ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ ΓΡΑΜΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΤΗΝ ΜΕΘΟΔΟ SIMPLEX

Διπλωματική Εργασία

του

ΔΗΜΗΤΡΙΟΥ ΤΥΧΑΛΑ

Επιβλέπων: Αναστάσιος Μωυσιάδης, Καθηγητής

Πανεπιστημιούπολη Σερρών, Ιανουάριος 2024

Ευχαριστίες

Στην διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας σημαντικός αριθμός ανθρώπων συνεργάστηκαν και προσέφεραν τη βοήθεια τους. Είμαι ευγνώμων σε όλους. Στο σημείο αυτό θεωρώ υποχρέωσή μου να απευθύνω τις πιο θερμές ευχαριστίες μου στους παρακάτω:

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της παρούσας εργασίας Δρ. Μωυσιάδη Αναστάσιο διότι χωρίς την συμβολή του δεν θα ήταν δυνατή η εκπόνηση αυτής της εργασίας. Συγκεκριμένα, θα ήθελα να τονίσω ότι ήταν για εμένα πολύ σημαντική η συνεχής βοήθεια και καθοδήγηση, αλλά και η στήριξη των επιλογών που μου πρόσφερε καθ' όλη την διάρκεια της εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω τον διπλωματούχο Μηχανικό Πληροφορικής και Τηλεπικοινωνιών, Δημήτριο Μανάφη, για σημαντική βοήθεια που μου πρόσφερε στο κομμάτι της δημιουργίας του γραφικού περιβάλλοντος χρήστη (GUI) της εφαρμογής.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και κυρίως την μητέρα μου, Χήτα Παναγιώτα, για την συνεχή υποστήριξη, αλλά και κατανόηση που μου πρόσφερε κατά την διάρκεια εκπόνησης της παρούσας εργασίας.

Περίληψη

Η παρούσα εργασία πραγματοποιήθηκε με σκοπό την δημιουργία ενός λογισμικού, το οποίο θα είναι φιλικό προς τον χρήστη και θα έχει τη δυνατότητα επίλυσης προβλημάτων Γραμμικού Προγραμματισμού με μια αντικειμενική συνάρτηση και πολλαπλούς περιορισμούς. Συγκεκριμένα, η μέθοδος που χρησιμοποιείται για την βελτιστοποίηση της αντικειμενικής συνάρτησης είναι η μέθοδος Simplex δύο φάσεων. Η δημιουργία του λογισμικού πραγματοποιήθηκε στο περιβάλλον ανάπτυξης Visual Studio 2019, το οποίο είναι εργαλείο της Microsoft Corporation, και η γλώσσα προγραμματισμού που επιλέχθηκε για την ανάπτυξή του είναι η Visual Basic. Το αποτέλεσμα της εργασίας είναι ένα εύχρηστο λογισμικό, το οποίο επιλύει προβλήματα Γραμμικού Προγραμματισμού με μεγάλη αποδοτικότητα και αποτελεσματικότητα δίνοντας ταυτόχρονα την δυνατότητα εισαγωγής μεγάλου όγκου δεδομένων μέσω υπολογιστικού φύλλου κάνοντάς το ένα χρήσιμο εργαλείο για ακαδημαϊκή χρήση, και ίσως με την βελτιστοποίηση του και την επέκταση των λειτουργιών του, και για επαγγελματική χρήση.

Abstract

This work was carried out in order to create a software, which will be user-friendly and will have the ability to solve Linear Programming problems with an objective function and multiple constraints. Specifically, the method used to optimize the objective function is the two-phase Simplex method. The creation of the software was carried out in the Visual Studio 2019 development environment, which is a tool of Microsoft Corporation, and the programming language chosen for its development is Visual Basic. The result of the work is an easy-to-use software, which solves Linear Programming problems with great efficiency and effectiveness while allowing the input of a large amount of data through a spreadsheet makes it a useful tool for academic use, and perhaps by optimizing it and extending its functions, and for professional use.

Πίνακας Περιεχομένων

1. Εισαγωγή

Στην παρούσα εργασία αναλύεται η διαδικασία ανάπτυξης ενός λογισμικού το οποίο εφαρμόζει την μέθοδο Simplex (2 φάσεων). Η Simplex είναι μια μέθοδος Γραμμικού Προγραμματισμού η οποία εφαρμόζεται σε προβλήματα βελτιστοποίησης, στα οποία υφίστανται περιορισμοί (ή οριακές συνθήκες), όπου τόσο η αντικειμενική συνάρτηση όσο και οι περιορισμοί είναι γραμμικές συναρτήσεις των μεταβλητών. Πολλά από τα προβλήματα που αντιμετωπίζει ένας μηχανικός στην δουλειά του έχουν ακριβώς αυτή την μορφή των προβλημάτων, με αποτέλεσμα η μέθοδος Simplex να αποτελεί ένα βασικό εργαλείο για κάθε μηχανικό, ο οποίος πρέπει να λαμβάνει αποφάσεις σε μια παραγωγική ή κατασκευαστική – σχεδιαστική διαδικασία με σκοπό να πετύχει το βέλτιστο αποτέλεσμα. Αυτό αποτέλεσε και το βασικό κίνητρο για την ανάπτυξη και εξέλιξη ενός αλγορίθμου, ο οποίος βασίζεται σε αυτή την μέθοδο.

1.1 Σκοπός της εργασίας

Ο κύριος σκοπός της εργασίας είναι η κατασκευή ενός λογισμικού το οποίο θα είναι απολύτως φιλικό προς το χρήστη με στόχο την διευκόλυνση αλλά και την επιτάχυνση της αλληλεπίδρασής του με το λογισμικό, το οποίο θα παρέχει την δυνατότητα επίλυσης, και μάλιστα με μεγάλη αποδοτικότητα, προβλημάτων Γραμμικού Προγραμματισμού πολύ μεγάλου μεγέθους (αναφορικά με τον αριθμό των μεταβλητών και το πλήθος των υφιστάμενων οριακών συνθηκών). Βασικός στόχος είναι μετά το πέρας της ολοκλήρωσης αλλά και βελτιστοποίησής του λογισμικού, να αποτελέσει ένα εργαλείο τόσο για τους φοιτητές του Μεταπτυχιακού Προγράμματος και τους υποψήφιους διδάκτορες του Τμήματος όσο και για τους επαγγελματίες μηχανικούς.

1.2 Δομή της εργασίας

Στο **Κεφάλαιο 0** περιγράφονται ο σκοπός της παρούσας εργασίας αλλά και η δομή του κάθε κεφαλαίου που την απαρτίζει.

Στο **Κεφάλαιο 2** αναλύεται το αναγκαίο, για την κατανόηση όλων των διαδικασιών και εννοιών, θεωρητικό υπόβαθρο.

Στο **Κεφάλαιο 3** περιγράφεται το υπολογιστικό κομμάτι εφαρμογής, δηλαδή το τμήμα του προγράμματος που εφαρμόζεται ο αλγόριθμός Simplex.

Στο **Κεφάλαιο 4** περιγράφεται το τμήμα του προγράμματος που αφορά το γραφικό περιβάλλον που έρχεται σε διεπαφή ο χρήστης.

Στο **Κεφάλαιο 5** αναλύονται τα συμπεράσματα που προέκυψαν από την διαδικασία κατασκευής του λογισμικού, καθώς και οι προτάσεις για το μέλλον.
Στο **Κεφάλαιο 6** αναφέρεται το σύνολο της βιβλιογραφίας που χρησιμοποιήθηκε για την ολοκλήρωση της εργασίας.

1.3 Βιβλιογραφική επισκόπηση

Για την δημιουργία του λογισμικού και για την επιλογή σωστής προσέγγισης και αντιμετώπισης των προβλημάτων που προέκυπταν, τόσο στην υπολογιστική διαδικασία όσο και στην δημιουργία του γραφικού περιβάλλοντος της εφαρμογής, ήταν αναγκαία η αναζήτηση και η μελέτη βιβλιογραφικών αλλά και ηλεκτρονικών πηγών. Παρακάτω αναλύονται στο σύνολό τους οι πηγές που χρησιμοποιήθηκαν.

Το βιβλίο Επιχειρησιακή Έρευνα του Βασίλη Κώστογλου (1) και το βιβλίο Αριθμητικές Μέθοδοι για Μηχανικούς των Steven C. Chapra και Raymond P. Canale (2) χρησιμοποιήθηκαν για την κατανόηση του θεωρητικού υπόβαθρου του Γραμμικού Προγραμματισμού και της μεθόδου Simplex, καθώς και την δημιουργία του υπολογιστικού αλγορίθμου που χρησιμοποιείται στο παρόν λογισμικό, ο οποίος βασίζεται εξ ολοκλήρου στις θεωρητικές γνώσεις που περιέχονται σε αυτά.

Οι υπόλοιπές πηγές (3), (4), (5), (6), (7), (8), (9) και (10) είναι όλες ηλεκτρονικές και συγκεκριμένα είναι ιστοσελίδες της εταιρίας Microsoft Corporation, στην οποία ανήκει και το Visual Studio στο οποίο και αναπτύχθηκε το λογισμικό. Από αυτές τις πηγές χρησιμοποιήθηκαν γενικές πληροφορίες για την κατασκευή ενός γραφικού περιβάλλοντος στο Visual Studio αλλά και πληροφορίες για την επίτευξη των λειτουργιών που πραγματοποιούνται πάνω σε αυτό, όπως είναι η δημιουργία των πλαισίων εισαγωγής κειμένου και εμφάνισης μηνυμάτων, η εισαγωγή και ανάγνωση υπολογιστικού φύλλου, αλλά και η κατασκευή λιστών πολλαπλής επιλογής.

2. Θεωρητικό υπόβαθρο

2.1 Γραμμικός Προγραμματισμός

2.1.1 Ορισμός Γραμμικού Προγραμματισμού

Ο Γραμμικός Προγραμματισμός είναι ένα κεφάλαιο των μαθηματικών που ασχολείται με προβλήματα βελτιστοποίησης, δηλαδή με την εύρεση της βέλτιστης λύσης ενός μαθηματικού μοντέλου, υπό συγκεκριμένους περιορισμούς. Στα προβλήματα αυτά τόσο η αντικειμενική συνάρτηση όσο και οι περιορισμοί είναι γραμμικές συναρτήσεις των μεταβλητών απόφασης. Ο στόχος είναι να βρεθούν οι τιμές αυτών των μεταβλητών που ελαχιστοποιούν ή μεγιστοποιούν την αντικειμενική συνάρτηση, με την προϋπόθεση ότι οι τιμές αυτές πληρούν τους περιορισμούς. Ο Γραμμικός Προγραμματισμός έχει πολλές εφαρμογές σε διάφορους τομείς, όπως οικονομική ανάλυση, διαχείριση αλυσίδων εφοδιασμού, παραγωγή, διαχείριση πόρων και πολλά άλλα.

2.1.2 Μοντέλο Γραμμικού Προγραμματισμού

Το μαθηματικό μοντέλο του Γραμμικού Προγραμματισμού (linear programming) περιλαμβάνει τον ορισμό της συνάρτησης που θέλουμε να μεγιστοποιήσουμε ή ελαχιστοποιήσουμε (αντικειμενική συνάρτηση), συνδυασμένη με περιορισμούς ισότητας ή ανισότητας που την επηρεάζουν. Πιο συγκεκριμένα, ο ορισμός ενός προβλήματος γραμμικού προγραμματισμού στην γενική του μορφή είναι ο εξής:

Αντικειμενική συνάρτηση:

Μεγιστοποίηση: $c_1x_1 + c_2x_2 + ... + c_nx_n$

Ελαχιστοποίηση: $c_1x_1 + c_2x_2 + ... + c_nx_n$

Περιορισμοί δομής:

$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \leq b_1$

$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \leq b_2$

...

$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \leq b_m$

Περιορισμοί μη αρνητικότητας:

$x_1, x_2, ..., x_n \geq 0$

Όπου, $c_1, c_2, ..., c_n$, είναι οι συντελεστές της αντικειμενικής συνάρτησης και $x_1, x_2, ..., x_n$ είναι οι μεταβλητές απόφασης που αναζητούμε. Οι αριθμοί $\alpha_{11}, \alpha_{12}, ..., \alpha_{mn}$ είναι οι συντελεστές των περιορισμών και $b_1, b_2, ..., b_m$ είναι οι σταθερές τιμές (τα δεξιά μέλη) των περιορισμών.

2.2 Μέθοδος Simplex

2.2.1 Η λειτουργία της μεθόδου Simplex

Ένας τρόπος έτσι ώστε να βρούμε την βέλτιστη λύση σε ένα πρόβλημα γραμμικού προγραμματισμού είναι να δημιουργήσουμε όλες τις βασικές λύσεις και να διαλέξουμε εκείνη που είναι αποδεκτή και αντιστοιχεί στην βέλτιστή τιμή της αντικειμενικής συνάρτησης. Αυτό μπορεί να γίνει επειδή η βέλτιστή λύση, εάν υπάρχει, πάντοτε βρίσκεται σε ένα ακραίο σημείο ή στην κορυφή μιας γωνίας στο πεδίο των αποδεκτών λύσεων.

Εάν υπάρχουν *m* εξισώσεις περιορισμών ισότητας και οι μεταβλητές με $n \geq m$, μια βασική λύση μπορεί να εξαχθεί θέτοντας κάποιες $n - m$ μεταβλητές ίσες με μηδέν. Ο αριθμός των βασικών λύσεων που πρέπει να ελεγχθούν είναι τότε ο αριθμός των δυνατών συνδυασμών των *n* μεταβλητών ανά *m* μεταβλητές (σε ομάδες μεταβλητών). Αυτός ο αριθμός δίνεται από την σχέση:

$$\binom{n}{m} = \frac{n!}{(n-m)!\,m!}$$

Έτσι για παράδειγμά εάν $n = 10$ και $m = 5$ η σχέση δίνει 252 συνδυασμούς άρα 252 βασικές λύσεις που πρέπει να ελεγχθούν ενώ εάν $n = 20$ και $m = 10$ τότε οι βασικές λύσεις γίνονται 184.756.

Για μεγάλους αριθμούς μεταβλητών *n* και εξισώσεων *m*, ο αριθμός των βασικών λύσεων που θα πρέπει να ελεγχθούν μεγαλώνει συνεπώς πάρα πολύ και γίνεται αναγκαία η ύπαρξη ενός υπολογιστικού σχήματος, το οποίο θα εξετάζει σε αλληλουχία τις βασικές αποδεκτές λύσεις αλλά κάθε φορά μόνο εκείνες που οδηγούν σε μικρότερη τιμή της αντικειμενικής συνάρτησης μέχρι να φτάσουμε στην ελάχιστη τιμή της. Αυτή την δυνατότητα προσφέρει η μέθοδος Simplex που πρότεινε ο Dantzig.

Η μέθοδος οδηγεί αρχικά στην απόκτηση μιας βασικής αποδεκτής λύσης. Εάν αυτή δεν είναι η βέλτιστη, τότε, δίνει την δυνατότητα μετακίνησης προς μια άλλη βασική αποδεκτή λύση, η οποία όμως θα δίνει μικρότερη ή ίδια τιμή στην αντικειμενική συνάρτηση. Η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί η ελάχιστη τιμή της αντικειμενικής συνάρτησης μετά από ένα περιορισμένο αριθμό επαναλήψεων.

Το πρώτο βήμα της μεθόδου είναι να κατασκευασθεί ένα βοηθητικό πρόβλημα, με την εισαγωγή τεχνητών μεταβλητών στο αρχικό σύστημα. Η προσθήκη αυτή γίνεται με σκοπό την μετατροπή του αρχικού προβλήματος στο βοηθητικό, το οποίο όμως θα είναι σε κανονική μορφή, τέτοια ώστε η βασική αποδεκτή λύση του να μπορεί να εξαχθεί αμέσως.

Ξεκινώντας από αυτή την κανονική μορφή, η βέλτιστη λύση του αρχικού προβλήματος μπορεί να βρεθεί σε δύο φάσεις. Στην πρώτη φάση γίνεται προσπάθεια να βρεθεί μια βασική αποδεκτή λύση του αρχικού προβλήματος. Αυτό γίνεται με μια αλληλουχία pivot operations με τις οποίες μετατρέπεται κάθε φορά το σύστημα σε κανονική μορφή, από τις οποίες μπορεί να βρεθεί η βέλτιστη λύση του βοηθητικού προβλήματος. Αυτό μας δίνει την δυνατότητα να βρούμε μια αποδεκτή λύση, εάν υπάρχει, του αρχικού προβλήματος.

Η δεύτερη φάση αφορά την εύρεση της βέλτιστης λύσης του αρχικού προβλήματος. Αυτό γίνεται με μια δεύτερη αλληλουχία pivot operations, η οποία επιτρέπει την μετακίνηση από μια βασική αποδεκτή λύση στην επόμενη του αρχικού προβλήματος, ώσπου να φθάσουμε στην βέλτιστη λύση, εάν υπάρχει.

Η μετατροπή του συστήματος σε κανονική μορφή, η οποία γίνεται και στις δύο φάσεις, γίνεται με το αλγόριθμο Simplex, ο οποίος αποτελεί την βάση της μεθόδου.

2.2.1.1 Ο αλγόριθμος Simplex

Ο αλγόριθμός Simplex ξεκινά πάντα από ένα σύστημα εξισώσεων, το οποίο περιλαμβάνει την αντικειμενική συνάρτηση μαζί με τις εξισώσεις των περιορισμών ισότητας του προβλήματος, σε κανονική μορφή.

Έτσι ο σκοπός του αλγορίθμου Simplex είναι να βρεθεί το διάνυσμα $x \geq 0$ το οποίο ελαχιστοποιεί την $f(x)$ και ικανοποιεί τις εξισώσεις:

$$
\begin{aligned}
&1x_1 + 0x_2 + \cdots + 0x_m + a''_{1,m+1}\, x_{m+1} + \cdots + a''_{1n}\, x_n = b''_1 \\
&0x_1 + 1x_2 + \cdots + 0x_m + a''_{2,m+1}\, x_{m+1} + \cdots + a''_{2n}\, x_n = b''_2 \\
&\qquad\qquad \cdots. \qquad\qquad (1) \\
&0x_1 + 0x_2 + \cdots + 1x_m + a''_{m,m+1}\, x_{m+1} + \cdots + a''_{mn}\, x_n = b''_m \\
&0x_1 + 0x_2 + \cdots + 0x_m - f + c''_{m+1}\, x_{m+1} + \cdots + c''_{mn}\, x_n = -f''_0
\end{aligned}
$$

Όπου a''_{ij} , c''_j , b''_i και f''_0 είναι σταθερές.

Να σημειωθεί ότι η $(-f)$ θεωρείται ως μια βασική μεταβλητή στην κανονική μορφή των εξισώσεων (1). Η βασική λύση που αμέσως εξάγεται από τις εξισώσεις (1) είναι:

$$
\begin{aligned}
&x_i = b_i\,, \quad i = 1,2, \ldots m \\
&f = f''_0 \qquad\qquad (2) \\
&x_i = 0\,, \quad i = m+1, m+2, \ldots n
\end{aligned}
$$

Εάν η βασική λύση είναι και αποδεκτή, οι τιμές των x_i , i = 1,2, … n είναι μη αρνητικές και έτσι ισχύει:

.

$$b''_{i.} \geq 0\,, \quad I = 1, 2, \ldots m$$

Στην πρώτη φάση της μεθόδου (φάση I), η βασική λύση που προκύπτει από την κανονική μορφή του συστήματος, όπως δημιουργείται με την προσθήκη των τεχνητών μεταβλητών, θα είναι αποδεκτή για το βοηθητικό πρόβλημα.

Η δεύτερη φάση της μεθόδου (φάση II) ξεκινά με μια βασική αποδεκτή λύση του αρχικού προβλήματος. Έτσι η αρχική κανονική μορφή στην αρχή του αλγορίθμου Simplex, θα είναι πάντα μια βασική αποδεκτή λύση.

Η βέλτιστη λύση θα είναι μια από τις βασικές αποδεκτές λύσεις. Συνεπώς εφόσον ο αλγόριθμος Simplex μετακινείται κάθε φορά από μια βασική σε μια άλλη βασική λύση με pivot operations, θα

πρέπει κάθε φορά να ελέγχεται, εάν η βασική αποδεκτή λύση στην οποία βρισκόμαστε είναι ή όχι η βέλτιστη.

Η ακόλουθη διαδικασία μας παρέχει την δυνατότητα να αποφανθούμε εάν βρισκόμαστε στην βέλτιστη λύση ή όχι, ελέγχοντας τους συντελεστές c''_j , $j = 1, 2, \dots n$ στην εξίσωση της αντικειμενικής συνάρτησης.

2.2.1.1.1 Η εξακρίβωση ενός σημείο βέλτιστής λύσης

Θεώρημα:

Μια βασική αποδεκτή λύση είναι μια βέλτιστη λύση με ελάχιστη τιμή της αντικειμενικής συνάρτησης ίση με f''_0 εάν όλοι οι συντελεστές c''_j , $j = m+1, m+2, \dots n$ στις εξισώσεις (1) είναι μη αρνητικοί.

Απόδειξη:

Από την τελευταία σειρά των εξισώσεων (1) μπορούμε να γράψουμε το εξής:

$$f''_0 + \sum_{i=m+1}^{n} c''_i x_i = f$$

Εφόσον οι μεταβλητές $x_{m+1}, x_{m+2}, \dots, x_n$ είναι στις εξισώσεις (1) μηδέν και περιορίζονται να μην είναι αρνητικοί αριθμοί, ο μόνος τρόπος να μεταβληθούν είναι να γίνουν θετικοί αριθμοί. Αλλά εάν $c''_i > 0$ για $i = m+1, m+2, \dots, n$, τότε αυξάνοντας οποιαδήποτε μεταβλητή x_i δεν μπορούμε να πετύχουμε μείωση της τιμής της αντικειμενικής συνάρτηση f. Από την στιγμή που καμία μεταβολή στις μη βασικές μεταβλητές δεν μπορεί να οδηγήσει σε μείωση της f , η παρούσα λύση πρέπει να είναι η βέλτιστη, με βέλτιστη τιμή της f την f''_0.

Βλέποντας επίσης τις τιμές των συντελεστών c''_i , μπορούμε να διαπιστώσουμε εάν υπάρχουν πολλαπλά βέλτιστα σημεία. Έστω όλα τα $c''_i > 0$, $i = m+1, m+2, \dots , k-1, k+1, \dots , n$ και $c''_k = 0$ για κάποια μη βασική μεταβλητή x_k. Εάν οι περιορισμοί επιτρέπουν στην μεταβλητή να γίνει θετικός αριθμός, από την παρούσα τιμή της που είναι μηδέν, τότε δεν μπορεί να επέλθει καμία αλλαγή στην τιμή της f και έτσι θα υπάρχουν περισσότερα βέλτιστα σημεία.

Ωστόσο είναι πιθανό η μεταβλητή να μην επιτρέπεται από τους περιορισμούς να γίνει θετικός αριθμός. Έτσι μπορούμε να πούμε ότι μια βασική αποδεκτή λύση είναι η μόνη βέλτιστη αποδεκτή λύση, εάν $c''_j > 0$ για όλες τις μη βασικές μεταβλητές $x''_j > 0$, $j = m+1, m+2, \dots, n$. Εάν η παρούσα βασική αποδεκτή λύση διαπιστωθεί ότι δεν είναι βέλτιστη, τότε μπορούμε ακολουθώντας την παρακάτω διαδικασία να πάμε σε μια βελτιωμένη λύση της παρούσας κανονικής μορφής.

2.2.1.1.2 Βελτιώνοντας μια μη βέλτιστη, βασική αποδεκτή λύση

Από την τελευταία σειρά των εξισώσεων (1) μπορούμε να γράψουμε την αντικειμενική συνάρτηση ως:

$$f = f''_0 + \sum_{i=1}^{m} c''_i x_i + \sum_{j=m+1}^{n} c''_j x_j$$
$$= f''_0 \text{ (για την λύση των εξισώσεων (2))}$$

Εάν τουλάχιστον ένας συντελεστής c''_j είναι αρνητικός, τότε η τιμή της f μπορεί να μειωθεί εάν αυξήσουμε την αντίστοιχη μεταβλητή x_j από μηδέν σε θετικό αριθμό. Δηλαδή η μη βασική μεταβλητή x_j , της οποίας ο συντελεστής c''_j είναι αρνητικός αριθμός, πρέπει να γίνει βασική μεταβλητή προκειμένου να μειωθεί περαιτέρω η αντικειμενική συνάρτηση. Την ίδια στιγμή, εξαιτίας της διαδικασίας εξάλειψης (pivot operation), θα πρέπει μια βασική μεταβλητή να γίνει μη βασική για να μπει η νέα στην θέση της. Έτσι, θα πρέπει να επιλεγεί εκείνη που θα γίνει μη βασική ώστε το αποτέλεσμα να είναι μια τιμή της f μικρότερη από την f''_0.
Εάν παραπάνω από ένας συντελεστής c''_j είναι αρνητικός, τότε επιλέγεται να γίνει η μεταβλητή x_s βασική, για την οποία ισχύει ότι:

$$c''_s = minimum\ c''_j < 0$$

Δηλαδή αυτή με τον μικρότερο αρνητικό συντελεστή. Αυτή η επιλογή αν και μπορεί να μην οδηγήσει αμέσως στην βέλτιστη λύση, ωστόσο βοηθάει στο να απαιτηθούν λιγότερες επαναλήψεις.

Εάν τώρα περισσότεροι συντελεστές c''_j έχουν την ίδια ελάχιστη αρνητική τιμή, τότε διαλέγουμε αυθαίρετα τον c''_s (δηλαδή την x_s που θα γίνει βασική). Αφού αποφασίσαμε ποια μεταβλητή x_s θα γίνει βασική, την αυξάνουμε από μηδέν σε έναν θετικό αριθμό, διατηρώντας όλες τις άλλες μη

βασικές μεταβλητές στην τιμή μηδέν. Παρατηρούμε το αποτέλεσμα αυτής της αύξησης σε όλες τις βασικές μεταβλητές.

Από τις εξισώσεις (1) έχουμε:

$$\begin{aligned} x_1 &= b''_1 - a''_{1s}\, x_s\,, \quad b''_1 \geq 0 \\ x_2 &= b''_2 - a''_{2s}\, x_s\,, \quad b''_2 \geq 0 \\ &\cdots. \\ x_m &= b''_m - a''_{ms}\, x_s\,, \quad b''_m \geq 0 \end{aligned} \qquad (3)$$

$$f = f''_0 + c''_s\, x_s\,, \quad c''_s < 0 \qquad (4)$$

Εφόσον είναι $c''_s < 0$ η εξίσωση (4) υποδεικνύει ότι η μεταβλητή x_s από μηδέν πρέπει να αυξηθεί όσο γίνεται περισσότερο ώστε αφαιρούμενη να μειώσει όσο γίνεται περισσότερο την τιμή της f.

Ωστόσο αυξάνοντας την x_s θα πρέπει να προσέξουμε γιατί με βάση τις εξισώσεις (3) κάποιες από τις βασικές μεταβλητές x''_i , $i = 1,2, \ldots, m$ μπορεί να γίνουν αρνητικές. Μπορούμε να διαπιστώσουμε ότι αν όλοι οι συντελεστές $a''_{is} \leq 0$, $i = 1,2, \ldots, m$ τότε η x_s μπορεί να αυξηθεί απεριόριστα χωρίς να γίνεται καμία μεταβλητή x''_i , $i = 1,2, \ldots, m$ αρνητική. Στην περίπτωση αυτή η ελάχιστη τιμή της f είναι το μείον άπειρο και το πρόβλημα λέμε ότι έχει απεριόριστη λύση. Αν τουλάχιστον ένας συντελεστής a''_{is} είναι θετικός, τότε η μέγιστη τιμή που μπορεί να πάρει η x_s δίνεται από την ελάχιστη τιμή του λόγου b''_i / a''_{is} για τον οποίο ισχύει $a''_{is} > 0$.

Δηλαδή:

$$x^*_s = \frac{b''_r}{a''_{rs}} = \text{minimum}\left(\frac{b''_i}{a''_{is}}\right) \quad \text{όπου } a''_{is} > 0 \qquad (5)$$

Η επιλογή του r εάν ισχύει το $\min (b''_i / a''_{is})$ για περισσότερους a''_{rs} , με την προϋπόθεση ότι όλα τα $b''_i > 0$, είναι αυθαίρετη.

Εάν κάποιο b''_i για το οποίο ισχύει ότι $a''_{is} > 0$ είναι μηδέν, τότε με βάση τις εξισώσεις (3) η x_s δεν μπορεί να αυξηθεί καθόλου. Στην περίπτωση αυτή η λύση ονομάζεται εκφυλισμένη (degenerated solution).

Στην περίπτωση που δεν ισχύει $b''_i = 0$, τότε από την μια βασική αποδεκτή λύση μπορούμε να πάμε σε μια νέα βασική αποδεκτή λύση ως ακολούθως:

Αντικαθιστώντας την τιμή της x^*_s από την εξίσωση (5) στις εξισώσεις (3) και (4) έχουμε:

$$\begin{aligned} x_s &= x^*_s \\ x_i &= b''_i - a''_{is} x^*_s, \quad i = 1, 2, \dots, m \ \text{και} \ i \neq r \\ x_r &= 0 \\ x_j &= 0 \quad j = m+1, m+2, \dots, n \ \text{και} \ j \neq r \end{aligned} \tag{6}$$

$$f = f''_0 + c''_s x^*_s \leq f''_0 \tag{7}$$

Εφόσον $a''_{rs} > 0$ στην εξίσωση (5) μια pivot operation στον συντελεστή a''_{rs} στο σύστημα των εξισώσεων (1) θα οδηγήσει σε μια νέα κανονική μορφή, από την οποία προκύπτει η βασική αποδεκτή λύση των εξισώσεων (6).

Επίσης, η εξίσωση (7) δείχνει ότι αυτή η βασική αποδεκτή λύση οδηγεί σε μικρότερη τιμή της αντικειμενικής συνάρτησης f σε σύγκριση με αυτή των εξισώσεων (2). Αυτή η λύση μπορεί να ελεγχθεί και πάλι εάν είναι η βέλτιστη βλέποντας εάν όλοι οι συντελεστές $c''_j > 0$ στην νέα κανονική μορφή. Εάν δεν είναι η βέλτιστη, τότε όλη η διαδικασία θα πρέπει να επαναληφθεί.

Στον αλγόριθμό Simplex αυτό γίνεται με έναν επαναλαμβανόμενο τρόπο μέχρι ο αλγόριθμος να βρει:

α) μια ομάδα αποδεκτών λύσεων για τις οποίες $f \to -\infty$ ή

β) μια βασική αποδεκτή λύση βέλτιστη με όλους τους $c''_j \geq 0$, $i = 1, 2, \dots, n$.

2.2.2 Περίγραμμα της μεθόδου Simplex

Για την επίλυση ενός προβλήματος Γραμμικού Προγραμματισμού και συγκεκριμένα για την βελτιστοποίηση της αντικειμενικής συνάρτησης ενός προτύπου μέσα στα πλαίσια που ορίζουν οι περιορισμοί δομής, πρέπει να βρεθούν αρχικά όλες οι ακραίες δυνατές λύσεις και να επιλεγεί η καλύτερη από αυτές, εκείνη δηλαδή που δίνει στην συνάρτηση την καλύτερη δυνατή τιμή (μέγιστη

ή ελάχιστη ανάλογα με τον αντικειμενικό στόχο του προβλήματος). Παρόλο που η διαδικασία αυτή είναι θεωρητικά ορθή και εύχρηστη σε ικανοποιητικό βαθμό, η μέθοδος Simplex είναι πολύ πιο αποτελεσματική. Το περίγραμμά της μπορεί να συνοψισθεί στην ακόλουθή επαναληπτική διαδικασία:

- **Αρχικό βήμα:** Εκκίνηση από μια ακραία δυνατή λύση.
- **Επαναληπτικό βήμα:** Μετακίνηση σε μία καλύτερη γειτονική ακραία λύση (το βήμα αυτό επαναλαμβάνεται όσες φορές χρειαστεί).
- **Κανόνας τερματισμού:** Τέλος της διαδικασίας όταν η τρέχουσα ακραία δυνατή λύση είναι καλύτερη από όλες τις γειτονικές της ακραίες δυνατές λύσεις. Τότε αυτή είναι η άριστη λύση.

Είναι γνωστό από την θεωρία του Γραμμικού Προγραμματισμού ότι το πλήθος των ακραίων γειτονικών λύσεων είναι πεπερασμένο. Ως εκ τούτου η μέθοδος Simplex θα πρέπει να βρει την άριστη λύση μετά από πεπερασμένο αριθμό επαναληπτικών βημάτων. Το πραγματικό όμως κλειδί της αποτελεσματικότητας αυτής της μεθόδου είναι ότι απαιτεί έναν πολύ μικρό αριθμό επαναλήψεων. Η λεπτομερής παρουσίαση της μαθηματικής απόδειξης για το σημαντικό αυτό γεγονός δεν είναι απαραίτητη, αρκούν τα συνεπή αποτελέσματα της επίλυσης εκατοντάδων χιλιάδων πραγματικών προβλημάτων εδώ και πέντε δεκαετίες.

2.2.3 Οργάνωση του μαθηματικού προτύπου

Ένα σημαντικό πλεονέκτημα της μεθόδου Simplex, πέρα από τον μικρό αριθμό επαναληπτικών βημάτων που απαιτούνται, είναι η δυνατότητα πλήρους τυποποίησής της σε τέτοιο βαθμό, ώστε να καθίσταται εύκολη τόσο η απομνημόνευσή της για την εκτέλεση με χαρτί και μολύβι, όσο και η σύνταξη ενός προγράμματος ηλεκτρονικού υπολογιστή, ο οποίος αποτελεί πλέον το συνηθέστερο μέσο επίλυσης προβλημάτων Γραμμικού Προγραμματισμού.

Πριν, ωστόσο, την εκτέλεση της κυρίως μεθοδολογίας της Simplex είναι απαραίτητη η διεξαγωγή ορισμένων προκαταρκτικών ενεργειών, οι οποίες ουσιαστικά μετατρέπουν το πρόβλημα σε μια εύχρηστη καθαρά υπολογιστή διαδικασία. Η προπαρασκευαστική αυτή οργάνωση του μαθηματικού πρότυπου του Γραμμικού Προγραμματισμού μπορεί να συνοψισθεί στα ακόλουθα στάδια:

1. Μετατροπή της αντικειμενικής συνάρτησης και των περιορισμών δομής και μη αρνητικότητας στη μορφή που υπαγορεύει το μοντέλο του Γραμμικού Προγραμματισμού (βλέπε ενότητα 2.1.2) .
2. Αλλαγή της φοράς της ανισότητας όσων περιορισμών δομής απαιτείται, έτσι ώστε τα δεξιά μέλη όλων των περιορισμών να είναι μη αρνητικοί αριθμοί. Επομένως, αν το δεξιό μέλος ενός περιορισμού είναι αρνητικό χρειάζεται η αλλαγή των πρόσημων και της φοράς της αντίστοιχης ανισότητας (πολλαπλασιάζοντας και τα δύο μέλη με -1).
3. Μετατροπή των περιορισμών δομής που αποτελούν ανισότητες ή ανισοϊσότητες σε ισοδύναμες ισότητες. Αυτό επιτυγχάνεται με την προσθήκη (πρόσθεση αν η φορά της ανισότητας είναι < ή ≤ και αφαίρεση αν η φορά είναι της μορφής > ή ≥) νέων μη αρνητικών μεταβλητών, οι οποίες λέγονται ψευδομεταβλητές. Για παράδειγμα στο περιορισμό δομής $3x_1 – 5x_2 + 2x_3 \leq 13$ χρειάζεται η πρόσθεση της ψευδομεταβλητής x_4 προκειμένου αυτός να μετατραπεί στην ισότητα $3x_1 – 5x_2 + 2x_3 + x_4 = 13$. Με τον ίδιο τρόπο ο περιορισμός $4x_1 + x_2 - 7x_3 \geq 8$ αφαιρώντας την ψευδομεταβλητή x_4 μετατρέπεται στην ισότητα $4x_1 + x_2 - 7x_3 - x_4 = 8.$ (Εξυπακούεται ότι και στις δύο παραπάνω περιπτώσεις η μεταβλητή x_4 δεν περιλαμβάνεται στις αρχικές μεταβλητές απόφασης του αντίστοιχου προτύπου).
4. Δημιουργία στο αριστερό τμήμα του πίνακα των περιορισμών δομής ενός μοναδιαίου πίνακα με την πρόσθεση κατάλληλου αριθμού νέων μη αρνητικών μεταβλητών, οι οποίες ονομάζονται τεχνητές μεταβλητές. Συγκεκριμένα, απαιτείται η δημιουργία στο αριστερό μέλος του συνόλου των περιορισμών δομής ενός ολοκληρωμένου μοναδιαίου πίνακα. Οι στήλες του επιτρέπεται να είναι διατεταγμένες σε οποιαδήποτε σειρά μεταξύ τους. Όπως είναι φανερό, υπάρχει πιθανότητα μία ή περισσότερες από τις στήλες του επιθυμητού μοναδιαίου πίνακα να προϋπάρχουν προερχόμενες είτε από τις αρχικές μεταβλητές απόφασης είτε, πολύ συχνότερα, από τις ψευδομεταβλητές που προστέθηκαν στους περιορισμούς στο αμέσως προηγούμενο στάδιο της οργάνωσης του μαθηματικού προτύπου. Επομένως, ο αριθμός των τεχνητών μεταβλητών που πρέπει να προστεθούν ποικίλλει ανάλογα με τη φορά και την φύση των περιορισμών.

2.2.4 Διαμόρφωσή του αρχικού πίνακα Simplex

Η διαδικασία που οδηγεί στη βέλτιστη λύση μπορεί να συστηματοποιηθεί πιο αποτελεσματικά αν παρασταθεί με την μορφή διαδοχικών πινάκων. Καθένας από αυτούς αντιστοιχεί σε ένα στάδιο της μεθοδολογίας και ονομάζεται πίνακας Simplex. Κάθε πίνακας περιέχει όλα τα στοιχεία του

προβλήματος που αντιπροσωπεύουν το αντίστοιχο στάδιο της διαδικασίας και είναι απαραίτητα για την μετάβαση στο επόμενο, πλησιέστερο στην άριστη λύση, στάδιο. Παρακάτω φαίνεται η αρχική μορφή ενός πίνακα Simplex, όπως και τα στοιχεία που περιλαμβάνονται σε αυτόν:

Βάση	x_1	x_2	...	x_n	x_{n+1}	x_{n+2}	...	x_{n+m}	Δ.Μ.
x_{n+1}	a_{11}	a_{12}	...	a_{1n}	1	0	...	0	b_1
x_{n+2}	a_{21}	a_{22}	...	a_{2n}	0	1	...	0	b_2
...	.....								...
x_{n+m}	a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1	b_m
-f	c_1	c_2	...	c_n	0	0	...	0	0

Όπου:

$x_1, x_2, ..., x_n$ οι μεταβλητές απόφασης (φυσικές),

$x_{n+1}, x_{n+2}, ..., x_{n+m}$ οι ψευδομεταβλητές και οι τεχνητές μεταβλητές,

$b_1, b_2, ..., b_m$ τα δεξιά μέλη των περιορισμών,

$c_1, c_2, ..., c_n$ οι συντελεστές όλων των μεταβλητών της αντικειμενικής συνάρτησης,

$a_{m1}, a_{m2}, ..., a_{mn}$ ο πίνακας τον συντελεστών των φυσικών μεταβλητών στους περιορισμούς,

1, 0, …, 1 ο μοναδιαίος πίνακας των συντελεστών των βασικών μεταβλητών στους περιορισμούς,

Δ.Μ. το δεξιό μέλος των περιορισμών.

2.2.5 Επαναληπτικός αλγόριθμός Simplex

Μετά την διαμόρφωση του μαθηματικού προτύπου του προβλήματος και την δημιουργία του αρχικού πίνακα Simplex ακολουθεί η κυρίως μεθοδολογία προσδιορισμού της βέλτιστης λύσης. Η μεθοδολογία αυτή αποτελείται από μια επαναληπτική διαδικασία που παριστάνεται από ένα μαθηματικό αλγόριθμο, τον αλγόριθμό Simplex. Παρακάτω αναλύονται τα βήματα της αλγοριθμικής διαδικασίας για ένα πρόβλημα ελαχιστοποίησης (Στην περίπτωση προβλήματος μεγιστοποίησης μετατρέπεται το πρόβλημα σε ελαχιστοποίησης πολλαπλασιάζοντας την αντικειμενική συνάρτηση με -1).

- **Επιλογή της μεταβλητής που θα εισέλθει στη βάση.**

  Στην βάση επιλέγεται να εισέλθει μια μη βασική μεταβλητή. Συγκεκριμένα, επιλέγεται αυτή με τον μικρότερο αρνητικό συντελεστή c_i, έτσι ώστε να εξασφαλισθεί ο μεγαλύτερος ρυθμός μείωσης της τιμής της αντικειμενικής συνάρτησης.

- **Προσδιορισμός της μεταβλητής που θα εγκαταλείψει την βάση.**

  Προκειμένου να προσδιορισθεί η μεταβλητή που πρόκειται να εγκαταλείψει τη βάση υπολογίζεται για κάθε μια από τις τρέχουσες βασικές μεταβλητές το θετικό πηλίκο b_i/a_{ij}, όπου b_i είναι το δεξιό μέλος που εκφράζει την τρέχουσα τιμή της αντίστοιχης βασικής μεταβλητής και a_{ij} ο συντελεστής της στη στήλη του πίνακα Simplex που αντιστοιχεί στη μεταβλητή που μόλις εισήλθε στη βάση. Από τα παραπάνω πηλίκα που βρέθηκαν θετικά επιλέγεται το μικρότερο, διότι αυτό επιβάλει την μικρότερη δυνατή μείωση της μεταβλητής που εισήλθε στην βάση.

- **Προσδιορισμός της νέας βασικής δυνατής λύσης.**

  Έχοντας προσδιορίσει την εισερχόμενη και την εξερχόμενη από την βάση μεταβλητή απομένει η εύρεση της νέας βελτιωμένης λύσης καθώς και ο προσδιορισμός όλων των στοιχείων του νέου πίνακα Simplex. Η διαδικασία αυτή εκτελείται με τα ακόλουθα βήματα:

1. <u>Προσδιορισμός του στοιχείο ‘‘οδηγός’’ (pivot element).</u>

   Αυτός είναι ο διαιρέτης a_{ij} του πηλίκου b_i/a_{ij}, το οποίο αποτελεί το κριτήριο προσδιορισμού της εξερχόμενης από την βάση μεταβλητής. Το στοιχείο ‘‘οδηγός’’ είναι δηλαδή ο συντελεστής του περιορισμού που βρίσκεται στην τομή μεταξύ της γραμμής της εξερχόμενης μεταβλητής και της στήλης της μεταβλητής που εισέρχεται στη νέα βάση.

2. <u>Σχεδιασμός νέου πίνακα Simplex κάτω από τον τρέχοντα.</u>

   Σύνταξη της νέας βάσης αντικαθιστώντας τη μεταβλητή που βγήκε από την προηγούμενη βάση με αυτήν που εισέρχεται στην νέα βάση.

3. Υπολογισμός της νέας γραμμής του οδηγού.

   Διαίρεση όλων των στοιχείων της προηγούμενης γραμμής του οδηγού με το στοιχείο ‘‘οδηγός’’, δηλαδή:

$$Στοιχείο\ νέας\ γραμμής\ οδηγού = \frac{Στοιχείο\ προηγούμενης\ γραμμής\ οδηγού}{Στοιχείο\ οδηγός}$$

4. Υπολογισμός των υπόλοιπων στοιχείων του νέου πίνακα Simplex.

   Από κάθε στοιχείο του προηγούμενου πίνακα αφαιρείται το γινόμενο του αντίστοιχου στοιχείου της προηγούμενης στήλης του οδηγού με το αντίστοιχο στοιχείο της νέας γραμμής του οδηγού.

$$Νέο\ Στοιχείο = Αντίστοιχο\ Προηγούμενο\ Στοιχείο -$$
$$(Αντιστοιχό\ Στοιχείο\ Προηγούμενης\ Στήλης) \times (Αντίστοιχο\ Στοιχείο\ Νέας\ Γραμμής)$$

- **Κανόνας τερματισμού.**

  Συνοψίζοντας, η διαδικασία βελτιστοποίησης της συνάρτησης που υιοθετεί η μέθοδος Simplex αποτελείται από: α) την επιλογή της μεταβλητής που θα εισέλθει στην βάση, β) από τον προσδιορισμό της μεταβλητής που θα εγκαταλείψει την βάση και γ) από την εύρεση της νέας βελτιωμένης λύσης. Η διαδικασία αυτή επαναλαμβάνεται με τον ίδιο ακριβώς τρόπο έως ότου οι συντελεστές όλων των μη βασικών μεταβλητών στην αντικειμενική συνάρτηση γίνουν θετικοί ή μηδέν. Στην περίπτωση αυτή δεν υπάρχει πλέον μη συμμετέχουσα στην λύση μεταβλητή, η οποία να μπορεί να συνεισφέρει στην μείωση της τιμής της αντικειμενικής συνάρτησης, επομένως η τρέχουσα λύση είναι η άριστη. Τέλος, στην διαδικασία αυτή δεν περιγράφονται ορισμένες περιπλοκές που είναι δυνατό να παρουσιασθούν οι οποίες δεν προβλέπονται από τους βασικούς κανόνες. Οι περιπτώσεις αυτές θα εξετασθούν αναλυτικά στην επόμενη ενότητα.

2.2.6 Περιπλοκές κατά την επαναληπτική διαδικασία

Στην προηγούμενη ενότητα αναλύθηκε η λειτουργία της επαναληπτικής διαδικασίας της μεθόδου Simplex, ωστόσο δεν έγινε κάποια αναφορά για την διαδικασία που πρέπει να ακολουθηθεί εάν υπάρχει κάποια περιπλοκή. Παρακάτω αναλύονται οι δυο βασικότερες περιπλοκές που μπορεί να προκύψουν αλλά και ποια είναι η ορθή αντιμετώπιση τους.

- **Μη πεπερασμένη τιμή της αντικειμενικής συνάντησης.**

  Η περιπλοκή αυτή παρουσιάζεται στο δεύτερο βήμα (απομάκρυνση μεταβλητής από την βάση) της επαναληπτικής διαδικασίας. Συγκεκριμένα συμβαίνει όταν καμία μεταβλητή δεν συγκεντρώνει τις προϋποθέσεις για να εγκαταλείψει την βάση. Στην πινακοποιημένη μορφή της μεθόδου αυτό σημαίνει ότι όλοι οι συντελεστές της στήλης της εισερχόμενης μεταβλητής είναι αρνητικοί ή μηδέν. Η ερμηνεία ενός τέτοιου ενδεχομένου είναι ότι οι περιορισμοί δομής δεν εμποδίζουν την επ' αόριστο αύξηση της τιμής της συνάρτησης προς την ευνοϊκή κατεύθυνση θετική ή αρνητική. Στην περίπτωση αυτή η μέθοδος Simplex σταματά, επειδή η τιμή της αντικειμενικής συνάρτησης είναι μη πεπερασμένη. Η πιθανότερη αιτία δημιουργίας μη πεπερασμένης τιμής είναι η λανθασμένη διαμόρφωση του προβλήματος.

- **Πολλαπλές άριστες λύσεις.**

  Η δεύτερη περιπλοκή είναι η επίλυση προβλήματος το οποίο καταλήγει να έχει πολλαπλές άριστες λύσεις. Αυτό συμβαίνει όταν η οικογένεια των ευθειών της αντικειμενικής συνάρτησης έχει τον ίδιο συντελεστή διεύθυνσης (είναι δηλαδή παράλληλη) με ένα από τους περιορισμούς δομής. Η παραπάνω περιπλοκή γίνεται αντιληπτή μόλις βρεθεί η πρώτη άριστη λύση, καθώς υπάρχει τουλάχιστον μία από τις μη βασικές μεταβλητές με μηδενικό συντελεστή στην γραμμή της αντικειμενικής συνάρτησης. Αυτό έχει σαν αποτέλεσμα να έχει βρεθεί η βέλτιστη τιμή της αντικειμενικής συνάρτησης, όμως παράλληλα αυτή η τιμή μένει σταθερή για όλες τις τιμές της μεταβλητής η οποία έχει μηδενικό συντελεστή. Στην περίπτωση αυτή έχουμε πολλαπλές (άπειρες) άριστες λύσεις.

2.2.7 Δύο φάσεις της μεθόδου Simplex

Το πρόβλημα όπως αναφέρθηκε είναι να βρεθούν μη αρνητικοί αριθμοί για τις μεταβλητές $x_1, x_2, \ldots, x_n$ οι οποίοι ικανοποιούν τις εξισώσεις (1) και ελαχιστοποιούν την αντικειμενική συνάρτηση (2).

$$\begin{aligned} a_{11}\,x_1 + a_{12}\,x_2 + \cdots + a_{1n}\,x_n &= b_1 \\ a_{21}\,x_1 + a_{22}\,x_2 + \cdots + a_{2n}\,x_n &= b_2 \\ \cdots & \\ a_{m1}\,x_1 + a_{m2}\,x_2 + \cdots + a_{mn}\,x_n &= b_m \end{aligned} \qquad (1)$$

$$c_1\,x_1 + c_2\,x_2 + \cdots + c_n\,x_n = f \qquad (2)$$

Τα προβλήματα που μπορούν να προκύψουν κατά την επίλυση του προβλήματος είναι:

1) Είναι δυνατόν να μην μπορεί να σχηματισθεί αμέσως η αρχική αποδεκτή κανονική μορφή του συστήματος. Αυτό μπορεί να συμβεί όταν το πρόβλημα γραμμικού προγραμματισμού δεν έχει ψευδομεταβλητές για κάποιες από τις εξισώσεις του ή όταν οι ψευδομεταβλητές έχουν αρνητικούς συντελεστές.
2) Το πρόβλημα μπορεί να έχει πλεονασματικούς ή ασυμπτωτικούς περιορισμούς και μπορεί να μην είναι επιλύσιμο με μη αρνητικούς αριθμούς.

Για την επίλυση αυτών των προβλημάτων μπορεί να χρησιμοποιηθεί η μέθοδος Simplex δύο φάσεων. Η φάση I της μεθόδου χρησιμοποιεί τον αλγόριθμο Simplex για να διαπιστωθεί εάν το πρόβλημα γραμμικού προγραμματισμού έχει μια αποδεκτή λύση. Εάν υφίσταται μια αποδεκτή λύση, αυτή δίνει μια βασική αποδεκτή λύση σε κανονική μορφή, έτοιμη για να εφαρμοσθεί η φάση II της μεθόδου. Η φάση II χρησιμοποιεί τον αλγόριθμο Simplex προκειμένου να βρεθεί εάν το πρόβλημα έχει ένα περιορισμένο βέλτιστο. Εάν υπάρχει ένα περιορισμένο βέλτιστο, βρίσκει την βασική αποδεκτή λύση, η οποία είναι βέλτιστη. Η μέθοδος περιγράφεται ως εξής:

1) Διαμορφώνουμε το αρχικό σύστημα των εξισώσεων (1) έτσι ώστε όλοι οι σταθεροί όροι b_i να είναι θετικοί ή μηδέν, αλλάζοντας το πρόσημο των εξισώσεων, στις οποίες είναι απαραίτητο, και στα δύο μέλη.
2) Εισάγουμε στο σύστημα ένα πλήθος τεχνητών μεταβλητών $y_1, y_2, \ldots, y_m$ *με* $y_i \geq 0$, οι οποίες λειτουργούν ως βασικές μεταβλητές στην φάση I, έτσι ώστε το σύστημα να μετατραπεί ως εξής:

$$\begin{aligned} a_{11}\,x_1 + a_{12}\,x_2 + \cdots + a_{1n}\,x_n + y_1 &= b_1 \\ a_{21}\,x_1 + a_{22}\,x_2 + \cdots + a_{2n}\,x_n + y_2 &= b_2 \\ \cdots & \\ a_{m1}\,x_1 + a_{m2}\,x_2 + \cdots + a_{mn}\,x_n + y_m &= b_m \\ b_i \geq 0 & \end{aligned} \qquad (3)$$

Στις εξισώσεις (3) υπάρχει το ενδεχόμενο κάποιος συντελεστής a_{ij} και ο αντίστοιχός b_i να είναι με αντίθετο πρόσημο από ότι ήταν στις εξισώσεις (1) λόγω του πρώτου βήματος.

Η αντικειμενική συνάρτηση (2) μπορεί να γραφεί ως:

$$c_1\,x_1 + c_2\,x_2 + \cdots + c_n\,x_n + (-f) = 0 \qquad (4)$$

3) Φάση Ι της μεθόδου. Ορίζουμε μια ποσότητα w ως το άθροισμα των τεχνητών μεταβλητών.

$$w = y_1 + y_2 + \ldots + y_m \tag{5}$$

Στην συνέχεια χρησιμοποιούμε τον αλγόριθμο Simplex για να βρούμε τις $x_i \geq 0$ (i = 1, 2, …, n) και $y_i \geq 0$ (i = 1, 2, …, m) οι οποίες ελαχιστοποιούν την w και ικανοποιούν τις εξισώσεις (3) και (4). Συνεπώς θεωρούμε το σύστημα:

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + y_1 = b_1\\
&a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + y_2 = b_2\\
&\cdots.\\
&a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + y_m = b_m\\
&c_1x_1 + c_2x_2 + \cdots + c_nx_n + (-f) = 0\\
&y_1 + y_2 + \ldots + y_m + (-w) = 0
\end{aligned} \tag{6}$$

Το σύστημα αυτό δεν είναι σε κανονική μορφή. Μπορεί όμως να μετατραπεί σε κανονική μορφή με βασικές μεταβλητές τις $y_1, y_2, \ldots, y_m, -f, -w$ αφαιρώντας το άθροισμα των m πρώτων εξισώσεων από την τελευταία εξίσωση, οπότε γίνεται:

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + y_1 = b_1\\
&a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + y_2 = b_2\\
&\cdots.\\
&a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + y_m = b_m\\
&c_1x_1 + c_2x_2 + \cdots + c_nx_n + (-f) = 0\\
&d_1x_1 + d_2x_2 + \cdots + d_nx_n + (-w) = -w_0
\end{aligned} \tag{7}$$

όπου $d_i = -(a_{1i} + a_{2i} + \ldots + a_{mi})$, i = 1, 2, …, n (8)

$-w_0 = -(b_1 + b_2 + \ldots + b_m)$ (9)

Οι εξισώσεις (7) δίνουν την πρώτη βασική αποδεκτή λύση, η οποία είναι αναγκαία για να ξεκινήσει η φάση Ι.

4) Η ποσότητα *w* λέγεται “φόρμα μη αποδεκτής λύσης” και έχει την ιδιότητα ότι εάν σαν αποτέλεσμα της φάσης Ι προκύψει ότι η ελάχιστη τιμή της *w* είναι $w > 0$, τότε δεν υπάρχει

αποδεκτή λύση για το αρχικό πρόβλημα γραμμικού προγραμματισμού, όπως ορίσθηκε με τις εξισώσεις (1) και (2) και έτσι η διαδικασία ολοκληρώνεται εδώ.

Αν η ελάχιστη τιμή της *w* προκύψει από την φάση I ότι είναι *w = 0*, τότε το σύστημα (7) θα είναι σε κανονική μορφή και έτσι ξεκινά η φάση II με την απάλειψη της εξίσωσης w καθώς και των στηλών που αντιστοιχούν σε κάθε μια από τις τεχνητές μεταβλητές $y_1, y_2, \ldots, y_m$ από το σύστημα.

5) Φάση II της μεθόδου.

   Εφαρμόζουμε τον αλγόριθμο Simplex στο τροποποιημένο σύστημα εξισώσεων σε κανονική μορφή, όπως διαμορφώθηκε στο τέλος της φάσης I, ώστε να προκύψει μια λύση, η οποία ελαχιστοποιεί την τιμή της *f*, εάν φυσικά υπάρχει.

3. Υπολογιστικός αλγόριθμος

3.1 Περιγραφή της υπολογιστικής διαδικασίας

Ο προγραμματισμός της υπολογιστικής διαδικασίας βασίζεται εξ' ολοκλήρου στην θεωρία που περιγράφεται στο 2ο Κεφάλαιο της παρούσας εργασίας. Παρακάτω περιγράφονται αναλυτικά τα βήματα που ακολουθεί ο υπολογιστικός αλγόριθμος του λογισμικού:

- Έλεγχος ύπαρξης αρνητικής τιμής στο δεξιό μέλος των περιορισμών (στην περίπτωση ύπαρξης η υπολογιστική διαδικασία τερματίζεται).
- Συλλογή δεδομένων για τον τύπο των περιορισμών (δηλ. ≤, ≥, =).
- Δημιουργία αρχικού πίνακα Simplex και εισαγωγή των δεδομένων σε αυτόν (δηλ. συντελεστές φυσικών μεταβλητών αντικειμενικής συνάρτησης και περιορισμών, συντελεστές τεχνητών μεταβλητών και ψευδομεταβλητών περιορισμών και δεξιά μέλη περιορισμών).
- Υπολογισμός τεχνητής συνάρτησης *w* και εισαγωγή της στον πίνακα Simplex.
- Εύρεση των βασικών και μη βασικών μεταβλητών και έλεγχος διαθεσιμότητας αρχικής λύσης.
- Εφαρμογή επαναληπτικού αλγορίθμου:
    1. Εύρεση της μεταβλητής που θα εξέλθει από την βάση.
    2. Εύρεση της μεταβλητής που θα εισέλθει στην βάση.
    3. Υπολογισμός του επόμενου πίνακα Simplex.
    4. Εύρεση των νέων βασικών και μη βασικών μεταβλητών.
    5. Έλεγχος διαθεσιμότητας επόμενης βελτιωμένης λύσης.
5. Έλεγχος του μέτρου της ποσότητας *w* και ανάλογα τερματισμός της διαδικασίας (w>0) ή εφαρμογή της 2ης Φάσης (w=0).
6. Εφόσον *w=0*, αφαιρώ τις στήλες των τεχνητών μεταβλητών και την γραμμή της τεχνητής συνάρτησης *w*.
7. Εύρεση των νέων βασικών και μη βασικών μεταβλητών και έλεγχος διαθεσιμότητας αρχικής λύσης για την 2η Φάση.
8. Εφαρμογή επαναληπτικού αλγορίθμου (βλέπε βήματα 1ης Φάσης.
9. Εξαγωγή των αποτελεσμάτων από τον τελικό πίνακα Simplex.

3.2 Τεκμηρίωση των βασικών συναρτήσεων της υπολογιστικής διαδικασίας

Στην παρούσα ενότητα τεκμηριώνονται οι βασικές συναρτήσεις που συμβάλουν στην διεκπεραίωση της υπολογιστικής διαδικασίας. Παρακάτω παρατίθενται οι πέντε βασικότερες από αυτές:

- **calculation_of_non_basic_variables()**

```
Public Shared Function calculation_of_non_basic_variables(n As Integer, number_of_variables(,) As Double,
  basic_variables(,) As Double, rev_ineq As Integer, artificial_variables(,) As Double, phase As Integer) As Double(,)

    Dim non_basic_variables(,) As Double

    If phase = 1 Then
        Dim non_basic_variablesAux(n + rev_ineq - 1, 0) As Double
        non_basic_variables = non_basic_variablesAux

    ElseIf phase = 2 Then
        Dim non_basic_variablesAux(n + rev_ineq - 1 - artificial_variables.Length, 0) As Double
        non_basic_variables = non_basic_variablesAux

    End If

    Dim k = 0

    For i = 0 To number_of_variables.Length - 1
        Dim contains = 0

        For j = 0 To basic_variables.Length - 1
            If number_of_variables(0, i) = basic_variables(j, 0) Then
                contains = 1
            End If
        Next

        If contains = 0 Then
            non_basic_variables(k, 0) = number_of_variables(0, i)
            k = k + 1
        End If
    Next
    Return non_basic_variables
End Function
```

Η *calculation_of_non_basic_variables* είναι μια συνάρτηση η οποία βρίσκει ποιες είναι οι μη βασικές μεταβλητές σε κάθε φάση του προβλήματος. Η πληροφορία αυτή είναι από τις σημαντικότερες καθώς είναι αναγκαία σε πολλά στάδια της υπολογιστικής διαδικασίας, όπως γίνεται αντιληπτό και από την περιγραφή της στην προηγούμενη υποενότητα. Από τον κώδικα φαίνεται ότι η συνάρτηση αυτή λαμβάνει ως είσοδο τον αριθμό των μεταβλητών, τις βασικές μεταβλητές, τις τεχνητές μεταβλητές, τον αριθμό των αντιστροφών ανισοτήτων (≥), τη φάση (phase) στην οποία βρίσκεται την δεδομένη στιγμή, και επιστρέφει έναν πίνακα ο οποίος περιέχει τις μη βασικές μεταβλητές. Αρχικά, δημιουργεί τον πίνακα *non_basic_variables* και ανάλογα σε ποια από τις δύο φάσεις βρίσκεται μεταβάλλεται το μέγεθος του πίνακα (διότι στη δεύτερη φάση οι τεχνητές μεταβλητές αφαιρούνται) και με την πληροφορία των βασικών μεταβλητών βρίσκει τις μη βασικές και τις καταχωρεί σε αυτόν με την βοήθεια της μεταβλητής *k*

- **Availability_of_optimized_solution()**

```
4 references
Public Shared Function Availability_of_optimized_solution(simplex(,) As Double,
     non_basic_variables(,) As Double, m As Integer, phase As Integer) As Integer

    Dim available As Integer = 0

    For i = 0 To non_basic_variables.Length - 1

        Dim pick As Double = non_basic_variables(i, 0)

        If phase = 1 Then
            If simplex(m + 1, pick) < 0 Then
                available = 1
            End If
        Else
            If simplex(m, pick) < 0 Then
                available = 1
            End If
        End If
    Next

    Return available

End Function
```

Αυτή είναι μια συνάρτηση η οποία χρησιμοποιείται για τον έλεγχο της διαθεσιμότητας επόμενης λύσης στον αλγόριθμο Simplex. Η συνάρτηση αυτή δέχεται ως είσοδο τον πίνακα Simplex, τον πίνακα *non_basic_variables* που περιέχει τις μη βασικές μεταβλητές, τον αριθμό των περιορισμών *m* και τη φάση του προβλήματος *phase*.

Στη συνέχεια, η συνάρτηση ανάλογα σε ποια φάση βρίσκεται ελέγχει κάθε μη βασική μεταβλητή από την τελευταία γραμμή του πίνακα Simplex για να αντιληφθεί αν κάποια έχει αρνητικό συντελεστή. Αν έστω και μία μη βασική μεταβλητή έχει αρνητικό συντελεστή, τότε υπάρχει επόμενη διαθέσιμη λύση και η μεταβλητή *available* θα πάρει την τιμή 1, διαφορετικά η τιμή θα παραμείνει 0. Τέλος, η συνάρτηση επιστρέφει την τιμή της μεταβλητής *available*.

- **Entry_Variable()**

```
Public Shared Function Entry_Variable(Simplex(,) As Double, m As Integer, nm As Integer, rev_ineq As Integer,
        phase As Integer, non_basic_variables(,) As Double, artificialVariablesLength As Integer) As Integer

    Dim entry As Integer

    Dim entry_vector(0, non_basic_variables.Length - 1) As Double

    If phase = 1 Then

        Dim myArray(0, nm + rev_ineq - 1) As Double

        For i = 0 To nm + rev_ineq - 1
            myArray(0, i) = Simplex(m + 1, i)
        Next

        For j = 0 To non_basic_variables.Length - 1
            entry_vector(0, j) = myArray(0, non_basic_variables(j, 0))
        Next

    Else

        Dim myArray(0, nm + rev_ineq - 1 - artificialVariablesLength) As Double

        For i = 0 To nm + rev_ineq - 1 - artificialVariablesLength
            myArray(0, i) = Simplex(m, i)
        Next

        For j = 0 To non_basic_variables.Length - 1
            entry_vector(0, j) = myArray(0, non_basic_variables(j, 0))
        Next

    End If

    For i = 0 To non_basic_variables.Length - 1
        If entry_vector(0, i) = 0 Then
            entry = -1
            Return entry
        End If
    Next

    Dim minElement As Double = 99999999999999
    Dim entryValue As Integer

    For i = 0 To entry_vector.Length - 1

        If (entry_vector(0, i) < 0) AndAlso (entry_vector(0, i) < minElement) Then
            minElement = entry_vector(0, i)
            entryValue = i
        End If
    Next

    entry = non_basic_variables(entryValue, 0)

    Return entry

End Function
```

Η συνάρτηση *Entry_Variable* υπολογίζει την κατάλληλη μη βασική μεταβλητή για να εισέλθει στη βάση στην επόμενη επανάληψη του αλγορίθμου Simplex. Παίρνει ως είσοδο τον πίνακα Simplex, όπως και τα απαραίτητα στοιχεία του πίνακα, την φάση του αλγορίθμου (phase), το διάνυσμα των μη βασικών μεταβλητών και το μέγεθος του διανύσματος των τεχνητών μεταβλητών. Ανάλογα με τη φάση του αλγορίθμου, αντλεί τις αντίστοιχες τιμές από τον πίνακα Simplex και δημιουργεί ένα διάνυσμα (entry_vector) από τους συντελεστές των μη βασικών μεταβλητών. Στη συνέχεια, ελέγχει αν υπάρχει κάποια μη βασική μεταβλητή που έχει συντελεστή 0, και αν ναι επιστρέφει -1, έτσι

ώστε να ανιχνευθεί η περίπτωση πολλαπλών άριστων λύσεων (βλέπε ενότητα 2.2.5). Τέλος, υπολογίζει τη μεταβλητή που έχει το μικρότερο αρνητικό συντελεστή από το διάνυσμα *entry_vector* και επιστρέφει την θέση της.

- **Exit_Variable()**

```
Public Shared Function Exit_Variable(Simplex(,) As Double, m As Integer, nm As Integer, rev_ineq As Integer,
                                     phase As Integer, artificial_variables(,) As Double, entry As Double) As Integer

    Dim exit_vector(m - 1, 0) As Double

    For i = 0 To m - 1
        If Simplex(i, entry) < 0 Then
            exit_vector(i, 0) = -1
        ElseIf phase = 1 Then
            exit_vector(i, 0) = Simplex(i, nm + rev_ineq) / Simplex(i, entry)
        ElseIf phase = 2 Then
            exit_vector(i, 0) = Simplex(i, nm + rev_ineq - artificial_variables.Length) / Simplex(i, entry)
        End If
    Next

    Dim _exit As Integer
    Dim minElement As Double = 999999999999999999
    Dim exitNegative As Boolean = False

    For i = 0 To m - 1
        If (exit_vector(i, 0) >= 0) AndAlso (exit_vector(i, 0) < minElement) Then
            minElement = exit_vector(i, 0)
            _exit = i
            exitNegative = True
        End If
    Next

    If exitNegative = False Then
        _exit = -1
    End If

    Return _exit
End Function
```

Η συνάρτηση *Exit_Variable* υπολογίζει την μεταβλητή που θα εξέλθει από την βάση σε κάθε επαναληπτική διαδικασία του αλγορίθμου Simplex. Λαμβάνει ως είσοδο τον πίνακα Simplex και τα κύρια χαρακτηριστικά του, καθώς και άλλες μεταβλητές όπως ο αριθμός των μεταβλητών απόφασης, ο αριθμός των περιορισμών, η φάση, οι τεχνητές μεταβλητές, και η μεταβλητή που εισέρχεται στη βάση και σύμφωνα με την στήλη στην οποία βρίσκεται επιλέγεται ποια μεταβλητή θα εξέλθει από τη βάση.

Η συνάρτηση αρχικά, δημιουργεί ένα διάνυσμα (exit_vector) με μέγεθος ανάλογο του αριθμού των περιορισμών του προβλήματος. Έπειτα υπολογίζει για κάθε μια από τις τρέχουσες βασικές μεταβλητές το θετικό πηλίκο b_i/a_{ij}, όπου b_i είναι το δεξιό μέλος που εκφράζει την τρέχουσα τιμή

της αντίστοιχης βασικής μεταβλητής και a_{ij} ο συντελεστής της στη στήλη του πίνακα Simplex που αντιστοιχεί στη μεταβλητή που μόλις εισήλθε στη βάση (βλέπε Κεφάλαιο 2) και στην συνέχεια προσθέτει το αποτέλεσμα του κάθε πηλίκου στο διάνυσμα *exit_vector*. Τέλος, ελέγχει κάθε στοιχείο του διανύσματος για να βρει το ελάχιστο θετικό στοιχείο, έτσι ώστε να βρεθεί η μεταβλητή που θα εξέλθει από την βάση.

- **new_Simplex_calculation()**

```
Public Shared Function new_Simplex_calculation(simplex(,) As Double, _exit As Integer, entry As Integer, m As Integer,
                  nm As Integer, rev_ineq As Integer, phase As Integer, artificial_variables(,) As Double) As Double(,)

    Dim driver_element As Double = simplex(_exit, entry)
    Dim Simplex_New(m + 1, nm + rev_ineq) As Double

    If phase = 1 Then

        For i = 0 To nm + rev_ineq
            simplex(_exit, i) = simplex(_exit, i) / driver_element
        Next

        Dim Simplex_NewAux(m + 1, nm + rev_ineq) As Double

        For i = 0 To m + 1
            For j = 0 To nm + rev_ineq
                Simplex_NewAux(i, j) = (simplex(i, j) - simplex(_exit, j) * simplex(i, entry))
            Next
        Next

        Simplex_New = Simplex_NewAux

        Dim driver_element_row(0, nm + rev_ineq) As Double

        For i = 0 To nm + rev_ineq
            driver_element_row(0, i) = simplex(_exit, i)
        Next

        simplex = Simplex_New

        For i = 0 To nm + rev_ineq
            simplex(_exit, i) = simplex(_exit, i) + driver_element_row(0, i)
        Next
```

```
        Else

            For i = 0 To nm + rev_ineq - artificial_variables.Length
                simplex(_exit, i) = simplex(_exit, i) / driver_element
            Next

            Dim Simplex_NewAux(m, nm + rev_ineq - artificial_variables.Length) As Double

            For i = 0 To m
                For j = 0 To nm + rev_ineq - artificial_variables.Length
                    Simplex_NewAux(i, j) = (simplex(i, j) - simplex(_exit, j) * simplex(i, entry))
                Next
            Next

            Simplex_New = Simplex_NewAux


            Dim driver_element_row(0, nm + rev_ineq - artificial_variables.Length) As Double

            For i = 0 To nm + rev_ineq - artificial_variables.Length
                driver_element_row(0, i) = simplex(_exit, i)
            Next

            simplex = Simplex_New

            For i = 0 To nm + rev_ineq - artificial_variables.Length
                simplex(_exit, i) = simplex(_exit, i) + driver_element_row(0, i)
            Next

        End If

        Return simplex

    End Function
```

Τέλος, η συνάρτηση *new_Simplex_calculation* είναι αυτή η οποία υπολογίζει τον νέο πίνακα Simplex, ο οποίος περιέχει την νέα βασική λύση, σε κάθε επαναληπτική διαδικασία και στις δυο φάσεις της μεθόδου. Συγκεκριμένα, η συνάρτηση λαμβάνει ως είσοδο τον πίνακα Simplex και τα κύρια χαρακτηριστικά του, την μεταβλητή που εισέρχεται στη βάση καθώς και τη μεταβλητή που εξέρχεται από αυτήν, τον αριθμό των μεταβλητών απόφασης και τον αριθμό των περιορισμών, την φάση στην οποία βρίσκεται το πρόβλημα και τις τεχνητές μεταβλητές, και δίνει σαν έξοδο τον νέο πίνακα Simplex.

Οι διαδικασίες που εκτελεί η συνάρτηση αυτή, είναι κατά γράμμα τα βήματα που περιγράφονται στην παράγραφο του προσδιορισμού της νέας βασικής δυνατής λύσης στην ενότητα 2.2.5 της παρούσας εργασίας.

4. Περιβάλλον διεπαφής με τον χρήστη

4.1 Περιγραφή του περιβάλλοντος διεπαφής

Κατά την εκκίνηση του λογισμικού ανοίγει το περιβάλλον με το οποίο έρχεται σε διεπαφή ο χρήστης. Η γραφική δομή του φαίνεται στην παρακάτω εικόνα και τα στοιχεία που περιέχονται σε αυτή είναι τα εξής:

(1) Κουμπί το οποίο ανοίγει την εξερεύνηση αρχείων του Η/Υ για την φόρτωση υπολογιστικού φύλλου (.csv),

(2) Κουμπί το οποίο ανοίγει το εγχειρίδιο (.pdf) με οδηγίες ως προς την απαραίτητη δομή του υπολογιστικού φύλλου αλλά και την διαδικασία εισαγωγής του στην εφαρμογή.

(3) Πλαίσια εισαγωγής κειμένου στα οποία ο χρήστης συμπληρώνει τον αριθμό των μεταβλητών της αντικειμενικής συνάρτησης και τον αριθμό των περιορισμών του προβλήματος που θέλει να επιλύσει.

(4) Λίστα από την οποία ο χρήστης επιλέγει αν το πρόβλημα που θέλει να επιλύσει είναι ελαχιστοποίησης (Min) ή μεγιστοποίησης (Max).

(5) Πλαίσια εισαγωγής στα οποία ο χρήστης συμπληρώνει τους συντελεστές των μεταβλητών απόφασης.

(6) Πλαίσια εισαγωγής στα οποία ο χρήστης συμπληρώνει τους συντελεστές των μεταβλητών του κάθε περιορισμού.

(7) Λίστα από την οποία ο χρήστης επιλέγει τον τύπο της ισότητας του κάθε περιορισμού (≥, ≤, =).

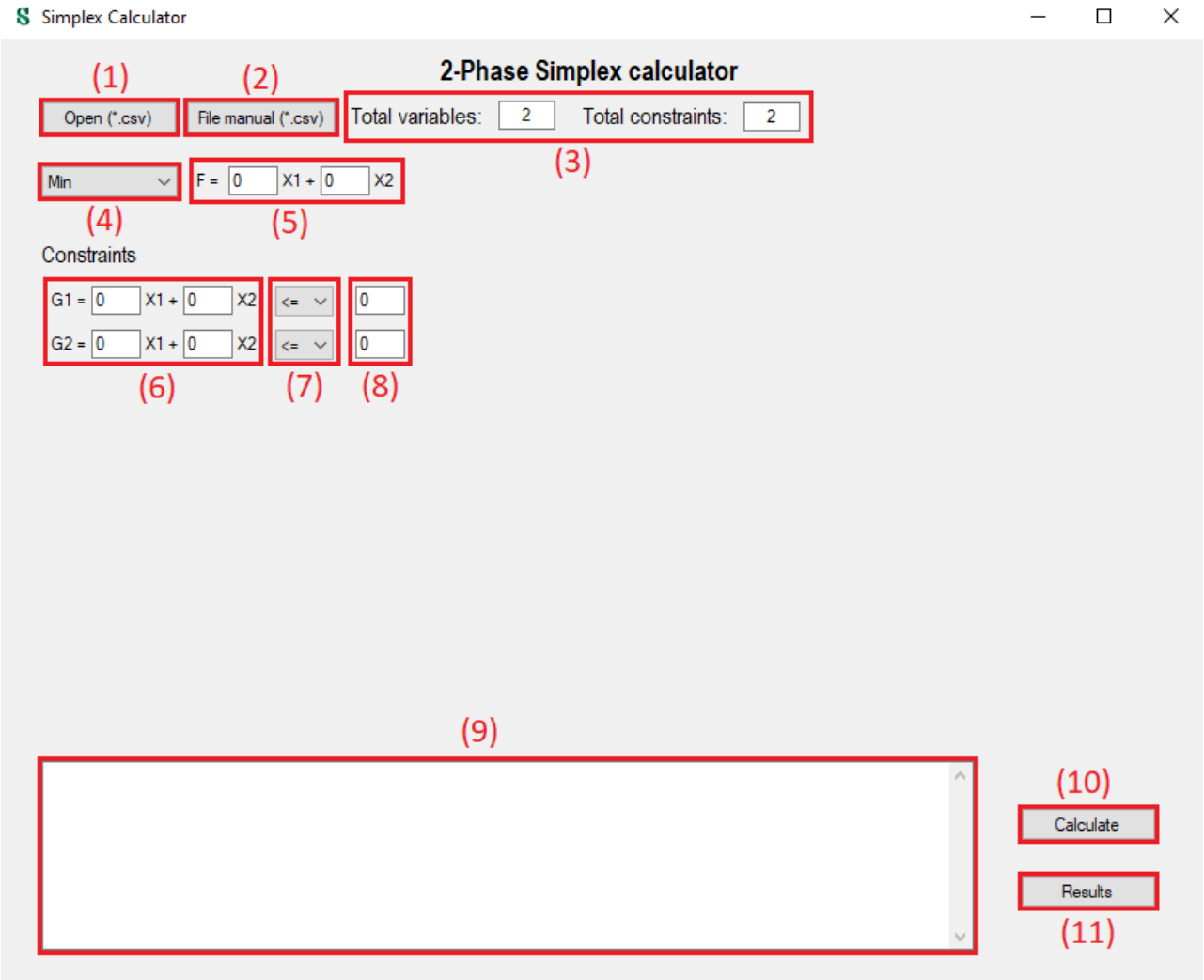
(8) Πλαίσια εισαγωγής στα οποία ο χρήστης συμπληρώνει την τιμή του δεξιού μέλους κάθε περιορισμού.

(9) Πλαίσιο εμφάνισης μηνυμάτων (π.χ. αποτέλεσμα, σφάλματα).

(10) Κουμπί το οποίο εκκινεί την διαδικασία του υπολογισμού.

(11) Κουμπί το οποίο ανοίγει ένα αρχείο .txt το οποίο περιέχει τους ενδιάμεσους πίνακες που δημιουργούνται σε κάθε επανάληψή του αλγορίθμου Simplex.

(12) Επιγραφές (π.χ. Total variables, Total constraints, Constraints κ.α.).



4.2 Ανάλυση λειτουργίας του περιβάλλοντος διεπαφής

Όπως παρατηρείται στην ενότητα 4.1 εκτός από την ύπαρξη των πλαισίων εισαγωγής κειμένου για την εισαγωγή των δεδομένων στο λογισμικό, δίνεται και η δυνατότητα φόρτωσης υπολογιστικού φύλλου. Στην ενότητα αυτή αναλύεται η λειτουργία του λογισμικού και για τις δυο περιπτώσεις εισαγωγής δεδομένων.

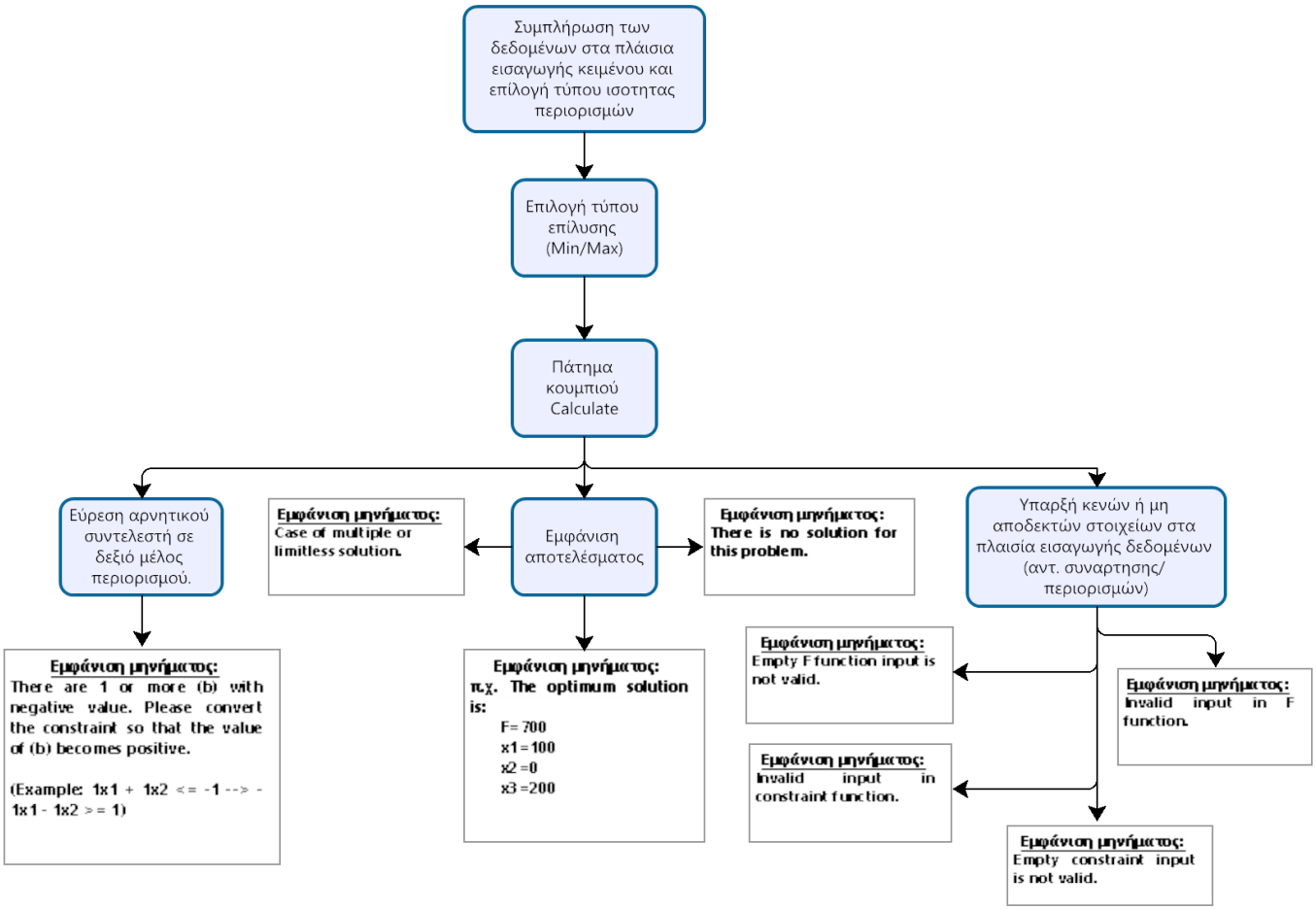
4.2.1 Εισαγωγή δεδομένων μέσω των πλαισίων εισαγωγής κειμένου

Για την περίπτωση επίλυσης προβλήματος γραμμικού προγραμματισμού μικρού μεγέθους (έως 10 μεταβλητές και έως 10 περιορισμούς) δίνεται η δυνατότητα από το λογισμικό, η εισαγωγή των δεδομένων να γίνει χειροκίνητα, με την συμπλήρωση των πλαισίων εισαγωγής κειμένου από τον χρήστη. Η διαδικασία αποτελείται από τα εξής βήματα:

1. Συμπλήρωση των δεδομένων του προβλήματος στα πλαίσια εισαγωγής κειμένου και επιλογή τύπου ισότητας περιορισμών.
2. Επιλογή τύπου επίλυσης προβλήματος (μεγιστοποίησης ή ελαχιστοποίησης).
3. Πάτημα του κουμπιού *Calculate*.

Έπειτα από την εκκίνηση της διαδικασίας υπολογισμού πραγματοποιούνται στα πρώτα στάδια του προγράμματος κάποιοι έλεγχοι, όπως η αναζήτηση για αρνητικό συντελεστή στο δεξιό μέλος περιορισμού και η ύπαρξη κενών ή μη αποδεκτών στοιχείων στα πλαίσια εισαγωγής κειμένου. Στην περίπτωση εύρεσης κάποιου από το προαναφερθέντα σφάλματα εμφανίζεται στο πλαίσιο εμφάνισης μηνυμάτων το αντίστοιχο μήνυμα. Εάν δεν βρεθεί κάποιο σφάλμα τότε η διαδικασία του υπολογισμού ολοκληρώνεται εμφανίζοντας το αποτέλεσμα στο πλαίσιο εμφάνισης μηνυμάτων και δίνοντας την δυνατότητα εμφάνισης των ενδιάμεσων πινάκων πατώντας το κουμπί *Results*. Το αποτέλεσμα, όπως περιγράφεται και στην ενότητα 2, μπορεί να είναι η βέλτιστη λύση, περίπτωση μη διαθέσιμης αποδεκτής λύσης ή περίπτωση μη περιορισμένης λύσης.

Στην παρακάτω εικόνα πραγματοποιείται μια διαγραμματική απεικόνιση της λειτουργίας του λογισμικού για την περίπτωση της εισαγωγής δεδομένων μέσω των πλαισίων εισαγωγής κειμένου απεικονίζοντας την ροή της διαδικασίας καθώς και τα μηνύματα που εμφανίζονται στο πλαίσιο εμφάνισης μηνυμάτων σε κάθε περίπτωση.



4.2.2 Εισαγωγή δεδομένων μέσω υπολογιστικού φύλλου

Για την επίλυση προβλημάτων μεγαλύτερου μεγέθους, το λογισμικό παρέχει την δυνατότητα εισαγωγής των δεδομένων του προβλήματος μέσω υπολογιστικού φύλλου τύπου *csv*. Για την εισαγωγή του υπολογιστικού φύλλου στο λογισμικό και την ορθή επίλυση του προβλήματος είναι αναγκαία η ορθή δόμηση των κελιών του, η οποία περιγράφεται αναλυτικά παρακάτω.

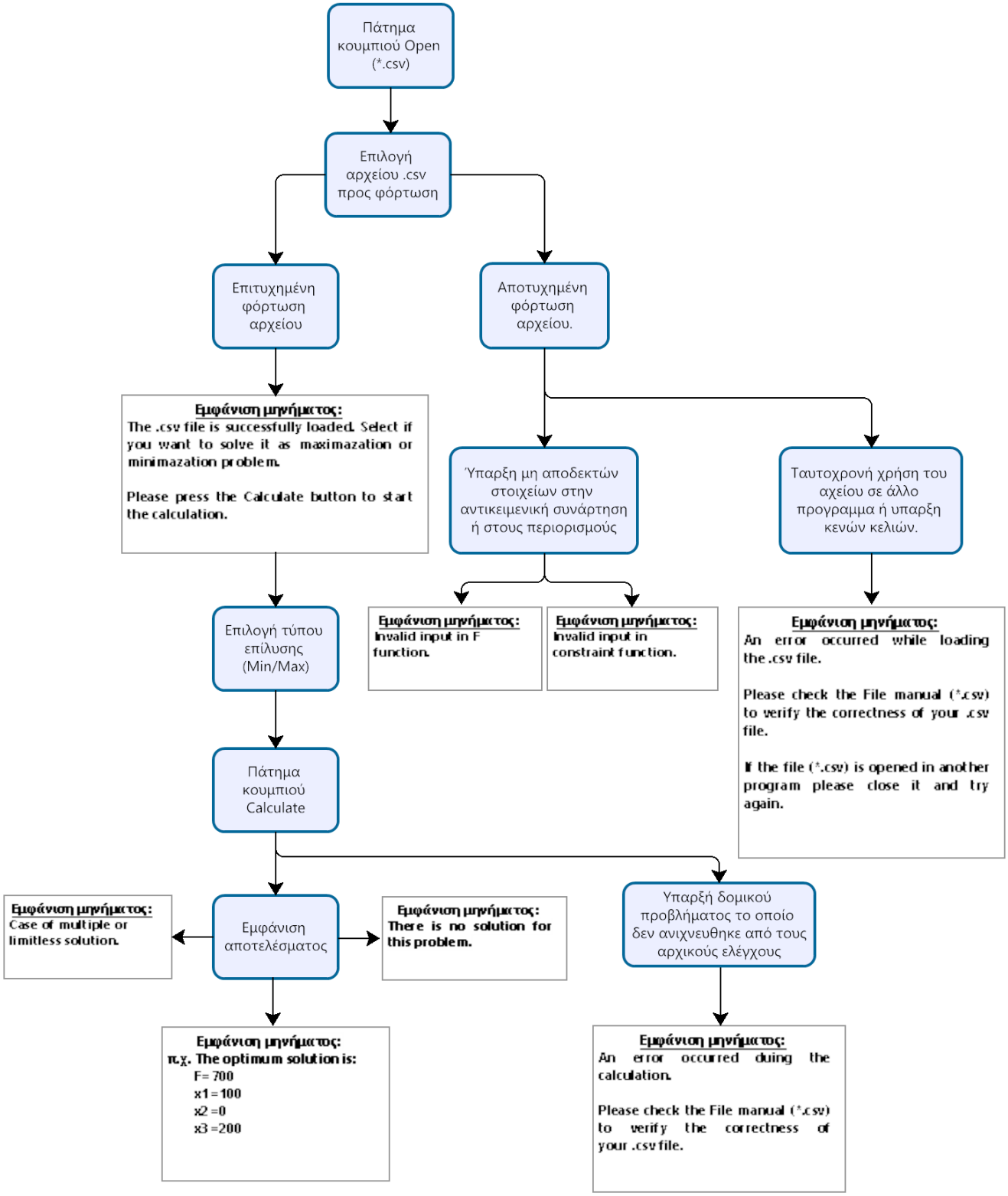
Αρχικά, όπως φαίνεται και στην παρακάτω εικόνα στην πρώτη γραμμή του φύλλου συμπληρώνονται οι συντελεστές των μεταβλητών της αντικειμενικής συνάρτησης. Στις παρακάτω σειρές συμπληρώνονται οι περιορισμοί δομής (ένας σε κάθε γραμμή χωρίς κάποιο περιορισμό για τον αριθμό των περιορισμών που μπορούν να εισαχθούν). Συγκεκριμένα, στα πρώτα κελιά της κάθε γραμμής τοποθετούνται οι συντελεστές των μεταβλητών απόφασης του περιορισμού, έπειτα στο επόμενο κελί τοποθετείται το δεξιό μέλος του (δηλαδή την τιμή που υπάρχει μετά την ισότητα του περιορισμού) και στο τελευταίο κελί συμπληρώνεται ένας αριθμός που αντιπροσωπεύει τον τύπο της ισότητας (1 → = , 2 → ≤ , 3 → ≥).

	A	B	C	D	E	F	G
1	x1	x2	...	xn	<-- F		
2	x1	x2	...	xn	Δ.Μ.	Τύπος Ισότητας	<-- G1
3	-//-	-//-	...	-//-	-//-	-//-	<-- G2
4	...	...	...	...	...	...	...
5	-//-	-//-	...	-//-	-//-	-//-	<-- Gn

Έπειτα από την σύνταξη του υπολογιστικού φύλλου και την αποθήκευσή του σε αρχείο τύπου *csv*, η διαδικασία για την επίλυση του προβλήματος είναι η εξής:

1. Πάτημα του κουμπιού Open (*.csv), το οποίο ανοίγει την εξερεύνηση αρχείων του υπολογιστή.
2. Επιλογή του αρχείου που θα εισαχθεί στο λογισμικό.
3. Εμφάνιση μηνύματος επιτυχημένης ή αποτυχημένης φόρτωσης του αρχείου.
4. Στην περίπτωση επιτυχημένης φόρτωσης πραγματοποιείται η επιλογή του «τύπου ισότητας».
5. Πάτημα του κουμπιού *Calculate*.

Η κύρια διαφορά αυτής της διαδικασίας με αυτή που περιγράφεται στην προηγούμενη υποενότητα είναι ότι οι έλεγχοι για αρνητικό συντελεστή στο δεξιό μέλος περιορισμού και για ύπαρξη κενών ή μη αποδεκτών στοιχείων στα πλαίσια εισαγωγής κειμένου πραγματοποιούνται πριν την εκκίνηση της διαδικασίας υπολογισμού κατά την φόρτωση του αρχείου, όπως και εάν το αρχείο είναι ανοιχτό σε κάποια άλλη εφαρμογή και στην περίπτωση εύρεσης κάποιου σφάλματος εμφανίζεται πάλι στο πλαίσιο εμφάνισης μηνυμάτων το αντίστοιχο μήνυμα. Τέλος, στην παρακάτω εικόνα απεικονίζεται και πάλι διαγραμματικά η λειτουργία του λογισμικού καθώς και τα μηνύματα που εμφανίζονται στο πλαίσιο εμφάνισης μηνυμάτων σε κάθε περίπτωση.


Πάτημα κουμπιού Open (*.csv)
Επιλογή αρχείου .csv προς φόρτωση
Επιτυχημένη φόρτωση αρχείου
Αποτυχημένη φόρτωση αρχείου.
Εμφάνιση μηνήματος:
The .csv file is successfully loaded. Select if you want to solve it as maximazation or minimazation problem.
Please press the Calculate button to start the calculation.
Ύπαρξη μη αποδεκτών στοιχείων στην αντικειμενική συνάρτηση ή στους περιορισμούς
Ταυτοχρονή χρήση του αχείου σε άλλο προγραμμα ή υπαρξη κενών κελιών.
Επιλογή τύπου επίλυσης (Min/Max)
Εμφάνιση μηνήματος:
Invalid input in F function.
Εμφάνιση μηνήματος:
Invalid input in constraint function.
Εμφάνιση μηνήματος:
An error occurred while loading the .csv file.
Please check the File manual (*.csv) to verify the correctness of your .csv file.
If the file (*.csv) is opened in another program please close it and try again.
Πάτημα κουμπιού Calculate
Εμφάνιση μηνήματος:
Case of multiple or limitless solution.
Εμφάνιση αποτελέσματος
Εμφάνιση μηνήματος:
There is no solution for this problem.
Υπαρξή δομικού προβλήματος το οποίο δεν ανιχνευθηκε από τους αρχικούς ελέγχους
Εμφάνιση μηνήματος:
π.χ. The optimum solution is:
F= 700
x1 =100
x2 =0
x3 =200
Εμφάνιση μηνήματος:
An error occurred duing the calculation.
Please check the File manual (*.csv) to verify the correctness of your .csv file.

4.3 Τεκμηρίωση των βασικών συναρτήσεων του περιβάλλοντος διεπαφής

Στην ενότητα αυτή πραγματοποιείται η τεκμηρίωση των βασικότερων συναρτήσεων με τη βοήθεια των οποίων δημιουργείται το γραφικό περιβάλλον διεπαφής του χρήστη με το λογισμικό αλλά πραγματοποιούνται και οι λειτουργίες σε αυτό. Παρακάτω παρατίθενται οι βασικότερες από αυτές:

- **Dispose()**

```
'Form overrides dispose to clean up the component list.
<System.Diagnostics.DebuggerNonUserCode()>
0 references
Protected Overrides Sub Dispose(ByVal disposing As Boolean)
    Try
        If disposing AndAlso components IsNot Nothing Then
            components.Dispose()
        End If
    Finally
        MyBase.Dispose(disposing)
    End Try
End Sub
```

Η παραπάνω συνάρτηση χρησιμοποιείται για να αποδεσμεύσει τους πόρους του συστήματος που αντιστοιχούν στα αντικείμενα της φόρμας όταν αυτή κλείνει. Αυτό συνήθως συμβαίνει όταν ο χρήστης κλείνει την εφαρμογή. Η *Dispose* συνάρτηση είναι υπεύθυνη για τη διαγραφή των αντικειμένων που έχουν δημιουργηθεί από τη φόρμα και των πόρων που έχουν καταναλωθεί από αυτήν. Αυτό γίνεται για να μην υπάρχει κατανάλωση μνήμης ή άλλων πόρων του συστήματος από την εφαρμογή όταν αυτή δεν είναι πλέον απαραίτητη.

- **InitializeComponent()**

```
Private Sub InitializeComponent()
    Dim resources As System.ComponentModel.ComponentResourceManager = New System.ComponentModel.ComponentResourceManager(GetType(Form1))
    'Me.GroupBox1 = New System.Windows.Forms.GroupBox()
    Me.Label3 = New System.Windows.Forms.Label()
    Me.ComboBox1 = New System.Windows.Forms.ComboBox()
    Me.Button1 = New System.Windows.Forms.Button()
    Me.TextBox2 = New System.Windows.Forms.TextBox()
    Me.TextBox1 = New System.Windows.Forms.TextBox()
    Me.Label2 = New System.Windows.Forms.Label()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.totalVariablesLabel = New System.Windows.Forms.Label()
    Me.totalVarsNumberTextBox = New System.Windows.Forms.TextBox()
    Me.BackgroundWorker1 = New System.ComponentModel.BackgroundWorker()
    Me.Label5 = New System.Windows.Forms.Label()
    Me.totalConstraintLabel = New System.Windows.Forms.Label()
    Me.totalConstNumberTextBox = New System.Windows.Forms.TextBox()
    Me.minMaxComboBox = New System.Windows.Forms.ComboBox()
    Me.constraintsPanel = New System.Windows.Forms.Panel()
    Me.Label4 = New System.Windows.Forms.Label()
    Me.calculateButton = New System.Windows.Forms.Button()
    Me.csvButton = New System.Windows.Forms.Button()
    Me.displayPannel = New System.Windows.Forms.ListBox()
    Me.csvManualButton = New System.Windows.Forms.Button()
    'Me.GroupBox1.SuspendLayout()
    Me.SuspendLayout()
```

Αρχικά, όπως φαίνεται στον κώδικα παραπάνω, η συνάρτηση δεσμεύει την απαραίτητη μνήμη για την αρχικοποίηση των οπτικών αντικειμένων που έχουν προστεθεί στην φόρμα.

```
    'totalConstNumberTextBox
    '
    Me.totalConstNumberTextBox.Location = New System.Drawing.Point(510, 43)
    Me.totalConstNumberTextBox.Name = "totalConstNumberTextBox"
    Me.totalConstNumberTextBox.Size = New System.Drawing.Size(39, 20)
    Me.totalConstNumberTextBox.TabIndex = 5
    Me.totalConstNumberTextBox.TextAlign = System.Windows.Forms.HorizontalAlignment.Center
    '
    'minMaxComboBox
    '
    Me.minMaxComboBox.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
    Me.minMaxComboBox.FormattingEnabled = True
    Me.minMaxComboBox.Items.AddRange(New Object() {"Min", "Max"})
    Me.minMaxComboBox.Location = New System.Drawing.Point(29, 87)
    Me.minMaxComboBox.Name = "minMaxComboBox"
    Me.minMaxComboBox.Size = New System.Drawing.Size(93, 21)
    Me.minMaxComboBox.TabIndex = 7
    '
    'constraintsPanel
    '
    Me.constraintsPanel.AutoScroll = True
    Me.constraintsPanel.Location = New System.Drawing.Point(29, 161)
    Me.constraintsPanel.Name = "constraintsPanel"
    Me.constraintsPanel.Size = New System.Drawing.Size(764, 328)
    Me.constraintsPanel.TabIndex = 9
```

Επίσης, στην συνάρτηση αυτή ορίζονται τα χαρακτηριστικά και οι ιδιότητες (μέγεθος, τοποθεσία κτλ.) κάθε οπτικού αντικειμένου που προβάλλονται στην φόρμα κατά το άνοιγμα της εφαρμογής.

```
    '
    'Form1
    '
    Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
    Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
    Me.ClientSize = New System.Drawing.Size(826, 648)
    Me.Controls.Add(Me.csvManualButton)
    Me.Controls.Add(Me.displayPannel)
    Me.Controls.Add(Me.csvButton)
    Me.Controls.Add(Me.calculateButton)
    Me.Controls.Add(Me.Label4)
    Me.Controls.Add(Me.constraintsPanel)
    Me.Controls.Add(Me.minMaxComboBox)
    Me.Controls.Add(Me.totalConstNumberTextBox)
    Me.Controls.Add(Me.totalConstraintLabel)
    Me.Controls.Add(Me.Label5)
    Me.Controls.Add(Me.totalVarsNumberTextBox)
    Me.Controls.Add(Me.totalVariablesLabel)
    Me.Font = New System.Drawing.Font("Microsoft Sans Serif", 8.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(161, Byte))
    Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
    Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
    Me.Name = "Form1"
    Me.Text = "Simplex Calculator"
    Me.ResumeLayout(False)
    Me.PerformLayout()

End Sub
```

Τέλος, πραγματοποιεί την πρόσθεση των αντικειμένων στην φόρμα και ορίζει ορισμένα χαρακτηριστικά της (π.χ. όνομα, εικονίδιο φόρμας).

- **Form1_Load()**

```
0 references
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
    ' Show the form in the center '
    Me.CenterToScreen()
End Sub
```

Αυτή η συνάρτηση εκτελείται όταν η φόρμα φορτώνει. Η μέθοδος *CenterToScreen* εξυπηρετεί για να τοποθετήσει τη φόρμα στο κέντρο της οθόνης κατά την εκκίνηση της εφαρμογής.

- **InitializeTextBox()**

```
5 references
Private Function InitializeTextBox(pointX As Double, pointY As Double, width As Double, height As Double, text As String) As TextBox

    Dim textBox As New TextBox With {
        .Size = New Size(width, height),
        .Location = New Point(pointX, pointY),
        .Text = text,
        .TextAlign = HorizontalAlignment.Left}

    Return textBox
End Function
```

Η συνάρτηση αυτή δέχεται ως ορίσματα τις συντεταγμένες (X, Y) ενός σημείου, το πλάτος και το ύψος ενός πλαισίου εισαγωγής κειμένου (TextBox) και το κείμενο που πρέπει να εμφανίζεται μέσα σε αυτό. Η συνάρτηση δημιουργεί ένα νέο αντικείμενο *TextBox* και το αρχικοποιεί με τις τιμές που παρέχονται ως ορίσματα. Τέλος, επιστρέφει αυτό το αντικείμενο *TextBox*. Η ρύθμιση της ιδιότητας *TextAlign to HorizontalAlignment.Left* καθορίζει τον τρόπο μορφοποίησης του κειμένου μέσα στο πλαίσιο κειμένου.

- **InitializeLabel()**

```
6 references
Private Function InitializeLabel(pointX As Double, pointY As Double, width As Double, height As Double, text As String) As Label
    Dim label As New Label() With {.Size = New Size(width, height), .Location = New Point(pointX, pointY), .Text = text}
    Return label
End Function
```

Η συνάρτηση αυτή δέχεται ως ορίσματα τις συντεταγμένες (X, Y) ενός σημείου, το πλάτος και το ύψος της επιγραφής (Label) και το κείμενο που πρέπει να εμφανίζει. Η συνάρτηση δημιουργεί μια νέα επιγραφή και την αρχικοποιεί με τις τιμές που παρέχονται ως ορίσματα και στην συνέχεια την επιστρέφει.

- **CreateFFunctionUIRow()**

```
1 reference
Private Sub CreateFFunctionUIRow(pointX As Integer, pointY As Double, functionName As String, totalVariables As Integer)

    Dim shiftRight As Integer = 25

    Dim labelsY As Double = pointY + 3
    Dim textBoxesY As Double = pointY


    For i = 0 To totalVariables - 1

        If i = 0 Then
            Dim functionSymbolLabel As Label = InitializeLabel(pointX, labelsY, 25, 30, functionName + " =")
            pointX += shiftRight
            functionLabelList.Add(functionSymbolLabel)

            Dim textBox As TextBox = InitializeTextBox(pointX, textBoxesY, 34, 50, "0")
            pointX += textBox.Size.Width
            functionTextBoxList.Add(textBox)


            Dim label As Label = InitializeLabel(pointX, labelsY, 29, 30, "X1")
            pointX += label.Size.Width
            functionLabelList.Add(label)
        Else

            functionLabelList.Last.Text += " +"

            Dim textBox As TextBox = InitializeTextBox(pointX, textBoxesY, 34, 50, "0")
            pointX += textBox.Size.Width
            functionTextBoxList.Add(textBox)


            Dim label As Label = InitializeLabel(pointX, labelsY, 29, 30, "X" + (i + 1).ToString())
            pointX += label.Size.Width
            functionLabelList.Add(label)

        End If
    Next


End Sub
```

Η παραπάνω συνάρτηση δημιουργεί επιγραφές (Labels) και πλαίσια εισαγωγής κειμένου (TextBoxes) για να εμφανίσει τα στοιχεία που απαιτούνται για τη δημιουργία της συνάρτησης. Το πρώτο *Label* που δημιουργείται εμφανίζει το όνομα της συνάρτησης, ενώ το πρώτο *TextBox* είναι για τον πρώτο συντελεστή της μεταβλητής της συνάρτησης. Κάθε επόμενο *Label* είναι για την επιγραφή της αντίστοιχης μεταβλητής (x_1, x_2, x_3 κλπ.), ενώ κάθε επόμενο *TextBox* είναι για την τιμή του αντίστοιχου συντελεστή της κάθε μεταβλητής.

- **CreateContraintUIRow()**

```
1 reference
Private Sub CreateContraintUIRow(pointX As Double, pointY As Double, totalVariables As Integer, constraintNumber As String)
    Dim shiftRight As Integer = 30
    Dim labelsY As Double = pointY + 3
    Dim textBoxesY As Double = pointY

    For i = 0 To totalVariables - 1
        If i = 0 Then

            If constraintNumber = 10 Then
                pointX -= 5
                shiftRight += 5
            End If

            Dim functionSymbolLabel As Label = InitializeLabel(pointX, labelsY, 40, 30, "G" + constraintNumber + " =") '25
            pointX += shiftRight
            constraintsLabelList.Add(functionSymbolLabel)

            Dim textBox As TextBox = InitializeTextBox(pointX, textBoxesY, 34, 50, "0")
            pointX += textBox.Size.Width
            constraintsTextBoxList.Add(textBox)

            Dim label As Label = InitializeLabel(pointX, labelsY, 29, 30, "X1")
            pointX += label.Size.Width
            constraintsLabelList.Add(label)
        Else

            constraintsLabelList.Last.Text += " +"

            Dim textBox As TextBox = InitializeTextBox(pointX, textBoxesY, 34, 50, "0")
            pointX += textBox.Size.Width
            constraintsTextBoxList.Add(textBox)

            Dim label As Label = InitializeLabel(pointX, labelsY, 29, 30, "X" + (i + 1).ToString())
            pointX += label.Size.Width
            constraintsLabelList.Add(label)

        End If
    Next

    Dim equalityTypeComboBox As New ComboBox()
    equalityTypeComboBox.Items.AddRange({"<=", ">=", "="})
    equalityTypeComboBox.Location = New Point(pointX, pointY)
    equalityTypeComboBox.Size = New Size(40, 30)
    equalityTypeComboBox.SelectedIndex = 0
    equalityTypeComboBox.DropDownStyle = ComboBoxStyle.DropDownList
    pointX += equalityTypeComboBox.Size.Width + 15
    equalityTypeComboBoxList.Add(equalityTypeComboBox)

    Dim constraintResultTextBox As TextBox = InitializeTextBox(pointX, textBoxesY, 34, 50, "0")
    constraintsTextBoxList.Add(constraintResultTextBox)
End Sub
```

Η συνάρτηση *CreateContraintUIRow* χρησιμοποιείται για τη δημιουργία μιας σειράς για μια σχέση περιορισμού στο γραφικό περιβάλλον της φόρμας. Το *pointX* και το *pointY* είναι οι συντεταγμένες στις οποίες θα αρχίσει η δημιουργία της σειράς επάνω στην φόρμα. Η μεταβλητή *totalVariables* καθορίζει πόσες μεταβλητές περιλαμβάνονται στη σχέση, και *constraintNumber* είναι ο αριθμός των περιορισμών της σχέσης. Η μεταβλητή *shiftRight* καθορίζει την απόσταση που θα μετακινηθεί δεξιά κάθε ετικέτα κειμένου και πλαίσιο κειμένου. Η σειρά αρχίζει με τη δημιουργία ετικέτας κειμένου για το σύμβολο της σχέσης (πχ. G1=), μια επιγραφή για την πρώτη μεταβλητή (x_1) και ένα πλαίσιο εισαγωγής κειμένου για την τιμή του συντελεστή της πρώτης μεταβλητής. Στη συνέχεια, για κάθε επιπλέον μεταβλητή δημιουργείται μια επιγραφή που περιέχει το σύμβολο '+' για να εμφανιστεί σωστά η σχέση, ένα πλαίσιο εισαγωγής κειμένου για την τιμή του συντελεστή κάθε

μεταβλητής και μια επιγραφή που δείχνει την αρίθμηση της μεταβλητής. Τέλος, η συνάρτηση αυτή δημιουργεί μια νέα λίστα επιλογής (ComboBox) για κάθε γραμμή περιορισμού. Το *ComboBox* αυτό έχει τους τρεις διαφορετικούς τύπους περιορισμών που μπορεί να έχει ένα πρόβλημα βελτιστοποίησης (≤, ≥, =). Επίσης, το *ComboBox* τοποθετείται σε μια συγκεκριμένη θέση στη φόρμα που ορίζεται από τις μεταβλητές *pointX* και *pointY*, ενώ έχει και μια συγκεκριμένη διάσταση. Τέλος, τα *ComboBox* προστίθενται στην λίστα *equalityTypeComboBoxList*, ενώ τα *TextBox* που δημιουργούνται στην συνάρτηση αυτή προστίθενται στην λίστα *constraintsTextBoxList*.

- **AppendDynamicUIComponents()**

```
Private Sub AppendDynamicUIComponents()

    For i = 0 To functionTextBoxList.Count - 1
        Controls.Add(functionTextBoxList(i))
    Next

    For i = 0 To functionLabelList.Count() - 1
        Controls.Add(functionLabelList(i))
    Next

    For i = 0 To constraintsTextBoxList.Count() - 1
        constraintsPanel.Controls.Add(constraintsTextBoxList(i))
    Next

    For i = 0 To constraintsLabelList.Count() - 1
        constraintsPanel.Controls.Add(constraintsLabelList(i))
    Next

    For i = 0 To equalityTypeComboBoxList.Count() - 1
        constraintsPanel.Controls.Add(equalityTypeComboBoxList(i))
    Next

End Sub
```

Η συνάρτηση AppendDynamicUIComponents προσθέτει τα δυναμικά στοιχεία του χρήστη στο κύριο παράθυρο της εφαρμογής. Προσθέτει τα πλαίσια εισαγωγής κειμένου και τις επιγραφές της συνάρτησης που δημιουργήθηκαν μέσω της συνάρτησης *CreateFFunctionUIRow*, και τα πλαίσια εισαγωγής κειμένου και τις επιγραφές της συνάρτησης που δημιουργήθηκαν μέσω της συνάρτησης *CreateContraintUIRow*, καθώς και τα *ComboBox* επιλογής του τύπου της ισότητας που δημιουργήθηκαν μέσω της ίδιας μεθόδου. Τέλος, τα παραπάνω στοιχεία προστίθενται είτε στο κύριο παράθυρο είτε στο παράθυρο περιορισμών, αντίστοιχα.

- **CreateDynamicUIComponents()**

```
2 references
Private Sub CreateDynamicUIComponents(totalVariables As Integer, totalConstraints As Integer)

    minMaxComboBox.SelectedIndex = 0

    Dim pointX = minMaxComboBox.Location.X + minMaxComboBox.Size.Width + 10
    Dim pointY = minMaxComboBox.Location.Y

    ' Creation of graphics for objective function '
    CreateFFunctionUIRow(pointX, pointY, "F", totalVariables)

    pointY = 8
    pointX = constraintsPanel.Location.X - 25

    'Creation of graphics for constrains '
    For i = 1 To totalConstraints
        CreateContraintUIRow(pointX, pointY, totalVariables, i)
        pointY += 30
    Next

    AppendDynamicUIComponents()

End Sub
```

Αυτή η ρουτίνα δημιουργεί τα δυναμικά στοιχεία της διεπαφής χρήστη. Λαμβάνει ως είσοδο τον αριθμό των μεταβλητών και περιορισμών που υπάρχουν στο πρόβλημα και δημιουργεί δυναμικά τα αντίστοιχα πεδία εισόδου για τις μεταβλητές και τους περιορισμούς. Καλεί τις συναρτήσεις *CreateFFunctionUIRow* και *CreateContraintUIRow* για τη δημιουργία των στοιχείων για την αντικειμενική συνάρτηση και τους περιορισμούς αντίστοιχα. Κατόπιν, καλεί τη συνάρτηση *AppendDynamicUIComponents* για την προσθήκη των δυναμικών στοιχείων διεπαφής χρήστη στην φόρμα.

- **ClearDynamicUIComponents()**

```
2 references
Private Sub ClearDynamicUIComponents()
    For Each element As TextBox In functionTextBoxList
        Controls.Remove(element)
    Next

    For Each element As Label In functionLabelList
        Controls.Remove(element)
    Next


    For Each element As Label In constraintsLabelList
        constraintsPanel.Controls.Remove(element)
    Next

    For Each element As TextBox In constraintsTextBoxList
        constraintsPanel.Controls.Remove(element)
    Next

    For Each element As ComboBox In equalityTypeComboBoxList
        constraintsPanel.Controls.Remove(element)
    Next

    functionTextBoxList.Clear()
    functionLabelList.Clear()

    constraintsLabelList.Clear()
    constraintsTextBoxList.Clear()
    equalityTypeComboBoxList.Clear()
End Sub
```

Αυτή η ρουτίνα καθαρίζει τα δυναμικά στοιχεία που δημιουργήθηκαν στο πρόγραμμα. Αυτό γίνεται με την αφαίρεση των στοιχείων *TextBox, Label* και *ComboBox* από τη φόρμα ή το πλαίσιο εμφάνισης μηνυμάτων, και στη συνέχεια καθαρίζει τις λίστες που περιέχουν τα στοιχεία αυτά.

- **CheckInitialVariablesTextBox()**

```
2 references
Private Sub CheckInitialVariablesTextBox(ByRef totalVarsInput As String)


    Dim inputAsArray As Char() = totalVarsInput.ToCharArray()

    For i = 0 To inputAsArray.Count() - 1

        Dim element = inputAsArray(i)

        If Not validInitialTextBoxCharacters.Contains(element) Then
            totalVarsInput = totalVarsInput.Remove(i)
        End If

    Next

    If String.IsNullOrEmpty(totalVarsInput) Or String.IsNullOrWhiteSpace(totalVarsInput) Or totalVarsInput.ToString.Equals("0") Then

        totalVarsInput = "2"

    End If


    Dim totalVarsInputNumber As Double = Double.Parse(totalVarsInput)

    If totalVarsInputNumber >= 11 Then totalVarsInput = "2"

End Sub
```

Η συνάρτηση *CheckInitialVariablesTextBox* ελέγχει το κείμενο εισαγωγής από τον χρήστη για τον αριθμό των μεταβλητών που θα χρησιμοποιηθούν στη συνάρτηση. Ελέγχει αν το κείμενο περιλαμβάνει έγκυρους χαρακτήρες, δηλαδή αριθμούς και κόμμα, και αν όχι, τους αφαιρεί. Στη συνέχεια, ελέγχει εάν το αποτέλεσμα είναι κενό ή ίσο με 0 και στη συνέχεια το θέτει σε 2. Τέλος, μετατρέπει το αποτέλεσμα σε αριθμό και ελέγχει αν είναι μεγαλύτερο ή ίσο με 11, σε περίπτωση που είναι, το θέτει σε 2.

- **TotalVariables_TextChanged**

```
0 references
Private Sub TotalVariables_TextChanged(sender As Object, e As EventArgs) Handles totalVarsNumberTextBox.TextChanged

    Dim totalVarsInput As String = sender.text

    CheckInitialVariablesTextBox(totalVarsInput)
    totalVarsNumberTextBox.Text = totalVarsInput

    clearDynamicUIComponents()

    Dim totalConstrains As Integer = 2

    If Not totalConstNumberTextBox.Text.Equals("") Then
        totalConstrains = Integer.Parse(totalConstNumberTextBox.Text.ToString)
    End If

    CreateDynamicUIComponents(totalVarsInput, totalConstrains)

End Sub
```

Η συνάρτηση *TotalVariables_TextChanged* εκτελείται κάθε φορά που αλλάζει το κείμενο του πλαισίου εισαγωγής κειμένου *totalVarsNumberTextBox*. Αρχικά, αποθηκεύει την τιμή του *totalVarsNumberTextBox* στην μεταβλητή *totalVarsInput*. Στη συνέχεια, ελέγχει τους χαρακτήρες της μεταβλητής *totalVarsInput* με τη χρήση της συνάρτησης *CheckInitialVariablesTextBox*, η οποία αφαιρεί κάθε χαρακτήρα που δεν είναι έγκυρος στο πλαίσιο εισαγωγής κειμένου. Επίσης, ελέγχει αν η τιμή της μεταβλητής *totalVarsInput* είναι κενή ή ίση με 0 και την θέτει σε 2 στην περίπτωση που ισχύει. Τέλος, μετατρέπει την τιμή της *totalVarsInput* σε αριθμό και την αποθηκεύει στη μεταβλητή *totalVarsInputNumber*. Εάν η *totalVarsInputNumber* είναι μεγαλύτερη ή ίση με 11, η τιμή της *totalVarsInput* τίθεται πάλι σε 2. Στη συνέχεια, η συνάρτηση καθαρίζει τα δυναμικά στοιχεία που έχουν δημιουργηθεί και καλεί τη συνάρτηση *CreateDynamicUIComponents* για να δημιουργήσει εκ νέου τα δυναμικά στοιχεία με βάση τις νέες τιμές των πλαισίων εισαγωγής κειμένου *totalVarsInput* και *totalConstrains*.

- **totalConstNumberTextBox_TextChanged()**

```
0 references
Private Sub totalConstNumberTextBox_TextChanged(sender As Object, e As EventArgs) Handles totalConstNumberTextBox.TextChanged

    Dim totalConstraintsInput As String = sender.text

    CheckInitialVariablesTextBox(totalConstraintsInput)
    totalConstNumberTextBox.Text = totalConstraintsInput

    clearDynamicUIComponents()

    Dim totalVariables As Integer = Integer.Parse(totalVarsNumberTextBox.Text.ToString)
    Dim totalConstrains As Integer = Integer.Parse(totalConstNumberTextBox.Text.ToString)

    CreateDynamicUIComponents(totalVariables, totalConstrains)

End Sub
```

Η παραπάνω συνάρτηση κάνει ότι και η *TotalVariables_TextChanged* απλά εκτελείται κάθε φορά που αλλάζει το κείμενο του πλαισίου εισαγωγής κειμένου *totalConstNumberTextBox*.

- **ParseInputTextBoxContent()**

```
4 references
Private Function ParseInputTextBoxContent(totalVariables As String) As Double

    If totalVariables.Contains(",") Then
        Dim replaceComa As String = totalVariables.Replace(",", ".")
        Return Double.Parse(replaceComa, CultureInfo.InvariantCulture)
    End If

    Return Double.Parse(totalVariables, CultureInfo.InvariantCulture)

End Function
```

Η συνάρτηση *ParseInputTextBoxContent* δέχεται ως όρισμα ένα αλφαριθμητικό (String) που αναπαριστά έναν αριθμό και επιστρέφει τον αριθμό σε μορφή δεκαδικού αριθμού (Double), ανεξαρτήτως αν το αλφαριθμητικό χρησιμοποιεί τον χαρακτήρα ',' ή '.' για τα δεκαδικά. Συγκεκριμένα, η συνάρτηση ελέγχει αν το αρχικό αλφαριθμητικό περιέχει τον χαρακτήρα ',' και αν ναι, τον αντικαθιστά με τον χαρακτήρα '.'. Στη συνέχεια, επιστρέφει το αποτέλεσμα της μετατροπής του τελικού αλφαριθμητικού σε δεκαδικό αριθμό χρησιμοποιώντας τη μέθοδο *Double.Parse(),* με δεδομένη την κατάλληλη παράμετρο *CultureInfo.InvariantCulture*. Αν το αρχικό αλφαριθμητικό δεν περιέχει τον χαρακτήρα ',', η συνάρτηση απλά επιστρέφει το αποτέλεσμα της μετατροπής του σε δεκαδικό αριθμό.

- **GetEqualityNumber()**

```
1 reference
Private Function GetEqualityNumber(input As String) As Integer
    If input.Equals("=") Then
        Return 1
    ElseIf input.Equals("<=") Then
        Return 2
    Else
        Return 3
    End If
End Function
```

Η συνάρτηση *GetEqualityNumber* δέχεται ένα αλφαριθμητικό ως είσοδο και επιστρέφει έναν ακέραιο αριθμό που αντιστοιχεί στον τελεστή της ισότητας ενός περιορισμού. Ειδικότερα, επιστρέφει την τιμή 1 αν το εισαγόμενο δεδομένο είναι "=", την τιμή 2 αν είναι "<=" και την τιμή 3 αν είναι ">=".

- **CreateFunctionArrayFromCSV()**

```
1 reference
Private Function CreateFunctionArrayFromCSV(totalVariablesNumber As Integer, inputs() As String,
                                            ByRef errorFlag As Boolean) As Double()

    Dim functionArray(totalVariablesNumber) As Double

    For i = 0 To inputs.Count() - 1

        Dim currentInput As String = inputs(i)

        If String.IsNullOrEmpty(currentInput) Or String.IsNullOrWhiteSpace(currentInput) Then
            displayPannel.Items.Add("Emtpy Function input is not valid.")
            errorFlag = True
            Exit Function
        End If

        For Each element As Char In currentInput.ToCharArray()
            If Not validTextBoxInputs.Contains(element) Then
                displayPannel.Items.Add("Invalid input in F function.")
                errorFlag = True
                Exit Function
            End If
        Next

        functionArray(i) = ParseInputTextBoxContent(currentInput)
    Next

    Return functionArray

End Function
```

Η συνάρτηση *CreateFunctionArrayFromCSV* δημιουργεί έναν πίνακα όπου τα δεδομένα εισόδου του χρήστη προέρχονται από αρχείο *csv*. Η συνάρτηση δέχεται ως είσοδο τον αριθμό των μεταβλητών (totalVariablesNumber), έναν πίνακα με τα δεδομένα εισόδου (inputs) και μια

μεταβλητή σφάλματος (errorFlag) που ορίζεται ως *Boolean* και χρησιμοποιείται για την εντοπισμό σφαλμάτων. Η συνάρτηση ελέγχει αν τα δεδομένα εισόδου είναι έγκυρα και εάν όλες οι τιμές των στοιχείων είναι αριθμητικές. Εάν υπάρχει οποιοδήποτε σφάλμα στην είσοδο, η σημαία σφάλματος θα οριστεί σε αληθής (True) και η συνάρτηση θα επιστρέψει κενό αποτέλεσμα, διαφορετικά η συνάρτηση επιστρέφει έναν πίνακα που περιέχει όλες τις τιμές των συντελεστών των μεταβλητών της αντικειμενικής συνάρτησης.

- **CreateFunctionArray()**

```
1 reference
Private Function CreateFunctionArray(totalVariablesNumber As Integer, ByRef errorFlag As Boolean) As Double()

    ' F FUNCTION ARRAY '
    Dim functionArray(totalVariablesNumber) As Double

    For i = 0 To functionTextBoxList.Count() - 1

        Dim currentTextBoxContents As String = functionTextBoxList(i).Text


        If String.IsNullOrEmpty(currentTextBoxContents) Or String.IsNullOrWhiteSpace(currentTextBoxContents) Then
            functionTextBoxList(i).Select()
            displayPannel.Items.Add("Emtpy Function input is not valid.")
            errorFlag = True
            Exit Function
        End If

        For Each element As Char In currentTextBoxContents

            If Not validTextBoxInputs.Contains(element) Then
                functionTextBoxList(i).Select()
                displayPannel.Items.Add("Invalid input in F function.")
                errorFlag = True
                Exit Function
            End If
        Next

        functionArray(i) = ParseInputTextBoxContent(functionTextBoxList(i).Text)
    Next

    Return functionArray

End Function
```

Η συνάρτηση *CreateFunctionArray* κάνει ότι και η *CreateFunctionArrayFromCSV* με την μόνη διαφορά ότι ελέγχει και λαμβάνει τιμές από την λίστα *functionTextBoxList*.

- **CreateConstraintsArrayFromCSV()**

```
1 reference
Private Function CreateConstraintsArrayFromCSV(totalVariablesNumber As Integer,
totalConstraintsNumber As Integer, ByRef errorFlag As Boolean, inputs As String()) As Double(,)

    ' Initialize constraints array '
    Dim constraintsArray(totalConstraintsNumber - 1, totalVariablesNumber + 1) As Double

    Dim constraintsListAux As New List(Of String)

    For i = 0 To inputs.Count() - 1

        Dim currentElement As String = inputs(i)

        If String.IsNullOrEmpty(currentElement) Or String.IsNullOrWhiteSpace(currentElement) Then
            displayPannel.Items.Add("Empty constraint input is not valid.")
            errorFlag = True
            Exit Function
        End If

        For Each element As Char In currentElement.ToCharArray()

            If Not validTextBoxInputs.Contains(element) Then
                displayPannel.Items.Add("Invalid input in constraint function.")
                errorFlag = True
                Exit Function
            End If
        Next

        constraintsListAux.Add(currentElement)
    Next

    Dim elementIndex As Integer = 0

    For i = 0 To totalConstraintsNumber - 1 '
        For j = 0 To totalVariablesNumber + 1
            Dim currentElement As String = constraintsListAux(elementIndex)
            constraintsArray(i, j) = ParseInputTextBoxContent(currentElement)
            elementIndex += 1
        Next
    Next

    Return constraintsArray

End Function
```

Αυτή η συνάρτηση δημιουργεί έναν πίνακα όπου τα δεδομένα εισόδου του χρήστη προέρχονται από αρχείο *csv*. Αρχικά δημιουργείται ένας πίνακας περιορισμών με διαστάσεις (totalConstraintsNumber, totalVariablesNumber + 1), όπου κάθε γραμμή αντιστοιχεί σε έναν περιορισμό και κάθε στήλη αντιστοιχεί σε μια μεταβλητή κάθε περιορισμού, ενώ η τελευταία στήλη αντιστοιχεί στο δεξιό μέλος του περιορισμού. Στη συνέχεια, γίνεται η επεξεργασία της σειράς δεδομένων που δίνονται ως είσοδος και αποθηκεύονται σε μια λίστα (constraintsListAux), ελέγχοντας για κενά και μη έγκυρες εισόδους. Έπειτα, τοποθετείται σε κάθε κελί του πίνακα περιορισμών η αντίστοιχη τιμή από τη λίστα με τα δεδομένα, με τη χρήση της συνάρτησης ParseInputTextBoxContent για τη μετατροπή του κειμένου σε αριθμό. Τέλος, η συνάρτηση επιστρέφει τον πίνακα περιορισμών, διαφορετικά επιστρέφει κενό και εμφανίζει το αντίστοιχο μήνυμα σφάλματος στον πίνακα εμφάνισης μηνυμάτων.

- **CreateConstraintsArray ()**

```
1 reference
Private Function CreateConstraintsArray(totalVariablesNumber As Integer, totalConstraintsNumber As Integer,
                                        ByRef errorFlag As Boolean) As Double(,)

    Dim elementIndex As Integer = 0

    Dim constraintsArray(totalConstraintsNumber - 1, totalVariablesNumber + 1) As Double

    For i = 0 To constraintsTextBoxList.Count() - 1

        Dim currentTextBoxContents As String = constraintsTextBoxList(i).Text

        If String.IsNullOrEmpty(currentTextBoxContents) Or String.IsNullOrWhiteSpace(currentTextBoxContents) Then
            constraintsTextBoxList(i).Select()
            displayPannel.Items.Add("Empty constraint input is not valid.")
            errorFlag = True
            Exit Function
        End If

        For Each element As Char In currentTextBoxContents

            If Not validTextBoxInputs.Contains(element) Then
                constraintsTextBoxList(i).Select()
                displayPannel.Items.Add("Invalid input in constraint function.")
                errorFlag = True
                Exit Function
            End If

        Next
    Next

    For i = 0 To totalConstraintsNumber - 1  '
        For j = 0 To totalVariablesNumber
            constraintsArray(i, j) = ParseInputTextBoxContent(constraintsTextBoxList(elementIndex).Text)
            elementIndex += 1
        Next
    Next

    For i = 0 To equalityTypeComboBoxList.Count() - 1
        constraintsArray(i, totalVariablesNumber + 1) = GetEqualityNumber(equalityTypeComboBoxList(i).Text)
    Next

    Return constraintsArray

End Function
```

Η συνάρτηση *CreateConstraintsArray* κάνει ότι και η *CreateConstraintsArrayFromCSV* με την μόνη διαφορά ότι ελέγχει και λαμβάνει τιμές από την λίστα *constraintsTextBoxList*.

- **EnableDisableUIComponents()**

```
Private Sub EnableDisableUIComponents(isEnable As Boolean)

    For i = 0 To functionTextBoxList.Count - 1
        functionTextBoxList(i).Enabled = isEnable
    Next

    For i = 0 To constraintsTextBoxList.Count() - 1
        constraintsTextBoxList(i).Enabled = isEnable
    Next

    For i = 0 To equalityTypeComboBoxList.Count() - 1
        equalityTypeComboBoxList(i).Enabled = isEnable
    Next

    totalConstNumberTextBox.Enabled = isEnable
    totalVarsNumberTextBox.Enabled = isEnable
End Sub
```

Η συνάρτηση *EnableDisableUIComponents* δέχεται σαν παράμετρο μια μεταβλητή τύπου Boolean (isEnable) που δηλώνει αν τα οπτικά αντικείμενα που περιέχονται στην φόρμα πρέπει να είναι ενεργοποιημένα ή απενεργοποιημένα. Η συνάρτηση προσπαθεί να ενεργοποιήσει ή να απενεργοποιήσει αυτά τα στοιχεία ανάλογα με την τιμή της μεταβλητής (isEnable) που δόθηκε. Εάν η τιμή της μεταβλητής (isEnable) είναι αληθής, τα στοιχεία ενεργοποιούνται , ενώ αν η τιμή της είναι ψευδής, τότε τα στοιχεία απενεργοποιούνται.

- **CsvButton_Click()**

```
Private Sub csvButton_Click(sender As Object, e As EventArgs) Handles csvButton.Click

    Dim csvFileReader As StreamReader = Nothing
    displayPannel.Items.Clear()

    Try
        Dim openFileDialog As New OpenFileDialog()
        openFileDialog.Title = "Open CSV File"
        openFileDialog.Filter = "CSV Files (*.csv)|*.csv"

        If openFileDialog.ShowDialog = DialogResult.OK Then

            Dim errorFlag As Boolean = False
            Dim csvFile As String = openFileDialog.FileName

            csvFileReader = My.Computer.FileSystem.OpenTextFileReader(csvFile)

            totalVariablesNumber = 0
            totalConstraintsNumber = 0

            Dim functionLine As String = csvFileReader.ReadLine()
            Dim functionLineArray As String() = functionLine.Split(";")

            Dim functionLineList As New List(Of String)

            For i = 0 To functionLineArray.Count() - 1
                Dim element As String = functionLineArray(i).Trim()
                If (Not String.IsNullOrEmpty(element)) AndAlso (Not String.IsNullOrWhiteSpace(element)) Then
                    functionLineList.Add(element)
                End If
            Next

            totalVariablesNumber = functionLineList.Count()
            functionArray = CreateFunctionArrayFromCSV(totalVariablesNumber - 1, functionLineList.ToArray(), errorFlag)

            If errorFlag = True Then Exit Sub

            Dim constraintsInputs As New List(Of String)

            Do
                Dim currentLine As String = csvFileReader.ReadLine
                Dim currentLineArray As String() = currentLine.Split(";")

                For i = 0 To currentLineArray.Length - 1

                    Dim element As String = currentLineArray(i).Trim()

                    If (Not String.IsNullOrEmpty(element)) AndAlso (Not String.IsNullOrWhiteSpace(element)) Then
                        constraintsInputs.Add(element)
                    End If
                Next

                totalConstraintsNumber = totalConstraintsNumber + 1

            Loop Until csvFileReader.EndOfStream

            constraintsArray = CreateConstraintsArrayFromCSV(totalVariablesNumber, totalConstraintsNumber, errorFlag, constraintsInputs.ToArray())

            If errorFlag = True Then Exit Sub

            csvFileIsLoaded = True
            EnableDisableUIComponents(False)

            displayPannel.Items.Add("The .csv file is successfully loaded. Select if you want to solve it as maximization or minimization problem.")
            displayPannel.Items.Add("Please press the Calculate button to start the calculation.")

            calculateButton.Select()
        End If

    Catch ex As Exception
        ' Emfanish mhnumatos sxetika me sfalma sto anoigma tou arxeiou '
        displayPannel.Items.Add("An error occurred while loading the .csv file.") '~> Emfanish munhmatos sto panel
        displayPannel.Items.Add("Please check the File manual (*.csv) to verify the correctness of your .csv file.")
        displayPannel.Items.Add("If the file (*.csv) is opened in another program please close it and try again.")
    Finally
        If csvFileReader IsNot Nothing Then
            csvFileReader.Close()
            csvFileReader.Dispose()
        End If
    End Try
End Sub
```

Η παραπάνω συνάρτηση είναι μια υποδοχή γεγονότος κλικ (Event Handler) για το κουμπί με την ονομασία *csvButton*. Ο βασικός στόχος της συνάρτησης είναι να ανοίγει ένα αρχείο *csv*, να επεξεργάζεται τα δεδομένα που περιέχει και να αντλεί από αυτά πληροφορίες που θα χρησιμοποιηθούν αργότερα για τον υπολογισμό της μεθόδου Simplex. Ο κώδικας της συνάρτησης περιλαμβάνει τις εξής ενέργειες:

1) Αρχικοποίηση της μεταβλητής *csvFileReader* σε κενό (null).
2) Εκκαθάριση του πλαισίου εμφάνισης μηνυμάτων (displayPanel).
3) Άνοιγμα της εξερεύνησης εργασιών του Η/Υ (OpenFileDialog) με σκοπό ο χρήστης να μπορεί να επιλέξει ένα αρχείο *csv* προς φόρτωση.
4) Αν το αρχείο επιλέγεται επιτυχώς, δημιουργείται μια μεταβλητή *errorFlag* που χρησιμοποιείται για την διαδικασία ελέγχου εγκυρότητας των δεδομένων εισαγωγής.
5) Αρχικοποίηση ενός αντικείμενο τύπου *StreamReader* με το οποίο θα γίνει η ανάγνωση των δεδομένων του αρχείου *csv*.
6) Αρχικοποίηση των μεταβλητών *totalVariablesNumber* και t*otalConstraintsNumber* σε μηδέν.
7) Ανάγνωση των δεδομένων του αρχείου γραμμή προς γραμμή όπου τα δεδομένα κάθε γραμμής αποθηκεύονται στην λίστα *constraintsInputs* και *functionLineList* αντίστοιχα.

- **CalculateButton_Click()**

```
0 references
Private Sub CalculateButton_Click(sender As Object, e As EventArgs) Handles calculateButton.Click

    Dim max As Integer = 0
    Dim simplexMessage As String = ""

    Try
        displayPannel.Items.Clear()

        If csvFileIsLoaded = True Then

            csvFileIsLoaded = False

            If minMaxComboBox.Text.Equals("Max") Then
                For i = 0 To functionArray.Count - 1
                    functionArray(i) = functionArray(i) * -1
                Next
                max = 1
            End If

            Dim simplexCalculation As New SimplexCalculation(totalVariablesNumber, totalConstraintsNumber, constraintsArray, functionArray)
            simplexCalculation.CalculateSimplex(simplexMessage, max)

            EnableDisableUIComponents(True)
        Else
```

```
            totalVariablesNumber = Integer.Parse(totalVarsNumberTextBox.Text)
            totalConstraintsNumber = Integer.Parse(totalConstNumberTextBox.Text)

            Dim errorFlag As Boolean = False
            functionArray = CreateFunctionArray(totalVariablesNumber, errorFlag)

            If errorFlag = True Then Return

            constraintsArray = CreateConstraintsArray(totalVariablesNumber, totalConstraintsNumber, errorFlag)

            If errorFlag = True Then Return

            If minMaxComboBox.Text.Equals("Max") Then
                For i = 0 To functionArray.Count - 1
                    functionArray(i) = functionArray(i) * -1
                Next
                max = 1
            End If

            Dim simplexCalculation As New SimplexCalculation(totalVariablesNumber, totalConstraintsNumber, constraintsArray, functionArray)
            simplexCalculation.CalculateSimplex(simplexMessage, max)
        End If


        Dim simplexMessageResult() As String = simplexMessage.Split(vbNewLine)

        For i = 0 To simplexMessageResult.Count() - 1
            displayPannel.Items.Add(simplexMessageResult(i))
        Next

    Catch ex As Exception
        EnableDisableUIComponents(True)
        displayPannel.Items.Add("An error occurred during the calculation.")
        displayPannel.Items.Add("Please check the File manual (*.csv) to verify the correctness of your .csv file.")
    End Try
End Sub
```

Αυτή η συνάρτηση εκτελείται όταν ο χρήστης πατά το κουμπί *Calculate* στο γραφικό περιβάλλον της εφαρμογής. Ανάλογα με την κατάσταση της εφαρμογής (αν έχει φορτωθεί ένα αρχείο *csv* ή όχι), η συνάρτηση επιλέγει ποιο μέρος του κώδικα πρέπει να εκτελέσει. Εάν έχει φορτωθεί ένα αρχείο *csv*, τότε χρησιμοποιούνται τα δεδομένα που έχουν διαβαστεί από το αρχείο για τον υπολογισμό την μεθόδου Simplex. Αν δεν έχει φορτωθεί αρχείο *csv*, τότε για τον υπολογισμό χρησιμοποιούνται τα δεδομένα που έχει εισάγει ο χρήστης στα πλαίσια κειμένου (TextBox) της φόρμας. Τέλος, η συνάρτηση εμφανίζει το αποτέλεσμα της διαδικασίας βελτιστοποίησης του Simplex στο πλαίσιο εμφάνισης μηνυμάτων (displayPanel) της φόρμας. Εάν προκύψει κάποιο σφάλμα κατά τη διαδικασία υπολογισμού, εμφανίζει το αντίστοιχο μήνυμα σφάλματος.

- **InitialTextBoxes_MouseClick()**

```
0 references
Private Sub InitialTextBoxes_MouseClick(sender As Object, e As MouseEventArgs) Handles _
        totalConstNumberTextBox.MouseClick, totalVarsNumberTextBox.MouseClick

    Dim clickedTextBox As TextBox = CType(sender, TextBox)
    clickedTextBox.Select(0, clickedTextBox.Text.Length)
End Sub
```

Η παραπάνω συνάρτηση εκτελείται όταν γίνεται κλικ σε ένα από δύο πεδία εισαγωγής κειμένου (TextBox) της φόρμας. Αυτά τα πλαίσια εισαγωγής κειμένου είναι το *totalConstNumberTextBox* και το *totalVarsNumberTextBox* και είναι συνδεδεμένα με αυτήν τη συνάρτηση μέσω του *Handles* στη δήλωση της συνάρτησης. Η συνάρτηση αυτή έχει ως στόχο να επιλέξει το περιεχόμενο του πλαισίου που έκανε κλικ ο χρήστης, ώστε αυτό να είναι έτοιμο για επεξεργασία.

- **csvManualButton_Click()**

```
0 references
Private Sub csvManualButton_Click(sender As Object, e As EventArgs) Handles csvManualButton.Click
    Dim csvManualFilePath As String = My.Application.Info.DirectoryPath + "\csvManual.pdf"
    Process.Start(csvManualFilePath)
End Sub
```

Η παραπάνω συνάρτηση εκτελείται όταν ο χρήστης πατάει το κουμπί με την ονομασία *csvManualButton*. Στην περίπτωση αυτή, η συνάρτηση αναθέτει στη μεταβλητή *csvManualFilePath* την απόλυτη διαδρομή του αρχείου *csvManual.pdf*, το οποίο βρίσκεται στον ίδιο φάκελο με την εφαρμογή (My.Application.Info.DirectoryPath είναι μια μεταβλητή που αναφέρεται στον φάκελο της εφαρμογής). Έπειτα, η συνάρτηση χρησιμοποιεί τη μέθοδο *Process.Start* για να ανοίξει το αρχείο c*svManualFilePath* με το πρόγραμμα που είναι προκαθορισμένο στα αρχεία τύπου *pdf* του υπολογιστή του χρήστη (π.χ. Adobe Reader). Με αυτόν τον τρόπο, το αρχείο τύπου *pdf* ανοίγει στον υπολογιστή του χρήστη για να διαβαστεί.

5. Συμπεράσματα και παραδείγματα

Στην παρούσα ενότητα αναλύονται τα συμπεράσματα που προέκυψαν από την εκπόνηση της παρούσας εργασίας, καθώς και προτάσεις για μελλοντική εξέλιξη του λογισμικού που υλοποιήθηκε. Τέλος, παρατίθενται παραδείγματα επίλυσης προβλημάτων Γραμμικού Προγραμματισμού με την βοήθεια του λογισμικού που αναπτύχθηκε.

5.1 Συμπεράσματα

Η δημιουργία ενός λογισμικού που λύνει προβλήματα βελτιστοποίησης με τη μέθοδο Simplex μπορεί να οδηγήσει σε αρκετά σημαντικά συμπεράσματα. Αρχικά, για την δημιουργία ενός τέτοιου λογισμικού απαιτείται η πλήρης κατανόηση των μαθηματικών αρχών της μεθόδου η οποία μπορεί να συντελέσει στην καλύτερη κατανόηση της βελτιστοποίησης και των μαθηματικών αλγορίθμων πίσω από αυτήν. Επιπλέον, δίνεται η δυνατότητα επίλυσης προβλημάτων Γραμμικού Προγραμματισμού με μεγάλη αποδοτικότητα και αποτελεσματικότητα κάνοντάς το ένα χρήσιμο εργαλείο τόσο για τους φοιτητές του Μεταπτυχιακού Προγράμματος, όσο και για τους υποψήφιους διδάκτορες του Τμήματος, το οποίο ήταν και ο κύριος σκοπός της εργασίας. Τέλος, με την βελτίωση αλλά και επέκταση των λειτουργιών που εκτελεί το λογισμικό, είναι δυνατό να μετατραπεί σε χρήσιμο εργαλείο με ποικίλες εφαρμογές, ακόμα και για επαγγελματίες μηχανικούς.

5.2 Προτάσεις για το μέλλον

Το λογισμικό καθώς αναπτύχθηκε στα πλαίσια μιας μεταπτυχιακής διπλωματικής εργασίας, είναι κατανοητό πως έχει αρκετά περιθώρια εξέλιξης και στο υπολογιστικό αλλά και στο λειτουργικό κομμάτι του. Παρακάτω παρατίθενται ορισμένες προτάσεις που θα οδηγούσαν στην βελτιστοποίησή του:

- Επίλυση προβλημάτων και με άλλες μορφές της μεθόδου Simplex, εκτός από αυτή των 2 φάσεων που χρησιμοποιείται, όπως η Αναθεωρημένη μέθοδος Simplex ή η Διττή μέθοδος Simplex.
- Επίλυση προβλημάτων Γραμμικού Προγραμματισμού με παραπάνω από μια αντικειμενικές συναρτήσεις.
- Υπολογισμός του χρόνου επίλυσης, ώστε να είναι δυνατή η σύγκριση των τιμών μεταξύ διαφορετικών μεθόδων.

5.3 Παραδείγματα

Για την καλύτερη κατανόηση της λειτουργίας του λογισμικού και ειδικότερα του περιβάλλοντος διεπαφής του χρήστη με αυτό, παρακάτω δίνονται παραδείγματα επίλυσης προβλημάτων Γραμμικού Προγραμματισμού με την βοήθεια του λογισμικού.

Παράδειγμα 1:

Στο παράδειγμα αυτό ζητείται να ελαχιστοποιηθεί η αντικειμενική συνάρτηση και τα δεδομένα εισάγονται στην εφαρμογή συμπληρώνοντας τα πλαίσια εισαγωγής κειμένου και επιλέγοντας από την λίστα τον τύπο της ισότητας για κάθε περιορισμό:

Αντικειμενική συνάρτηση:

min $F= x_1 + 2x_2$

Περιορισμοί δομής:

$2x_1 + x_2 \leq 6$

$x_1 + x_2 = 4$

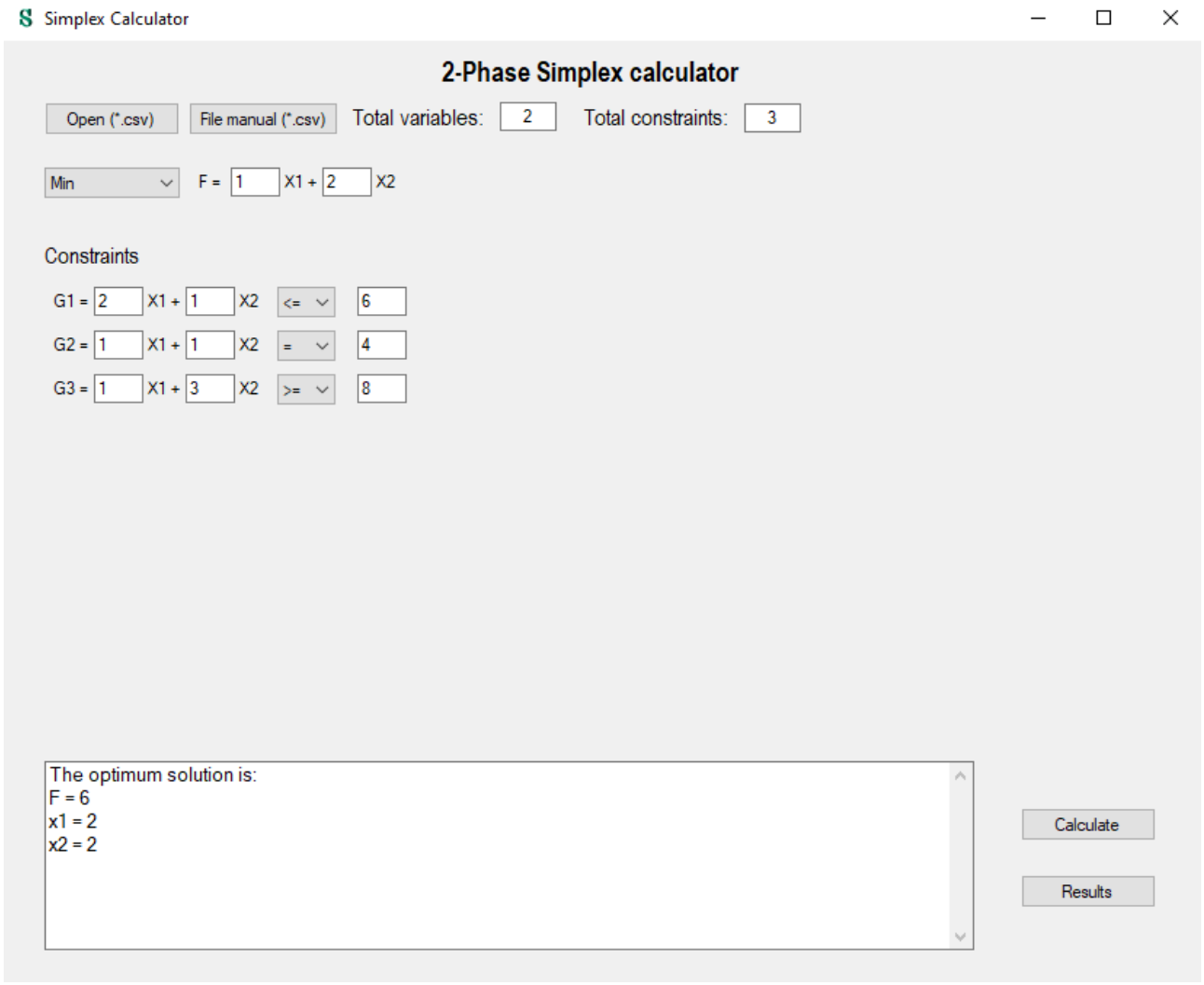
$x_1 + 3x_2 \geq 8$

Περιορισμοί μη αρνητικότητας:

$x_1, x_2 \geq 0$

Λύση:

Η ελάχιστη τιμή της αντικειμενικής συνάρτησης είναι *F=6* και η άριστη λύση είναι: $x_1=2, x_2=2.$



Παρακάτω απεικονίζονται οι ενδιάμεσοι πίνακες του παραδείγματος όπως προκύπτουν από το λογισμικό για κάθε επανάληψη του αλγορίθμου Simplex.

SimplexCalculatorResult.txt - Σημειωματάριο

Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

PHASE 1:

BV	x1	x2	x3	x4	x5	x6	b
x4	2,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	6,000E+000
x5	1,000E+000	1,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	4,000E+000
x6	1,000E+000	3,000E+000	-1,000E+000	0,000E+000	0,000E+000	1,000E+000	8,000E+000
-F	1,000E+000	2,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000
-w	-2,000E+000	-4,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	-1,200E+001

BV	x1	x2	x3	x4	x5	x6	b
x4	1,667E+000	0,000E+000	3,333E-001	1,000E+000	0,000E+000	-3,333E-001	3,333E+000
x5	6,667E-001	0,000E+000	3,333E-001	0,000E+000	1,000E+000	-3,333E-001	1,333E+000
x2	3,333E-001	1,000E+000	-3,333E-001	0,000E+000	0,000E+000	3,333E-001	2,667E+000
-F	3,333E-001	0,000E+000	6,667E-001	0,000E+000	0,000E+000	-6,667E-001	-5,333E+000
-w	-6,667E-001	0,000E+000	-3,333E-001	0,000E+000	0,000E+000	1,333E+000	-1,333E+000

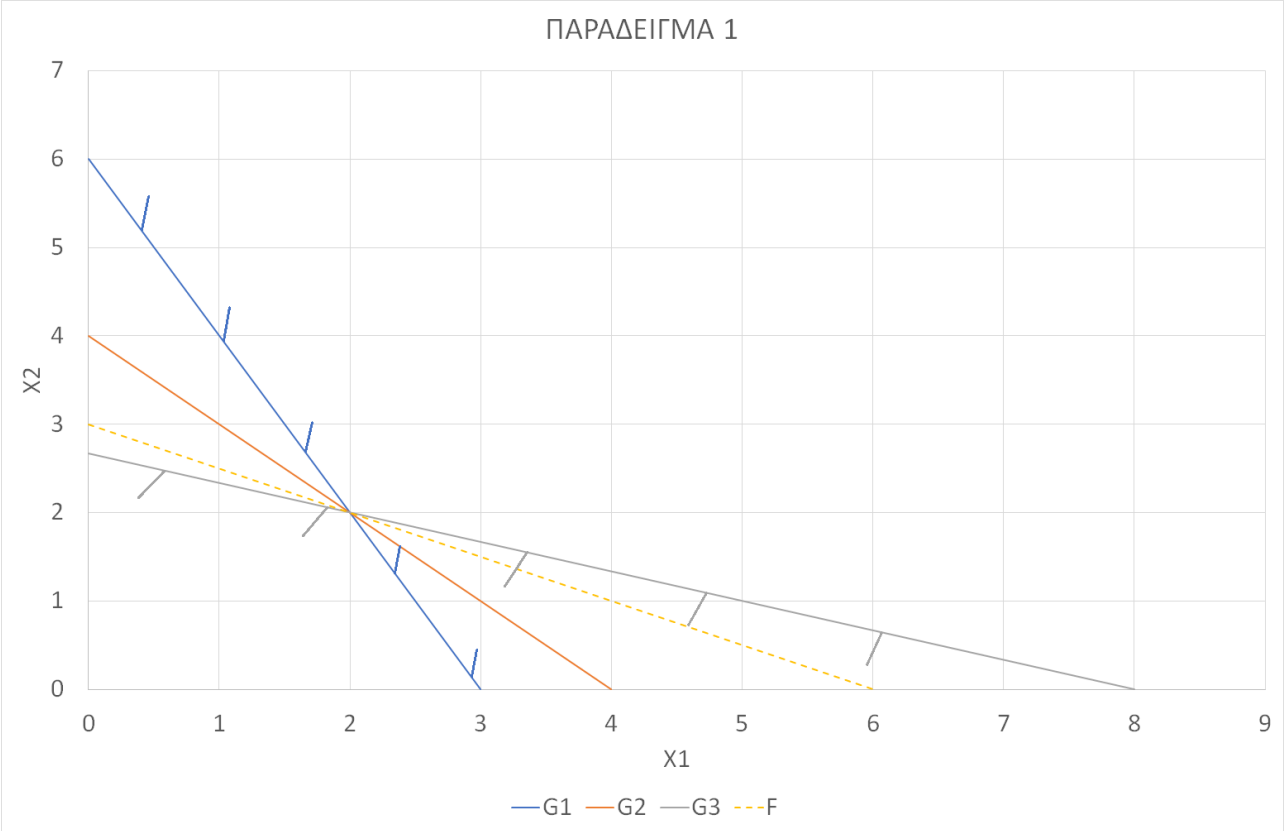
BV	x1	x2	x3	x4	x5	x6	b
x1	1,000E+000	0,000E+000	2,000E-001	6,000E-001	0,000E+000	-2,000E-001	2,000E+000
x5	0,000E+000	0,000E+000	2,000E-001	-4,000E-001	1,000E+000	-2,000E-001	0,000E+000
x2	0,000E+000	1,000E+000	-4,000E-001	-2,000E-001	0,000E+000	4,000E-001	2,000E+000
-F	0,000E+000	0,000E+000	6,000E-001	-2,000E-001	0,000E+000	-6,000E-001	-6,000E+000
-w	0,000E+000	0,000E+000	-2,000E-001	4,000E-001	0,000E+000	1,200E+000	-4,441E-016

BV	x1	x2	x3	x4	x5	x6	b
x1	1,000E+000	0,000E+000	0,000E+000	1,000E+000	-1,000E+000	0,000E+000	2,000E+000
x3	0,000E+000	0,000E+000	1,000E+000	-2,000E+000	5,000E+000	-1,000E+000	0,000E+000
x2	0,000E+000	1,000E+000	0,000E+000	-1,000E+000	2,000E+000	0,000E+000	2,000E+000
-F	0,000E+000	0,000E+000	0,000E+000	1,000E+000	-3,000E+000	0,000E+000	-6,000E+000
-w	0,000E+000	0,000E+000	0,000E+000	5,551E-017	1,000E+000	1,000E+000	-4,441E-016

PHASE 2:

BV	x1	x2	x3	x4	b
x1	1,000E+000	0,000E+000	0,000E+000	1,000E+000	2,000E+000
x3	0,000E+000	0,000E+000	1,000E+000	-2,000E+000	0,000E+000
x2	0,000E+000	1,000E+000	0,000E+000	-1,000E+000	2,000E+000
-F	0,000E+000	0,000E+000	0,000E+000	1,000E+000	-6,000E+000

Τέλος, στο παρακάτω διάγραμμα απεικονίζονται οι περιορισμοί δομής του παραδείγματος όπως και η αντικειμενική συνάρτηση *F* (με τιμή F=6), επιβεβαιώνοντας την ορθότητα της επίλυσης του λογισμικού.



Παράδειγμα 2:

Στο παράδειγμα αυτό ζητείται να μεγιστοποιηθεί η αντικειμενική συνάρτηση και να εισαχθούν τα δεδομένα στην εφαρμογή μέσω υπολογιστικού φύλλου.

Αντικειμενική συνάρτηση:

max $F= 3x_1 + 4x_2 + 2x_3$

Περιορισμοί δομής:

$x_1 + x_2 \leq 100$

$x_2 + x_3 \leq 200$

$x_1 + x_2 + x_3 \leq 400$

$x_2 \leq 80$

Περιορισμοί μη αρνητικότητας:

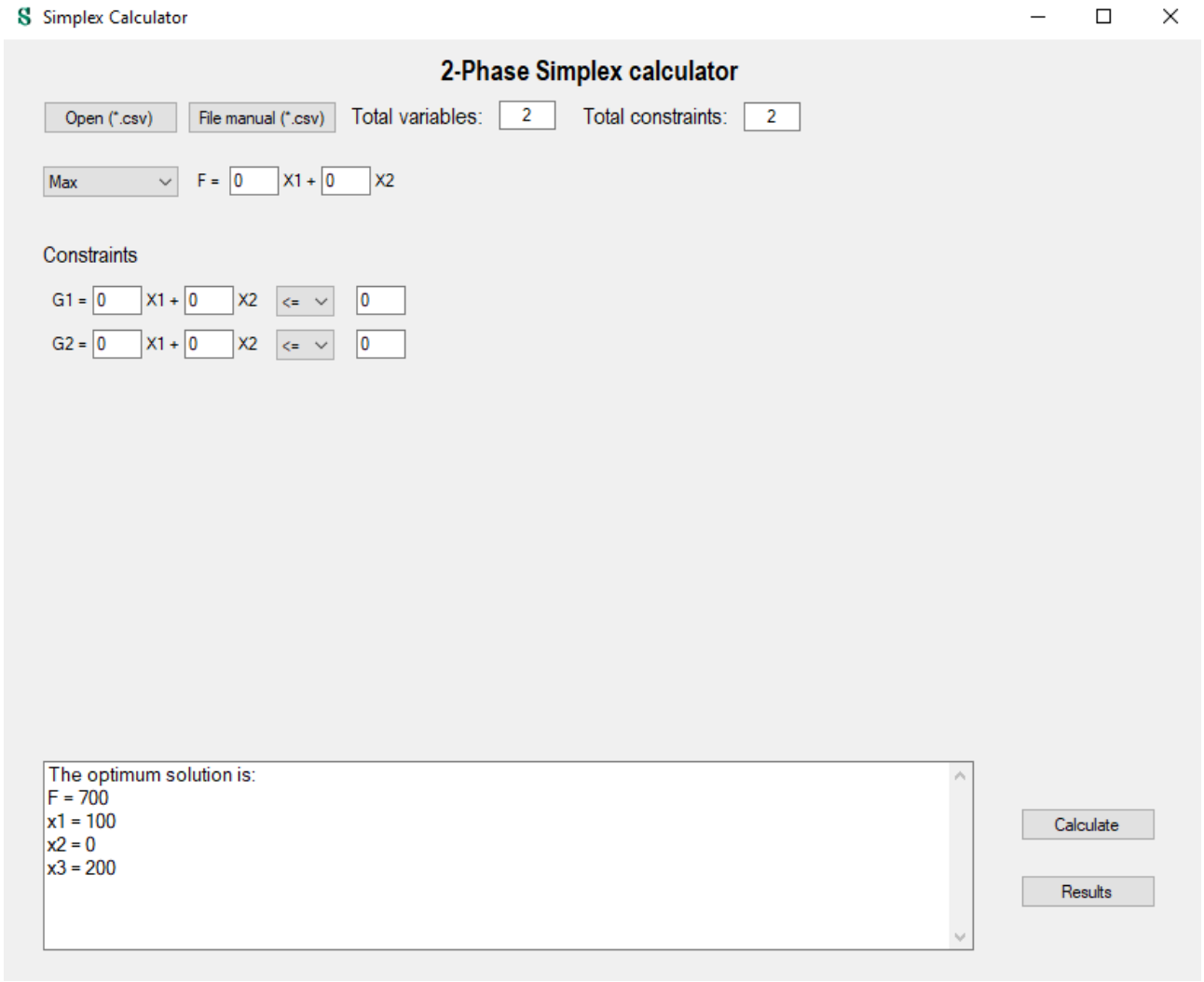
$x_1, x_2, x_3 \geq 0$

Λύση:

Η μέγιστη τιμή της αντικειμενικής συνάρτησης είναι *F=700* και η άριστη λύση είναι: $x_1=100$, $x_2=0$, $x_3=200$.

Η δομή του υπολογιστικού φύλλου που εισάγεται ώστε να επιλυθεί το παραπάνω πρόβλημα είναι η εξής:

	A	B	C	D	E	F
1	3	4	2			
2	1	1	0	100	2	
3	0	1	1	200	2	
4	1	1	1	400	2	
5	0	2	0	80	2	
6						



Παρακάτω απεικονίζονται οι ενδιάμεσοι πίνακες του παραδείγματος όπως προκύπτουν από το λογισμικό για κάθε επανάληψη του αλγορίθμου Simplex.

SimplexCalculatorResult.txt - Σημειωματάριο

Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

PHASE 1:

BV	x1	x2	x3	x4	x5	x6	x7	b
x4	1,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+002
x5	0,000E+000	1,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	2,000E+002
x6	1,000E+000	1,000E+000	1,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	4,000E+002
x7	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	8,000E+001
-F	-3,000E+000	-4,000E+000	-2,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000
-w	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000

PHASE 2:

BV	x1	x2	x3	x4	x5	x6	x7	b
x4	1,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+002
x5	0,000E+000	1,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	2,000E+002
x6	1,000E+000	1,000E+000	1,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	4,000E+002
x7	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	8,000E+001
-F	-3,000E+000	-4,000E+000	-2,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000

BV	x1	x2	x3	x4	x5	x6	x7	b
x4	1,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	-1,000E+000	2,000E+001
x5	0,000E+000	0,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	-1,000E+000	1,200E+002
x6	1,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	1,000E+000	-1,000E+000	3,200E+002
x2	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	8,000E+001
-F	-3,000E+000	0,000E+000	-2,000E+000	0,000E+000	0,000E+000	0,000E+000	4,000E+000	3,200E+002

BV	x1	x2	x3	x4	x5	x6	x7	b
x1	1,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	-1,000E+000	2,000E+001
x5	0,000E+000	0,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	-1,000E+000	1,200E+002
x6	0,000E+000	0,000E+000	1,000E+000	-1,000E+000	0,000E+000	1,000E+000	0,000E+000	3,000E+002
x2	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	8,000E+001
-F	0,000E+000	0,000E+000	-2,000E+000	3,000E+000	0,000E+000	0,000E+000	1,000E+000	3,800E+002

BV	x1	x2	x3	x4	x5	x6	x7	b
x1	1,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	-1,000E+000	2,000E+001
x3	0,000E+000	0,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	-1,000E+000	1,200E+002
x6	0,000E+000	0,000E+000	0,000E+000	-1,000E+000	-1,000E+000	1,000E+000	1,000E+000	1,800E+002
x2	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	8,000E+001
-F	0,000E+000	0,000E+000	0,000E+000	3,000E+000	2,000E+000	0,000E+000	-1,000E+000	6,200E+002

BV	x1	x2	x3	x4	x5	x6	x7	b
x1	1,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+002
x3	0,000E+000	1,000E+000	1,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	2,000E+002
x6	0,000E+000	-1,000E+000	0,000E+000	-1,000E+000	-1,000E+000	1,000E+000	0,000E+000	1,000E+002
x7	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	8,000E+001
-F	0,000E+000	1,000E+000	0,000E+000	3,000E+000	2,000E+000	0,000E+000	0,000E+000	7,000E+002

Παράδειγμα 3:

Στο παράδειγμα αυτό ζητείται να ελαχιστοποιηθεί η αντικειμενική συνάρτηση και τα δεδομένα εισάγονται στην εφαρμογή συμπληρώνοντας τα πλαίσια εισαγωγής κειμένου και επιλέγοντας από την λίστα τον τύπο της ισότητας για κάθε περιορισμό:

Αντικειμενική συνάρτηση:

min $F= -3x_1 - 2x_2$

Περιορισμοί δομής:

$x_1 - x_2 \leq 6$

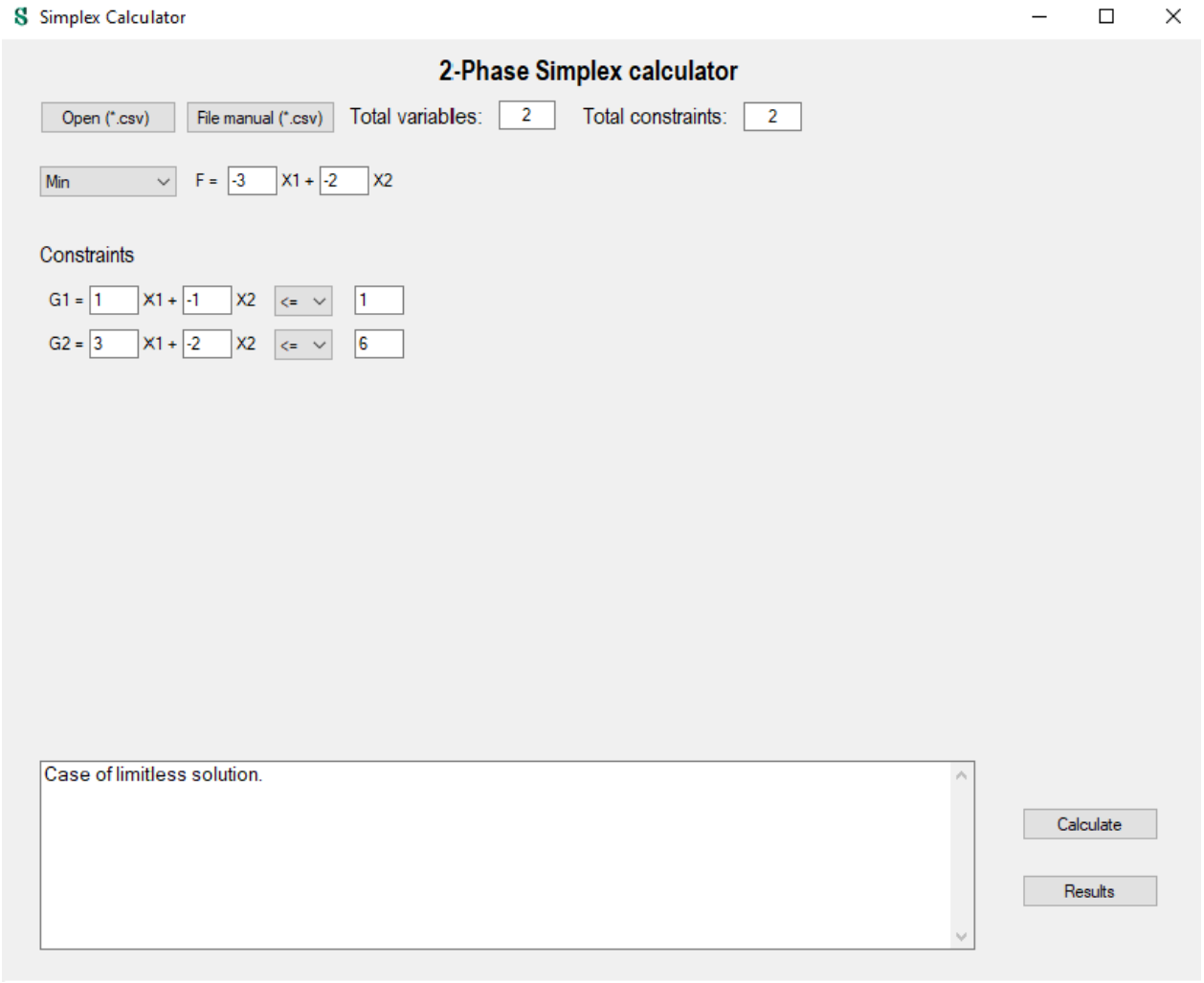
$3x_1 - 2x_2 \leq 4$

Περιορισμοί μη αρνητικότητας:

$x_1, x_2 \geq 0$

Λύση:

Περίπτωση μη περιορισμένης λύσης. Δηλαδή η ελάχιστη τιμή της αντικειμενικής συνάρτησης είναι το μείον άπειρο.



Παρακάτω απεικονίζονται οι ενδιάμεσοι πίνακες του παραδείγματος όπως προκύπτουν από το λογισμικό για κάθε επανάληψη του αλγορίθμου Simplex.

PHASE 1:

BV	x1	x2	x3	x4	b
x3	1,000E+000	-1,000E+000	1,000E+000	0,000E+000	1,000E+000
x4	3,000E+000	-2,000E+000	0,000E+000	1,000E+000	6,000E+000
-F	-3,000E+000	-2,000E+000	0,000E+000	0,000E+000	0,000E+000
-w	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000

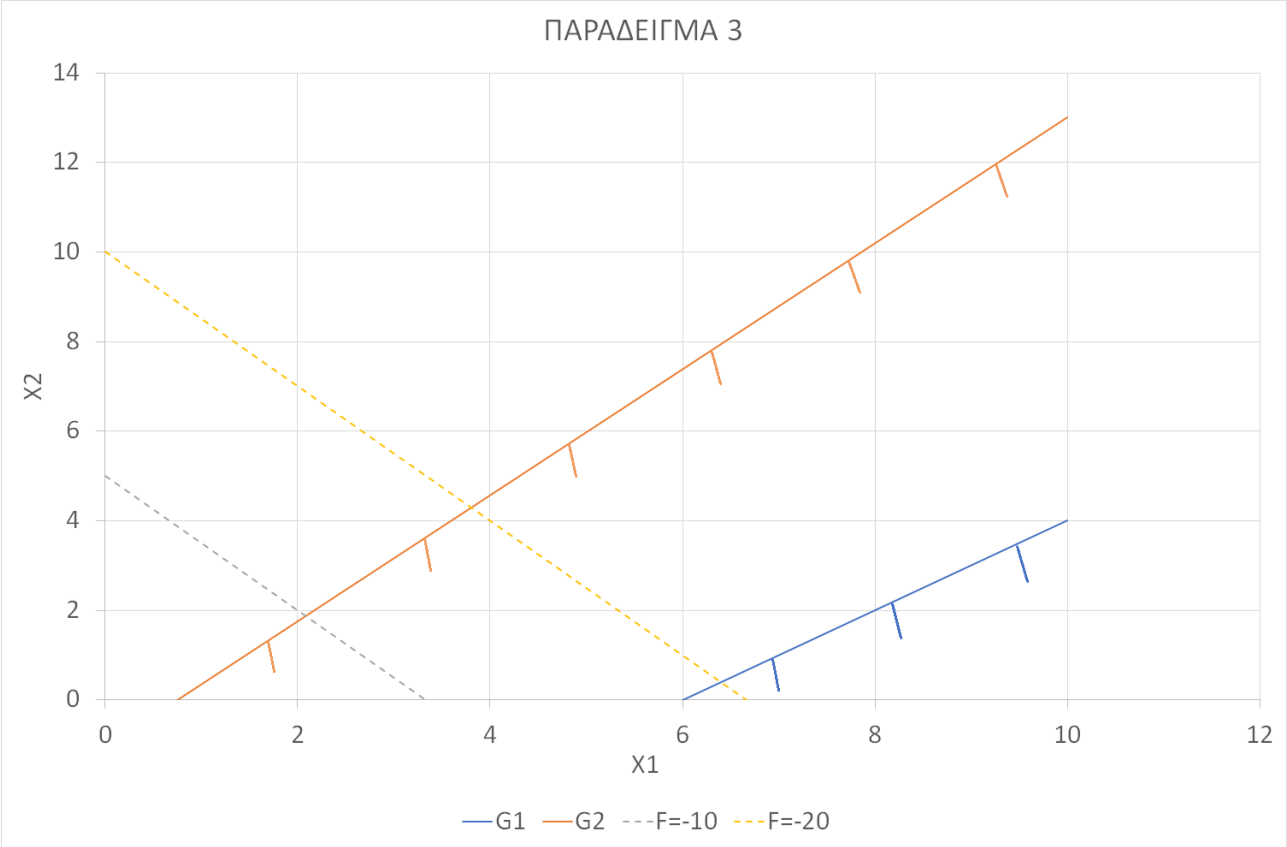
PHASE 2:

BV	x1	x2	x3	x4	b
x3	1,000E+000	-1,000E+000	1,000E+000	0,000E+000	1,000E+000
x4	3,000E+000	-2,000E+000	0,000E+000	1,000E+000	6,000E+000
-F	-3,000E+000	-2,000E+000	0,000E+000	0,000E+000	0,000E+000

BV	x1	x2	x3	x4	b
x1	1,000E+000	-1,000E+000	1,000E+000	0,000E+000	1,000E+000
x4	0,000E+000	1,000E+000	-3,000E+000	1,000E+000	3,000E+000
-F	0,000E+000	-5,000E+000	3,000E+000	0,000E+000	3,000E+000

BV	x1	x2	x3	x4	b
x1	1,000E+000	0,000E+000	-2,000E+000	1,000E+000	4,000E+000
x2	0,000E+000	1,000E+000	-3,000E+000	1,000E+000	3,000E+000
-F	0,000E+000	0,000E+000	-1,200E+001	5,000E+000	1,800E+001

Τέλος, στο παρακάτω διάγραμμα απεικονίζονται οι περιορισμοί δομής του παραδείγματος όπως και η αντικειμενική συνάρτηση *F* για δυο διαφορετικές τιμές της (F=-10 και F=-20), επιβεβαιώνοντας την αδυναμία των περιορισμών να περιορίσουν την μείωση της αντικειμενικής συνάρτησης *F*.



<u>Παράδειγμα 4:</u>

Στο παράδειγμα αυτό ζητείται να μεγιστοποιηθεί η αντικειμενική συνάρτηση και τα δεδομένα εισάγονται στην εφαρμογή συμπληρώνοντας τα πλαίσια εισαγωγής κειμένου και επιλέγοντας από την λίστα τον τύπο της ισότητας για κάθε περιορισμό:

<u>Αντικειμενική συνάρτηση:</u>

$\max F = 3x_1 + 2x_2$

<u>Περιορισμοί δομής:</u>

$9x_1 + 10x_2 \leq 330$

$-21x_1 + 4x_2 \leq 36$

$1x_1 + 2x_2 \leq 6$

$6x_1 - x_2 \leq 72$

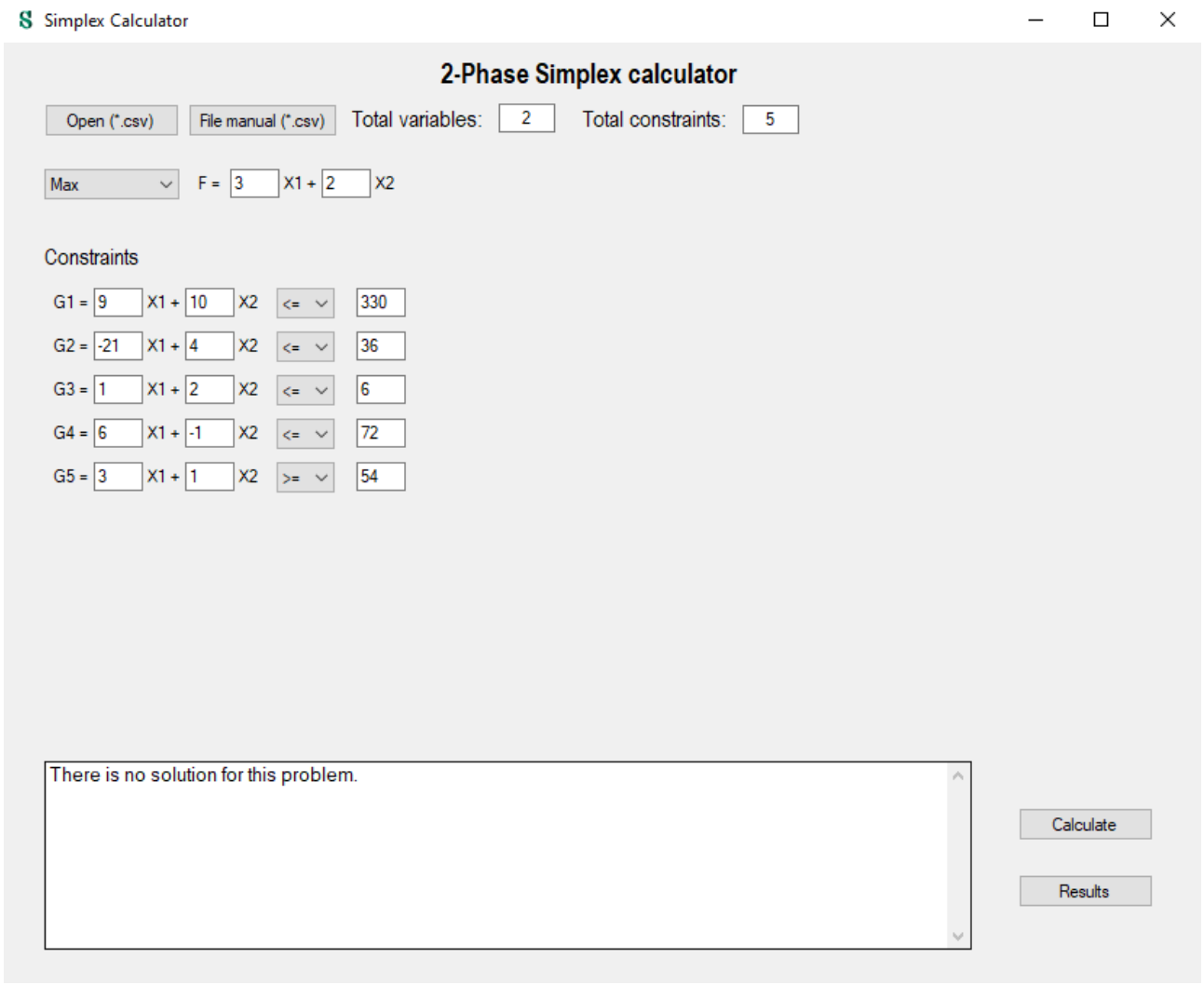
$3x_1 + x_2 \geq 54$

<u>Περιορισμοί μη αρνητικότητας:</u>

$x_1, x_2 \geq 0$

<u>Λύση:</u>

Περίπτωση μη διαθέσιμης λύσης.



Παρακάτω απεικονίζονται οι ενδιάμεσοι πίνακες του παραδείγματος όπως προκύπτουν από το λογισμικό για κάθε επανάληψη του αλγορίθμου Simplex.

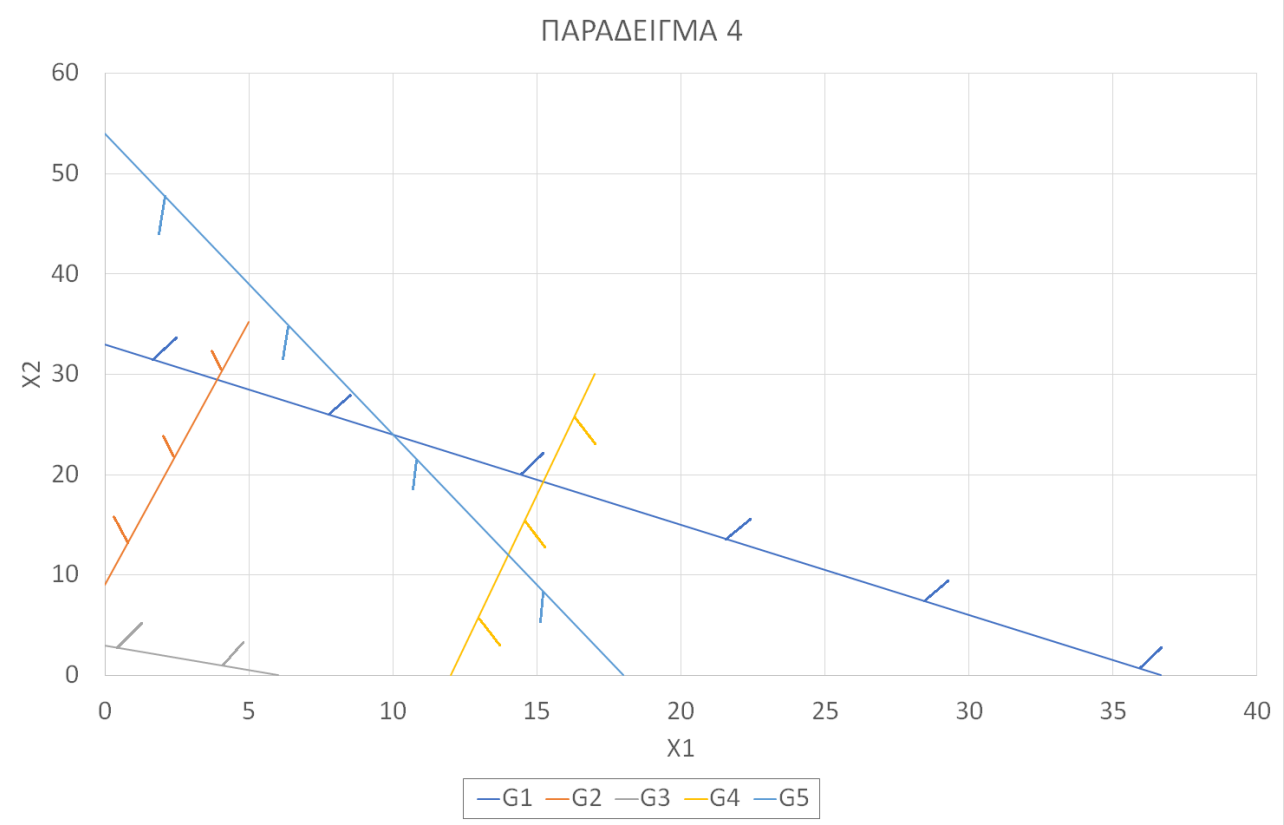
SimplexCalculatorResult.txt - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

PHASE 1:

BV	x1	x2	x3	x4	x5	x6	x7	x8	b
x4	9,000E+000	1,000E+001	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	3,300E+002
x5	-2,100E+001	4,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	3,600E+001
x6	1,000E+000	2,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	6,000E+000
x7	6,000E+000	-1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	7,200E+001
x8	3,000E+000	1,000E+000	-1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	5,400E+001
-F	-3,000E+000	-2,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000
-w	-3,000E+000	-1,000E+000	1,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	0,000E+000	-5,400E+001

BV	x1	x2	x3	x4	x5	x6	x7	x8	b
x4	0,000E+000	-8,000E+000	0,000E+000	1,000E+000	0,000E+000	-9,000E+000	0,000E+000	0,000E+000	2,760E+002
x5	0,000E+000	4,600E+001	0,000E+000	0,000E+000	1,000E+000	2,100E+001	0,000E+000	0,000E+000	1,620E+002
x1	1,000E+000	2,000E+000	0,000E+000	0,000E+000	0,000E+000	1,000E+000	0,000E+000	0,000E+000	6,000E+000
x7	0,000E+000	-1,300E+001	0,000E+000	0,000E+000	0,000E+000	-6,000E+000	1,000E+000	0,000E+000	3,600E+001
x8	0,000E+000	-5,000E+000	-1,000E+000	0,000E+000	0,000E+000	-3,000E+000	0,000E+000	1,000E+000	3,600E+001
-F	0,000E+000	4,000E+000	0,000E+000	0,000E+000	0,000E+000	3,000E+000	0,000E+000	0,000E+000	1,800E+001
-w	0,000E+000	5,000E+000	1,000E+000	0,000E+000	0,000E+000	3,000E+000	0,000E+000	0,000E+000	-3,600E+001

Τέλος, όπως φαίνεται και στην παρακάτω διαγραμματική απεικόνιση του παραδείγματος, λόγω των περιορισμών δομής δεν υπάρχει χωρίο δυνατών λύσεων. Δηλαδή δεν υπάρχει σημείο (x_1, x_2) που να ικανοποιεί όλους τους περιορισμούς.



6. Βιβλιογραφία

1. **Κώστογλου, Βασίλης.** *Επιχειρησιακή Έρευνα & Οργάνωση Συστημάτων Παραγωγής.* Θεσσαλονίκη : Εκδόσεις ΤΖΙΟΛΑ, 2019. 978-960-418-568-9.

2. **Steven C. Chapra, Raymond P. Canale.** *Αριθμητικές Μέθοδοι για Μηχανικούς.* Θεσσαλονίκη : Εκδόσεις ΤΖΙΟΛΑ, 2018. 978-960-418-763-8.

3. **Tutorial: Create a WinForms app with Visual Basic. [Ηλεκτρονικό] Microsoft Corporation, 28 Φεβρουαρίου 2023.** (https://learn.microsoft.com/en-us/visualstudio/ide/create-a-visual-basic-winform-in-visual-studio?view=vs-2019&fbclid=IwAR2ZgLyAetVeKyfbnhjcjEDeE_Tzvt7Tw2ZOOqqXRwQMoi5cNFPjzC7Ae-I.)

4. **FileSystem.OpenTextFileReader Method. [Ηλεκτρονικό] Microsoft Corporation.** (https: //learn.microsoft.com/en-us/dotnet/api/microsoft.visualbasic.fileio.filesystem.opentextfilereader?view=netframework-4.7.2&fbclid=IwAR1Rzw5YpG8L6an1M7iRJRaI2IdJrO9SOtL_l6hcjl4_m3D-DFdfgesrxMk.)

5. **How to: Read Text from Files with a StreamReader (Visual Basic). [Ηλεκτρονικό] Microsoft Corporation, 15 Σεπτέμβριος 2021.** (https://learn.microsoft.com/fi-FI/dotnet/visual-basic/developing-apps/programming/drives-directories-files/how-to-read-text-from-files-with-a-streamreader?fbclid=IwAR3HdynJ_O5gJ0FDANgGHvZMtYY6fltNx3cO5ta1rULKhKfmqPXMHf0V Ars.)

6. **OpenFileDialog Component Overview (Windows Forms). [Ηλεκτρονικό] Microsoft Corporation, 18 Αυγούστου 2022.** (https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/openfiledialog-component-overview-windows-forms?view=netframeworkdesktop-4.8&fbclid=IwAR1oksKfQqDRTrUZtCRdzLpSPToOEa-ug-m2_hDZiLScAeEaRQkolcyDcd4.)

7. **OpenFileDialog Class. [Ηλεκτρονικό] Microsoft .** (https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.openfiledialog?view=windowsdesktop-7.0&fbclid=IwAR0HxXlFzMj3a7Yup7g4HjlLVHl1pykWa8WT4e8tShRktICq6r4nAZvlzGY.)

8. **ListBox Class. [Ηλεκτρονικό] Microsoft Corporation.** (https://learn.microsoft.com/en-us/ dotnet/api/system.windows.forms.listbox?view=netframework-4.7.2&fbclid=IwAR0o67-7g1jRk_UO2NinC4TzdEUM46eYXoKpg8t_LkBRExl3alIZLOLlkqk.)

9. **ListBox Control Overview (Windows Forms). [Ηλεκτρονικό] Microsoft Corporation, 6 Φερβρουάριος 2023.** (https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/ listbox-control-overview-windows-forms?view=netframeworkdesktop-4.8&fbclid=IwAR3PgW3cF3oitxmkh2uCDMv8wyeCZHR8qmpF298_JzoAHeMdXdZ4BjJLgYY.)

10. **Label Class. [Ηλεκτρονικό] Microsoft Corporation.** (https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.label?view=netframework-4.7.2&fbclid=IwAR2m08MrrkMB3ibxL5d8Sra0hHk1FRiUJha1HnesbCGuNko820ziIcHP9dI.)