ΕΝΟΤΗΤΑ 12
Πολυκριτήριος Γραμμικός Προγραμματισμός (AUGMECON-R)

Εισαγωγή

Τα προβλήματα πολυκριτήριου μαθηματικού προγραμματισμού χωρίζονται σε τρεις κατηγορίες, ανάλογα με το στάδιο στο οποίο οι αποφασίζοντες εμπλέκονται στη διαδικασία λήψης αποφάσεων εκφράζοντας τις προτιμήσεις τους. Αυτές είναι:

Α) Μέθοδοι εκ των προτέρων έκφρασης προτίμησης (a priori methods)

Β) Αλληλεπιδραστικές μέθοδοι (interactive methods)

Γ) Μέθοδοι εκ των υστέρων έκφρασης προτίμησης (a posteriori methods)

Στην πρώτη περίπτωση, οι αποφασίζοντες εκφράζουν τις προτιμήσεις τους πριν τη διαδικασία επίλυσης του προβλήματος, θέτοντας στόχους ή βάρη στις αντικειμενικές συναρτήσεις. Οι επικριτές αυτής της μεθόδου ισχυρίζονται ότι είναι δύσκολο για τους αποφασίζοντες να γνωρίζουν και να μπορούν να ποσοτικοποιήσουν εκ των προτέρων τις προτιμήσεις τους. Στη δεύτερη περίπτωση, φάσεις διαλόγου με τους αποφασίζοντες εναλλάσσονται με φάσεις επίλυσης του προβλήματος με αποτέλεσμα μετά από διαδοχικές επαναλήψεις να πραγματοποιείται σύγκλιση των προτιμήσεων και των λύσεων που προκύπτουν. Δηλαδή οι αποφασίζοντες κατευθύνουν τη διαδικασία επίλυσης με στόχο την απόκτηση της πιο επιθυμητής. Το μειονέκτημα αυτών των μεθόδων είναι ότι ποτέ δεν έχουν εποπτεία ολόκληρου του μετώπου Pareto ή μίας προσέγγισής του. Στην τρίτη περίπτωση, το πρόβλημα επιλύεται, προκύπτουν οι Pareto αποτελεσματικές λύσεις και εν συνεχεία οι αποφασίζοντες επιλέγουν εξ αυτών την πλέον επιθυμητή.

Μια άλλη διάκριση των αλγορίθμων επίλυσης προβλημάτων πολυκριτήριου μαθηματικού προγραμματισμού αφορά στην ακρίβεια του μετώπου Pareto με αποτέλεσμα να διακρίνονται σε:

Α) Ευρετικούς (heuristics) αλγόριθμους (π.χ γενετικοί αλγόριθμοι) όπου επιτυγχάνεται μία προσέγγιση του μετώπου Pareto και η ακρίβεια δεν ενδιαφέρει τους αποφασίζοντες

Β) Μη ευρετικούς αλγορίθμους, όπου αποκτάται το ακριβές μέτωπο Pareto.

Η AUGMECON-R, η οποία αποτελεί εξέλιξη των AUGMECON (Mavrotas, 2009) και AUGMECON-2 (Florios and Mavrotas 2013) και ανήκει στην οικογένεια των αλγορίθμων ε-constraint, αποτελεί έναν a posteriori ταχύτατο αλγόριθμο εύρεσης του ακριβούς μετώπου Pareto (στην περίπτωση ακέραιου προγραμματισμού). Η βασική ιδέα ανάπτυξης του αλγορίθμου ήταν η εισαγωγή συντελεστών παράκαμψης σε όλους τους βρόχους επαναλήψεων και η επιστράτευση μιας δομής δεδομένων προκειμένου να αποθηκεύονται τα κατάλληλα δεδομένα και να αποφεύγονται οι περιττές βελτιστοποιήσεις-επαναλήψεις.

Η μέθοδος ε-constraint

Έστω το ακόλουθο πρόβλημα πολυκριτήριου μαθηματικού προγραμματισμού:

$$\max\{f_1(x), f_2(x), \dots, f_p(x)\}$$

$$subject\ to: x \in S$$

όπου: x είναι το διάνυσμα των μεταβλητών απόφασης, $f_1(x), f_2(x), \dots, f_p(x)$ είναι οι p αντικειμενικές συναρτήσεις, και S το χωρίο των αποτελεσματικών λύσεων.

Η βασική ιδέα της ε-constraint έγκειται στη βελτιστοποίηση της μίας αντικειμενικής θεωρώντας όλες τις υπόλοιπες ως περιορισμούς. Το πρόβλημα μετατρέπεται ως εξής:

$$\max\{f_1(x)\}$$

$$s.t.:$$

$$f_2(x) \geq e_2$$

$$f_3(x) \geq e_3$$

$$\dots$$

$$f_p(x) \geq e_p$$

$$x \in S$$

Μεταβάλλοντας το δεξί μέλος των περιορισμένων αντικειμενικών (e_i) αποκτούμε τις αποτελεσματικές λύσεις, με το πρόβλημα να επιλύεται πάνω σε ένα πλέγμα μεγέθους N_2*N_3*…*N_P όπου N_i είναι το εύρος της αντικειμενικής συνάρτησης f_i, δηλαδή η διαφορά μεταξύ μέγιστης και ελάχιστης τιμής, προσαυξημένη κατά μία μονάδα. Έτσι, η διαδικασία διαμορφώνεται ως εξής:

Αφού επιλυθεί το παραπάνω πρόβλημα, εν συνεχεία αυξάνεται το e_2 κατά μια ποσότητα $step_2$ (που αντιπροσωπεύει το βήμα), η οποία μπορεί να είναι και μοναδιαία. Επιλύοντας το πρόβλημα, λαμβάνεται μια δεύτερη λύση και αυξάνεται το e_2' κατά την ίδια ποσότητα $step_2$. Επιλύεται εκ νέου το πρόβλημα, λαμβάνοντας μια τρίτη λύση κ.ο.κ. μέχρι να φτάσουμε στο άνω όριο της f_2 οπότε τότε το δεξί μέλος γίνεται πάλι ίσο με το αρχικό και αυξάνεται κατά $step_3$ το e_3 οπότε πάλι το e_2 ακολουθεί την περιγραφείσα πορεία μέχρι να φτάσει στο άνω όριο, οπότε πάλι το $e_3'=e_3+step_3$ αυξάνεται κατά $step_3$. Όταν το δεξί μέλος της f_3 φτάσει στο άνω όριο τότε γίνεται πάλι ίσο με το αρχικό και τώρα αυξάνεται κατά $step_4$ το e_4 κ.ο.κ. Η διαδικασία επαναλαμβάνεται για όλες τις αντικειμενικές, μέχρι να καλυφθεί όλο το πλέγμα. Έτσι λαμβάνονται οι λύσεις του προβλήματος βελτιστοποίησης, οι οποίες είναι p-άδες.

Ένα πλεονέκτημα της ε-constraint είναι ότι μπορούμε να ελέγξουμε το πλήθος των παραγόμενων αποτελεσματικών λύσεων, προσαρμόζοντας κατάλληλα το πλήθος των σημείων του πλέγματος (grid points), δηλαδή των σημείων πάνω στα οποία θα πραγματοποιείται η εκάστοτε βελτιστοποίηση, στα εύρη των αντικειμενικών συναρτήσεων.

Ωστόσο, η μέθοδος ε-constraint χαρακτηρίζεται από τρεις βασικές αδυναμίες: η ανάγκη υπολογισμού του εύρους κάθε αντικειμενικής συνάρτησης πάνω στο σύνολο που περιέχει τις μεταβλητές απόφασης για τις οποίες λαμβάνουμε αποτελεσματικές κατά Pareto λύσεις· η αδυναμία εξασφάλισης ότι οι λύσεις που λαμβάνονται ανήκουν στο αποτελεσματικό μέτωπο Pareto και όχι στο ασθενές μέτωπο Pareto·και ο πολύ μεγάλος χρόνος επίλυσης για προβλήματα με παραπάνω από 2 αντικειμενικές συναρτήσεις.

Η AUGMECON

Η AUGMECON αποτελεί βελτίωση της ε-constraint, η οποία συνίσταται στις εξής τροποποιήσεις:

1. Οι περιορισμοί που εμπλέκουν μέσα τις p-1 το πλήθος αντικειμενικές συναρτήσεις μετατρέπονται απο ανισότητες σε ισότητες.
2. Εισάγονται μεταβλητές περιθωρίου (surplus ή slack variables, ανάλογα με το αν επιδιώκεται μεγιστοποίηση ή ελαχιστοποίηση, αντίστοιχα) τόσο στους περιορισμούς όσο και στην κύρια αντικειμενική, με τη μορφή αθροίσματος πολλαπλασιασμένο με την κατάλληλη σταθερά (eps).
3. Γίνεται πρόωρη έξοδος από τον τελευταίο εσωτερικό βρόχο (innermost loop) με στόχο την εξοικονόμηση χρόνου σε περίπτωση οπού το πρόβλημα είναι αδύνατο.

Οι δύο πρώτες αποσκοπούν στην απόκτηση μόνο αποτελεσματικών και όχι ασθενών εφικτών λύσεων ενώ η τρίτη στην ταχύτερη απόκριση του αλγορίθμου. Η φιλοσοφία ωστόσο των δύο μεθόδων είναι η ίδια. Μια καινοτόμος προσθήκη στον αλγόριθμο έγκειται στο ότι η AUGMECON εκμεταλλεύεται περιπτώσεις όπου το πρόβλημα είναι αδύνατο (infeasibile) με αποτέλεσμα την πρόωρη έξοδο από τον εμφωλευμένο βρόχο. Αυτή η προσθήκη επιταχύνει σημαντικά τον αλγόριθμο σε περιπτώσεις περισσότερων από τριών αντικειμενικών συναρτήσεων. Ο αλγόριθμος ξεκινά με τις περιορισμένες αντικειμενικές να φράσσονται από κάτω όρια, τα οποία σταδιακά κατά τη διάρκεια του αλγορίθμου γίνονται πιο αυστηρά. Αυτό σημαίνει ότι για μεγιστοποίηση (ελαχιστοποίηση) μίας αντικειμενικής συνάρτησης, ο αλγόριθμος ξεκινάει από ένα ελάχιστο (μέγιστο) και σταδιακά αυξάνεται (μειώνεται) το δεξί μέλος των αντίστοιχων περιορισμών. Κατά τη διάρκεια της διαδικασίας αυτής, εάν προκύψει ότι το πρόβλημα είναι αδύνατο, αυτό σημαίνει ότι δεν χρειάζεται να αυξήσουμε περεταίρω το δεξί μέλος της αντίστοιχης αντικειμενικής, αφού προφανώς πάλι θα οδηγηθούμε σε μια κατάσταση στην οποία το πρόβλημα είναι αδύνατο—από εδώ και στο εξής, αυτές οι καταστάσεις θα ονομάζονται infeasibilities. Δηλαδή, εάν το πρόβλημα είναι αδύνατο, τότε προφανώς όσο αυστηρότεροι γίνονται οι περιορισμοί τόσο πιο σίγουρο είναι ότι το πρόβλημα θα παραμείνει αδύνατο. Συνεπώς, ο αλγόριθμος εξέρχεται από τον πλέον εμφωλευμένο βρόχο και η διαδικασία συνεχίζεται από το επόμενο σημείο του πλέγματος της αντικειμενικής που αντιστοιχεί στον αμέσως εξωτερικότερο βρόχο από αυτόν που εξήλθε ο αλγόριθμος. Το πρόβλημα τώρα με αυτή την μικρή προσθήκη λαμβάνει τη μορφή:

$$\max\{f_1(x) + eps \times (s_2 + s_3 + \cdots + s_p)\}, eps \in (10^{-6}, 10^{-3})$$

$$subject\ to:$$

$$f_2(x) - s_2 = e_2$$

$$f_3(x) - s_3 = e_3$$

$$\dots$$

$$f_p(x) - s_p = e_p$$

$$x \in S$$

H AUGMECON-2

Συγκριτικά με την AUGMECON, η AUGMECON-2 τροποποιεί ελάχιστα την αντικειμενική συνάρτηση προς μεγιστοποίηση, ως εξής:

$$\max\{f_1(x) + eps \times \left(s_2/r_2 + 10^{-1} s_3/r_3 + \dots + 10^{-(p-2)} s_p/r_p\right)\}$$

Αυτή η τροποποίηση γίνεται προκειμένου να επιτευχθεί ένα είδος λεξικογραφικής βελτιστοποίησης στις υπόλοιπες αντικειμενικές στην περίπτωση που υπάρχουν εναλλακτικά βέλτιστα. Για παράδειγμα, με αυτήν τη μοντελοποίηση, το λογισμικό επίλυσης θα βρει το βέλτιστο για την f_1 και έπειτα θα προσπαθήσει να βελτιστοποιήσει την f_2, μετά την f_3 κ.ο.κ. Στην περίπτωση της AUGMECON η σειρά βελτιστοποίησης των f_2-f_p δεν έπαιζε κάποιο ρόλο, ενώ τώρα ο νέος αλγόριθμος πραγματοποιεί σειριακή βελτιστοποίηση των περιορισμένων αντικειμενικών (στην περίπτωση εναλλακτικών βέλτιστων). Για κάθε f_i με i=2,…,p, υπολογίζουμε το εύρος της από τον πίνακα πληρωμών (ή αποδόσεων, τιμών, ανταμοιβών, εκ του payoff table). Εν συνεχεία, διαιρούμε το εύρος της k-οστής αντικειμενικής σε q_k το πλήθος ίσα διαστήματα, χρησιμοποιώντας q_k-1 ισαπέχοντα εσωτερικά σημεία. Επομένως, έχουμε συνολικά q_k+1 σημεία πλέγματος (grid points) που χρησιμοποιούνται για να μεταβάλλουν παραμετρικά το δεξί μέλος (e_k) της k-οστής αντικειμενικής συνάρτησης. Ο συνολικός αριθμός των βελτιστοποιήσεων θα είναι $(q_1+1) \times (q_2+1) \times \dots (q_p + 1)$. Έστω λοιπόν r_k το εύρος της k-οστής αντικειμενικής με k=2, 3, …, p. Τότε το διακριτό βήμα μετά του οποίου θα μεταβάλλεται το δεξί μέλος είναι:

$$step_k = r_k/q_k$$

Το δεξί μέλος του αντίστοιχου περιορισμού γίνεται στην t-οστη επανάληψη:

$$e_{k,t} = fmin_k + t * step_k$$

όπου $fmin_k$ είναι το ελάχιστο της f_k από τον πίνακα πληρωμών και t είναι ο μετρητής της επανάληψης.

Σε κάθε επανάληψη ελέγχουμε τον συντελεστή περιθωρίου (sv) που αντιστοιχεί στην αντικειμενική του τελευταίου εσωτερικού βρόχου που στην προκειμένη περίπτωση είναι η f_2. Έπειτα υπολογίζουμε τον συντελεστή παράκαμψης (bypass coefficient) ως εξής:

$$b = int(s_2/step_2)$$

όπου η συνάρτηση $int()$ επιστρέφει το ακέραιο μέρος του ορίσματος.

Όταν ο συντελεστής περιθωρίου s_2 είναι μεγαλύτερη από το $step_2$, μπορούμε να εξάγουμε το συμπέρασμα ότι στην επόμενη επανάληψη θα πάρουμε την ίδια λύση από την βελτιστοποίηση με την μόνη διαφορά ότι θα έχει μειωθεί το sv κατά την τιμή του βήματος, οπότε η νέα του τιμή θα είναι s_2-$step_2$. Αυτό κάνει την συγκεκριμένη επανάληψη περιττή και συνεπώς μπορούμε να την αποφύγουμε αφού δεν συνεισφέρει νέα λύση στο σύνολο των αποτελεσματικών λύσεων. Στην ουσία, ο συντελεστής παράκαμψης μάς υποδεικνύει πόσες διαδοχικές επαναλήψεις μπορούμε να αποφύγουμε.

Μαθηματικό μοντέλο της AUGMECON-R

Έστω προς επίλυση το ακόλουθο πρόβλημα:

$$\max\{f_1(x), f_2(x), \dots, f_p(x)\}$$

$$subject\ to: x \in S$$

όπου x είναι το διάνυσμα των μεταβλητών απόφασης, $f_1(x), f_2(x), \dots, f_p(x)$ οι p το πλήθος αντικειμενικές συναρτήσεις και S ο χώρος των εφικτών λύσεων. Η AUGMECON-R (όπως και κάθε αλγόριθμος αυτής της οικογένειας) θεωρεί την μια αντικειμενική ελεύθερη να βελτιστοποιηθεί ενώ οι υπόλοιπες τίθενται ως περιορισμοί. Με βάση την AUGMECON-R το πρόβλημα λαμβάνει πλέον την ακόλουθη μορφή (πρόβλημα P):

$$\max\{f_1(x) + eps \times (s_2/r_2 + 10^{-1}s_3/r_3 + \dots + 10^{-(p-2)}s_p/r_p)\}$$

$$subject\ to:$$

$$f_2(x) - s_2 = e_2$$

$$f_3(x) - s_3 = e_3$$

$$\dots$$

$$f_p(x) - s_p = e_p$$

$$x \in S$$

όπου: τα s_i οι μεταβλητές περιθωρίου/απόκλισης (surplus or slack variables), τα e_i είναι τα δεξιά μέλη των περιορισμών και r_i είναι το ακέραιο εύρος της i-οστής αντικειμενικής.

Οι αρχικές τιμές των δεξιών μελών είναι τα θεωρητικά σημεία ναδίρ, δηλαδή οι ελάχιστες τιμές που λαμβάνουν οι αντικειμενικές συναρτήσεις. Αυξάνοντας τα δεξιά μέλη των περιορισμένων αντικειμενικών και προχωρώντας βήμα-βήμα στο $N_2 \times N_3 \times \dots \times N_p$ πλέγμα που δημιουργείται, αποκτούμε τα σημεία-λύσεις του μετώπου Pareto, όπου N_i είναι το ακέραιο εύρος της i-οστής αντικειμενικής προσαυξημένο κατά 1. Οι συντελεστές παράκαμψης που θα εισάγουμε χρησιμοποιούν την πληροφορία που προκύπτει από τους συντελεστές περιθωρίου των περιορισμένων αντικειμενικών με στόχο την αποφυγή περιττών επαναλήψεων και την επιτάχυνση της επίλυσης. Για την απόκτηση του ακριβούς μετώπου Pareto, βασική προϋπόθεση είναι οι συντελεστές των αντικειμενικών να είναι ακέραιοι.

Βήματα επίλυσης του προβλήματος

1. Από τον πίνακα πληρωμών λαμβάνουμε τα διαγώνια στοιχεία που αντιστοιχούν στις μέγιστες τιμές των αντικειμενικών. Από κάθε στήλη, για την i-οστή αντικειμενική λαμβάνουμε την ελάχιστη τιμή. Αυτή τη τιμή την πολλαπλασιάζουμε με έναν μειωτικό συντελεστή (π.χ. 0,1) με στόχο να λάβουμε μια υποεκτίμηση των nadir points (ιδανικά οι ελάχιστες τιμές των αντικειμενικών θα μπορούσαν να τεθούν μηδενικές χωρίς αυτό να επηρεάζει ουσιαστικά τη λειτουργεία ή τον χρόνο εκτέλεσης της AUGMECON-R).

2. Υπολογίζουμε το εύρος κάθε αντικειμενικής ως εξής:

$$r_i = maxf_i - minf_i$$

3. Το πλέγμα θα έχει μέγεθος $(r_2 + 1) \times (r_3 + 1) \times \dots (r_p + 1)$.

4. Τα e_i για την πρώτη επανάληψη τίθενται ίσα με $minf_i$.

5. Ανάλογα με την πυκνότητα του μετώπου Pareto που επιθυμούμε, θέτουμε το κατάλληλο βήμα:

$$step_i = r_i/q_i$$

με το οποίο ο αλγόριθμος θα προχωράει πάνω στο πλέγμα, όπου q_i είναι το πλήθος των ίσων διαστημάτων στα οποία χωρίζεται το εύρος r_i για να σχηματίσει το πλέγμα. Ιδανικά, για να μην παραλειφθεί ούτε μια λύση, τίθεται μοναδιαίο βήμα $step_i = 1$.

6. Σε κάθε βήμα, πραγματοποιείται η βελτιστοποίηση του προβλήματος (P) από το λογισμικό επίλυσης και προκύπτει ένα σημείο του μετώπου Pareto $[f_1(x^*), f_2(x^*), \dots, f_p(x^*)]$ και τα s_i. Υπολογίζουμε τους συντελεστές παράκαμψης για τις περιορισμένες αντικειμενικές ως εξής:

$$b_2 = int(s_2/step_2)$$

$$b_3 = int(s_3/step_3)$$

$$\dots$$

$$b_p = int(s_p/step_p)$$

όπου η συνάρτηση $int()$ επιστρέφει το ακέραιο μέρος του ορίσματος.

7. Έστω j_i ο δείκτης του for loop που αντιστοιχεί στην i-οστή αντικειμενική, τότε αμέσως μετά τον υπολογισμό των b_i συμβαίνει η εξής αποθήκευση δεδομένων στον $N_2 \times N_3 \times \dots \times N_p$ πίνακα δεδομένων:

```
For (jp, currentp to currentp+bp) {
        For (jp-1, currentp-1 to currentp-1+bp-1) {
                ...
                        For (jp, currentp)
                                {flag[jp][jp-1]...[j2]=b2+1}...}}
```

Με αυτό τον τρόπο, όταν σε μια επανάληψη $s_i > step_i$, στην επόμενη επανάληψη για το b_i' που αντιστοιχεί στο $e_i' = e_i' + step_i$, η βελτιστοποίηση θα οδηγήσει στην ίδια ακριβώς λύση, με την μόνη διαφορά ότι πλέον $s_i' = s_i - step_i$, η οποία όμως είναι αχρείαστη και πρέπει να την αποφύγουμε. Ο συντελεστής παράκαμψης b_i υποδεικνύει πόσες επαναλήψεις πρέπει να παρακάμψουμε, δεδομένου ότι αυτές οι επαναλήψεις αφορούν στην i-οστή αντικειμενική συνάρτηση και ότι τα δεξιά μέλη των υπόλοιπων αντικειμενικών παραμένουν σταθερά.

Η γενική μορφή της μετάβασης και της εξερεύνησης πάνω στο πλέγμα είναι η ακόλουθη:

For (i, 0 to max_i) {

$e_p = minf_p + i * step_p$

For (j, 0 to max_j) {

$e_{p-1} = minf_{p-1} + j * step_{p-1}$

```
        …
                For (k, 0 to max_k) {
                        e_2 = minf_2 + k * step_2
                        Solve (P)
                        Next k
                …
        Next j
    Next i
```

όπου έχουμε υποθέσει ότι η f_2 βρίσκεται στον πιο εμφωλευμένο βρόχο.

8. Στην επόμενη επανάληψη (αφού οι δείκτες των βρόχων έχουν αυξηθεί κατάλληλα), έστω ότι οι δείκτες έχουν τιμές j_p*, j_{p-1}*, …, j_2*. Ελέγχουμε την θέση flag[j_p*][j_{p-1}*]…[j_2] του (p-1)-διάστατου πίνακα μνήμης (που πριν την έναρξη της διαδικασίας ήταν αρχικοποιημένος με μηδενικά, αλλά σταδιακά ορισμένες θέσεις του όπως είδαμε σε προηγούμενο βήμα αλλάζουν τιμή ώστε μελλοντικά όταν φτάσουμε σε αυτές να γλιτώσουμε αχρείαστες βελτιστοποιήσεις). Αν η θέση αυτή εμπεριέχει τον αριθμό μηδέν τότε αυτό μας υποδεικνύει ότι πρέπει να κάνουμε την βελτιστοποίηση, αλλιώς αν η τιμή είναι μη μηδενική τότε η βελτιστοποίηση δεν πραγματοποιείται και προχωράμε πάνω στο πλέγμα τόσα βήματα παρακάτω όσα μας υποδεικνύει ο μη μηδενικός αριθμός.

Υποθέτοντας τώρα ότι διαθέτουμε την απαιτούμενη χωρητικότητα για να αποθηκεύσουμε στον πίνακα flag τις τιμές των b_i κατά την βελτιστοποίηση h, και ορίζοντας ως *γνήσια* κάθε βελτιστοποίηση που δίνει μια πλειάδα λύσεων διαφορετική από εκείνη που θα πρόκυπτε αν μειώναμε κατά μία μονάδα οποιοδήποτε από τα e_i (αλλά ένα και μόνο ένα από αυτά), τότε το πλήθος των παραπάνω βελτιστοποιήσεων που γλιτώνει η AUGMECON-R σε σχέση με την AUGMECON-2 είναι:

$$\sum_{h\in D}\{b_{3,h}\}, if\ p=3$$

$$\sum_{h\in D}\{b_{3,h}+b_{4,h}*(b_{3,h}+1)\}, if\ p=4$$

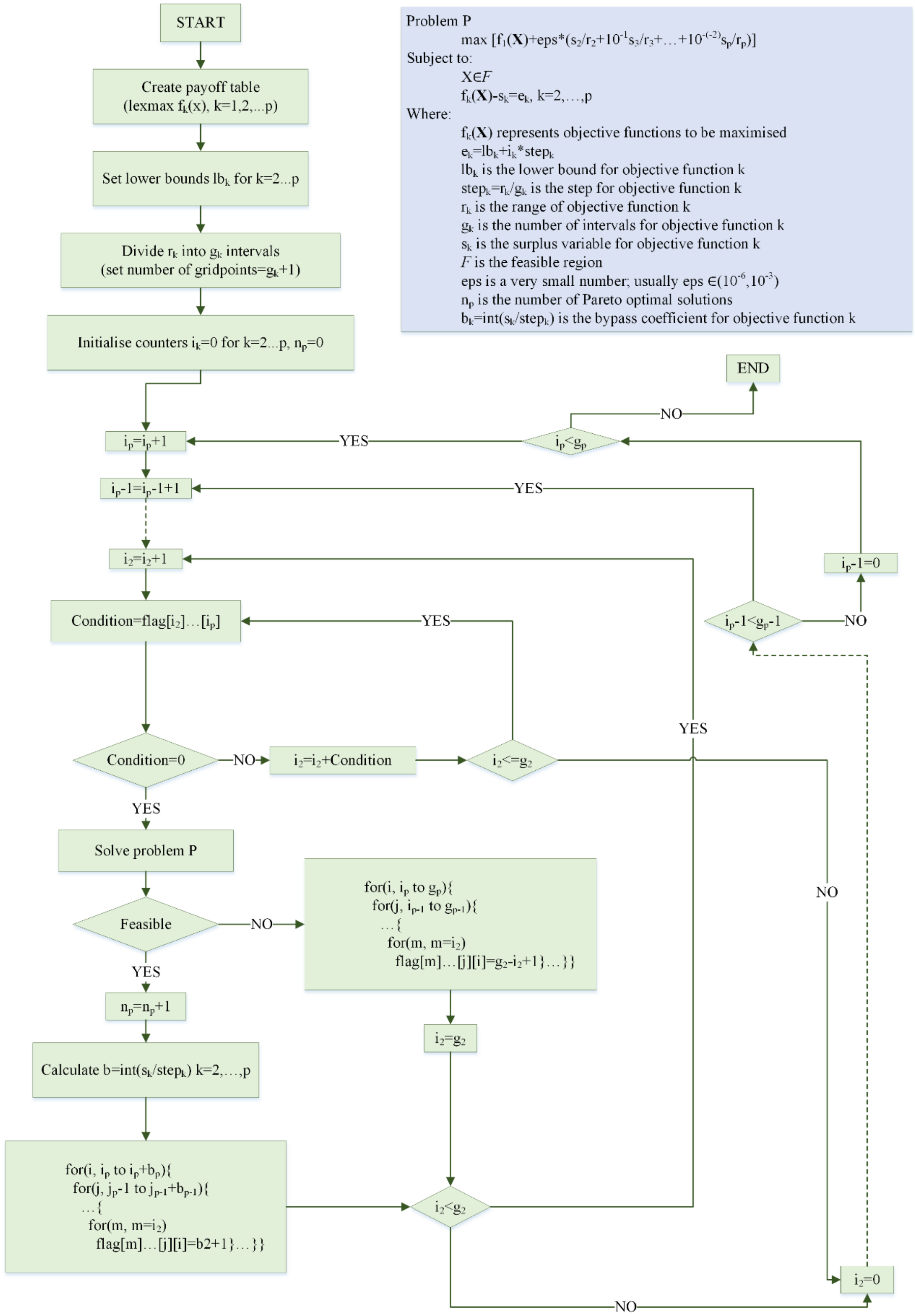
$$\sum_{h\in D}\{b_{3,h}+b_{4,h}*(b_{3,h}+1)+b_{5,h}*(b_{4,h}+1)(b_{3,h}+1)\}, if\ p=5$$

$$\sum_{h\in D}\{b_{3,h}+b_{4,h}(b_{3,h}+1)+b_{5,h}(b_{4,h}+1)(b_{3,h}+1)+\cdots \\ +b_{p,h}(b_{p-1,h}+1)(b_{p-2,h}+1)\dots(b_{3,h}+1)\}, if\ p\geq 6$$

όπου h μια γνήσια βελτιστοποίηση και D το σύνολο των γνήσιων βελτιστοποιήσεων.

Η AUGMECON-R μπορεί ταυτόχρονα να αποφύγει βελτιστοποιήσεις που θα κατέληγαν σε αδύνατη λύση. Αν για παράδειγμα για τιμές e_2*, e_3*,…, e_p* των δεξιών μελών των περιορισμών προκύψει αδύνατη λύση, τότε οποιαδήποτε αύξηση ενός εκ των e_2, e_3, …, e_p με όλα τα υπόλοιπα ίσα η μεγαλύτερα από e_2*, e_3*,…, e_p* θα οδηγήσει επίσης σε αδύνατη λύση. Με άλλα λόγια, αν για ένα σημείο του πλέγματος, το μοντέλο είναι αδύνατο να επιλυθεί, τότε προφανώς για κάθε βήμα πάνω στο πλέγμα (με συνέπεια οι περιορισμοί να γίνουν πιο αυστηροί) η βελτιστοποίηση θα είναι έτσι κι αλλιώς αδύνατη, και επομένως δεν υπάρχει λόγος να πραγματοποιηθεί. Συνεπώς, για $\delta_i \in N$, κάθε $\{e_i{}^* + \delta_i\}$ συνδυασμός των δεξιών μελών των περιορισμένων αντικειμενικών θα έδινε αδύνατη λύση. Η AUGMECON-R γλιτώνει όλες αυτές τις περιττές βελτιστοποιήσεις για $\left\{e_2{}^* + \delta_2, e_3{}^* + \delta_3, \dots, e_p{}^* + \delta_p\right\}$ ενώ η AUGMECON-2 θα απέφευγε εκείνες που αφορούσαν τις περιπτώσεις $\left\{e_2{}^* + \delta_2, e_3{}^*, \dots, e_p{}^*\right\}, \delta_2 > 0$.

Παρακάτω παρουσιάζεται και το διάγραμμα ροής της AUGMECON-R:



Βιβλιογραφία

- Mavrotas, G. (2009). Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. Applied mathematics and computation, 213(2), 455-465.

- Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. Applied Mathematics and Computation, 219(18), 9652-9669.
- Hwang, C. L., Paidy, S. R., Yoon, K., & Masud, A. S. M. (1980). Mathematical programming with multiple objectives: A tutorial. Computers & Operations Research, 7(1-2), 5-31.
- Florios, K., & Mavrotas, G. (2014). Generation of the exact pareto set in multi-objective traveling salesman and set covering problems. Applied Mathematics and Computation, 237, 1-19.
- Nikas, A., Fountoulakis, A., Forouli, A., & Doukas, H. (2020). A robust augmented ε-constraint method (AUGMECON-R) for finding exact solutions of multi-objective linear programming problems. Operational Research, 1-42.
- Forouli, A., Gkonis, N., Nikas, A., Siskos, E., Doukas, H., & Tourkolias, C. (2019). Energy efficiency promotion in Greece in light of risk: Evaluating policies as portfolio assets. Energy, 170, 818-831.
- Forouli, A., Doukas, H., Nikas, A., Sampedro, J., & Van de Ven, D. J. (2019). Identifying optimal technological portfolios for European power generation towards climate change mitigation: A robust portfolio analysis approach. Utilities Policy, 57, 33-42.
- Van de Ven, D. J., Sampedro, J., Johnson, F. X., Bailis, R., Forouli, A., Nikas, A., ... & Doukas, H. (2019). Integrated policy assessment and optimisation over multiple sustainable development goals in Eastern Africa. Environmental Research Letters, 14(9), 094001.
- Forouli, A., Nikas, A., Van de Ven, D. J., Sampedro, J., & Doukas, H. (2020). A multiple-uncertainty analysis framework for integrated assessment modelling of several sustainable development goals. Environmental Modelling & Software, 131, 104795.