Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Κατανεμημένα Συστήματα, Ασφάλεια και Αναδυόμενες Τεχνολογίες Πληροφορίας»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ανάπτυξη αυτοματοποιημένης διαδικασίας υπολογισμού μοντέλων της Περιβάλλουσας Ανάλυσης Δεδομένων σε παραγωγικές διαδικασίες δύο σταδίων** **Software Development of Network Data Envelopment Analysis models for two-stage processes**
Ονοματεπώνυμο Φοιτητή	**Ελένη Τραυλού**
Πατρώνυμο	**Γεώργιος**
Αριθμός Μητρώου	**ΜΠΚΣΑ/ 18022**
Επιβλέπων	**Δημήτριος Δεσπότης, Καθηγητής**

Ημερομηνία Παράδοσης **Ιανουάριος 2021**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Δημήτριος Δεσπότης Καθηγητής	Δημήτριος Αποστόλου Καθηγητής	Διονύσιος Σωτηρόπουλος Επίκουρος Καθηγητής

ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή διατριβή στοχεύει στην εκτίμηση της αποδοτικότητας παραγωγικών μονάδων μέσω διακεκριμένων μοντέλων της Πολυσταδιακής Περιβάλλουσας Ανάλυσης Δεδομένων. Η Πολυσταδιακή ΠΑΔ αποτελεί επέκταση της κλασικής ΠΑΔ με την έννοια της σύνθετης εσωτερικής δομής που παρουσιάζουν οι παραγωγικές μονάδες. Συγκεκριμένα, η διατριβή θα εστιάσει σε διαδικασίες δύο σταδίων διατεταγμένων σε σειρά. Διαδεδομένες προσεγγίσεις σε διαδικασίες με αυτή την μορφή είναι αυτές των Kao et al. (2008) και Chen et al. (2009) με το πολλαπλασιαστικό και το προσθετικό μοντέλο αντίστοιχα καθώς και η προσέγγιση των Despotis et al. (2016) με την μέθοδο του αδύναμου κρίκου. Η ανάπτυξη των προαναφερθέντων μοντέλων θα εκτελεστεί μέσω VBA scripts για την εφαρμογή του Excel με πρόθεση την δημιουργία μιας αυτοματοποιημένης διαδικασίας αποτίμησης της αποδοτικότητας των παραγωγικών μονάδων στην Πολυσταδιακή ΠΑΔ.

ABSTRACT

Subject of this thesis is the performance assessment of production units through prominent approaches in Network Data Envelopment Analysis. Network DEA is an extension of conventional DEA where each production unit has a complex internal structure. This thesis will focus on two-stage series processes. Common approaches in such context are those proposed by Kao et al. (2008) and Chen et al. (2009) with the multiplicative and additive model respectively as well as the weak link approach proposed by Despotis et al. (2016). The development of the aforementioned models will be performed through VBA scripts for Excel Application with the intention of automating the process of performance assessment of production units in Network DEA.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Η μέτρηση της αποδοτικότητας αποτελεί καθοριστικό παράγοντα για την βελτίωση και την εύρυθμη λειτουργία των οργανισμών. Ως αποδοτικότητα εννοείται ο βαθμός στον οποίο οι οργανισμοί μετασχηματίζουν αποτελεσματικά το εισαγόμενο κόστος σε εξαγόμενη αξία. Μια από τις δημοφιλέστερες τεχνικές για την εκτίμηση της αποδοτικότητας αποτελεί η Περιβάλλουσα Ανάλυση Δεδομένων – ΠΑΔ (Data Envelopment Analysis - DEA). Οι οργανισμοί προς ανάλυση αναφέρονται ως μονάδες παραγωγής ή μονάδες απόφασης (Decision Making Units – DMUs). Για την εκτίμηση των μονάδων αυτών είναι απαραίτητο να είναι ομοειδής, συγκρίσιμες καθώς και να δέχονται ίδιο αριθμό εισροών για την παραγωγή ίδιου αριθμού εκροών. Στις κλασικές μεθοδολογίες της ΠΑΔ οι μονάδες παραγωγής προς μελέτη συγκροτούνται από ένα μόνο στάδιο το οποίο χρησιμοποιεί εισροές για την παραγωγή εκροών. Σε πολλές περιπτώσεις ωστόσο η εσωτερική δομή είναι γνωστή και επηρεάζει την μέτρηση της αποδοτικότητας. Η Πολυσταδιακή Περιβάλλουσα Ανάλυση Δεδομένων (Network DEA), ως επέκταση της κλασικής ΠΑΔ, είναι μια μεθοδολογία για την αξιολόγηση μονάδων παραγωγής με γνωστή εσωτερική δομή. Κάθε παραγωγική μονάδα αντιμετωπίζεται πλέον ως δίκτυο διεργασιών (σταδίων) οι οποίες συνδέονται και αλληλοεπιδρούν μέσω ενδιάμεσων μεγεθών. Ως εκ τούτου τα μοντέλα της Πολυσταδιακής ΠΑΔ παρέχουν καλύτερες μετρήσεις για την αποδοτικότητα σε διαδικασίες με σύνθετη εσωτερική δομή.

Η διατύπωση των μοντέλων της Πολυσταδιακής ΠΑΔ καθιστά δυνατή την επίλυση τους ως προβλήματα γραμμικού προγραμματισμού. Η αποδοτικότητας μιας μονάδας απόφασης μπορεί να εκτιμηθεί με την λύση ενός γραμμικού προβλήματος ενώ για την εκτίμηση πολλαπλών μονάδων απόφασης χρειάζεται η επίλυση αντίστοιχου αριθμού προβλημάτων. Μάλιστα για την Πολυσταδιακή ΠΑΔ ορισμένα μοντέλα απαιτούν την επίλυση επιπλέον προβλημάτων ώστε να προσδιοριστεί αποτελεσματικά η αποδοτικότητα μιας μονάδας απόφασης. Η μέτρηση της αποδοτικότητας για αυτά τα μοντέλα μπορεί να καταστεί αρκετά χρονοβόρα ειδικά όσο ο αριθμός των μονάδων απόφασης μεγαλώνει.

Η παρούσα εργασία θα εστιάσει στην διαμόρφωση ενός προγράμματος για την αυτοματοποίηση των διαδικασιών που απαιτούνται για την εκτίμηση της αποδοτικότητας των μονάδων απόφασης σε επιλεγμένα μοντέλα της ΠΑΔ. Η ανάπτυξη του προγράμματος θα πραγματοποιηθεί με την χρήση της γλώσσας προγραμματισμού του MS Excel (Visual Basic for Applications) και του πρόσθετου προγράμματος Solver που εμπεριέχεται σε αυτό. Τα μοντέλα προς υλοποίηση ανήκουν στο πλαίσιο της Πολυσταδιακής Περιβάλλουσας Ανάλυσης Δεδομένων για την εκτίμηση της αποδοτικότητας μονάδων που αποτελούνται από δύο στάδια παραγωγής διατεταγμένα σε σειρά.

Στο κεφάλαιο που ακολουθεί γίνεται μια σύντομη αναφορά στην κλασική θεωρία της ΠΑΔ και τις βασικές της έννοιες, ενώ στην συνέχεια του κεφαλαίου παρουσιάζεται αναλυτικότερα η Πολυσταδιακή ΠΑΔ και τα μοντέλα προς υλοποίηση. Το τρίτο κεφάλαιο επικεντρώνεται στην ανάπτυξη του προγράμματος και τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή του. Στο τέταρτο κεφάλαιο γίνεται η παρουσίαση του προγράμματος για την αυτοματοποιημένη επίλυση των μοντέλων ενώ η εργασία κλείνει με το πέμπτο κεφάλαιο όπου παρουσιάζονται τα συμπεράσματα και οι περιορισμοί του προγράμματος.

Κεφάλαιο 2

Περιβάλλουσα Ανάλυση Δεδομένων

Η Περιβάλλουσα Ανάλυση Δεδομένων (ΠΑΔ) αποτελεί βασικό εργαλείο για την εκτίμηση της αποδοτικότητας παραγωγικών μονάδων. Οι παραγωγικές μονάδες που μελετώνται, αναφέρονται ως Μονάδες Απόφασης και βασική λειτουργία τους είναι η μετατροπή πολλαπλών εισροών σε εκροές. Η ΠΑΔ είναι μια μη-παραμετρική μέθοδος γραμμικού προγραμματισμού καθώς καμία υπόθεση δεν γίνεται σε σχέση με την συνάρτηση παραγωγής. Συνεπώς ο μηχανισμός μετατροπής των εισροών σε εκροές παραμένει άγνωστος και η μέτρηση της αποδοτικότητας βασίζεται σε πραγματικές παρατηρήσεις. Βασική ιδέα της ΠΑΔ είναι η δημιουργία ενός συνόρου μέγιστων παραγωγικών δυνατοτήτων βασισμένο στις μονάδες απόφασης που παρουσιάζουν την μέγιστη αποδοτικότητα. Το σύνορο αυτό λειτουργεί σαν σημείο αναφοράς για τις μη-αποδοτικές μονάδες καθώς συγκρινόμενες με αυτό μπορεί να προσδιοριστεί το επίπεδο της αποδοτικότητα τους.

Η ικανότητα κάθε παραγωγικής μονάδας να μετασχηματίζει όσο το δυνατόν αποτελεσματικότερα τις εισροές σε εκροές αποτελεί και την ουσία της αποδοτικότητας της. Η αποδοτικότητα λοιπόν μιας μονάδας μπορεί να οριστεί ως το πηλίκο του σταθμισμένου αθροίσματος των εκροών προς το σταθμισμένο άθροισμα των εισροών όπου τα βάρη ορίζονται από την ίδια την αποτιμωμένη μονάδα κατά τρόπο τέτοιο ώστε να μεγιστοποιεί τον δείκτη αποδοτικότητα της.

Τις βάσεις της Περιβάλλουσας Ανάλυσης Δεδομένων έθεσε ο Farell (1957). Η μεθοδολογία της ΠΑΔ εφαρμόστηκε για πρώτη φορά από τους Charnes, Cooper και Rhodes (1978) και αναπτύχθηκε περαιτέρω από τους Banker et al (1984). Οι δύο παραπάνω εφαρμογές είχαν σαν συνέπεια τη δημιουργία των δύο βασικών μοντέλων της ΠΑΔ, του CCR και BCC με τις επωνυμίες κατά αντιστοιχία με τα αρχικά των συγγραφέων. Τα μοντέλα αυτά διαφέρουν ως προς την υπόθεση κλίμακας αποδόσεων η οποία επιλέγεται για την εφαρμογή τους. Συγκεκριμένα, το CCR βρίσκει εφαρμογή υπό την υπόθεση περί κλίμακας σταθερών αποδόσεων (Constant Returns to Scale - CRS) σύμφωνα με την οποία κάθε ποσοστιαία μεταβολή των εισροών αποφέρει αντίστοιχη ποσοστιαία μεταβολή στις εκροές. Αντίθετα, το BCC εφαρμόζεται υπό την υπόθεση περί κλίμακας μεταβλητών αποδόσεων (Variable Returns to Scale - VRS) όπου η σχέση εισροών/εκροών είναι μη γραμμική καθώς δεν υφίσταται η αναλογική μεταβολή τους.

2.1 Βασικές Έννοιες

Μονάδες Απόφασης (DMUs)

Όπως προαναφέρθηκε οι Μονάδες Απόφασης (Decision making units - DMUs) είναι παραγωγικές μονάδες οι οποίες μετατρέπουν εισροές σε εκροές. Ως εισροές (Inputs) εννοούνται οι πόροι οι οποίοι καταναλώνονται (εισερχόμενο κόστος) από την εκάστοτε DMU, ενώ ως εκροές (Outputs) τα αποτελέσματα της παραγωγικής διαδικασίας (εξαγόμενη αξία). Πρόκειται για ομοειδής παραγωγικές μονάδες οι οποίες καθίστανται συγκρίσιμες καθώς εκτελούν τις ίδιες λειτουργίες όσον αφορά το σύνολο των εισροών που λαμβάνουν και το σύνολο των εκροών που παράγουν. Για την αποτελεσματικότερη αποτίμηση της αποδοτικότητας των Μονάδων Απόφασης είναι ιδιαιτέρως σημαντικό οι εισροές και οι εκροές που τις χαρακτηρίζουν να αντικατοπτρίζουν όσον το δυνατόν καλύτερα την λειτουργία τους. Οι εισροές και εκροές της παραγωγικής διαδικασίας μπορεί να είναι διαφορετικού τύπου και μονάδας μέτρησης.

Σχήμα 1: Σχηματική Αναπαράσταση μιας Μονάδας Απόφασης (DMU)

Προσανατολισμός Μοντέλων (Input/Output Oriented)

Στόχος της ΠΑΔ είναι η εκτίμηση της αποδοτικότητας των παραγωγικών μονάδων. Οι μονάδες με την μέγιστη αποδοτικότητα διαμορφώνουν το σύνορο αποδοτικότητας ως προς το οποίο θα συγκριθούν οι μη-αποδοτικές μονάδες. Η μετάβαση μιας μη-αποδοτικής μονάδας στο σύνορο αποδοτικότητας μπορεί να επιτευχθεί μέσω δύο προσεγγίσεων. Αν το ενδιαφέρον εντοπίζεται στην ελαχιστοποίηση των εισροών (Input Oriented) τότε υπολογίζεται ο βαθμός στον οποίο η μονάδα μπορεί να μειώσει τις εισροές της για την παραγωγή συγκεκριμένης ποσότητας εκροής, ενώ αν ενδιαφέρει η μεγιστοποίηση των εκροών (Output Oriented) υπολογίζεται ο βαθμός στον οποίο μπορεί να αυξήσει τις εκροές της για μια συγκεκριμένη ποσότητα εισροής.

Το Μοντέλο Charnes, Cooper και Rhodes (1978)

Το CCR μοντέλο αναπτύχθηκε από τους Charnes et al (1978) για την μέτρηση της σχετικής αποδοτικότητας των Μονάδων Απόφασης υπό κλίμακα σταθερών αποδόσεων(CRS). Η αποδοτικότητα μιας Μονάδας Απόφασης (DMU) ορίζεται ως το σταθμισμένο άθροισμα των εκροών προς το σταθμισμένο άθροισμα των εισροών. Το μοντέλο που δίνει την CCR - αποδοτικότητα της DMU j_0 έχει ως εξής:

$$\max e_{j_0} = \frac{\sum u_r Y_{rj_0}}{\sum v_i X_{ij_0}} \qquad (1.1)$$

$$\text{s.t.} \quad \frac{\sum u_r Y_{rj}}{\sum v_i X_{ij}} \leq 1$$

$$v_i, u_r \geq 0 \,, \; j = 1, \dots, n$$

Το παραπάνω μοντέλο επιλύεται ξεχωριστά για κάθε DMU ώστε να υπολογιστεί η αποδοτικότητα της, με τα βάρη v_i, u_r να επιλέγονται από την εκάστοτε DMU j_0 έτσι ώστε να επιτύχει την όσο το δυνατόν μεγιστοποίηση της αποδοτικότητας της. Ο τρόπος με τον οποίο είναι ορισμένο το μοντέλο περιορίζει τις τιμές αποδοτικότητας που μπορεί να επιτύχει κάθε DMU στο διάστημα [0,1].

Το αντίστοιχο μοντέλο γραμμικού προγραμματισμού (1.2) προκύπτει από το (1.1) μέσω του C–C μετασχηματισμού (Charnes and Cooper (1962)) και έχει τη εξής μορφή:

$$\max e_{j_0} = \sum u_r Y_{rj_0} \quad (1.2)$$
$$\text{s.t.} \quad \sum v_i X_{ij_0} = 1$$
$$\sum u_r Y_{rj} - \sum v_i X_{ij} \leq 0$$
$$v_i, u_r \geq 0\,,\ j = 1, \dots, n$$

Για μία βέλτιστη λύση του μοντέλου (1.2), έστω (v_i^*, u_r^*), η αποδοτικότητα της DMUj_0, $e_{j_0}^*$ προκύπτει απευθείας από την αντικειμενική συνάρτηση. Η DMU j_0 είναι CCR-αποδοτική αν και μόνο αν $e_{j_0}^* = 1$ και υπάρχει μία τουλάχιστον βέλτιστη λύση (v_i^*, u_r^*) με $v_i^* > 0$ *και* $u_r^* > 0$. Διαφορετικά η DMU j_0 είναι CCR-μη αποδοτική.

Το μοντέλο CCR επιτρέπει την επιλογή του προσανατολισμού είτε στις εισροές είτε στις εκροές. Το (1.2) έχει προσανατολισμό στις εισροές, ενώ αν το ενδιαφέρον εντοπίζεται στις εκροές τότε το αντίστοιχο CCR μοντέλο με προσανατολισμό στις εκροές έχει την μορφή:

$$min\, e_{j_0} = \sum v_i X_{ij_0} \quad (1.3)$$
$$\text{s.t.} \quad \sum u_r Y_{rj_0} = 1$$
$$\sum u_r Y_{rj} - \sum v_i X_{ij} \leq 0$$
$$v_i, u_r \geq 0\,,\ j = 1, \dots, n.$$

Το Μοντέλο Banker, Charnes και Cooper (1984)

Το μοντέλο BCC, αποτελεί επέκταση του κλασικού CCR μοντέλου με στόχο την εφαρμογή του υπό την υπόθεση περί κλίμακας μεταβλητών αποδόσεων (VRS). Η διαφορά των δύο μοντέλων ως προς την δομή τους είναι η επιπλέον μεταβλητή, έστω d, στο μοντέλο BCC. Το μοντέλο BCC με προσανατολισμό στις εισροές είναι το ακόλουθο:

$$\max e_{j_0} = \sum u_r Y_{rj_0} - d \quad (1.4)$$
$$\text{s.t.} \quad \sum v_i X_{ij_0} = 1$$
$$\sum u_r Y_{rj} - d - \sum v_i X_{ij} \leq 0$$
$$v_i, u_r \geq 0,\ d \in \mathbb{R},\ j = 1, \dots, n$$

Ενώ το αντίστοιχο μοντέλο με προσανατολισμό στις εκροές:

$$min\, e_{j_0} = \sum v_i X_{ij_0} - p \quad (1.5)$$
$$\text{s.t.} \quad \sum u_r Y_{rj_0} = 1$$
$$\sum u_r Y_{rj} - \sum v_i X_{ij} + p \leq 0$$
$$v_i, u_r \geq 0,\ p \in \mathbb{R},\ j = 1, \dots, n.$$

Η αποδοτικότητα της DMU j_0 είναι ανάλογη με το CCR μοντέλο, με την DMU j_0 αποδοτική αν και μόνο αν $e_{j_0}^* = 1$ και υπάρχει μία τουλάχιστον βέλτιστη λύση (v_i^*, u_r^*) με $v_i^* > 0$ *και* $u_r^* > 0$.

2.2 Πολυσταδιακή Περιβάλλουσα Ανάλυση Δεδομένων

Στις κλασικές μεθοδολογίες της ΠΑΔ οι μονάδες απόφασης μετατρέπουν εισροές σε εκροές ενώ η εσωτερική τους δομή παραμένει άγνωστη. Κατά αυτόν τον τρόπο το σύστημα αντιμετωπίζεται σαν μια ενιαία διαδικασία ενώ στην πραγματικότητα μπορεί να αποτελείται από διακριτές διαδικασίες των οποίων η εσωτερική δομή είναι γνωστή αλλά και κρίσιμη για την αποτίμηση της αποδοτικότητας μιας Μονάδας Απόφασης. Μια επέκταση της κλασικής ΠΑΔ που καλύπτει αυτή την απαίτηση είναι η Πολυσταδιακή Περιβάλλουσα Ανάλυση Δεδομένων (Network DEA).

Η Πολυσταδιακή ΠΑΔ εφαρμόζεται για την αξιολόγηση μονάδων απόφασης οι οποίες αποτελούνται από πολλαπλά στάδια με σύνθετη δομή. Συγκεκριμένα, κάθε μονάδα απόφασης λειτουργεί ως δίκτυο από διατεταγμένα στάδια τα οποία συνδέονται μέσω ενδιάμεσων μεγεθών. Τα ενδιάμεσα αυτά μεγέθη αποτελούν ταυτόχρονα τις εκροές ενός σταδίου και τις εισροές άλλου. Συνεπώς, η εκτίμηση της συνολικής αποδοτικότητας του συστήματος γίνεται λαμβάνοντας υπόψη τις αποδοτικότητες των μελών της.

Η αποτίμηση της αποδοτικότητας των Μονάδων Απόφασης στην Πολυσταδιακή ΠΑΔ μπορεί να πραγματοποιηθεί μέσω διαφορετικών προσεγγίσεων. Τα δύο βασικά πρότυπα είναι αυτά της ανεξάρτητης αξιολόγησης των Μονάδων Απόφασης και των επιμέρους σταδίων τους και την ολιστική αξιολόγηση. Η ανεξάρτητη αξιολόγηση του συστήματος αποτελεί μια στοιχειώδη μέθοδο για την εκτίμηση της αποδοτικότητας των Μονάδων Απόφασης και των σταδίων τους. Παρότι αναγνωρίζεται η εσωτερική δομή των Μονάδων Απόφασης, η αποδοτικότητα του συστήματος και των επιμέρους σταδίων υπολογίζονται ως ξεχωριστές διαδικασίες χωρίς να λαμβάνεται υπόψιν οποιαδήποτε αλληλεξάρτηση έχουν μεταξύ τους με αποτέλεσμα να μην διαφαίνεται η επίδραση των επιμέρους σταδίων στην συνολική αποδοτικότητα. Αντίθετα, με την ολιστική αξιολόγηση των Μονάδων Απόφασης η εσωτερική δομή αλλά και η αλληλεξάρτηση των σταδίων λαμβάνονται υπόψιν ενώ η συνολική αποδοτικότητα καθώς και οι αποδοτικότητες των σταδίων αποτιμώνται από κοινού μέσω ενός γραμμικού προγράμματος.

Δύο κατηγορίες στο πρότυπο της ολιστικής αξιολόγησης των Μονάδων Απόφασης είναι αυτή της ανάλυσης της αποδοτικότητας (efficiency decomposition approach) και της σύνθεσης της αποδοτικότητας (composition approach). Οι μέθοδοι που θα αναπτυχθούν παρακάτω ανήκουν και στις δύο κατηγορίες με το πολλαπλασιαστικό και προσθετικό μοντέλο να χρησιμοποιεί την προσέγγιση της ανάλυσης της συνολικής αποδοτικότητας στην αποδοτικότητα των επιμέρους σταδίων και την μέθοδο του αδύναμου κρίκου να συνθέτει τις επιμέρους αποδοτικότητες για την εκτίμηση της συνολικής αποδοτικότητας της παραγωγικής μονάδας.

2.3 Μοντέλα Προς Υλοποίηση

Η παρούσα εργασία επικεντρώνεται σε μοντέλα της ΠΑΔ που περιέχουν δύο στάδια διατεταγμένα σε σειρά. Συγκεκριμένα τα μοντέλα που θα αναπτυχθούν είναι το πολλαπλασιαστικό μοντέλο των Kao et al. (2008), το προσθετικό μοντέλο των Chen et al. (2009) καθώς και η μέθοδος του Αδύναμου κρίκου των Despotis et al. (2016).

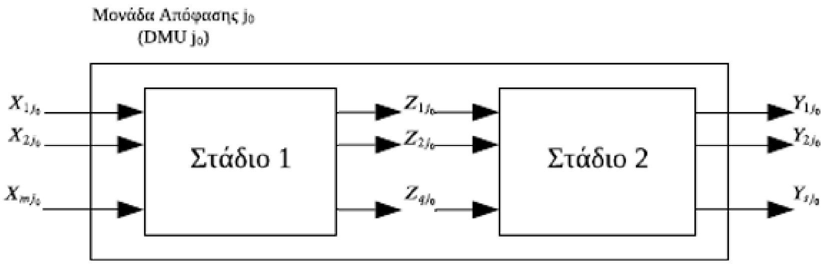
Οι **Kao et al. (2008)** παρουσιάζουν το πολλαπλασιαστικό μοντέλο σε παραγωγικές διαδικασίες δύο σταδίων. Υποθέτουν ότι η συνολική αποδοτικότητα του συστήματος ισούται με το πηλίκο τον επιμέρους σταδίων. Προτείνουν έτσι ένα γραμμικό πρόγραμμα υπολογισμού της συνολικής αποδοτικότητας του συστήματος την οποία αναλύουν εκ των υστέρων στις επιμέρους αποδοτικότητες των σταδίων. Επίσης, παρουσιάζουν και μια τεχνική για τον έλεγχο μοναδικότητας της αποδοτικότητας των σταδίων.

Οι **Chen et al. (2009)**, ορίζουν την συνολική αποδοτικότητα του συστήματος ως τον σταθμισμένο μέσο των αποδόσεων των επιμέρους σταδίων. Σύμφωνα με αυτό προτείνουν το προσθετικό μοντέλο σε παραγωγικές διαδικασίες δύο σταδίων. Υποθέτοντας ότι τα βάρη των επιμέρους σταδίων αντικατοπτρίζουν την σημασία τους, αναπαριστούν το μέγεθος κάθε σταδίου ως τμήμα των συνολικών εισροών που εισέρχονται σε αυτό. Οι αποδοτικότητες των επιμέρους σταδίων προκύπτουν από τον υπολογισμό της συνολικής αποδοτικότητας με τρόπο παρόμοιο με αυτόν του μοντέλου των Kao et al. (2008). Το πλεονέκτημα του μοντέλου των Chen et al. (2009) έναντι των Kao et al. (2008) είναι η εφαρμογή του σε προβλήματα υπό την υπόθεση περί κλίμακας μεταβλητών αποδόσεων.

Οι **Despotis et al. (2016)** διαφοροποιούνται από τις προηγούμενες μεθόδους παρουσιάζοντας την συνθετική προσέγγιση για την εκτίμηση της αποδοτικότητας. Προτείνουν ένα μοντέλο με οδηγό την βελτιστοποίηση τον επιμέρους σταδίων και την συνολική αποδοτικότητα να προκύπτει εκ των υστέρων. Βασιζόμενοι στον ρόλο του αδύναμου κρίκου στις εφοδιαστικές αλυσίδες διατυπώνουν έναν νέο ορισμό για την εύρεση της συνολικής αποδοτικότητας του συστήματος ως τη μέγιστη ροή του δικτύου που σχηματίζεται από τα δύο στάδια και υπολογίζεται από την ελάχιστη τομή του. Το μοντέλο αυτό παρέχει μοναδικές και αμερόληπτες μετρήσεις για τις αποδοτικότητες των επιμέρους σταδίων.

2.3.1 Πολλαπλασιαστικό μοντέλο (Kao, Hwang et al. 2008)

Οι Kao et al. (2008) εισάγουν μια καινοτόμα προσέγγιση για την εκτίμηση της αποδοτικότητας των μονάδων που αποτελούνται από δύο στάδια παραγωγής με σειριακή διάταξη. Σε αυτό το μοντέλο η εσωτερική δομή της παραγωγικής διαδικασίας αποτελείται από ενδιάμεσα μεγέθη, Z_{pj_0},τα οποία λειτουργούν ως εκροές για το πρώτο στάδιο και ως εισροές για το δεύτερο. Το πολλαπλασιαστικό μοντέλο βασίζεται στο κλασικό μοντέλο CCR (Charnes et al. 1978) υπό κλίμακα σταθερών αποδόσεων (CRS). Η προσέγγιση των Kao και Hwang (2008) βασίζεται στην υπόθεση ότι τα βάρη w_p για τα ενδιάμεσα μεγέθη Z_{pj_0} είναι ίδια και για τα δύο στάδια ανεξάρτητα από το ότι αποτελούν εκροές και εισροές για το κάθε στάδιο αντίστοιχα.



Σχήμα 2: Σχηματική Αναπαράσταση Μονάδας Απόφασης (DMU) με δύο στάδια διατεταγμένα σε σειρά.

Έστω η υπό αξιολόγηση Μονάδα Απόφασης j_0 (DMU j_0).

Η συνολική απόδοση του συστήματος ορίζεται ως το πηλίκο του σταθμισμένου αθροίσματος των εκροών ως προς το σταθμισμένο άθροισμα των εισροών, $E_{j_0} = \frac{\sum u_r Y_{rj_0}}{\sum v_i X_{ij_0}}$ και οι αποδοτικότητες του πρώτου και δευτέρου σταδίου αντίστοιχα ως,

$E_{j_0}^1 = \frac{\sum w_p Z_{pj_0}}{\sum v_i X_{ij_0}}$ $\kappa\alpha\iota$ $E_{kj_0}^2 = \frac{\sum u_r Y_{rj_0}}{\sum w_p Z_{pj_0}}$.

Σύμφωνα με τα παραπάνω η συνολική αποδοτικότητα του συστήματος αναλύεται ως το γινόμενο των αποδόσεων των επιμέρους σταδίων, $E_{j_0} = E_{j_0}^1 \times E_{j_0}^2$.

Για τον υπολογισμό της συνολικής αποδοτικότητας του συστήματος χρησιμοποιούνται τα παρακάτω μοντέλα, όπου το (2.1) αποτελεί την κλασματική μορφή υπολογισμού της απόδοσης και το (2.2) το αντίστοιχο γραμμικό του με εφαρμογή του C–C μετασχηματισμού:

$$
\begin{aligned}
E_{j_0} = \max\ & \frac{\sum u_r Y_{rj_0}}{\sum v_i X_{ij_0}} \qquad (2.1)\\
\text{s.t.}\quad & \frac{\sum u_r Y_{rj}}{\sum v_i X_{ij}} \le 1\\
& \frac{\sum w_p Z_{pj}}{\sum v_i X_{ij}} \le 1\\
& \frac{\sum u_r Y_{rj}}{\sum w_p Z_{pj}} \le 1\\
& v_i, w_p, u_r \ge 0\,,\ j = 1, \dots, n
\end{aligned}
$$

$$
\begin{aligned}
E_{j_0} = \max\ & \sum u_r Y_{rj_0} \qquad (2.2)\\
\text{s.t.}\quad & \sum v_i X_{ij_0} = 1\\
& \sum u_r Y_{rj} - \sum v_i X_{ij} \le 0\\
& \sum w_p Z_{pj} - \sum v_i X_{ij} \le 0\\
& \sum u_r Y_{rj} - \sum w_p Z_{pj} \le 0\\
& v_i, w_p, u_r \ge 0\,,\ j = 1, \dots, n
\end{aligned}
$$

Έστω (u_r^*, v_i^*, w_p^*) μια βέλτιστη λύση του (2.2) για την DMU j_0. Ο υπολογισμός της συνολικής αποδοτικότητας E_{j_0} και των επιμέρους σταδίων $E_{j_0}^1$ $\kappa\alpha\iota$ $E_{j_0}^2$ προκύπτουν ως εξής:

$E_{j_0} = \sum u_r^* Y_{rj_0}$ $\kappa\alpha\iota$ $E_{j_0}^1 = \sum w_p^* Z_{pj_0}$, $E_{j_0}^2 = \frac{\sum u_r^* Y_{rj_0}}{\sum w_p^* Z_{pj_0}} = \frac{E_{j_0}}{E_{j_0}^1}$

Όπως σημειώνεται από τους Kao και Hwang (2008) είναι πολύ πιθανό τα βέλτιστα βάρη που προκύπτουν από το (2.2) να μην είναι μοναδικά με αποτέλεσμα και η ανάλυση της συνολικής αποδοτικότητας στις δύο επιμέρους να μην είναι μοναδική. Προτείνουν έτσι έναν επιπλέον έλεγχο μοναδικότητας με την εύρεση βαρών τα οποία μεγιστοποιούν την απόδοση του πρώτου σταδίου $E_{j_0}^1$ ενώ διατηρούν την τιμή της συνολικής αποδοτικότητας E_{j_0} σταθερή και όπως υπολογίστηκε από το (2.2). Η μέγιστη αποδοτικότητα για το πρώτο στάδιο δίνεται από το μοντέλο:

$$
\begin{aligned}
& E_{j_0}^{1max} = max \sum w_p Z_{pj_0} \qquad (2.3)\\
\text{s.t.} \quad & \sum v_i X_{ij_0} = 1\\
& \sum u_r Y_{rj} - E_{j_0} \sum v_i X_{ij} \leq 0\\
& \sum u_r Y_{rj} - \sum v_i X_{ij} \leq 0\\
& \sum w_p Z_{pj} - \sum v_i X_{ij} \leq 0\\
& \sum u_r Y_{rj} - \sum w_p Z_{pj} \leq 0\\
& v_i, w_p, u_r \geq 0\ ,\ j = 1, \dots, n
\end{aligned}
$$

Μετά τον υπολογισμό της μέγιστης αποδοτικότητας του πρώτου σταδίου $E_{j_0}^{1max}$, η αποδοτικότητα του δευτέρου σταδίου προκύπτει ως $E_{j_0}^{2min} = {E_{j_0}}/{E_{j_0}^{1max}}$.

Εναλλακτικά, αν το ενδιαφέρον εντοπίζεται στην βελτιστοποίηση του δευτέρου σταδίου $E_{j_0}^{2max}$ τότε αντικαθιστώντας την αντικειμενική συνάρτηση του (2.3) με $\sum u_r Y_{rj_0}$ και τον πρώτο περιορισμό με $\sum w_p Z_{pj_0} = 1$ προκύπτει το ζητούμενο, ενώ η αποδοτικότητα για το πρώτο στάδιο πλέον προκύπτει ως $E_{j_0}^{1min} = {E_k}/{E_{j_0}^{2max}}$.

Επιπλέον, αν $E_{j_0}^{1max} \neq E_{j_0}^{1min}$ ή αν $E_{j_0}^{2max} \neq E_{j_0}^{2min}$ τότε η ανάλυση της αποδοτικότητας των δύο επιμέρους σταδίων δεν είναι μοναδική και υπάρχουν εναλλακτικές βέλτιστες λύσεις που αποδίδουν την ίδια συνολική αποδοτικότητα στην μονάδα απόφασης.

2.3.2 Προσθετικό μοντέλο (Chen et al. 2009)

Στο πλαίσιο του προσθετικού μοντέλου των Chen et al. (2009), η συνολική αποδοτικότητα καθώς και οι αποδοτικότητες των επιμέρους σταδίων για την DMU j_0, υπό κλίμακα σταθερών αποδόσεων (CRS), ορίζονται ως εξής:

$$E_{j_0} = \frac{\sum w_p Z_{pj_0} + \sum u_r Y_{rj_0}}{\sum v_i X_{ij_0} + \sum w_p Z_{pj_0}}, \quad E_{j_0}^1 = \frac{\sum w_p Z_{pj_0}}{\sum v_i X_{ij_0}} \quad \text{και} \quad E_{j_0}^2 = \frac{\sum u_r Y_{rj_0}}{\sum w_p Z_{pj_0}}.$$

Οι ορισμοί των αποδόσεων των επιμέρους σταδίων παραμένουν ίδιοι με το πολλαπλασιαστικό μοντέλο ενώ η διαφοροποίηση εντοπίζεται στον ορισμό της συνολικής αποδοτικότητας της παραγωγικής διαδικασίας. Σύμφωνα με τον ορισμό των Chen et al. (2009), τα ενδιάμεσα μεγέθη εμφανίζονται και στους δύο όρους του κλάσματος της συνολικής αποδοτικότητας και κατά συνέπεια υπολογίζονται ταυτόχρονα ως εισροές και εκροές. Η ανάλυση της συνολικής αποδοτικότητας λοιπόν ορίζεται ως ο σταθμισμένος μέσος των αποδόσεων των επιμέρους σταδίων και προκύπτει από την παρακάτω σχέση:

$$E_{j_0} = t_1 \cdot \frac{\sum w_p Z_{pj_0}}{\sum v_i X_{ij_0}} + t_2 \cdot \frac{\sum u_r Y_{rj_0}}{\sum w_p Z_{pj_0}} = t_1 \cdot E_{j_0}^1 + t_2 \cdot E_{j_0}^2 \quad \text{με } t_1, t_2 \text{ τέτοια ώστε } t_1 + t_2 = 1.$$

Τα βάρη ορίζονται ως συναρτήσεις των μεταβλητών απόφασης με τρόπο τέτοιο ώστε να αντικατοπτρίζουν το μέγεθος των εισροών του κάθε σταδίου ($\sum v_i X_{ij_0}$ $και$ $\sum w_p Z_{pj_0}$ $αντίστοιχα$) ως προς τις συνολικές εισροές του συστήματος, ως εξής:

$$t_1 = \frac{\sum v_i X_{ij_0}}{\sum v_i X_{ij_0} + \sum w_p Z_{pj_0}} \quad και \quad t_2 = \frac{\sum w_p Z_{pj_0}}{\sum v_i X_{ij_0} + \sum w_p Z_{pj_0}}$$

Η συνολική αποδοτικότητα του συστήματος προκύπτει από τα παρακάτω μοντέλα, όπου το (2.4) αποτελεί την κλασματική μορφή υπολογισμού της απόδοσης και το (2.5) το αντίστοιχο γραμμικό του με εφαρμογή του C–C μετασχηματισμού:

$$E_{j_0} = \max \frac{\sum w_p Z_{pj_0} + \sum u_r Y_{rj_0}}{\sum v_i X_{ij_0} + \sum w_p Z_{pj_0}} \qquad (2.4)$$

$$\begin{aligned} \text{s.t.} \quad & \frac{\sum w_p Z_{pj}}{\sum v_i X_{ij}} \leq 1 \\ & \frac{\sum u_r Y_{rj}}{\sum w_p Z_{pj}} \leq 1 \\ & v_i, w_p, u_r \geq 0\ ,\ j = 1, \dots, n \end{aligned}$$

$$E_{j_0} = \max\ \sum u_r Y_{rj_0} + \sum w_p Z_{pj_0} \qquad (2.5)$$

$$\begin{aligned} \text{s.t.} \quad & \sum v_i X_{ij_0} + \sum w_p Z_{pj_0} = 1 \\ & \sum w_p Z_{pj} - \sum v_i X_{ij} \leq 0 \\ & \sum u_r Y_{rj} - \sum w_p Z_{pj} \leq 0 \\ & v_i, w_p, u_r \geq 0\ ,\ j = 1, \dots, n \end{aligned}$$

Έστω (u_r^*, v_i^*, w_p^*) μια βέλτιστη λύση του (2.5) για την DMU j_0. Ο υπολογισμός της συνολικής αποδοτικότητας E_{j_0} και των επιμέρους σταδίων $E_{j_0}^1$ $και$ $E_{j_0}^2$ προκύπτουν ως εξής:

$$E_{j_0} = \sum w_p^* Z_{pj_0} + \sum u_r^* Y_{rj_0} \quad και \quad E_{j_0}^1 = \frac{\sum w_p^* Z_{pj_0}}{\sum v_i^* X_{ij_0}} \quad , \quad E_{j_0}^2 = \frac{\sum u_r^* Y_{rj_0}}{\sum w_p^* Z_{pj_0}}.$$

Παρόμοια με την περίπτωση του πολλαπλασιαστικού μοντέλου η ανάλυση της συνολικής αποδοτικότητας στις επιμέρους μπορεί να μην είναι μοναδική. Χρησιμοποιώντας τον έλεγχο της μοναδικότητας όπως στο πολλαπλασιαστικό μοντέλο, οι Chen et al. (2009) προτείνουν το παρακάτω μοντέλο για την μεγιστοποίηση της απόδοσης του πρώτου σταδίου:

$$E_{j_0}^{1max} = \max \sum w_p Z_{pj_0} \qquad (2.6)$$

$$\begin{aligned} \text{s.t.} \quad & \sum v_i X_{ij_0} = 1 \\ & \left(1 - E_{j_0}\right) \sum w_p Z_{pj_0} + \sum u_r Y_{rj_0} = E_{j_0} \\ & \sum w_p Z_{pj} - \sum v_i X_{ij} \leq 0 \\ & \sum u_r Y_{rj} - \sum w_p Z_{pj} \leq 0 \\ & v_i, w_p, u_r \geq 0\ ,\ j = 1, \dots, n \end{aligned}$$

Με την αποδοτικότητα του δευτέρου σταδίου να προκύπτει από την σχέση $E_{j_0}^{2min} = \frac{E_{j_0} - t_1^* \cdot E_{j_0}^{1max}}{t_2^*}$ όπου t_1^*, t_2^* τα βέλτιστα βάρη που προέκυψαν από το μοντέλο (2.5).

Εναλλακτικά, για την μεγιστοποίηση της απόδοσης του δευτέρου σταδίου $E_{j_0}^{2max}$ έχουμε:

$$E_{j_0}^{2max} = max\ \sum u_r Y_{rj_0} \qquad (2.7)$$

$$\begin{aligned} s.t. \quad & \sum w_p Z_{pj_0} = 1 \\ & \sum w_p Z_{pj_0} + \sum u_r Y_{rj_0} - E_{j_0} \sum v_i X_{ij_0} = E_{j_0} \end{aligned}$$

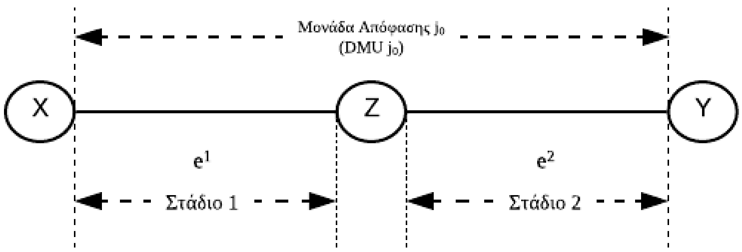
$$\sum w_p\, Z_{pj} - \sum v_i\, X_{ij} \leq 0$$
$$\sum u_r Y_{rj} - \sum w_p\, Z_{pj} \leq 0, \quad v_i, w_p, u_r \geq 0\ ,\ j = 1, \ldots, n$$

Και την αποδοτικότητα του πρώτου σταδίου να προκύπτει ως

$$E_{j_0}^{1min} = \frac{E_{j_0} - t_2^* \cdot E_{j_0}^{2max}}{t_1^*}.$$

2.3.3 Η μέθοδος του Αδύναμου Κρίκου (Despotis et al. 2016)

Στην μέθοδο του αδύναμου κρίκου οι Despotis et al. (2016) προτείνουν μια διαφορετική προσέγγιση για την εκτίμηση της αποδοτικότητας. Σε σύγκριση με τις προηγούμενες μεθόδους οι οποίες χρησιμοποιούν την ανάλυση της συνολικής αποδοτικότητας στις επιμέρους (decomposition approach) με κριτήριο την βελτιστοποίηση της συνολικής αποδοτικότητας του συστήματος, οι Despotis et al. (2016) προτείνουν τη συνθετική προσέγγιση (composition approach) όπου πρώτα βελτιστοποιείται η αποδοτικότητα του κάθε σταδίου ενώ ο υπολογισμός της συνολικής αποδοτικότητας του συστήματος γίνεται εκ των υστέρων.



Σχήμα 3: Εναλλακτική αναπαράσταση σε παραγωγικές διαδικασίες δύο σταδίων.

Οι αποδοτικότητες για το πρώτο και το δεύτερο στάδιο της DMU j_0 ορίζονται ως εξής:

$$e_{j_0}^1 = \frac{\sum w_p\, Z_{pj_0}}{\sum v_i X_{ij_0}} \quad \kappa\alpha\iota \quad e_{j_0}^{\,2} = \frac{\sum u_r Y_{rj_0}}{\sum w_p\, Z_{pj_0}}$$

Για την εκτίμηση της αποδοτικότητας των Μονάδων Απόφασης οι Despotis et al. (2016) προτείνουν το παρακάτω μοντέλο.

Οι αντικειμενικές συναρτήσεις στο πολυκριτήριο μοντέλο (2.8) εκφράζουν τις αποδοτικότητες της DMU j_0 για το πρώτο και δεύτερο στάδιο αντίστοιχα. Το μοντέλο αυτό στοχεύει στην από κοινού μεγιστοποίηση των αποδόσεων των δύο σταδίων και ως εκ τούτου στη μεγιστοποίηση της συνολικής αποδοτικότητας. Το μοντέλο (2.9) προέρχεται από το (2.8) μέσω του C–C μετασχηματισμού αναφορικά με την πρώτη αντικειμενική συνάρτηση.

$$\begin{aligned}
e_1 &= max\ \frac{\sum w_p Z_{pj_0}}{\sum v_i X_{ij_0}} \\
e_2 &= max\ \frac{\sum u_r Y_{rj_0}}{\sum w_p Z_{pj_0}} \qquad (2.8) \\
s.t.\quad & \sum w_p\, Z_{pj} - \sum v_i\, X_{ij} \leq 0 \\
& \sum u_r Y_{rj} - \sum w_p\, Z_{pj} \leq 0 \\
& v_i, w_p, u_r \geq \varepsilon\ ,\ j = 1, \ldots, n
\end{aligned}$$

$$\begin{aligned}
e_1 &= max\ \sum w_p\, Z_{pj_0} \\
e_2 &= max\ \frac{\sum u_r Y_{rj_0}}{\sum w_p Z_{pj_0}} \qquad (2.9) \\
s.t.\quad & \sum v_i\, X_{ij_0} = 1 \\
& \sum w_p\, Z_{pj} - \sum v_i\, X_{ij} \leq 0 \\
& \sum u_r Y_{rj} - \sum w_p\, Z_{pj} \leq 0 \\
& v_i, w_p, u_r \geq \varepsilon\ ,\ j = 1, \ldots, n
\end{aligned}$$

Σύμφωνα με την πολλαπλασιαστική μέθοδο η συνολική απόδοση του συστήματος ορίζεται ως το γινόμενο των αποδόσεων των επιμέρους σταδίων, $e_{j_0} = e^1_{j_0} \times e^2_{j_0}$. Εξ' ορισμού οι επιμέρους αποδοτικότητες είναι μικρότερες ή ίσες από τη μονάδα $e^1_{j_0} \leq 1 \; και \; e^2_{j_0} \leq 1$ συνεπώς και η συνολική αποδοτικότητα θα είναι μικρότερη ή ίση από την αποδοτικότητα των σταδίων $e_{j_0} \leq \min\{e^1_{j_0}, e^2_{j_0}\}$, με την ισότητα να ισχύει όταν κάποιο από τα επιμέρους στάδια ή όλα είναι αποδοτικά, δηλαδή $e^1_{j_0} = 1$ ή/και $e^2_{j_0} = 1$.

Η συγκεκριμένη ιδιότητα καθιστά το στάδιο με τη μικρότερη απόδοση καθοριστικό στον υπολογισμό της συνολικής απόδοσης του συστήματος και σύμφωνα με τους Despotis et al. (2016) αποτελεί και τον αδύναμο κρίκο του συστήματος.

Με στόχο την μεγιστοποίηση της απόδοσης του σταδίου (αδύναμου κρίκου) με τη χαμηλότερη αποδοτικότητα ώστε τελικά και η συνολική απόδοση του συστήματος να λάβει τη μέγιστη δυνατή τιμή οι Despotis et al. (2016) διατυπώνουν την μαθηματική αναπαράσταση της παραπάνω ιδέας ως:

$e_{j_0} = max\ [min\ \{ q_1 e^1_{j_0}\ , q_2 e^2_{j_0}\}]$, όπου (q_1, q_2) γνησίως θετικά βάρη. (2.10)

Σύμφωνα με το πολυκριτήριο μοντέλο (2.9) ο έλεγχος των επιμέρους αποδόσεων των δύο σταδίων πραγματοποιείται σε δύο φάσεις. Η Φάση Ι εντοπίζει ένα σημείο στο ανώτερο σύνορο της περιοχής των εφικτών λύσεων στον χώρο τον αντικειμενικών συναρτήσεων του (2.9) μέσω του (2.11) που μεγιστοποιεί την τιμή του αδύναμου κρίκου, ενώ στην Φάση ΙΙ παρέχεται η βέλτιστη λύση κατά Pareto.

Φάση Ι:

$$
\begin{aligned}
& \max\ \theta && (2.11)\\
\text{s.t.}\quad & \sum w_p Z_{pj_0} \geq \theta E^1_{j_0} \\
& \frac{\sum u_r Y_{rj_0}}{\sum w_p Z_{pj_0}} \geq \theta E^2_{j_0} \\
& \sum v_i X_{ij_0} = 1 \\
& \sum w_p Z_{pj} - \sum v_i X_{ij} \leq 0 \\
& \sum u_r Y_{rj} - \sum w_p Z_{pj} \leq 0 \\
& v_i, w_p, u_r \geq \varepsilon\ , \theta \geq 0\ ,\ j = 1, \dots, n
\end{aligned}
$$

Το μοντέλο (2.11) αποτελεί την κανονική μορφή του (2.10). Παρότι το μοντέλο (2.11) είναι μη γραμμικό, η παραμετρική του λύση μπορεί να επιτευχθεί σύμφωνα με τους Despotis et al. (2016) με τη μέθοδο της διχοτομικής αναζήτησης (bisection search) χρησιμοποιώντας τη μεταβλητή θ ως παράμετρο στο κλειστό διάστημα [0,1], αφού $0 \leq \theta \leq 1$. Οι λύσεις που προκύπτουν από το μοντέλο (2.11) είναι ασθενώς βέλτιστες κατά Pareto για το μοντέλο (2.9).

Για την επίλυση του μοντέλου οι Despotis et al. (2016) ορίζουν τις τιμές που μπορεί να λάβει η μεταβλητή θ, ως κάτω φράγμα το $\underline{\theta}$ και ως άνω φράγμα το $\overline{\theta}$. Αρχικά, θέτουν $\underline{\theta} = 0$ για το οποίο οι περιορισμοί του (2.11) είναι συνεπείς και $\overline{\theta} = 1 + \varepsilon$ για το οποίο οι περιορισμοί δεν είναι συνεπείς, όπου ε ένας πολύ μικρός θετικός αριθμός. Στην συνέχεια ελέγχεται η συνέπεια των περιορισμών για $\theta' = \frac{(\underline{\theta}+\overline{\theta})}{2}$. Στην περίπτωση που είναι συνεπείς το θ′ αντικαθιστά το $\underline{\theta}$ διαφορετικά αντικαθιστά το $\overline{\theta}$. Ο έλεγχος συνεχίζεται έως ότου οι τιμές και των δύο φραγμάτων φτάσουν αρκετά κοντά.

Υποθέτοντας μια βέλτιστη λύση $(\theta^*, v^*_i, w^*_p, u^*_r)$ του (2.11) οι αποδοτικότητες των σταδίων υπολογίζονται ως:

$e_{j_0}^{1*} = \frac{\sum w_p^* Z_{pj_0}}{\sum v_i^* X_{ij_0}} = \sum w_p^* Z_{pj_0}$ και $e_{j_0}^{2*} = \frac{\sum u_r^* Y_{rj_0}}{\sum w_p^* Z_{pj_0}}$.

Όπως προαναφέρθηκε οι λύσεις του (2.11) είναι ασθενώς βέλτιστες κατά Pareto για το μοντέλο (2.9). Σύμφωνα με τους Despotis et al. (2016) μια βέλτιστη λύση για το (2.9) επιτυγχάνεται με την Φάση II.

Φάση II:

$$
\begin{aligned}
& \max s_1 + s_2 \qquad (2.12) \\
\text{s.t.} \quad & \sum w_p Z_{pj_0} - s_1 = e_{j_0}^{1*} \\
& \sum u_r Y_{rj_0} - s_2 \sum w_p^* Z_{pj_0} - e_{j_0}^{2*} \sum w_p Z_{pj_0} = 0 \\
& \sum v_i X_{ij_0} = 1 \\
& \sum w_p Z_{pj} - \sum v_i X_{ij} \le 0 \\
& \sum u_r Y_{rj} - \sum w_p Z_{pj} \le 0 \\
& 0 \le s_1 \le E_{j_0}^1, 0 \le s_2 \le E_{j_0}^2 \\
& v_i, w_p, u_r \ge \varepsilon, \quad j = 1, \dots, n
\end{aligned}
$$

Αν η λύση που προκύπτει από τη Φάση I είναι ασθενώς βέλτιστη κατά Pareto, στη βέλτιστη λύση της Φάσης II το πολύ μία από τις μεταβλητές $\hat{s}_1$ και $\hat{s}_2$ θα είναι γνησίως θετική. Στην περίπτωση που $\hat{s}_1 = \hat{s}_2 = 0$ τότε οι δύο φάσεις παράγουν την ίδια βέλτιστη κατά Pareto λύση.
Η λύση $(\hat{v}_i, \hat{w}_p, \hat{u}_r)$ που προκύπτει από το μοντέλο (2.12) είναι βέλτιστη κατά Pareto για το μοντέλο (2.9). Οι αποδόσεις των σταδίων υπολογίζονται από τις σχέσεις:

$\hat{e}_{j_0}^1 = \frac{\sum \hat{w}_p Z_{pj_0}}{\sum \hat{v}_i X_{ij_0}} = \sum \hat{w}_p Z_{pj_0}$ $\textit{και}$ $\hat{e}_{j_0}^2 = \frac{\sum \hat{u}_r Y_{rj_0}}{\sum \hat{w}_p Z_{pj_0}}$

ενώ η συνολική αποδοτικότητα προκύπτει ως $\hat{e}_{j_0}^0 = min\{\hat{e}_{j_0}^1, \hat{e}_{j_0}^2\}$.

Κεφάλαιο 3

Η ανάπτυξη του προγράμματος αυτοματοποιημένης ανάλυσης αποδοτικότητας για την αξιολόγηση παραγωγικών μονάδων πραγματοποιήθηκε με την χρήση του Microsoft Excel και των συμβατικών λειτουργιών που προσφέρει καθώς και την επέκταση αυτών μέσω της γλώσσας προγραμματισμού Visual Basic For Applications (VBA) που είναι ενσωματωμένη στο Excel. Η επίλυση των μοντέλων βασίζεται στην χρήση του Solver που εμπεριέχεται στο Excel σαν πρόσθετο πρόγραμμα (Solver Add-in). Πρόκειται για ένα εργαλείο το οποίο χρησιμοποιεί τεχνικές της επιχειρησιακής έρευνας για την εύρεση βέλτιστων λύσεων σε προβλήματα απόφασης.

Οι απαιτήσεις για την χρήση του προγράμματος περιορίζονται στην ύπαρξη του λογισμικού του MS Excel στον υπολογιστή του χρήστη και την εισαγωγή του Solver Add-in. Πέραν αυτού έχει δημιουργηθεί ένα πρόσθετο πρόγραμμα Add-in (NetworkDEAadd-in) καθώς και ένα αρχείο Microsoft Office ExportedUI Customization τα οποία περιέχουν τις απαραίτητες διαδικασίες για την επίλυση των μοντέλων.

3.1 Εργαλεία και Τεχνολογίες

Microsoft Visual Basic for Applications (VBA)

Η Microsoft Visual Basic for Applications (VBA) χρησιμοποιείται σε συνδυασμό με το MS Excel για την δημιουργία αυτοματοποιημένων εργαλείων. Τα προγράμματα σε VBA αναφέρονται επίσης ως Excel Macros, VBA Macros ή απλά Macros. Η VBA ελέγχει το MS Excel μέσω δημιουργίας εκτελέσιμων διαδικασιών στον VBA Editor που περιέχεται στο Developer Tab του Excel. Ο VBA Editor είναι ένα πρόγραμμα επεξεργασίας VBA το οποίο δίνει την δυνατότητα δημιουργίας, επεξεργασίας, εντοπισμού σφαλμάτων (debugging) και εκτέλεσης κώδικα VBA. Ο κώδικας αυτός είναι οργανωμένος μέσα στον Editor σαν VBAProject. Κάθε project τέτοιου τύπου ορίζεται σαν μία συλλογή από υπομονάδες, φόρμες χρήστη και διαφόρων άλλων προγραμματιστικών στοιχείων. Το πρόγραμμα για την επίλυση των μοντέλων έχει αναπτυχθεί ως VBA Project αποτελούμενο από διαφορετικές υπομονάδες (modules) με συναρτήσεις και υπορουτίνες (functions and subroutines) για την αυτοματοποίηση της επίλυσης των προβλημάτων καθώς και από φόρμες χρήστη (UserForms) για την εισαγωγή των δεδομένων προς ανάλυση.

Solver

Ο Solver ως επιπρόσθετο πρόγραμμα του Excel αποτελεί σημαντικό κομμάτι στην επίλυση προβλημάτων γραμμικού προγραμματισμού. Η χρήση του επιτρέπει την εύρεση της βέλτιστης τιμής της αντικειμενικής συνάρτησης που υπόκειται σε συγκεκριμένους περιορισμούς μέσω των μεταβλητών απόφασης. Προσαρμόζοντας τις τιμές των μεταβλητών απόφασης με τρόπο κατά τον οποίο να ικανοποιούνται τα όρια των περιορισμών δημιουργείται το ζητούμενο (μεγιστοποίηση ή ελαχιστοποίηση) για την αντικειμενική συνάρτηση. Οι μέθοδοι επίλυσης που προσφέρει διαφέρουν ως προς την ομαλότητα και την γραμμικότητα των προβλημάτων προς επίλυση ως εξής:

- **Μη γραμμική γενικευμένη μειωμένη διαβάθμιση (GRG)**
  για ομαλά, μη γραμμικά προβλήματα.
- **LP Simplex** για γραμμικά προβλήματα.
- **Evolutionary** για μη-ομαλά προβλήματα.

Ο Excel Solver περιορίζει το ανώτατο όριο του αριθμού των μεταβλητών απόφασης και των περιορισμών κάθε μοντέλου. Το όριο για τις μεταβλητές απόφασης είναι 200, ενώ το όριο των περιορισμών εξαρτάται από την γραμμικότητα του μοντέλου και την μορφή των περιορισμών. Με την χρήση της βασικής έκδοσης του Solver οι περιορισμοί του μοντέλου μπορούν να καλύψουν έως 100 κελιά τα οποία δεν αποτελούν μεταβλητές απόφασης. Συγκεκριμένα, τα μοντέλα των Kao and Hwang (2008) και Chen et al. (2009) απαιτούν για την λύση τους 2 περιορισμούς για κάθε μονάδα απόφασης του μοντέλου και ένα ακόμα περιορισμό για την μονάδα απόφασης υπό ανάλυση ενώ το μοντέλο των Despotis et al. (2016) απαιτεί τους 2 περιορισμούς για κάθε μονάδα απόφασης και επιπλέον 3 περιορισμούς για την μονάδα απόφασης υπό ανάλυση. Σύμφωνα με τα παραπάνω τα μοντέλα περιορίζονται σε λύσεις προβλημάτων με μέγιστο αριθμό Μονάδων Απόφασης τις 48.

Η χρήση του Solver μέσω της VBA προϋποθέτει την συμπλήρωση απαιτούμενων πεδίων για τον προσδιορισμό του προβλήματος προς επίλυση. Οι συναρτήσεις και η ερμηνεία τους έχουν όπως παρακάτω:

Συνάρτηση	**Ερμηνεία**
SolverReset	Επαναφορά του Solver στις αρχικές ρυθμίσεις. Χρησιμοποιείται για τον εκ νέου προσδιορισμό των αντικειμενικών συναρτήσεων, περιορισμών και μεταβλητών απόφασης.
SolverOptions	Επιπρόσθετες επιλογές για την επίλυση του μοντέλου. (MaxTime, Iterations, Precision, AssumeLinear, StepThru, Estimates, Derivatives, SearchOption, IntTolerance, Scaling, Convergence, AssumeNonNeg etc.)
SolverOk	Προσδιορισμός βασικού μοντέλου.
SetCell MaxMinVal ValueOf ByChange Engine EngineDesc	Αναφορά στο κελί της αντικειμενικής συνάρτησης. 1: Maximize, 2: Minimize, 3: Προσδιορισμός συγκεκριμένης τιμής Προαιρετικό για την τιμή 3 του MaxMinVal. Εύρος κελιών για τις μεταβλητές απόφασης. 1: GRG Nonlinear, 2: Simplex LP, 3: Evolutionary Εναλλακτικός τρόπος επιλογής μεθόδου επίλυσης "Simplex LP", "GRG Nonlinear", ή "Evolutionary".
SolverAdd	Προσθήκη περιορισμού του μοντέλου.
CellRef Relation FormulaText	Αναφορά στο κελί της αριστερής πλευράς του περιορισμού 1: <=, 2: =, 3: >= Τιμή της δεξιάς πλευράς του περιορισμού
SolverSolve	Εκκίνηση επίλυσης
UserFinish	True: Επιστροφή αποτελεσμάτων χωρίς πλαίσιο διαλόγου. False: Επιστροφή αποτελεσμάτων και παρουσίαση πλαισίου διαλόγου.
SolverFinish	Διαχείριση αποτελεσμάτων και δημιουργία αναφοράς.
KeepFinal	1: Διατήρηση λύσεων στα κελιά απόφασης και αντικατάσταση τιμών. 2: Απόρριψη λύσεων και επαναφορά αρχικών τιμών.

3.2 Ανάπτυξη Λειτουργιών

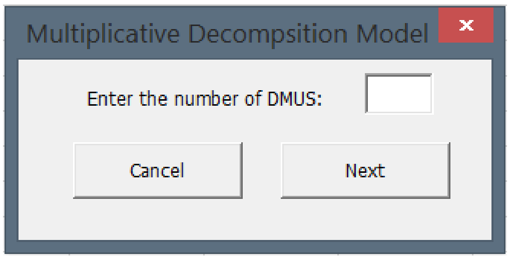
Το πρόγραμμα αναπτύχθηκε μέσω χρήσης υπομονάδων (modules) και φορμών χρήστη (UserForms). Οι υπομονάδες εμπεριέχουν συναρτήσεις και υπορουτίνες σε VBA οι οποίες εκπληρώνουν τις βασικές λειτουργίες του προγράμματος. Για την αποτελεσματική λειτουργία τους είναι απαραίτητη η αλληλεπίδραση του χρήστη με το πρόγραμμα ώστε να ορίσει τις προδιαγραφές σύμφωνα με τις οποίες θα λειτουργήσει. Παρακάτω περιγράφεται αναλυτικά η ανάπτυξη του προγράμματος με την χρήση UserForms και Modules.

3.2.1 Εκκίνηση Εφαρμογής

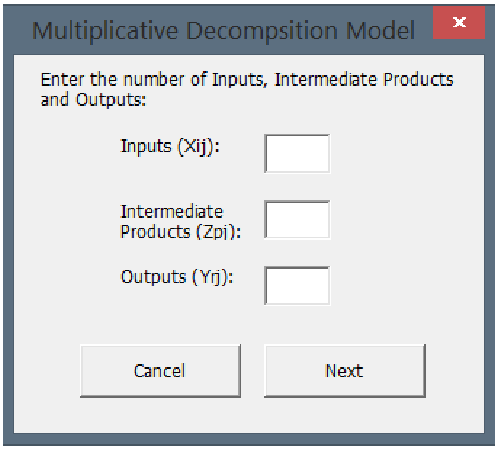
Η επικοινωνία του χρήστη με την εφαρμογή αποτελεί ιδιαιτέρως σημαντικό κομμάτι για τον προσδιορισμό του προβλήματος προς επίλυση. Η ανάπτυξη της διεπαφής χρήστη (UI) βασίστηκε στην δημιουργία φορμών χρήστη (UserForms) οι οποίες είναι ελεγχόμενες μέσω VBA. Οι φόρμες χρήστη που δημιουργήθηκαν αποτελούν παράθυρα ή πλαίσια διαλόγου τα οποία επιτρέπουν την εισαγωγή δεδομένων, επιλογή τιμών από λίστες καθώς και την προβολή ενημερωτικών μηνυμάτων.

Με στόχο την δημιουργία μίας εύχρηστης διεπαφής χρήστη η εφαρμογή ζητά όσο το δυνατών λιγότερες πληροφορίες ώστε να αποφευχθούν τυχόν λάθη από πλευράς χρήστη. Επιλέγοντας οποιοδήποτε μοντέλο από το Solver Models tab εμφανίζονται διαδοχικά οι δύο πρώτες φόρμες για την εισαγωγή των χαρακτηριστικών του μοντέλου προς επίλυση. Τα ζητούμενα αφορούν αποκλειστικά τον αριθμό των μονάδων απόφασης καθώς και τον αριθμό εισροών (X_{ij}), ενδιάμεσων μεγεθών (Z_{pj}) και εκροών (Y_{rj}).

Σημειώνεται ότι σε κάθε βήμα συμπλήρωσης των πεδίων κάθε φόρμας υπάρχουν οι κατάλληλοι έλεγχοι επαλήθευσης των τιμών που εισάγονται σύμφωνα με τους περιορισμούς των μοντέλων και του Solver. Επίσης, οι πρώτες φόρμες που εμφανίζονται είναι σχεδιασμένες ως vbModal και η αλληλεπίδραση του χρήστη με την εφαρμογή του MS Excel δεν είναι δυνατή.

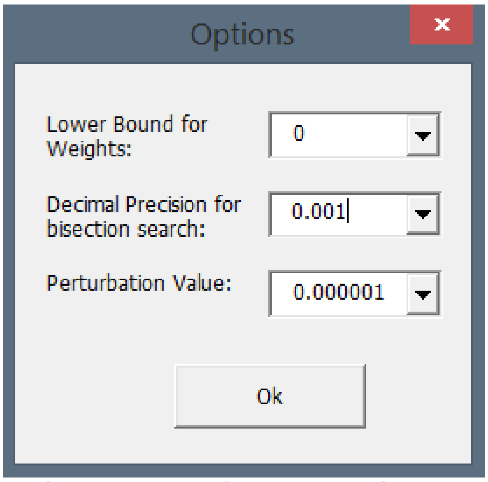


Εικόνα 3.1: Εισαγωγή αριθμού Μονάδων Απόφασης



Εικόνα 3.2: Εισαγωγή αριθμού εισροών, ενδιάμεσων μεγεθών και εκροών

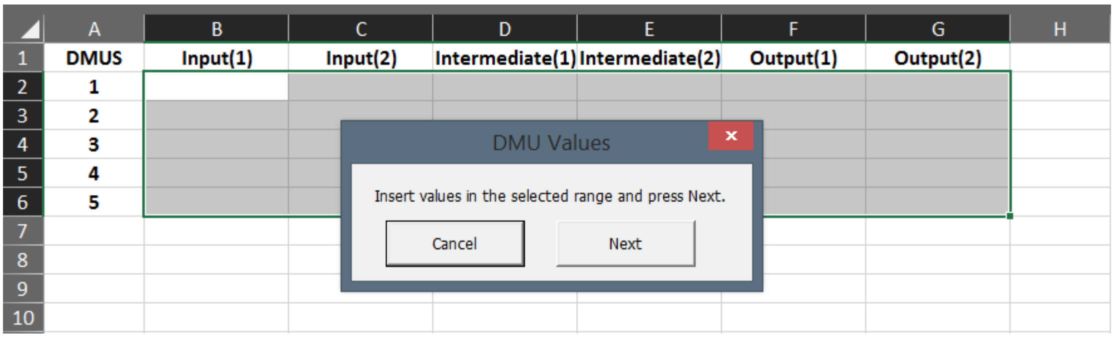
Για την μέθοδο του αδύναμου κρίκου απαιτείται η συμπλήρωση μιας επιπλέον φόρμας με τις πρόσθετες επιλογές που απαιτούνται για τον καθορισμό του μοντέλου. Συγκεκριμένα δίνεται η επιλογή του κατώτατου ορίου των βαρών ($v_i, w_p, u_r \geq \varepsilon$), η επιλογή ακρίβειας και η τιμή διαταραχής (Perturbation value) για τη μέθοδο της διχοτομικής αναζήτησης. Η διχοτομική αναζήτηση, κατά την Φάση Ι της επίλυσης του μοντέλου των Despotis et al. (20016), γίνεται επαναληπτικά και οι αποδοτικότητες των μονάδων απόφασης που θα προκύψουν εξαρτώνται σημαντικά από τις τιμές που θα επιλεχθούν.



Εικόνα 3.3: Επιλογές για το μοντέλο του Αδύναμου κρίκου

Τα πεδία τιμών στις λίστες της φόρμας είναι προεπιλεγμένα σε προτεινόμενες τιμές. Το εύρος των πεδίων μπορεί να μεταβληθεί αναλόγως με τις ανάγκες του χρήστη.

Με την επιτυχή συμπλήρωση των παραπάνω πεδίων εμφανίζεται η τελική φόρμα για την εισαγωγή των δεδομένων στον χώρο που έχει προκαθοριστεί στο Worksheet του Excel.



Εικόνα 3.4: Εισαγωγή δεδομένων στο Worksheet για πρόβλημα με 5 Μονάδες Απόφασης με δύο εισροές, δύο ενδιάμεσα μεγέθη και δύο εκροές

Η εισαγωγή των δεδομένων αποτελεί το τελικό βήμα από πλευράς χρήστη και η διαχείριση των προβλημάτων πραγματοποιείται στο αντίστοιχο module του μοντέλου που έχει επιλεγεί.

3.2.2 Επίλυση Μοντέλων

Στις υπομονάδες (Modules) του VBAProject βρίσκεται η βασική λειτουργία του προγράμματος. Κάθε μοντέλο έχει το δικό του module με τον κώδικα που αφορά το μοντέλο και την υλοποίηση του. Ακόμα συμπεριλαμβάνεται ένα module με κώδικα που αφορά διαδικασίες του προγράμματος οι οποίες είναι κοινές για όλα τα μοντέλα. Οι διαδικασίες για την επίλυση των μοντέλων περιγράφονται περιληπτικά παρακάτω με ενδεικτικά παραδείγματα της χρήσης του Solver μέσω VBA ενώ μια αναλυτικότερη παρουσίαση παρατίθεται στο Παράρτημα της εργασίας.

Ο υπολογισμός των μοντέλων πραγματοποιείται με την χρήση τριών Worksheets. Τα δεδομένα τα οποία εισάγει ο χρήστης βρίσκονται στο Data Sheet ενώ για την επίλυση του μοντέλου δημιουργείται ένα επιπλέον Worksheet ως Calculations καθώς και ένα Result Sheet με τα τελικά αποτελέσματα. Η επίλυση των μοντέλων γίνεται στο Calculations Sheet το οποίο δημιουργείται εξαρχής για κάθε μοντέλο. Σε αυτό δημιουργούνται οι αντικειμενικές συναρτήσεις, οι περιορισμοί και οι μεταβλητές απόφασης όπως ορίζονται από κάθε μοντέλο. Το συγκεκριμένο Sheet λειτουργεί ως συνδετικός κρίκος μεταξύ του Data Sheet και Result Sheet, παραμένει κρυφό από τον χρήστη καθ' όλη την διάρκεια των υπολογισμών και διαγράφεται πριν την τελική παρουσίαση των αποτελεσμάτων.

Για κάθε μοντέλο οι συναρτήσεις που περιέχονται στο Calculations Sheet για τις μεταβλητές αλλάζουν ενώ προστίθενται ή αφαιρούνται περιορισμοί ανάλογα με τις προδιαγραφές του. Με την χρήση των μεταβλητών που ορίζονται για κάθε μοντέλο και την λειτουργία του Solver μέσω VBA όπως παρουσιάστηκε παραπάνω παρουσιάζονται δύο παραδείγματα για την εύρεση της αποδοτικότητας.

Module Επίλυσης Πολλαπλασιαστικού Μοντέλου

Για την επίλυση του μοντέλου των Kao et al. (2008) δημιουργείται μια επαναληπτική διαδικασία για τον αριθμό των μονάδων απόφασης προς επίλυση (numberOfDMUs). Η επίλυση του μοντέλου γίνεται αποκλειστικά στο Calculations Sheet (Worksheets(Dmus)). Για κάθε μία από τις μονάδες αυτές ορίζονται εξαρχής η αντικειμενική συνάρτηση (objectiveValue) και ο περιορισμός $\sum v_i X_{ij_0} = 1$ (stVX).

```
For i = 0 To numberOfDMUS

    Worksheets(Dmus).Activate

    objectiveValue = uYi.Offset(i, 0).Address
    stVX = vXi.Offset(i, 0).Address
```

Σε κάθε επανάληψη γίνεται επανεκκίνηση του Solver για τον προσδιορισμό της νέας αντικειμενικής συνάρτησης και των περιορισμών του μοντέλου. Θέτοντας Assumenonneg:=True οι μεταβλητές απόφασης για το μοντέλο θα έχουν πάντα θετικές τιμές χωρίς να απαιτείται να συμπεριληφθούν στους περιορισμούς.

```
    SolverReset
    SolverOptions Assumenonneg:=True
    SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
      Engine:=2, EngineDesc:="Simplex LP"

    SolverAdd CellRef:=stVX, Relation:=2, FormulaText:="1"
    SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"
    SolverAdd CellRef:=conC, Relation:=1, FormulaText:="0"

    SolverSolve (True)
    SolverFinish KeepFinal:=1
```

Με την επίλυση του προβλήματος οι τιμές της αντικειμενικής συνάρτησης και των πολλαπλασιαστών που προέκυψαν αποθηκεύονται στο ResultSheet για την τελική παρουσίαση.

```
    destinationObjectiveValue.Offset(i, 0).Value = uYi.Offset(i, 0).Value
    multiplierRng.Copy destinationMultiplierRange.Offset(i, 0)

    Application.CutCopyMode = False

  Next
```

Module Επίλυσης της Μεθόδου του Αδύναμου Κρίκου

Για την επίλυση του μοντέλου του αδύναμου κρίκου σε κάθε επανάληψη του Solver γίνεται αρχικοποίηση των τιμών του θ, θmax και θmin. Πέραν των objectiveValue και stVX που αναφέρονται στην αντικειμενική συνάρτηση και τον περιορισμό του $\sum v_i X_{ij_0} = 1$, προστίθενται οι επιπλέον περιορισμοί stWZE1, stWZE2 και stUY όπως ορίζονται στην εργασία των Despotis et al. (2016).

```
For i = 0 To numberOfDMUS

    Worksheets(Dmus).Activate

    θmax = 1
    θmin = 0
    θ = (θmax + θmin) / 2

    objectiveValue = vXi.Offset(i, 0).Address
    stVX = vXi.Offset(i, 0).Address
    stWZE1 = wke1.Offset(i, 0).Address
    stWZE2 = wke2.Offset(i, 0).Address
    stUY = uYi.Offset(i, 0).Address
```

Για το συγκεκριμένο μοντέλο απαιτείται η δημιουργία μιας ακόμα επαναληπτικής διαδικασίας για την διχοτομική αναζήτηση. Ο Solver θα επιλύει επαναλαμβανόμενα για την ίδια μονάδα απόφασης έως ότου η διαφορά των θmax και θmin φτάσει σε έναν αρκετά μικρό αριθμό (dp) όπως έχει οριστεί από τον χρήστη.

```
    Do While (θmax - θmin) >= dp And θ <= 1

        SolverReset
        SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
        Engine:=2, EngineDesc:="Simplex LP"

        SolverAdd CellRef:=stWZE1, Relation:=3, FormulaText:="=" & θ
        SolverAdd CellRef:=stUY, Relation:=3, FormulaText:="=" & θ & "*" & stWZE2

        SolverAdd CellRef:=stVX, Relation:=2, FormulaText:="=1"
        SolverAdd CellRef:=conA, Relation:=1, FormulaText:="0"
        SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"
        'weights >= ε
        SolverAdd CellRef:=v, Relation:=3, FormulaText:="=" & ε
        SolverAdd CellRef:=w, Relation:=3, FormulaText:="=" & ε
        SolverAdd CellRef:=u, Relation:=3, FormulaText:="=" & ε
        SolverAdd CellRef:=θ, Relation:=3, FormulaText:="0"

        Results = SolverSolve(True)
        SolverFinish KeepFinal:=1

        Counter = Counter + 1
```

Η διχοτομική αναζήτηση επιτυγχάνεται μέσω μιας τιμής που επιστρέφει ο Solver σχετικά με τα αποτελέσματα των λύσεων. Οι τιμές που θα μας απασχολήσουν για την επίλυση είναι η τιμή 0 και 5 τις οποίες επιστρέφει ο Solver σε περίπτωση εύρεσης λύσης και αδυναμία εύρεσης λύσης αντίστοιχα. Σε διαφορετική περίπτωση χρησιμοποιείται η μεταβλητή διαταραχής (p) όπως ορίστηκε από τον χρήστη.

```
    Select Case Results
        Case 0
          θmin = θ
          θ = (θmax + θmin) / 2
        Case 5
          θmax = θ
          θ = (θmax + θmin) / 2
        Case Else
          θ = θ + p
        End Select
```

Ο Counter χρησιμοποιείται εξολοκλήρου για την καταμέτρηση των επαναλήψεων του Solver και την ενημέρωση του χρήστη σε σταθερά χρονικά διαστήματα (ανά 200 επαναλήψεις). Ο χρήστης ενημερώνεται για την εξέλιξη της επίλυσης μέχρι εκείνη την στιγμή και αν επιθυμεί μπορεί να τερματίσει την διαδικασία.

```
      If Counter Mod 200 = 0 Then
        If MsgBox("Solver has not found a complete solution yet." & vbCrLf & " Do you wish to
continue the process?", _
        vbQuestion + vbYesNo + vbDefaultButton1, "Solution Found For: " & i & " of " & numDmu & "
DMUS.") _
          = vbNo Then BreakCode
      End If

    Loop
```

Αν ικανοποιηθούν όλοι οι παραπάνω περιορισμοί, γίνεται αποθήκευση των μεταβλητών στο ResultSheet και ξεκινάει η επίλυση για την επόμενη μονάδα απόφασης.

```
    destinationObjectiveValue.Offset(i, 0).Value = θ
    destinationE1Value.Offset(i, 0).Value = wZi.Offset(i, 0).Value
    destinationE2Value.Offset(i, 0).Value = uywz.Offset(i, 0).Value

    Application.CutCopyMode = False

  Next
```

Κεφάλαιο 4

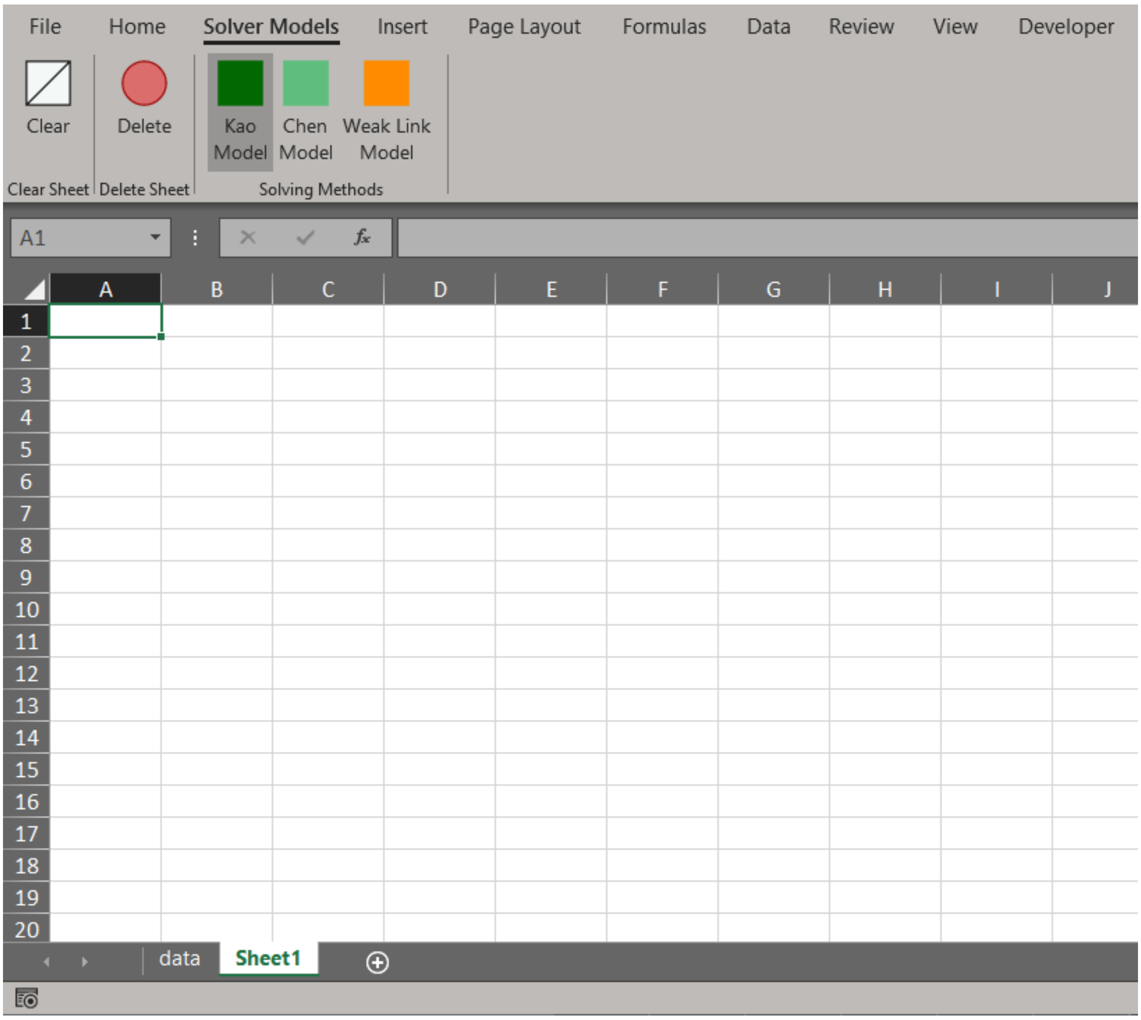
Σε αυτό το κεφάλαιο θα γίνει παρουσίαση της λειτουργίας της αυτοματοποιημένης διαδικασίας αποτίμησης της αποδοτικότητας. Ως παράδειγμα θα χρησιμοποιηθούν συνθετικά δεδομένα για 30 Μονάδες Απόφασης με δύο εισροές, δύο εκροές και δύο ενδιάμεσα μεγέθη.

DMUS	X1	X2	Z1	Z2	Y1	Y2
1	69,5	68,6	56,6	84,4	48,7	62,8
2	40,2	66,2	88	47,2	85,8	28,3
3	81,3	89,8	44,4	18,4	38,3	20,7
4	55	97,9	28,7	41,6	38,2	10,3
5	56,2	59,1	26,5	52,7	44,2	17,4
6	64,8	64,4	14,7	70,5	86,6	22,9
7	79,2	68,1	63,5	39,3	47,6	35
8	36	74,3	66,6	57,4	40,3	94,8
9	10,8	10,3	46,5	47,9	57,5	95,2
10	17,7	93,6	35,9	58,7	45,9	12
11	38,8	97,5	55,2	41,7	60,5	82,7
12	60,9	96,4	86	28,9	93,1	72,3
13	70,3	45,8	65,3	35,3	34,3	98,8
14	20,5	75,6	13,1	60	53,3	18,3
15	17,9	74,8	54,2	66,7	52,1	15,8
16	51,8	19,8	52,3	74,2	73,6	84,7
17	11,3	27,3	42,7	72,3	68,9	37,4
18	58,7	42	95,9	26,6	51,6	96,4
19	41,4	51,6	83	75,4	20,5	72
20	99,7	87,1	87,5	96,9	58,6	39
21	25,6	14,6	52	19,1	44,3	51,3
22	65,1	97,3	79,4	68	53,8	55,5
23	40,4	33	74,5	21,7	13,9	55,7
24	19,4	20,1	77,5	74,1	60,9	71
25	54,2	99,3	20,8	69,9	47,8	12,2
26	80,1	27,5	51,3	95,8	21,7	12,6
27	82,9	38,1	43,3	75,3	16,8	26,6
28	98,6	81,8	93,8	15,9	40,3	35,8
29	77,3	40,3	95,6	52,5	96,1	44,2
30	38,6	58,3	37,8	66,1	16	69,9

Εικόνα 4.1: Συνθετικά Δεδομένα (πηγή: Despotis et al.(2016b))

4.1 Kao Model

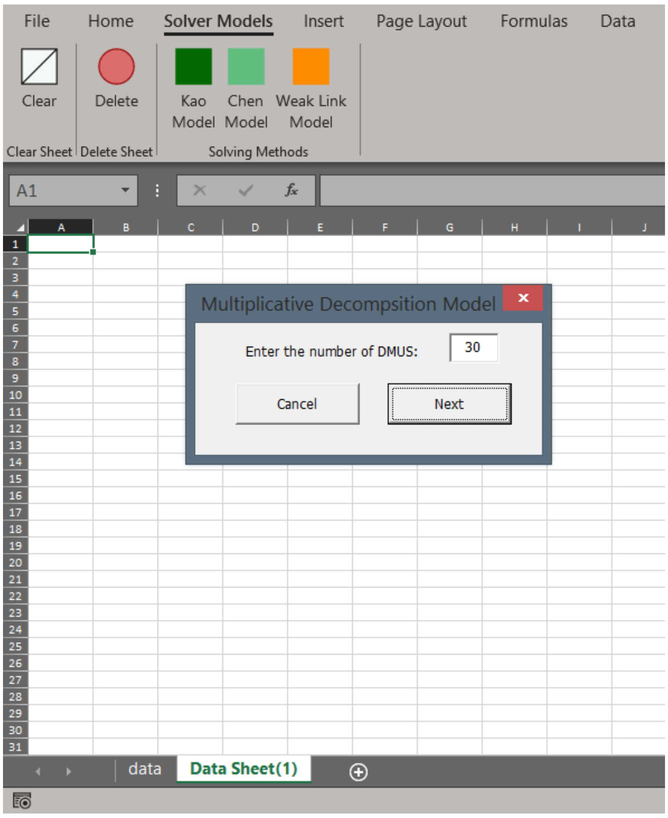
Με την είσοδο στο Excel ο χρήστης μπορεί να επιλέξει ένα από τα προτεινόμενα μοντέλα από το tab Solver Models.



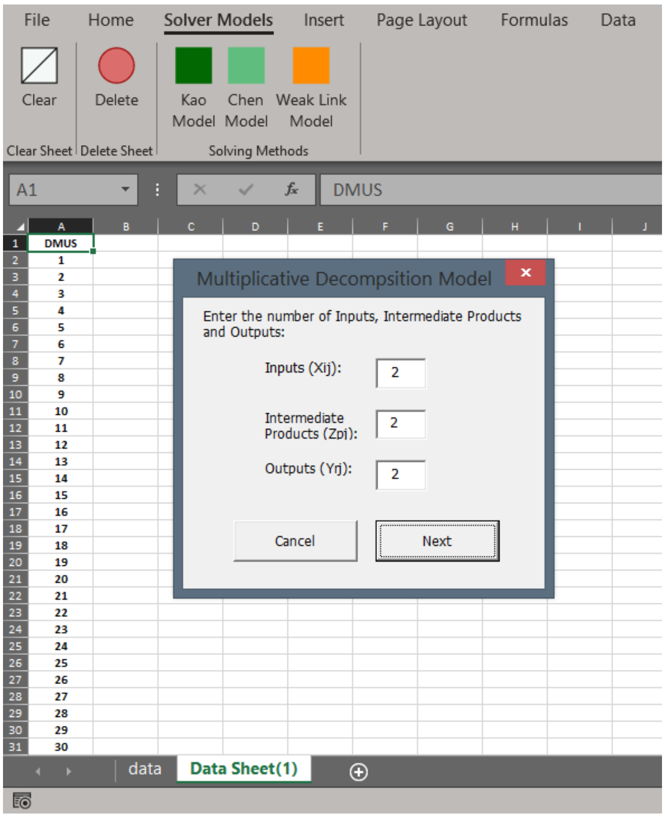
Εικόνα 4.2: Επιλογή μοντέλου σε κενό Worksheet

Επιλέγοντας το Kao Model από το group Solving Methods γίνεται εκκίνηση της εφαρμογής και ο χρήστης εισάγει τις απαραίτητες πληροφορίες για την δημιουργία του μοντέλου.

Για κάθε φόρμα που συμπληρώνεται από τον χρήστη το ενεργό Worksheet ενημερώνεται αναλόγως με την πληροφορία που έχει εισαχθεί.

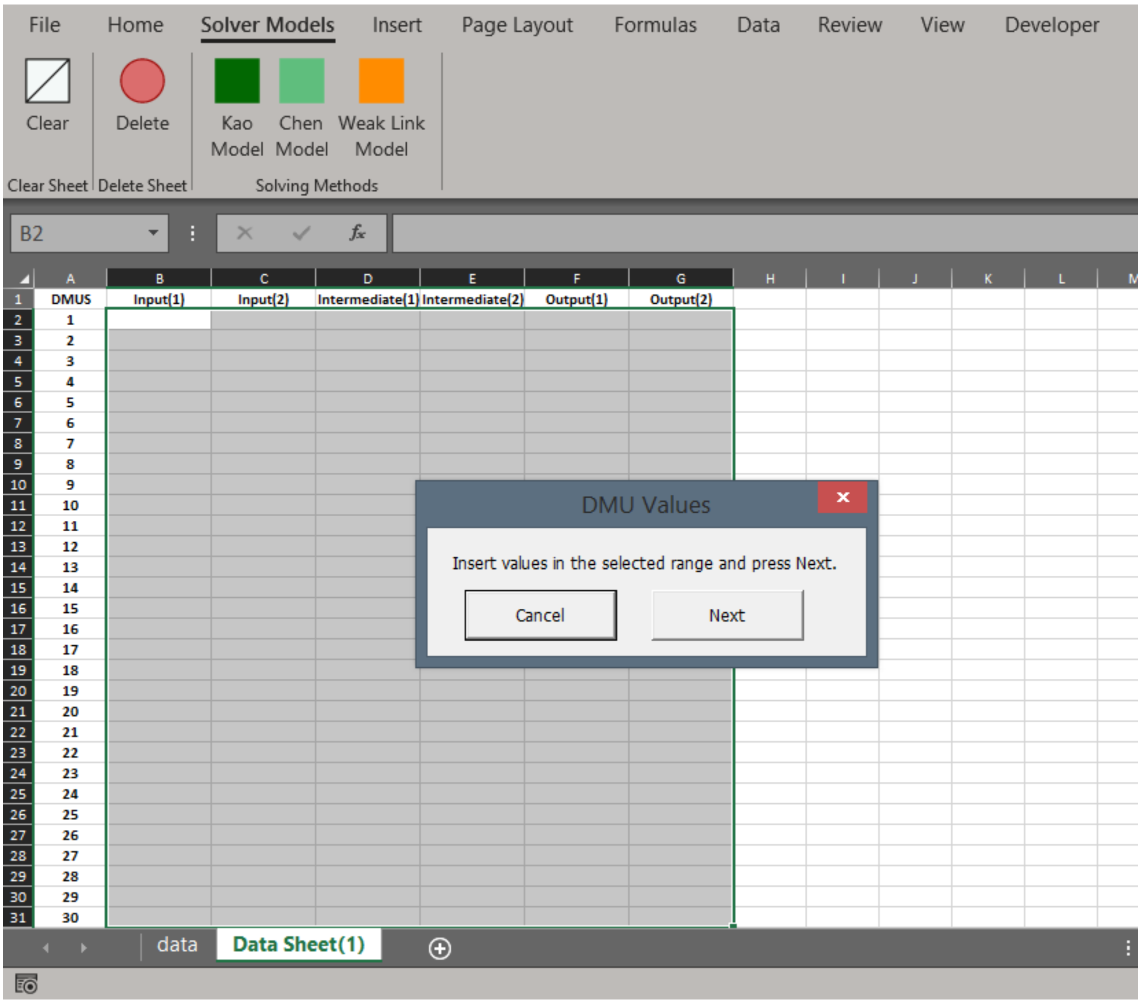


Εικόνα 4.3: Εισαγωγή αριθμού Μονάδων Απόφασης

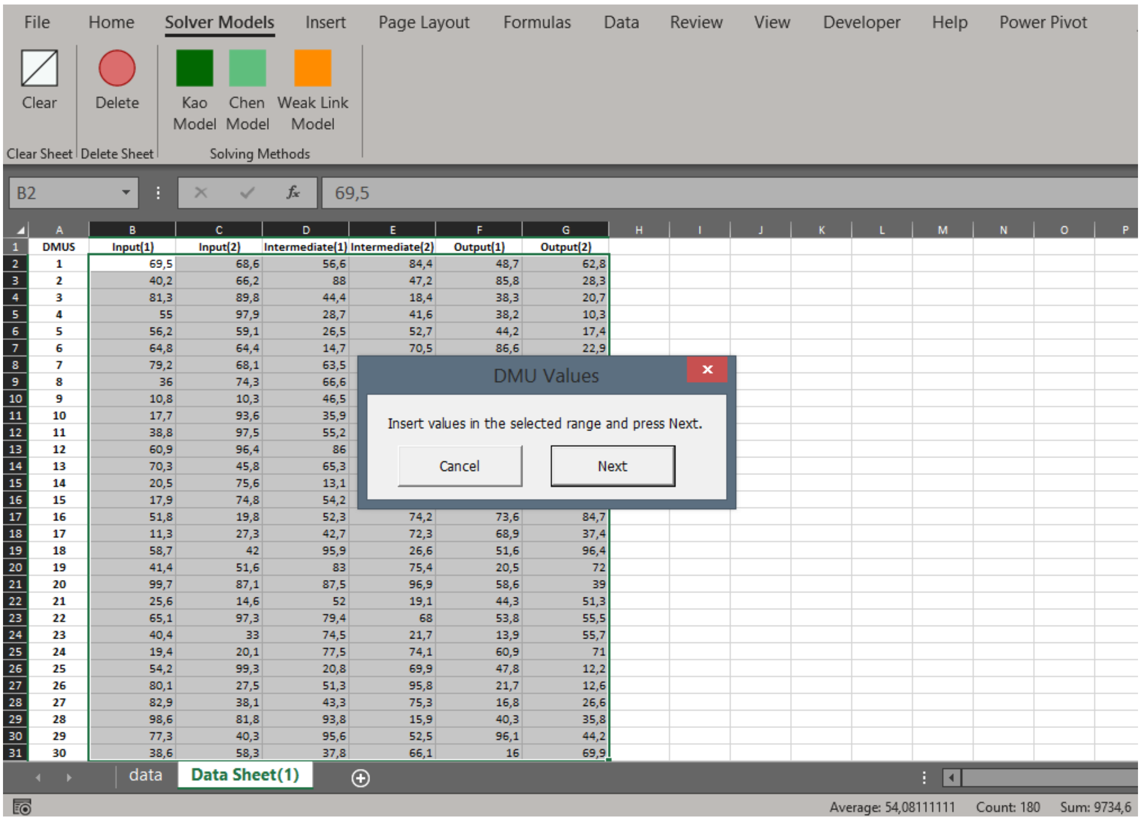


Εικόνα 4.4: Εισαγωγή αριθμού Εισροών, Ενδιάμεσων Μεγεθών και Εκροών

Με την συμπλήρωση των απαιτούμενων πεδίων ο χρήστης μπορεί να εισάγει τα δεδομένα προς ανάλυση στον προκαθορισμένο χώρο του Worksheet.

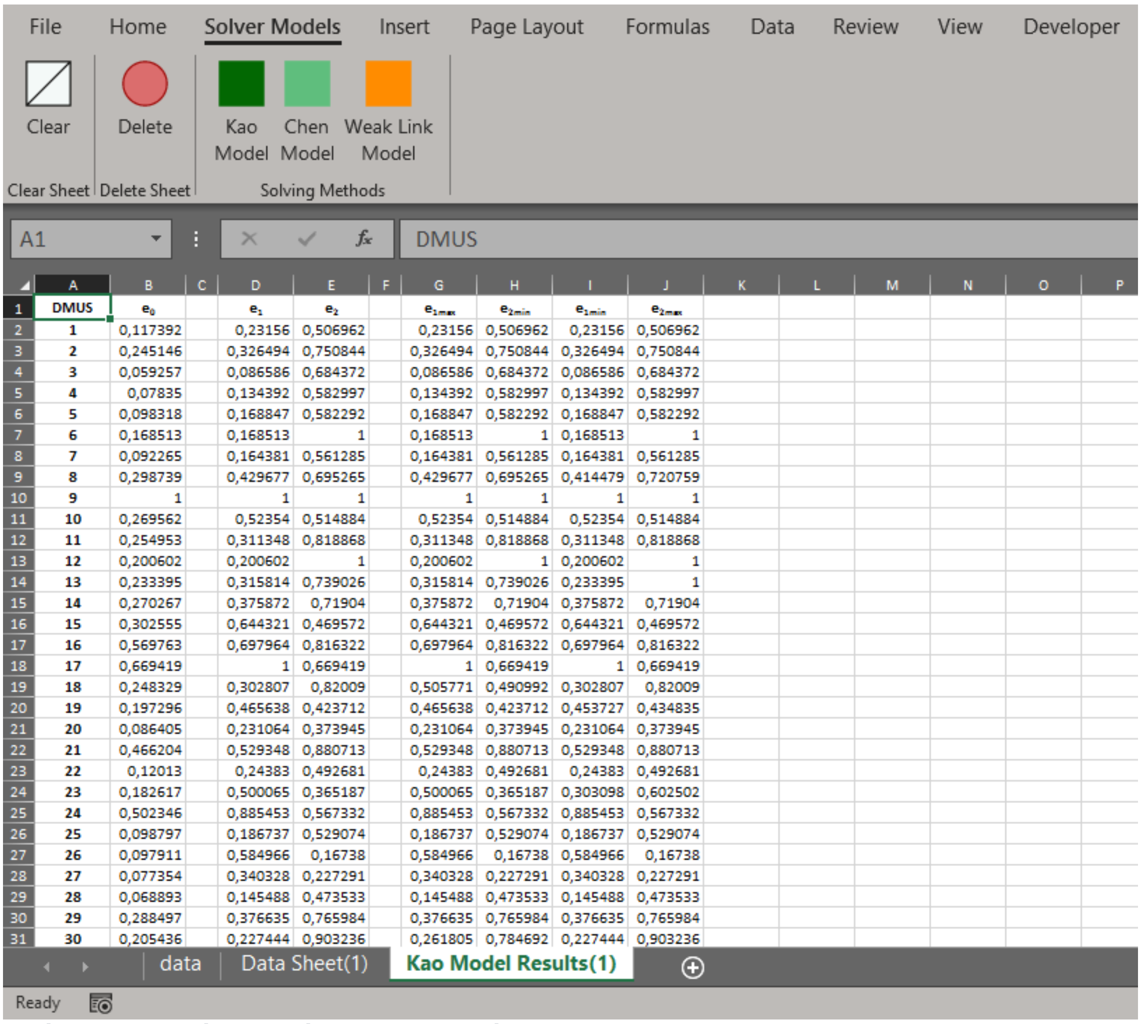


Εικόνα 4.5: Εισαγωγή δεδομένων



Εικόνα 4.6: Εκκίνηση επίλυσης

Με την επιλογή Next γίνεται η εκκίνηση του προγράμματος επίλυσης του μοντέλου. Με το τέλος των υπολογισμών θα γίνει αυτόματη προβολή των λύσεων σε καινούργιο Worksheet.

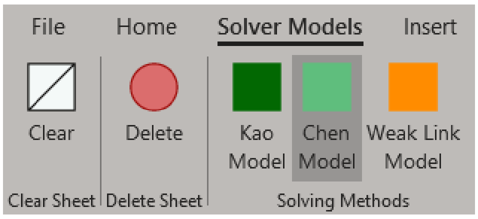


DMUS	e_0	e_1	e_2	e_{1max}	e_{2min}	e_{1min}	e_{2max}
1	0,117392	0,23156	0,506962	0,23156	0,506962	0,23156	0,506962
2	0,245146	0,326494	0,750844	0,326494	0,750844	0,326494	0,750844
3	0,059257	0,086586	0,684372	0,086586	0,684372	0,086586	0,684372
4	0,07835	0,134392	0,582997	0,134392	0,582997	0,134392	0,582997
5	0,098318	0,168847	0,582292	0,168847	0,582292	0,168847	0,582292
6	0,168513	0,168513	1	0,168513	1	0,168513	1
7	0,092265	0,164381	0,561285	0,164381	0,561285	0,164381	0,561285
8	0,298739	0,429677	0,695265	0,429677	0,695265	0,414479	0,720759
9	1	1	1	1	1	1	1
10	0,269562	0,52354	0,514884	0,52354	0,514884	0,52354	0,514884
11	0,254953	0,311348	0,818868	0,311348	0,818868	0,311348	0,818868
12	0,200602	0,200602	1	0,200602	1	0,200602	1
13	0,233395	0,315814	0,739026	0,315814	0,739026	0,233395	1
14	0,270267	0,375872	0,71904	0,375872	0,71904	0,375872	0,71904
15	0,302555	0,644321	0,469572	0,644321	0,469572	0,644321	0,469572
16	0,569763	0,697964	0,816322	0,697964	0,816322	0,697964	0,816322
17	0,669419	1	0,669419	1	0,669419	1	0,669419
18	0,248329	0,302807	0,82009	0,505771	0,490992	0,302807	0,82009
19	0,197296	0,465638	0,423712	0,465638	0,423712	0,453727	0,434835
20	0,086405	0,231064	0,373945	0,231064	0,373945	0,231064	0,373945
21	0,466204	0,529348	0,880713	0,529348	0,880713	0,529348	0,880713
22	0,12013	0,24383	0,492681	0,24383	0,492681	0,24383	0,492681
23	0,182617	0,500065	0,365187	0,500065	0,365187	0,303098	0,602502
24	0,502346	0,885453	0,567332	0,885453	0,567332	0,885453	0,567332
25	0,098797	0,186737	0,529074	0,186737	0,529074	0,186737	0,529074
26	0,097911	0,584966	0,16738	0,584966	0,16738	0,584966	0,16738
27	0,077354	0,340328	0,227291	0,340328	0,227291	0,340328	0,227291
28	0,068893	0,145488	0,473533	0,145488	0,473533	0,145488	0,473533
29	0,288497	0,376635	0,765984	0,376635	0,765984	0,376635	0,765984
30	0,205436	0,227444	0,903236	0,261805	0,784692	0,227444	0,903236

Εικόνα 4.7: Τελικά αποτελέσματα του μοντέλου των Kao et al. (2008)

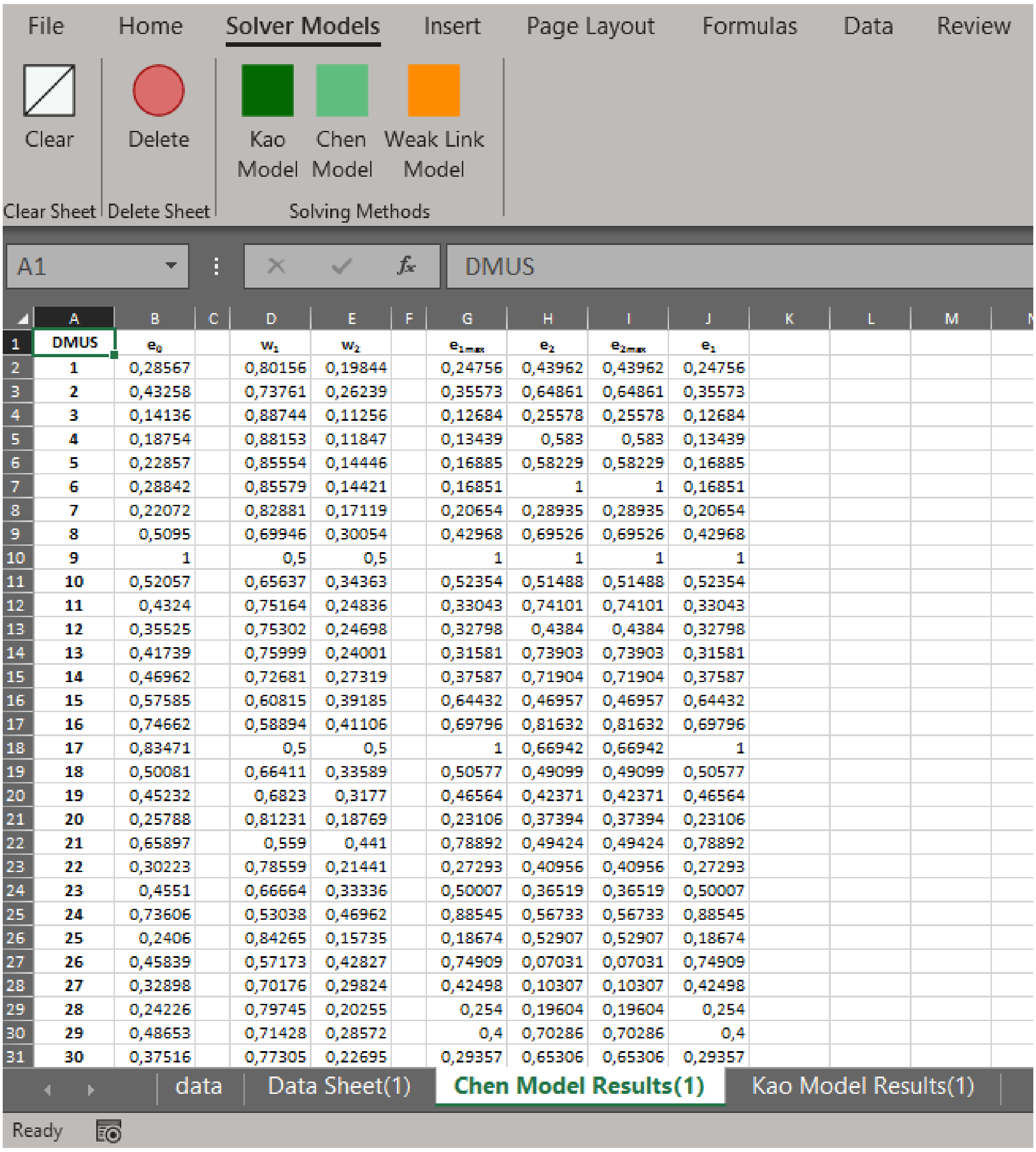
4.2 Chen Model

Η ίδια ακολουθία ενεργειών απαιτείται και για την επίλυση του μοντέλου των Chen et al. (2008). Με την επιλογή του Chen Model από το group Solving Methods



Εικόνα 4.8: Επιλογή μοντέλου

Γίνεται η εκκίνηση του μοντέλου και ο χρήστης εισάγει τις τιμές στα απαιτούμενα πεδία. Τα τελικά αποτελέσματα εμφανίζονται στο αντίστοιχο Worksheet του μοντέλου.

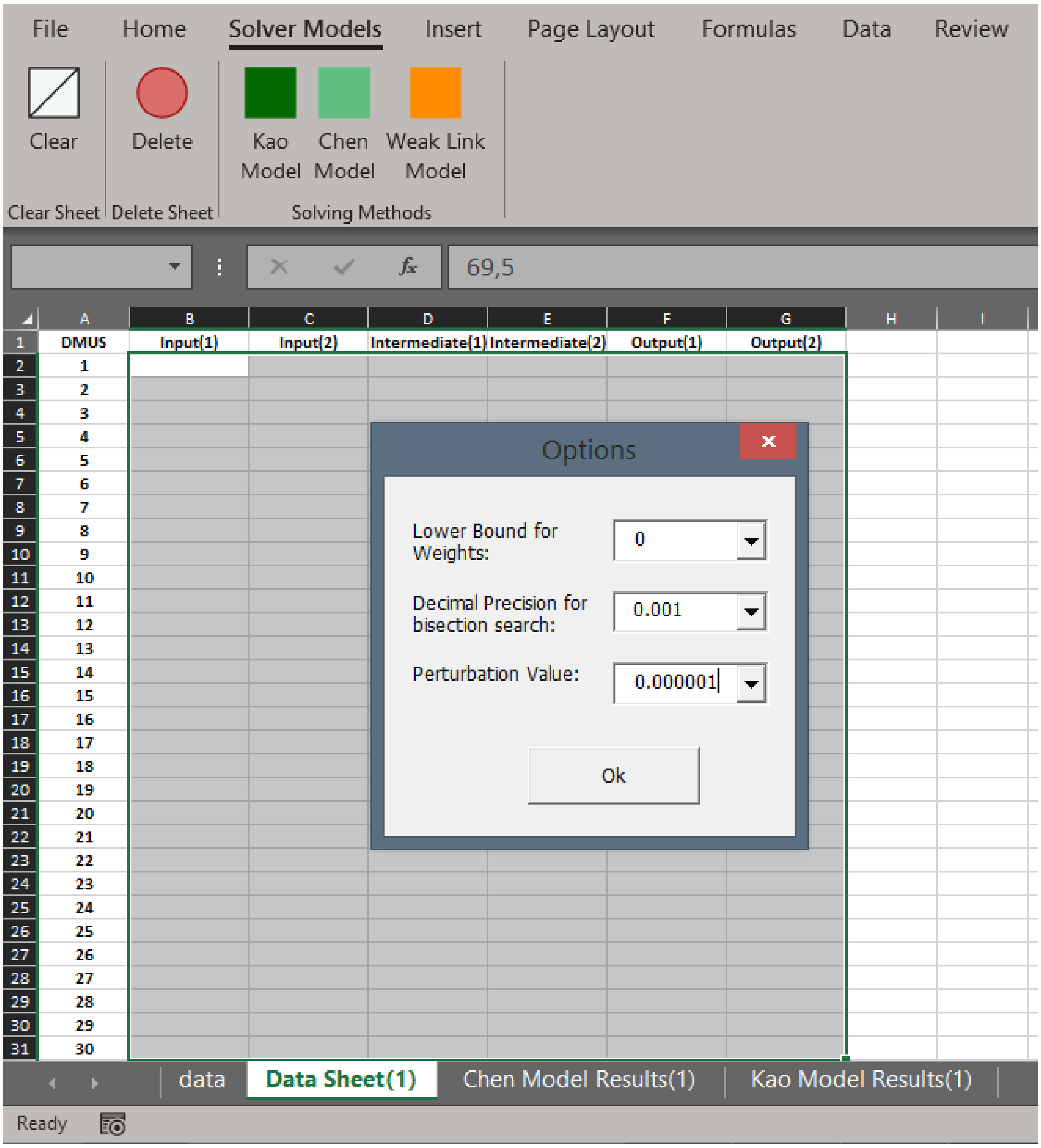


DMUS	e_0	w_1	w_2	e_{1max}	e_2	e_{2max}	e_1
1	0,28567	0,80156	0,19844	0,24756	0,43962	0,43962	0,24756
2	0,43258	0,73761	0,26239	0,35573	0,64861	0,64861	0,35573
3	0,14136	0,88744	0,11256	0,12684	0,25578	0,25578	0,12684
4	0,18754	0,88153	0,11847	0,13439	0,583	0,583	0,13439
5	0,22857	0,85554	0,14446	0,16885	0,58229	0,58229	0,16885
6	0,28842	0,85579	0,14421	0,16851	1	1	0,16851
7	0,22072	0,82881	0,17119	0,20654	0,28935	0,28935	0,20654
8	0,5095	0,69946	0,30054	0,42968	0,69526	0,69526	0,42968
9	1	0,5	0,5	1	1	1	1
10	0,52057	0,65637	0,34363	0,52354	0,51488	0,51488	0,52354
11	0,4324	0,75164	0,24836	0,33043	0,74101	0,74101	0,33043
12	0,35525	0,75302	0,24698	0,32798	0,4384	0,4384	0,32798
13	0,41739	0,75999	0,24001	0,31581	0,73903	0,73903	0,31581
14	0,46962	0,72681	0,27319	0,37587	0,71904	0,71904	0,37587
15	0,57585	0,60815	0,39185	0,64432	0,46957	0,46957	0,64432
16	0,74662	0,58894	0,41106	0,69796	0,81632	0,81632	0,69796
17	0,83471	0,5	0,5	1	0,66942	0,66942	1
18	0,50081	0,66411	0,33589	0,50577	0,49099	0,49099	0,50577
19	0,45232	0,6823	0,3177	0,46564	0,42371	0,42371	0,46564
20	0,25788	0,81231	0,18769	0,23106	0,37394	0,37394	0,23106
21	0,65897	0,559	0,441	0,78892	0,49424	0,49424	0,78892
22	0,30223	0,78559	0,21441	0,27293	0,40956	0,40956	0,27293
23	0,4551	0,66664	0,33336	0,50007	0,36519	0,36519	0,50007
24	0,73606	0,53038	0,46962	0,88545	0,56733	0,56733	0,88545
25	0,2406	0,84265	0,15735	0,18674	0,52907	0,52907	0,18674
26	0,45839	0,57173	0,42827	0,74909	0,07031	0,07031	0,74909
27	0,32898	0,70176	0,29824	0,42498	0,10307	0,10307	0,42498
28	0,24226	0,79745	0,20255	0,254	0,19604	0,19604	0,254
29	0,48653	0,71428	0,28572	0,4	0,70286	0,70286	0,4
30	0,37516	0,77305	0,22695	0,29357	0,65306	0,65306	0,29357

Εικόνα 4.9: Τελικά αποτελέσματα του μοντέλου των Chen et al. (2008)

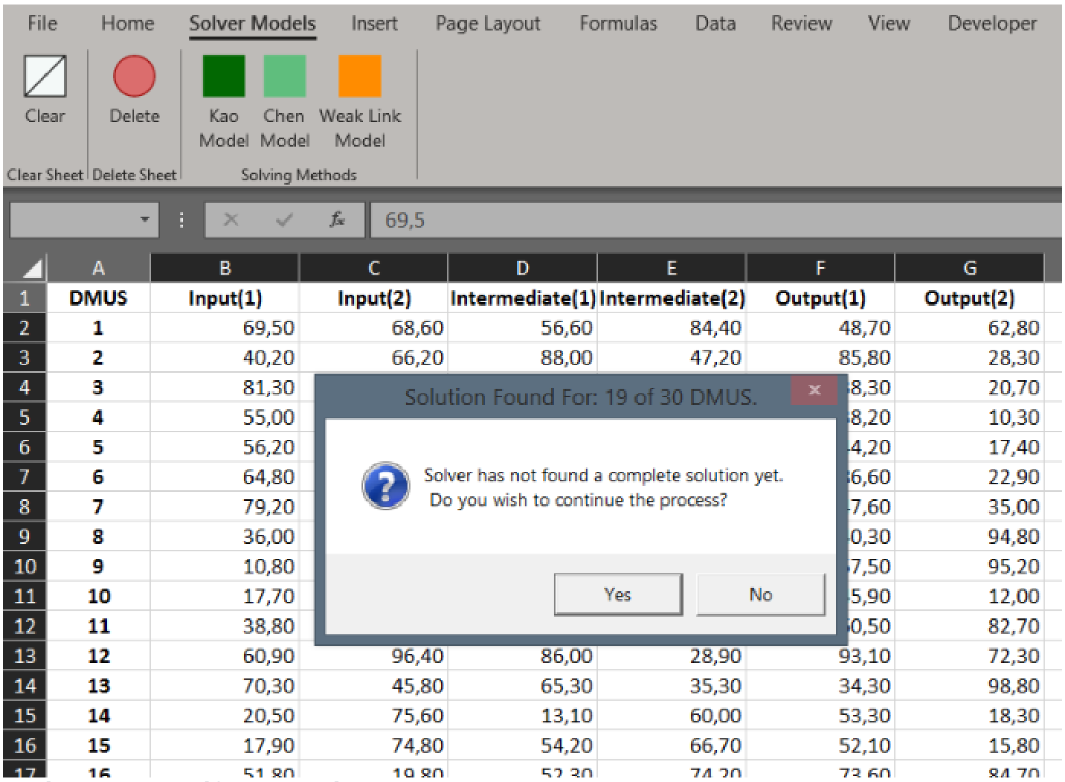
4.3 Weak Link Model

Όπως προαναφέρθηκε το μοντέλο των Despotis et al. (2016) απαιτεί την συμπλήρωση μιας ακόμα φόρμας χρήστη. Αφού συμπληρωθούν τα αρχικά πεδία του αριθμού των Μονάδων απόφασης, εισροών, ενδιάμεσων μεγεθών και εκροών ο χρήστης επιλέγει τους επιπλέον περιορισμούς του μοντέλου μέσω drop down λιστών επιλογών.



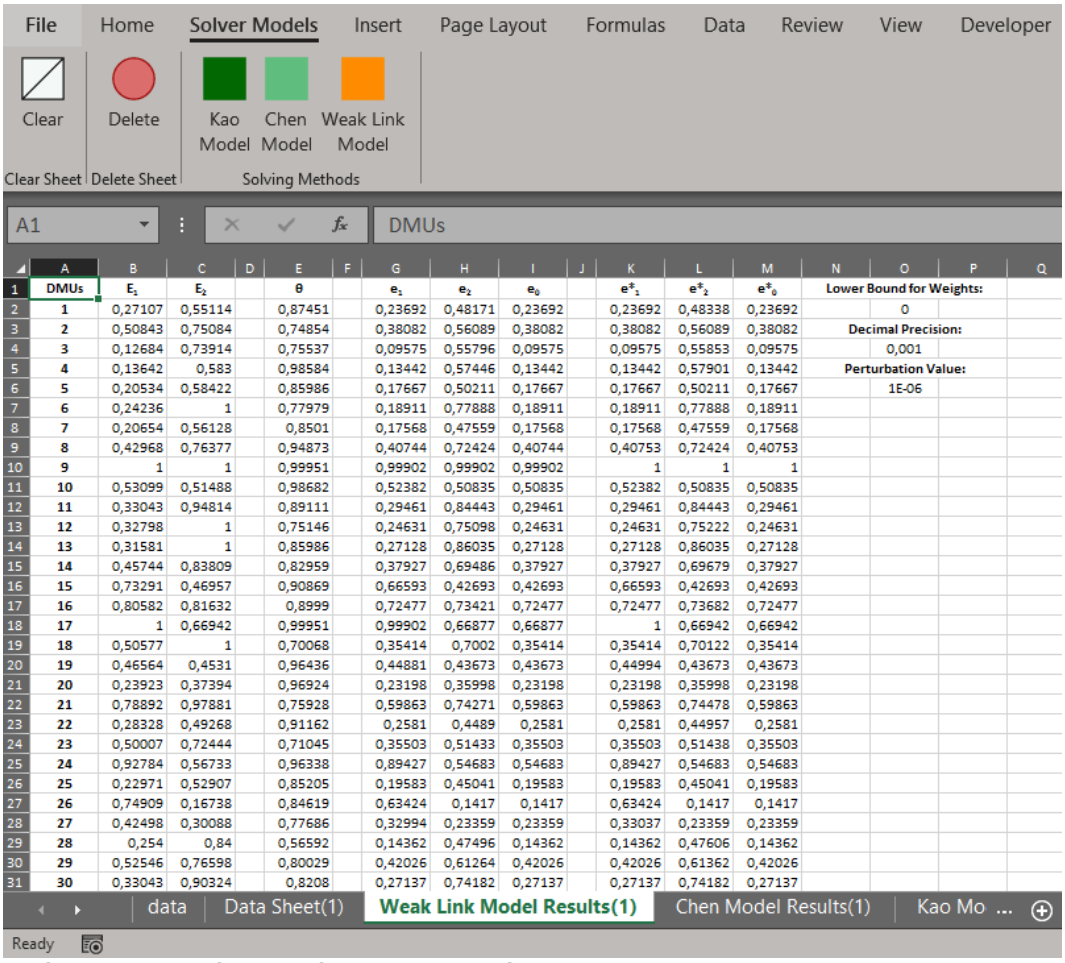
Εικόνα 4.10: Λίστες επιλογών για το Weak Link Model

Η επίλυση του συγκεκριμένου μοντέλου είναι αρκετά πιο απαιτητική συγκριτικά με τα υπόλοιπα μοντέλα που υλοποιήθηκαν. Συγκεκριμένα, η επίλυση του γραμμικού προβλήματος (2.11) με την μέθοδο της διχοτομικής αναζήτησης ενδέχεται να καταστεί αρκετά χρονοβόρα ενώ σε πολλές περιπτώσεις ενδέχεται το πρόγραμμα να μην είναι ικανό να βρει τις βέλτιστες λύσεις για συγκεκριμένες Μονάδες Απόφασης υπό τις παραπάνω επιλογές . Για την κάλυψη περιπτώσεων όπως αυτή υπάρχει μια επιπλέον ενημέρωση χρήστη η οποία ενεργοποιείται ανά σταθερά διαστήματα, βασισμένα στις επαναλήψεις του Solver για την εύρεση λύσης.



Εικόνα 4.11: Παράθυρο ενημέρωσης για το Weak Link Model

Όπως φαίνεται στην εικόνα, ο χρήστης ενημερώνεται για την μέχρι στιγμής κατάσταση επίλυσης του μοντέλου μέσω ενός ενημερωτικού πλαισίου το οποίο πληροφορεί τον χρήστη για τον αριθμό των μονάδων για τις οποίες έχει βρεθεί λύση καθώς και αν επιθυμεί την συνέχιση ή διακοπή της διαδικασίας. Με την συνέχιση της διαδικασίας ο Solver συνεχίζει την επίλυση μέχρι την ολοκλήρωση των υπολογισμών ή μέχρι την εμφάνιση νέου ενημερωτικού πλαισίου κατάστασης. Με την διακοπή της διαδικασίας ο Solver σταματάει την επίλυση, οι μέχρι τότε λύσεις διαγράφονται από το Worksheet των αποτελεσμάτων και ο χρήστης μπορεί να κάνει επανεκκίνηση για το μοντέλο με διαφορετικές επιλογές στους περιορισμούς του μοντέλου. Με την ολοκλήρωση του μοντέλου τα αποτελέσματα εμφανίζονται στο αντίστοιχο Worksheet.



DMUs	E_1	E_2	θ	e_1	e_2	e_0	e^*_1	e^*_2	e^*_0	Lower Bound for Weights:
1	0,27107	0,55114	0,87451	0,23692	0,48171	0,23692	0,23692	0,48338	0,23692	0
2	0,50843	0,75084	0,74854	0,38082	0,56089	0,38082	0,38082	0,56089	0,38082	Decimal Precision:
3	0,12684	0,73914	0,75537	0,09575	0,55796	0,09575	0,09575	0,55853	0,09575	0,001
4	0,13642	0,583	0,98584	0,13442	0,57446	0,13442	0,13442	0,57901	0,13442	Perturbation Value:
5	0,20534	0,58422	0,85986	0,17667	0,50211	0,17667	0,17667	0,50211	0,17667	1E-06
6	0,24236	1	0,77979	0,18911	0,77888	0,18911	0,18911	0,77888	0,18911	
7	0,20654	0,56128	0,8501	0,17568	0,47559	0,17568	0,17568	0,47559	0,17568	
8	0,42968	0,76377	0,94873	0,40744	0,72424	0,40744	0,40753	0,72424	0,40753	
9	1	1	0,99951	0,99902	0,99902	0,99902	1	1	1	
10	0,53099	0,51488	0,98682	0,52382	0,50835	0,50835	0,52382	0,50835	0,50835	
11	0,33043	0,94814	0,89111	0,29461	0,84443	0,29461	0,29461	0,84443	0,29461	
12	0,32798	1	0,75146	0,24631	0,75098	0,24631	0,24631	0,75222	0,24631	
13	0,31581	1	0,85986	0,27128	0,86035	0,27128	0,27128	0,86035	0,27128	
14	0,45744	0,83809	0,82959	0,37927	0,69486	0,37927	0,37927	0,69679	0,37927	
15	0,73291	0,46957	0,90869	0,66593	0,42693	0,42693	0,66593	0,42693	0,42693	
16	0,80582	0,81632	0,8999	0,72477	0,73421	0,72477	0,72477	0,73682	0,72477	
17	1	0,66942	0,99951	0,99902	0,66877	0,66877	1	0,66942	0,66942	
18	0,50577	1	0,70068	0,35414	0,7002	0,35414	0,35414	0,70122	0,35414	
19	0,46564	0,4531	0,96436	0,44881	0,43673	0,43673	0,44994	0,43673	0,43673	
20	0,23923	0,37394	0,96924	0,23198	0,35998	0,23198	0,23198	0,35998	0,23198	
21	0,78892	0,97881	0,75928	0,59863	0,74271	0,59863	0,59863	0,74478	0,59863	
22	0,28328	0,49268	0,91162	0,2581	0,4489	0,2581	0,2581	0,44957	0,2581	
23	0,50007	0,72444	0,71045	0,35503	0,51433	0,35503	0,35503	0,51438	0,35503	
24	0,92784	0,56733	0,96338	0,89427	0,54683	0,54683	0,89427	0,54683	0,54683	
25	0,22971	0,52907	0,85205	0,19583	0,45041	0,19583	0,19583	0,45041	0,19583	
26	0,74909	0,16738	0,84619	0,63424	0,1417	0,1417	0,63424	0,1417	0,1417	
27	0,42498	0,30088	0,77686	0,32994	0,23359	0,23359	0,33037	0,23359	0,23359	
28	0,254	0,84	0,56592	0,14362	0,47496	0,14362	0,14362	0,47606	0,14362	
29	0,52546	0,76598	0,80029	0,42026	0,61264	0,42026	0,42026	0,61362	0,42026	
30	0,33043	0,90324	0,8208	0,27137	0,74182	0,27137	0,27137	0,74182	0,27137	

Εικόνα 4.12: Τελικά αποτελέσματα του μοντέλου των Despotis et al. (2016)

Κεφάλαιο 5

Συμπεράσματα

Βασικός στόχος της εργασίας ήταν η δημιουργία μιας αυτοματοποιημένης διαδικασίας στο περιβάλλον του MS Excel για την αποτίμηση της αποδοτικότητας σε βασικά μοντέλα της Πολυσταδιακή ΠΑΔ. Τα μοντέλα που υλοποιήθηκαν στο πλαίσιο αυτό αφορούν μονάδες που αποτελούνται από δύο στάδια παραγωγής με σειριακή διάταξη υπό την υπόθεση περί κλίμακας σταθερών αποδόσεων. Από τα μοντέλα των Kao et al. (2008), Chen et al. (2009) και Despotis et al. (2016) που μελετήθηκαν το μοντέλο των Chen et al. (2009) μπορεί να επεκταθεί και σε κλίμακα μεταβλητών αποδόσεων ενώ αυτό των Despotis et al. (2016) μπορεί να εφαρμοστεί και σε περισσότερο σύνθετες δομές πολλών σταδίων με σειριακή διάταξη.

Η υλοποίηση του συστήματος βασίστηκε στο MS Excel και τον Solver. Η διαδεδομένη χρήση τους καθιστά το πρόγραμμα αυτοματοποίησης προσιτό σε όλους ανεξαρτήτως γνωστικού υπόβαθρου στο κομμάτι του γραμμικού προγραμματισμού και της Περιβάλλουσας Ανάλυσης Δεδομένων. Επιπλέον, η επιλογή που προσφέρει το Excel για την δημιουργία αυτοματοποιημένων εργαλείων με μία γλώσσα όπως η Visual Basic for Applications (VBA) παρέχει την δυνατότητα επεξεργασίας και τροποποίησης των μοντέλων που έχουν δημιουργηθεί. Με αυτόν τον τρόπο γίνεται εύκολα η επέκταση της χρήσης τους ή η προσαρμογή τους στις απαιτήσεις του εκάστοτε αναλυτή.

Παρόλα αυτά, τα μοντέλα που διαμορφώνονται σε τέτοιο περιβάλλον περιορίζονται σημαντικά από τα ίδια τα εργαλεία που χρησιμοποιήθηκαν στην δημιουργία τους. Βασικά περιοριστικά στοιχεία αποτελούν τα όρια που θέτει ο Solver στον αριθμό των περιορισμών και των μεταβλητών απόφασης που μπορεί να διαχειριστεί καθώς και η περιορισμένη υπολογιστική ισχύς του Excel. Ο συνδυασμός αυτός είναι αρκετά δεσμευτικός σχετικά με το μέγεθος και την πολυπλοκότητα των προβλημάτων προς επίλυση. Η γνώση των περιορισμών είναι σημαντική ώστε να μην αναιρείται η χρησιμότητα των αυτοματοποιημένων διαδικασιών όπως αυτή που δημιουργήθηκε για την εργασία και να βρίσκουν εφαρμογή στην επίλυση προβλημάτων αντίστοιχου μεγέθους και πολυπλοκότητας.

Βιβλιογραφία

Kao, Chiang & Hwang, Shiuh-Nan. (2008). Efficiency decomposition in two-stage data envelopment analysis: An application to non-life insurance companies in Taiwan. European Journal of Operational Research. 185. 418-429

Chen, Yao & Cook, Wade & Li, Ning & Zhu, Joe. (2009). Additive efficiency decomposition in two-stage DEA. European Journal of Operational Research. 196. 1170-1176

Despotis, Dimitris & Koronakos, Gregory & Sotiros, Dimitris. (2016). The "weak-link" approach to network DEA for two-stage processes. European Journal of Operational Research. 254

Koronakos, Grigorios (2017). The composition and the weak-link approaches to Network Data Envelopment Analysis. Doctoral Thesis, University of Piraeus.

Sotiros, Dimitrios – Georgios (2016). Value based and network Data Envelopment Analysis: new models and applications. Doctoral Thesis, University of Piraeus.

Κοπελιάρη, Μαρία Ε. (2013). Μοντέλα Περιβάλλουσας Ανάλυσης Δεδομένων σε παραγωγικές διαδικασίες δύο σταδίων. Μεταπτυχιακή Διατριβή, Πανεπιστήμιο Πειραιώς.

Jablonsky, Josef. (2008). A spreadsheet-based system for DEA models. http://www.mcdm.ue.katowice.pl/files/papers/mcdm07(3)_4.pdf

Ηλεκτρονικές Πηγές

https://www.solver.com/

https://peltiertech.com/Excel/SolverVBA.html

https://docs.microsoft.com/en-us/office/vba/api/overview/excel

http://www.cpearson.com/excel/mainpage.aspx

https://stackoverflow.com/

Παράρτημα

Kao Module

```
'************************************************************
'Desc: Multiplicative Decomposition Model overall efficiency
'************************************************************
Private Sub Solve_e0()

    Dim i As Integer

    byChangeVarCells = multiplierRng.Address
    conB = constraintB.Address
    conC = constraintC.Address

    Set destinationMultiplierRange = Worksheets(ResultSheet).Cells(2, 4)
    Set destinationObjectiveValue = Worksheets(ResultSheet).Range("$B$2")

    For i = 0 To numberOfDMUS

        Worksheets(Dmus).Activate

        objectiveValue = uYi.Offset(i, 0).Address
        stVX = vXi.Offset(i, 0).Address

        SolverReset
        SolverOptions Assumenonneg:=True
        SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
            Engine:=2, EngineDesc:="Simplex LP"

        SolverAdd CellRef:=stVX, Relation:=2, FormulaText:="1"
        SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"
        SolverAdd CellRef:=conC, Relation:=1, FormulaText:="0"

        SolverSolve (True)
        SolverFinish KeepFinal:=1

        destinationObjectiveValue.Offset(i, 0).Value = uYi.Offset(i, 0).Value
        multiplierRng.Copy destinationMultiplierRange.Offset(i, 0)

        Application.CutCopyMode = False

    Next

End Sub
```

```
'************************************************************
'Desc: Multiplicative Decomposition Model Stage I efficiency
'************************************************************
Private Sub e1()

   Dim X_MultiplierRange As Range, Z_MultiplierRange As Range

   Worksheets(Dmus).Activate

   Set e1Value = C.Offset(0, 1)
   Set e1Range = constraintC.Offset(0, 1)

   With Sheets(ResultSheet)
      Set Z_MultiplierRange = Range(Cells(2, 4 + numX), Cells(2, 3 + numX + numZ))
      Set X_MultiplierRange = Range(Cells(2, 4), Cells(2, 3 + numX))
   End With

   e1Value.FormulaArray = "=MMULT(" & ziRng.Address(RowAbsolute:=False, ColumnAbsolute:=False)_
                     & ",TRANSPOSE('" & Sheets(ResultSheet).Name & "'!" &
                     Z_MultiplierRange.Address(RowAbsolute:=False, ColumnAbsolute:=False) & "))" _
                     & "/" & "MMULT(" & xiRng.Address(RowAbsolute:=False, ColumnAbsolute:=False) _
                     & ",TRANSPOSE('" & Sheets(ResultSheet).Name & "'!" &
                     X_MultiplierRange.Address(RowAbsolute:=False, ColumnAbsolute:=False) & "))"

   e1Range.FillDown
   Application.Calculate

   e1Range.Copy
   rngE1.Offset(1, 0).PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False

End Sub
```

```
'*************************************************************
'Desc: Multiplicative Decomposition Model Stage II efficiency
'*************************************************************
Private Sub e2()

    Dim E2Fullrng As Range

    With Worksheets(ResultSheet)
            Set rngE1Value = rngE1.Offset(1, 0)
            Set rngE2Value = rngE1.Offset(1, 1)
            Set E0Value = Range("B2")
            Set fullrng = Range(rngE1.Offset(1, 0), rngE1.Offset(1, 0).End(xlDown))
            Set E2Fullrng = fullrng.Offset(0, 1)
    End With

    rngE2Value.FormulaArray = "=" & E0Value.Address(RowAbsolute:=False, ColumnAbsolute:=False) & _
        "/" & rngE1Value.Address(RowAbsolute:=False, ColumnAbsolute:=False)

    E2Fullrng.FillDown
    Application.Calculate
    E2Fullrng.Value = E2Fullrng.Value

End Sub
```

```
'************************************************************
'Desc: Multiplicative Decomposition Model Stage I maximum efficiency score
'Post-optimality procedure
'************************************************************
Private Sub Solve_e1max()

  Dim i As Integer
  Dim e1maxCon As String

  E1maxConstraint
  byChangeVarCells = multiplierRng.Address
  conB = constraintB.Address
  conC = constraintC.Address
  Set E1maxValue = Worksheets(ResultSheet).Range(rngE1.Address).Offset(1, 3)

  For i = 0 To numberOfDMUS

    Worksheets(Dmus).Activate

    objectiveValue = wZi.Offset(i, 0).Address
    stVX = vXi.Offset(i, 0).Address
    e1maxCon = e1MaxRng.Offset(i, 0).Address

    SolverReset
    SolverOptions Assumenonneg:=True
    SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
      Engine:=2, EngineDesc:="Simplex LP"

    SolverAdd CellRef:=stVX, Relation:=2, FormulaText:="1"
    SolverAdd CellRef:=e1maxCon, Relation:=2, FormulaText:="0"
    SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"
    SolverAdd CellRef:=conC, Relation:=1, FormulaText:="0"

    SolverSolve (True)
    SolverFinish KeepFinal:=1

    E1maxValue.Offset(i, 0).Value = wZi.Offset(i, 0).Value
    Application.CutCopyMode = False

  Next

  Solve_e2min

End Sub
```

```
'************************************************************
'Desc: Multiplicative Decomposition Model Stage II
'Post-optimality procedure
'************************************************************

Private Sub Solve_e2min()

    Dim E2minValue As Range, E2minRange As Range

    Worksheets(ResultSheet).Activate
    Set E2minValue = E1maxValue.Offset(0, 1)
    Set E2minRange = Range(E1maxValue, E1maxValue.End(xlDown)).Offset(0, 1)

    E2minValue.FormulaArray = "=" & E0Value.Address(RowAbsolute:=False, ColumnAbsolute:=False) &_
"/" & E1maxValue.Address(RowAbsolute:=False, ColumnAbsolute:=False)

    E2minRange.FillDown
    Application.Calculate
    E2minRange.Value = E2minRange.Value

End Sub



'************************************************************
'Desc: Multiplicative Decomposition Model Stage II maximum efficiency score
'Post-optimality procedure
'************************************************************
Private Sub Solve_e2max()

    Dim i As Integer
    Dim stWZ As String, e2maxCon As String
    byChangeVarCells = multiplierRng.Address
    conB = constraintB.Address
    conC = constraintC.Address

    Set E2maxValue = Worksheets(ResultSheet).Range(rngE1.Address).Offset(1, 6)

    For i = 0 To numberOfDMUS

    Worksheets(Dmus).Activate

    objectiveValue = uYi.Offset(i, 0).Address
    stWZ = wZi.Offset(i, 0).Address
    e2maxCon = e1MaxRng.Offset(i, 0).Address
```

```
    SolverReset
    SolverOptions Assumenonneg:=True
    SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
        Engine:=2, EngineDesc:="Simplex LP"

    SolverAdd CellRef:=stWZ, Relation:=2, FormulaText:="1"
    SolverAdd CellRef:=e2maxCon, Relation:=2, FormulaText:="0"
    SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"
    SolverAdd CellRef:=conC, Relation:=1, FormulaText:="0"

    SolverSolve (True)
    SolverFinish KeepFinal:=1

    E2maxValue.Offset(i, 0).Value = uYi.Offset(i, 0).Value
    Application.CutCopyMode = False

    Next

    Solve_e1min

End Sub



'***********************************************************
'Desc: Multiplicative Decomposition Model Stage I
'Post-optimality procedure
'***********************************************************
Private Sub Solve_e1min()

    Dim E1minValue As Range, E1minRange As Range

    Worksheets(ResultSheet).Activate

    Set E1minValue = E1maxValue.Offset(0, 2)
    Set E1minRange = Range(E1maxValue, E1maxValue.End(xlDown)).Offset(0, 2)

    E1minValue.FormulaArray = "=" & E0Value.Address(RowAbsolute:=False, ColumnAbsolute:=False) &_
"/" & E2maxValue.Address(RowAbsolute:=False, ColumnAbsolute:=False)

    E1minRange.FillDown
    Application.Calculate
    E1minRange.Value = E1minRange.Value

End Sub
```

Chen Module

```
'*************************************************************
'Desc: Additive Decomposition Model overall efficiency
'*************************************************************
Private Sub Solve_θo()

    Dim i As Integer
    Dim conC As String

    byChangeVarCells = multiplierRng.Address
    conA = constraintA.Address
    conB = constraintB.Address
    Set destinationMultiplierRange = Worksheets(ResultSheet).Cells(2, 4)
    Set destinationObjectiveValue = Worksheets(ResultSheet).Range("$B$2")

    For i = 0 To numberOfDMUS

    Worksheets(Dmus).Activate

    objectiveValue = maxValue.Offset(i, 0).Address
    conC = constraintCValue.Offset(i, 0).Address

    SolverReset
    SolverOptions Assumenonneg:=True
    SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
        Engine:=2, EngineDesc:="Simplex LP"

    SolverAdd CellRef:=conC, Relation:=2, FormulaText:="=1"
    SolverAdd CellRef:=conA, Relation:=1, FormulaText:="0"
    SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"

    SolverSolve (True)
    SolverFinish KeepFinal:=1

    destinationObjectiveValue.Offset(i, 0).Value = maxValue.Offset(i, 0).Value
    multiplierRng.Copy destinationMultiplierRange.Offset(i, 0)

    'weights w1, w2 for optimization variables
    Worksheets(ResultSheet).Range("D2").End(xlToRight).Offset(i, 2).Value = w1.Offset(i, 0).Value
    Worksheets(ResultSheet).Range("D2").End(xlToRight).Offset(i, 3).Value = w2.Offset(i, 0).Value

    Application.CutCopyMode = False

    Next

End Sub
```

```
'*************************************************************
'Desc: Additive Decomposition Model Stage I maximum efficiency score
'Post-optimality procedure
'*************************************************************
Private Sub Solve_θ1()

    Dim i As Integer
    Dim conC As String, conD As String

    Calculateθ1Constraint
    byChangeVarCells = multiplierRng.Address
    conA = constraintA.Address
    conB = constraintB.Address

    For i = 0 To numberOfDMUS

    Worksheets(Dmus).Activate

    objectiveValue = wZi.Offset(i, 0).Address
    conC = stè1Value.Offset(i, 0).Address
    conD = vXi.Offset(i, 0).Address

    SolverReset
    SolverOptions Assumenonneg:=True
    SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
        Engine:=2, EngineDesc:="Simplex LP"

    SolverAdd CellRef:=conA, Relation:=1, FormulaText:="0"
    SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"
    SolverAdd CellRef:=conC, Relation:=2, FormulaText:="=0"
    SolverAdd CellRef:=conD, Relation:=2, FormulaText:="=1"

    SolverSolve (True)
    SolverFinish KeepFinal:=1

    Worksheets(ResultSheet).Range("D2").End(xlToRight).Offset(i, 5).Value = wZi.Offset(i, 0).Value

    Application.CutCopyMode = False

    Next

    Calculateθ2

End Sub
```

```
'************************************************************
'Desc: Additive Decomposition Model Stage II
'Post-optimality procedure
'************************************************************
Private Sub Calculateθ2()

    Dim θ1maxValue As Range, θ1maxRange As Range

    Worksheets(ResultSheet).Activate

    Set w1AnswerReportValue = Range(Cells(2, 4), Cells(2, 4)).End(xlToRight).Offset(0, 2)
    Set w2AnswerReportValue = w1AnswerReportValue.Offset(0, 1)
    Set θ1maxValue = w1AnswerReportValue.Offset(0, 3)
    Set θ1maxRange = _
Range(w1AnswerReportValue.Address,w1AnswerReportValue.End(xlDown)).Offset(0, 3)

    Set θ2Value = θ1maxValue.Offset(0, 1)
    Set θ2Range = θ1maxRange.Offset(0, 1)

    θ2Value.FormulaArray = "=(" & θ0Value.Address(RowAbsolute:=False, ColumnAbsolute:=False) & "-"
& _
        w1AnswerReportValue.Address(RowAbsolute:=False, ColumnAbsolute:=False) & "*" & _
        θ1maxValue.Address(RowAbsolute:=False, ColumnAbsolute:=False) & ")/" & _
        w2AnswerReportValue.Address(RowAbsolute:=False, ColumnAbsolute:=False)

    θ2Range.FillDown
    Application.Calculate

    θ2Range.Value = θ2Range.Value

End Sub
```

```
'************************************************************
'Desc: Additive Decomposition Model Stage II maximum efficiency score
'Post-optimality procedure
'************************************************************
Private Sub Solve_θ2()

    Dim i As Integer
    Dim conC As String, conD As String, θ0 As String

    Calculateθ2Constraint
    byChangeVarCells = multiplierRng.Address
    conA = constraintA.Address
    conB = constraintB.Address

    For i = 0 To numberOfDMUS

    Worksheets(Dmus).Activate

    objectiveValue = uYi.Offset(i, 0).Address
    conC = stθ2Value.Offset(i, 0).Address
    conD = wZi.Offset(i, 0).Address
    'θ0 = θ0Value.Offset(i, 0).Address

    SolverReset
    SolverOptions Assumenonneg:=True
    SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
        Engine:=2, EngineDesc:="Simplex LP"

    SolverAdd CellRef:=conA, Relation:=1, FormulaText:="0"
    SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"
    SolverAdd CellRef:=conC, Relation:=2, FormulaText:="=0" '& Sheets(ResultSheet).Name & "'!" & θ0
    SolverAdd CellRef:=conD, Relation:=2, FormulaText:="=1"

    SolverSolve (True)
    SolverFinish KeepFinal:=1

    Worksheets(ResultSheet).Range("D2").End(xlToRight).Offset(i, 7).Value = uYi.Offset(i, 0).Value
    Application.CutCopyMode = False

    Next

    Calculateθ1

End Sub
```

```
'************************************************************
'Desc: Additive Decomposition Model Stage I
'Post-optimality procedure
'************************************************************
Private Sub Calculateθ1()

    Dim θ1Value As Range, θ1Range As Range

    Worksheets(ResultSheet).Activate

    Set θ2maxValue = θ2Value.Offset(0, 1)
    Set θ2maxRange = θ2Range.Offset(0, 1)

    Set θ1Value = θ2maxValue.Offset(0, 1)
    Set θ1Range = θ2maxRange.Offset(0, 1)

    Application.CutCopyMode = False
    θ1Value.FormulaArray = "=(" & θ0Value.Address(RowAbsolute:=False, ColumnAbsolute:=False) & "-"
& _ w2AnswerReportValue.Address(RowAbsolute:=False, ColumnAbsolute:=False) & "*" & _
        θ2maxValue.Address(RowAbsolute:=False, ColumnAbsolute:=False) & ")/" & _
        w1AnswerReportValue.Address(RowAbsolute:=False, ColumnAbsolute:=False)

    θ1Range.FillDown
    Application.Calculate
    θ1Range.Value = θ1Range.Value

End Sub
```

Weak link Module

```
'***********************************************************
'Desc: Weak link Model Stage I efficiency score
'Independent Assessment
'***********************************************************
Private Sub Solve_E1()

    Dim i As Integer

    byChangeVarCells = multiplierRng.Address
    conA = constraintA.Address
    conB = constraintB.Address

    Set destinationObjectiveValue = Worksheets(ResultSheet).Range("$B$2")

    For i = 0 To numberOfDMUS

        Worksheets(Dmus).Activate

        objectiveValue = wZi.Offset(i, 0).Address
        stVX = vXi.Offset(i, 0).Address

        SolverReset
        SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
            Engine:=2, EngineDesc:="Simplex LP"
        SolverAdd CellRef:=stVX, Relation:=2, FormulaText:="=1"
        SolverAdd CellRef:=conA, Relation:=1, FormulaText:="0"
        SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"

        SolverSolve (True)
        SolverFinish KeepFinal:=1

        destinationObjectiveValue.Offset(i, 0).Value = wZi.Offset(i, 0).Value
        Application.CutCopyMode = False

    Next

    Worksheets(ResultSheet).Activate

End Sub
```

```
'***********************************************************
'Desc: Weak link Model Stage II efficiency score
'Independent Assessment
'***********************************************************
Private Sub Solve_E2()

   Dim i As Integer
   Dim stWZ As String

   byChangeVarCells = multiplierRng.Address
   conA = constraintA.Address
   conB = constraintB.Address

   Set destinationObjectiveValue = Worksheets(ResultSheet).Range("$C$2")

   For i = 0 To numberOfDMUS

      Worksheets(Dmus).Activate

      objectiveValue = uYi.Offset(i, 0).Address
      stWZ = wZi.Offset(i, 0).Address

      SolverReset
      SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
         Engine:=2, EngineDesc:="Simplex LP"
      SolverAdd CellRef:=stWZ, Relation:=2, FormulaText:="=1"
      SolverAdd CellRef:=conA, Relation:=1, FormulaText:="0"
      SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"

      SolverSolve (True)
      SolverFinish KeepFinal:=1

      destinationObjectiveValue.Offset(i, 0).Value = uYi.Offset(i, 0).Value

      Application.CutCopyMode = False

   Next

   Worksheets(ResultSheet).Activate

End Sub
```

```
'***********************************************************
'Desc: Weak link Model Phase I efficiency score
'***********************************************************
Private Sub Solve_maxθ()

    Dim i As Integer, Results As Integer, Counter As Integer
    Dim destinationE1Value As Range, destinationE2Value As Range
    Dim θmax As Double, θmin As Double, θ As Double
    Dim stWZE1 As String, stWZE2 As String, stUY As String
    Dim e0 As Range, e0Range As Range
    Dim v As String, w As String, u As String

    maxθConstraints

    byChangeVarCells = multiplierRng.Address
    conA = constraintA.Address
    conB = constraintB.Address
    v = x_Multipliers.Address
    w = z_Multipliers.Address
    u = y_Multipliers.Address

    Set destinationObjectiveValue = Worksheets(ResultSheet).Range("$E$2")
    Set destinationE1Value = destinationObjectiveValue.Offset(0, 2)
    Set destinationE2Value = destinationE1Value.Offset(0, 1)

    For i = 0 To numberOfDMUS

        Worksheets(Dmus).Activate

        θmax = 1
        θmin = 0
        θ = (θmax + θmin) / 2

        objectiveValue = vXi.Offset(i, 0).Address
        stVX = vXi.Offset(i, 0).Address
        stWZE1 = wke1.Offset(i, 0).Address
        stWZE2 = wke2.Offset(i, 0).Address
        stUY = uYi.Offset(i, 0).Address

        Do While (θmax - θmin) >= dp And θ <= 1

            SolverReset
            SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
            Engine:=2, EngineDesc:="Simplex LP"

            SolverAdd CellRef:=stWZE1, Relation:=3, FormulaText:="=" & θ
            SolverAdd CellRef:=stUY, Relation:=3, FormulaText:="=" & θ & "*" & stWZE2

            SolverAdd CellRef:=stVX, Relation:=2, FormulaText:="=1"
            SolverAdd CellRef:=conA, Relation:=1, FormulaText:="0"
            SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"
            'weights >= å
            SolverAdd CellRef:=v, Relation:=3, FormulaText:="=" & å
            SolverAdd CellRef:=w, Relation:=3, FormulaText:="=" & å
            SolverAdd CellRef:=u, Relation:=3, FormulaText:="=" & å
            SolverAdd CellRef:=θ, Relation:=3, FormulaText:="0"
```

```
        Results = SolverSolve(True)
        SolverFinish KeepFinal:=1

        Counter = Counter + 1

          'Bisection search

          Select Case Results
          Case 0 'Solver found a solution. All constraints and optimality conditions are satisfied.
            θmin = θ
            θ = (θmax + θmin) / 2
          Case 5 'Solver could not find a feasible solution.
            θmax = θ
            θ = (θmax + θmin) / 2
          Case Else
            θ = θ + p
          End Select

        If Counter Mod 200 = 0 Then
          If MsgBox("Solver has not found a complete solution yet." & vbCrLf & " Do you wish to
continue the process?", _
          vbQuestion + vbYesNo + vbDefaultButton1, "Solution Found For: " & i & " of " & numDmu & "
DMUS.") _
            = vbNo Then BreakCode
        End If

      Loop

      destinationObjectiveValue.Offset(i, 0).Value = θ
      destinationE1Value.Offset(i, 0).Value = wZi.Offset(i, 0).Value
      destinationE2Value.Offset(i, 0).Value = uywz.Offset(i, 0).Value

      Application.CutCopyMode = False

    Next

    Application.Calculate

    Worksheets(ResultSheet).Activate
    Set e0 = destinationE2Value.Offset(0, 1)
    Set e0Range = Range(destinationE2Value, destinationE2Value.End(xlDown)).Offset(0, 1)
    'Overall efficiency as the minimum of stage efficiencies
    e0.FormulaArray = "=MIN(" & destinationE1Value.Address(RowAbsolute:=False,_
ColumnAbsolute:=False)  & ":" & destinationE2Value.Address(RowAbsolute:=False,_
ColumnAbsolute:=False) & ")"

    e0Range.FillDown
    Application.Calculate
    e0Range.Value = e0Range.Value

End Sub
```

```
'**********************************************************
'Desc: Weak link Model Phase II efficiency score
'**********************************************************
Private Sub Solve_maxS1plusS2()

    Dim i As Integer
    Dim destinationE1Value As Range, destinationE2Value As Range
    Dim stS1 As String, stS2 As String
    Dim e0 As Range, e0Range As Range
    Dim e1 As String, e2 As String
    Dim v As String, w As String, u As String

    S1S2Constraints
    byChangeVarCells = multiplierRng.Address
    conA = constraintA.Address
    conB = constraintB.Address
    v = x_Multipliers.Address
    w = z_Multipliers.Address
    u = y_Multipliers.Address

    Set destinationE1Value = Worksheets(ResultSheet).Range("$K$2")
    Set destinationE2Value = destinationE1Value.Offset(0, 1)

    For i = 0 To numberOfDMUS

        Worksheets(Dmus).Activate

        objectiveValue = s1s2.Offset(i, 0).Address
        stVX = vXi.Offset(i, 0).Address
        stS2 = s2.Offset(i, 0).Address
        stS1 = s1.Offset(i, 0).Address
        e1 = Worksheets(ResultSheet).Cells(2, 2).Offset(i, 0).Address
        e2 = Worksheets(ResultSheet).Cells(2, 3).Offset(i, 0).Address

        SolverReset
        SolverOk SetCell:=objectiveValue, MaxMinVal:=1, ValueOf:=0, ByChange:=byChangeVarCells, _
        Engine:=2, EngineDesc:="Simplex LP"

        SolverAdd CellRef:=stVX, Relation:=2, FormulaText:="=1"
        SolverAdd CellRef:=conA, Relation:=1, FormulaText:="0"
        SolverAdd CellRef:=conB, Relation:=1, FormulaText:="0"

        SolverAdd CellRef:=v, Relation:=3, FormulaText:="=" & ε
        SolverAdd CellRef:=w, Relation:=3, FormulaText:="=" & ε
        SolverAdd CellRef:=u, Relation:=3, FormulaText:="=" & ε

        SolverAdd CellRef:=stS1, Relation:=1, FormulaText:="=" & e1
        SolverAdd CellRef:=stS2, Relation:=1, FormulaText:="=" & e2
        SolverAdd CellRef:=stS1, Relation:=3, FormulaText:="=0"
        SolverAdd CellRef:=stS2, Relation:=3, FormulaText:="=0"

        SolverSolve (True)
        SolverFinish KeepFinal:=1
```

```
        destinationE1Value.Offset(i, 0).Value = wZi.Offset(i, 0).Value
        destinationE2Value.Offset(i, 0).Value = uywz.Offset(i, 0).Value

    Next

    Application.Calculate

    Set e0 = destinationE2Value.Offset(0, 1)
    Set e0Range = Range(destinationE2Value, destinationE2Value.End(xlDown)).Offset(0, 1)
    'Overall efficiency as the minimum of stage efficiencies
    e0.FormulaArray = "=MIN(" & destinationE1Value.Address(RowAbsolute:=False,
ColumnAbsolute:=False) _
                    & ":" & destinationE2Value.Address(RowAbsolute:=False, ColumnAbsolute:=False) & ")"
    e0Range.FillDown
    Application.Calculate

    e0Range.Value = e0Range.Value

End Sub
```