισμικό που χρησιμοποιήθηκε ήταν το *Microsoft Visual Studio 2012*. Για την επίλυσή του χρησιμοποιήθηκε ο solver *IBM CPLEX Optimization Studio v.12.4*.

4.2 Ανάλυση αλγορίθμου

Ο αλγόριθμος που αναπτύσσεται για την επίλυση του υπό εξέταση προβλήματος συνοψίζεται στα τέσσερα παρακάτω βήματα:

<u>Βήμα 1^{ο}:</u>

Εισάγονται τα δεδομένα του προβλήματος μέσα από ένα αρχείο κειμένου (.txt) τα οποία εμφανίζονται παρακάτω:

H Σύνολο χρονικών περιόδων

U Σύνολο μονάδων παραγωγής

$P_{u,h}$ Τιμή προσφοράς ενέργειας του παραγωγού u την χρονική περίοδο h

$Q_u^{\max}$ Τεχνικό μέγιστο του παραγωγού u

$Q_u^{\min}$ Τεχνικό ελάχιστο του παραγωγού u

$P^{\max}$ Άνω όριο για την προσφοράς ενέργειας του παραγωγού 1

$P^{\min}$ Κάτω όριο για την προσφοράς ενέργειας του παραγωγού 1

c_1 Μοναδιαίο κόστος παραγωγής του μεμονωμένου παραγωγού (μονάδα 1)

SUC_u Κόστος εκκίνησης του παραγωγού u

D_h Ζήτηση ενέργειας την χρονική περίοδο h

Στη συνέχεια επιλέγεται από τον χρήστη η μέθοδος αποζημίωσης του μεμονωμένου παραγωγού (αποζημίωση ανάλογη της *Υποβληθείσας Προσφοράς* ή αποζημίωση ανάλογης της *Οριακής Τιμής του Συστήματος*) καθώς επίσης και το *βήμα αλλαγής* με το οποίο θα εναλλάσσεται η υποβολή της πιθανής προσφορά του.

<u>Βήμα 2ο:</u>

Αφού επιλεγεί από τον χρήστη η μέθοδος αποζημίωσης του μεμονωμένου παραγωγού επιλέγεται και η μέθοδος με την οποία θα εφαρμοστεί ο αλγόριθμος για το παραπάνω πρόβλημα διεπίπεδου γραμμικού προγραμματισμού. Οι εναλλακτικές μέθοδοι εφαρμογής του αλγορίθμου παρουσιάζονται και αναλύονται εκτενέστερα στο πέμπτο κεφάλαιο της παρούσας διπλωματικής εργασίας. Ο σκοπός της δημιουργίας και εφαρμογής αυτών των μεθόδων είναι τόσο η επιβεβαίωση της ακρίβειας των αποτελεσμάτων που θα προκύψουν από την εκτέλεση των πειραμάτων όσο και η πιθανή βελτίωσή τους. Βέβαια, για την εφαρμογή των εναλλακτικών αυτών μεθόδων στο πρόβλημα απαιτούνται τροποποιήσεις του αλγορίθμου ως προς κάποιες παραμέτρους. Στο σημείο αυτό κρίνεται σκόπιμο να αναφερθεί ότι οι εναλλακτικές μέθοδοι εκτέλεσης του αλγορίθμου είναι ευρετικού χαρακτήρα και δεν εξασφαλίζουν ότι η λύση που θα προκύψει από την εφαρμογή τους είναι και η ολικά βέλτιστη, μπορεί όμως να βελτιώσουν την ήδη υπάρχουσα λύση.

<u>Βήμα 3ο:</u>

Στη συνέχεια, επιλύεται το πρόβλημα του κάτω επιπέδου για τον πρώτο συνδυασμό τιμών των προσφορών $P_{1,h}$, οι οποίες ορίζονται στο μοναδιαίο κόστος του μεμονωμένου παραγωγού $P_1^{\min}$, και καθορίζεται ως τοπικό μέγιστο η λύση του προβλήματος άνω επιπέδου για τον συγκεκριμένο συνδυασμό. Η διαδικασία αυτή συνεχίζεται με τον ίδιο τρόπο, μέχρι να εντοπιστεί το σύνολο των διαφορετικών (όσον αφορά τις τιμές των μεταβλητών απόφασης) βέλτιστων λύσεων του προβλήματος του κάτω επιπέδου για όλους τους πιθανούς συνδυασμούς τιμών που μπορούν να προκύψουν από τις προσφορές $P_{1,h}$ του μεμονωμένου παραγωγού για κάθε χρονική περίοδο h σε συνδυασμό με το *βήμα αλλαγής*. Για να γίνει κατανοητό το παραπάνω ακολουθεί ένα αριθμητικό παράδειγμα:

Έστω ότι ο μεμονωμένος παραγωγός πρέπει να δώσει προσφορές για τέσσερεις περιόδους ($H = 4$). Το μεταβλητό κόστος (κάτω όριο τιμής προσφοράς) του μεμονωμένου παραγωγού ($u=1$) είναι $c_1 = 40$ €/MWh και το άνω όριο στην τιμή προσφοράς του είναι $P_1^{\max} = 80$ €/MWh. Έστω ότι επιλέγεται *βήμα αλλαγής των προσφορών* από τον χρήστη το $vima = 10$ €/MWh. Ο αριθμός των πιθανών συνδυασμών που μπορούν να προκύψουν υπολογίζεται από την παρακάτω σχέση:

$$\Sigma\upsilon\nu o\lambda\iota\kappa\acute{o}\varsigma\ \alpha\rho\iota\theta\mu\acute{o}\varsigma\ \pi\iota\theta\alpha\nu\acute{\omega}\nu\ \sigma\upsilon\nu\delta\upsilon\alpha\sigma\mu\acute{\omega}\nu = \left[div\left[(P_{\max} - c_1)/vima\right] + 1\right]^H \quad (4.1)$$

Οι πιθανοί συνδυασμοί τιμών που μπορούν να προκύψουν για τις προσφορές $P_{1,h}$ του μεμονωμένου παραγωγού για κάθε χρονική περίοδο h απεικονίζονται στον Πίνακα 4.1.

Περίοδος (h)	$h=1$	$h=2$	$h=3$	$h=4$
P_1^h	40	40	40	40
P_1^h	40	40	40	50
P_1^h	40	40	40	60
P_1^h	40	40	40	70
P_1^h	40	40	40	80
P_1^h	40	40	50	40
P_1^h	40	40	50	50
P_1^h	40	40	50	60
P_1^h	40	40	50	70
P_1^h	40	40	50	80
P_1^h	40	40	60	40
P_1^h	…	…	…	…
P_1^h	80	80	80	80

Πίνακας 4.1 Πιθανοί συνδυασμοί τιμών που μπορούν να προκύψουν για τις προσφορές του μεμονωμένου παραγωγού

Βήμα 4^{ο}:

Τέλος, επιλέγεται αυτός ο συνδυασμός των τιμών $P_{1,h}$ για κάθε χρονική περίοδο $h \in H$ ο οποίος θα μεγιστοποιεί το κέρδος του μεμονωμένου παραγωγού (πρόβλημα ανώτερου επίπεδου). Στην περίπτωση όπου υπάρχουν πολλαπλές βέλτιστες λύσεις επιλέγεται αυτή που θα προκύψει πρώτη, η οποία συνήθως αποτελείται από το συνδυασμό των προσφορών με τις μικρότερη τιμή σε σχέση με τις υπόλοιπες. Αυτή είναι και η ολική βέλτιστη λύση του διεπίπεδου προβλήματος που μελετάται στην παρούσα διπλωματική εργασία.

4.3 Συνοπτική απεικόνιση αλγορίθμου

Μια συνοπτική απεικόνιση του αλγορίθμου όπου εμφανίζονται περιεκτικά τα παραπάνω βήματα εμφανίζεται στον Πίνακα 4.2.

Βήμα 1. Εισαγωγή δεδομένων. Εισάγονται τα δεδομένα μέσω αρχείου (.txt) και επιλέγεται η μέθοδος αποζημίωσης του μεμονωμένου παραγωγού καθώς επίσης και το *βήμα αλλαγής* των προσφορών του.

Βήμα 2. Επιλογή μεθόδου επίλυσης προβλήματος κατώτερου επιπέδου. Επιλέγεται μία από τις πέντε διαθέσιμες μεθόδους για την επίλυση του προβλήματος κατώτερου επιπέδου.

Βήμα 3. Επίλυση προβλήματος κατώτερου επιπέδου. Υπολογίζεται ο συνολικός αριθμός όλων των πιθανών συνδυασμών των προσφορών του μεμονωμένου παραγωγού για κάθε περίοδο σύμφωνα με τη σχέση $\left[div\left[(P_{\max} - c_1) / vima \right] + 1 \right]^H$. Στη συνέχεια επιλύεται το πρόβλημα του κατώτερου επιπέδου για κάθε πιθανό συνδυασμό των προσφορών του μεμονωμένου παραγωγού.

Βήμα 4. Επίλυση προβλήματος ανώτερου επιπέδου. Επιλέγεται αυτός ο συνδυασμός των τιμών $P_{1,h}$ για κάθε χρονική περίοδο $h \in H$ ο οποίος θα επιφέρει το μεγαλύτερο κέρδος στον μεμονωμένο παραγωγό σύμφωνα με τη σχέση $\underset{P_{1,h}}{Max} F = \sum_{h \in H} (p_h - c_1) Q_{1,h}$.

Πίνακας 4.2 Συνοπτική απεικόνιση αλγορίθμου

Κεφάλαιο 5 Εφαρμογή Αλγορίθμου - Υπολογιστικά πειράματα

5.1 Εισαγωγή

Στο κεφάλαιο αυτό πραγματοποιείται η εφαρμογή του αλγορίθμου που παρουσιάστηκε στο προηγούμενο κεφάλαιο πάνω σε δύο ρεαλιστικά παραδείγματα. Αναλύεται η απόδοσή του και ο χρόνος εκτέλεσης του ανάλογα με το μέγεθος και τη δυσκολία του προβλήματος. Τέλος, υλοποιείται η ανάλυση ευαισθησίας του προβλήματος και προτείνονται κάποιοι τρόποι βελτίωσης της απόδοσής του.

5.2 Ανάπτυξη κώδικα

Το παραπάνω πρόβλημα καθώς και η μεθοδολογία επίλυσης προγραμματίστηκε με τη χρήση της γλώσσας προγραμματισμού *C#* και το λογισμικό που χρησιμοποιήθηκε ήταν το *Microsoft Visual Studio 2012* . Για την επίλυση του χρησιμοποιήθηκε ο solver *IBM CPLEX Optimization Studio v.12.4.*

Ο υπολογιστής που χρησιμοποιήθηκε για την εκτέλεση των πειραμάτων έχει τα εξής χαρακτηριστικά:

- *Επεξεργαστής:* Intel Core i3-3120M CPU @ 2.50Ghz
- *Εγκατεστημένη μνήμη (RAM):* 4,00 GB
- *Λογισμικό:* Windows 8.1 Professional 64-bit

Σε κάθε πείραμα που εκτελέστηκε τα δεδομένα εισήχθηκαν στο πρόγραμμα με την μορφή πινάκων από αρχείο κειμένου τύπου .txt.

5.3 Εφαρμογή Αλγορίθμου

5.3.1 Παραδοχές

Οι παραδοχές που λαμβάνονται υπ' όψιν αφορούν στην εφαρμογή του αλγορίθμου και δεν επηρεάζουν την γενικότερη εφαρμογή της μεθοδολογίας επίλυσης. Έχουν σχεδιαστεί έτσι ώστε οι υπολογιστικοί χρόνοι εκτέλεσης του αλγορίθμου να είναι εφικτοί και συγχρόνως να συμβάλουν στην ρεαλιστικότητα του συγκεκριμένου παραδείγματος όπου αφορά στην ελληνική αγορά ενέργειας. Οι παραδοχές αυτές είναι οι ακόλουθες:

1η Παραδοχή (Αρχική κατάσταση των παραγωγικών μονάδων): Αρχικά, δηλαδή πριν την πρώτη χρονική περίοδο, όλες οι μονάδες παραγωγής είναι σβηστές. Η παραδοχή αυτή έχει να κάνει με το γεγονός ότι για να παραχθεί μια μονάδα την πρώτη χρονική περίοδο, ο παραγωγός πρέπει να πληρώσει το κόστος εκκίνησης SUC_u.

2η Παραδοχή (Στρατηγική υποβολής προσφορών): Η στρατηγική υποβολής προσφοράς του μεμονωμένου παραγωγού έχει ως αρχή την εξαντλητική δοκιμή όλων των πιθανών συνδυασμών (“brute-force”) των προσφορών για κάθε χρονική περίοδο με τον εξής τρόπο:

- Υπολογίζεται το κέρδος για όλους του πιθανούς συνδυασμούς μεταξύ του μεταβλητού κόστους και του άνω ορίου με *βήμα αλλαγής* που καθορίζεται από τον χρήστη του αλγορίθμου και επιλέγεται αυτός ο συνδυασμός που επιφέρει το υψηλότερο κέρδος στον μεμονωμένο παραγωγό. Επίσης, το *βήμα αλλαγής* ορίζεται ως ακέραιος αριθμός.
- Μεταξύ όλων των συνδυασμών που επιφέρουν το υψηλότερο κέρδος επιλέγεται ο συνδυασμός που βρέθηκε πρώτος .

Η τιμή της προσφοράς πρέπει να είναι μεταξύ του μοναδιαίου κόστους παραγωγής c_1 και του άνω ορίου $P^{\max}$.

5.4 Πρώτο παράδειγμα με τη μέθοδο αποζημίωσης βάσει της Οριακής Τιμής του Συστήματος

Στο πρώτο παράδειγμα υπάρχουν τέσσερις (4) μονάδες παραγωγής ενέργειας και τέσσερις (4) χρονικοί περίοδοι. Τα τεχνικά χαρακτηριστικά, οι τιμές-προσφοράς και τα κόστη εκκίνησης των μονάδων παραγωγής παρουσιάζονται στον Πίνακα 5.1. Τα τεχνικά ελάχιστα και μέγιστα δίνονται σε MW, τα κόστη εκκίνησης σε € και οι τιμές-προσφοράς για την ενέργεια σε €/MWh. Το μεταβλητό κόστος του μεμονωμένου παραγωγού (της μονάδας 1) είναι 50 €/MWh, το άνω όριο στην τιμή προσφοράς του είναι 100 €/MWh και η ζήτηση για ενέργεια είναι ίση με 1,000 MWh για κάθε χρονική περίοδο. Τα δεδομένα των μονάδων παραγωγής δεν είναι πλασματικά, αλλά αντιστοιχούν σε πραγματικές μονάδες που συμμετέχουν στην ελληνική αγορά ηλεκτρικής ενέργειας, όπως περιγράφεται από τους Andrianesis et al. (2012).

Μονάδα (u)	$Q_u^{\min}$	$Q_u^{\max}$	SUC_u
1	240	377	13000
2	144	476	10000
3	240	384	15000
4	105	188	27000

Πίνακας 5.1 Δεδομένα Πρώτου Παραδείγματος

Στον Πίνακα 5.2 απεικονίζονται οι προσφορές που υποβάλει ο κάθε παραγωγός για κάθε χρονική περίοδο h. Οι τιμές είναι σε €/MWh.

	Περίοδος ((h)			
Μονάδα (u)	**1**	**2**	**3**	**4**
1	---	---	---	---
2	58	55	68	60
3	54	50	65	69
4	62	60	58	56

Πίνακας 5.2 Προσφορές μονάδων παραγωγής για κάθε χρονική περίοδο

Στον Πίνακα 5.3 απεικονίζεται η ζήτηση για ενέργεια για κάθε χρονική περίοδο σε MWh.

	Περίοδος ((h)			
	1	**2**	**3**	**4**
Ζήτηση (D_h)	1000	1000	1000	1000

Πίνακας 5.3 Ζήτηση για ενέργεια σε κάθε χρονική περίοδο

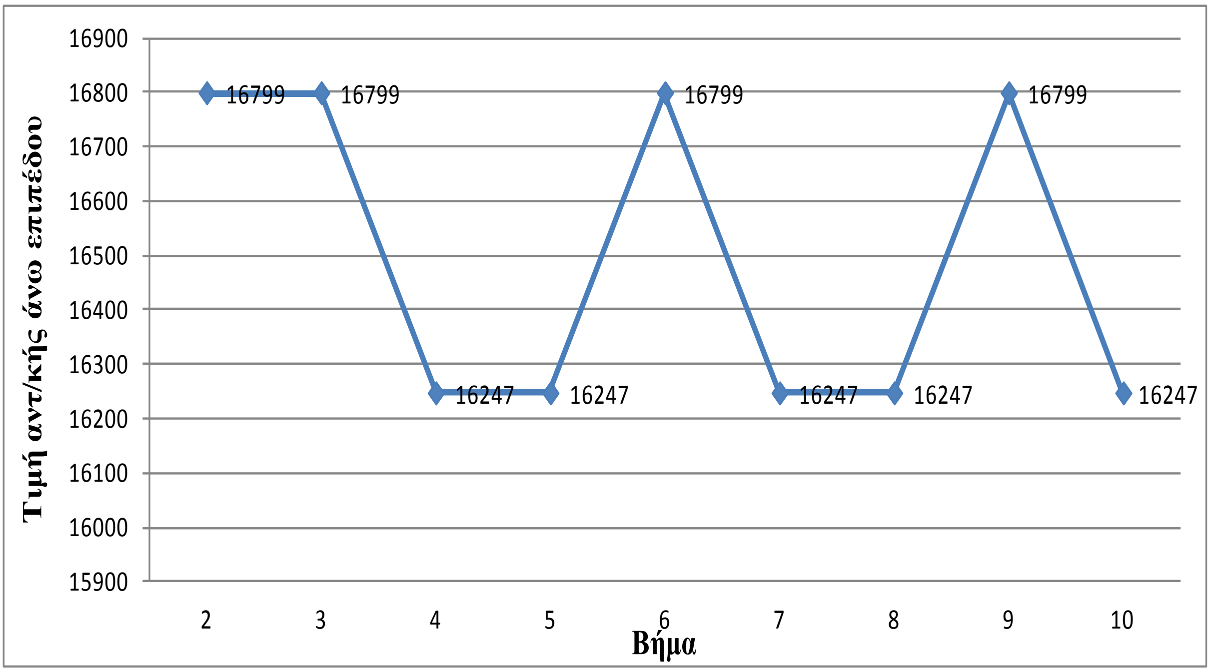
5.4.1 Αριθμητικά Αποτελέσματα πειραμάτων

Τα πειράματα που υλοποιηθήκαν για το πρώτο παράδειγμα αφορούν στην εφαρμογή διαφορετικού *βήματος αλλαγής* και κατά ποιον τρόπο επηρεάζεται η τιμή της αντικειμενικής συνάρτησης καθώς και ο χρόνος υλοποίησης του αλγορίθμου από αυτήν την αλλαγή. Το βήμα που χρησιμοποιήθηκε πήρε τις τιμές 2, 3, 4, 5, 6, 7, 8, 9,10. Ο λόγος που δεν λαμβάνεται υπ’ όψιν το βήμα 1 είναι ότι, σύμφωνα με την Σχέση 4.1, οι πιθανοί συνδυασμοί των προσφορών που προκύπτουν για αυτό το βήμα είναι 6,765,201 συνδυασμοί, γεγονός το οποίο καθιστά τον χρόνο υλοποίησης του αλγορίθμου μη εφικτό (περίπου 112,28 ώρες). Η μέθοδος αποζημίωσης που επιλέχθηκε είναι η *Οριακή Τιμή του Συστήματος (ΟΤΣ)*. Τα αποτελέσματα παρατίθενται στον Πίνακα 5.4.

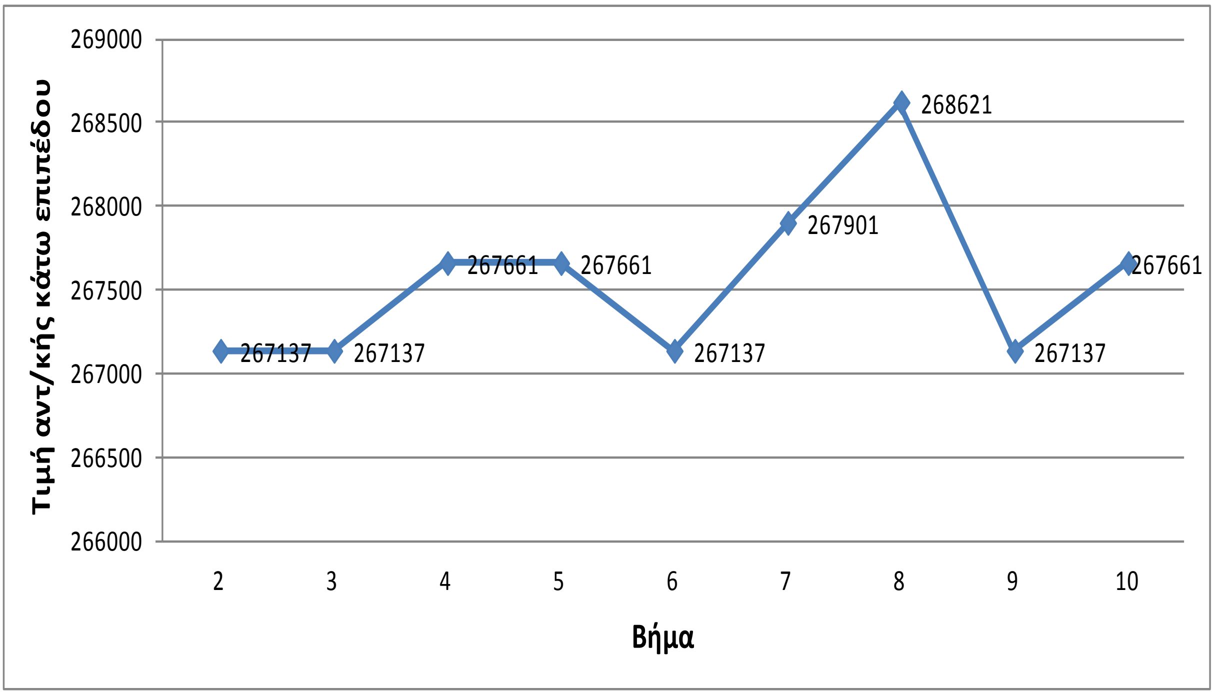
Βήμα αλλαγής	**Λύση κάτω επιπέδου** $(Q_{u,h})$	**Λύση άνω επιπέδου (Προσφορές** $P_{u,h}$**)**	**Τιμή αντ/κής κάτω επιπέδου** (f^*)	**Τιμή αντ/κής άνω επιπέδου** (F^*)	**Χρόνος υλοποίησης (minutes)**
2	(377,377,377,284)	(50,50,50,68)	267137	16799	456,2
3	(377,377,377,284)	(50,50,50,68)	267137	16799	102,01
4	(377,377,377,240)	(50,50,50,70)	267661	16247	38,02
5	(377,377,377,240)	(50,50,50,70)	267661	16247	13,06
6	(377,377,377,284)	(50,50,50,68)	267137	16799	8,81
7	(377,377,377,240)	(50,50,50,71)	267901	16247	6,15
8	(377,377,377,240)	(50,50,50,74)	268621	16247	3,83
9	(377,377,377,284)	(50,50,50,68)	267137	16799	1,98
10	(377,377,377,240)	(50,50,50,70)	267661	16247	1,18

Πίνακας 5.4 Αριθμητικά αποτελέσματα του πρώτου παραδείγματος με την μέθοδο αποζημίωσης ΟΤΣ

Στα παρακάτω Διαγράμματα 5.1 και 5.2 εμφανίζονται οι τιμές των αντικειμενικών συναρτήσεων άνω και κάτω επιπέδου αντίστοιχα συναρτήσει του βήματος αλλαγής.

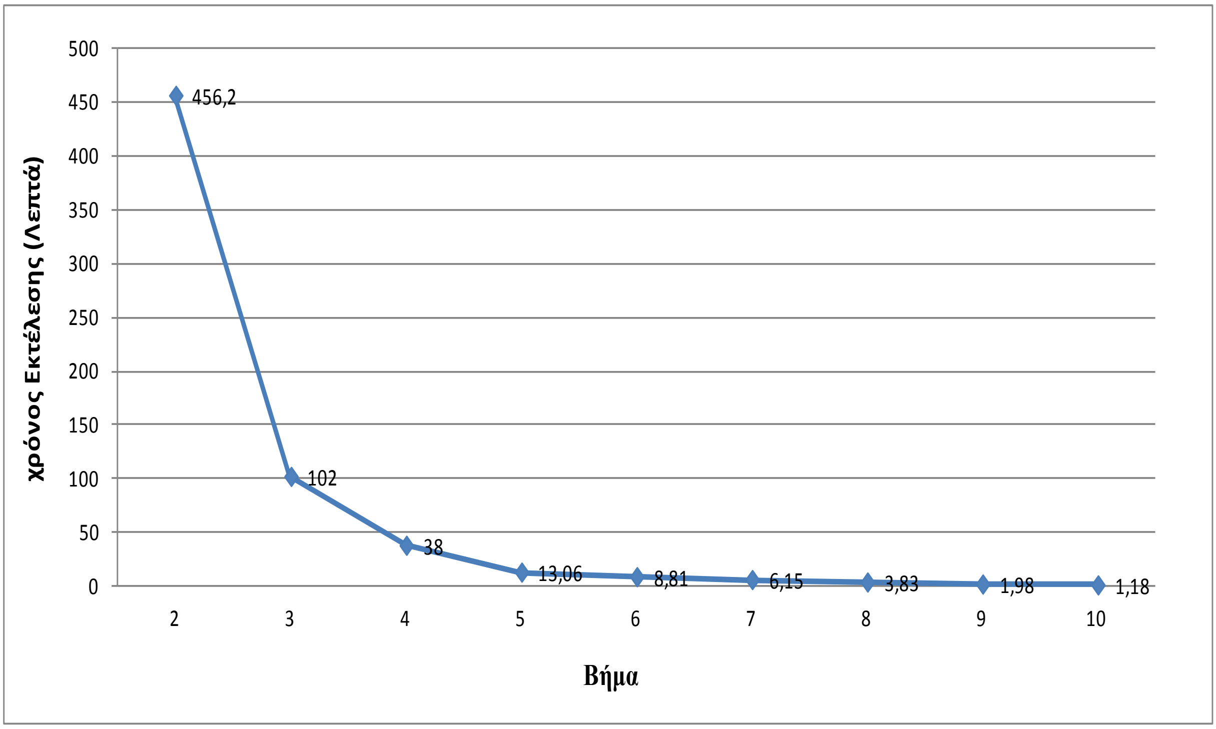


Διάγραμμα 5.1 Τιμές αντικειμενικής συνάρτησης άνω επιπέδου συναρτήσει του βήματος (ΟΤΣ)



Διάγραμμα 5.2 Τιμές αντικειμενικής συνάρτησης κάτω επιπέδου συναρτήσει του βήματος (ΟΤΣ)

Μία ακόμα ενδιαφέρουσα σύγκριση είναι αυτή του χρόνου εκτέλεσης του αλγορίθμου συναρτήσει του βήματος αλλαγής. Όπως είναι αναμενόμενο, όσο μεγαλύτερο είναι το βήμα τόσο μικρότερος είναι και χρόνος εκτέλεσης καθώς μειώνεται ο αριθμός των πιθανών συνδυασμών των προσφορών που υποβάλλονται, συνεπώς και ο αριθμός των επαναλήψεων για τις οποίες εκτελείται ο αλγόριθμος. Ο ρυθμός με τον οποίο μεταβάλλεται ο χρόνος εκτέλεσης συναρτήσει του βήματος αλλαγής απεικονίζεται στο Διάγραμμα 5.3.



Διάγραμμα 5.3 Χρόνος εκτέλεσης του αλγορίθμου συναρτήσει του βήματος αλλαγής (ΟΤΣ)

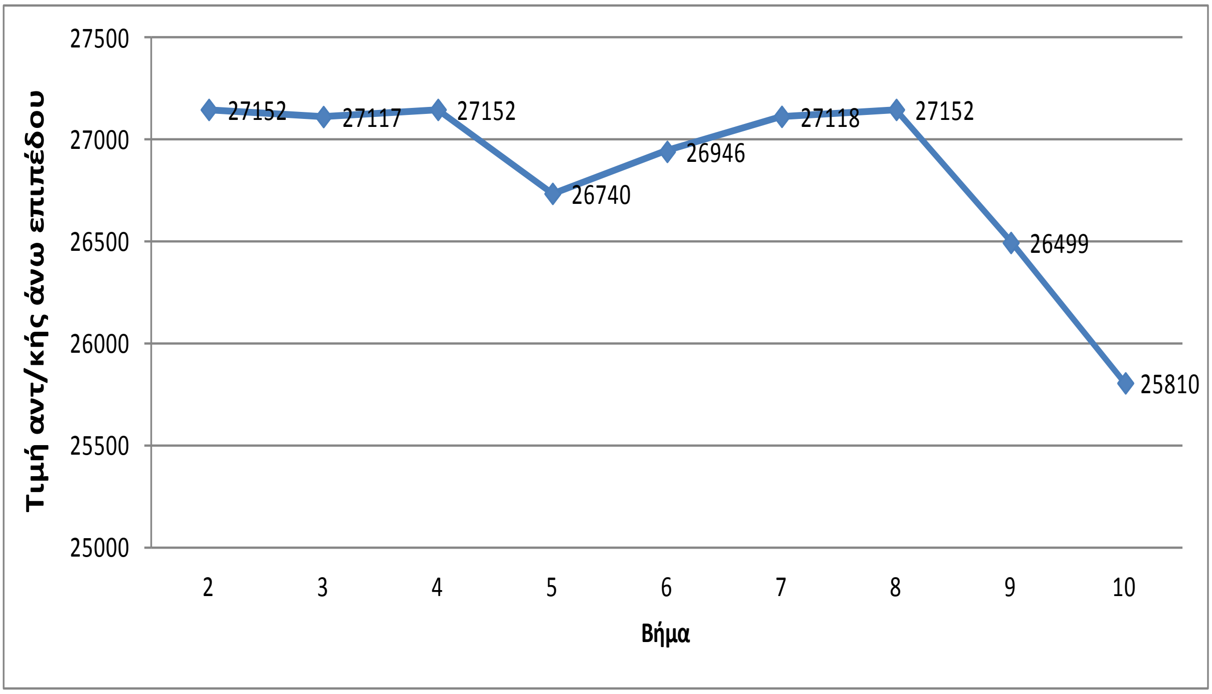
5.3.4 Πρώτο παράδειγμα με τη μέθοδο αποζημίωσης βάσει της υποβληθείσας προσφοράς

Τα πειράματα που υλοποιήθηκαν σε αυτή την περίπτωση αφορούν πάλι στην εφαρμογή διαφορετικού *βήματος αλλαγής* και κατά ποιον τρόπο επηρεάζεται η τιμή της αντικειμενικής συνάρτησης καθώς και ο χρόνος υλοποίησης του αλγορίθμου από αυτήν την αλλαγή. Το βήμα που χρησιμοποιήθηκε πήρε τις τιμές 2, 3, 4, 5, 6, 7, 8, 9,10. Ο λόγος που δεν λαμβάνεται υπ’ όψιν το βήμα 1 προαναφέρθηκε στην περίπτωση της *Οριακής Τιμής Συστήματος*. Η μέθοδος αποζημίωσης που επιλέχθηκε είναι η *Αποζημίωση βάσει της Υποβληθείσας Προσφοράς (ΑΑΠ)*. Τα δεδομένα που χρησιμοποιούνται αντλούνται από τους Πίνακες 5.2, 5.3 και 5.4. Τα αποτελέσματα παρατίθενται στον Πίνακα 5.5.

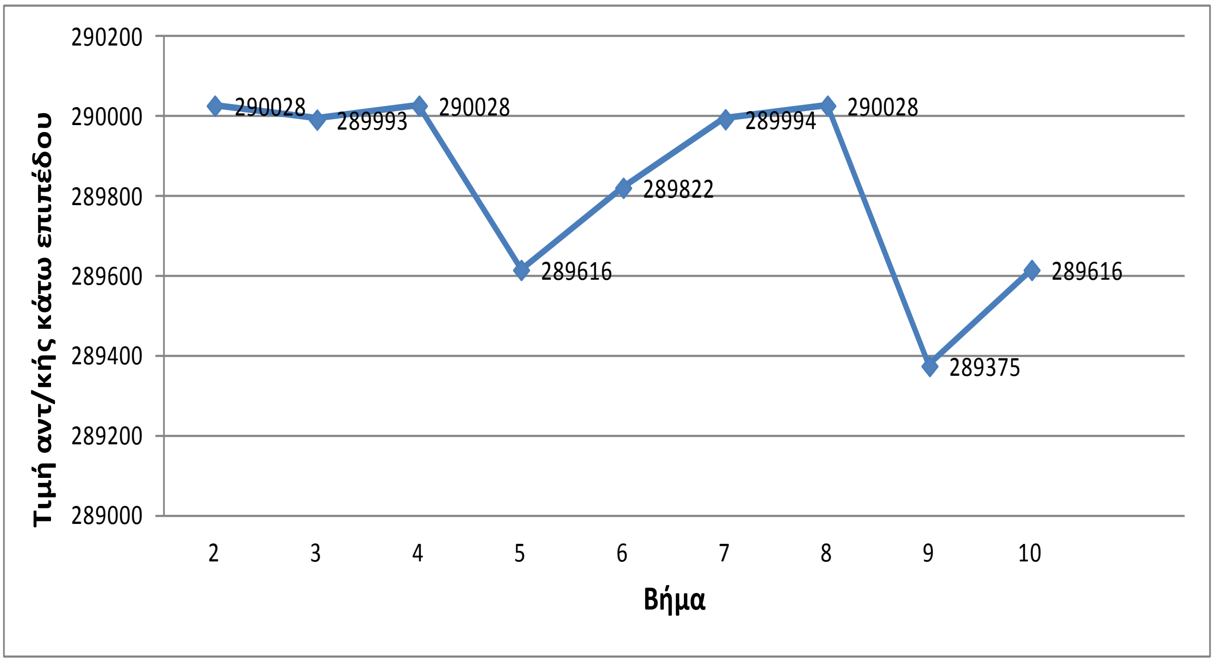
Βήμα αλλαγής	Λύση κάτω επιπέδου $(Q_{u,h})$	Λύση άνω επιπέδου (Προσφορές $P_{u,h}$)	Τιμή αντ/κής κάτω επιπέδου (f^*)	Τιμή αντ/κής άνω επιπέδου (F^*)	Χρόνος υλοποίησης (minutes)
2	(240,240,377,377)	(88,100,58,58)	290028	27152	444,48
3	(240,240,377,377)	(80,100,62,59)	289993	27117	104,3
4	(240,240,377,377)	(90,98,58,58)	290028	27152	35,83
5	(240,240,377,377)	(80,100,65,55)	289616	26740	14,58
6	(240,240,377,377)	(86,98,62,56)	289822	26946	10,31
7	(240,240,377,377)	(92,99,57,57)	289994	27118	8,12
8	(240,240,377,377)	(90,98,58,58)	290028	27152	4,16
9	(240,240,377,377)	(60,100,68,59)	289375	26499	2,63
10	(240,240,377,284)	(80,100,60,60)	289616	25810	1,51

Πίνακας 5.5 Αριθμητικά αποτελέσματα του πρώτου παραδείγματος με την μέθοδο αποζημίωσης ΑΑΠ

Στα παρακάτω Διαγράμματα 5.4 και 5.5 εμφανίζονται οι τιμές των αντικειμενικών συναρτήσεων άνω και κάτω επιπέδου αντίστοιχα συναρτήσει του βήματος αλλαγής.

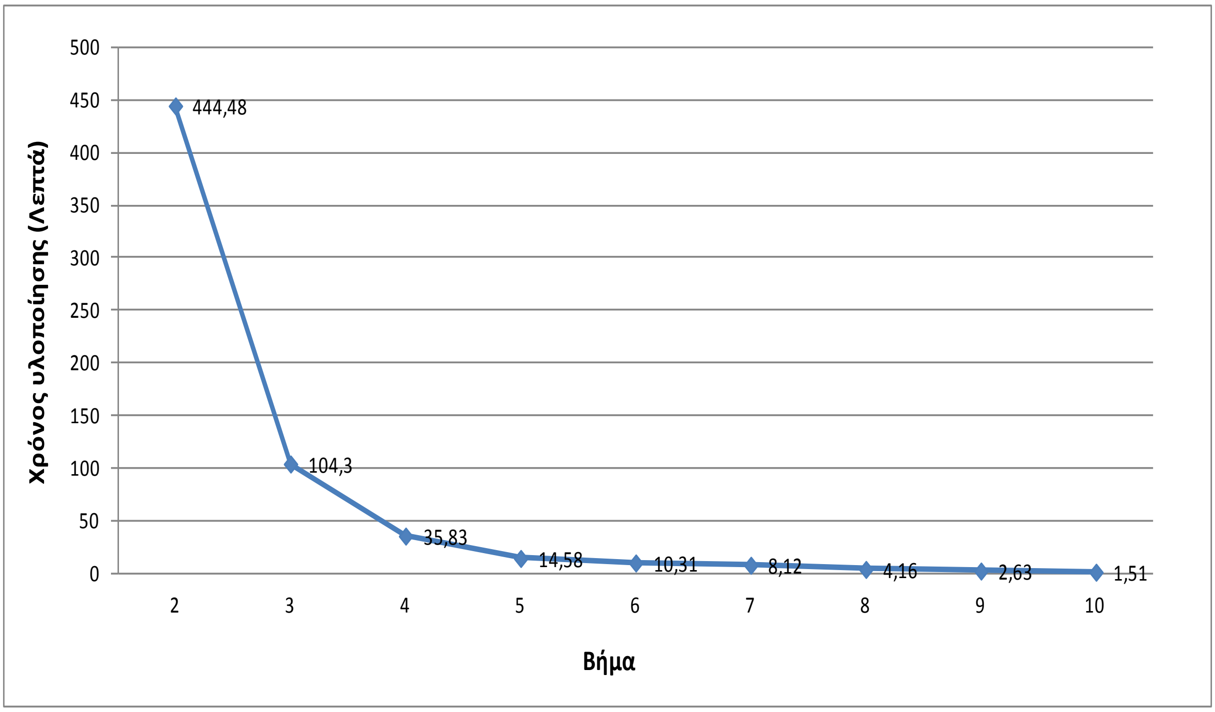


Διάγραμμα 5.4 Τιμές αντικειμενικής συνάρτησης άνω επιπέδου συναρτήσει του βήματος (ΑΑΠ)



Διάγραμμα 5.5 Τιμές αντικειμενικής συνάρτησης κάτω επιπέδου συναρτήσει του βήματος (ΑΑΠ)

Στη συνέχεια πραγματοποιείται η σύγκριση μεταξύ του χρόνου εκτέλεσης του αλγορίθμου και του βήματος αλλαγής. Όσο μεγαλύτερο είναι το βήμα τόσο μικρότερος είναι και χρόνος εκτέλεσης καθώς μειώνεται ο αριθμός των πιθανών συνδυασμών των προσφορών που υποβάλλονται. Ο ρυθμός με τον οποίο μεταβάλλεται ο χρόνος εκτέλεσης συναρτήσει του βήματος αλλαγής για τη μέθοδο *ΑΑΠ* απεικονίζεται στο Διάγραμμα 5.6.



Διάγραμμα 5.6 Χρόνος εκτέλεσης του αλγορίθμου συναρτήσει του βήματος αλλαγής (ΑΑΠ)

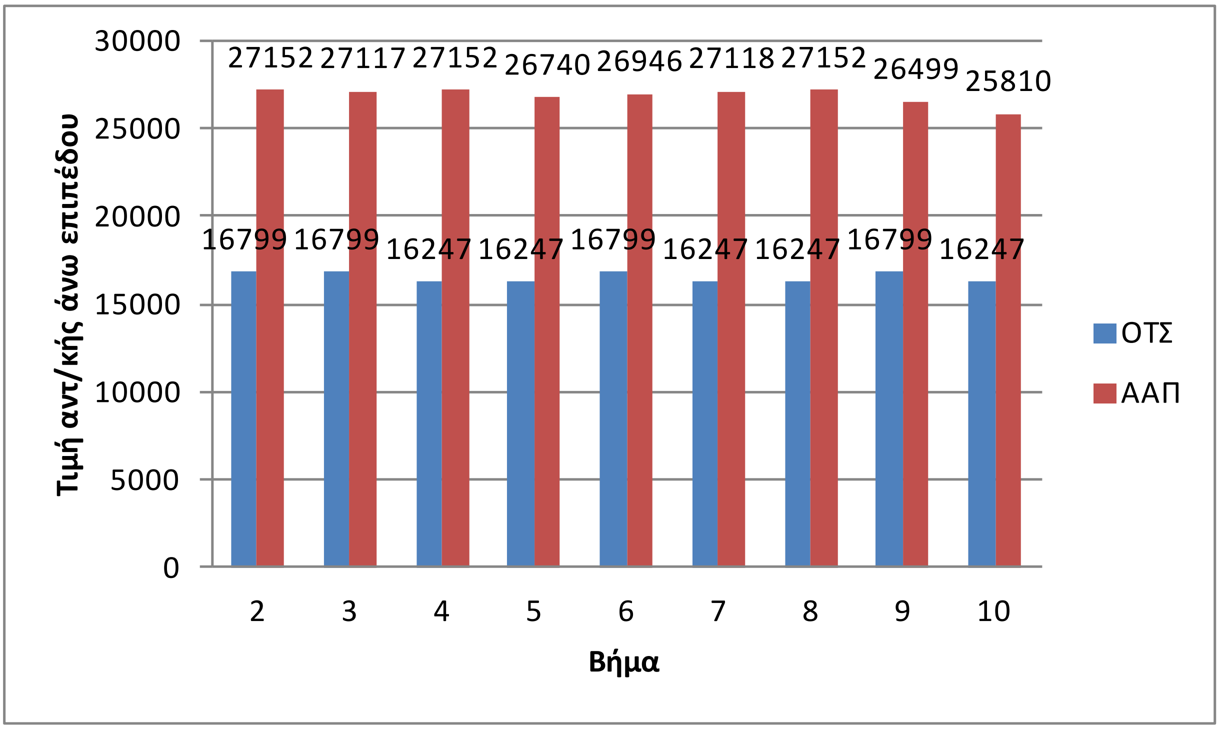
5.4.2 Συγκριτική ανάλυση των δύο μεθόδων αποζημίωσης για το πρώτο παράδειγμα

Τα στοιχεία που διαδραματίζουν το σημαντικότερο ρόλο στη σύγκριση των δύο μεθόδων αποζημίωσης είναι τιμή της άνω αντικειμενικής συνάρτησης και ο χρόνος υλοποίησης του αλγορίθμου, δίνοντας μεγαλύτερη βαρύτητα στο πρώτο στοιχείο ως βασικότερο μέτρο σύγκρισης.

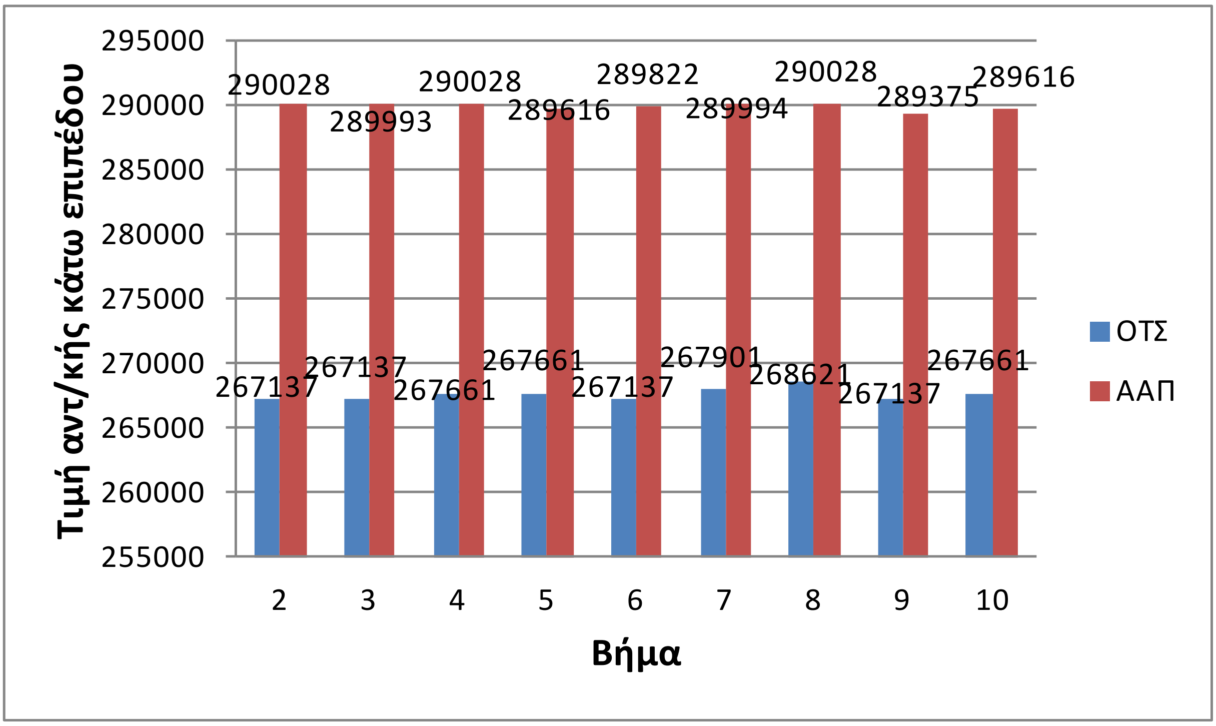
Παρατηρώντας τις τιμές της αντικειμενικής συνάρτησης άνω επιπέδου, δηλαδή το κέρδος του μεμονωμένου παραγωγού, για τις δύο μεθόδους αποζημίωσης προκύπτει ότι η μέθοδος αποζημίωσης *ΑΑΠ* επιφέρει μεγαλύτερο κέρδος στον μεμονωμένο παραγωγό. Πιο συγκεκριμένα, το μέγιστο κέρδος της μεμονωμένης παραγωγικής μονάδας με την μέθοδο *ΟΤΣ* ανέρχεται σε 16799 € ενώ με την μέθοδο *ΑΑΠ* το κέρδος αυξάνεται σε 27152 €. Η διαφορά μεταξύ των δύο κερδών ανέρχεται σε 10353 €, δηλαδή κατά 38,12 % μεγαλύτερο το κέρδος της μεθόδου *ΑΑΠ* από αυτό της μεθόδου *ΟΤΣ* , μια διαφορά αρκετά σημαντική.

Βέβαια, στο σημείο αυτό προκύπτει μια νέα ενδιαφέρουσα παρατήρηση. Στην περίπτωση της μεθόδου *ΑΑΠ* το κόστος του συστήματος, δηλαδή της τιμής της αντικειμενικής συνάρτησης κάτω επιπέδου, είναι μεγαλύτερο κατά 7 % από αυτό της μεθόδου *ΟΤΣ*, γεγονός το οποίο επιφέρει μεγαλύτερη ζημία στον Ανεξάρτητο Διαχειριστή του Συστήματος. Σύμφωνα με τους Kozanidis et.al (2011), μια αγορά ενέργειας που αποζημιώνει τους παραγωγούς σύμφωνα με τη μέθοδο *ΑΑΠ* είναι ανεπαρκώς σχεδιασμένη καθώς επιτρέπει την πιθανή χειραγώγηση των τιμών από τους παραγωγούς και δίνει κίνητρο για υποβολή ακριβών προσφορών αφού μετά την εκκαθάριση της αγοράς ο παραγωγός θα αποζημιωθεί σύμφωνα με την τιμή της προσφοράς που έχει υποβάλει.

Η σύγκριση μεταξύ των δύο μεθόδων αποζημίωσης ως προς τις τιμές της αντικειμενικής συνάρτησης άνω και κάτω επιπέδου ανά βήμα απεικονίζεται στα Διαγράμματα 5.7 και 5.8 αντίστοιχα.

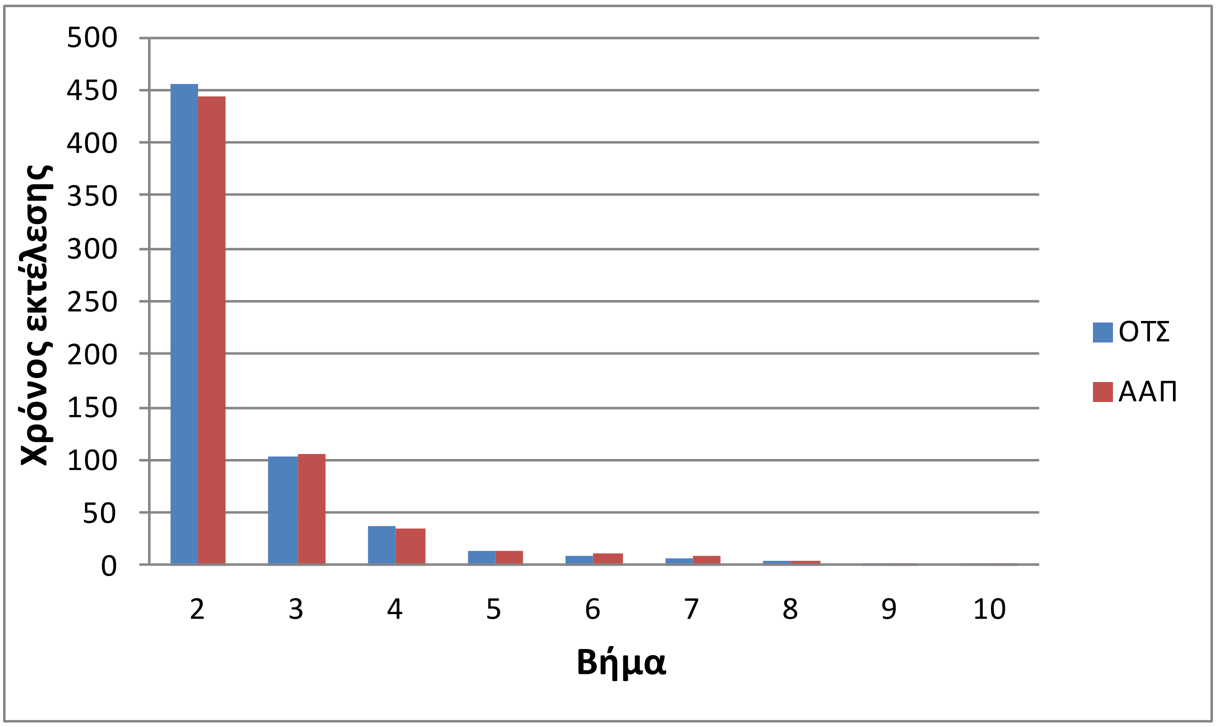


Διάγραμμα 5.7 Σύγκριση των μεθόδων αποζημίωσης ως προς τις τιμές της αντικειμενικής συνάρτησης άνω επιπέδου



Διάγραμμα 5.8 Σύγκριση των μεθόδων αποζημίωσης ως προς τις τιμές της αντικειμενικής συνάρτησης κάτω επιπέδου

Όσον αφορά τους χρόνους εκτέλεσης του αλγορίθμου για τις δύο μεθόδους αποζημίωσης δεν παρουσιάζονται σημαντικές διαφορές. Το αποτέλεσμα αυτό είναι αναμενόμενο καθώς το μέγεθος και η πολυπλοκότητα του προβλήματος είναι παρόμοια και για τις δύο περιπτώσεις. Αυτό που διαφοροποιεί ελάχιστα τους χρόνους είναι ο τρόπος υπολογισμού της Οριακής Τιμής Συστήματος σε κάθε επανάληψη του αλγορίθμου, γεγονός το οποίο μπορεί να προβεί περισσότερο χρονοβόρο από το να αποζημιωθεί ο παραγωγός απλά με την τιμή της προσφοράς που έχει υποβάλλει. Η σύγκριση μεταξύ των δύο μεθόδων αποζημίωσης ως προς το χρόνο εκτέλεσής τους παρατίθεται στο Διάγραμμα 5.9.



Διάγραμμα 5.9 Σύγκριση των μεθόδων αποζημίωσης ως προς τον χρόνο υλοποίησής τους

5.5 Δεύτερο Παράδειγμα βάσει της μεθόδου αποζημίωσης Οριακής Τιμής Συστήματος

Στο παράδειγμα που ακολουθεί υπάρχουν τέσσερις (4) μονάδες παραγωγής ενέργειας και τέσσερις (4) χρονικές περίοδοι. Τα τεχνικά χαρακτηριστικά, οι τιμές-προσφοράς και τα κόστη εκκίνησης των μονάδων παραγωγής παρουσιάζονται στον Πίνακα 5.6. Τα τεχνικά ελάχιστα και μέγιστα δίνονται σε MW, τα κόστη εκκίνησης σε € και οι τιμές-προσφοράς για την ενέργεια σε €/MWh. Το μεταβλητό κόστος του μεμονωμένου παραγωγού (της μονάδας 1) είναι 40 €/MWh, το άνω όριο στην τιμή προσφοράς του είναι 100 €/MWh. Η ζήτηση για ενέργεια διαφέρει για κάθε χρονική περίοδο και απεικονίζεται στον Πίνακα 5.8. Τα δεδομένα των μονάδων παραγωγής προσεγγίζουν την πραγματικότητα της ελληνικής αγοράς ενέργειας και οι τελικές λύσεις του προβλήματος είναι γνωστές και απεικονίζονται στον Πίνακα 5.9.

Μονάδα (u)	$Q_u^{\min}$	$Q_u^{\max}$	SUC_u
1	240	400	1100
2	200	500	1200
3	100	300	1000
4	150	350	1300

Πίνακας 5.6 Δεδομένα Δεύτερου Παραδείγματος

Στον Πίνακα 5.7 απεικονίζονται οι προσφορές που υποβάλει ο κάθε παραγωγός για κάθε χρονική περίοδο h. Οι τιμές είναι σε €/MWh.

	Περίοδος ((h)			
Μονάδα (u)	**1**	**2**	**3**	**4**
1	---	---	---	---
2	58	55	68	60
3	54	50	65	69
4	62	60	58	56

Πίνακας 5.7 Προσφορές μονάδων παραγωγής για κάθε χρονική περίοδο

Στον Πίνακα 5.8 απεικονίζεται η ζήτηση για ενέργεια για κάθε χρονική περίοδο σε MWh.

	Περίοδος ((h)			
	1	**2**	**3**	**4**
Ζήτηση (D_h)	1000	900	850	1050

Πίνακας 5.8 Ζήτηση για ενέργεια σε κάθε χρονική περίοδο

Στον Πίνακα 5.9 απεικονίζονται οι βέλτιστες λύσεις του προβλήματος για τις δύο μεθόδους αποζημίωσης.

Μέθοδος Αποζημίωσης	**Λύση κάτω επιπέδου** ($Q_{u,h}$)	**Λύση άνω επιπέδου (Προσφορές $P_{u,h}$)**	**Τιμή αντ/κής κάτω επιπέδου** (f^*)	**Τιμή αντ/κής άνω επιπέδου** (F^*)
ΟΤΣ	(400,400,400,400)	(53,49,58,55)	215800	31200
ΑΑΠ	(400,400,300,400)	(58,55,67,60)	225000	29300

Πίνακας 5.9 Βέλτιστες λύσεις του προβλήματος για τις δύο μεθόδους αποζημίωσης

5.5.1 Αριθμητικά Αποτελέσματα πειραμάτων

Τα πειράματα που υλοποιηθήκαν για το δεύτερο παράδειγμα αφορούν στην εφαρμογή διαφορετικού *βήματος αλλαγής* και κατά ποιον τρόπο επηρεάζεται η τιμή της αντικειμενικής συνάρτησης καθώς και ο χρόνος υλοποίησης του αλγορίθμου από αυτήν την αλλαγή. Η μέθοδος αποζημίωσης που επιλέχθηκε είναι η *Οριακή Τιμή του Συστήματος (ΟΤΣ)*. Τα αποτελέσματα που προέκυψαν από την εκτέλεση των πειραμάτων με την συγκεκριμένη μέθοδο αποζημίωσης είναι τα ίδια ανεξαρτήτως του *βήματος*. Αυτό οφείλεται στο γεγονός ότι η βέλτιστη λύση βρίσκεται στον πρώτο συνδυασμό των προσφορών, ο οποίος δημιουργείται ανεξάρτητα από το *βήμα*. Τα αποτελέσματα παρατίθενται στον Πίνακα 5.10.

Λύσεις κάτω επιπέδου $(Q_{u,h})$	Λύσεις άνω επιπέδου (Προσφορές $P_{u,h}$)	Τιμή αντ/κής κάτω επιπέδου (f^*)	Τιμή αντ/κής άνω επιπέδου (F^*)	Χρόνος εκτέλεσης (minutes)
(400,400,400,400)	(40,40,40,40)	193800	31200	913,3

Πίνακας 5.10 Αριθμητικά αποτελέσματα του δεύτερου παραδείγματος με την μέθοδο αποζημίωσης ΟΤΣ

Στο συγκεκριμένο παράδειγμα η βέλτιστη λύση είναι η ίδια ανεξάρτητα από το *βήμα αλλαγής* και επειδή η ολική βέλτιστη λύση είναι γνωστή και επιβεβαιώνεται, η σύγκριση των τιμών των αντικειμενικών συναρτήσεων άνω και κάτω επιπέδου δεν παρουσιάζει ιδιαίτερο ενδιαφέρον γι' αυτό και δεν παρατίθεται.

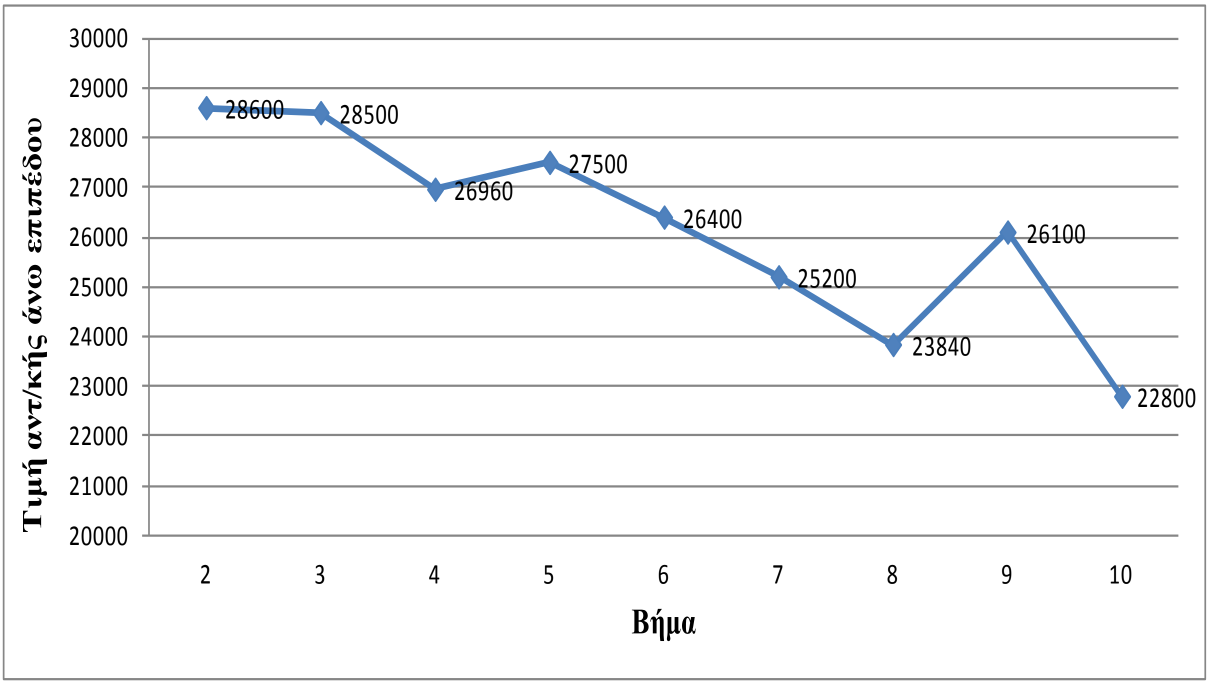
5.5.2 Δεύτερο παράδειγμα με τη μέθοδο Αποζημίωσης βάσει της Υποβληθείσας Προσφοράς

Τα πειράματα που υλοποιήθηκαν σε αυτή την περίπτωση αφορούν πάλι στην εφαρμογή διαφορετικού *βήματος αλλαγής* και κατά ποιον τρόπο επηρεάζεται η τιμή της αντικειμενικής συνάρτησης καθώς και ο χρόνος υλοποίησης του αλγορίθμου από αυτήν την αλλαγή. Το βήμα που χρησιμοποιήθηκε πήρε τις τιμές 2, 3, 4, 5, 6, 7, 8, 9,10. Ο λόγος που δεν λαμβάνεται υπ' όψιν το βήμα 1 προαναφέρθηκε στην περίπτωση της *Οριακής Τιμής Συστήματος*. Η μέθοδος αποζημίωσης που επιλέχθηκε είναι η *Αποζημίωση βάσει της Υποβληθείσας Προσφοράς (ΑΑΠ)*. Τα δεδομένα που χρησιμοποιούνται αντλούνται από τους Πίνακες 5.6, 5.7 και 5.8. Τα αποτελέσματα παρατίθενται στον Πίνακα 5.11.

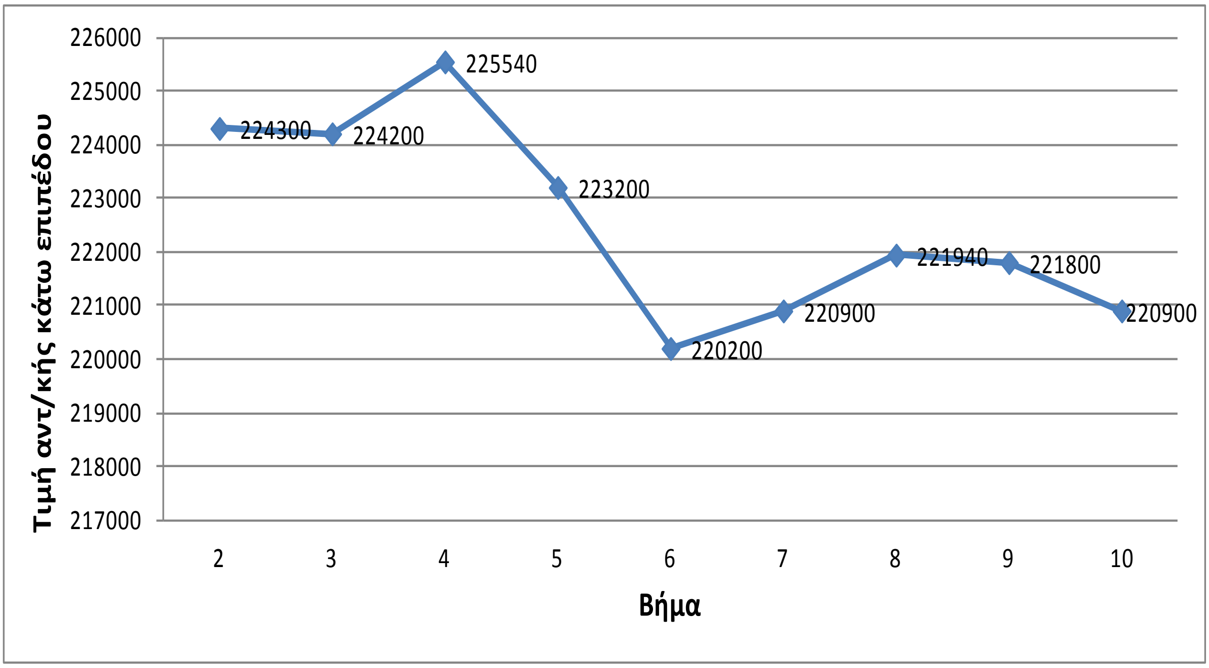
Βήμα αλλαγής	**Λύσεις κάτω επιπέδου** $(Q_{u,h})$	**Λύσεις άνω επιπέδου (Προσφορές** $P_{u,h}$**)**	**Τιμή αντ/κής κάτω επιπέδου** (f^*)	**Τιμή αντ/κής άνω επιπέδου** (F^*)	**Χρόνος εκτέλεσης (minutes)**
2	(400,400,300,400)	(58,54,66,60)	224300	28600	905,3
3	(400,400,300,400)	(58,55,67,58)	224200	28500	179
4	(240,400,300,400)	(64,52,68,60)	225540	26960	64,7
5	(400,400,300,400)	(55,55,65,60)	223200	27500	25,7
6	(400,400,400,400)	(58,52,58,58)	220200	26400	12,4
7	(400,400,300,400)	(54,54,68,54)	220900	25200	10,5
8	(400,240,300,400)	(56,56,64,56)	221940	23840	4,1
9	(400,400,300,400)	(58,49,67,58)	221800	26100	2,1
10	(400,240,300,400)	(50,60,60,60)	220900	22800	1,9

Πίνακας 5.11 Αριθμητικά αποτελέσματα του δεύτερου παραδείγματος με την μέθοδο αποζημίωσης ΑΑΠ

Στα παρακάτω Διαγράμματα 5.10 και 5.11 εμφανίζονται οι τιμές των αντικειμενικών συναρτήσεων άνω και κάτω επιπέδου αντίστοιχα συναρτήσει του βήματος αλλαγής.

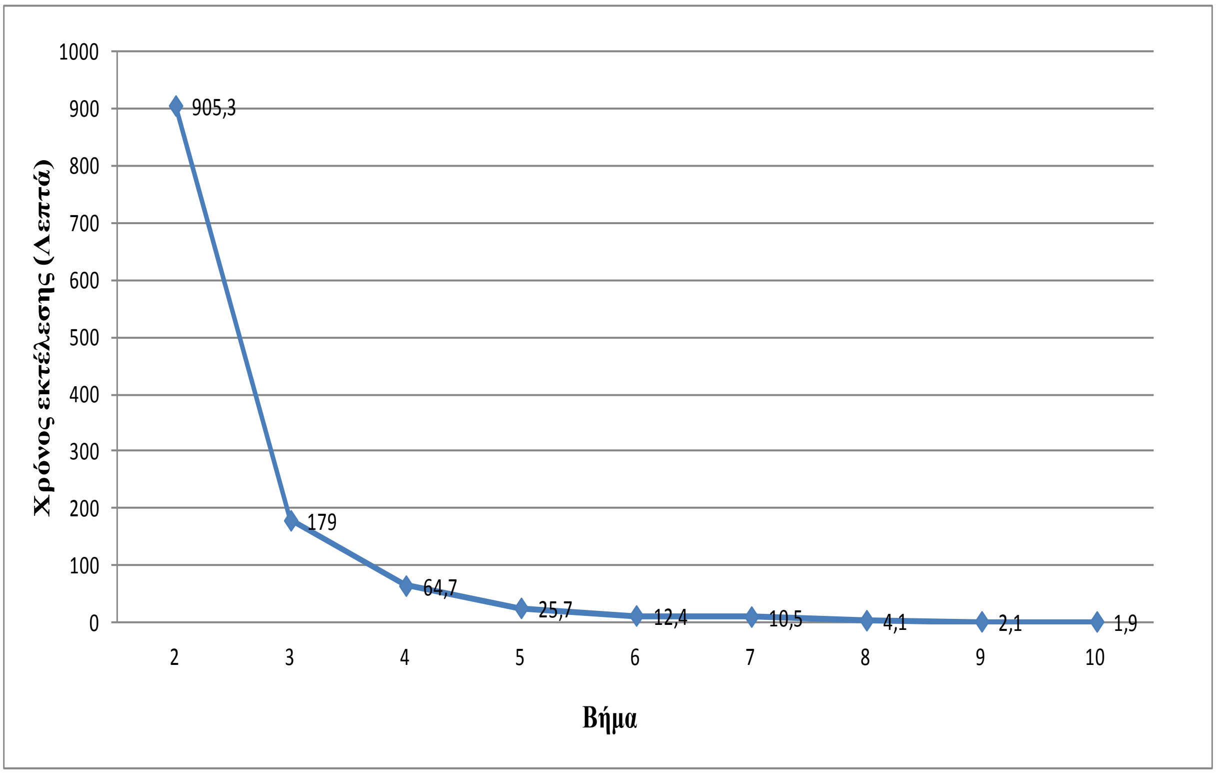


Διάγραμμα 5.10 Τιμές αντικειμενικής συνάρτησης άνω επιπέδου συναρτήσει του βήματος (ΑΑΠ)



Διάγραμμα 5.11 Τιμές αντικειμενικής συνάρτησης κάτω επιπέδου συναρτήσει του βήματος (ΑΑΠ)

Στη συνέχεια πραγματοποιείται η σύγκριση μεταξύ του χρόνου εκτέλεσης του αλγορίθμου και του βήματος αλλαγής. Όσο μεγαλύτερο είναι το βήμα τόσο μικρότερος είναι και χρόνος εκτέλεσης καθώς μειώνεται ο αριθμός των πιθανών συνδυασμών των προσφορών που υποβάλλονται. Ο ρυθμός με τον οποίο μεταβάλλεται ο χρόνος εκτέλεσης συναρτήσει του βήματος αλλαγής για τη μέθοδο *ΑΑΠ* απεικονίζεται στο Διάγραμμα 5.12.



Διάγραμμα 5.12 Χρόνος εκτέλεσης του αλγορίθμου συναρτήσει του βήματος αλλαγής (ΑΑΠ)

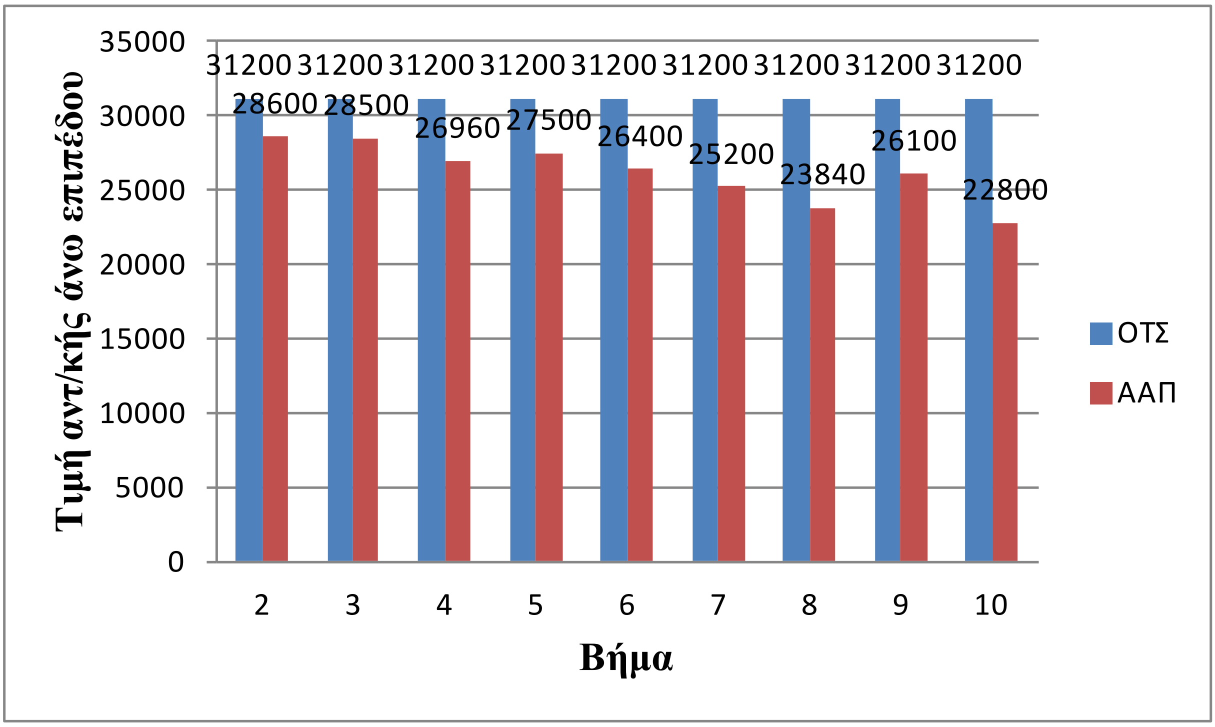
5.5.3 Συγκριτική ανάλυση των δύο μεθόδων αποζημίωσης για το δεύτερο παράδειγμα

Τα στοιχεία που διαδραματίζουν το σημαντικότερο ρόλο στη σύγκριση των δύο μεθόδων αποζημίωσης είναι τιμή της άνω αντικειμενικής συνάρτησης και ο χρόνος υλοποίησης του αλγορίθμου, δίνοντας μεγαλύτερη βαρύτητα στο πρώτο στοιχείο ως βασικότερο μέτρο σύγκρισης.

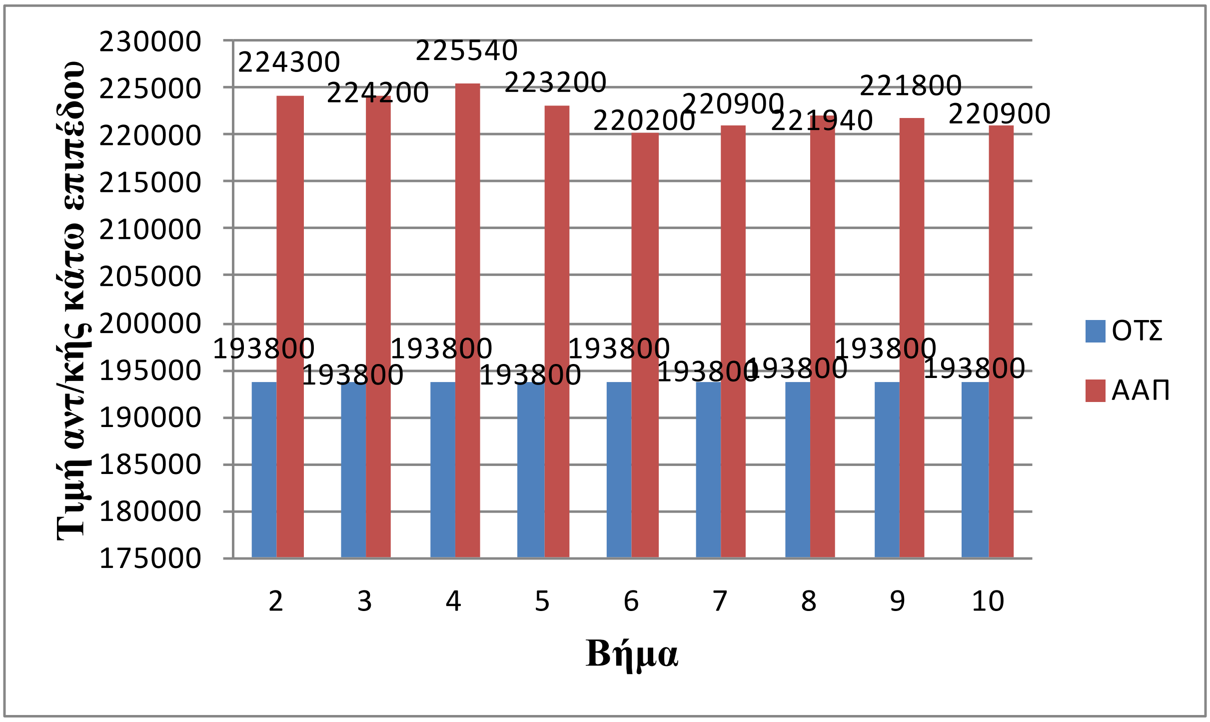
Παρατηρώντας τις τιμές της αντικειμενικής συνάρτησης άνω επιπέδου, δηλαδή το κέρδος του μεμονωμένου παραγωγού, για τις δύο μεθόδους αποζημίωσης προκύπτει ότι η μέθοδος αποζημίωσης *ΟΤΣ* επιφέρει μεγαλύτερο κέρδος στον παραγωγό. Πιο συγκεκριμένα, το μέγιστο κέρδος της μεμονωμένης παραγωγικής μονάδας με την μέθοδο *ΑΑΠ* ανέρχεται σε 28600 € ενώ με την μέθοδο *ΟΤΣ* το κέρδος αυξάνεται σε 31200 €. Η διαφορά μεταξύ των δύο κερδών ανέρχεται σε 2600 €, δηλαδή κατά 8,08 % μεγαλύτερο το κέρδος της μεθόδου *ΟΤΣ* από αυτό της μεθόδου *ΑΑΠ*.

Στο σημείο αυτό παρατηρείται ότι στην περίπτωση της μεθόδου *ΑΑΠ* το κόστος του συστήματος, δηλαδή της τιμής της αντικειμενικής συνάρτησης κάτω επιπέδου, είναι μεγαλύτερο κατά 13,59 % από αυτό της μεθόδου *ΟΤΣ*, γεγονός το οποίο επιφέρει μεγαλύτερη ζημία στον Ανεξάρτητο Διαχειριστή του Συστήματος χωρίς αυτό να συνεπάγεται μεγαλύτερο κέρδος για τον μεμονωμένο παραγωγό όπως στο πρώτο παράδειγμα. Από τα παραπάνω προκύπτει ότι για το συγκεκριμένο παράδειγμα δεν συμφέρει ούτε τον Ανεξάρτητο Διαχειριστή του Συστήματος ούτε τον μεμονωμένο παραγωγό να χρησιμοποιηθεί η μέθοδος *ΑΑΠ*.

Η σύγκριση μεταξύ των δύο μεθόδων αποζημίωσης ως προς τις τιμές της αντικειμενικής συνάρτησης άνω και κάτω επιπέδου ανά βήμα απεικονίζεται στα Διαγράμματα 5.13 και 5.14 αντίστοιχα.

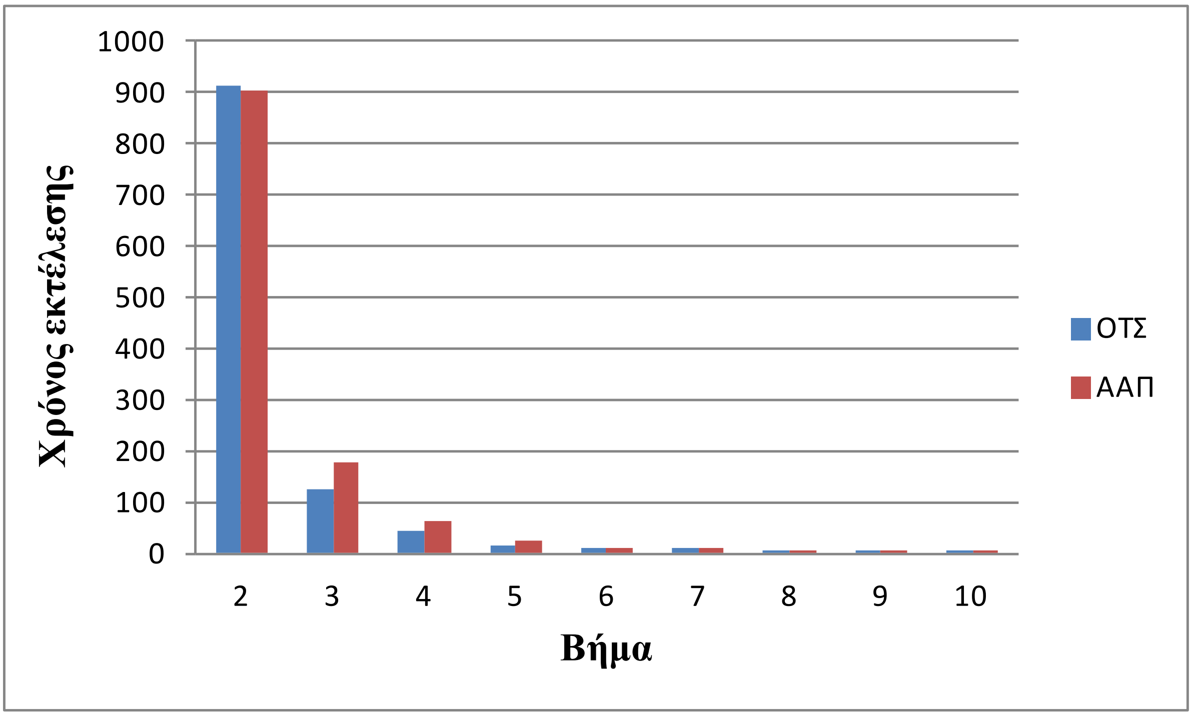


Διάγραμμα 5.13 Σύγκριση των μεθόδων αποζημίωσης ως προς τις τιμές της αντικειμενικής συνάρτησης άνω επιπέδου



Διάγραμμα 5.14 Σύγκριση των μεθόδων αποζημίωσης ως προς τις τιμές της αντικειμενικής συνάρτησης κάτω επιπέδου

Όσον αφορά τους χρόνους εκτέλεσης του αλγορίθμου για τις δύο μεθόδους αποζημίωσης δεν παρουσιάζονται σημαντικές διαφορές, όπως και στο πρώτο παράδειγμα. Το αποτέλεσμα αυτό είναι αναμενόμενο καθώς το μέγεθος και η πολυπλοκότητα του προβλήματος είναι παρόμοια και για τις δύο περιπτώσεις. Αυτό που διαφοροποιεί ελάχιστα τους χρόνους είναι ο τρόπος υπολογισμού της Οριακής Τιμής Συστήματος σε κάθε επανάληψη του αλγορίθμου, γεγονός το οποίο μπορεί να προβεί περισσότερο χρονοβόρο από το να αποζημιωθεί ο παραγωγός απλά με την τιμή της προσφοράς που έχει υποβάλλει. Η σύγκριση μεταξύ των δύο μεθόδων αποζημίωσης ως προς το χρόνο υλοποίησής τους παρατίθεται στο Διάγραμμα 5.15.



Διάγραμμα 5.15 Σύγκριση των μεθόδων αποζημίωσης ως προς τον χρόνο εκτέλεσης τους

5.6 Εναλλακτικές μέθοδοι εφαρμογής του αλγορίθμου

Στην ενότητα αυτή παρουσιάζονται κάποιες εναλλακτικές μέθοδοι με τις οποίες μπορεί να εφαρμοστεί ο αλγόριθμος για το παραπάνω πρόβλημα διεπίπεδου γραμμικού προγραμματισμού. Ο σκοπός της δημιουργίας και εφαρμογής αυτών των μεθόδων είναι τόσο η επιβεβαίωση της ακρίβειας των αποτελεσμάτων που προέκυψαν από την εκτέλεση των πειραμάτων όσο και η βελτίωσης τους. Βέβαια, για την εφαρμογή των εναλλακτικών αυτών μεθόδων στο πρόβλημα απαιτούνται τροποποιήσεις του αλγορίθμου ως προς κάποιες παραμέτρους. Οι παράμετροι ως προς τις οποίες τροποποιείται ο αλγόριθμος συνοψίζονται στις παρακάτω υποενότητες 5.6.1, 5.6.2, 5.6.3 και 5.6.4.

5.6.1 Επίλυση ως προς μία χρονική περίοδο

Στη μέθοδο αυτή αυτό το πρόβλημα μεγιστοποίησης του κέρδους για τον μεμονωμένο παραγωγό επιλύεται για κάθε μία χρονική περίοδο ξεχωριστά, κρατώντας τις προσφορές των υπόλοιπων χρονικών περιόδων σταθερές σε μια τιμή η οποία ορίζεται από τον χρήστη του αλγορίθμου και είναι μεταξύ του μοναδιαίου κόστους c_1 και του $P^{\max}$. Δηλαδή, λαμβάνονται υπ’ όψιν μόνο οι συνδυασμοί των προσφορών που προκύπτουν για την χρονική περίοδο που έχει επιλεχθεί ως η περίοδος για την οποία η προσφορά θα μεταβάλλεται. Το γεγονός αυτό μειώνει κατά πολύ το μέγεθος του προβλήματος αφού μειώνονται σε μεγάλο βαθμό οι επαναλήψεις για τις οποίες εκτελείται ο αλγόριθμος.

➢ *Εφαρμογή στο πρώτο παράδειγμα*

Από την εκ νέου εκτέλεση των πειραμάτων για το πρώτο παράδειγμα για κάθε μία περίοδο ξεχωριστά για την περίπτωση της *ΟΤΣ* παρατηρείται ότι οι προσφορές που επιφέρουν το μέγιστο κέρδος στον μεμονωμένο παραγωγό δεν ξεπερνούν την τιμή 69. Οπότε, επιλύοντας ξανά το πρόβλημα θέτοντας ως νέο άνω όριο στην τιμή προσφοράς το 69 ($P^{\max} = 69$) έχουμε την λύση 17083 (τιμή αντικειμενικής συνάρτησης άνω επιπέδου) για τον συνδυασμό των προσφορών 50,50,50,69. Στο σημείο αυτό επισημάνεται ότι πλέον είναι εφικτή η χρήση του 1 ως *βήμα αλλαγής*, γεγονός το οποίο αυξάνει την ακρίβεια της τελικής λύσης.

Η νέα λύση που προκύπτει είναι μεγαλύτερη από την λύση που βρέθηκε στα προηγούμενα υπολογιστικά πειράματα για *βήμα αλλαγής* 2,3,4,5,6,7,8,9,10 κατά 284 €, γεγονός το οποίο επιβεβαιώνει την συμβολή της παραπάνω μεθόδου στην βελτίωση της τελικής λύσης.

Όσον αφορά τον χρόνο εκτέλεσης του αλγορίθμου είναι 201,71 λεπτά, δηλαδή 221 λεπτά λιγότερο από το πείραμα με *βήμα αλλαγής* 2. Για να επιλυθεί το παραπάνω πρόβλημα με βήμα 1 θα χρειαζόταν 6.765.201 επαναλήψεις ενώ τώρα χρειάστηκαν 160.000.

Στην περίπτωση της *ΑΑΠ* η εφαρμογή της παραπάνω μεθόδου δεν επιφέρει κάποια βελτίωση της τελική λύσης καθώς από την επιμέρους επίλυση του προβλήματος για κάθε χρονική περίοδο χωριστά οι προσφορές που επιφέρουν το μέγιστο κέρδος στον μεμονωμένο παραγωγό ανέρχονται στο 100, τιμή η οποία είναι ήδη το άνω όριο του προβλήματος.

➢ *Εφαρμογή στο δεύτερο παράδειγμα*

Εφαρμόζοντας τη μέθοδο αυτή και στο δεύτερο παράδειγμα για την περίπτωση της *ΑΑΠ* παρατηρείται ότι οι προσφορές που επιφέρουν το μέγιστο κέρδος στον μεμονωμένο παραγωγό δεν ξεπερνούν την τιμή 67. Οπότε, επιλύοντας ξανά το πρόβλημα θέτοντας ως νέο άνω όριο στην τιμή προσφοράς το 67 $(P^{\max} = 67)$ έχουμε την λύση 29300 (τιμή αντικειμενικής συνάρτησης άνω επιπέδου) για τον συνδυασμό των προσφορών 58,55,67,60 η οποία είναι και η ολική βέλτιστη λύση. Στο σημείο αυτό επισημάνεται ότι πλέον είναι εφικτή η χρήση του 1 ως *βήμα αλλαγής*, γεγονός το οποίο αυξάνει την ακρίβεια της τελικής λύσης.

Η νέα λύση που προκύπτει είναι μεγαλύτερη από την λύση που βρέθηκε στα προηγούμενα υπολογιστικά πειράματα για *βήμα αλλαγής* 2,3,4,5,6,7,8,9,10 κατά 700 €, γεγονός το οποίο επιβεβαιώνει την συμβολή της παραπάνω μεθόδου στην βελτίωση της τελικής λύσης και για το δεύτερο παράδειγμα.

Όσον αφορά τον χρόνο εκτέλεσης του αλγορίθμου για το δεύτερο παράδειγμα είναι 601,1 λεπτά, δηλαδή 304 λεπτά λιγότερο από το πείραμα με *βήμα αλλαγής* 2.

Στην περίπτωση της *ΟΤΣ* η εφαρμογή της παραπάνω μεθόδου δεν επιφέρει κάποια βελτίωση της τελική λύσης καθώς από την αρχική επίλυση του προβλήματος η τελική βέλτιστη λύση έχει ήδη βρεθεί.

5.6.2 Ορισμός των τιμών προσφοράς για κάθε χρονική περίοδο από τον χρήστη

Από τις λύσεις του πρώτου παραδείγματος για την περίπτωση της *ΟΤΣ* παρατηρείται ότι οι προσφορές που υποβάλει ο μεμονωμένος παραγωγός τείνουν να συγκλίνουν στην περιοχή τιμών 66 έως 72 για τη τέταρτη περίοδο ενώ παραμένουν σταθερές στο 50 για τις υπόλοιπες τρείς περιόδους.

Το γεγονός αυτός συντέλεσε στην τροποποίηση του αλγορίθμου έτσι ώστε να δίνεται η δυνατότητα στον χρήστη να δοκιμάζει από μόνος του προσφορές που εκτιμάει αυτός ότι μπορεί να βελτιώσουν την τελική λύση.

- *Εφαρμογή στο πρώτο παράδειγμα*

Στα πειράματα που εκτελέστηκαν για το πρώτο παράδειγμα παρατηρείται ότι για την τέταρτη περίοδο η τιμή της προσφοράς αδυνατεί να πάρει την τιμή 69 λόγω του μεγέθους του *βήματος*, οπότε ορίζεται από τον χρήστη ο συνδυασμός προσφορών 50,50,50,69. Η νέα λύση που προκύπτει έχει την τιμή 17083 και είναι μεγαλύτερη από την λύση που βρέθηκε στα προηγούμενα υπολογιστικά πειράματα για *βήμα αλλαγής* 2,3,4,5,6,7,8,9,10 κατά 284 €, γεγονός το οποίο επιβεβαιώνει την συμβολή και αυτής της μεθόδου στην βελτίωση της τελικής λύσης.

Όσον αφορά το χρόνο εκτέλεσης του αλγορίθμου, με την παρούσα μέθοδο ο χρόνος καθίσταται αμελητέος καθώς ο αλγόριθμος εκτελείται μόνο μία φορά και ανέρχεται στα 0,007 δευτερόλεπτα.

➢ *Εφαρμογή στο δεύτερο παράδειγμα*

Στα πειράματα που εκτελέστηκαν για το δεύτερο παράδειγμα παρατηρείται ότι για την τέταρτη περίοδο η τιμή της προσφοράς έχει τάση προς την τιμή 60 όποτε της επιτρέπει το *βήμα αλλαγής* να πάρει αυτή την τιμή ενώ για την δεύτερη περίοδο η προσφορά τείνει προς την περιοχή τιμών 54 έως 56. Στη συνέχεια ορίζονται διάφοροι συνδυασμοί προσφορών που είναι κοντά σε αυτές τις τιμές και κοντά στον συνδυασμό που επιφέρει το μεγαλύτερο κέρδος, στην συγκεκριμένη περίπτωση το 58,54,66,60 για *βήμα* 2. Ένας από τους συνδυασμούς που προκύπτει βάσει της παραπάνω λογικής είναι και ο συνδυασμός προσφορών 58,55,67,60. Η νέα λύση που προκύπτει για αυτό τον συνδυασμό έχει την τιμή 29300 και ισούται με την τελική βέλτιστη λύση. Είναι μεγαλύτερη από την λύση που βρέθηκε στα προηγούμενα υπολογιστικά πειράματα για *βήμα αλλαγής* 2,3,4,5,6,7,8,9,10 κατά 700 €.

Επιπλέον, ο χρόνος εκτέλεσης του αλγορίθμου καθίσταται αμελητέος και για το δεύτερο παράδειγμα καθώς δεν χρειάστηκε πάνω από 10 επαναλήψεις για να βρεθεί ένας συνδυασμός προσφορών που να βελτιώνει την προϋπάρχουσα βέλτιστη λύση.

5.6.3 Επίλυση ως προς $H-1$ χρονικές περιόδους

Με τη χρήση αυτής της μεθόδου, αντί το πρόβλημα να λυθεί για H χρονικές περιόδους, λύνεται για μία λιγότερη $(H-1)$ κρατώντας σταθερή την προσφορά για την περίοδο που θα επιλεγεί από τον χρήστη του αλγορίθμου, με βάση κάποιας λογικής, σε μία τιμή η ο όποια επιλέγεται επίσης από τον χρήστη. Με τον τρόπο αυτό επιτυγχάνεται μείωση του χρόνου εκτέλεσης του αλγορίθμου σε σημαντικό βαθμό. Η λογική ως προς την οποία επιλέγεται η περίοδος για την οποία η προσφορά δεν θα μεταβάλλεται είναι να επιλέγεται η περίοδος αυτή για την οποία η προσφορά έχει την ίδια τιμή για την εκτέλεση πειραμάτων με εφαρμογή διαφορετικού *βήματος αλλαγής*.

➢ *Εφαρμογή στο πρώτο παράδειγμα*

Στα πειράματα που εκτελέστηκαν για το πρώτο παράδειγμα παρατηρείται ότι για την πρώτη περίοδο η τιμή της υποβάλλουσας προσφοράς παίρνει την τιμή 50 ανεξαρτήτως του βήματος αλλαγής. Έτσι, κρατώντας σταθερή την τιμή της πρώτης περιόδου στο 50 επιλύεται το πρόβλημα για μία λιγότερη περίοδο. Λόγω της μείωσης της διάστασης του προβλήματος δύναται να χρησιμοποιηθεί ως *βήμα αλλαγής* το 1, γεγονός το οποίο αυξάνει την ακρίβεια της λύσης.

Όσον αφορά τον χρόνο εκτέλεσης του αλγορίθμου με τη συγκεκριμένη μέθοδο για το συγκεκριμένο παράδειγμα είναι 146,7 λεπτά, δηλαδή 298 λεπτά λιγότερο από το πείραμα με *βήμα αλλαγής* 2. Για να επιλυθεί το παραπάνω πρόβλημα με *βήμα* 1 θα χρειαζόταν 6.765.201 επαναλήψεις (περίπου 4,7 μέρες) ενώ τώρα χρειάστηκαν μόλις 132.651 και η λύση που προκύπτει είναι καλύτερη κατά 284 €.

5.6.4 Επίλυση μέσω της εύρεσης τοπικών μέγιστων

Οι αγορές ενέργειας στα δύο ρεαλιστικά παραδείγματα που μελετήθηκαν παραπάνω αποτελούνταν από τέσσερεις παραγωγούς και τέσσερεις χρονικές περιόδους. Υπάρχουν περιπτώσεις όπου στις αγορές ενέργειας συμμετέχουν πάνω από οκτώ ή εννιά παραγωγικές μονάδες και οι χρονικές περίοδοι για τις οποίες πρέπει να υποβάλει προσφορές ο παραγωγός είναι 24, ίσες δηλαδή με τις ώρες μιας ημέρας. Στις περιπτώσεις αυτές, το μέγεθος του προβλήματος αυξάνεται κατά μεγάλο βαθμό και η επίλυση του με τον αλγόριθμο συνδυαστικής απαρίθμησης ή και των εναλλακτικών μεθόδων που αναφέρθηκαν παραπάνω καθίσταται αρκετά χρονοβόρα. Το γεγονός αυτό υπήρξε η αφορμή για τον σχεδιασμό της μεθόδου "*Επίλυσης μέσω της εύρεσης τοπικών μέγιστων*". Με τη μέθοδο αυτή, αρχικά όλες οι προσφορές του μεμονωμένου παραγωγού θέτονται ίσες με το μοναδιαίο κόστος $P^{\min}$. Στη συνέχεια επιλύεται το πρόβλημα για την πρώτη χρονική περίοδο και η προσφορά που θα προκύψει ως βέλτιστη ορίζεται ως η λύση για την πρώτη περίοδο και παραμένει σταθερή ενώ για όλες τις άλλες περιόδους οι προσφορές παραμένουν στο $P^{\min}$. Το πρόβλημα επιλύεται και για την επόμενη χρονική περίοδο και η προσφορά που θα προκύψει ως βέλτιστη ορίζεται ως η λύση για την περίοδο εκείνη και παραμένει και αυτή σταθερή. Με τον τρόπο αυτό επιλύεται το πρόβλημα μέχρι και για την τελευταία χρονική περίοδο. Η τελική λύση η οποία προκύπτει δεν είναι ολικά βέλτιστη, αποτελείται όμως από λύσεις που αποτελούν τοπικά μέγιστα για τα υποπροβλήματα που προέκυψαν από την επιμέρους επίλυση του κυρίου προβλήματος για κάθε χρονική περίοδο ξεχωριστά . Ένα τέτοιο παράδειγμα αγοράς ενέργειας με 9 παραγωγούς και 24 χρονικές περιόδους μελετήθηκε από τους Andrianesis et.al. (2012). Το μεταβλητό κόστος του μεμονωμένου παραγωγού είναι 49 €/MWh και το άνω όριο στην τιμή προσφοράς του είναι 100 €/MWh. Τα δεδομένα αυτού του παραδείγματος παρατίθενται στου Πίνακες 5.12, 5.13 και 5.14. Η μέθοδος αποζημίωσης που επιλέχτηκε είναι η *ΟΤΣ*.

Μονάδα (u)	$Q_u^{\min}$	$Q_u^{\max}$	SUC_u
1	2400	3800	1500000
2	240	377	13000
3	144	476	10000
4	155	550	25000
5	240	384	15000
6	65	151	18000
7	105	188	27000
8	120	287	50000
9	60	144	24000

Πίνακας 5.12 Δεδομένα Προβλήματος εννέα παραγωγών

Στον Πίνακα 5.13 απεικονίζονται οι προσφορές που υποβάλει ο κάθε παραγωγός για κάθε χρονική περίοδο h. Οι τιμές είναι σε €/MWh.

	Περίοδος ((h)																							
Μονάδα (u)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
1	--																							
2	49	51	53	55	57	60	58	56	54	52	51	50	49	51	53	55	57	60	58	56	54	52	51	50
3	52	54	56	58	60	62	60	58	54	50	54	59	52	54	56	58	60	62	60	58	54	50	54	59
4	55	57	59	61	63	65	64	63	62	61	60	56	55	57	59	61	63	65	64	63	62	61	60	56
5	57	59	61	63	65	67	69	68	64	62	68	70	57	59	61	63	65	67	69	68	64	62	68	70
6	64	66	68	70	72	74	76	75	73	71	65	64	64	66	68	70	72	74	76	75	73	71	65	64
7	65	65	65	65	65	65	67	69	70	67	71	66	65	65	65	65	65	65	67	69	70	67	71	66
8	70	72	74	76	80	82	84	86	75	74	73	72	70	72	74	76	80	82	84	86	75	74	73	72
9	72	74	76	78	80	82	84	86	73	74	75	72	72	74	76	78	80	82	84	86	73	74	75	72

Πίνακας 5.13 Προσφορές μονάδων παραγωγής για κάθε χρονική περίοδο

Στον Πίνακα 5.14 απεικονίζεται η ζήτηση για ενέργεια για κάθε χρονική περίοδο σε MWh.

	Περίοδος ((h)											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Ζήτηση (D_h)	4200	3900	3800	3700	3700	3600	4000	4300	4800	5200	5550	5500

	Περίοδος ((h)											
	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
Ζήτηση (D_h)	5450	5450	5300	5000	4950	4900	5000	5200	5100	5000	4800	4500

Πίνακας 5.14 Ζήτηση για ενέργεια σε κάθε χρονική περίοδο

Τα αποτελέσματα τα οποία προέκυψαν από την εκτέλεση του αλγορίθμου με την χρήση αυτή της μεθόδου για την αποζημίωση του μεμονωμένου παραγωγού παρατίθενται στου Πίνακες 5.15 και 5.16 αντίστοιχα.

	Περίοδος ((h)											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Λύσεις κάτω επιπέδου $(Q_{u,h})$	2413	2400	2552	2452	2452	2400	2400	2400	2674	3074	3424	3374
Λύσεις άνω επιπέδου ($P_{u,h}$ **)**	72	100	58	61	62	65	71	71	100	100	100	92

	Περίοδος ((h)											
	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
Λύσεις κάτω επιπέδου $(Q_{u,h})$	3324	3324	3174	2874	2824	2774	2874	3074	2974	2874	2674	2758
Λύσεις άνω επιπέδου ($P_{u,h}$ **)**	88	100	100	80	80	82	84	86	73	74	75	70

Πίνακας 5.15 Λύσεις κάτω και άνω επιπέδου με τη μέθοδο ΟΤΣ

Μέθοδος	Τιμή αντ/κής κάτω επιπέδου (f^*)	Τιμή αντ/κής άνω επιπέδου (F^*)	Χρόνος εκτέλεσης (minutes)
ΟΤΣ	9.857.050	2.101.230	5,01

Πίνακας 5.16 Τιμές αντ/κής συνάρτησης άνω και κάτω επιπέδου

Παρατηρείται ότι ο χρόνος εκτέλεσης του αλγορίθμου για ένα πρόβλημα τόσο μεγάλων διαστάσεων όσο αυτό το οποίο μελετάται είναι σχετικά μικρός και ανέρχεται στα 5,01 λεπτά. Αυτό καθιστά τη μέθοδο αυτή ως τη μοναδική που έχει τη δυνατότητα να λύσει προβλήματα τόσο μεγάλων διαστάσεων σε έναν εφικτό χρόνο. Βέβαια, η λύση η οποία προκύπτει δεν είναι η ολικά βέλτιστη αποτελεί όμως μια καλή προσέγγισή της.

Κεφάλαιο 6 Συμπεράσματα και Μελλοντική έρευνα

Ο αντικειμενικός σκοπός της παρούσας διπλωματικής εργασίας ήταν η ανάπτυξη ενός αλγορίθμου συνδυαστικής απαρίθμησης ο οποίος μπορεί να χρησιμοποιηθεί για την εύρεση του βέλτιστου συνδυασμού υποβολής προσφορών σε μια αγορά ηλεκτρικής ενέργειας ημερήσιου προγραμματισμού.

Αρχικά, στο πρώτο και δεύτερο κεφάλαιο έγινε μια εισαγωγή στις αγορές ενέργειας και στον τρόπο λειτουργίας τους καθώς επίσης και στον μικτό ακέραιο διεπίπεδο γραμμικό προγραμματισμό, ο οποίος αποτέλεσε και το υπόβαθρο για την ανάπτυξη του μαθηματικού μοντέλου στη συνέχεια.

Στο τρίτο κεφάλαιο διατυπώθηκε το πρόβλημα το οποίο μελετήθηκε και αναπτύχθηκε το μαθηματικό μοντέλο το οποίο το περιγράφει. Στη συνέχεια παρουσιάστηκαν οι δύο εναλλακτικοί μέθοδοι αποζημίωσης του μεμονωμένου παραγωγού και μελετήθηκε κατά πόσο επηρεάζουν την βέλτιστη λύση. Από την σύγκριση των δύο παραπάνω μεθόδων προέκυψε ότι η μέθοδος "*Αποζημίωση βάσει της Υποβάλλουσας Προσφοράς*" είναι ανεπαρκώς σχεδιασμένη καθώς επιτρέπει την πιθανή χειραγώγηση των τιμών από τους παραγωγούς και δίνει κίνητρο για υποβολή προσφορών υψηλότερων τιμών, γεγονός το οποίο επιβεβαιώνεται και από τα αποτελέσματα των πειραμάτων που εκτελέστηκαν με την χρήση του αλγορίθμου στη συνέχεια.

Στο πειραματικό κομμάτι της εργασίας, αρχικά παρουσιάστηκε και αναλύθηκε ο αλγόριθμος επίλυσης του μαθηματικού μοντέλου για περισσότερες από μία χρονικές περιόδους. Στη συνέχεια, ο αλγόριθμος εφαρμόστηκε σε δύο ρεαλιστικά παραδείγματα, τα αποτελέσματα των οποίων καταγράφηκαν και παρουσιάστηκαν στο πέμπτο κεφάλαιο.

Μια ενδιαφέρουσα παρατήρηση που προέκυψε από την εκτέλεση των πειραμάτων είναι ότι η αύξηση του βήματος με το οποίο αλλάζει η τιμή της προσφοράς του μεμονωμένου παραγωγού δεν οδηγεί απαραίτητα σε χειροτέρευση της λύσης όπως αρχικά αναμένονταν. Σε πολλές μάλιστα περιπτώσεις η λύση του προβλήματος με το μικρότερο δυνατό βήμα για το οποίο καθίσταται εφικτή η εκτέλεση του αλγορίθμου συνάπτει με την λύση του προβλήματος για βήμα τρείς και τέσσερεις φορές μεγαλύτερο. Αυτό συμβαίνει επειδή στα παραδείγματα που μελετήθηκαν υπήρχαν πολλαπλά βέλτιστα οπότε και η πιθανότητα να προκύψει ένας συνδυασμός που θα επιφέρει τη βέλτιστη λύση ήταν μεγαλύτερη. Από το γεγονός αυτό προκύπτει ότι αλγόριθμος μπορεί να εφαρμοστεί και να αποδώσει λύση ακόμα και σε προβλήματα μεγάλου μεγέθους ορίζοντας απλά ένα μεγαλύτερο βήμα χωρίς να υπάρχουν σημαντικές αποκλίσεις από τη βέλτιστη λύση.

Μια ακόμα ενδιαφέρουσα παρατήρηση είναι ότι με την εφαρμογή των εναλλακτικών μεθόδων εκτέλεσης του αλγορίθμου επετεύχθη βελτίωση της αρχικής λύσης και σε πολύ μικρότερο μάλιστα χρονικό διάστημα. Ενδεικτικά αναφέρεται ότι στο πρώτο παράδειγμα χρειάστηκε 456,2 λεπτά για να βρεθεί λύση με βήμα δύο, ενώ με την χρήσης της δεύτερης εναλλακτικής μεθόδου βρέθηκε καλύτερη λύση κατά 284 € σε 201,7 λεπτά με βήμα ένα.

Εν κατακλείδι, αντικείμενο μελλοντικής έρευνας θα μπορούσε να αποτελέσει η μελέτη της συμπεριφοράς των αποτελεσμάτων που θα προέκυπταν από την εφαρμογή του αλγορίθμου σε διαφορετικά παραδείγματα αγορών ενέργειας με διαφορετικά τεχνικά χαρακτηριστικά και σύγκριση με τη συμπεριφορά των αποτελεσμάτων που παρουσιάζονται στην παρούσα διπλωματική εργασία. Θα μπορούσε ακόμα να δοκιμαστεί και η απόδοση των εναλλακτικών μεθόδων εκτέλεσης του αλγορίθμου στα παραδείγματα αυτά τόσο ως προς το κέρδος του μεμονωμένου παραγωγού όσο και ως προς τον χρόνο εκτέλεσης που ενδεχομένως θα μειωνόταν.

Βιβλιογραφία

Kozanidis G, Kostarelou E, Andrianesis P, Liberopoulos G. (2011) “Mixed integer bilevel programming for optimal bidding strategies in day-ahead electricity markets with indivisibilities.” In 1st International Symposium & 10th Balkan Conference on Operational Research (BAL-COR), Sep 22–25; Thessaloniki, Greece. p. 8.

Andrianesis P, Liberopoulos G, Kozanidis G, Papalexopoulos (2012) “A. Recovery mechanisms in day-ahead electricity markets with non-convexities – part I: design and evaluation methodolgy.” IEEE Trans. Power Syst. 28:960–968.

H. v. Stackelberg, (1952) “English translation: The theory of market economy”, Marktform und Gleichgewicht, Springer Verlag, Berlin, 1934, Oxford University Press.

S. Dempe, S. Franke, (1991), “Bilevel Optimization Problems with Vectorvalued Objective Functions in Both Levels”, Paper by TU Bergakademie Freiberg, Germany.

Garcia-Martos, C, Rodriguez, J. and Sanchez, M.J. (2007), “Mixed models for short-run forecasting of electricity prices: Application for the Spanish market”, IEEE Transactions on Power Systems, 22(2), 544-552.

Ragupathi, R. and Das, T.K. (2004), “A stochastic game approach for modeling wholesale energy bidding in deregulated power markets”, IEEE Transactions on Power Systems, 19(2), 849-856.

Weber, J.D. and Overbye, T.J. (2002), “An individual welfare maximization algorithm for electricity markets”, IEEE Transactions on Power Systems, 17(3), 590-596.

Gountis, V.P. and Bakirtzis, A.G. (2004), “Bidding strategies for electricity producers in a competitive electricity marketplace”, IEEE Transactions on Power Systems, 19(1), 356-365.

Fampa, M., Barroso, L.A., Candal, D. and Simonetti, L. (2008), “Bilevel optimization applied to strategic pricing in competitive electricity markets”, Computational Optimization & Applications, 39, 121-142.

Pereira, M.V., Granville, S., Fampa, M.H.C., Dix, R. and Barroso, L.A. (2005), “Strategic bidding under uncertainty: A binary expansion approach”, IEEE Transactions on Power Systems, 20(1), 180-188.

Barroso, L.A., Carneiro, R.D., Granville, S., Pereira, M.V. and Fampa, M.H.C. (2006), “Nash equilibrium in strategic bidding: A binary expansion approach”, IEEE Transactions on Power Systems, 21(2), 629-638.

Bakirtzis, A.G., Ziogos, N.P., Tellidou, A.C. and Bakirtzis, G.A. (2007), “Electricity producer offering strategies in day-ahead energy market with step-wise offers”, IEEE Transactions on Power Systems, 22(4), 1804-1818.

Ruiz, C. and Conejo, A.J. (2009), “Pool strategy of a producer with endogenous formation of locational marginal prices”, IEEE Transactions on Power Systems, 24(4), 1855-1866.

Li, T., Shahidehpour, M. and Keyhani, A. (2004), “Market power analysis in electricity markets using supply function equilibrium model”, IMA Journal of Management Mathematics, 15, 339-354.

Hu, X. and Ralph, D. (2007), “Using EPECs to model bilevel games in restructured electricity markets with locational prices”, Operations Research, 55(5), 809-827.

Hobbs, B.F., Metzler, C.B. and Pang, J.-S. (2000), “Strategic gaming analysis for electric power systems: An MPEC approach”, IEEE Transactions on Power Systems, 15(2), 638-645.

Li, T., Shahidehpour, M. and Keyhani, A. (2004), “Market power analysis in electricity markets using supply function equilibrium model”, IMA Journal of Management Mathematics, 15, 339-354.

O’Neill RP, Sotkiewicz PM, Hobbs BF, Rothkopf MH, Stewart WR, Jr. (2005) “Efficient market-clearing prices in markets with nonconvexities.” Eur. J. Oper. Res, 164:269–285.

Andrianesis P, Liberopoulos G, Kozanidis G, Papalexopoulos A. (2012) “Recovery mechanisms in day-ahead electricity markets with non-convexities – part II: implementation and numerical evaluation.” IEEE Trans. Power Syst. 28:969–977.

Dempe S. (2002) “Foundations of bilevel programming.” New York (NY): Kluwer Academic;

ILOG AMPL CPLEX System Version 11.0 User’s Guide, 2008. [Online]. Available: http://www.ampl.com/BOOKLETS/am-plcplex100userguide.pdf.

Παράρτημα: Κώδικας αλγορίθμου στη C#

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
#include <ilcplex/ilocplex.h>


// important note: all arrays, matrices, etc. are 0-based
double   **P;
double   *Qmin;
double   *Qmax;
double   *D;
double   *SC;
double   *q1max;    //Decision variables qt for the 1st producer that gives
double   *P1max;    //Optimal bid price for 1st producer for each period
int      *count;
int      *step;
int      *c;

double   *q;
double   *z;
double   *y;
double   *SMP;     //SMP and PAB

int      Pmin;
int      Pmax;
double   Max;      //Maximum value of Sum[(SMP1t-Pmin)*q1t]
double   MaxSum;

int      N,T;      //Number of Producers, Number of Periods
int      i,t,w;    //counter of combinations (w) and counter of periods (t)
int      Maxm;     //number of combination that holds the final solution
int      pay;      //compensation method
int        print;    //selection to print results
double   a;        //how many numbers are altered in P1
int      vima;
int      k;        //how many numbers are altered in P1 if modulus=!0
double   W;        //number of combinations
int      c1;       //Pmin
double   objval;
double   objvalmax;
int      cntmax,cntmin;      //for SMP (case where q is both Qmin and Qmax)
int      Maxq,Minq,Minqs;
double   Maxq1,MaxP,Max1;

int      fix;
int      l;
int      counter;
int      sens;
double   Pfix;
int      v;
int      tnew/*=T*/;
int      Tnew;
int        u;
```

```
CPXENVptr   Env = NULL;  // CPLEX environment pointers
CPXLPptr    Lp = NULL;   // CPLEX problem pointers

void print_error(char *name)
{
   printf("%s failed, exiting...\n", name);
   printf("Press return to continue...\n");
   getchar();
}

bool CombinationGen()
{

    for(t=0; t<T; t++){
        if (t==(T-1)){
           if (count[t]==a) {
              count[t]=0;
           }
            P[0][t]=Pmin+count[t]*vima;
           if (P[0][t]>Pmax){
              P[0][t]=Pmax;
           }
           count[t]++;
        }

        else{
               if (count[t]==pow(a,T-t)-1){
                    count[t]=-1;
                    step[t]=0;
                }
                P[0][t]=Pmin+step[t]*vima;
                if (P[0][t]>Pmax){
                     P[0][t]=Pmax;
                }
                count[t]++;
                if (w==c[t]){
                     step[t]+=1;
                     c[t]=c[t]+pow(a,(T-t-1));
                }
        }
    }

    return true;
}
```

```
bool ChangeObjSense()
{

   CPXchgobjsen (Env, Lp, CPX_MIN);

   return true;
}

bool  init_env_lp()
{
   int status;

   Env = CPXopenCPLEX (&status);
   if (status != 0){
       print_error("CPXopenCPLEX");
       return false;
   }

   status = CPXsetintparam(Env, CPX_PARAM_DATACHECK, CPX_ON);
   if (status != 0){
      print_error("CPXsetintparam");
      return false;
   }

   return true;
}

bool  CreateProb()
{
   int status;

   Lp = CPXcreateprob (Env, &status, "iso");
   if (status != 0){
       print_error("CPXcreateprob");
       return false;
   }

   return true;
}

bool Solvprob()
{
      int status;
      status = CPXmipopt (Env, Lp);
      if (status != 0){
             printf("%d",status);
         print_error("CPXmipopt");
         return true;
      }
      return true;
}
```

```
bool PrintProblem ()
{
   int status;

   status = CPXwriteprob (Env, Lp, "prob.lp", NULL);
   if (status != 0){
      print_error("CPXwriteprob");
         return false;
   }
   return true;
}

bool NewCols (CPXCENVptr env, CPXLPptr lp, int ccnt, double const * obj,
double const * lb, double const * ub, char const * xctype, char ** colname)
{
   int status;
   status = CPXnewcols (env, lp, 1, obj, lb, ub, xctype, NULL);
   if (status != 0){
      print_error("CPXnewcols");
      return false;
   }
   return true;
}

bool AddDecVars()
{
   double obj, lb, ub;
   char xctype;

   lb = 0;
   ub = CPX_INFBOUND;

   xctype = 'C';         // continuous variables (Q), index = 0,...,N*T-1
   for (i = 0; i <= N-1; i++){      // for each unit
        for (t = 0; t <= T-1; t++){    // for each period
             obj =P[i][t];
             if (NewCols(Env, Lp, 1, &obj, &lb, &ub, &xctype, NULL) ==
false)
                  return false;
        }
   }

   xctype = 'B';  // binary variables (status), index = N*T,...,2*N*T-1
   obj = 0;       // status vars do not appear on the objective
   for (i = 0; i <= N-1; i++){      // for each unit
       for (t = 0; t <= T-1; t++){    // for each period
           if (NewCols(Env, Lp, 1, &obj, &lb, &ub, &xctype, NULL) == false)
               return false;
       }
   }

   xctype = 'B';  // binary variables (start-up), index = 2*N*T,...,3*N*T-1
   for (i = 0; i <= N-1; i++){      // for each unit
        obj = SC[i];
        for (t = 0; t <= T-1; t++){    // for each period
        if (NewCols(Env, Lp, 1, &obj, &lb, &ub, &xctype, NULL) == false)
            return false;
        }
   }
```

```
return true;
}

bool Addrows(CPXCENVptr env, CPXLPptr lp, int nzcnt, double const * rhs,
char const * sense, int const * rmatind, double const * rmatval, char **
rowname)
{
   int  status, rmatbeg = 0;

   status = CPXaddrows (env, lp, 0, 1, nzcnt, rhs, sense, &rmatbeg,
   rmatind, rmatval, NULL, rowname);
   if (status != 0){
      print_error("CPXaddrows");
      return false;
   }
   return true;
}

bool DemandConstraints()
{
   double rhs, *rmatval;
   char *rowname, sense;
   int nzcnt, *rmatind;


   rowname = (char*) malloc(5*sizeof(char));
   rmatval = (double*)malloc((T)*sizeof(double));
   rmatind = (int*)malloc((T)*sizeof(int));

   sense = 'E';

   for (t = 0; t <= T-1; t++){  // for each time period
         nzcnt = 0;
         rhs = D[t];
         sprintf(rowname, "D%d", t+1);
         for (i = 0; i <= N-1; i++){ // add quantities of all units
              rmatind[nzcnt] = t + i*T;
              rmatval[nzcnt] = 1;
              nzcnt++;
         }
         if (Addrows(Env, Lp, N, &rhs, &sense, rmatind, rmatval, &rowname)
== false)
          return false;

   }

   free(rowname);
   free(rmatval);
   free(rmatind);

   return true;
}
```

```
bool AddMinMax()
{
   double rhs, *rmatval;
   char *rowname, sense;
   int  *rmatind;


   rowname = (char*) malloc(10*sizeof(char));
   rmatval = (double*)malloc((2)*sizeof(double));
   rmatind = (int*)malloc((2)*sizeof(int));

   rhs = 0;
   sense = 'L';

   // technical minimum constraints
   for (i = 0; i <= N-1; i++){  // for each unit
         for (t = 0; t <= T-1; t++){  // for each period
              sprintf(rowname, "MIN%d%d", i+1, t+1);
              rmatind[0] = N*T + i*T + t;
              rmatval[0] = Qmin[i];
              rmatind[1] = i*T + t;
              rmatval[1] = -1;
              if (Addrows(Env, Lp, 2, &rhs, &sense, rmatind, rmatval,
&rowname) == false)
              return false;
         }
   }

   // technical maximum constraints
   sense = 'G';
   for (i = 0; i <= N-1; i++){ // for each unit
         for (t = 0; t <= T-1; t++){  // for each period
              sprintf(rowname, "MAX%d%d", i+1, t+1);
              rmatind[0] = N*T + i*T + t;
              rmatval[0] = Qmax[i];
              rmatind[1] = i*T + t;
              rmatval[1] = -1;
              if (Addrows(Env, Lp, 2, &rhs, &sense, rmatind, rmatval,
&rowname) == false)
              return false;
         }
   }

   free(rowname);
   free(rmatval);
   free(rmatind);

   return true;
}
```

```
bool startupConstraint()
{
   double rhs,rhs1, *rmatval;
   char *rowname, sense;
   int  *rmatind;

   rowname = (char*) malloc(10*sizeof(char));
   rmatval = (double*)malloc((3)*sizeof(double));
   rmatind = (int*)malloc((3)*sizeof(int));

   rhs = 0;
   rhs1 = -1;
   sense = 'G';

   // Start up constraint
   for (i = 0; i <= N-1; i++){  // for each unit
         for (t = 0; t <= T-1; t++){  // for each period
             if (t>0){
                 sprintf(rowname, "start%d%d", i+1, t+1);
                 rmatind[0] = N*T + i*T + t; //zit
                 rmatval[0] = -1;
                 rmatind[1] = 2*N*T + i*T + t; //yit
                 rmatval[1] = 1;
                 rmatind[2] = N*T + i*T + t-1; //zit-1
                 rmatval[2] = 1;
                 if (Addrows(Env, Lp, 3, &rhs, &sense, rmatind, rmatval,
&rowname) == false)
                 return false;
                 }
                 else {
                      sprintf(rowname, "start%d%d", i+1, t+1);
                       rmatind[0] = N*T + i*T + t; //zit
                       rmatval[0] = -1;
                      rmatind[1] = 2*N*T + i*T + t; //yit (zit-1 = 0 for t=0)
                      rmatval[1] = 1;
                if (Addrows(Env, Lp, 2, &rhs, &sense, rmatind, rmatval,
&rowname) == false)
                return false;
                }
        }
   }

   free(rowname);
   free(rmatval);
   free(rmatind);

   return true;
}
```

```
bool  PrintSolStatus()
{
   int    lpstat;
   int    status;

   status = CPXsolution (Env, Lp, &lpstat, &objval, NULL, NULL, NULL,
NULL);

   if (status != 0){
      print_error("CPXsolution");
      printf("%d",status);
      return false;
   }

   if(print==1){
      if (lpstat != CPXMIP_OPTIMAL && lpstat!=CPXMIP_OPTIMAL_TOL) {
          printf ("Solution not optimal!!\n");
       }

       printf ("\nSolution status:  %d\n\n", lpstat);
       printf ("Objective value:  %g\n\n", objval);
   }

}//end PrintSolStatus()

bool GetDecVars()
{
  int status;

  q=(double*)malloc((N*T)*sizeof(double));
  z=(double*)malloc((N*T)*sizeof(double));
  y=(double*)malloc((N*T)*sizeof(double));

  status = CPXgetx (Env, Lp, q, 0, N*T-1); //get q
  status = CPXgetx (Env, Lp, z, N*T, 2*N*T-1);// get z
  status = CPXgetx (Env, Lp, y, 2*N*T, 3*N*T-1); // get y

  printf ("\n#Loop= %d%s%g\n", w+1," of ",W);
  return true;
}
```

```c
bool Print()
{
  printf("%5s","\nTable P[i][t]: ""\n\n");

  for(i=0; i<N; i++) {
      for(t=0; t<T; t++) {
          printf("%1.1f  ",P[i][t]);
      }
      printf("\n");
  }
  printf ("\n");

  printf ("%5s","Decesion Variable qit: "); //print qit
  printf ("\n");

  for (i = 0; i <= N-1; i++){
        printf ("%5s%1d%2s"," Producer",i+1, "  ");
        for (t = 0; t <= T-1; t++){
            printf ("%1.0f  ",q[i*T+t]);
        }
        printf ("\n");
  }

  printf ("\n%5s","Decesion Variable zit: "); //print zit
  printf ("\n");

  for (i = 0; i <= N-1; i++){
        printf ("%5s%1d"," Producer",i+1);
        for (t = 0; t <= T-1; t++){
             printf ("  %1.0f  ",z[i*T+t]);
        }
        printf ("\n");
  }

  printf ("\n%5s","Decesion Variable yit: "); //print yit
  printf ("\n");

  for (i = 0; i <= N-1; i++){
         printf ("%5s%1d"," Producer",i+1);
         for (t = 0; t <= T-1; t++){
              printf ("  %1.0f  ",y[i*T+t]);
         }
         printf ("\n");
  }

  printf ("\n\n%5s","q1[t]=    "); //print q1t (qit for the first producer)


  for (t = 0; t <= T-1; t++){
      printf ("%2.0f ",q[t]);
  }
  printf ("\n");

  return true;

} //End Print
```

```
bool  PrintFinalSol()
{
  SMP=(double*)malloc((T)*sizeof(double));

  cntmin=0;
  cntmax=0;
  Maxq=0;
  Minq=1000000;
  Minqs=1000000;

  if (pay==1){
     if (print==1){
         printf ("\n%5s","SMP[t]=  "); //print SMP[t]
     }
     for (t = 0; t <= T-1; t++){
          SMP[t]=0;
         for (i = 0; i <= N-1; i++){
                if (q[i*T+t]<Qmax[i]-0.001 && q[i*T+t]>Qmin[i]+0.001){
                    SMP[t]=P[i][t];
                }
          }
                if (SMP[t]==0){
                 for (i = 0; i <= N-1; i++){
                  if (q[i*T+t]>=Qmax[i]-0.001 && q[i*T+t]<=Qmax[i]+0.001 ){
                      cntmax=1;
                       if (P[i][t]>=Maxq){
                           Maxq=P[i][t];
                           SMP[t]=P[i][t];
                        }
                  }
                 if (q[i*T+t]>=Qmin[i]-0.001 && q[i*T+t]<=Qmin[i]+0.001 ){
                      cntmin=1;
                      if (P[i][t]<=Minq){
                          Minq=P[i][t];
                          SMP[t]=P[i][t];
                      }
                 }
                }
                if (cntmin+cntmax==2){
                    for (i = 0; i <= N-1; i++){
                        if (q[i*T+t]<=Minqs && q[i*T+t]>0){
                            Minqs=q[i*T+t];
                            SMP[t]=P[i][t];
                        }
                    }
                 }
         }
         if (print==1){
             printf ("%2g ",SMP[t]);
                }
     }
     printf ("\n");
  }

  else{
         for (t = 0; t <= T-1; t++){
                SMP[t]=P[0][t];
          }
         if (print==1){
```

```
            printf ("\n%5s","PAB[t]=  "); //print PAB[t]
            for (t = 0; t <= T-1; t++){
                printf ("%2g ",SMP[t]);
            }
         }
         printf ("\n");
   }

   MaxSum=0;

   for (t = 0; t <= T-1; t++){
       MaxSum=MaxSum+(SMP[t]-c1)*q[t];
   }

   if (print==1){
      if (w==0){
        printf ("\n%2s%d%2s","MaxSum[for ",w+1,"st combination of P1]=  ");
      }
      else {
        printf ("\n%2s%d%2s","MaxSum[for ",w+1,"nd combination of P1]=  ");
      }
      printf ("%10.0f \n\n",MaxSum);
   }

   if (w==0){
        Max=MaxSum;
        Maxm=w;
        objvalmax=objval;
        printf ("\n%2s%2.0f\n","Max= ",Max);
        for (t = 0; t <= T-1; t++){ //store q1max[t]
            q1max[t]=q[t];
            P1max[t]=P[0][t];//optimal bid price for 1st producer
         }
   }

   else{
        if (MaxSum>Max){
            Max=MaxSum;
            Maxm=w;
            objvalmax=objval;
            printf ("\n%2s%g\n","Max= ",Max);
            for (t = 0; t <= T-1; t++){ //store q1max[t]
                q1max[t]=q[t];
                 P1max[t]=P[0][t];
            }
         }
   }

   if (w==W-1) {
       printf ("%2s\n\n", "--------------------------------------------------
------");
        printf ("\n%2s%g%2s%d%2s\n\n","TotalMaxSum= ",Max, " (for w=
",Maxm,") " );
        printf ("%2s","q1max[t]= ");
        for (t = 0; t <= T-1; t++){
             printf ("%2g ",q1max[t]);
        }
        printf ("\n\n");
        printf ("\n%2s\n\n","Optimal bid price for 1st producer: ");
```

```
        for (t = 0; t <= T-1; t++){
             printf ("%2.0f ",P1max[t]);
        }
        printf ("\n\n");
        printf("Maxobjval= %7g",objvalmax);
        printf ("\n\n");
  }

  if (MaxSum>Maxq1){
       Maxq1=MaxSum;
       MaxP=P[0][u];
  }

  free(q);
  free(z);
  free(y);
  free(SMP);

  return true;
} //end PrintFinalSol()

bool SensMenu()
{
   fix=0;
   //counter=-1;
   printf("Select one of the following options: \n\n");
   printf("--->Individual biding (Place manually the bids for each
period)\n");
   printf("    Type 0 to select\n\n");
   printf("--->Fix strategic Bids for T-1 periods\n");
   printf("    Type 1 to select\n\n");
   printf("--->Fix Strategic Bid for 1 period\n");
   printf("    Type 2 to select\n\n");
   printf("--->Use Andrianesis method\n");
   printf("    Type 3 to select\n\n");

   scanf("%d",&fix);

   if (fix==0){
       W=1;
   }
   else if (fix==1){
            W=a;
   }
   else if (fix==2){
            W=pow(a,T-1);
   }
   else if (fix==3){
            W=a*T;
   }

   return true;
}
```

```
bool SensAnalysis()
{
   if (fix==0){
       printf("\n");
        printf("Please enter the values of the strategic Bids for each
period (Enter a value and press enter each time)  \n");
        for (t=0;t<T;t++){
              scanf("%d",&l);
               P[0][t]=l;
        }
    }

      else if (fix==1){
              if (w==0){
                  counter=-1;
                  printf("\n");
                  printf("Please enter for which period the strategic Bids
will be shifting (Enter a value between 0 and T-1)  \n");
                  scanf("%d",&l);
              }
              for (t=0;t<T;t++){
                  P[0][t]=Pmin;
              }

              counter=counter++;
              P[0][l]=Pmin+(counter)*vima;
              if (P[0][l]>Pmax){
                  P[0][l]=Pmax;
              }
      }

      else if (fix==2){
               if (w==0){
                  printf("\n");
                  printf("Please enter for which period you wish the Bid to
be fixed (Enter a value between 0 and T-1)  \n");
                  scanf("%d",&l);
                  printf("Please enter the value of the bid that you wish
be kept fix \n");
                  scanf("%d",&v);
                  Pfix=v;
                  if (l==T-1){
                  Tnew=T-1;
                  tnew=0;
                  }
                  else if (l==0){
                           tnew=1;
                            Tnew=T;
                  }
                  else{
                        Tnew=T;
                        tnew=0;
                  }

                   for (t=tnew;t<Tnew;t++){
                         step[t]=0;
                         if (t==Tnew-1){
                             count[t]=0;
                         }
```

```
            else {
                count[t]=-1;
            }
            if (t<Tnew-2){
                c[t]=pow(a,(Tnew-t-1))-1;
            }
            else {
                  c[t]=a-1;
            }
    }

    if (l!=0 && l!=T){
        for (t=0;t<l;t++){
            c[t]=pow(a,(T-t-1-1))-1;
        }
    }
}

for (t=tnew;t<Tnew;t++){
    if (t!=l){
        if (t==(Tnew-1)){
            if (count[t]==a) {
                count[t]=0;
            }
            P[0][t]=Pmin+count[t]*vima;
            if (P[0][t]>Pmax){
                P[0][t]=Pmax;
            }
            count[t]++;
        }

        else{
            if (t<l){
                if (count[t]==pow(a,T/*new*/-t-1)-1){
                    count[t]=-1;
                    step[t]=0;
                    }
                }
                else{
                    if (count[t]==pow(a,T/*new*/-t)-1){
                        count[t]=-1;
                        step[t]=0;
                    }

                }

                P[0][t]=Pmin+step[t]*vima;
                if (P[0][t]>Pmax){
                    P[0][t]=Pmax;
                }
                count[t]++;
                if (w==c[t]){
                    step[t]+=1;
                    if (t<l){
                        c[t]=c[t]+pow(a,(T/*new*/-t-1-1));
                    }
                    else{
                        c[t]=c[t]+pow(a,(T/*new*/-t-1));
                    }
```

```
                        }
                    }

                }// end of if t=!l
            }// end of t=tnew; t<Tnew; t++
            P[0][l]=Pfix;
    }// enf of fix==2

    else if (fix==3) {
            if (w==0){
                u=0;
                counter=0;
                Maxq1=0;
                for (t=0;t<T;t++){
                    P[0][t]=Pmin;
                }
            }
            else {
                if(counter==a-1){
                    counter=-1;
                    P[0][u]=MaxP;
                    u++;
                }
                counter++;
                P[0][u]=Pmin+counter*vima;
                if (P[0][u]>Pmax){
                    P[0][u]=Pmax;
                }
            }

    } //end of fix==3

    return true;
}
```

```
int main()
{
   int start,stop;

   long double duration;

   int x;

   FILE *file;

   file = fopen("C:/Users/Xpert-pc/Desktop/cpp
files/Kozanidis/data_enumeration_problem.txt","r"); //import data from file

   if (file == NULL){
      printf("Error Reading File\n");
   }

   //Import data from file

   fscanf(file, " %d",&N);

   fscanf(file, " %d",&T);

   Qmin=(double*)malloc(N*sizeof(double));

   Qmax=(double*)malloc(N*sizeof(double));

   D=(double*)malloc(T*sizeof(double));

   SC=(double*)malloc(N*sizeof(double));

   count=(int*)malloc(T*sizeof(int));

   step=(int*)malloc(T*sizeof(int));

   c=(int*)malloc(T*sizeof(int));

   q1max=(double*)malloc(T*sizeof(double));

   P1max=(double*)malloc(T*sizeof(double));

   P=(double**)malloc(N*sizeof(double*));
   for (i=0; i<N; i++){
        P[i]=(double*)malloc(T*sizeof(double));
   }

   for(i=1; i<N; i++) {
       for(t=0; t<T; t++) {
            fscanf(file, "%d",&x);
             P[i][t]=x;
       }
   }

   for(i=0; i<N; i++) {
        fscanf(file, "%d",&x);
         Qmin[i]=x;
   }

   for(i=0; i<N; i++) {
```

```
        fscanf(file, " %d",&x);
         Qmax[i]=x;
    }

    for(t=0; t<T; t++) {
        fscanf(file, " %d",&x);
         D[t]=x;
    }

    for(i=0; i<N; i++) {
        fscanf(file, " %d",&x);
         SC[i]=x;
    }

    fscanf(file, " %d",&x);
    Pmin=x;
    fscanf(file, " %d",&x);
    Pmax=x;
    c1=Pmin;

    printf("Please Enter the pace in which the bids of the strategic
producer will be shifting (vima) \n");
    scanf("%d",&vima);

    printf("Please Enter Compensation Method\n");

    printf("Type 0 for ''Pay as Bid'' or 1 for ''System Marginal
Pricing'':\n");
    scanf("%d",&pay);

    printf("Do you want to print solutions for each repetition?\n");

    printf("Type 0 for ''No'' or 1 for ''Yes'' (No printing saves 20 percent
time):\n");
    scanf("%d",&print);

    a=((int)((Pmax-Pmin)/vima)+1);// # of possible values that P[0][t] can
take

    k=(Pmax-Pmin)%vima;//modulus of the devision

    if (k>0){
       a=((int)((Pmax-Pmin)/vima)+2); // # of possible values that the
strategic Bids can take if the modulus of the devision (Pmax-Pmin)/vima is
>0
    }

    W=pow(a,T);

    printf("Do you want to enter Sensitivity Analysis Menu?\n");
    printf("Type 0 for ''No'' or 1 for ''Yes'':\n");
    scanf("%d",&sens);

    if (sens==1){SensMenu();}

    double b=(W/1000)*0.99;

    printf("W (Total repetitions) = %2g ",W);
    printf("\n\n");
```

```
    if (b>1){
        printf("Estimated Execution Time: %f%s\n",b," minutes");
        printf("\n\n");
    }

    else{
         printf("Estimated Execution Time: %f%s\n",b*60," seconds");
         printf("\n\n");
    }

    for (t=0;t<T;t++){
           step[t]=0;
           if (t==T-1){
              count[t]=0;
           }
           else {
           count[t]=-1;
           }
           if (t<T-2){
            c[t]=pow(a,(T-t-1))-1;
           }
           else {
                c[t]=a-1;
           }
    }

    if (init_env_lp() == false) exit;
    if (ChangeObjSense() == false) exit;

    start=clock();

    for (w = 0; w <  W; w++){

         if (sens!=1){
              CombinationGen();//Creation of all possible combinations for the
bids of the Strategic Producer
            }
            else{
                 SensAnalysis();
            }
            if (CreateProb() == false) exit;
            if (AddDecVars() == false) exit;
             if (DemandConstraints() == false) exit;
             if (AddMinMax() == false) exit;
             if (startupConstraint() == false) exit;
             if (PrintProblem() == false) exit;
             if (Solvprob() == false) exit;
            if (GetDecVars() == false) exit;
            if (PrintSolStatus() == false) exit;
            if (print==1){
                if (Print() == false) exit;
            }
           if (PrintFinalSol() == false) exit;

           CPXfreeprob (Env, &Lp);

    }// end of repetition
```

```
    stop=clock();

    duration=(long double) ((stop-start)/CLOCKS_PER_SEC)/60;

    if (duration>1){
        printf("time= %Lf%s\n",duration,"min");
    }

    else{
         printf("time= %Lf%s\n",duration*60,"sec");
    }

    CPXcloseCPLEX (&Env);

    free(q1max);
    free(P1max);
    free(count);
    free(step);
    free (c);
    free(Qmin);
    free(Qmax);
    free(SC);
    free(P);
    free(D);

    system ("pause");
    return 0;

} //end main
```