ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Μεταπτυχιακή Εργασία

Εύρεση βέλτιστων λύσεων σε αγορές ημερήσιου προγραμματισμού ηλεκτρικής ενέργειας μέσω της εφαρμογής συνθηκών βελτιστότητας

υπό

ΚΑΪΜΑΚΑΜΗ ΧΡΗΣΤΟΥ

Διπλωματούχου Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών Δ.Π.Θ,

2014

ΥΦΑΝΤΗ ΛΑΜΠΡΟΥ

Διπλωματούχου Πολιτικού Μηχανικού Π.Θ, 2015

2016

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

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

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του Μεταπτυχιακού Προγράμματος Σπουδών «Σύγχρονες Μέθοδοι Σχεδιασμού και Ανάλυσης στη Βιομηχανία» του Τμήματος Μηχανολόγων Μηχανικών στην κατεύθυνση της Οργάνωσης Παραγωγής και Βιομηχανικής Διοίκησης από τους μεταπτυχιακούς φοιτητές Καϊμακάμη Χρήστο και Υφαντή Λάμπρο.

Αρχικά, θα θέλαμε να ευχαριστήσουμε θερμά τον επιβλέποντα Καθηγητή κ. Γεώργιο Κοζανίδη για την παρότρυνσή του να ασχοληθούμε με το συγκεκριμένο θέμα, την άριστη συνεργασία μας και την πολύτιμη καθοδήγησή του σε κάθε στάδιο της παρούσας Διπλωματικής εργασίας. Επίσης, θα θέλαμε να εκφράσουμε τις ευχαριστίες μας στους Καθηγητές κ. Γεώργιο Λυμπερόπουλο και κ. Γεώργιο Σαχαρίδη που δέχτηκαν να είναι μέλη της τριμελούς επιτροπής αξιολόγησης της παρούσας.

Τέλος, θα θέλαμε να ευχαριστήσουμε όλους αυτούς τους υπέροχους ανθρώπους που ήταν καθημερινά γύρω μας και κατέστησαν τόσο ενδιαφέρουσα την όλη εμπειρία των μεταπτυχιακών σπουδών αλλά περισσότερο απ' όλους τις οικογένειές μας για την υποστήριξή και τη βοήθειά τους στη συνολική πορεία μας έως τώρα.

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η διερεύνηση και τελικά η εύρεση της βέλτιστης προσφοράς που δύναται να υποβληθεί από έναν παραγωγό που συμμετέχει σε μια αγορά ημερήσιου προγραμματισμού ηλεκτρικής ενέργειας σε κάθε περίοδο λειτουργίας. Βασική θεώρηση για την ανάλυση του συγκεκριμένου προβλήματος, αποτελεί το γεγονός ότι ο παραγωγός έχει πλήρη επίγνωση των παραμέτρων της αγοράς, όπως της ζήτησης σε ενέργεια και των τιμών/προσφορών που έχουν κατατεθεί από τους υπόλοιπους παραγωγούς.

Το πρόβλημα μορφοποιείται ως ένα μεικτό ακέραιο μοντέλο διεπίπεδου (bilevel) προγραμματισμού. Στο ανώτερο επίπεδο, ο παραγωγός προσπαθεί να μεγιστοποιήσει το ατομικό του κέρδος, ενώ στο κατώτερο ένας ανεξάρτητος διαχειριστής του συστήματος (Independent System Operator) είναι υπεύθυνος για την εκκαθάριση της αγοράς και τον καθορισμό της ποσότητας ενέργειας που εν τέλει θα προσφερθεί από κάθε συμμετέχοντα παραγωγό, με στόχο την ικανοποίηση της ζήτησης σε ενέργεια στο ελάχιστο συνολικό δυνατό κόστος βάσει των υποβληθεισών προσφορών. Το μοντέλο, το οποίο απαγορεύει την εφαρμογή παραδοσιακών μεθοδολογιών για την επίλυση του προβλήματος όπως είναι η αντικατάσταση του κάτω προβλήματος με τις πρώτης τάξης συνθήκες βελτιστότητάς του, χρησιμοποιεί διακριτές μεταβλητές για να αναπαραστήσει την κατάσταση λειτουργίας των μονάδων.

Η ανάπτυξη ενός ακριβούς αλγόριθμου κρίθηκε αναγκαία για την εύρεση της βέλτιστης λύσης του προβλήματος, και πιο συγκεκριμένα για την περίπτωση κατά την οποία οι συμμετέχοντες παραγωγοί θα αποζημιωθούν με την οριακή τιμή του συστήματος (smp), χρησιμοποιώντας συγκεκριμένους κανόνες βελτιστότητας που διέπουν τη συγκεκριμένη αγορά. Ο αλγόριθμος αυτός συμβάλλει σημαντικά στη μείωση του υπολογιστικού χρόνου επίλυσης σε σχέση με τα υφιστάμενα μοντέλα, γεγονός που θα επαληθευτεί και από την εφαρμογή του αλγορίθμου σε ποικίλες περιπτώσεις με διαφορετικά δεδομένα και παραμέτρους.

Περιεχόμενα

Κατάλογος Πινάκων

1. Εισαγωγή

Σε αυτό το κεφάλαιο, παρουσιάζονται πληροφορίες εισαγωγικού χαρακτήρα, που στόχο έχουν να διαμορφώσουν στον αναγνώστη μία γενική αλλά επαρκή εικόνα του αντικειμένου που θίγει η συγκεκριμένη μεταπτυχιακή εργασία, παραθέτοντας μια ανασκόπηση της σχετικής με την εργασία βιβλιογραφίας και περιγράφοντας συνοπτικά τις βασικές ενότητες της εργασίας.

1.1 Περιγραφή του γενικού προβλήματος

Τα μοντέλα μαθηματικού προγραμματισμού δύο επιπέδων προκύπτουν στο πλαίσιο εφαρμογής διαφόρων διεπιστημονικών πεδίων, όπως ο γεωργικός σχεδιασμός, η χάραξη της πολιτικής της κυβέρνησης, ο οικονομικός προγραμματισμός, η χρηματοοικονομική διαχείριση, η βελτιστοποίηση πολεμικών επιχειρήσεων, ο σχεδιασμός των μεταφορών, η βέλτιστη τιμολόγηση, ο οικολογικός προγραμματισμός, ο χημικός σχεδιασμός, ο προγραμματισμός της παραγωγής, η βέλτιστη κατανομή των πόρων, όπως και σε προβλήματα εγκατάστασης διοδίων, κατασκευαστικά προβλήματα και σε στρατιωτικές εφαρμογές (Dempe [1]). Αυτή η ευρεία εφαρμογή σε συνδυασμό με τη δυσκολία επίλυσης που έχουν τα προγράμματα δύο επιπέδων, έχει παρακινήσει τους ερευνητές να αναπτύξουν εξειδικευμένες αλγοριθμικές μεθόδους για την επίλυσή τους. Αν και αυτό κατέστησε τη σχετική περιοχή έρευνας πολύ ενεργή, καμία από τις γενικές μεθοδολογίες επίλυσης που έχουν αναπτυχθεί μέχρι σήμερα δεν είναι σε θέση να εξυπηρετήσει τα μοντέλα προγραμματισμού δύο επιπέδων.

Η απελευθέρωση των αγορών ηλεκτρικής ενέργειας αποτελεί μια σημαντική οικονομική εξέλιξη που έχει συμβεί σε πολλές χώρες παγκοσμίως τα τελευταία χρόνια. Παρότι οι συγκεκριμένοι σχεδιασμοί των αγορών που έχουν υιοθετηθεί από διάφορες χώρες ποικίλουν σημαντικά, πολλές από τις βασικές αρχές παραμένουν ως επί το πλείστον οι ίδιες. Οι περισσότεροι σχεδιασμοί έχουν θεσπίσει μία χονδρική και μία λιανική αγορά ηλεκτρικής ενέργειας που λειτουργούν σε μακροπρόθεσμους και βραχυπρόθεσμους χρονικούς ορίζοντες.

Γενικά, κάθε μονάδα που συμμετέχει χαρακτηρίζεται από το τεχνικό της ελάχιστο και μέγιστο και από τα κόστη εκκίνησης και παραγωγής, και καλείται να υποβάλλει μια προσφορά για την ποσότητα ενέργειας που θα προσφέρει στο σύστημα. Στο επίπεδο του ημερήσιου προγραμματισμού της χονδρικής αγοράς ηλεκτρικής ενέργειας, οι παραγωγοί υποβάλλουν ελεύθερα προσφορές (οι οποίες συνήθως υπόκεινται σε κάποιο κατώτατο και ανώτατο όριο) για την παραγωγή ενέργειας. Με τα τεχνικά χαρακτηριστικά και τις προσφορές των μονάδων γνωστά, ένας ανεξάρτητος διαχειριστής του συστήματος (ISO), κάνει εκκαθάριση της αγοράς κατανέμοντας ποσότητες ενέργειας στους συμμετέχοντες παραγωγούς, έτσι ώστε να ελαχιστοποιηθεί το συνολικό κόστος που απαιτείται για την ικανοποίηση της ζήτησης για ενέργεια, βάσει των υποβληθεισών προσφορών.

Στην παρούσα εργασία, εξετάζεται το πρόβλημα από την σκοπιά ενός στρατηγικού παραγωγού που συμμετέχει σε μια αγορά προγραμματισμού ηλεκτρικής ενέργειας διαφόρων περιόδων λειτουργίας, στην οποία η εκκίνηση λειτουργίας και η ποσότητα ενέργειας των μονάδων παραγωγής καθορίζονται από έναν ISO. Υποθέτοντας ότι ο παραγωγός αυτός έχει πλήρη γνώση των παραμέτρων της αγοράς (της ζήτησης για ενέργεια και των προσφορών/στοιχείων-κόστους όλων των άλλων παραγωγών), μελετάται το πρόβλημα της επιλογής της βέλτιστης τιμής-προσφοράς του στρατηγικού παραγωγού για κάθε μονάδα ενέργειας που παρέχει στο σύστημα. Εδώ, η λέξη βέλτιστη αφορά το γεγονός ότι μετά την εκκαθάριση της αγοράς από τον ISO, το κέρδος του στρατηγικού παραγωγού πρέπει να είναι το μέγιστο δυνατό.

Το πρόβλημα διατυπώνεται ως ένα μεικτό ακέραιο μοντέλο βελτιστοποίησης δύο επιπέδων, με τον στρατηγικό παραγωγό να μεγιστοποιεί το κέρδος του στο ανώτερο επίπεδο, και τον ISO να κάνει εκκαθάριση της αγοράς με το ελάχιστο συνολικό κόστος του συστήματος, βάσει των υποβληθεισών προσφορών, στο κατώτερο επίπεδο. Αξιοποιώντας σημαντικά ευρήματα από τη θεωρία του ακέραιου παραμετρικού προγραμματισμού, αναπτύσσεται ένας αποτελεσματικός αλγόριθμος που βρίσκει την ολικά βέλτιστη λύση του προβλήματος. Η συμβολή της παρούσας εργασίας είναι διττή. Από τη μία, μπορεί να βοηθήσει τους στρατηγικούς παραγωγούς στην ανάπτυξη προσφορών προς υποβολή που θα μεγιστοποιήσουν το ατομικό τους κέρδος. Από την άλλη, επιτρέπει στους εκάστοτε ISO την ανίχνευση πιθανών αθέμιτων μεθόδων χειραγώγησης των τιμών εκκαθάρισης της αγοράς από τους μεμονωμένους παραγωγούς και την κατασκευή κανόνων που θα τους αποτρέψουν.

Βασικές παράμετροι που διέπουν το πρόβλημα μίας αγοράς ημερήσιου προγραμματισμού ηλεκτρικής ενέργειας είναι οι εξής:

1. Η τιμή/προσφορά (bid) των παραγωγών (εκτός του στρατηγικού), κάθε παραγωγός υποβάλλει μία συγκεκριμένη προσφορά στον ISO, η οποία αντικατοπτρίζει και το κέρδος ανά MWh που θα έχει στην περίπτωση που συμπεριλαμβάνεται τελικά στη λύση του προβλήματος
2. Τα τεχνικά ελάχιστα και μέγιστα των ποσοτήτων σε MWh που δύναται να παράγει κάθε παραγωγός
3. Το μεταβλητό κόστος του κάθε παραγωγού για κάθε MWh που μπορεί να παράξει, το οποίο για ευνόητους λόγους αποτελεί και το κατώτατο όριο της προσφοράς που επιτρέπεται να καταθέσει ο κάθε παραγωγός
4. Το ανώτατο επιτρεπτό όριο τιμή/προσφοράς (price cap) για κάθε παραγωγό
5. Η συνολική ζήτηση σε MWh

Η αδυναμία του στρατηγικού παραγωγού να καθοδηγήσει τον ανεξάρτητο διαχειριστή του συστήματος (ISO) στην πιο συμφέρουσα λύση για αυτόν κατέστησε αναγκαία την εύρεση μίας αλγοριθμικής μεθόδου, σύμφωνα με την οποία ο στρατηγικός παραγωγός επιλέγει τόσο τη βέλτιστη ποσότητα που επιθυμεί να παράξει όσο και τη βέλτιστη αποζημίωση που επιθυμεί να έχει για κάθε MWh. Στη συνέχεια, λαμβάνοντας υπόψη τις παραμέτρους του προβλήματος και συγκεκριμένους κανόνες που διέπουν το συγκεκριμένο είδος αγοράς, ο ανεξάρτητος διαχειριστής του συστήματος χρησιμοποιεί τα δεδομένα που υποβάλλει ο στρατηγικός παραγωγός και εξάγει πιθανούς συνδυασμούς παραγωγών, ελαχιστοποιώντας το κόστος της παραγωγής και ικανοποιώντας τη συνολική ζήτηση σε ηλεκτρική ενέργεια. Η μορφοποίηση του αλγορίθμου είναι τέτοια ώστε, σε περίπτωση ύπαρξης πιθανών λύσεων, ο στρατηγικός παραγωγός θα συμμετέχει σε αυτές και οι λύσεις καθίστανται βέλτιστες και για τα 2 επίπεδα του προβλήματος.

1.2 Βιβλιογραφική Ανασκόπηση

Ένα σημαντικό κομμάτι της σχετικής έρευνας έχει αντιμετωπίσει το πρόβλημα της ανάπτυξης βέλτιστων στρατηγικών υποβολής προσφορών για παραγωγούς ενέργειας οι οποίοι συμμετέχουν σε μία αγορά ημερήσιου προγραμματισμού ηλεκτρικής ενέργειας. Πολλές από τις δημοσιευμένες εργασίες (π.χ., Garcia-Martos et al. [2]) 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 [3]) έχουν αναπτύξει στοχαστικά μοντέλα προγραμματισμού, προκειμένου να αντιμετωπίσουν τις αβεβαιότητες που παρουσιάζουν ορισμένες από τις παραμέτρους του προβλήματος. Στη σύντομη επισκόπηση της βιβλιογραφίας που ακολουθεί, παρατίθενται διεπίπεδα μοντέλα προγραμματισμού που έχουν αναπτυχθεί για βέλτιστη στρατηγική υποβολής προσφορών σε αγορές ηλεκτρικής ενέργειας, καθώς οι συγκεκριμένες εργασίες συνδέονται πιο στενά με την παρούσα.

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 [4] ανέπτυξαν ένα διεπίπεδο μοντέλο βελτιστοποίησης που μεγιστοποιεί την ευημερία 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 [5] και Fampa et al. [6] ανέπτυξαν διεπίπεδα στοχαστικά μοντέλα βελτιστοποίησης για το ίδιο πρόβλημα. Στην πρώτη εργασία, οι συγγραφείς χρησιμοποίησαν προσομοίωση Monte-Carlo και γενετικούς αλγόριθμους για την εύρεση της βέλτιστης λύσης, ενώ στη δεύτερη, οι συγγραφείς χρησιμοποίησαν μία ευρετική διαδικασία και μία μεικτή ακέραια αναδιατύπωση του προβλήματος. Μεταξύ άλλων, οι Pereira et al. [7], Barroso et al. [8], Bakirtzis et al. [9] και Ruiz and Conejo [10] 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. Τα μη γραμμικά προβλήματα μετατράπηκαν στη συνέχεια σε μεικτά ακέραια γραμμικά προβλήματα μέσω κατάλληλων μορφοποιήσεων και επιλύθηκαν μέσω γενικών λογισμικών βελτιστοποίησης. Οι Li et al. [11] ανέπτυξαν επίσης ένα διεπίπεδο μοντέλο βελτιστοποίησης και το χρησιμοποίησαν για την αναζήτηση σημείων ισορροπίας Nash. Οι Hu and Ralph [12] 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. [13] ανέπτυξαν ένα διεπίπεδο μοντέλο βελτιστοποίησης και χρησιμοποίησαν έναν αλγόριθμο ποινής εσωτερικών σημείων (penalty interior point algorithm) για την επίλυσή του. Οι Li and Shahidehpour [14] ανέπτυξαν ένα διεπίπεδο μοντέλο βελτιστοποίησης και χρησιμοποίησαν συναρτήσεις ευαισθησίας και μια πρωτεύουσα-δυική μεθοδολογία εσωτερικών σημείων (primal-dual interior point method) προκειμένου να το επιλύσουν. Οι Ma et al. [15] και Zhang et al. [16] ανέπτυξαν επίσης διεπίπεδα μοντέλα προγραμματισμού, και πρότειναν τεχνικές τύπου βελτιστοποίησης σμήνους σωματιδίων (particle swarm optimization) για την επίλυσή τους. Οι Badri et al. [17] ανέπτυξαν ένα ακόμη διεπίπεδο μοντέλο βελτιστοποίησης, το οποίο λαμβάνει υπόψη του διμερείς συμβάσεις και περιορισμούς μεταφοράς, και το έλυσαν μέσω μιας πρωτεύουσας-δυικής μεθοδολογίας εσωτερικών σημείων. Οι Vahidinasab και Jadid [18] ανέπτυξαν ένα μοντέλο βελτιστοποίησης που εξετάζει πολλαπλούς αντικειμενικούς στόχους. Μετά τη χρησιμοποίηση της μεθόδου μειωμένης εφικτής περιοχής ε-constraint (ε-constraint reduced feasible region) για την αντιμετώπιση των πολλαπλών στόχων, αντικατέστησαν το πρόβλημα του κάτω επιπέδου με τις πρώτης τάξης συνθήκες βελτιστότητάς του και έλυσαν το πρόβλημα που προέκυψε με γενικό λογισμικό βελτιστοποίησης. Οι Gabriel and Leuthold [19] παρουσίασαν μια διεπίπεδη μαθηματική μορφοποίηση και χρησιμοποιώντας διαζευκτικούς (disjunctive) περιορισμούς και γραμμικοποίηση την αναδιατύπωσαν ως ένα μεικτό ακέραιο γραμμικό πρόβλημα, το οποίο στη συνέχεια έλυσαν με γενικό λογισμικό βελτιστοποίησης.

Όπως φαίνεται παραπάνω, οι περισσότεροι συγγραφείς που αναπτύσσουν διεπίπεδα μοντέλα βελτιστοποίησης για το πρόβλημα που εξετάζουμε χρησιμοποιούν είτε μία κατάλληλη αναδιατύπωσή του και γενικό λογισμικό βελτιστοποίησης για την επίλυσή του, είτε μία ευρετική διαδικασία επίλυσης. Στην παρούσα εργασία, υιοθετείται μια ελαφρώς διαφορετική προσέγγιση. Πρώτα απ' όλα, χρησιμοποιούνται δυαδικές μεταβλητές για τη μοντελοποίηση της εκκίνησης των μονάδων παραγωγής ενέργειας. Αυτή η επιλογή, σε συνδυασμό με το γεγονός ότι επιβάλλονται αυστηρώς θετικά κάτω όρια για τις ποσότητες ενέργειας που οι μονάδες αυτές προσφέρουν αν συμμετάσχουν στην αγορά, προσδίδει ισχυρά συνδυαστικές ιδιότητες στο μοντέλο, οι οποίες απαγορεύουν τη χρήση συνθηκών βελτιστότητας πρώτης τάξης για την απλοποίηση της μορφοποίησής του. Αντ' αυτού, αναπτύσσεται μια αλγοριθμική μεθοδολογία, η οποία χρησιμοποιεί σημαντικά αποτελέσματα από τη θεωρία του μεικτού ακέραιου παραμετρικού προγραμματισμού, και είναι σε θέση να βρει την ολικά βέλτιστη λύση του προβλήματος, όταν αυτό τίθεται για περισσότερες από μία περιόδους λειτουργίας

2. Μορφοποίηση του Μαθηματικού Μοντέλου

Ο προγραμματισμός πολλαπλών επιπέδων αποτελεί έναν ειδικό κλάδο του μαθηματικού προγραμματισμού που ασχολείται με προγράμματα των οποίων το εφικτό σύνολο λύσεων καθορίζεται από μία σειρά ένθετων προβλημάτων βελτιστοποίησης. Η πιο μελετημένη περίπτωση είναι η περίπτωση των μοντέλων δύο επιπέδων, όπου ένα υποσύνολο των μεταβλητών απόφασης του άνω επιπέδου πρέπει να είναι η βέλτιστη λύση του ένθετου μαθηματικού προβλήματος (κάτω επίπεδο). Το πρόβλημα μπορεί να θεωρηθεί ως ένα παιχνίδι δύο ατόμων με τους δύο ιθύνοντες λήψης αποφάσεων να παίρνουν τις αποφάσεις ιεραρχικά. Ο πρώτος ιθύνων λήψης αποφάσεων (που αναφέρεται ως ηγέτης) ελέγχει ένα υποσύνολο των μεταβλητών απόφασης του προβλήματος, προσπαθώντας να λύσει ένα πρόβλημα βελτιστοποίησης που περιλαμβάνει στο σύνολο των περιορισμών του ένα δεύτερο πρόβλημα βελτιστοποίησης που επιλύεται από τον δεύτερο ιθύνοντα λήψης αποφάσεων (που αναφέρεται ως ακόλουθος), ο οποίος ελέγχει τις υπόλοιπες μεταβλητές απόφασης. Σε γενικές γραμμές, ένα πρόγραμμα δύο επιπέδων είναι μη-κυρτό, και η εξεύρεση του συνολικού βέλτιστου είναι ένα πολύ δύσκολο έργο.

Αρχικά, θεωρείται ένα σύνολο μονάδων παραγωγής που συμμετέχουν σε μια αγορά προγραμματισμού ηλεκτρικής ενέργειας πολλών περιόδων λειτουργίας (σε ημερήσιο, εβδομαδιαίο και άλλο ορίζοντα-βάση λειτουργίας). Οι παραγωγοί ενέργειας υποβάλλουν τιμές-προσφορές (bids) για κάθε περίοδο του προγραμματισμένου ορίζοντα λειτουργίας σε έναν ανεξάρτητο διαχειριστή του συστήματος (ISO, δεύτερος ιθύνων λήψης αποφάσεων - ακόλουθος) που κάνει εκκαθάριση της αγοράς και καθορίζει την εκκίνηση και την ποσότητα ενέργειας της κάθε μονάδας παραγωγής, διασφαλίζοντας ότι η συνολική ζήτηση ενέργειας ικανοποιείται με το ελάχιστο συνολικό κόστος για το σύστημα, βάσει των υποβληθεισών προσφορών. Τα τεχνικά χαρακτηριστικά της κάθε μονάδας παραγωγής (ελάχιστο και μέγιστο παραγωγής), καθώς και το κόστος εκκίνησης είναι σταθερά και γνωστά στον ISO. Αφού προσδιοριστεί η ποσότητα ενέργειας για κάθε παραγωγό, υιοθετείται ένα σύστημα πληρωμών εκκαθάρισης που αποζημιώνει κάθε συμμετέχοντα παραγωγό, με την πλήρη καταβολή του κόστους εκκίνησής του και την τιμή εκκαθάρισης της αγοράς για κάθε MWh που αυτός παρέχει στο σύστημα.

Η τιμή εκκαθάρισης της αγοράς μπορεί να είναι σύμφωνα με έναν πρώτο τρόπο αποζημίωσης η ακριβής τιμή προσφοράς που υποβάλλεται από τον αντίστοιχο παραγωγό (pay-as-bid) ή η ίδια για όλους τους παραγωγούς, σύμφωνα με έναν δεύτερο τρόπο αποζημίωσης με βάση την οριακή τιμή του συστήματος (uniform – System Marginal Price). Στην δεύτερη περίπτωση, η τιμή εκκαθάρισης είναι γνωστή ως οριακή τιμή του συστήματος, δεδομένου ότι αντιπροσωπεύει το οριακό κόστος για την ενέργεια, δηλαδή το επιπλέον κόστος που πρέπει να καταβληθεί, όταν η ζήτηση αυξάνεται κατά 1 MWh.

Κάθε παραγωγός αντιμετωπίζει το πρόβλημα της επιλογής της βέλτιστης τιμής προσφοράς που ο ίδιος θα πρέπει να υποβάλει στον ISO για κάθε περίοδο λειτουργίας του ορίζοντα προγραμματισμού. Εδώ, ο όρος βέλτιστη αφορά το γεγονός ότι το κέρδος του παραγωγού μετά την εκκαθάριση της αγοράς από τον ISO (που επιθυμεί το μικρότερο δυνατό κόστος) θα πρέπει να είναι το μέγιστο δυνατό. Ακόμη και στη σπάνια περίπτωση που ένας τέτοιος παραγωγός έχει πλήρη ενημέρωση των παραμέτρων της αγοράς (δηλαδή, τα τεχνικά χαρακτηριστικά και τις προσφορές /κόστη όλων των συμμετεχόντων παραγωγών, καθώς και τη ζήτηση για ενέργεια), το πρόβλημα της μεγιστοποίησης του κέρδους που αντιμετωπίζει είναι διεπίπεδο.

Στα πλαίσια της παρούσας διπλωματικής εργασίας, αναλύεται η μορφοποίηση και εν τέλει η επίλυση του προβλήματος με ενιαία αποζημίωση των συμμετεχόντων με το οριακό κόστος του συστήματος (SMP), το οποίο θα καθορίζεται από συγκεκριμένο παραγωγό (marginal) σε κάθε περίπτωση σύμφωνα με συγκεκριμένους κανόνες βελτιστότητας.

2.1 Περιγραφή μαθηματικού μοντέλου με τιμή εκκαθάρισης το οριακό κόστος του συστήματος (SMP)

Αρχικά, προσδιορίζονται οι συμβολισμοί και οι ονοματολογίες των συνόλων, μεταβλητών απόφασης και παραμέτρων του προβλήματος:

Σύνολα	
I	Μονάδες παραγωγής με δείκτη i

Μεταβλητές απόφασης	
F	Το συνολικό κέρδος του στρατηγικού παραγωγού στο σύνολο του χρονικού ορίζοντα
f	Το κόστος του ISO για την ικανοποίηση της ζήτησης στο σύνολο του χρονικού ορίζοντα
p_1	Ακέραια μεταβλητή που δείχνει την τιμή προσφοράς που δίνει ο στρατηγικός παραγωγός στον ISO, για την ενέργεια που παράγει
q_i	Συνεχής μεταβλητή που καθορίζει την ποσότητα σε MWh ενέργειας που τελικά θα παράξει ο κάθε παραγωγός
z_i	Δυαδική μεταβλητή που ισούται με 1 αν ο παραγωγός *i* συμμετέχει στην παραγωγή και 0 αν όχι
smp	Συνεχής μεταβλητή που καθορίζει την οριακή τιμή εκκαθάρισης του συστήματος από τον ISO

Παράμετροι	
p_i	Η τιμή της προσφοράς που δίνει ο παραγωγός *i* στον ISO για την ενέργεια

	που θα παράξει
m_i	Το τεχνικό ελάχιστο του παραγωγού i
k_i	Το τεχνικό μέγιστο του παραγωγού i
P	Το ανώτατο όριο της προσφοράς που επιτρέπεται να καταθέσει κάθε παραγωγός (price cap)
c_1	Μοναδιαίο κόστος ενέργειας για το στρατηγικό παραγωγό και ταυτόχρονα το κατώτατο όριο της προσφοράς που μπορεί να υποβάλλει
s_i	Κόστος εκκίνησης του παραγωγού i
D	Η συνολική ζήτηση σε ενέργεια

Πίνακας 2.1.1: Συμβολισμός και ονοματολογία των συνόλων, μεταβλητών απόφασης και των παραμέτρων του μαθηματικού μοντέλου με ενιαία τιμή εκκαθάρισης την οριακή τιμή του συστήματος SMP

Το μαθηματικό μοντέλο μεικτού ακέραιου διεπίπεδου προγραμματισμού για τον συγκεκριμένο τρόπο αποζημίωσης-εκκαθάρισης είναι το εξής:

$$Max_{p_1}\, F = (smp - c_1) * q_1 \tag{2.1}$$

$$s.t\;\; c_1 \leq p_1 \leq P \tag{2.2}$$

$$p_1 \; ακέραια\; μεταβλητή \tag{2.3}$$

$$smp \; ακέραια\; μεταβλητή \tag{2.4}$$

$$(z_i, q_i) \;\in \arg min_{z_i, q_i}\; f = \sum_{i=1}^{I} (p_i * q_i + s_i) \tag{2.5}$$

$$s.t\; \sum_{i=1}^{I} q_i = d \tag{2.6}$$

$$m_i * z_i \leq q_i \leq k_i * z_i \quad \forall i \tag{2.7}$$

$$z_i, δυαδική\; μεταβλητή\; \forall ι \tag{2.8}$$

$$q_i \geq 0\;\; \forall i \tag{2.9}$$

Αρχικά, η αντικειμενική συνάρτηση του ανωτέρου επιπέδου (2.1) μεγιστοποιεί το κέρδος του στρατηγικού παραγωγού. Το κέρδος αυτό εξαρτάται από την οριακή τιμή εκκαθάρισης του συστήματος από τον ISO και φυσικά από την ποσότητα που εν τέλει θα παράξει ο στρατηγικός παραγωγός. Το κόστος εκκίνησης δεν περιλαμβάνεται στην αντικειμενική συνάρτηση (2.1) διότι σύμφωνα με το συγκεκριμένο σύστημα αποζημίωσης-εκκαθάρισης το κόστος αυτό επιβαρύνει τον ISO, ο οποίος θα αναλάβει εν τέλει την αποπληρωμή του στους παραγωγούς. Ο περιορισμός (2.2) προσδιορίζει τα όρια μέσα στα οποία μπορεί να διακυμανθεί η προσφορά του στρατηγικού παραγωγού, ενώ ο περιορισμός (2.3) δηλώνει την ακεραιότητα της προσφοράς που τελικά θα υποβάλλει. Στη συνέχεια, ο περιορισμός (2.4) επιβάλλει ότι η οριακή τιμή του συστήματος είναι ακέραια.

Το πρόβλημα του κατωτέρου επίπεδου (πρόβλημα του ISO) ορίζεται από τις σχέσεις (2.5) – (2.9) και λειτουργεί ως περιορισμός για το ανώτερο επίπεδο. Η αντικειμενική συνάρτηση του κατωτέρου επιπέδου (2.5) ελαχιστοποιεί το κόστος στη διαδικασία ανάθεσης της ενέργειας στους υποψήφιους παραγωγούς από τον ISO. Έπειτα, ο περιορισμός (2.6) εισάγει στο μοντέλο την ανάγκη κάλυψης της συνολικής ζήτησης d από τους παραγωγούς που εν τέλει θα επιλεχθούν για να την ικανοποιήσουν, ενώ ο περιορισμός (2.7) οριοθετεί τις ποσότητες που μπορούν να παραχθούν από κάθε παραγωγό σε περίπτωση που επιλεχθούν. Τέλος, οι περιορισμοί (2.8) και (2.9) υποδηλώνουν την δυαδικότητα και τη μη αρνητικότητα των μεταβλητών που ορίζουν την ανάθεση και την ποσότητα αντίστοιχα.

2.2 Κανόνες βελτιστότητας

Όπως προαναφέρθηκε, η αποζημίωση των παραγωγών που τελικά συμμετέχουν στην παραγωγική διαδικασία είναι ενιαία και ισούται με την οριακή τιμή του συστήματος. Πιο συγκεκριμένα, η οριακή τιμή του συστήματος (smp) ισούται με τη σκιώδη τιμή του περιορισμού κάλυψης της ζήτησης (2.6). Ο προσδιορισμός της σκιώδους αυτής τιμής πραγματοποιείται με βάση κάποιες κατευθυντήριες γραμμές της αγοράς (μοντελοποιήσεις ακέραιου προγραμματισμού που χρησιμοποιούνται για την επιβολή των κανόνων σχεδιασμού της αγοράς), οι οποίες αποτελούν και κανόνες βελτιστότητας για το πρόβλημα του ISO.

Η εύρεση της οριακής τιμής του συστήματος καθορίζεται από τη βέλτιστη λύση του προβλήματος του ISO σύμφωνα με τους ακόλουθους κανόνες της αγοράς, όπως τεκμηριώνεται και στη συνέχεια.

- *Κανόνας 1*: Αν υπάρχει μια μονάδα i, της οποίας η παραγόμενη ενέργεια τη χρονική περίοδο t είναι αυστηρά μεταξύ του τεχνικού της ελαχίστου και μεγίστου ($m_i < q_{i,t} < k_i$), τότε η σκιώδης τιμή του συστήματος smp_t είναι ίση με την τιμή προσφοράς της μονάδας αυτής ($p_{i,t}$).
- *Κανόνας 2*: Εάν δεν υπάρχει καμία μονάδα i, της οποίας η παραγόμενη ενέργεια τη χρονική περίοδο t να είναι αυστηρά μεταξύ του τεχνικού της ελαχίστου και μεγίστου (κανόνας 1) και υπάρχει τουλάχιστον μία που παράγει ποσότητα ίση με το τεχνικό της ελάχιστο, τότε η σκιώδης τιμή του συστήματος (smp_t) είναι ίση με την ελάχιστη τιμή προσφοράς μεταξύ των μονάδων που παράγουν στο τεχνικό τους ελάχιστο.
- *Κανόνας 3*: Εάν όλες οι συμμετέχουσες μονάδες παράγουν ενέργεια ίση με το τεχνικό τους μέγιστο, τότε η σκιώδης τιμή του συστήματος (smp) είναι ίση με την μέγιστη τιμή προσφοράς μεταξύ όλων των μονάδων που παράγουν στο τεχνικό τους μέγιστο.

Οι παραπάνω κανόνες τεκμηριώνονται και επαληθεύονται μέσω των ακολούθων 5 ενδεχόμενων περιπτώσεων, οι οποίες καλύπτουν τους πιθανούς συνδυασμούς λύσεων για το σύνολο του προβλήματος. Για το σκοπό αυτό, χρησιμοποιείται ένα παράδειγμα με 2 παραγωγούς-μονάδες ηλεκτρικής ενέργειας, με συγκεκριμένα χαρακτηριστικά λειτουργίας, τα τεχνικά μέγιστα και ελάχιστα τους που είναι η ελάχιστη και αντίστοιχα η μέγιστη ποσότητα ενέργειας που μπορεί να παράγει η κάθε μονάδα. Ορίζονται ως M_1και K_1 το τεχνικό ελάχιστο και μέγιστο του 1ου παραγωγού, ενώ M_2 και K_2, τα τεχνικά ελάχιστα και μέγιστα αντίστοιχα του 2ου παραγωγού. Να σημειωθεί ότι οι παραχθείσες ποσότητες ενέργειας θεωρούνται ακέραιες.

1η περίπτωση : Έστω βέλτιστη λύση με παραχθείσες ποσότητες ενέργειας Q_1, Q_2 ανάμεσα από τα τεχνικά ελάχιστα και μέγιστα, $M_i < Q_i < K_i$ με $i = 1,2$. Η προσφορά-τιμή P_1 που δίνει ο πρώτος παραγωγός είναι μεγαλύτερη από αυτή του δεύτερου παραγωγού P_2, $P_1 > P_2$. Αν τώρα μειωθεί κατά μία μονάδα η ποσότητα που παράγει ο πρώτος παραγωγός και παράλληλα αυξηθεί κατά μία μονάδα η ποσότητα που παράγει ο δεύτερος, παραμένουν και οι 2 παραγωγοί μέσα στη λύση καλύπτοντας την συνολική ζήτηση. Το κόστος όμως κάλυψης της ζήτησης μειώνεται κατά $(P_1 - P_2)$. Συνεπώς καταλήγουμε σε ΑΤΟΠΟ, εφόσον δεν μπορούν οι ποσότητες και των 2 παραγωγών να βρίσκονται ανάμεσα στα αντίστοιχα ελάχιστα και μέγιστα τους όταν οι τιμές προσφοράς τους είναι διαφορετικές. Ως εκ τούτου επαληθεύεται η συνθήκη βελτιστότητας κατά την οποία μόνο ένας από τους 2 παραγωγούς μπορεί να παράγει ποσότητα ανάμεσα στο τεχνικό του μέγιστο και ελάχιστο στη βέλτιστη λύση.

2η περίπτωση : Έστω βέλτιστη λύση με παραχθείσες ποσότητες ενέργειας Q_1, Q_2, με την Q_1 ανάμεσα από το τεχνικό της μέγιστο/ελάχιστο, $M_1 < Q_1 < K_1$ και την Q_2 στο τεχνικό της ελάχιστο $Q_2 = M_2$. Η προσφορά-τιμή P_1 που δίνει ο πρώτος παραγωγός είναι μεγαλύτερη από αυτή του δεύτερου παραγωγού P_2, $P_1 > P_2$. Αν τώρα μειωθεί κατά μία μονάδα η ποσότητα που παράγει ο πρώτος παραγωγός και παράλληλα αυξηθεί κατά μία μονάδα η ποσότητα που παράγει ο δεύτερος, παραμένουν και οι 2 παραγωγοί μέσα στη λύση καλύπτοντας την συνολική ζήτηση. Το κόστος όμως κάλυψης της ζήτησης μειώνεται κατά $(P_1 - P_2)$. Συνεπώς καταλήγουμε σε ΑΤΟΠΟ, δεν μπορεί ο παραγωγός που παράγει ποσότητα ίση με το τεχνικό του ελάχιστο και βρίσκεται στη βέλτιστη λύση να έχει προσφορά-τιμή μικρότερη από αυτή του παραγωγού που παράγει ποσότητα μεταξύ του τεχνικού μεγίστου/ελαχίστου του.

3η περίπτωση : Έστω βέλτιστη λύση με παραχθείσες ποσότητες ενέργειας Q_1, Q_2, με την Q_1 ανάμεσα από το τεχνικό της μέγιστο/ελάχιστο, $M_1 < Q_1 < K_1$ και την Q_2 στο τεχνικό της μέγιστο $Q_2 = K_2$. Η προσφορά-τιμή P_1 που δίνει ο πρώτος παραγωγός είναι μικρότερη από αυτήν του δεύτερου παραγωγού P_2, $P_1 < P_2$. Αν τώρα μειωθεί κατά μία μονάδα η ποσότητα που παράγει ο δεύτερος παραγωγός και παράλληλα αυξηθεί κατά μία μονάδα η ποσότητα που παράγει ο πρώτος, παραμένουν και οι 2 παραγωγοί μέσα στη λύση καλύπτοντας την συνολική ζήτηση. Το κόστος όμως κάλυψης της ζήτησης μειώνεται κατά $(P_2 - P_1)$. Συνεπώς καταλήγουμε σε ΑΤΟΠΟ, δεν μπορεί ο παραγωγός που παράγει ποσότητα ίση με το τεχνικό του μέγιστο και βρίσκεται στη βέλτιστη λύση να έχει προσφορά-τιμή μεγαλύτερη από αυτή του παραγωγού που παράγει ποσότητα μεταξύ του τεχνικού μεγίστου/ελαχίστου του.

4η περίπτωση : Έστω βέλτιστη λύση με παραχθείσες ποσότητες ενέργειας Q_1, Q_2, με την Q_1 στο τεχνικό ελάχιστο, $Q_1 = M_1$ και την Q_2 στο τεχνικό της μέγιστο $Q_2 = K_2$. Η προσφορά-τιμή P_1 που δίνει ο πρώτος παραγωγός είναι μικρότερη από αυτή του δεύτερου παραγωγού P_2, $P_1 < P_2$. Αν τώρα μειωθεί κατά μία μονάδα η ποσότητα που παράγει ο δεύτερος παραγωγός και παράλληλα αυξηθεί κατά μία μονάδα η ποσότητα που παράγει ο πρώτος, παραμένουν και οι 2 παραγωγοί μέσα στη λύση καλύπτοντας την συνολική ζήτηση. Το κόστος όμως κάλυψης της ζήτησης μειώνεται κατά $(P_2 - P_1)$. Συνεπώς καταλήγουμε σε ΑΤΟΠΟ, δεν μπορεί ο παραγωγός που παράγει ποσότητα ίση με το τεχνικό του μέγιστο και βρίσκεται στη βέλτιστη λύση να έχει προσφορά-τιμή μεγαλύτερη από αυτή του παραγωγού που παράγει ποσότητα ίση με το τεχνικό του ελάχιστο.

5η περίπτωση : Έστω βέλτιστη λύση με παραχθείσες ποσότητες ενέργειας Q_1, Q_2 ανάμεσα από τα τεχνικά ελάχιστα και μέγιστα, $M_i < Q_i < K_i$ με i = 1,2. Η προσφορά-τιμή P_1 που δίνει ο πρώτος παραγωγός είναι ίση με αυτή του δεύτερου παραγωγού P_2, $P_1 = P_2$. Αν τώρα μειωθεί κατά μία μονάδα η ποσότητα που παράγει ο πρώτος παραγωγός και παράλληλα αυξηθεί κατά μία μονάδα η ποσότητα που παράγει ο δεύτερος, διακρίνονται οι εξής περιπτώσεις:

1. Να παραμείνουν και οι δύο ποσότητες ανάμεσα στα τεχνικά τους όρια.
2. Η ποσότητα του πρώτου παραγωγού να γίνει όση το τεχνικό του ελάχιστο και η ποσότητα του δεύτερου να παραμείνει ανάμεσα από τα τεχνικά του όρια.
3. Η ποσότητα του πρώτου παραγωγού να παραμείνει ανάμεσα από τα τεχνικά του όρια και του δεύτερου να γίνει όση το τεχνικό του μέγιστο.
4. Η ποσότητα του πρώτου παραγωγού να γίνει όση το τεχνικό του ελάχιστο και η ποσότητα του δεύτερου να γίνει όση το τεχνικό του μέγιστο.

Και στις τέσσερις περιπτώσεις η λύση δεν μεταβάλλεται, αφού $P_1 = P_2$. Συνεπώς, στη περίπτωση που οι τιμές/προσφορές των μονάδων είναι ίσες και υπάρχει μία βέλτιστη λύση, με οποιοδήποτε συνδυασμό ποσοτήτων, τότε θα υπάρχει σίγουρα και μία εναλλακτική βέλτιστη σύμφωνη με τους κανόνες βελτιστότητας. Επομένως, η μόνη περίπτωση στην οποία είναι δυνατόν να υπάρχουν στη βέλτιστη λύση 2 παραγωγοί που παράγουν αυστηρά ανάμεσα στο τεχνικό τους ελάχιστο και στο τεχνικό τους μέγιστο, είναι όταν οι τιμές προσφοράς των 2 παραγωγών είναι ίσες. Σε μια τέτοια περίπτωση όμως, είναι δυνατόν να πάρουμε μια εναλλακτική βέλτιστη λύση στην οποία το πολύ ένας παραγωγός παράγει ανάμεσα στο τεχνικό του ελάχιστο και στο τεχνικό του μέγιστο, πολύ απλά μεταφέροντας ποσότητα παραγόμενης ενέργειας από τον έναν στον άλλον. Ως εκ τούτου, επαληθεύεται η συνθήκη βελτιστότητας κατά την οποία υπάρχει πάντα βέλτιστη λύση στην οποία το πολύ ένας παραγωγός παράγει ανάμεσα στο τεχνικό του ελάχιστο και στο τεχνικό του μέγιστο.

3. Μεθοδολογία επίλυσης

Στο διεπίπεδο προγραμματισμό ο υπεύθυνος λήψης αποφάσεων του ανωτέρου επιπέδου δεν έχει τη δυνατότητα να εξαναγκάσει τον αντίστοιχα υπεύθυνο του κατωτέρου επιπέδου σε μία λύση η οποία θα είναι η βέλτιστη για αυτόν. Πάνω σε αυτή τη διαπίστωση στηρίχθηκε και η ανάγκη εύρεσης μίας προσέγγισης, κατά την οποία ο στρατηγικός παραγωγός των προβλημάτων που θα αναλυθούν σε επόμενο κεφάλαιο επιλέγει τη βέλτιστη λύση για αυτόν και ταυτόχρονα ο ISO ελαχιστοποιεί το κόστος της παραγωγής διατηρώντας ωστόσο ως δεδομένη τη λύση του στρατηγικού παραγωγού. Ο αλγόριθμος που εν τέλει αναπτύχθηκε, μοντελοποιήθηκε μαθηματικά και με τη βοήθεια της γλώσσας προγραμματισμού C επιλύθηκε μία πληθώρα προβλημάτων με διαφορετικά δεδομένα

Πιο συγκεκριμένα, αναπτύχθηκε ένας αλγόριθμος, για το πρόβλημα με ενιαία αποζημίωση το οριακό κόστος του προβλήματος (smp), σύμφωνα με τον οποίο η ποσότητα Q_1 του στρατηγικού παραγωγού και το οριακό κόστος του συστήματος smp εισάγονται ως δεδομένα στο πρόβλημα και ο ανεξάρτητος διαχειριστής του συστήματος (ISO) θα επιχειρήσει την ελαχιστοποίηση του κόστους, χωρίς αυτό φυσικά να σημαίνει ότι θα υπάρχει και λύση με τις τιμές των δεδομένων που δόθηκαν. Όπως, αποδείχτηκε στο προηγούμενο κεφάλαιο, οι 3 κατευθυντήριες γραμμές της αγοράς την παραγωγής ηλεκτρικής ενέργειας είναι ταυτόχρονα και κανόνες βελτιστότητας για τον πρόβλημα που καλείται να επιλύσει ο ISO. Αυτό σημαίνει ότι η λύση (ή οι λύσεις) η οποία θα προκύψει με βάση τα δεδομένα Q_1 και smp θα υπακούει έναν από τους 3 αυτούς κανόνες και θα είναι και βέλτιστη για τον ISO. Αυτό σημαίνει ότι η εισαγωγή των κανόνων αυτών μαθηματικά ως περιορισμούς θα κατευθύνει εν τέλει τον αλγόριθμο στην εξαγωγή του βέλτιστου συνδυασμού ποσοτήτων για τον ISO.

Σημείωση: *Ο αλγόριθμος δημιουργήθηκε με στόχο να επιλύσει το πρόβλημα της μίας περιόδου.*

3.1 Σύνολα, Παράμετροι και Μεταβλητές απόφασης του Προβλήματος

Οι παράμετροι του προβλήματος είναι οι εξής:

Σύνολα	
I	Το πλήθος των παραγωγών

Πίνακας 3.1.1: Σύνολα του προβλήματος με ενιαία αποζημίωση (smp)

Παράμετροι	
Q_1	Η δεδομένη ποσότητα του στρατηγικού παραγωγού
smp	Η δεδομένη τιμή εκκαθάρισης της αγοράς ή αλλιώς το οριακό κόστος του συστήματος
M_ι	Το τεχνικό ελάχιστο που δύναται να παράξει ο κάθε παραγωγός *i* (εκτός του στρατηγικού παραγωγού)
K_i	Το τεχνικό μέγιστο που δύναται να παράξει ο κάθε παραγωγός *i* (εκτός του στρατηγικού παραγωγού)
c_1	Το μεταβλητό κόστος του στρατηγικού παραγωγού ή αλλιώς το ελάχιστο όριο της προσφοράς που δύναται να καταθέσει
P_{max}	Το ανώτατο όριο προσφοράς που δύναται να καταθέσουν όλοι οι παραγωγοί (price cap)
M_1	Το τεχνικό ελάχιστο που δύναται να παράξει ο στρατηγικός παραγωγός
K_1	Το τεχνικό μέγιστο που δύναται να παράξει ο στρατηγικός παραγωγός
D	Η συνολική ζήτηση σε ενέργεια
P_i	Οι τιμές/προσφορές των παραγωγών (εκτός του στρατηγικού παραγωγού)

Πίνακας 3.1.2: Παράμετροι του προβλήματος με ενιαία αποζημίωση (smp)

Μεταβλητές απόφασης	
Q_i	Οι μεταβλητές απόφασης που υποδεικνύουν τις ποσότητες των παραγωγών *i* (εκτός του στρατηγικού παραγωγού)
Z_i	Οι μεταβλητές απόφασης που λαμβάνουν την τιμή 1 εφ' όσον ο παραγωγός *i* συμμετέχει στη λύση και 0 αν όχι
Y_i	Βοηθητικές μεταβλητές απόφασης που χρησιμοποιούνται όταν οι ποσότητες Q_iείναι είτε στο τεχνικό τους ελάχιστος είτε στο τεχνικό τους μέγιστο

Πίνακας 3.1.3: Μεταβλητές απόφασης του προβλήματος με ενιαία αποζημίωση (smp)

3.2 Επεξήγηση διαδικασίας αλγορίθμου

Η λογική, με βάση την οποία προσεγγίστηκε και τελικά διαμορφώθηκε ο αλγόριθμος επίλυσης του προβλήματος, ξεκινά με την αναγνώριση του παραγωγού εκείνου που τελικά θα καθορίσει το οριακό κόστος (smp). Εφ' όσον η ενιαία αποζημίωση των παραγωγών που θα συμμετέχουν στην παραγωγή είναι δεδομένη, καθίσταται δυνατή η σύγκριση των τιμών/προσφορών (bids) όλων των υπολοίπων παραγωγών (εκτός του στρατηγικού) με το οριακό κόστος που έχει επιλεγεί για το πρόβλημα. Στην περίπτωση που το οριακό κόστος (smp) ισούται με κάποια από τις προσφορές των παραγωγών i του προβλήματος ($P_i = smp$) τότε ο παραγωγός για τον οποίο ισχύει η ισότητα είναι αυτόματα και αυτός ο οποίος θα καθορίσει εν τέλει το οριακό κόστος (marginal producer). Ο προσδιορισμός του κρίνεται αναγκαίος, καθώς με γνώμονα αυτόν γίνονται όλες οι συγκρίσεις ποσοτήτων και τιμών των υπολοίπων παραγωγών (όπως και του στρατηγικού παραγωγού), με στόχο την επαλήθευση των κανόνων βελτιστότητας που αναλύθηκαν στο Κεφάλαιο 2. Στην περίπτωση κατά την οποία η τιμή του οριακού κόστους (smp) δεν ισούται με κάποια από τις προσφορές P_i των υπολοίπων παραγωγών τότε εξυπακούεται ότι αυτός που καθορίζει το οριακό κόστος είναι ο στρατηγικός παραγωγός. Στην περίπτωση αυτή, η εισαγωγή ενός περιορισμού επιβάλλεται έτσι ώστε η μεταβλητή απόφασης P_1 να ισούται με το οριακό κόστος του συστήματος (smp) που έχει εισαχθεί ως δεδομένο ($P_1 = smp$).

Ο αλγόριθμος συμπεριλαμβάνει 2 γενικούς περιορισμούς που αφορούν τη ζήτηση και τα όρια της ποσότητας Q_1 του στρατηγικού παραγωγού. Αρχικά, εισάγεται ο περιορισμός της κάλυψης της ζήτησης, σύμφωνα με τον οποίο πρέπει το άθροισμα των ποσοτήτων Q_i των παραγωγών i να ισούται με τη συνολική ζήτηση D. Η ποσότητα Q_1 του στρατηγικού παραγωγού είναι δεδομένη και, με τη λογική ότι ο αλγόριθμος κατασκευάστηκε για να εντοπίσει τις πιθανές λύσεις στις οποίες όμως θα συμμετέχει σίγουρα ο στρατηγικός παραγωγός, αφαιρείται από τη συνολική ζήτηση, δηλαδή $\sum_{i=1}^{I} Q_i = D - Q_1$. Έπειτα, εισάγεται και ο περιορισμός της οριοθέτησης των ποσοτήτων Q_i των υπολοίπων παραγωγών, σύμφωνα με τον οποίο πρέπει η ποσότητα αυτή να βρίσκεται ανάμεσα στο τεχνικό του ελάχιστο και στο τεχνικό του μέγιστο, δηλαδή $M_i \leq Q_i \leq K_i$.

Μετά τον εντοπισμό, λοιπόν, του παραγωγού εκείνου (marginal) που θα καθορίζει το οριακό κόστος (smp) και των 2 γενικών περιορισμών (ζήτηση και οριοθέτηση ποσότητας παραγωγών), ο αλγόριθμος κατηγοριοποιεί τη διαδικασία λύσης του με βάση την ποσότητα Q_1 που έχει εισαχθεί ως δεδομένη για το στρατηγικό παραγωγό. Οι κατηγορίες είναι οι εξής :

1. $\boldsymbol{M_1 < Q_1 < K_1}$

   Στην περίπτωση αυτή εξετάζεται το ενδεχόμενο η δεδομένη ποσότητα Q_1 να βρίσκεται αυστηρά ανάμεσα στο τεχνικό της ελάχιστο και μέγιστο. Με βάση και τον 1ο κανόνα βελτιστότητας που αναλύθηκε σε προηγούμενο κεφάλαιο εφ' όσον η ποσότητα βρίσκεται αυστηρά ανάμεσα στο μέγιστο και ελάχιστο της τότε ο παραγωγός αυτός θα καθορίσει και το οριακό κόστος (smp). Έτσι, εξετάζεται το ενδεχόμενο εκείνο στο οποίο ο στρατηγικός παραγωγός καθορίζει το οριακό κόστος (marginal producer).

   Όπως αναφέρθηκε και παραπάνω η εισαγωγή της ισότητας του οριακού κόστους (smp) και της προσφοράς P_1 ως περιορισμό είναι αναγκαία. Στη συνέχεια, εισάγεται ο κανόνας βελτιστότητας που ταιριάζει για την περίπτωση αυτή και αυτός είναι ο 1ος. Για να καθορίζει ο στρατηγικός παραγωγός το οριακό κόστος (smp) θα πρέπει όσοι παραγωγοί έχουν τιμή/προσφορά χαμηλότερη από αυτόν ($P_i < smp$), αν συμμετέχουν, να συμμετέχουν με τη μέγιστη τιμή τους, δηλαδή

$Q_i = Z_i * K_i$, ενώ όσοι παραγωγοί έχουν τιμή/προσφορά υψηλότερη από αυτόν ($P_i > smp$), αν συμμετέχουν, να συμμετέχουν με την ελάχιστη τιμή τους, δηλαδή $Q_i = Z_i * M_i$. Στην περίπτωση, ωστόσο, της ισότητας ($P_i = smp$) οι παραγωγοί, εφ' όσον συμμετέχουν, θα συμμετέχουν είτε με την ελάχιστη είτε με την μέγιστη ποσότητα τους ($Q_i = Y_{2i} * M_i + Y_{2i+1} * K_i$ για $0 \leq i \leq I$). Ακριβώς επειδή οι βοηθητικές μεταβλητές Y χρησιμοποιούνται για να καθοδηγήσουν την ποσότητα Q είτε στο ελάχιστο είτε στο μέγιστο προστέθηκε ο περιορισμός όπου το άθροισμα των Y για κάθε παραγωγό θα πρέπει να ισούται με την τιμή της μεταβλητής απόφασης Z ($Y_{2i} + Y_{2i+1} = Z_i$). Όταν το Z θα παίρνει την τιμή 0 ο παραγωγός i δε θα συμμετέχει στη λύση οπότε χρειαζόμαστε μηδενικές βοηθητικές μεταβλητές για μηδενική ποσότητα, ενώ όταν το Z θα παίρνει την τιμή 1 τότε ο παραγωγός θα συμμετάσχει στη λύση και το άθροισμα των βοηθητικών μεταβλητών Y θα πρέπει να ισούται ακριβώς με 1.

2. $\boldsymbol{Q_1 = M_1}$

Η 2η περίπτωση είναι εκείνη στην οποία η δεδομένη ποσότητα Q_1 είναι ίση με την ελάχιστη επιτρεπόμενη τιμή της. Η ιδιομορφία που παρουσιάζει αυτή η περίπτωση είναι ότι ο παραγωγός που θα καθορίσει το οριακό κόστος (smp) μπορεί να είναι τόσο ο ίδιος ο στρατηγικός παραγωγός όσο και οι άλλοι (σε αντίθεση με την 1η περίπτωση όπου σίγουρα είναι ο στρατηγικός). Έτσι κρίνεται αναγκαία η κατηγοριοποίηση της 2ης περίπτωσης σε 2 άλλες υποπεριπτώσεις.

- Όταν ο στρατηγικός παραγωγός είναι τελικά αυτός που καθορίζει το οριακό κόστος του συστήματος (προκύπτει από τη σύγκριση των τιμών/προσφορών με το smp) τότε πρέπει να επαληθεύεται ο 2ος κανόνας βελτιστότητας και φυσικά εισάγεται ο περιορισμός της ισότητας του οριακού κόστους smp και της τιμής/προσφοράς του στρατηγικού παραγωγού P_1, δηλαδή $P_1 = smp$.
  Για να επαληθεύεται ο 2ος κανόνας βελτιστότητας, πρέπει οι παραγωγοί i των οποίων η τιμή/προσφορά (P_i) είναι μικρότερη από την αντίστοιχη του στρατηγικού παραγωγού [$P_i < (P_1 = smp)$], εφ' όσον συμμετέχουν, να συμμετέχουν με τη μέγιστη ποσότητα τους, δηλαδή $Q_i = Z_i * K_i$. Ταυτόχρονα, οι παραγωγοί των οποίων η τιμή/προσφορά (P_i) είναι μεγαλύτερη από την αντίστοιχη του στρατηγικού παραγωγού, εφ' όσον συμμετέχουν, πρέπει να συμμετέχουν με την ελάχιστη ποσότητα τους, δηλαδή $Q_i = Z_i * M_i$. Τέλος στην περίπτωση που εμφανιστεί το φαινόμενο της ισότητας των τιμών του στρατηγικού παραγωγού με κάποια/κάποιες εκ των υπολοίπων παραγωγών, όπως και στην προηγούμενη περίπτωση έτσι και τώρα, δύναται η ποσότητα του παραγωγού i να έχει είτε την ελάχιστη είτε τη μέγιστη τιμή της, δηλαδή ($Q_i = Y_{2i} * M_i + Y_{2i+1} * K_i$ για $0 \leq i \leq I$). Η χρήση των βοηθητικών μεταβλητών Y έχει την ίδια φιλοσοφία με την ανάλυση που πραγματοποιήθηκε στην προηγούμενη περίπτωση.
- Στην περίπτωση που κάποιος άλλος εκτός του στρατηγικού παραγωγού καθορίζει το οριακό κόστος (smp) τότε τα πράγματα γίνονται λίγο πιο πολύπλοκα. Αρχικά, η ποσότητα Q_i του παραγωγού που θα καθορίσει το

οριακό κόστος πρέπει να είναι μικρότερη ή ίση από την μέγιστη ποσότητα που μπορεί να παράξει ($Q_i < K_i$). Σύμφωνα με τον 3^{ο} κανόνα βελτιστότητας, αν ο παραγωγός που καθορίζει το οριακό κόστος βρίσκεται στο μέγιστο του τότε όλοι όσοι θα συμμετέχουν πρέπει και αυτοί να συμμετέχουν με τη μέγιστη ποσότητα τους. Έτσι, κατά την εξέταση της περίπτωσης όπου ο στρατηγικός παραγωγός βρίσκεται στην ελάχιστη τιμή του, είναι ανώφελο να επιτραπεί στον παραγωγό που καθορίζει το οριακό κόστος να πάρει τη μέγιστη τιμή του, εφ' όσον ο στρατηγικός δε μπορεί να συμμετάσχει.

Στη συνέχεια, είτε ο παραγωγός που καθορίζει το οριακό κόστος (marginal) βρίσκεται ανάμεσα στο τεχνικό του ελάχιστο και μέγιστο είτε ακριβώς στο τεχνικό του ελάχιστο, η τιμή του στρατηγικού παραγωγού πρέπει να είναι μεγαλύτερη από αυτού, δηλαδή $P_1 > P_I$, όπου σαν I συμβολίζεται ο marginal παραγωγός. Αν ο marginal παραγωγός βρίσκεται αυστηρά ανάμεσα στο τεχνικό του ελάχιστο και μέγιστο και ο στρατηγικός παραγωγός είναι φιξαρισμένος στην ελάχιστη ποσότητα του, τότε εξυπακούεται πως ο στρατηγικός παραγωγός πρέπει να είναι πιο ακριβός, σύμφωνα πάντα με τον 1^{ο} κανόνα βελτιστότητας. Αν ο marginal παραγωγός βρίσκεται στο ελάχιστο του τότε, σύμφωνα με τον 3^{ο} κανόνα βελτιστότητας, αυτός θα είναι και ο φθηνότερος. Σημαντικό καθίσταται να εξασφαλιστεί, επίσης, μέσω περιορισμού η συμμετοχή του marginal παραγωγού και αυτό επιτυγχάνεται με την υποχρεωτική εξίσωση της μεταβλητής Z_i του marginal παραγωγού i ίσης με τη μονάδα ($Z_i = 1$).

Η επαλήθευση του 2^{ου} κανόνα βελτιστότητας, σε αυτήν την περίπτωση, πραγματοποιείται με την ίδια λογική που χρησιμοποιήθηκε στην 1^{η} περίπτωση ($M_1 < Q_1 < K_1$). Οι ακριβότεροι του marginal παραγωγού, εφ' όσον συμμετέχουν, θα συμμετέχουν με την ελάχιστη ποσότητα τους, οι φθηνότεροι με την μέγιστη ποσότητα τους και στην περίπτωση της ισότητας ισχύει και πάλι η συμμετοχή είτε στο ελάχιστο είτε στο μέγιστο με τη χρήση των βοηθητικών μεταβλητών Y και τη φιλοσοφίας τους.

3. $Q_1 = K_1$

Η τελευταία περίπτωση που δύναται να προκύψει είναι εκείνη στην οποία η δεδομένη ποσότητα Q_1 του στρατηγικού παραγωγού παίρνει την μέγιστη τιμή της. Όπως και στην προηγούμενη περίπτωση έτσι και εδώ, ο marginal παραγωγός ενδέχεται να είναι τόσο ο στρατηγικός παραγωγός του προβλήματος όσο και κάποιος από τους υπόλοιπους, ανάλογα πάντα με την ταύτιση ή μη τις τιμής/προσφοράς των παραγωγών με το οριακό κόστος του συστήματος. Οι 2 υποκατηγορίες που προκύπτουν και η ανάλυση τους είναι οι εξής:

- Στην περίπτωση όπου ο marginal παραγωγός είναι και ο στρατηγικός παραγωγός, εισάγεται ο περιορισμός εκείνος που εξισώνει την τιμή/προσφορά του στρατηγικού παραγωγού με το οριακό κόστος του συστήματος ($P_1 = smp$).

  Στη συνέχεια, αφού η ποσότητα του στρατηγικού παραγωγού Q_i βρίσκεται στη μέγιστη τιμή της και ο ίδιος είναι αυτός που καθορίζει το οριακό κόστος, θα πρέπει να επαληθεύεται ο 3ος κανόνας βελτιστότητας. Σύμφωνα τον κανόνα, όσοι παραγωγοί έχουν τιμή/προσφορά μικρότερη από αυτή του στρατηγικού παραγωγού ή ίση με αυτή θα συμμετέχουν ,εφ' όσον συμμετέχουν, με τη μέγιστη τιμή τους. Οι παραγωγοί των οποίων η τιμή/προσφορά είναι μεγαλύτερη από του στρατηγικού παραγωγού υποχρεωτικά δε θα συμμετέχουν στην παραγωγή. Αυτό συμβαίνει διότι, σύμφωνα με τον 3ο κανόνα, όλοι οι παραγωγοί που θα συμμετέχουν θα είναι στο μέγιστο τους και αυτός που καθορίζει το οριακό κόστος έχει τη μεγαλύτερη τιμή.

- Στην περίπτωση όπου κάποιος εκτός του στρατηγικού παραγωγού καθορίζει το οριακό κόστος, η μεταβλητή απόφασης Z_i που υποδεικνύει τη συμμετοχή ή μη αυτού του παραγωγού i εξισώνεται αυτόματα ίση με τη μονάδα ($Z_i = 1$).

  Στη συνέχεια, εισάγεται ένας περιορισμός ο οποίος έχει στόχο τόσο να περιορίσει το εύρος του πεδίου λύσεων της μεταβλητής P_1 όσο και να καθοδηγήσει την ίδια μεταβλητή στην τιμή που πρέπει να έχει ώστε να μπορεί να συμμετέχει στην παραγωγή. Κατά συνέπεια, με τη λογική ότι ο marginal παραγωγός θα βρίσκεται στο μέγιστο του και πρέπει να έχει τη μεγαλύτερη τιμή από αυτούς που βρίσκονται στο μέγιστο τους συμπεραίνεται ότι η τιμή του πρέπει να είναι μικρότερη από αυτή που έχει καταθέσει ο παραγωγός που καθορίζει το οριακό κόστος ($P_1 < P_I$).

  Στην περίπτωση αυτή ο marginal παραγωγός μπορεί να παράξει οποιαδήποτε ποσότητα από το τεχνικό του ελάχιστο μέχρι το τεχνικό του μέγιστο. Εφ' όσον η ποσότητα είναι στο τεχνικό του ελάχιστο ή και ανάμεσα, οι υπόλοιπες μονάδες ακολουθούν είτε τον 1ο είτε τον 2ο κανόνα βελτιστότητας. Οι ακριβότεροι του marginal παραγωγού, εφ' όσον συμμετέχουν, θα συμμετέχουν με την ελάχιστη ποσότητα τους, οι φθηνότεροι με την μέγιστη ποσότητα τους και στην περίπτωση της ισότητας ισχύει και πάλι η συμμετοχή είτε στο ελάχιστο είτε στο μέγιστο με τη χρήση των βοηθητικών μεταβλητών Y και τη φιλοσοφίας τους (εκτός του στρατηγικού παραγωγού που ξέρουμε εξ αρχής ότι είναι στο τεχνικό του μέγιστο). Όταν όμως η ποσότητα που παράγει ο marginal παραγωγός είναι στο τεχνικό του μέγιστο στη λύση μπαίνουν μονάδες που έχουν μικρότερη τιμή/προσφορά από το οριακό κόστος και με ποσότητα ίση με το τεχνικό τους μέγιστο.

  Η διαφοροποίηση αυτή δημιούργησε την ανάγκη να γνωρίζει ο αλγόριθμος πότε η ποσότητα που παράγει ο marginal παραγωγός βρίσκεται στο τεχνικό του μέγιστο. Τη λύση σ' αυτό το πρόβλημα έδωσε η βοηθητική μεταβλητή J (δυαδική), η οποία γίνεται μονάδα όταν η ποσότητα του marginal είναι στο τεχνικό του μέγιστο.

  Οι περιορισμοί τώρα, όπως και στις προηγούμενες περιπτώσεις εισάγονται με την σύγκριση των τιμών/προσφορών των υπόλοιπων μονάδων παραγωγής με το οριακό κόστος. Αν η τιμή/προσφορά της εκάστοτε μονάδας παραγωγής είναι μικρότερη του οριακού κόστους τα πράγματα είναι ξεκάθαρα, η ποσότητα παραγωγής της θα είναι ή μηδενική ή μέγιστη. Για τις περιπτώσεις τώρα που είναι μεγαλύτερη ή ίση, προσθέτουμε στον περιορισμό μια ποσότητα ίση με (D +1) επί την μεταβλητή απόφασης J, έτσι ώστε όταν J=1 οι υπόλοιπες μονάδες που βάση των συνθηκών βελτιστότητας θα μπορούσαν να παράξουν ποσότητες μόνο ίσες με το τεχνικό τους ελάχιστο, να τις ωθεί ο αλγόριθμος τελικά να παράγουν ποσότητα μεγαλύτερη της ζήτησης. Αυτό εξ' ορισμού τις βγάζει εκτός λύσης.

Η επίλυση, ωστόσο, του αλγορίθμου μπορεί να μας δώσει περισσότερες από μία λύσεις. Για το λόγο αυτό, στην περίπτωση που ο αλγόριθμος υπολογίζει την πρώτη λύση και βρίσκει τις μονάδες παραγωγής που θα συμμετέχουν, εισάγει στον αλγόριθμο ένα νέο περιορισμό (cut) που αποκλείει την λύση αυτή και γεννά μία ενδεχόμενη διαφορετική βέλτιστη. Στη συνέχεια, επιλύεται ξανά ο αλγόριθμος και η διαδικασία συνεχίζεται έως ο αλγόριθμος να μη δύναται να εντοπίσει άλλες ενδεχόμενες λύσεις. Χαρακτηριστικά, αναλύεται και επεξηγείται ένα παράδειγμα τέτοιου περιορισμού στο Παράρτημα.

4. Εφαρμογή Αλγορίθμου

Στο κεφάλαιο αυτό, παρουσιάζεται η εφαρμογή της προτεινόμενης μεθοδολογίας, που αναλύθηκε εκτενώς στο προηγούμενο κεφάλαιο, σε 5 παραδείγματα με διαφορετικά δεδομένα. Οι μεταξύ τους διαφορές επικεντρώνονται κατά κύριο λόγο στον αριθμό των μονάδων παραγωγής (σύνολο I) που συμμετέχουν στην αγορά ηλεκτρικής ενέργειας καθώς και την ζήτηση (D) που απαιτείτε να καλυφθεί, την ποσότητα που θα παράξει ο στρατηγικός παραγωγός (Q_1) και το οριακό κόστος (smp). Με τον τρόπο αυτό, το πρόβλημα εξετάζεται σε διάφορες κλίμακες, μελετώντας την συμπεριφορά του αλγορίθμου στις παραπάνω μεταβολές.

4.1 Παράδειγμα 1ο

Σε πρώτη φάση, παρουσιάζεται ένα παράδειγμα με 10 μονάδες παραγωγής ηλεκτρικής ενέργειας. Τα τεχνικά χαρακτηριστικά (τεχνικό ελάχιστο και τεχνικό μέγιστο σε MWh) και οι τιμές/προσφορές, εκτός φυσικά της τιμής/προσφοράς του στρατηγικού παραγωγού που είναι μεταβλητή απόφασης και καθορίζεται από τον αλγόριθμο (σε €/MWh), παρουσιάζονται στον *Πίνακα 4.1.1* Το μεταβλητό κόστος c (κόστος παραγωγής) του μεμονωμένου στρατηγικού παραγωγού είναι 35 €/MWh, και το άνω όριο στην τιμή/προσφορά είναι 150 €/MWh. Το παράδειγμα αυτό το εφαρμόζουμε για ζήτηση ίση με 4200 MWh και 5200 MWh.

Μονάδα (i)	M_I	K_i	P_i
1	240	377	-
2	2400	3800	35
3	144	476	52
4	155	550	55
5	240	384	57
6	65	151	64
7	105	188	65
8	120	287	70
9	60	144	72
10	0	141	150

Πίνακας 4.1.1: Τεχνικά χαρακτηριστικά και τιμές/προσφορές των μονάδων παραγωγής

Αρχικά, η ζήτηση που απαιτείται να καλυφθεί από τους 10 παραγωγούς του συγκεκριμένου παραδείγματος είναι 4200 MWh. Όπως αναλύθηκε στο προηγούμενο κεφάλαιο, η ζήτηση πρέπει να καλυφθεί με τον στρατηγικό παραγωγό πάντα εντός της βέλτιστης λύσης, με συγκεκριμένη ποσότητα και για συγκεκριμένο οριακό κόστος (smp). Συνεπώς, για το παραπάνω παράδειγμα των 10 μονάδων παραγωγής θεωρήθηκε η ποσότητα που παράγει ο στρατηγικός παραγωγός ίση με το τεχνικό του μέγιστο, 377 MWh και το οριακό κόστος smp ίσο με 150 €/MWh.

Με την εισαγωγή των παραπάνω δεδομένων στον αλγόριθμο, βρίσκουμε κατά πόσο υπάρχουν βέλτιστες λύσεις για τον παραπάνω συνδυασμό ποσότητας Q_1 του στρατηγικού παραγωγού και ενιαίας τιμή εκκαθάρισης smp. Τα αποτελέσματα παρατίθενται στον *Πίνακα 4.1.2*. Η στήλη των μεταβλητών απόφασης Z_i μας δείχνει την κατάσταση λειτουργίας και των Q_i την ποσότητα σε MWh που συμμετέχουν.

Μονάδα (i)	Z_i	Q_i
1	1	377
2	1	3800
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	1	23

Πίνακας 4.1.2: Καταστάσεις λειτουργίας και ποσότητες με τις οποίες συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση για ζήτηση ίση με 4.200 MWh

Το παραπάνω παράδειγμά έχει μόνο μία βέλτιστη λύση, με marginal παραγωγό τον 10^{ο} παραγωγό και με ποσότητα παραγωγής μεταξύ των τεχνικών ορίων του ($0 < 23 < 141$). Συμμετέχει επίσης ο 2^{ος} παραγωγός με ποσότητα ίση με το τεχνικό του μέγιστο, 3800 MWh, ακολουθώντας τον 1^{ο} κανόνα βελτιστότητας που αναλύθηκε σε προηγούμενο κεφάλαιο. Εφόσον η τιμή/προσφορά του είναι 35 €/MWh και μικρότερη των 150€/MWh του οριακού κόστους smp, η ποσότητα του 2^{ου} παραγωγού ορθώς παίρνει τη μέγιστη τιμή της. Στη λύση επίσης φαίνεται και η τιμή προσφοράς του στρατηγικού παραγωγού P_1 που είναι 35 €/MWh.

Στη συνέχεια, για το ίδιο παράδειγμα αυξάνεται η ζήτηση στην τιμή των 5200 MWh με στόχο να αξιολογηθεί η συμπεριφορά του αλγορίθμου στη μεταβολή αυτή. Τα αποτελέσματα του αλγορίθμου για την ζήτηση αυτή δίνονται στους *Πίνακες 4.1.3* με *4.1.6.*

Λύσεις	1^{η}		2^{η}		3^{η}		4^{η}	
Μονάδα (i)	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i
1	1	377	1	377	1	377	1	377
2	1	3800	1	3800	1	3800	1	3800
3	0	0	1	476	1	476	1	476
4	0	0	0	0	0	0	0	0
5	1	384	0	0	1	384	1	384
6	0	0	0	0	0	0	1	151
7	1	188	0	0	0	0	0	0
8	1	287	1	287	0	0	0	0
9	1	144	1	144	1	144	0	0
10	1	20	1	116	1	19	1	12

Πίνακας 4.1.3: Καταστάσεις λειτουργίας και ποσότητες με τις οποίες συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση για ζήτηση ίση με 5200 MWh

Λύσεις	5^{η}		6^{η}		7^{η}		8^{η}	
Μονάδα(i)	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i
1	1	377	1	377	1	377	1	377
2	1	3800	1	3800	1	3800	1	3800

3	0	0	0	0	0	0	1	476
4	0	0	1	550	0	0	0	0
5	1	384	0	0	1	384	0	0
6	1	151	1	151	1	151	1	151
7	1	188	1	188	0	0	1	188
8	1	287	0	0	1	287	0	0
9	0	0	0	0	1	144	1	144
10	1	13	1	134	1	57	1	64

Πίνακας 4.1.4: Καταστάσεις λειτουργίας και ποσότητες με τις οποίες συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση για ζήτηση ίση με 5200 MWh

Λύσεις	**$9^{η}$**		**$10^{η}$**		**$11^{η}$**		**$12^{η}$**	
Μονάδα(i)	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i
1	1	377	1	377	1	377	1	377
2	1	3800	1	3800	1	3800	1	3800
3	1	476	0	0	0	0	0	0
4	0	0	1	550	1	550	1	550
5	0	0	1	384	0	0	0	0
6	0	0	0	0	1	151	0	0
7	1	188	0	0	0	0	1	188
8	1	287	0	0	1	287	0	0
9	0	0	0	0	0	0	1	144
10	1	72	1	89	1	35	1	141

Πίνακας 4.1.5 Καταστάσεις λειτουργίας και ποσότητες με τις οποίες συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση για ζήτηση ίση με 5200 MWh

Λύσεις	**$13^{η}$**		**$14^{η}$**	
Μονάδα(i)	Z_i	Q_i	Z_i	Q_i
1	1	377	1	377
2	1	3800	1	3800
3	0	0	1	476
4	1	550	0	0
5	0	0	0	0
6	0	0	1	151
7	0	0	0	0
8	1	287	1	287
9	1	144	0	0
10	1	42	1	109

Πίνακας 4.1.6: Καταστάσεις λειτουργίας και ποσότητες με τις οποίες συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση για ζήτηση ίση με 5200 MWh

Από τα αποτελέσματα της εφαρμογής του αλγορίθμου διαπιστώνεται ότι ο αριθμός των λύσεων μεγάλωσε δραματικά, φτάνοντας τις 14. Η αύξηση της ζήτησης οδήγησε στην αύξηση της πιθανότητας να υπάρχουν περισσότεροι συνδυασμοί καταστάσεων λειτουργίας, που μπορούν να καλύπτουν την ζήτηση σύμφωνα πάντα με τους κανόνες βελτιστότητας.

Στην $1^{η}$ λύση ο marginal παραγωγός είναι ο $10^{ος}$ με παραγόμενη ποσότητα ανάμεσα στα τεχνικά του όρια ($0 < 20 < 141$) και τις μονάδες παραγωγής 1,2,5,7,8,9 να παράγουν ποσότητες ίσες με το τεχνικό τους μέγιστο, εφόσον οι τιμές-προσφορές αυτών των μονάδων παραγωγής είναι μικρότερες του οριακού κόστους smp που είναι ίσο με 150 €/MWh όπως φαίνεται στον *Πίνακα 4.1.1*. Όπως παρατηρούμε και στις υπόλοιπες λύσεις marginal παραμένει ο 10^{ος} παραγωγός με διαφορετική παραγόμενη ποσότητα κάθε φορά και τις υπόλοιπες συμμετέχουσες μονάδες, διαφορετικές σε κάθε λύση, η παραγωγή τους να βρίσκεται στο τεχνικό τους μέγιστο. Η μόνη διαφορά έγκειται στην 12^{η} λύση, όπου η 10^{η} μονάδα παράγει ποσότητα ίση με το τεχνικό της μέγιστο και όχι ανάμεσα στα τεχνικά τις όρια. Παραμένει όμως marginal αφού η τιμή/προσφορά της είναι ίση με το οριακό κόστος smp. Και εφόσον και υπόλοιπες συμμετέχουσες μονάδες παράγουν ποσότητες ίσες με το τεχνικό τους μέγιστο επαληθεύεται ο 3^{ος} κανόνας βελτιστότητας. Η επίλυση επίσης δίνει την τιμή προσφοράς του στρατηγικού παραγωγού ίση με 35 €/MWh για κάθε λύση.

4.2 Παράδειγμα 2^{ο}

Στο παράδειγμα αυτό εξετάζεται ένα μεγαλύτερο πρόβλημα με 15 μονάδες παραγωγής ηλεκτρικής ενέργειας. Τα τεχνικά χαρακτηριστικά (τεχνικό ελάχιστο και τεχνικό μέγιστο, σε MWh) και οι τιμές/προσφορές, εκτός της τιμής/προσφοράς του στρατηγικού παραγωγού που είναι μεταβλητή απόφασης και καθορίζεται από τον αλγόριθμο (σε €/MWh) παρουσιάζονται στον *Πίνακα 4.2.1*. Το μεταβλητό κόστος c (κόστος παραγωγής) του μεμονωμένου παραγωγού είναι 30 €/MWh, και το άνω όριο στην τιμή προσφοράς είναι 90 €/MWh. Στο παράδειγμα αυτό η ζήτηση ισούται με 4500 MWh.

Μονάδα (*i*)	M_i	K_i	P_i
1	750	935	-
2	1500	3000	42
3	500	850	35
4	1200	1900	64
5	200	250	53
6	50	175	77
7	110	199	40
8	70	135	30
9	50	110	77
10	210	450	66
11	35	145	55
12	75	230	50
13	110	300	70
14	60	155	80
15	635	840	90

Πίνακας 4.2.1: Τεχνικά χαρακτηριστικά και τιμές/προσφορές των μονάδων παραγωγής

Μαζί με τα δεδομένα του *Πίνακα 5.4* εισάγουμε στον αλγόριθμο την ποσότητα που παράγει ο στρατηγικός παραγωγός και είναι ίση με 800 MWh, ποσότητα μεταξύ των τεχνικών ορίων του στρατηγικού παραγωγού (750 < 800 < 935) και το οριακό κόστος smp ίσο με 70 €/MWh. Στο παράδειγμα αυτό, είναι φανερό εξ αρχής ότι marginal παραγωγός στις τυχόν λύσεις που θα βρεθούν μπορεί να είναι μόνο ο στρατηγικός, σύμφωνα με κανόνες βελτιστότητας. Τα αποτελέσματα του αλγορίθμου δίνονται στους *Πίνακες 4.2.2* και *4.2.3*.

Λύσεις	**1η**		**2η**		**3η**		**4η**	
Μονάδα(i)	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i
1	1	800	1	800	1	800	1	800
2	0	0	1	3000	0	0	1	3000
3	1	850	0	0	0	0	0	0
4	1	1900	0	0	1	1900	0	0
5	0	0	1	250	1	250	1	250
6	1	50	0	0	1	50	1	50
7	0	0	0	0	0	0	0	0
8	0	0	0	0	1	135	0	0
9	1	50	0	0	1	50	0	0
10	1	450	1	450	1	450	0	0
11	0	0	0	0	0	0	0	0
12	1	230	0	0	1	230	1	230
13	1	110	0	0	0	0	1	110
14	1	60	0	0	0	0	1	60
15	0	0	0	0	1	635	0	0

Πίνακας 4.2.2: Καταστάσεις λειτουργίας και ποσότητες που συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση

Λύσεις	**5η**		**6η**		**7η**		**8η**	
Μονάδα(i)	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i
1	1	800	1	800	1	800	1	800
2	1	3000	1	3000	0	0	0	0
3	0	0	0	0	1	850	1	850
4	0	0	0	0	1	1900	1	1900

5	1	250	1	250	0	0	0	0
6	0	0	0	0	1	50	0	0
7	0	0	0	0	0	0	0	0
8	1	135	0	0	1	135	1	135
9	0	0	1	50	0	0	1	50
10	0	0	0	0	1	450	1	450
11	1	145	0	0	1	145	1	145
12	0	0	1	230	0	0	0	0
13	1	110	1	110	1	110	1	110
14	1	60	1	60	1	60	1	60
15	0	0	0	0	0	0	0	0

Πίνακας 4.2.3: Καταστάσεις λειτουργίας και ποσότητες που συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση.

Από του παραπάνω πίνακες των αποτελεσμάτων παρατηρείται ότι οι λύσεις που μπορούν να καλύψουν την απαιτούμενη ζήτηση είναι 8. Σε όλες τις λύσεις όπως ήταν αναμενόμενο marginal είναι ο στρατηγικός παραγωγός με παραγόμενη ποσότητα 800 MWh όπως εισάχθηκε στα δεδομένα του προβλήματος. Επίσης και στις 8 λύσεις, οι υπόλοιπες συμμετέχουσες μονάδες ανάλογα με τιμή/προσφορά τους παράγουν ποσότητα είτε στο τεχνικό τους ελάχιστο είτε στο μέγιστο.

Ενδεικτικά θα αναλυθεί η 1^{η} λύση. Οι μονάδες που συμμετέχουν εκτός του στρατηγικού παραγωγού που είναι ο marginal είναι οι εξής: 3^{η}, 4^{η}, 6^{η}, 9^{η}, 10^{η}, 12^{η}, 13^{η} και η 14^{η}. Από αυτές οι μονάδες με μικρότερη τιμή/προσφορά από τα 70 €/MWh παράγουν ποσότητα ίση με το τεχνικό τους μέγιστο και είναι οι εξής : 3^{η}, 4^{η}, 10^{η} και η 12^{η}. Η 6^{η}, η 9^{η} και η 14^{η} με τιμή προσφορά μεγαλύτερη από τα 70 €/MWh παράγουν ποσότητα ίση με το τεχνικό τους ελάχιστο. Υπάρχει όμως και η 13^{η} μονάδα παραγωγής που έχει τιμή/προσφορά ίση με το οριακό κόστος, η μονάδα αυτή έχει την δυνατότητα να παράξει ποσότητα είτε ίση με το τεχνικό της ελάχιστο είτε με το μέγιστο. Στην προκειμένη περίπτωση ο αλγόριθμος την ώθησε στο τεχνικό της ελάχιστο.

4.3 Παράδειγμα 3^{ο}

Στο παράδειγμα αυτό εξετάζουμε ένα πρόβλημα με 20 μονάδες παραγωγής ηλεκτρικής ενέργειας. Τα τεχνικά χαρακτηριστικά (τεχνικό ελάχιστο και τεχνικό μέγιστο, σε MWh) και οι τιμές προσφοράς, εκτός της τιμής προσφοράς του στρατηγικού παραγωγού που είναι μεταβλητή απόφασης και καθορίζεται από τον αλγόριθμο, (σε €/MWh) παρουσιάζονται στον *Πίνακα 4.3.1*. Το μεταβλητό κόστος (κόστος παραγωγής) του μεμονωμένου παραγωγού είναι 20 €/MWh, και το άνω όριο στην τιμή προσφοράς είναι 65 €/MWh. Το παράδειγμα αυτό εφαρμόζεται για ζήτηση 233 MWh.

Μονάδα (i)	M_i	K_i	P_i
1	53	89	-

2	53	64	27
3	58	79	33
4	43	74	51
5	44	81	44
6	61	95	20
7	33	86	30
8	15	93	25
9	22	103	41
10	77	134	53
11	65	143	21
12	53	98	55
13	54	77	61
14	81	204	29
15	33	151	39
16	22	57	48
17	51	89	65
18	13	92	32
19	65	149	44
20	77	201	55

Πίνακας 4.3.1: Τεχνικά χαρακτηριστικά και τιμές προσφορές των μονάδων παραγωγής

Μαζί με τα παραπάνω δεδομένα του *Πίνακα 4.3.1* εισάγουμε στον αλγόριθμο την ποσότητα που παράγει ο στρατηγικός παραγωγός ίση με 53 MWh, ποσότητα ίση με το τεχνικό ελάχιστο του στρατηγικού παραγωγού και το οριακό κόστος smp ίσο με 61 €/MWh. Τα αποτελέσματα του αλγορίθμου δίνονται στον *Πίνακα 4.3.2*.

Λύσεις	**1η**		**2η**		**3η**		**4η**	
Μονάδα (i)	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i
1	1	53	1	53	1	53	1	53
2	0	0	0	0	1	64	1	64
3	0	0	0	0	0	0	0	0
4	1	74	0	0	0	0	0	0

5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	1	55	1	72	1	65	1	59
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	1	57	0	0	1	57
17	1	51	1	51	1	51	0	0
18	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0

Πίνακας 4.3.2: Καταστάσεις λειτουργίας και ποσότητες που συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση

Στον παραπάνω πίνακα των αποτελεσμάτων βλέπουμε ότι οι λύσεις που δόθηκαν από τον αλγόριθμο είναι 4. Παρατηρούμε ότι και στις 4 λύσεις marginal είναι η 13η μονάδα παραγωγής, διότι και στις 4 λύσεις η ποσότητα που παράγει είναι ανάμεσα (διαφορετική και στις 4 λύσεις) από τα τεχνικά της όρια. Οι υπόλοιπες τώρα μονάδες παραγωγής που συμμετέχουν σε κάθε λύση, ανάλογα με τιμή-προσφορά τους παράγουν ποσότητα είτε στο τεχνικό τους ελάχιστο είτε στο μέγιστο.

Ενδεικτικά θα αναλύσουμε την 3η λύση. Οι μονάδες που συμμετέχουν εκτός της 13ης που είναι ο marginal είναι οι εξής: 1η, 2ηκαι η 17η. Από αυτές η μονάδα με μικρότερη τιμή-προσφορά από τα 61 €/MWh παράγει ποσότητα ίση με το τεχνικό της μέγιστο και είναι η 2η. Η 17η με τιμή-προσφορά μεγαλύτερη από τα 61 €/MWh παράγει ποσότητα ίση με το τεχνικό της ελάχιστο. Υπάρχει όμως και ο στρατηγικός παραγωγός (1η μονάδα παραγωγής) που η τιμή-προσφορά του δίνεται από την λύση και είναι ίση με το οριακό κόστος, η μονάδα αυτή έχει την δυνατότητα να παράξει ποσότητα είτε ίση με το τεχνικό της ελάχιστο είτε με το μέγιστο. Στην προκειμένη περίπτωση την θέσαμε εξ' αρχής στην εισαγωγή των δεδομένων στο τεχνικό της ελάχιστο.

4.4 Παράδειγμα 4ο

Στο παράδειγμα αυτό εξετάζουμε μικρό πρόβλημα με 4 μονάδες παραγωγής ηλεκτρικής ενέργειας. Τα τεχνικά χαρακτηριστικά (τεχνικό ελάχιστο και τεχνικό μέγιστο, σε MWh) και οι τιμές προσφοράς, εκτός της τιμής προσφοράς του στρατηγικού παραγωγού που είναι μεταβλητή απόφασης και καθορίζεται από τον αλγόριθμο, (σε €/MWh) παρουσιάζονται στον *Πίνακα 4.4.1*. Το μεταβλητό κόστος (κόστος παραγωγής) του μεμονωμένου παραγωγού είναι 40 €/MWh, και το άνω όριο στην τιμή προσφοράς είναι 100 €/MWh. Το παράδειγμα αυτό το εφαρμόζουμε για ζήτηση 1050 MWh.

Μονάδα (i)	M_i	K_i	P_i
1	240	400	-
2	200	500	60
3	100	300	69
4	150	350	56

Πίνακας 4.4.1: Τεχνικά χαρακτηριστικά και τιμές προσφορές των μονάδων παραγωγής

Μαζί με τα παραπάνω δεδομένα του *Πίνακα 5.8* εισάγουμε στον αλγόριθμο την ποσότητα που παράγει ο στρατηγικός παραγωγός ίση με 400 MWh, ποσότητα ίση με το τεχνικό μέγιστο του στρατηγικού παραγωγού και το οριακό κόστος SMP ίσο με 100 €/MWh. Τα αποτελέσματα του αλγορίθμου δίνονται στον *Πίνακα 4.4.2*.

Μονάδα (i)	Z_i	Q_i
1	1	400
2	0	0
3	1	300
4	1	350

Πίνακας 4.4.2: Καταστάσεις λειτουργίας και ποσότητες που συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση

Ο αλγόριθμος για το παραπάνω παράδειγμα μας έδωσε μόνο μία λύση. Η τιμή-προσφορά του στρατηγικού παραγωγού είναι ίση με 100 €/MWh όσο το οριακό κόστος SMP. Συνεπώς, εφόσον δεν υπάρχει άλλη μονάδα παραγωγής με τιμή-προσφορά ίση με το οριακό κόστος, ο στρατηγικός παραγωγός είναι ο marginal. Για το λόγο αυτό και σύμφωνα με την συνθήκη βελτιστότητας οι υπόλοιπες συμμετέχουσες στη λύση μονάδες παράγουν ποσότητες ίσες με το τεχνικό τους μέγιστο.

4.5 Παράδειγμα 5^{ο}

Στο παράδειγμα αυτό εξετάζουμε ένα μεγάλης κλίμακας πρόβλημα με τον αριθμό των μονάδων παραγωγής να ανεβαίνει στις 40. Τα τεχνικά χαρακτηριστικά (τεχνικό ελάχιστο, τεχνικό μέγιστο, σε MWh) και οι τιμές προσφοράς, εκτός της τιμής προσφοράς του στρατηγικού παραγωγού που είναι μεταβλητή απόφασης και καθορίζεται από τον αλγόριθμο, (σε €/MWh) παρουσιάζονται στον Πίνακα 4.5.1. Το μεταβλητό κόστος (κόστος παραγωγής) του μεμονωμένου παραγωγού είναι 30 €/MWh, και το άνω όριο στην τιμή προσφοράς είναι 150 €/MWh. Το παράδειγμα αυτό το εφαρμόζουμε για ζήτηση 2000 MWh.

Παράλληλα με τα δεδομένα του *Πίνακα 4.5,1* εισάγουμε στον αλγόριθμο την ποσότητα που παράγει ο στρατηγικός παραγωγός ίση με 1563 MWh, ποσότητα ίση με το τεχνικό του μέγιστο και το οριακό κόστος *smp* ίσο με 56 €/MWh. Τα αποτελέσματα του αλγορίθμου δίνονται στον *Πίνακα 4.5.2*.

Μονάδα (i)	M_i	K_i	P_i
1	433	1563	-
2	183	833	30
3	152	432	34
4	165	578	32
5	177	234	39
6	234	543	41
7	132	556	49
8	111	1187	52
9	357	1732	56
10	44	124	62
11	122	256	59
12	135	345	67
13	168	721	61
14	189	652	74
15	311	432	79
16	515	1050	72
17	135	899	69
18	165	677	58
19	433	567	80
20	211	411	82
21	111	356	85
22	532	1589	87
23	190	1120	92
24	577	786	99
25	344	564	81
26	89	198	93
27	173	253	95
28	123	432	97
29	234	431	102
30	345	456	107
31	68	157	101
32	195	231	109
33	141	532	111
34	189	344	113
35	37	121	112
36	92	289	120
37	478	1560	133
38	567	2000	129
39	124	789	144
40	354	567	150

Πίνακας 4.5.1: Τεχνικά χαρακτηριστικά και τιμές προσφορές των μονάδων παραγωγής

Λύσεις	$1^η$		$2^η$		$3^η$		$4^η$	
Μονάδα(i)	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i	Z_i	Q_i
1	1	1563	1	1563	1	1563	1	1563
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	1	0	1	0	1	0	1	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0
9	0	437	1	393	1	400	1	369
10	0	0	1	44	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	1	68
32	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0
35	0	0	0	0	1	37	0	0
36	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0

Πίνακας 4.5.2: Καταστάσεις λειτουργίας και ποσότητες που συμμετέχουν οι μονάδες παραγωγής στη βέλτιστη λύση

Στον παραπάνω πίνακα των αποτελεσμάτων παρατηρείται ότι οι λύσεις που δόθηκαν από τον αλγόριθμο είναι 4. Παρατηρούμε ότι και στις 4 λύσεις marginal είναι η $9^η$ μονάδα παραγωγής, διότι και στις 4 λύσεις η ποσότητα που παράγει είναι ανάμεσα (διαφορετική και στις 4 λύσεις) από τα τεχνικά της όρια. Οι υπόλοιπες τώρα μονάδες παραγωγής που συμμετέχουν σε κάθε λύση, ανάλογα με τιμή-προσφορά τους παράγουν ποσότητα είτε στο τεχνικό τους ελάχιστο είτε στο μέγιστο.

Η διαφορά του παραδείγματος αυτού σε σχέση με τα υπόλοιπα είναι στην τιμή προσφορά του στρατηγικού παραγωγού που δίνεται στις 4 λύσεις. Στην $1^{η}$ λύση είναι ίση με 56 €/MWh ίση με το οριακό κόστος, ενώ στις υπόλοιπες τρεις 30 €/MWh. Στα παραπάνω παραδείγματα που εξετάσαμε δεν συνέβαινε αυτό, με την τιμή-προσφορά του στρατηγικού παραγωγού να είναι κοινή σε όλες τις λύσεις του εκάστοτε παραδείγματος.

Ενδεικτικά θα αναλύσουμε την $4^{η}$ λύση. Οι μονάδες που συμμετέχουν εκτός της $9^{ης}$ που είναι ο marginal είναι η $1^{η}$ και η $31^{η}$. Ο στρατηγικός παραγωγός ($1^{η}$ μονάδα παραγωγής) έχει μικρότερη τιμή-προσφορά από τα 56 €/MWh του οριακού κόστους και παράγει ποσότητα ίση με το τεχνικό του μέγιστο (δεδομένο) όπως προϋποθέτουν οι κανόνες βελτιστότητας. Η $31^{η}$ με τιμή-προσφορά μεγαλύτερη από τα 56 €/MWh παράγει ποσότητα ίση με το τεχνικό της ελάχιστο.

5. Συμπεράσματα και Προτάσεις

Σε αυτή την εργασία παρουσιάστηκε ένα μεικτό ακέραιο μοντέλο βελτιστοποίησης δύο επιπέδων, το οποίο χρησιμοποιείται για τη βέλτιστη στρατηγική υποβολής προσφορών ενεργειακών παραγωγών στις αγορές ημερήσιου προγραμματισμού ηλεκτρικής ενέργειας. Η διαφορά του μοντέλου που αναπτύχθηκε, σε σχέση με παρόμοια μοντέλα που έχουν μελετηθεί στη σχετική βιβλιογραφία, είναι η εισαγωγή της ποσότητας που επιθυμεί ο στρατηγικός παραγωγός και της τιμής εκκαθάρισης του συστήματος ως δεδομένα στον αλγόριθμο, ενώ το αποτέλεσμα είναι οι συνδυασμοί ενεργών παραγωγών που οδηγούν σε συγκεκριμένο βέλτιστο.

Η διαδικασία αυτή ακολουθεί την λογική των συνθηκών βελτιστότητας που αναλύθηκαν. Οι συνθήκες βελτιστότητας έπαιξαν τον σημαντικότερο ρόλο στην επίλυση αυτού του προβλήματος. Παρόλο που μπορεί να υπάρξει βέλτιστη λύση με διαφορετικά χαρακτηριστικά από αυτά των συνθηκών βελτιστότητας, αποδείχτηκε ότι πάντοτε θα υπάρχουν βέλτιστες λύσεις που επαληθεύουν στις συνθήκες αυτές. Για το λόγο αυτό, ο αλγόριθμος βασίστηκε στις συνθήκες βελτιστότητας, έτσι ώστε να περιοριστεί το εύρος των πιθανών λύσεων χωρίς να επηρεάζεται η βελτιστότητα, περιορίζοντας ταυτόχρονα τόσο τη δυσκολία του προβλήματος όσο και το χρόνο εύρεσης μίας βέλτιστης λύσης.

Αναλύθηκαν παραδείγματα με διαφορετικό αριθμό μονάδων παραγωγών κάθε φορά, από 4 μέχρι 40, και με διαφορετικό αριθμό λύσεων που κυμαίνεται μεταξύ 1 και 14 χωρίς όμως αυτό να επηρεάζει την ταχύτητα επίλυσης του αλγορίθμου με διαφορές μικρότερες του δευτερολέπτου.

Σε ανάλογες περιπτώσεις, όπως η επίλυση του προβλήματος με την χρήση επίπεδων τομών, η δυσκολία της μεθόδου, περιόριζε την εφαρμογή του παραβλήματος σε μικρό αριθμό μονάδων παραγωγών. Η διαδικασία αυτή ήταν αρκετά χρονοβόρα σε αντίθεση με την μεθοδολογία που παρουσιάστηκε στην παρούσα εργασία η οποία μπορεί να εφαρμοστεί σε μεγάλο όγκο δεδομένων χωρίς προβλήματα.

Αντικείμενο μελλοντικής έρευνας μπορεί να αποτελέσει η εφαρμογή των συνθηκών βελτιστότητας και η ανάπτυξη ενός αλγορίθμου για το πρόβλημα της αγοράς ημερήσιου προγραμματισμού ηλεκτρική ενέργειας με τρόπο αποζημίωσης και τιμής εκκαθάρισης pay-as-bid. Στην συγκεκριμένη μέθοδο, δεν υπάρχει οριακό κόστος *smp* και η αποζημίωση γίνεται βάση της τιμής/προσφοράς που καταθέτει η κάθε συμμετέχουσα μονάδα παραγωγής.

Παράρτημα Α

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

double          *M,*K,*Q,*Z,*P,*Y,*Zi;
int             D,c,PMAX,N,SMP,Qi,Q2,Z2,K2,M2,P2,I,J;

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

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

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

        return true;
}
```

// Προσθήκη μεταβλητών απόφασης Q_i

```
bool  Add_Vars_Q()
{
	int i,status;
	double obj, lb, ub;
	char type;
	obj = 0;
	lb = 0;
	ub = 3800;
	type = CPX_INTEGER ;

	int count;
	for(i = 0; i < N; i++) {
		status = CPXnewcols (Env, Lp, 1, &obj, &lb, &ub, &type, NULL);
		Q[i]=CPXgetnumcols (Env, Lp)-1;
		if (status != 0){
			print_error("CPXnewcols_Add_Vars_Q");
			return false;
		}
	}
	return true;
}
```

// Προσθήκη μεταβλητών απόφασης Z_i

```
bool  Add_Vars_Z()
{
	int i,status;
	double obj, lb, ub;
	char type;
	obj = 0;
	lb = 0;
	ub = 1;
	type = CPX_BINARY;

	for(i = 0; i < N; i++) {
		status = CPXnewcols (Env, Lp, 1, &obj, &lb, &ub, &type,NULL);
		Z[i]=CPXgetnumcols (Env, Lp)-1;
		if (status != 0){
			print_error("CPXnewcols_Add_Vars_Z");
			return false;
		}
	}
	return true;
}
```

// Προσθήκη μεταβλητής απόφασης P_2 (τιμή προσφορά του στρατηγικού παραγωγού

```
bool  Add_Var_P2()
{
	int i,status;
	double obj, lb, ub;
	char type;
	obj = 0;
	lb = c;
	ub = PMAX;
	type = CPX_INTEGER;


	status = CPXnewcols (Env, Lp, 1, &obj, &lb, &ub, &type,NULL);
	P2=CPXgetnumcols (Env, Lp)-1;
	if (status != 0){
		print_error("CPXnewcols_Add_Var_P2");
		return false;
	}

	return true;
}
```

// Προσθήκη βοηθητικών μεταβλητών απόφασης Y

```
bool  Add_Vars_Y()
{
	int i,status;
	double obj, lb, ub;
	char type;
	obj = 0;
	lb = 0;
	ub = 1;
	type = CPX_BINARY;

	for(i = 0; i < 2*N; i++) {
		status = CPXnewcols (Env, Lp, 1, &obj, &lb, &ub, &type,NULL);
		Y[i]=CPXgetnumcols (Env, Lp)-1;
		if (status != 0){
			print_error("CPXnewcols_Add_Vars_Y");
			return false;
		}
	}
	return true;
}
```

// Προσθήκη μεταβλητής απόφασης J (δείχνει αν ο marginal είναι στο μέγιστο)

```
bool  Add_Var_J()
{
	int i,status;
	double obj, lb, ub;
	char type;
	obj = 0;
	lb = 0;
	ub = 1;
	type = CPX_BINARY;


	status = CPXnewcols (Env, Lp, 1, &obj, &lb, &ub, &type,NULL);
	J=CPXgetnumcols (Env, Lp)-1;
	if (status != 0){
		print_error("CPXnewcols_Add_Var_J");
		return false;
	}

	return true;
}
```

// Περιορισμός της ζήτησης

```
bool demand()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='E';
	rmatbeg=0;
	rhs=D-Qi;
	nzcnt=N;
	rcnt=1;
	ccnt=0;

	int *rmatind=NULL;
	rmatind = (int*) malloc((N) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_demand");
		return false;
	}

	double *rmatval=NULL;

	rmatval = (double*) malloc((N) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_demand");
```

```
		return false;
	}

	for(i = 0; i < N; i++){
		rmatind[i]=Q[i];
		rmatval[i]=1;
	}

	status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense, &rmatbeg,
	rmatind, rmatval,NULL, NULL);

	if (status != 0){
		print_error("CPXaddrows_demand");
		return false;
	}
	free(rmatind);
	free(rmatval);
	return true;
}
```

// Κάτω όριο των Q_i

```
bool Lower_Bound_Q()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='L';
	rmatbeg=0;
	rhs=0;
	nzcnt=2;
	rcnt=1;
	ccnt=0;

	int *rmatind=NULL;
	rmatind = (int*) malloc((2) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_Lower_Bound_Q");
		return false;
	}

	double *rmatval=NULL;

	rmatval = (double*) malloc((2) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_Lower_Bound_Q");
		return false;
	}

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

```
		rmatind[0]=Z[i];
		rmatval[0]=M[i];

		rmatind[1]=Q[i];
		rmatval[1]=-1;

		status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
		&rmatbeg, rmatind, rmatval,NULL, NULL);

		if (status != 0){
			print_error("CPXaddrows_Lower_Bound_Q");
			return false;
		}
	}
	free(rmatind);
	free(rmatval);
	return true;
}
```

// Άνω όριο των Q_i

```
bool Upper_Bound_Q()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='G';
	rmatbeg=0;
	rhs=0;
	nzcnt=2;
	rcnt=1;
	ccnt=0;


	int *rmatind=NULL;
	rmatind = (int*) malloc((2) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_Upper_Bound_Q");
		return false;
	}

	double *rmatval=NULL;

	rmatval = (double*) malloc((2) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_Upper_Bound_Q");
		return false;
	}

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

```
		rmatind[0]=Z[i];
		rmatval[0]=K[i];

		rmatind[1]=Q[i];
		rmatval[1]=-1;

		status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
		&rmatbeg, rmatind, rmatval,NULL, NULL);

		if (status != 0){
			print_error("CPXaddrows_Upper_Bound_Q");
			return false;
		}
	}
	free(rmatind);
	free(rmatval);
	return true;
}
```

// Το Z του marginal είναι ίσο με 1

```
bool Zmarg_EQU_1()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='E';
	rmatbeg=0;
	rhs=1;
	nzcnt=1;
	rcnt=1;
	ccnt=0;


	int *rmatind=NULL;
	rmatind = (int*) malloc((1) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_Zmarg_EQU_1");
		return false;
	}

	double *rmatval=NULL;

	rmatval = (double*) malloc((1) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_Zmarg_EQU_1");
		return false;
	}
```

```
		rmatind[0]=Z[I];
		rmatval[0]=1;

		status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense, &rmatbeg,
		rmatind, rmatval,NULL, NULL);

		if (status != 0){
			print_error("CPXaddrows_Zmarg_EQU_1");
			return false;
		}
		free(rmatind);
		free(rmatval);
		return true;
}
```

// Ο στρατηγικός παραγωγός είναι marginal

// Η τιμή/προσφορά του στρατηγικού είναι ίση με το οριακό κόστος

```
bool P2_EQU_SMP()
{
		int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
		double rhs;
		char sense;
		sense='E';
		rmatbeg=0;
		rhs=SMP;
		nzcnt=1;
		rcnt=1;
		ccnt=0;


		int *rmatind=NULL;
		rmatind = (int*) malloc((1) *sizeof(int));
		if(rmatind == NULL) {
			print_error("Cannot allocate variable pointers_P2_EQU_SMP");
			return false;
		}

		double *rmatval=NULL;

		rmatval = (double*) malloc((1) *sizeof(double));
		if(rmatval == NULL) {
			print_error("Cannot allocate coefficients_P2_EQU_SMP");
			return false;
		}


		rmatind[0]=P2;
		rmatval[0]=1;
```

```
        status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense, &rmatbeg,
        rmatind, rmatval,NULL, NULL);

        if (status != 0){
                print_error("CPXaddrows_P2_EQU_SMP");
                return false;
        }
        free(rmatind);
        free(rmatval);
        return true;
}
```

// Ο στρατηγικός παραγωγός marginal με ποσότητα ανάμεσα στα τεχνικά όρια

```
bool MARG_US_Between()
{
        int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
        double rhs;
        char sense;
        sense='E';
        rmatbeg=0;

        nzcnt=2;
        rcnt=1;
        ccnt=0;
        rhs=0;

        int *rmatind=NULL;
        rmatind = (int*) malloc((3) *sizeof(int));
        if(rmatind == NULL) {
                print_error("Cannot allocate variable pointers_MARG_US_Between");
                return false;
        }

        double *rmatval=NULL;

        rmatval = (double*) malloc((3) *sizeof(double));
        if(rmatval == NULL) {
                print_error("Cannot allocate coefficients_MARG_US_Between");
                return false;
        }

        for(i = 0; i < N; i++){
                if (P[i] < SMP){
                        rmatind[0]=Q[i];
                        rmatval[0]=1;
```

```
				rmatind[1]=Z[i];

				rmatval[1]=-K[i];

				status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
					&rmatbeg, rmatind, rmatval,NULL, NULL);

				if (status != 0){
					print_error("CPXaddrows_MARG_US_Between");
					return false;
				}
			}
			if (P[i] > SMP){
				rmatind[0]=Q[i];
				rmatval[0]=1;

				rmatind[1]=Z[i];

				rmatval[1]=-M[i];

				status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
				&rmatbeg, rmatind, rmatval,NULL, NULL);

				if (status != 0){
					print_error("CPXaddrows_MARG_US_Between");
					return false;
				}
			}
			if (P[i] == SMP){
				nzcnt = 3;

				rmatind[0]=Q[i];
				rmatval[0]=1;

				rmatind[1]=Y[2*i];
				rmatval[1]=-M[i];

				rmatind[2]=Y[2*i+1];
				rmatval[2]=-K[i];

				status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
				&rmatbeg, rmatind, rmatval,NULL, NULL);

				if (status != 0){
					print_error("CPXaddrows_MARG_US_Between");
					return false;
				}
			}

		}
```

```
	free(rmatind);
	free(rmatval);
	return true;
}
```

// Βοηθητικοί περιορισμοί Y

```
bool Add_coef_Y_EQU_Z()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='E';
	rmatbeg=0;

	nzcnt = 3;
	rcnt=1;
	ccnt=0;
	rhs=0;


	int *rmatind=NULL;
	rmatind = (int*) malloc((3) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable
		pointers_Add_coef_Y_EQU_Z");
		return false;
	}

	double *rmatval=NULL;

	rmatval = (double*) malloc((3) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_Add_coef_Y_EQU_Z");
		return false;
	}

	for(i = 0; i < N; i++){
		if (i==I) continue;
		if (P[i] == SMP){

			rmatind[0]=Z[i];
			rmatval[0]=1;

			rmatind[1]=Y[2*i];
			rmatval[1]=-1;

			rmatind[2]=Y[2*i+1];
			rmatval[2]=-1;
```

```
			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_Add_coef_Y_EQU_Z");
				return false;
			}
		}

	}
	free(rmatind);
	free(rmatval);
	return true;
}
```

// Ο στρατηγικός παραγωγός marginal με ποσότητα στο ελάχιστο

```
bool MARG_US_Min()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='E';
	rmatbeg=0;

	nzcnt=2;
	rcnt=1;
	ccnt=0;
	rhs=0;


	int *rmatind=NULL;
	rmatind = (int*) malloc((3) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_MARG_US_Min");
		return false;
	}

	double *rmatval=NULL;
	rmatval = (double*) malloc((3) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_MARG_US_Min");
		return false;
	}

	for(i = 0; i < N; i++){
		if (P[i] < SMP){
			rmatind[0]=Q[i];
			rmatval[0]=1;
```

```
			rmatind[1]=Z[i];

			rmatval[1]=-K[i];

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_MARG_US_Min");
				return false;
			}
		}
		if (P[i] > SMP){
			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Z[i];
			rmatval[1]=-M[i];

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_MARG_US_Min");
				return false;
			}
		}
		if (P[i] == SMP){
			nzcnt = 3;

			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Y[2*i];
			rmatval[1]=-M[i];

			rmatind[2]=Y[2*i+1];
			rmatval[2]=-K[i];

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_MARG_US_Min");
				return false;
			}
		}

	}
	free(rmatind);
```

```
	free(rmatval);
	return true;
}
```

// Ο στρατηγικός παραγωγός marginal με ποσότητα στο μέγιστο

```
bool MARG_US_MAX()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='E';
	rmatbeg=0;

	nzcnt=2;
	rcnt=1;
	ccnt=0;
	rhs=0;

	int *rmatind=NULL;
	rmatind = (int*) malloc((2) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_MARG_US_MAX");
		return false;
	}


	double *rmatval=NULL;

	rmatval = (double*) malloc((2) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_MARG_US_MAX");
		return false;
	}

	for(i = 0; i < N; i++){
		if (P[i] <= SMP){
			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Z[i];
			rmatval[1]=-K[i];

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_MARG_US_MAX");
				return false;
			}
```

```
		}else {
			nzcnt=1;
			rmatind[0]=Z[i];
			rmatval[0]=1;

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);
		}
	}
	free(rmatind);
	free(rmatval);
	return true;
}
```

// Ο στρατηγικός παραγωγός στο ελάχιστο με τον marginal είτε ανάμεσα είτε στο ελάχιστο

```
bool Qmarg_Less_Qmax()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='G';
	rmatbeg=0;
	rhs=0;
	nzcnt=2;
	rcnt=1;
	ccnt=0;


	int *rmatind=NULL;
	rmatind = (int*) malloc((2) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_Qmarg_Less_Qmax");
		return false;
	}

	double *rmatval=NULL;

	rmatval = (double*) malloc((2) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_Qmarg_Less_Qmax");
		return false;
	}


	rmatind[0]=Z[I];
	rmatval[0]=K[I]-1;
```

```
	rmatind[1]=Q[I];
	rmatval[1]=-1;

	 status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense, &rmatbeg,
	rmatind, rmatval,NULL, NULL);

	if (status != 0){
		print_error("CPXaddrows_Qmarg_Less_Qmax");
		return false;
	}
	free(rmatind);
	free(rmatval);
	return true;
}
```

// Η τιμή/προσφορά του στρατηγικού μεγαλύτερη από αυτή του marginal

```
bool P2_greater_Pmarg()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='G';
	rmatbeg=0;
	rhs=P[I];
	nzcnt=1;
	rcnt=1;
	ccnt=0;


	int *rmatind=NULL;
	rmatind = (int*) malloc((1) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_P2_greater_Pmarg");
		return false;
	}

	double *rmatval=NULL;
	rmatval = (double*) malloc((1) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_P2_greater_Pmarg");
		return false;
	}


	rmatind[0]=P2;
	rmatval[0]=1;
```

```
	status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense, &rmatbeg,
	rmatind, rmatval,NULL, NULL);

	if (status != 0){
		print_error("CPXaddrows_P2_greater_Pmarg");
		return false;
	}
	free(rmatind);
	free(rmatval);
	return true;
}
```

// Περιορισμοί βάση του marginal

```
bool Add_coef_MARG()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='E';
	rmatbeg=0;

	nzcnt=2;
	rcnt=1;
	ccnt=0;
	rhs=0;

	int *rmatind=NULL;
	rmatind = (int*) malloc((3) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_Add_coef_MARG");
		return false;
	}


	double *rmatval=NULL;
	rmatval = (double*) malloc((3) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_Add_coef_MARG");
		return false;
	}

	for(i = 0; i < N; i++){
		if (i==I) continue;
		if (P[i] < SMP){
			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Z[i];
			rmatval[1]=-K[i];
```

```
			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_Add_coef_MARG");
				return false;
			}
		}

		if (P[i] > SMP){
			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Z[i];
			rmatval[1]=-M[i];

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_Add_coef_MARG");
				return false;
			}
		}

		if (P[i] == SMP){
			nzcnt = 3;

			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Y[2*i];
			rmatval[1]=-M[i];

			rmatind[2]=Y[2*i+1];
			rmatval[2]=-K[i];

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_Add_coef_MARG");
				return false;
			}
		}
	}

	if (status != 0){
		print_error("CPXaddrows_Add_coef_MARG");
```

```
		return false;
	}

	free(rmatind);
	free(rmatval);
	return true;
}
```

// Περιορισμοί βάση του marginal, όταν είναι στο μέγιστο

```
bool Add_coef_MARG_Qmax()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='E';
	rmatbeg=0;

	nzcnt=2;
	rcnt=1;
	ccnt=0;
	rhs=0;


	int *rmatind=NULL;
	rmatind = (int*) malloc((4) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_Add_coef_MARG");
		return false;
	}

	double *rmatval=NULL;
	rmatval = (double*) malloc((4) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_Add_coef_MARG");
		return false;
	}

	for(i = 0; i < N; i++){
		if (i==I) continue;
		if (P[i] < SMP){
			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Z[i];
			rmatval[1]=-K[i];

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);
```

```
			if (status != 0){
				print_error("CPXaddrows_Add_coef_MARG");
				return false;
			}
		}

		if (P[i] > SMP){
			nzcnt=3;

			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Z[i];
			rmatval[1]=-M[i];

			rmatind[2]=J;
			rmatval[2]=-(D+1);

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_Add_coef_MARG");
				return false;
			}
		}

		if (P[i] == SMP){
			nzcnt = 4;
			rmatind[0]=Q[i];
			rmatval[0]=1;

			rmatind[1]=Y[2*i];
			rmatval[1]=-M[i];

			rmatind[2]=Y[2*i+1];
			rmatval[2]=-K[i];

			rmatind[3]=J;
			rmatval[3]=-(D+1);

			status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
			&rmatbeg, rmatind, rmatval,NULL, NULL);

			if (status != 0){
				print_error("CPXaddrows_Add_coef_MARG");
				return false;
			}
		}
	}
```

```
	if (status != 0){
		print_error("CPXaddrows_Add_coef_MARG");
		return false;
	}

	free(rmatind);
	free(rmatval);
	return true;
}
```

// Η τιμή/προσφορά του στρατηγικού μικρότερη από αυτή του marginal

```
bool P2_lower_Pmarg()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='L';
	rmatbeg=0;
	rhs=P[I];
	nzcnt=1;
	rcnt=1;
	ccnt=0;


	int *rmatind=NULL;
	rmatind = (int*) malloc((1) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_P2_lower_Pmarg");
		return false;
	}

	double *rmatval=NULL;
	rmatval = (double*) malloc((1) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_P2_lower_Pmarg");
		return false;
	}
		rmatind[0]=P2;
		rmatval[0]=1;

		status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
		&rmatbeg, rmatind, rmatval,NULL, NULL);

		if (status != 0){
			print_error("CPXaddrows_P2_lower_Pmarg");
			return false;
		}
```

```
	free(rmatind);
	free(rmatval);
	return true;
}
```

// J ≤ Qmarg/Qmax

```
bool J_lower()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='L';
	rmatbeg=0;
	rhs=0;
	nzcnt=2;
	rcnt=1;
	ccnt=0;

	int *rmatind=NULL;
	rmatind = (int*) malloc((2) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_J_lower");
		return false;
	}

	double *rmatval=NULL;

	rmatval = (double*) malloc((2) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_J_lower");
		return false;
	}

		rmatind[0]=J;
		rmatval[0]=K[I];

		rmatind[1]=Q[I];
		rmatval[1]=-1;

		status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
		&rmatbeg, rmatind, rmatval,NULL, NULL);

		if (status != 0){
			print_error("CPXaddrows_J_lower");
			return false;
		}

	free(rmatind);
```

```
	free(rmatval);
	return true;
}
```

// J ≥ Qmarg-(Qmax-1)/(Qmax-1)

```
bool J_Upper()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='G';
	rmatbeg=0;
	rhs=-(K[I]-1);
	nzcnt=2;
	rcnt=1;
	ccnt=0;

	int *rmatind=NULL;
	rmatind = (int*) malloc((2) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_J_Upper");
		return false;
	}

	double *rmatval=NULL;
	rmatval = (double*) malloc((2) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_J_Upper");
		return false;
	}

		rmatind[0]=J;
		rmatval[0]=(K[I]-1);

		rmatind[1]=Q[I];
		rmatval[1]=-1;

		status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense,
		&rmatbeg, rmatind, rmatval,NULL, NULL);

		if (status != 0){
			print_error("CPXaddrows_J_Upper");
			return false;
		}

	free(rmatind);
	free(rmatval);
	return true;
}
```

// Επίλυση προβλήματος

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

// Εκτύπωση προβλήματος

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

// Έλεγχος του αποτελέσματος της βελτιστοποίησης

```
bool lp_stat()
{
	int lpstat;
	lpstat = CPXgetstat (Env, Lp);
	if (lpstat==0){
		print_error("CPXgetstat");
		return false;
	}
	if (lpstat==101){
		printf("Optimal integer solution found\n");
	}
	if (lpstat==103){
		printf("Integer infeasible\n");
}
	if (lpstat==118){
		printf("Node file size limit, integer solution exists\n");
	}
	 return true;
}
```

// **Εμφάνιση της λύσης του προβλήματος**

```
bool  PrintSolStatus()
{
	int status,i;
	double *sol;

	sol = (double*)malloc(CPXgetnumcols (Env, Lp)*sizeof(double));
	status = CPXgetx (Env, Lp, sol, 0, CPXgetnumcols(Env, Lp)-1);
	if (status != 0) {
		 print_error("infeasible");
		 return false;
	}

	for (i = 0; i < CPXgetnumcols (Env, Lp); i++) {
		if (sol[i]==0) continue;
		printf ( "X%d:  Value = %17.10g\n", i+1, sol[i]);
	}

	for (i = N; i < 2*N; i++) {
		Zi[i-N] = sol[i];
	}

	free(sol);
	return true;
}
```

// **Εισαγωγή περιορισμών CUT**

```
bool cut()
{
	int i,k,j,status,ccnt,rcnt,nzcnt,rmatbeg;
	double rhs;
	char sense;
	sense='G';
	rmatbeg=0;
	nzcnt=N;
	rcnt=1;
	ccnt=0;

	double sum=0;
	for (i=0; i<N; i++) {
		sum = sum + Zi[i];
	}
	rhs=1-sum;

	int *rmatind=NULL;
	rmatind = (int*) malloc((N) *sizeof(int));
	if(rmatind == NULL) {
		print_error("Cannot allocate variable pointers_demand");
```

```
		return false;
	}

	double *rmatval=NULL;

	rmatval = (double*) malloc((N) *sizeof(double));
	if(rmatval == NULL) {
		print_error("Cannot allocate coefficients_demand");
		return false;
	}

	for(i = 0; i < N; i++){
		if(Zi[i] == 0) {
			rmatind[i]=Z[i];
			rmatval[i]=1;
		}else{
			rmatind[i]=Z[i];
			rmatval[i]=-1;
		}
	}


	status = CPXaddrows (Env, Lp, ccnt, rcnt, nzcnt, &rhs,&sense, &rmatbeg,
	rmatind, rmatval,NULL, NULL);

	if (status != 0){
		print_error("CPXaddrows_demand");
		return false;
	}
	free(rmatind);
	free(rmatval);
	return true;
}
```

// MAIN

```
int main()
{
	int x,i,lpstat;
	I=-1;
	FILE *file;
 //import data from file

file=fopen("E:/Users/xristos/Desktop/metaptuxiako/diplomatiki/new/data5.txt","r";


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

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

Y=(double*)malloc((2*N)*sizeof(double));
M=(double*)malloc(N*sizeof(double));
K=(double*)malloc(N*sizeof(double));
Q=(double*)malloc(N*sizeof(double));
Z=(double*)malloc(N*sizeof(double));
P=(double*)malloc(N*sizeof(double));
Zi=(double*)malloc(N*sizeof(double));

for(i=0; i<N; i++) {
        fscanf(file, "%d",&x);          //εισαγωγή από txt των ελαχίστων
        M[i]=x;
        }

for(i=0; i<N; i++) {
        fscanf(file, "%d",&x);          //εισαγωγή από txt των μεγίστων
        K[i]=x;
        }

for(i=0; i<N; i++) {
        fscanf(file, "%d",&x);          //εισαγωγή από txt των bids
        P[i]=x;
}

fscanf(file, " %d",&x);                 //εισαγωγή από txt της ζήτησης
D=x;

fscanf(file, " %d",&x);                 //εισαγωγή από txt του ελαχίστου bid
c=x;

fscanf(file, " %d",&x);                 //εισαγωγή από txt του μεγίστου bid
PMAX=x;

fscanf(file, " %d",&x);                 //εισαγωγή από txt της ποσότητας μας
Qi=x;

fscanf(file, " %d",&x);                 //εισαγωγή από txt του SMP
SMP=x;

fscanf(file, " %d",&x);         //εισαγωγή από txt της ελάχιστης ποσότητας μας
M2=x;

fscanf(file, " %d",&x);         //εισαγωγή από txt της μέγιστης ποσότητας μας
K2=x;

if (init_env_lp() == false) exit;
if (Add_Vars_Q() == false) exit;
if (Add_Vars_Z() == false) exit;
if (Add_Var_P2() == false) exit;
```

```
if (Add_Vars_Y() == false) exit;
if (Add_Var_J() == false) exit;

if (demand() == false) exit;
if (Lower_Bound_Q() == false) exit;
if (Upper_Bound_Q() == false) exit;

I=-1;

for (i=0; i<N; i++) {
        if (P[i] == SMP){
                I=i;          //καθοριζουμε το marginal, αν I=-1 ειμαστε εμεις
        }
}

if (I!=-1) {
        if (J_lower() == false) exit;
        if (J_Upper() == false) exit;
}

if (Qi<K2 && Qi>M2){
        if (P2_EQU_SMP() == false) exit;

        if (MARG_US_Between() == false) exit;
        if (Add_coef_Y_EQU_Z() == false) exit;
}
else if (Qi == M2){
        if (I==-1){
                if (P2_EQU_SMP() == false) exit;
                if (MARG_US_Min() == false) exit;
                if (Add_coef_Y_EQU_Z() == false) exit;
        }else{
                if (Qmarg_Less_Qmax() == false) exit;
                if (P2_greater_Pmarg() == false) exit;
                if (Add_coef_MARG() == false) exit;
                if (Add_coef_Y_EQU_Z() == false) exit;
                if (Zmarg_EQU_1() == false) exit;
        }
}

else if (Qi == K2){
        if (I==-1){
                if (P2_EQU_SMP() == false) exit;
                if (MARG_US_MAX() == false) exit;
        }else{

                if (Add_coef_MARG_Qmax() == false) exit;
                if (Add_coef_Y_EQU_Z() == false) exit;
                if (Zmarg_EQU_1() == false) exit;
                if (P2_lower_Pmarg() == false) exit;
```

```
		}
	}

	if (PrintProblem() == false) exit;
	if (Solvprob() == false) exit;
	if (lp_stat() == false) exit;
	if (PrintSolStatus() == false) exit;


	// Εισαγωγή περιορισμών CUT

	lpstat = CPXgetstat (Env, Lp);

	while ( lpstat==101 ){
		if (cut() == false) exit;
		if (PrintProblem() == false) exit;
		if (Solvprob() == false) exit;
		if (lp_stat() == false) exit;
		if (PrintSolStatus() == false) exit;
	}

	free(M);
	free(K);
	free(Q);
	free(Z);
	free(P);

	printf("Press return to continue...\n");
	getchar();
	return 0;
}
```

Παράρτημα Β

Παρουσιάζεται ενδεικτικά ο τρόπος εφαρμογής και λειτουργίας του περιορισμού που λειτουργεί σαν cut και αναφέρθηκε στο Κεφάλαιο 3. Εφαρμόζεται για το Παράδειγμα 3 του Κεφαλαίου 4.

- Αρχικά ο αλγόριθμος που αναπτύχθηκε χωρίς την προσθήκη των νέων περιορισμών (cut) δίνει την πρώτη βέλτιστη λύση που εντοπίζει. Για το παράδειγμα 3 είναι η εξής:

  $\boldsymbol{Z_i}$ **= 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0**

  Ο περιορισμός που έχει εισαχθεί στον αλγόριθμο για να επιλυθεί ξανά είναι ο εξής:

  $$\boldsymbol{(1-Z_1) + Z_2 + Z_3 + (1-Z_4) + Z_5 + Z_6 + Z_7 + Z_8 + Z_9 + Z_{10} + Z_{11} + Z_{12} + (1-Z_{13}) + Z_{14} + Z_{15} + Z_{16} + (1-Z_{17}) + Z_{18} + Z_{19} + Z_{20} \geq 1}$$

- Μετά την νέα επίλυση εξάγεται η λύση:

  $\boldsymbol{Z_i}$ **= 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0**

  Ο περιορισμός που εισάγεται στον αλγόριθμο για να επιλυθεί ξανά είναι ο εξής:

  $$\boldsymbol{(1-Z_1) + Z_2 + Z_3 + Z_4 + Z_5 + Z_6 + Z_7 + Z_8 + Z_9 + Z_{10} + Z_{11} + Z_{12} + (1-Z_{13}) + Z_{14} + Z_{15} + (1-Z_{16}) + (1-Z_{17}) + Z_{18} + Z_{19} + Z_{20} \geq 1}$$

- Μετά την νέα επίλυση εξάγεται η λύση:

  $\boldsymbol{Z_i}$ **= 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0**

  Ο περιορισμός που εισάγεται στον αλγόριθμο για να επιλυθεί ξανά είναι ο εξής:

  $$\boldsymbol{(1-Z_1) + (1-Z_2) + Z_3 + Z_4 + Z_5 + Z_6 + Z_7 + Z_8 + Z_9 + Z_{10} + Z_{11} + Z_{12} + (1-Z_{13}) + Z_{14} + Z_{15} + Z_{16} + (1-Z_{17}) + Z_{18} + Z_{19} + Z_{20} \geq 1}$$

- Μετά την νέα επίλυση εξάγεται η λύση:

  $\boldsymbol{Z_i}$ **= 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0**

  Ο περιορισμός που εισάγεται στον αλγόριθμο για να επιλυθεί ξανά είναι ο εξής:

  $$\boldsymbol{(1-Z_1) + (1-Z_2) + Z_3 + Z_4 + Z_5 + Z_6 + Z_7 + Z_8 + Z_9 + Z_{10} + Z_{11} + Z_{12} + (1-Z_{13}) + Z_{14} + Z_{15} + (1-Z_{16}) + Z_{17} + Z_{18} + Z_{19} + Z_{20} \geq 1}$$

- Μετά την νέα επίλυση ο αλγόριθμος δίνει αποτέλεσμα, ανέφικτο. Συνεπώς το σύνολο των βέλτιστων λύσεων που εντόπισε ο αλγόριθμος είναι οι παραπάνω 4.

Βιβλιογραφία

[1] Dempe S. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. Optimization 2010; 52(3): 333–359.

[2] García-Martos C, Rodríguez, J and Sánchez MJ. Mixed models for short-run forecasting of electricity prices: application for the Spanish market. IEEE Trans. Power Syst. 2007; 22(2): 544-552.

[3] Ragupathi R, & Das T K. A stochastic game approach for modeling wholesale energy bidding in deregulated power markets. IEEE Trans. Power Syst. 2004; 19(2): 849-856.

[4] Weber JD, Overbye TJ. An individual welfare maximization algorithm for electricity markets. IEEE Trans. Power Syst. 2002; 17: 590–596.

[5] Gountis VP, Bakirtzis AG. Bidding strategies for electricity producers in a competitive electricity marketplace. IEEE Trans. Power Syst. 2004; 19: 356–365.

[6] Fampa M, Barroso LA, Candal D, Simonetti L. Bilevel optimization applied to strategic pricing in competitive electricity markets. Comput. Optim. Appl. 2008; 39: 121–142.

[7] Pereira MV, Granville S, Fampa MHC, Dix R, Barroso LA. Strategic bidding under uncertainty: a binary expansion approach. IEEE Trans. Power Syst. 2005; 20: 180–188.

[8] Barroso LA, Carneiro RD, Granville S, Pereira MV, Fampa MHC. Nash equilibrium in strategic bidding: a binary expansion approach. IEEE Trans. Power Syst. 2006; 21: 629–638.

[9] Bakirtzis AG, Ziogos NP, Tellidou AC, Bakirtzis GA. Electricity producer offering strategies in day-ahead energy market with step-wise offers. IEEE Trans. Power Syst. 2007; 22: 1804–1818.

[10] Ruiz C, Conejo AJ. Pool strategy of a producer with endogenous formation of locational marginal prices. IEEE Trans. Power Syst. 2009; 24: 1855–1866.

[11] Li T, Shahidehpour M, Keyhani A. Market power analysis in electricity markets using supply function equilibrium model. IMA J. Manage. Math. 2004; 15: 339–354.

[12] Hu X, Ralph D. Using EPECs to model bilevel games in restructured electricity markets with locational prices. Oper. Res. 2007; 55: 809–827.

[13] Hobbs BF, Metzler CB, Metzler J-S. Strategic gaming analysis for electric power systems: an MPEC approach. IEEE Trans. Power Syst. 2000; 15: 638–645.

[14] Li T, Shahidehpour M. Strategic bidding of transmission-constrained GENCOs with incomplete information. IEEE Trans. Power Syst. 2005; 20: 437–447.

[15] Ma Y, Jiang C, Hou Z, Wang C. The formulation of the optimal strategies for the electricity producers based on the particle swarm optimization algorithm. IEEE Trans. Power Syst. 2006; 21: 1663–1671.

[16] Zhang G, Zhang G, Gao Y, Lu J. A bilevel optimization model and a PSO-based algorithm in day-ahead electricity markets. In: IEEE International Conference on Systems, Man, and Cybernetics; 2009; San Antonio TX, USA. p. 611–616.

[17] Badri A, Jadid S, Rashidinejad M, Moghaddamc MP. Optimal bidding strategies in oligopoly markets considering bilateral contracts and transmission constraints. Electr. Power Syst. Res. 2008; 78: 1089–1098.

[18] Vahidinasab V, Jadid S. Multiobjective environmental/techno-economic approach for strategic bidding in energy markets. Appl. Energy 2009; 86: 496–504. 80

[19] Gabriel SA, Leuthold FU. Solving discretely-constrained MPEC problems with applications in electric power markets. Energy Econ. 2010; 32: 3–14.

[20] Bard JF. Practical bilevel optimization. Kluwer, Boston, MA, USA; 1998.

[21] Candler W, Norton R. Multi-level programming and development policy. World Bank, Bank Staff Working Paper No. 258, 1977.

[22] Dempe S. Foundations of bilevel programming. New York (NY): Kluwer Academic; 2002.

[23] Kozanidis G, Kostarelou E, Andrianesis P, Liberopoulos G. Mixed integer parametric bilevel programming for optimal strategic bidding of energy producers in day-ahead electricity markets with indivisibilities. *Optimization* 2013;62(8):1045-1068.

[24] Geoffrion AM, Nauss R. Parametric and postoptimality analysis in integer linear programming. Manage. Sci. 1977; 23: 453–466.

[25] Andrianesis P, Liberopoulos G, Kozanidis G, Papalexopoulos A. Recovery mechanisms in day-ahead electricity markets with non-convexities – part I: design and evaluation methodology. IEEE Trans. Power Syst. 2012; 28: 960–968.

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

[27] Zhang G, Zhang G, Gao Y, Lu J. Competitive strategic bidding optimization in electricity markets using bilevel programming and swarm technique. IEEE Trans. Ind. Electron. 2011; 58: 2138–2146.

[28] Foroud AA, Amirahmadi M, Bahmanzadeh M, Abdoos AA. Optimal bidding strategy for all market players in a wholesale power market considering demand response programs. Eur.Trans. Electr. Power 2011; 21: 293–311.

[29] Moore JT, Bard JF. The mixed integer linear bilevel programming problem. Oper. Res. 1990; 38: 911–921.

[30] Andrianesis P, Biskas P, Liberopoulos G. An overview of Greece's wholesale electricity market with emphasis on ancillary services. Electr. Power Syst. Res. 2011; 81: 1631–1642.

[31] Zhao F, Luh PB, Yan JH, Stern GA, Chang SC. Bid cost minimization versus payment cost minimization: a game theoretic study of electricity auctions. IEEE Trans. Power Syst. 2010; 25: 181–194.

[32] O'Neill RP, Sotkiewicz PM, Hobbs BF, Rothkopf MH, Stewart WR, Jr. Efficient marketclearing prices in markets with nonconvexities. Eur. J. Oper. Res. 2005; 164: 269–285.

[33] LINGO 13.0, User's guide, Chicago, IL: LINDO Systems, Inc; 2016. Available from: http:// www.lindo.com/.