

# Φιλτράρισμα και Κατάταξη Επίπεδης Ιμαντοκίνησης σε μικρή Ανεμογεννήτρια Κάθετου Άξονα με έμφαση στην γενικοποίηση αντίστοιχων προβλημάτων

Δημήτρης Σπαθούλας

21 Απριλίου 2023

## Περίληψη

Η παρούσα εργασία υλοποιήθηκε στα πλαίσια του μαθήματος Συστήματα Μετάδοσης Κίνησης 8ο εξάμηνο 2023.

Η παρούσα αναφορά διατίθεται ως μια περίληψη επάνω στα προγράμματα που δημιουργήθηκαν.

## 1 Εισαγωγή

Η εργασία έχει ως σκοπό την έρευνα και την ανάλυση του καλύτερου είδους και τύπου επίπεδου ιμάντα με χρήση φίλτρων εύρωστων υπερπαραμέτρων τροφοδοτούμενα από τον χρήστη για τον εκάστοτε σκοπό και τύπο μετάδοσης. Η ιμαντοκίνηση που χρησιμοποιείται αυτή τη στιγμή επάνω στην ANM ( ανεμογεννήτρια ) είναι τραπεζοειδής, οπότε θεωρήθηκε ότι μια αντίστοιχη έρευνα στον χώρο των εναλλακτικών του είδους αυτού έγινε ήδη.

Μέσα από μη καταστροφικές μετρήσεις που διεκπεραιώθηκαν υπολογίστηκαν οι μέσες τιμές των κυριότερων χαρακτηριστικών του. Αυτές είναι οι εξής ( με ένα σφάλμα  $\pm 5$  ) :

- Διάκεντρος : 350 χιλιοστά
- Διάμετρος μικρής τροχαλίας (ηλεκτρογεννήτριας) : 40 χιλ
- Διάμετρος μεγάλης τροχαλίας (κινητήρια) : 300 χιλ
- Κάθετο Μήκος φτερού : 0.6 μέτρα
- Πλάτος φτερού ANM : 25 δεκατόμετρα
- Β τροχαλίας : 20 χιλιοστά
- Πάχος ιμάντα : 0.56 εκατοστά

Αυτά τα μεγέθη θεωρούνται εμπειρικά για την συγκεκριμένη κατασκευή και η λύση που πρόκειται να διερευνηθεί θα βρίσκεται γύρω από αυτές τις τιμές, ως πεδία ορισμού.

## 2 Υπολογισμός και Κατωφλίωση Βασικών Εμπειρικών και Σχεδιαστικών Παραμέτρων

Παρακάτω γίνεται ανάλυση και αναφορά σε κάποιες βασικές υποθέσεις και περιορισμούς, για την μορφή και τον τύπο της ANM.

## 2.1 Υπολογισμός Στρεπτικής Ροπής Άξονα ANM

Γνωρίζοντας εμπειρικά την διάμεσο και την μέση τιμή της ταχύτητας του ανέμου για διάφορες εποχές της Ξάνθης τα τελευταία 30 χρόνια, θεωρείται μια ταχύτητα ανέμου 15 χιλ/ώρα, δηλαδή 4.5 μέτρα το δευτερόλεπτο ( αυτή είναι μια μέση τιμή που οπωσδήποτε η ANM θα πρέπει να λειτουργεί ομαλά ). Η μέση πυκνότητα του αέρα στο εύρος θερμοκρασίας της Ξάνθης είναι 1.225 χιλιόγραμμα ανά κυβικό μέτρο ( στατιστική ). Θεωρείται ότι ο αέρας “πέφτει” κάθετα και ομοιόμορφα πάνω στην επιφάνεια κάθετη της διεύθυνσης του, άρα ο συντελεστής οπισθέλκουσας υπολογίζεται ως 1.05.

$$F_D = \frac{1}{2} \rho v^2 C_D A \quad (1)$$

Οι υπερπαράμετροι στο πρόβλημα στο συγκεκριμένο σημείο είναι συντελεστές που προσθέτονται στην ροή του αλγόριθμου με σκοπό την μεγιστοποίηση πιθανόν σφαλμάτων και τοπικών ορίων, στην λειτουργία της γεννήτριας και στην μετάδοση της ροπής στον άξονα. Παρακάτω δίνεται ο αναφερόμενος κώδικας στην C.

---

```
// Spathoulas Dimitris 3/2023
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <malloc.h>
#define P_ave 1.225 // g/m3
#define vel_air 20.0 // km/h
#define Cd 1.05 // orthog region
#define Length_w 0.6 // m
#define Windth_w 0.25 // of one m
#define my_corr 5
#define my_c 1.4
double Fd;
double Td;
double Pr;
double vel;
double A;
double find_vel(double v){

    return v=(v*1000.0)/3600.0;
}

double area(double l , double w , double c){
    return (l*w*c);
}

int main(){

    vel=find_vel(vel_air);
    printf("vel is %5.2lf",vel);
    A = area(Length_w,Windth_w,my_corr);
    printf("\nA is %5.2lf",A);
    Fd = (0.5)*pow(vel,2.0)*P_ave*Cd*A;
    printf("\nforce is : %5.3lf",Fd);
    Td = Fd*(Windth_w*my_c*(0.5));
    printf("\nTd is : %5.2lf",Td);
    return 0;
}
```

---

Η έξοδος που θα μας δοθεί και η τιμή που ψάχνουμε απο αυτή τη ροή είναι η στρεπτική ροπή σε μονάδες διεθνούς συστήματος που μεταφέρεται απο τα φτερά της ανεμογεννήτριας στον άξονα της.

## 2.2 Το υποσύνολο λύσεων υπερπαραμέτρων - Η εξόρυξη γεννητόρων των εναλλακτικών του προβλήματος

Γνωρίζοντας ή θεωρώντας εμπειρικά ή πειραματικά θα χρειαστεί το πρόβλημα τύπου NP hard , να μειωθεί σε ένα αρκετά ορατό και αντιληπτό εύρος πιθανών λύσεων μέσα απο οριοθέτηση σε ακρότατα παραμέτρων του εκάστοτε συγκεκριμένου ασαφούς προβλήματος.

Στο συγκεκριμένο πρόβλημα θεωρήθηκαν κάποιες μέσες τιμές απο την ήδη υπαρκτή λύση που υπάρχει, άλλες όπως η γωνιακή ταχύτητα του κάθε άξονα παραμετροποιήθηκε με βάση το είδος, την χρήση και την μορφή του του συστήματος, κάτι περισσότερο ποιοτικό. Οι βασικότερες παράμετροι του προβλήματος , οι διάμετροι κάθε τροχαλίας τροφοδοτήθηκαν στο πρόβλημα ως σταθερές μέσα σε λίστες, έτσι ο έλεγχος και το φιλτράρισμα γίνεται σε  $K \cdot \Lambda$  συνδυασμούς τροχαλιών.

Κάθε μοναδικός συνδυασμός περνάει από διάφορα εύρωστα φίλτρα ορίων, κάτι που μειώνει ωστόσο δραστικά το σύνολο των τιμών των παραμέτρων και υπερπαραμέτρων που θα δοθούν στο επόμενο βήμα. Έτσι το πρόβλημα κατέληξε στην παρακάτω αλγοριθμική μορφή.

---

```
// Spathoulas Dimitris 3/2023
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <malloc.h>
#define N 21
#define M 21
#define NM 5
#define K 5
#define Td 10.0 // Nm
#define vel_sub 0.9
#define a_par_min 0.3 // m
#define a_par_max 0.7 // m
#define M_PI 3.141592
#define vel_max 40.0 // m/s
#define L_max 1.3 // m
#define n2_min 720.0 // rpm
#define n1_max 140.0 // rpm giati den ua antejei kai poly parapano
#define ang_vel_1_max 13.0 // rad/s 12.56
char m[64];
char name[64];
FILE *fptr=NULL;
int i,j,k,v;
double dd_1[N] = {0.08,0.09,0.1,0.112,0.125,0.14,0.16,
0.18,0.2,0.224,0.25,0.28,0.315,0.355,0.4,0.45,0.5,0.56,0.63,0.71,0.8};
double dd_2[M] = {0.0315,0.0355,0.04,0.045,0.05,0.056,0.063,0.071,
0.08,0.09,0.1,0.112,0.125,0.14,0.16,0.18,0.2,0.224,0.25,0.28,0.315};
double bbtr[NM] = {0.025,0.032,0.04,0.05,0.063};
double bb[NM] = {0.02 , 0.025 , 0.032, 0.04 , 0.05};
double trans;
int kk=0;
double L;
double s;
double f;
double velocity;
double a;
double ang_vel;

int main() {
    printf("Create a NEW file for data: ");
    scanf("%s",m);
    sprintf(name,"%s.txt",m);
    fptr=fopen(name,"w");

    if (fptr==NULL) {
```

```

    printf("\nfile couldnt be created! \n");
}

else {
    for (i=0;i<N;i++) {
        for (j=0;j<M;j++) {
            trans = (dd_1[i]/dd_2[j]);
            if ( (n2_min/trans) < n1_max ) { // exo mperdeytei edo ligo
                a = fmin(((1.5)*(dd_1[i]+dd_2[j])),fmax(
                    ((1.5*dd_2[j])+(dd_1[i]/2.0)),dd_1[i]));
                if ( (a_par_min<a) && (a_par_max>a) ) {
                    L = ( (a*2.0) + (M_PI*((dd_1[i]+dd_2[j])/2.0)) + (
                        (1.0/a)*pow((dd_1[i]-dd_2[j])/2.0,2.0) ) );
                    if (L < L_max) {
                        f = M_PI - (2.0*asin((dd_1[i]-dd_2[j])/(2.0*a))); // rads
                        ang_vel=((2.0*M_PI)*(n2_min/(trans*60.0)));
                        velocity =ang_vel*(dd_1[i]/2.0);
                        printf(" vel = %3.3lf ",velocity);
                        if ((velocity<vel_max)&&(ang_vel<=ang_vel_1_max)) {
                            fprintf(fptr,"%3.3lf m || %3.3lf m || a = %3.4lf m || trans = %3.3lf
                                || L = %4.4lf m || f = %4.4lf rad || vel = %4.4lf || ang_vel =
                                    %4.4lf \n",dd_1[i],dd_2[j],a,trans,L,f,velocity,ang_vel);
                            kk++; }
                        }
                    }
                }
            }
            s=0.0;
            trans= 0.0;
            a = 0.0;
            f = 0.0;
            L =0.0;
            velocity = 0.0;
            ang_vel=0.0;
        }
    }
    fclose(fptr);
    printf("\n\n%3d",kk);

return 0;
}

```

Ακριβώς παρακάτω γίνεται εμφάνιση των πέντε διαφορετικών μεταδόσεων με τις αντίστοιχες παραμέτρους που μπορούν να αντιστοιχισθούν με τον αντίστοιχο κώδικα.

Listing 1: Generators

---

```

0.315 m || 0.032 m || a = 0.3150 m || trans = 10.000 || L = 1.2381 m || f = 2.2081 rad ||
    vel = 1.1875 || ang_vel = 7.5398
0.315 m || 0.035 m || a = 0.3150 m || trans = 8.873 || L = 1.2426 m || f = 2.2223 rad || vel
    = 1.3383 || ang_vel = 8.4973
0.315 m || 0.040 m || a = 0.3150 m || trans = 7.875 || L = 1.2477 m || f = 2.2382 rad || vel
    = 1.5080 || ang_vel = 9.5744
0.315 m || 0.045 m || a = 0.3150 m || trans = 7.000 || L = 1.2533 m || f = 2.2558 rad || vel
    = 1.6965 || ang_vel = 10.7712
0.315 m || 0.050 m || a = 0.3150 m || trans = 6.300 || L = 1.2591 m || f = 2.2733 rad || vel
    = 1.8850 || ang_vel = 11.9680

```

---

Η λίστα για το τί είναι κάθε τιμή είναι η εξής ( το πρώτο στοιχείο της λίστας είναι αυτό στην πρώτη στήλη κάθε γραμμής, και ούτω καθεξής)

- Διάμετρος κινητήριας τροχαλίας (γενικά της πρώτης που επιλέγουμε)

- Διάμετρος μικρής τροχαλίας
- Διάκεντρος
- Μετάδοση
- Ανοιγμένο μήκος
- γωνία περιέλιξης σε ακτίνια
- Εφαπτόμενη ταχύτητα
- Γωνιακή ταχύτητα

Όπως είναι εμφανές το πρόγραμμα αυτό δίνει και μια απο τις βασικές παραμέτρους του επίπεδου ιμάντα. Το μήκος του.

### 3 Το σύνολο των εναλλακτικών

Μέχρι τώρα έγιναν οι υπολογισμοί των παραμέτρων για την έρευνα ενός υποσυνόλου γεννητορων από το τεράστιο σύνολο εναλλακτικών δερμάτινων και υφασμάτων ιμάντων. Με άλλα λόγια μέχρι τώρα κλείναμε το πρόβλημα σε όρια απο τα οποία θα γεννηθούν οι εναλλακτικές.

Σε αυτό το κομμάτι γίνεται το πέρασμα των υπερπαραμέτρων και παραμέτρων του προβλήματος στην πλήρη και λεπτομερή ανάλυση για την κάθε κλάση ιμάντα.

Σε αυτό το κομμάτι θα εμφανιστεί μόνο η κλάση της βάσης δεδομένων και πως περνάνε σε ένα πλήρη αυτοματοποιημένο και γενικό σχέδιο. Στο κομμάτι που δεν εμφανίζεται υπολογίζονται όλες οι υπόλοιπες παράμετροι μαζί με τους εκάστοτε περιορισμούς πχ πάχος.

Εάν υπάρχει ανάγκη για τον έλεγχο αυτού του κομματιού, πολύ ευχαρίστως μπορώ να κάνω μια λεπτομερή επίδειξη.

Οι ιμάντες που διερευνώνται ως προτεινόμενες λύσεις είναι :

- Δέρμα HG
- Δέρμα H
- Δέρμα F-S
- Gummi-Baumwolle
- Balata-Baum

---

```
// Spathoulas Dimitris 3/2023
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <malloc.h>
#define n 8
#define M 30
#define NM 5
#define K 5
#define factor 1.0*pow(10.0,6.0)
#define omagad 13
#define M_E 2.7182819
#define yoaga 64
#define grav 9.81

// HYPERPAR
#define KG_MAX 0.4
#define vel_mx_f 4.5
#define Td 10.0 // Nm
#define H_min 15000.0
```

```

#define vel_sub 0.67
#define safety_f 0.85
double bbtr[NM] = {0.025,0.032,0.04,0.05,0.063};
double bb[NM] = {0.02 , 0.025 , 0.032, 0.04 , 0.05};
FILE *fptr;
FILE *fptr1;
int m;
char name[yoaga];
int i,j,x;
char name1[yoaga];
int counter = 0;
struct ttls {
    double obj[n];
    // dd_1[i], dd_2[j], a, trans, L, f, velocity, ang_vel
};
struct ttls alts[M];

struct typ {

    double Eb; // N/mm2
    double gamma; // kg/m3
    double Sall; // N/mm2
    double dsm;
    double bmax;
    double velmax;
    double s_min;
    double s_max;
    double trib;
    double scor;
};
struct typ type[14];

double s=0.0;
double e =0.0;
double Sf=0.0;
double S_1=0.0;
double S_2= 0.0;
double S_0 = 0.0;
double ss_0=0.0;
double s_b1=0.0;
double s_k=0.0;
double s_u=0.0;
double seq=0.0;
double H = 0.0;
double B = 0.0;
int test = 0;
double KG = 0.0;
int main() {

    do {
        printf("Please enter number of alternatives: ");
        scanf("%d",&m); }
    while (!(m<M) );

    // FILE OPEN
    printf("\nWrite name of txt file containing values: ");
    scanf("%s",&name);

    fptr=fopen(name,"r");
    if (fptr==NULL) {
        printf("\nFile couldnt be found! \n");
    }

```

```

// TXT VALUES INPUT
else {
    test = 1;
    for (i=0;i<m;i++){
        for (j=0;j<n;j++){
            fscanf(fptr,"%lf",&alts[i].obj[j]);
            printf("%4.3lf ",alts[i].obj[j]); } printf("\n"); }
    }
fclose(fptr);

if (test==1) {
    test = 0;
    type[0].Eb=3.0;
    type[1].Eb=5.0;
    type[2].Eb=7.0;
    type[0].dsm=20.0;
    type[1].dsm=25.0;
    type[2].dsm=35.0;

    type[3].Eb=4.0;
    type[4].Eb=6.0;
    type[5].Eb=8.0;
    type[3].dsm=25.0;
    type[4].dsm=30.0;
    type[5].dsm=40.0;

    type[6].Eb=5.0;
    type[7].Eb=7.0;
    type[8].Eb=9.0;
    type[6].dsm=30.0;
    type[7].dsm=35.0;
    type[8].dsm=45.0;

    type[9].bmax=7.5;
    type[10].bmax=7.5;
    type[11].bmax=25.0;
    type[9].s_min=0.003;
    type[9].scor=1.3;
    type[9].s_max=0.007;
    type[10].s_min=0.003;
    type[10].scor=1.1;
    type[10].s_max=0.007;
    type[11].s_min=0.003;
    type[11].scor=0.7;
    type[11].s_max=0.007;

    type[12].bmax=7.5;
    type[13].bmax=25.0;
    type[12].s_min=0.003;
    type[12].scor=1.2;
    type[12].s_max=0.008;
    type[13].s_min=0.003;
    type[13].scor=0.6;
    type[13].s_max=0.008;

    for (i =0;i<9;i++) {
        type[i].scor=1.0;
    }

    for (i=0;i<3;i++) {
        type[i].Sall=0.44;
        type[i].bmax=25.0;

```

```

    type[i].gamma=900.0;
    type[i].velmax=50.0*vel_sub;
    type[i].trib=0.3 + ( (type[i].velmax/vel_mx_f)/100.0); }

for (i=3;i<6;i++) {
    type[i].Sall=0.44;
    type[i].bmax=10.0;
    type[i].gamma=950.0;
    type[i].velmax=40.0*vel_sub;
    type[i].trib=0.3 + ( (type[i].velmax/vel_mx_f)/100.0); }

for (i=6;i<9;i++) {
    type[i].Sall=0.39;
    type[i].bmax=5.0;
    type[i].gamma=1000.0;
    type[i].velmax=30.0*vel_sub;
    type[i].trib=0.3 + ( (type[i].velmax/vel_mx_f)/100.0); }

for (i=9;i<12;i++) {
    type[i].Sall=0.39;
    type[i].gamma=1200.0;
    type[i].velmax=40.0*vel_sub;
    type[i].trib=0.5;
    type[i].dsm=30.0; }

for (i=12;i<14;i++) {
    type[i].Sall=0.44;
    type[i].gamma=1250.0;
    type[i].velmax=40.0*vel_sub;
    type[i].trib=0.5;
    type[i].dsm=25.0; }

for(i=0;i<8;i+=3) {
    type[i].s_min=0.003;
    type[i].s_max=0.007;
}
for(i=1;i<9;i+=3) {
    type[i].s_min=0.008;
    type[i].s_max=0.012;
}
for(i=2;i<10;i+=3) {
    type[i].s_min=0.014;
    type[i].s_max=0.020;
}
for (i=9;i<14;i++){
    type[i].Eb=5.0;
}

```

---

Τελικά το παραπάνω πρόγραμμα θα δώσει ως έξοδο 8 παραμέτρους σε κάθε γραμμή που θα θεωρηθούν κριτήρια για την αξιολόγηση και κατάταξη του καλύτερου ιμάντα με βάση την αντίληψη του υποκειμένου χρήστη.

Η λίστα για το τί είναι κάθε τιμή είναι η εξής ( το πρώτο στοιχείο της λίστας είναι αυτό στην πρώτη στήλη κάθε γραμμής, και ούτω καθεξής)

- Διάμετρος κινητήριας τροχαλίας (γενικά της πρώτης που επιλέγουμε)
- Διάμετρος μικρής τροχαλίας
- Πλάτος
- Πάχος
- Ανοιγμένο μήκος



- Μάζα ιμάντα ( εδώ μπορεί να αλλάζει η χρηματική μονάδα ανα κιλό για κάθε ιμάντα, αλλά δεν υπήρχαν σχετικά δεδομένα )
- Αντοχή Ιμάντα ( θα μπορούσε να είναι και το πηλίκο με το όριο )
- Οι συνολικές ώρες λειτουργίας

Η κατάταξη γίνεται με την υιοθέτησή της αναλυτικής ιεραρχικής διαδικασίας AHP και την μέθοδο ιδεατών σημείων TOPSIS .

Παρακάτω φαίνεται ένα υποσύνολο των εναλλακτικών του προβλήματος. Το κενό και ο αριθμός υποδεικνύει αλλαγή τύπου ιμάντα από τους 5 καθώς και το σύνολο των εναλλακτικών μέσα στο υποσύνολο αυτό. Οι εναλλακτικές που θεωρήθηκαν καλές έως αυτό το σημείο με όλα τα παραπάνω φίλτρα και υπερπαραμέτρους είναι 170.

## Listing 2: ALTERNATIVES

0

0

0.3150		0.0320		0.0200		0.0060		1.2381		0.1486		0.1956		34317.9038
0.3150		0.0320		0.0250		0.0060		1.2381		0.1857		0.1756		65649.2316
0.3150		0.0320		0.0320		0.0060		1.2381		0.2377		0.1580		123447.3014
0.3150		0.0320		0.0400		0.0060		1.2381		0.2971		0.1455		202630.2988
0.3150		0.0320		0.0500		0.0060		1.2381		0.3714		0.1355		310938.2106
0.3150		0.0350		0.0200		0.0060		1.2426		0.1491		0.1953		30877.9198
0.3150		0.0350		0.0250		0.0060		1.2426		0.1864		0.1753		58986.8534
0.3150		0.0350		0.0320		0.0060		1.2426		0.2386		0.1578		110753.0356
0.3150		0.0350		0.0400		0.0060		1.2426		0.2982		0.1453		181558.6708
0.3150		0.0350		0.0500		0.0060		1.2426		0.3728		0.1354		278267.7730
0.3150		0.0400		0.0200		0.0060		1.2477		0.1497		0.1949		27827.8017
0.3150		0.0400		0.0250		0.0060		1.2477		0.1872		0.1750		53076.9026
0.3150		0.0400		0.0320		0.0060		1.2477		0.2396		0.1576		99487.7158
0.3150		0.0400		0.0400		0.0060		1.2477		0.2994		0.1452		162853.2687
0.3150		0.0400		0.0500		0.0060		1.2477		0.3743		0.1352		249258.9996
0.3150		0.0450		0.0200		0.0060		1.2533		0.1504		0.1945		25148.6422
0.3150		0.0450		0.0250		0.0060		1.2533		0.1880		0.1747		47883.3976
0.3150		0.0450		0.0320		0.0060		1.2533		0.2406		0.1574		89584.0798
0.3150		0.0450		0.0400		0.0060		1.2533		0.3008		0.1450		146404.2647
0.3150		0.0450		0.0500		0.0060		1.2533		0.3760		0.1351		223744.1992
0.3150		0.0500		0.0200		0.0060		1.2591		0.1511		0.1941		23007.4613
0.3150		0.0500		0.0250		0.0060		1.2591		0.1889		0.1744		43729.5035
0.3150		0.0500		0.0320		0.0060		1.2591		0.2417		0.1572		81657.0935
0.3150		0.0500		0.0400		0.0060		1.2591		0.3022		0.1449		133231.1519
0.3150		0.0500		0.0500		0.0060		1.2591		0.3777		0.1350		203301.9055

25

0.3150		0.0320		0.0200		0.0066		1.2381		0.1961		0.1777		61066.8325
0.3150		0.0320		0.0250		0.0066		1.2381		0.2451		0.1631		101959.2319
0.3150		0.0320		0.0320		0.0066		1.2381		0.3138		0.1504		166018.6498
0.3150		0.0320		0.0400		0.0066		1.2381		0.3922		0.1413		241363.1772
0.3150		0.0350		0.0200		0.0066		1.2426		0.1968		0.1775		54706.1135
0.3150		0.0350		0.0250		0.0066		1.2426		0.2460		0.1630		91238.1942

....

## 4 Πολυκριτήριος Προγραμματισμός

Για την κατανόηση του σχεδίου αντιμετώπισης του πλέον προβλήματος, παρακάτω εμφανίζονται πίνακες των αποτελεσμάτων με κάποια λόγια.

**\*ΣΗΜΕΙΩΣΗ\*** : Οι παρακάτω αλγόριθμοι που χρησιμοποιούνται είναι ελεγμένοι σε ακαδημαϊκό επίπεδο ως την ορθότητα των αποτελεσμάτων τους σε χρήση τους σε άλλα αντίστοιχα προβλήματα.

### 4.1 Η δημιουργία και ο υπολογισμός κύριου ιδιοδιανύσματος για παγκόσμια και τοπικά κριτήρια και υποκριτήρια

Παρακάτω εμφανίζονται τα βάρη των κριτηρίων και υποκριτηρίων.

Listing 3: GLOBAL WEIGHTS

crit_1	crit_2
0.250	0.750

Listing 4: LOCAL CHAR WEIGHTS

crit_1	crit_2	crit_3	crit_4	crit_5
0.206	0.206	0.110	0.110	0.368

CL is : 0.003  
CR is : 0.0027

Listing 5: LOCAL DER WEIGHTS

crit_1	crit_2	crit_3
0.400	0.200	0.400

Κάθε κριτήριο θεωρήθηκε φθίνων ( όσο μικρότερο τόσο καλύτερο ) εκτός του τελευταίου που είναι ο συνολικός χρόνος λειτουργίας. Ο τρόπος δημιουργίας αυτών των πινάκων και σχέσεων έγκειται στην υποκειμενική και εμπειρική γνώση και άποψη του σχεδιαστή ανάλογα το πρόβλημα.

Κατά τον δικό μου σχεδιασμό χώρισα τα κριτήρια σε 2 κατηγορίες η μία είναι οι γεωμετρικές, ενώ η άλλη περιέχει τιμές που υπολογίστηκαν από αυτές.

Οι τιμές κάτω στον φάκελο με τα 5 κριτήρια υποδεικνύουν τον βαθμό ασυνέπειας των τιμών λόγω υποκειμενικότητας στην θεμελιώδη κλίμακα προτίμησης, είναι κάτω του 10 της εκατό, άρα είναι αποδεχτή.

### 4.2 Ο χώρος των εναλλακτικών και η καλύτερη λύση

Τελικά, όλα τα παραπάνω βήματα και ο τρόπος της λύσης που παραδίδω βασίζεται πάρα πολύ στην έννοια της ευρωστίας και της στοχαστικότητας. Η λύση που βρίσκω είναι σίγουρα από τις καλύτερες στον χώρο των λύσεων σε ένα τόσο μεγάλο πρόβλημα αλλά σίγουρα με σαφείς περιορισμούς και εμπειρική γνώση πάνω σε άλλα προβλήματα.

Το τελικό αρχείο που περνάει στο πρόγραμμα κατάταξης είναι το εξής (δεν εμφανίζεται ολόκληρο, λόγω χώρου).

Listing 6: DECISION MATRIX

0	0	0	0	0	0	0	1
0.0515	0.0515	0.0275	0.0275	0.092	0.3	0.15	0.3
0.3150	0.0320	0.0200	0.0060	1.2381	0.1486	0.1956	34317.9038
0.3150	0.0320	0.0250	0.0060	1.2381	0.1857	0.1756	65649.2316
0.3150	0.0320	0.0320	0.0060	1.2381	0.2377	0.1580	123447.3014

0.3150	0.0320	0.0400	0.0060	1.2381	0.2971	0.1455	202630.2988
0.3150	0.0320	0.0500	0.0060	1.2381	0.3714	0.1355	310938.2106
....							

Παρακάτω γίνεται η εμφάνιση των αποτελεσμάτων για διαφορετικούς συντελεστές P στον υπολογισμό των αποστάσεων των εναλλακτικών από την ιδεατή και Άντι-ιδεατή λύση. καθώς και η τιμή της συγγενούς εγγύτητας.

Μια χοντρική εικόνα : όσο μεγαλύτερο το P τόσο πιο πλαστική η λύση (καταλήγουμε σε κάτι απόλυτο με βάση την γενική εικόνα των κριτηρίων και τις τιμές των εναλλακτικών).

Όσο πιο μικρό το P, τόσο μεγαλύτερη η πιθανότητα η καλύτερη λύση να είναι αυτή γιατί έχει την καλύτερη απόκριση πχ στα δύο ή και τρία πιο σημαντικά κριτήρια από τα 8.

Όσο μεγαλύτερη η συγγενής εγγύτητα, τόσο πιο κοντά η εναλλακτική στην ιδεατή λύση και μακριά από την αντι-ιδεατή.

```
Please enter number of alternatives: 200
Please enter number of criteria: 8
Please enter number of P (double): 1.0

Write name (and destination) of txt file containing weights, type of criteria and values: decision.txt

Best Alternative(s) is (are): 73 ,
-----
Process exited after 9.218 seconds with return value 0
Press any key to continue . . .
```

Figure 1: P=1

Για P=1 η καλύτερη λύση είναι ο τύπος ιμάντα Balata-Baumwolle με τις παρακάτω παραμέτρους :  
0.3150 „ 0.0350 „ 0.0320 „ 0.0043 „ 1.2426 „ 0.2147 „ 0.1381 „ 509798.5873

```
Please enter number of alternatives: 200
Please enter number of criteria: 8
Please enter number of P (double): 2.0

Write name (and destination) of txt file containing weights, type of criteria and values: decision.txt

Best Alternative(s) is (are): 17 , 22
-----
Process exited after 11.71 seconds with return value 0
Press any key to continue . . .
```

Figure 2: P=2

Για P=2 η καλύτερη λύση είναι οι ιμάντες με παραμέτρους :

0.3150 „ 0.0450 „ 0.0250 „ 0.0060 „ 1.2533 „ 0.1880 „ 0.1747 „ 47883.3976  
0.3150 „ 0.0500 „ 0.0250 „ 0.0060 „ 1.2591 „ 0.1889 „ 0.1744 „ 43729.5035

Η έξοδος στην προκειμένη περίπτωση είναι δύο διανύσματα γιατί τα δύο αναμεταξύ τους είχαν μια πάρα πολύ κοντινή συγγενή εγγύτητα διαφοράς χιλιοστών ( ο αλγόριθμος είναι προγραμματισμένος σε τέτοια κατάσταση να δίνει και τις δύο επιλογές ).

Και οι δύο ιμάντες είναι κατηγορίας δέρματος F-S .

Για P=(10,20) η καλύτερη λύση είναι ο τύπος ιμάντα F-S με τις παρακάτω παραμέτρους :

```

Please enter number of alternatives: 200
Please enter number of criteria: 8
Please enter number of P (double): 5.0

Write name (and destination) of txt file containing weights, type of criteria and values: decision.txt

Best Alternative(s) is (are): 19 ,
-----
Process exited after 11.44 seconds with return value 0
Press any key to continue . . .

```

Figure 3: P=5

```

Please enter number of alternatives: 200
Please enter number of criteria: 8
Please enter number of P (double): 10.0

Write name (and destination) of txt file containing weights, type of criteria and values: decision.txt

Best Alternative(s) is (are): 19 ,
-----
Process exited after 14.45 seconds with return value 0
Press any key to continue . . .

```

Figure 4: P=10

0.3150 „ 0.0450 „ 0.0400 „ 0.0060 „ 1.2533 „ 0.3008 „ 0.1450 „ 146404.2647

```

Please enter number of alternatives: 200
Please enter number of criteria: 8
Please enter number of P (double): 20.0

Write name (and destination) of txt file containing weights, type of criteria and values: decision.txt

Best Alternative(s) is (are): 20 ,
-----
Process exited after 8.219 seconds with return value 0
Press any key to continue . . .

```

Figure 5: P=20

Για  $P=(20,50)$ -infh καλύτερη λύση είναι ο τύπος ιμάντα F-S με τις παρακάτω παραμέτρους :

0.3150 „ 0.0450 „ 0.0500 „ 0.0060 „ 1.2533 „ 0.3760 „ 0.1351 „ 223744.1992

Να αναφέρω ότι η επιλογή της μεθόδου για την λύση του πίνακα απόφασης θα μπορούσε να είναι και η PROMETHEE για την οποία υπάρχει αντίστοιχο πρόγραμμα που έχω αναπτύξει.

## 5 Η τελική λύση

Τελικά, για αυτό το πρόβλημα με αυτές τις παραμέτρους για αυτά τα όρια, όπως φαίνεται ο πολυκριτήριος προγραμματισμός με αυτή την αξιολόγηση βαρών κριτηρίων δίνει τις παραπάνω 5 εναλλακτικές με μεγαλύτερη έμφαση στις τελευταίες δύο για μεγάλα P εάν χρειάζεται ο ιμάντας να νε γενικά καλός στα κριτήρια που έχουμε δώσει ή στις πρώτες 3 εάν μας ενδιαφέρει ο ιμάντας να νε καλός στα κριτήρια με το μεγαλύτερο βάρος.

Εγώ θα διάλεγα τον 19ο ή τον 20ο, δηλαδή σίγουρα τύπου F-S και θα επέλεγα τον 20ο γιατί έχει περισσότερες ώρες λειτουργίας.

```

Please enter number of alternatives: 200
Please enter number of criteria: 8
Please enter number of P (double): 15.0

Write name (and destination) of txt file containing weights, type of criteria and values: decision.txt

Best Alternative(s) is (are): 20 , 19
-----
Process exited after 11.82 seconds with return value 0
Press any key to continue . . .

```

Figure 6: P=15

## 6 Επίλογος

Όλη η παραπάνω περίληψη του σχεδιασμού και του τρόπου αντιμετώπισής έχει ως βάση την γενικοποίηση ενός προβλήματος τέτοιου είδους με πιο χαλαρές παραμετροποιήσεις. Εάν ξέραμε ή θέλαμε πιο συγκεκριμένα και πράγματα, πολύ εύκολα θα μπορούσαμε να αλλάζουμε κάποιες παραμέτρους ή να παραλείψουμε κάποια φίλτρα και να τα αντικαταστήσουμε με πιο σθεναρές τιμές.

Για παράδειγμα ο αλγόριθμος μπορεί να ελέγξει εάν υπάρχουν πιθανές λύσεις πριν καν γίνει λεπτομερής ανάλυση στους ιμάντες για το εάν υπάρχει μετάδοση από 2000 στροφές σε 5000 με ένα εύρος διακέντρου 500-1500 χιλιοστά, ή ποιοι είναι οι πιθανοί συνδυασμοί τροχαλιών εάν θέλουμε το εύρος ανοιγμένου μήκους ιμάντα να είναι 2000-2030 χιλιοστά.

Τέλος μπορούμε πολύ εύκολα να αλλάζουμε τις υπερπαραμέτρους και να βρούμε τελείως διαφορετικά αποτελέσματα ανάλογα την δικιά μας υποκειμενική αντίληψη.

Εγώ πχ θεώρησα έναν συντελεστή ασφαλείας για τον ιμάντα μικρότερο του δεδομένου γιατί είχα το παράθυρο της δυνατότητας αφού οι αναπτυσσόμενες δυνάμεις είναι πολύ μικρές, άρα και οι ιμάντες θα είναι πιο κοντά σε ένα ιδανικό κατώφλι προτίμησης, αλλά και η διάρκεια ζωής σε ώρες θα είναι μεγαλύτερη εάν δεν υπάρξει κάποιος καταστροφικός παλμός ή κάτι ανάλογο.

Με πάρα πολύ εύκολο τρόπο επίσης μπορεί να αλλάχθει και το ποιά αλλά και πόσα είναι τα τελικά κριτήρια. Το μόνο σχετικά πλαστικό σε αλλαγές κομμάτι είναι η τελική αξιολόγηση των 5 ιμάντων καθώς η δημιουργία μίας ακόμα κλάσης ή ακόμα και η προσθήκη άλλων ειδών ιμάντων θα απαιτεί πολύ πιθανόν μια διαφορετική επιπρόσθετη προσέγγιση στην ήδη δομημένη, καθώς και η εισαγωγή των σταθερών απαιτεί μια καλή αντίληψη στον χώρο των κλάσεων.

Αυτό που λέω εδώ είναι εάν χρειαστεί να προστεθεί μια ή δύο ή ν κλάσεις γραναζιών, καλό θα ήταν επειδή τα γρανάζια είναι ένα άλλο σώμα να δημιουργηθεί ένα ανεξάρτητο πρόγραμμα για εκείνα, εάν χρειαστεί να γίνει έλεγχος και για αλυσίδες ή επίπεδους ιμάντες πολύ πιθανό να γίνεται να ενταχθούν στο γενικό σύνολο του ήδη δομημένου προγράμματος αλλά με πάρα πολύ πιο προσεκτική και πολύπλοκη προσέγγιση, κάτι που έγκειται στο πρόβλημα μοτίβων για την δημιουργία γενικών κλάσεων.