Βελτιστοποίηση Συστημάτων &
Υδροπληροφορική

Γενετικοί Αλγόριθμοι

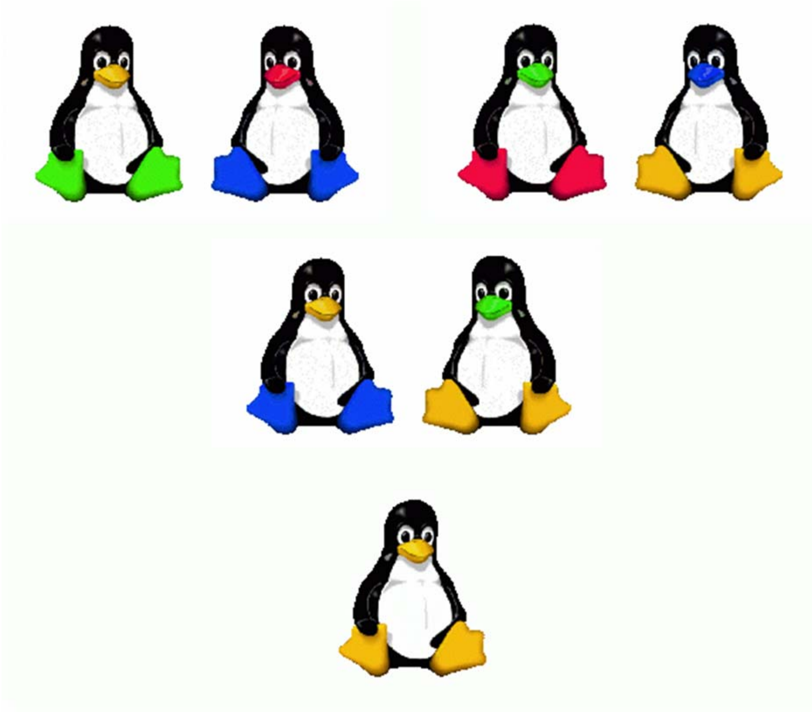
Χρήστος Μακρόπουλος & Ανδρέας Ευστρατιάδης
Τομέας Υδατικών Πόρων και Περιβάλλοντος
Εθνικό Μετσόβιο Πολυτεχνείο, Μάρτιος 2011

“The Gene is by far the most sophisticated program around.”

\- Bill Gates, *Business Week*, June 27, 1994

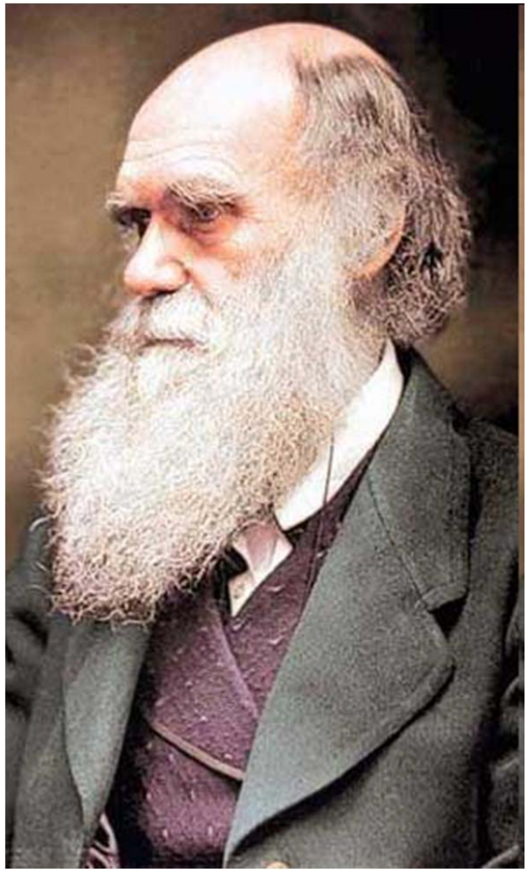
Γενετικοί αλγόριθμοι: Τι είναι;

- Αλγόριθμοι επίλυσης προβλημάτων που βασίζονται (είναι εμπνευσμένοι) από τις αρχές της *Βιολογικής Εξέλιξης (Δαρβίνος)*.



Γενετικοί Αλγόριθμοι

- Θεωρία της εξέλιξης (C. Darwin, 1858)
- **Εξέλιξη** = Διαδικασία που οδηγεί στην αύξηση της ικανότητας ενός πληθυσμού να επιβιώνει σε ένα δεδομένο περιβάλλον (Εξελικτική προσαρμογή)
- Με την αναπαραγωγή, η ικανότητα αυτή περνά στις επόμενες γενιές (από τα άτομα που την είχαν και άρα επέζησαν για να αναπαραχθούν): **Φυσική επιλογή**
- Εμείς αντί για **άτομα** έχουμε **λύσεις**



Η αρχή..

- Holland, John H (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor (η βασική ιδέα)
- Michalewicz, Zbigniew (1999), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag (το καλύτερο textbook μέχρι σήμερα)

Ο βασικός εξελικτικός κύκλος

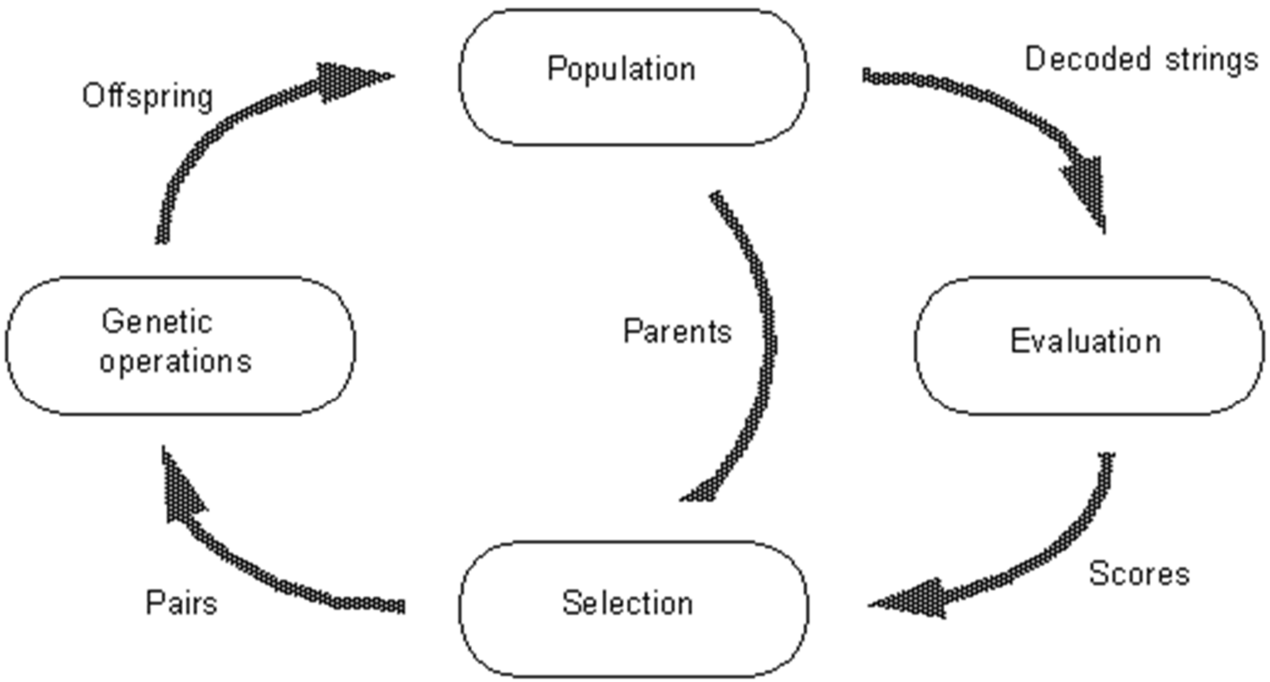


Figure 5.2: The "reproduction" cycle.

Γενετικοί αλγόριθμοι: Βασικά Χαρακτηριστικά

- Κάνουν αναζήτηση σε πολλά σημεία ταυτόχρονα και όχι μόνο σε ένα
- Χρησιμοποιούν **μόνο*** **την αντικειμενική συνάρτηση** και καμία επιπρόσθετη πληροφορία
- Χρησιμοποιούν πιθανοτικούς κανόνες αναζήτησης νέων λύσεων και όχι ντετερμινιστικούς

Πλεονεκτήματα

- Απλή βασική ιδέα
- Γενικής φύσης – μπορεί να εφαρμοστεί σε οποιοδήποτε πρόβλημα
- Υποστηρίζει πολύ-κριτηριακή βελτιστοποίηση
- Αντέχει θόρυβο/αβεβαιότητα
- Βγάζει πάντα μια απάντηση, η οποία και βελτιώνεται με το χρόνο
- Μπορούν να επιλύουν δύσκολα προβλήματα γρήγορα και αξιόπιστα.
- Μπορούν εύκολα να συνδεθούν με υπάρχοντα μοντέλα και συστήματα
- Είναι εύκολα επεκτάσιμοι και εξελίξιμοι.
- Μπορούν να συνδυαστούν (σε υβριδικές μορφές) με άλλες μεθόδους.
- Εφαρμόζονται σε πολύ περισσότερα πεδία από κάθε άλλη μέθοδο.
- Έχουν από τη φύση τους το στοιχείο του παραλληλισμού και άρα προσφέρεται για παράλληλη υλοποίηση.

Γενετικοί Αλγόριθμοι: Ορολογία…

- Δανεισμένη από το χώρο της Γενετικής.
- Αναφέρονται σε ***άτομα*** μέσα σε ένα πληθυσμό. Πολύ συχνά αυτά τα άτομα καλούνται επίσης ***χρωμοσώματα***.
- Τα **χρωμοσώματα** αποτελούνται από διάφορα στοιχεία που ονομάζονται ***γονίδια.***
- Κάθε γονίδιο επηρεάζει την κληρονομικότητα ενός ή περισσότερων χαρακτηριστικών (*είναι δηλαδή συνδεδεμένη με μια παράμετρο της λύσης*)

Γενετικοί αλγόριθμοι: Πώς λειτουργούν;

- **Διατηρούν** έναν πληθυσμό κωδικοποιημένων πιθανών λύσεων
- Υπολογίζουν την αντικειμενική συνάρτηση για κάθε άτομο (λύση) του δημιουργούμενου πληθυσμού (fitness evaluation - επίδοση)
- **Εξελίσσουν** τον πληθυσμό εφαρμόζοντας γενετικές διαδικασίες που επηρεάζονται από την επίδοση:
  - Διαδικασίες **επιλογής**,
  - Διαδικασίες **αναπαραγωγής**,
  - Διαδικασίες **μετάλλαξης**.
- Δημιουργούν νέο πληθυσμό που αντικαθιστά τον προηγούμενο με βάση την επίδοση.
- Επαναλαμβάνουν τη διαδικασία μέχρι να «βρουν λύση».

Γενετικοί αλγόριθμοι: Πώς λειτουργούν;

- Ένας Γ.Α. αποτελείται από:
  - **Γενετική αναπαράσταση**
  - **Τρόπο δημιουργίας ενός αρχικού πληθυσμού**
  - **Αντικειμενική συνάρτηση αξιολόγησης**
  - **Γενετικούς τελεστές**
  - **Τιμές για τις διάφορες παραμέτρους (του).**

Γενετικοί αλγόριθμοι: Πώς λειτουργούν;

- Γενετικοί Τελεστές
  - **Επιλογή**: επιλέγει με κάποιο τρόπο τα «καταλληλότερα» μέλη του πληθυσμού και τα χρησιμοποιεί για:
  - **Διασταύρωση**: συνδυάζει τα στοιχεία δύο χρωμοσωμάτων γονέων για να δημιουργήσει δύο νέους απογόνους ανταλλάσσοντας αντίστοιχα κομμάτια από τους γονείς.
  - **Μετάλλαξη**: αλλάζει αυθαίρετα ένα ή περισσότερα γονίδια ενός συγκεκριμένου χρωμοσώματος.

Γενετική Αναπαράσταση: (δυαδική) κωδικοποίηση

(11, 6, 9) → **1011 0110 1001**

Στο δυαδικό κάθε αριθμός είναι μια αύξουσα δύναμη του 2 ξεκινώντας από το 2^0

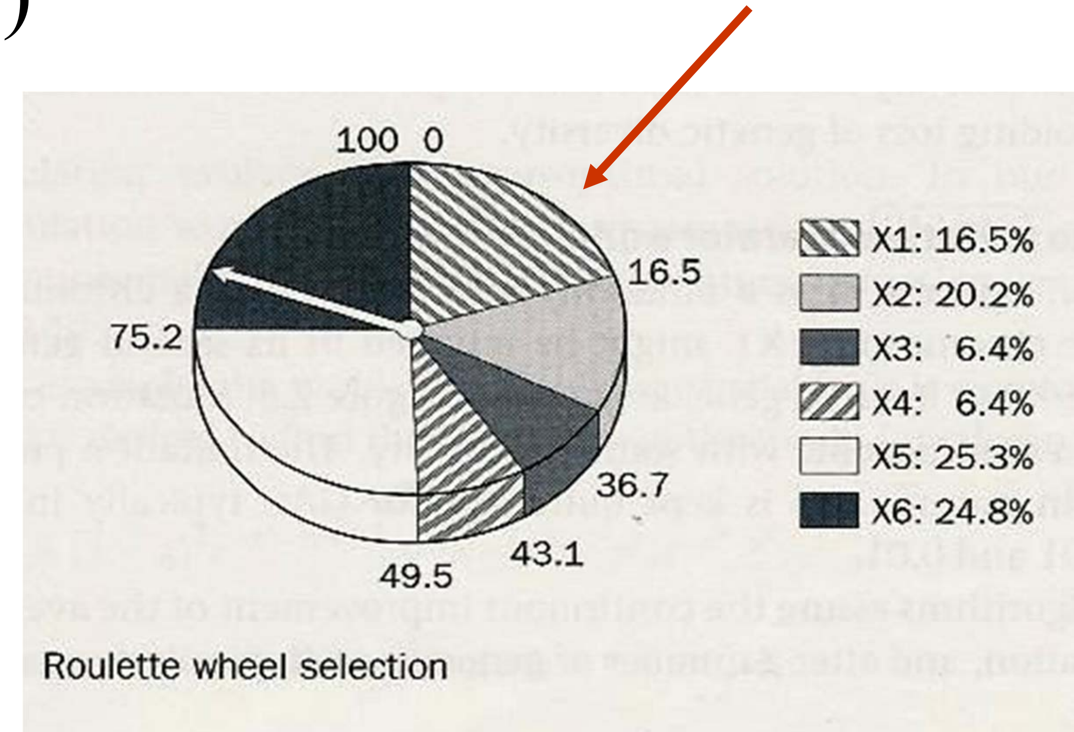
π.χ. 100101 = $[(\mathbf{1}) \times 2^5] + [(\mathbf{0}) \times 2^4] + [(\mathbf{0}) \times 2^3] + [(\mathbf{1}) \times 2^2] + [(\mathbf{0}) \times 2^1] + [(\mathbf{1}) \times 2^0] = 37$

1010=?

Επιλογή: Μηχανισμός ρουλέτας

$$\Pr(x_i) = \frac{f(x_i)}{\sum_j f(x_j)}$$

Όπου f(x): η τιμή της αντικειμενικής στο x



Roulette wheel selection

Επίσης: Επιλογή Τουρνουά (Tournament Selection) μετά..

Γενετικοί Τελεστές

Διασταύρωση

1 0 0 1 | 1 1 1 0	→	1 0 0 1 | 0 0 1 0
1 0 1 1 | 0 0 1 0		1 0 1 1 | 1 1 1 0

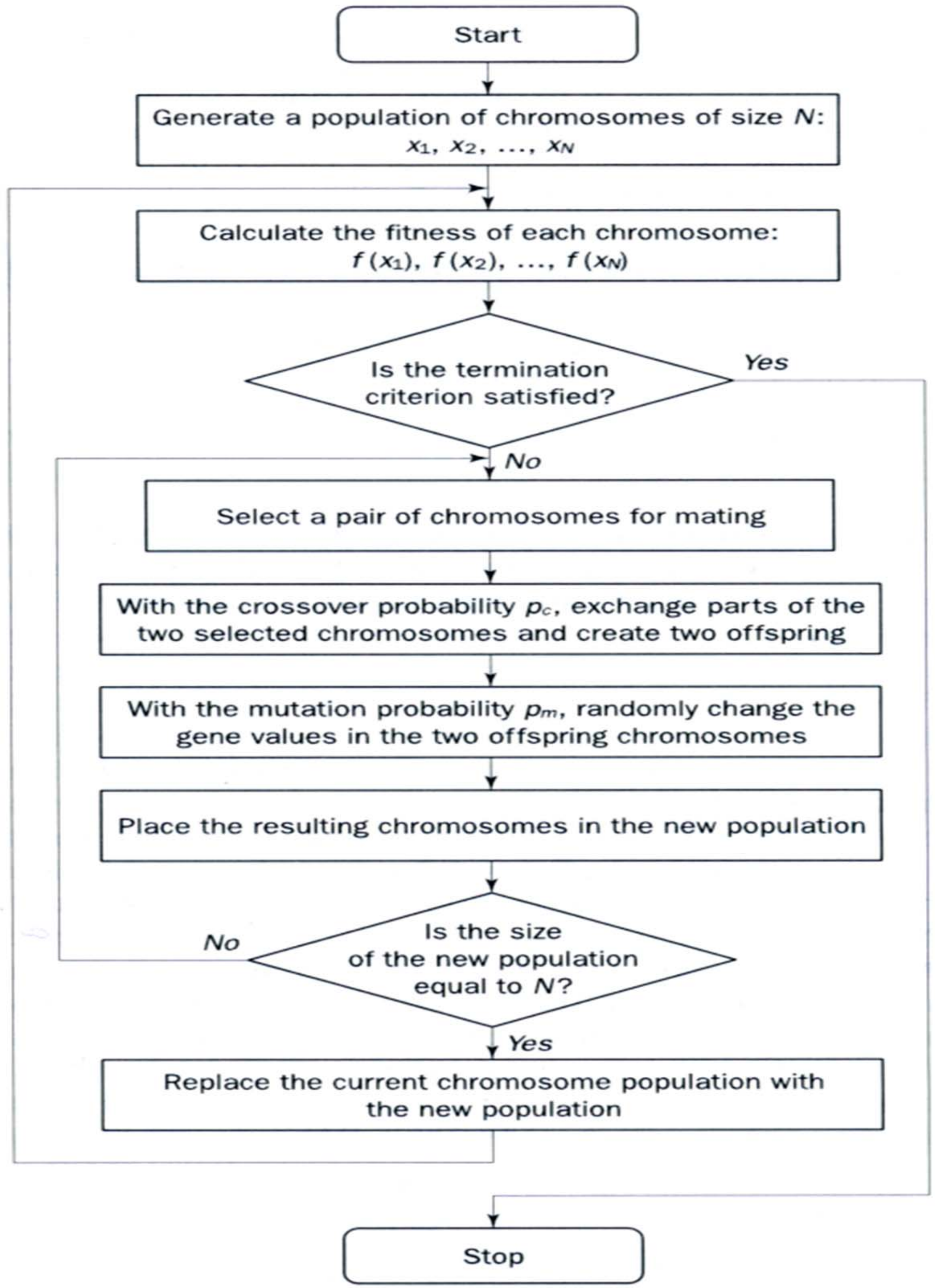
Σημείο διασταύρωσης

Μετάλλαξη

1 0 0 1 1 1 1 0	→	1 0 0 1 1 0 1 0

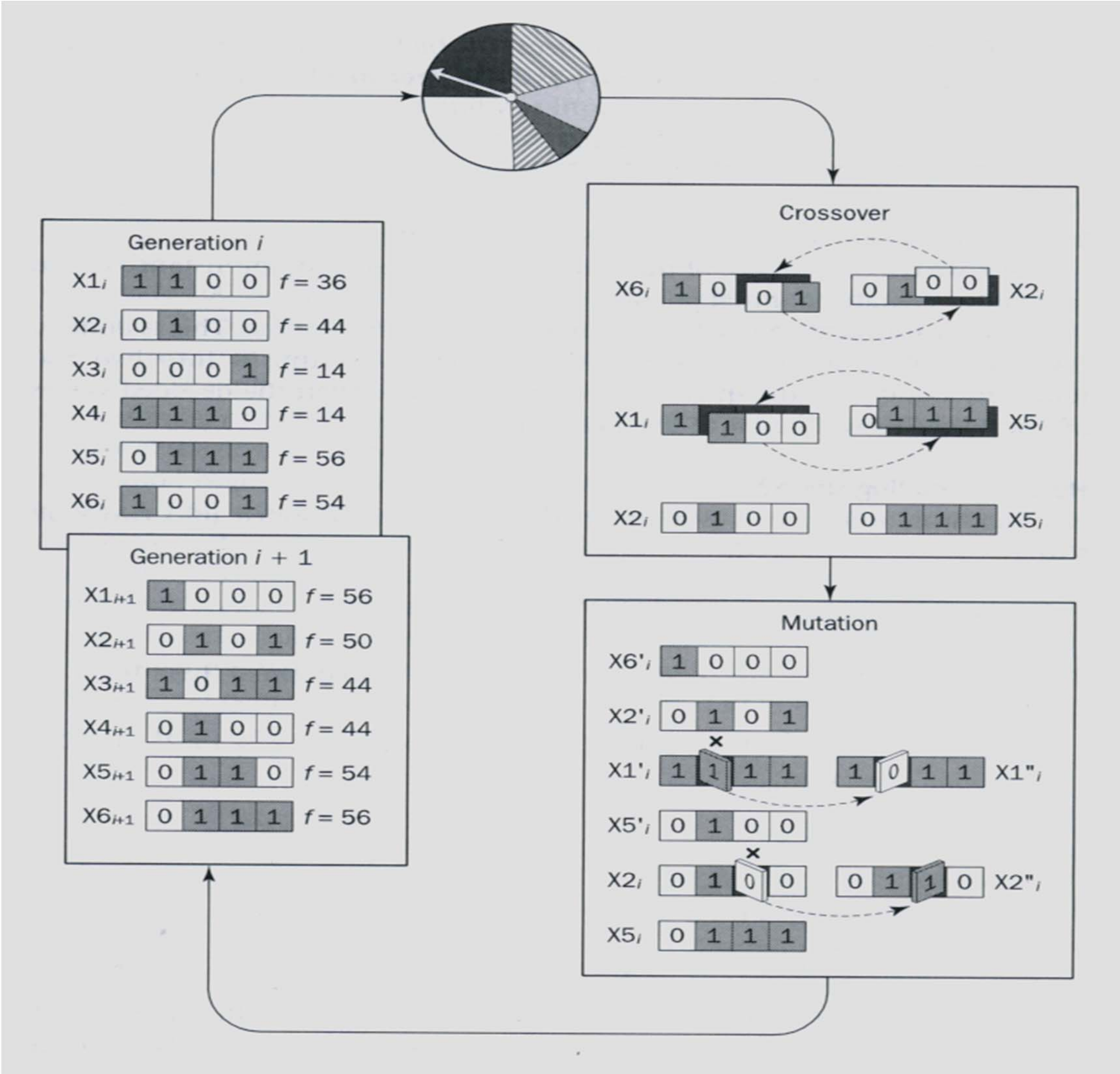
Ψηφίο μετάλλαξης

Διάγραμμα ροής



Γενετικός Κύκλος: η κάθε γενιά είναι κατά μέσο όρο **καλύτερη** από τη προηγούμενη

(και τουλάχιστον το ίδιο καλή)



Το θεώρημα των σχημάτων (Holland, 1975)

Σχήμα (Schema) = Ακολουθία που περιλαμβάνει 0, 1 και ‘*’ (“οτιδήποτε”)

Π.χ. το σχήμα

1	*	*	0

παριστάνει το σύνολο των δυαδικών ακολουθιών (χρωμοσωμάτων)

1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	0

Μια δυαδική ακολουθία μήκους L είναι εκπρόσωπος καθενός από τα 2^L διαφορετικά σχήματα με τα οποία ταιριάζει.

Το θεώρημα αποδεικνύει ότι σχήματα μικρού μήκους με απόδοση άνω του μέσου όρου **αυξάνουν εκθετικά μετά την εφαρμογή γενετικών τελεστών, σε επόμενες γενιές**.

Εξίσωση ανάπτυξης σχημάτων

Κάτω φράγμα της αναμενόμενης συχνότητας του σχήματος *s* (λαμβάνοντας υπόψη μόνο την αρνητική επίδραση των τελεστών διασταύρωσης και μετάλλαξης):

$$E[\alpha(s,t+1)] \geq \frac{f(s,t)}{F(t)}\alpha(s,t)\left(1-p_c\frac{l(s)}{L-1}\right)(1-p_m)^{\tau(s)}$$

Προσοχή:

- Το θεώρημα αναφέρεται ουσιαστικά σε πληθυσμό με άπειρα μέλη – στην πράξη η αναμενόμενη συχνότητα μπορεί να διαφοροποιείται.
- Η ακριβής μορφή της διατύπωσης εξαρτάται από τους συγκεκριμένους τελεστές.
- Μιλάει για «συχνότητα εμφάνισης» και δε μας λέει τίποτα για το ποια σχήματα (δηλ. πόσο καλά σχήματα) περνάνε στην επόμενη γενιά.

Δείτε επίσης: http://www.cse.unr.edu/~sushil/class/gas/notes/GASchemaTheorem2.pdf

Παράδειγμα:

Εύρεση μεγίστου της

$$F(x)=x^2$$

όπου x είναι ακέραιος στο διάστημα [1, 31].

Παράδειγμα: Η κωδικοποίηση

- Θέλουμε να αναπαραστήσουμε αριθμούς μέχρι το 31, οπότε θα χρησιμοποιήσουμε χρωμοσώματα των 5 γονιδίων στο δυαδικό: $2^5=32>31$.

Παράδειγμα: Η αρχικοποίηση

- Δημιουργία αρχικού πληθυσμού (έστω μεγέθους 4) με τυχαίο τρόπο:

$\mathbf{A_1 = 0\ 1\ 1\ 0\ 1 = 13_{10}}$

$\mathbf{A_2 = 1\ 1\ 0\ 0\ 0 = 24_{10}}$

$\mathbf{A_3 = 0\ 1\ 0\ 0\ 0 = 8_{10}}$

$\mathbf{A_4 = 1\ 0\ 0\ 1\ 1 = 19_{10}}$

Παράδειγμα: Η αξιολόγηση

$F(A_1) = 13^2 = 169$

$F(A_2) = 24^2 = 576$

$F(A_3) = 8^2 = 64$

$F(A_4) = 19^2 = 361$

Συνολική Απόδοση: 1170

Μέση απόδοση: 293

Παράδειγμα: Η επιλογή

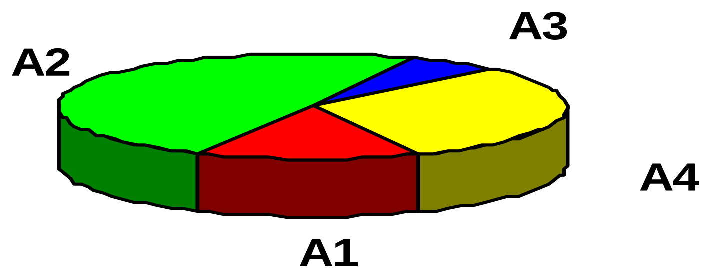
Ρουλέτα με βάρη: κάθε μέλος του πληθυσμού έχει πιθανότητα επιλογής ίση με τη σχετική του απόδοση στον τρέχοντα πληθυσμό.

$P(A_1) = 0.14$

$P(A_2) = 0.49$

$P(A_3) = 0.06$

$P(A_4) = 0.31$



Παράδειγμα: Η αναπαραγωγή

- Ο προσωρινός πληθυσμός μετά την εφαρμογή της ρουλέτας:

A'_1 = 0 1 1 0 1

A'_2 = 1 1 0 0 0

A'_3 = 1 1 0 0 0

A'_4 = 1 0 0 1 1

Με αρχικό:
A1 = 0 1 1 0 1
A2 = 1 1 0 0 0
A3 = 0 1 0 0 0
A4 = 1 0 0 1 1

Παράδειγμα: Η διασταύρωση

Επιλογή με τυχαίο τρόπο των ατόμων που θα διασταυρώσουν το γενετικό υλικό τους:

Έστω ότι διασταυρώνονται το A'_1 με το A'_2 με σημείο διασταύρωσης το 4 και το A'_3 με το A'_4 με σημείο διασταύρωσης το 2:

A'_1 = 0 1 1 0 | 1 → **A''_1 = 0 1 1 0 | 0**

A'_2 = 1 1 0 0 | 0 → **A''_2 = 1 1 0 0 | 1**

A'_3 = 1 1 | 0 0 0 → **A''_3 = 1 1 | 0 1 1**

A'_4 = 1 0 | 0 1 1 → **A''_4 = 1 0 | 0 0 0**

Παράδειγμα: Η μετάλλαξη

Με τυχαίο τρόπο επιλέγονται γονίδια των οποίων η τιμή αντιστρέφεται:

$$A''_1 = 0\ 1\ 1\ 0\ 0 \qquad A'''_1 = 0\ 1\ 1\ 0\ 0$$

$$A''_2 = 1\ 1\ 0\ 0\ 1 \qquad A'''_2 = 1\ 1\ 0\ 0\ 1$$

$$A''_3 = 1\ 1\ 0\ 1\ 1 \Rightarrow A'''_3 = 1\ 1\ 0\ 1\ 1$$

$$A''_4 = 1\ 0\ 0\ 0\ 0 \qquad A'''_4 = 1\ 0\ 0\ 1\ 0$$

Παράδειγμα: Ο νέος πληθυσμός

$\mathbf{A_1 = 0\ 1\ 1\ 0\ 0} = 12_{10}$ =>F(12)=144

$\mathbf{A_2 = 1\ 1\ 0\ 0\ 1} = 25_{10}$ =>F(25)=625

$\mathbf{A_3 = 1\ 1\ 0\ 1\ 1} = 27_{10}$ =>F(27)=729

$\mathbf{A_4 = 1\ 0\ 0\ 1\ 0} = 18_{10}$ =>F(18)=324

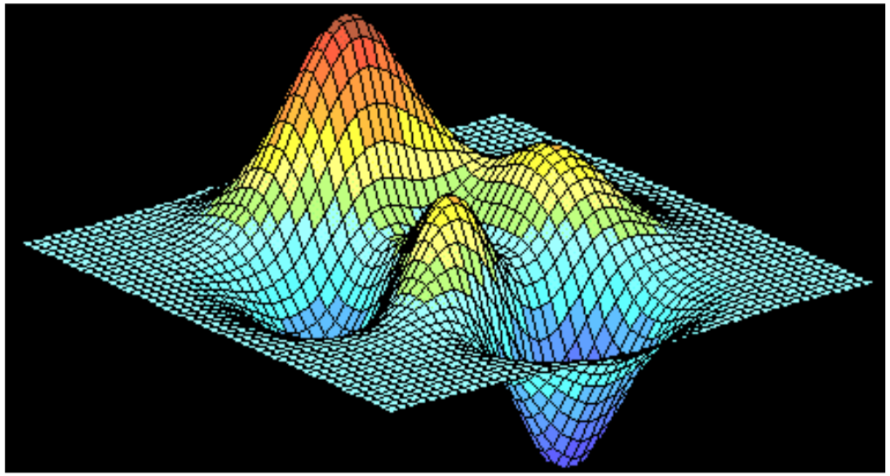
Συνολική Απόδοση: 1822 (από 1170)

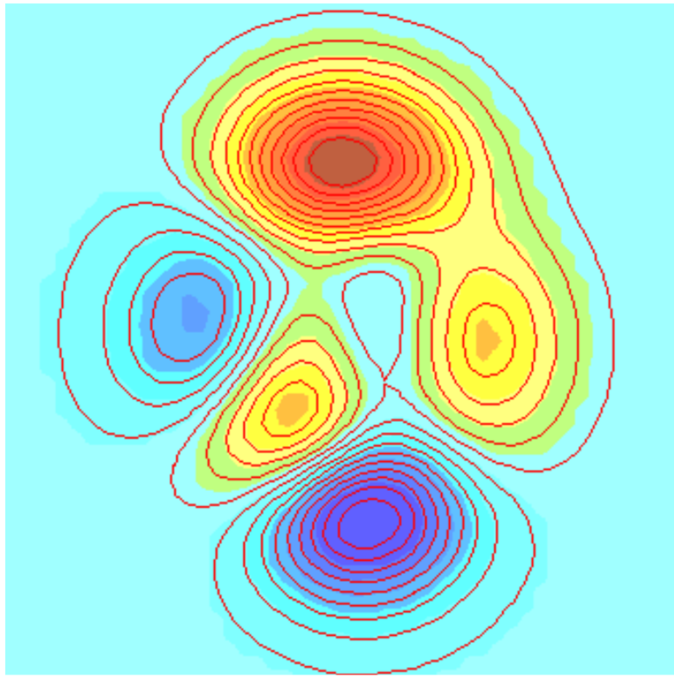
Μέση απόδοση:455.5 (από 293)

Προσέξτε ότι ο αλγόριθμος δεν γνωρίζει τη μορφή του χώρου των λύσεων…

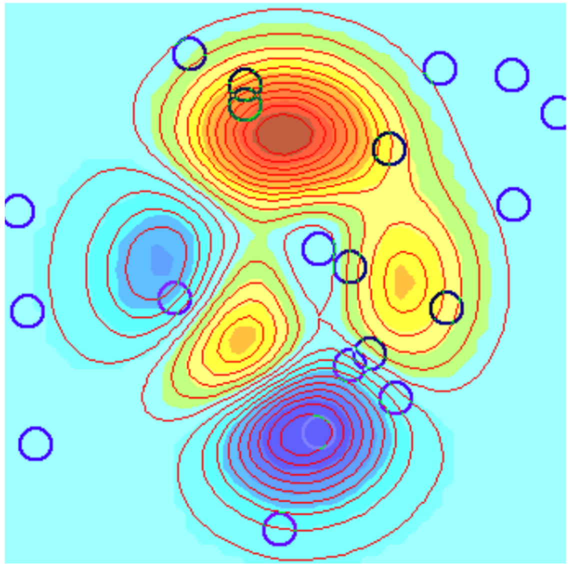
Παράδειγμα: μέγιστο της συνάρτησης peak

z = f(x, y) = 3*(1-x)^2*exp(-(x^2) - (y+1)^2) - 10*(x/5 - x^3 - y^5)*exp(-x^2-y^2) -1/3*exp(-(x+1)^2 - y^2).

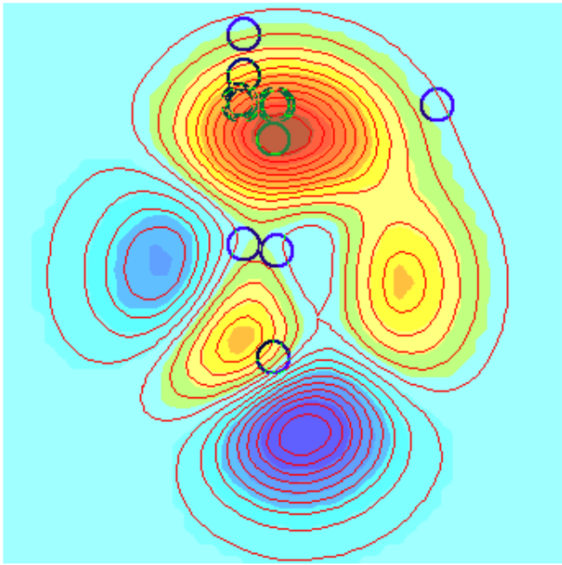




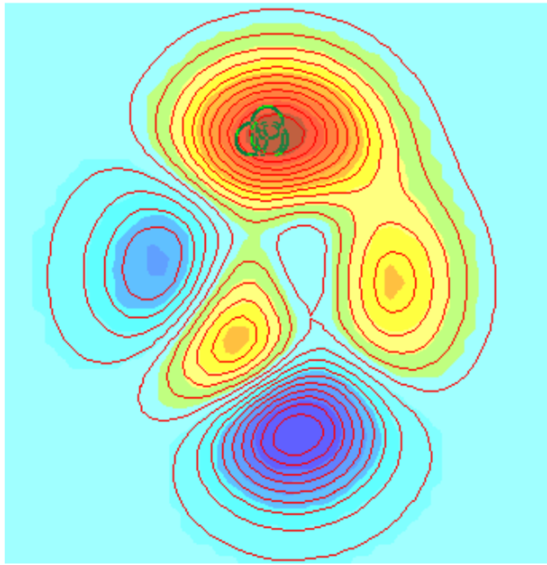
Παράδειγμα επίλυσης



Αρχικός πληθυσμός

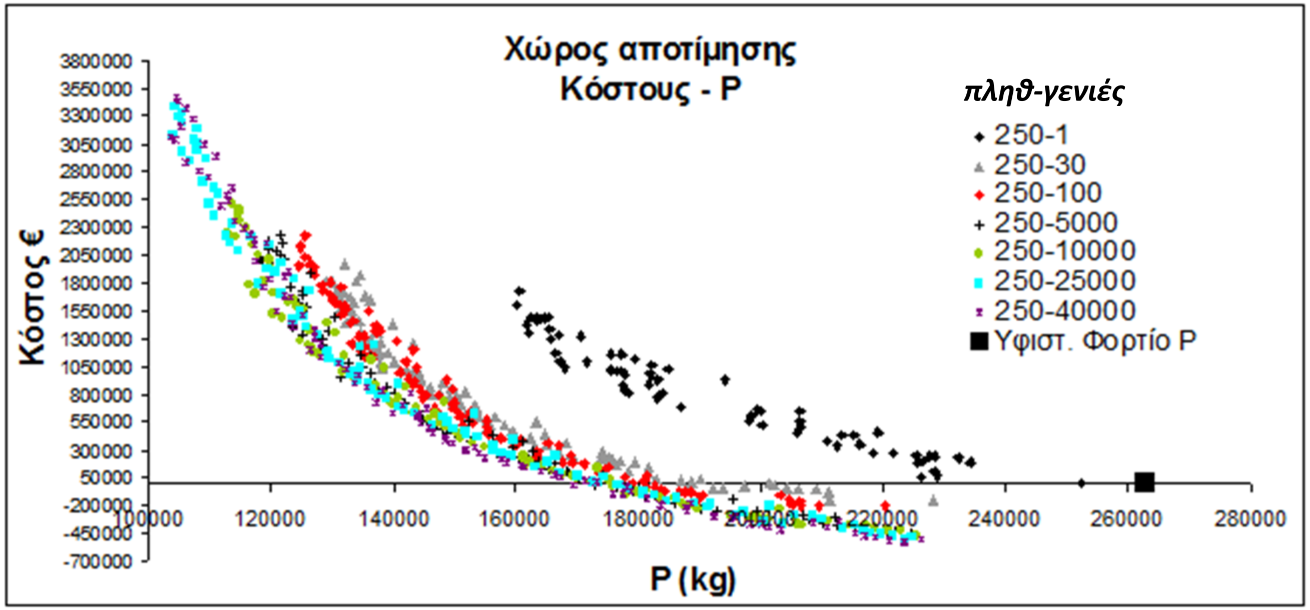


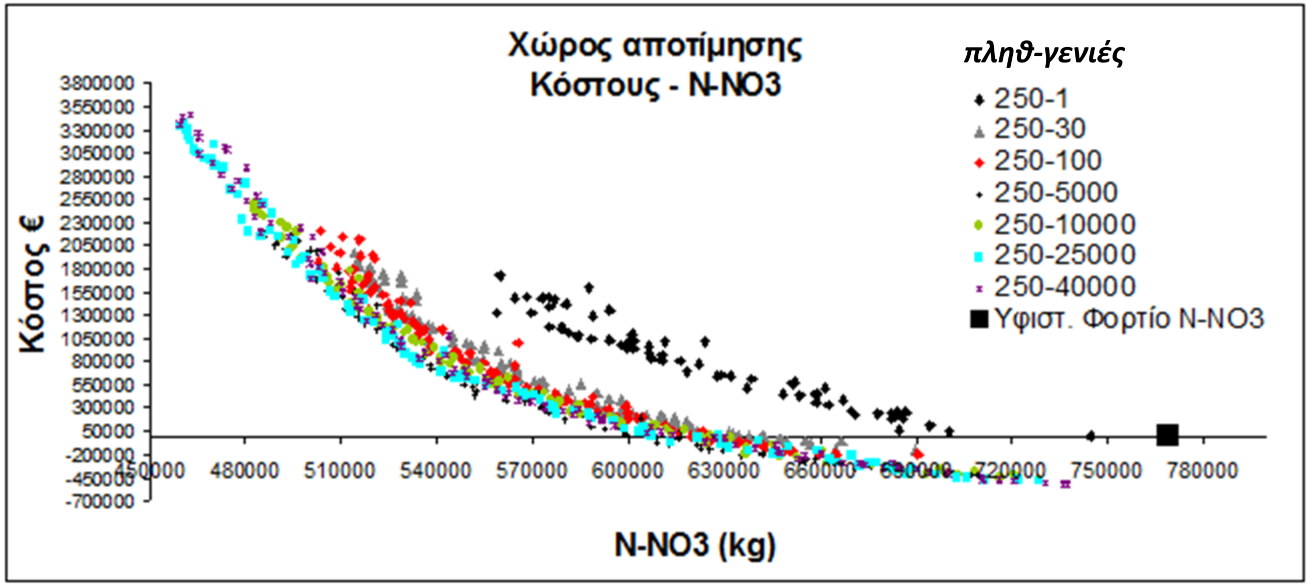
5η γενιά

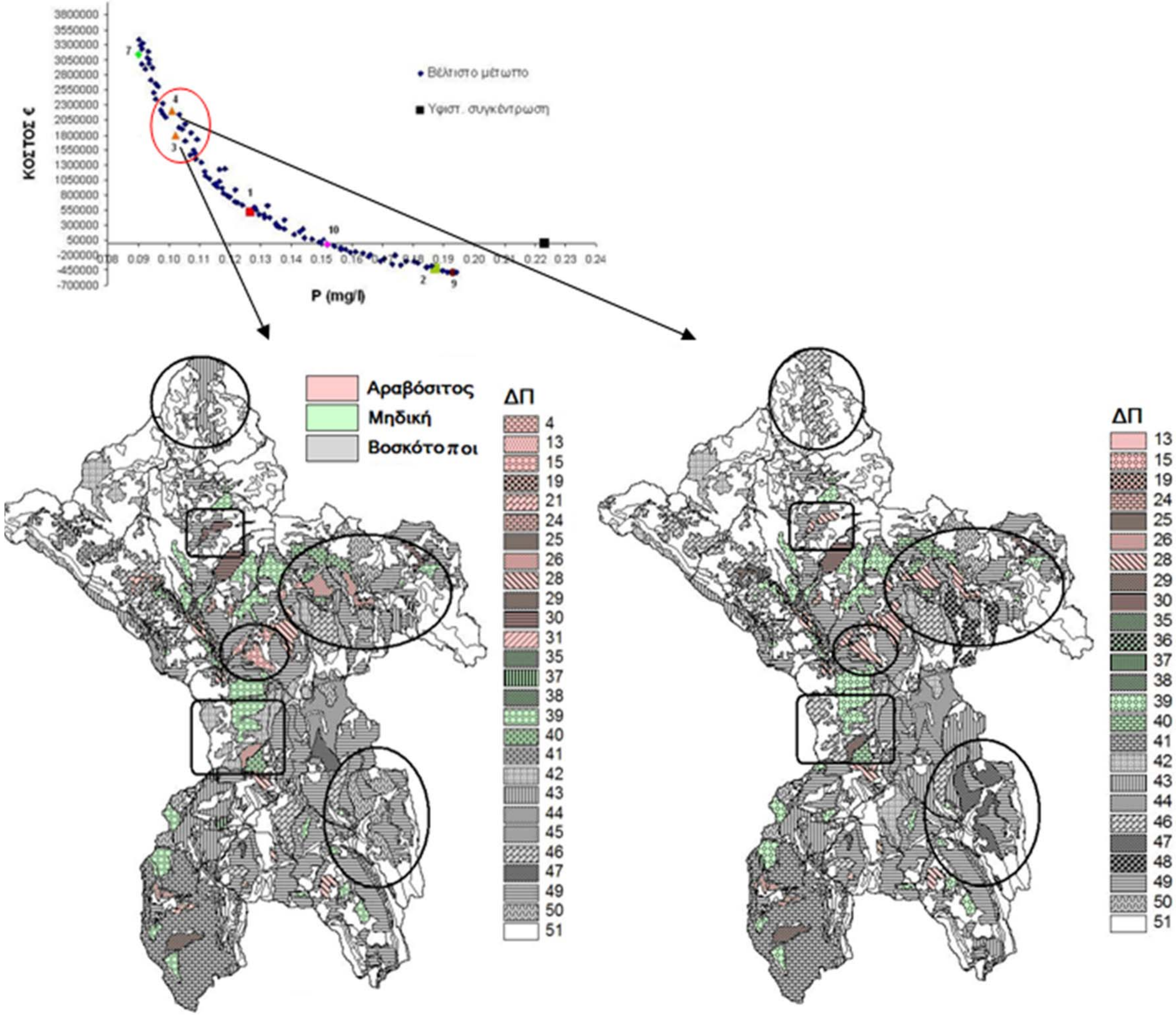


10η γενιά

Παράδειγμα εξέλιξης λύσεων ΓΑ για πρόβλημα διαχείρισης μη-σημειακής ρύπανσης



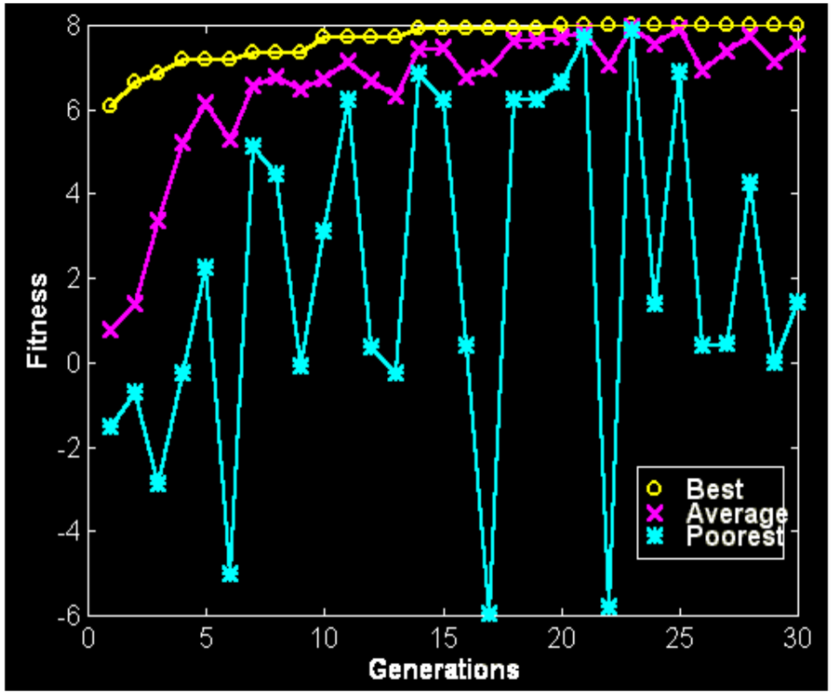




3800000
3550000
3300000
3050000
2800000
2550000
2300000
2050000
1800000
1550000
1300000
1050000
800000
550000
300000
50000
-200000
-450000
-700000
ΚΟΣΤΟΣ €
0.08
0.09
0.10
0.11
0.12
0.13
0.14
0.15
0.16
0.17
0.18
0.19
0.20
0.21
0.22
0.23
0.24
P (mg/l)
Βέλτιστο μέτωπο
Υφιστ. συγκέντρωση
7
4
3
1
10
2
9
Αραβόσιτος
Μηδική
Βοσκότοποι
ΔΠ
4
13
15
19
21
24
25
26
28
29
30
31
35
37
38
39
40
41
42
43
44
45
46
47
49
50
51
ΔΠ
13
15
19
24
25
26
28
29
30
35
36
37
38
39
40
41
42
43
44
46
47
48
49
50
51

Επίδοση

- Εγκλωβισμός σε τοπικά ακρότατα;
  - Διασταύρωση: χρήση της υπάρχουσας πληροφορίας
  - Μετάλλαξη: αναζήτηση νέας πληροφορίας
- P(mutation)→0 όσο εξελίσσεται η βελτιστοποίηση



Επαναλήψεις

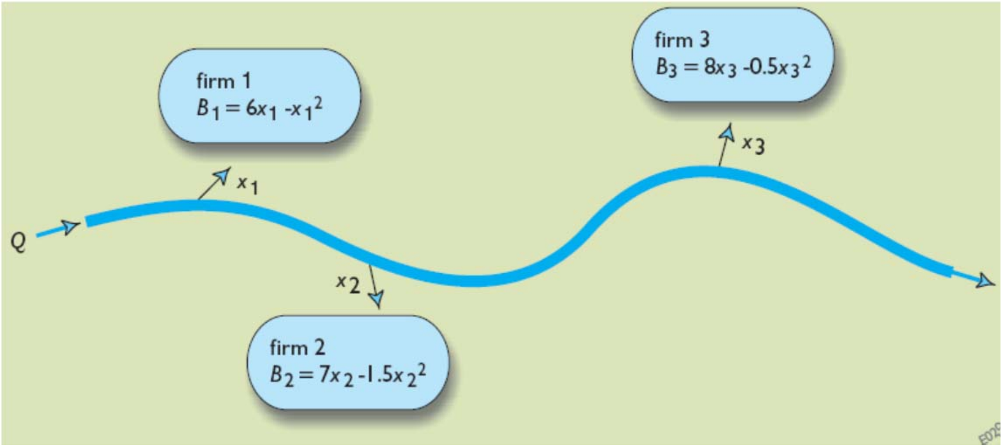
- Η διαδικασία αυτή μεταφέρει στην επόμενη γενιά τις καλύτερες από την προηγούμενη γενιά: Όσο πιο υψηλό fitness (ουσιαστικά μεγαλύτερη τιμή στην αντικειμενική συνάρτηση) τόσο μεγαλύτερη η **πιθανότητα** να συμπεριληφθεί στον πληθυσμό της επόμενης γενιάς.
- Οι γενετικοί αλγόριθμοι περιλαμβάνουν πολλές επαναλήψεις των διαδικασιών που αναφέρθηκαν.
- Κάθε επανάληψη (ή γενιά) είναι καλύτερη από την προηγούμενη (σαν μέσος όρος). **Η απόλυτα καλύτερη λύση κάθε γενιάς συνήθως μεταφέρεται αυτούσια στην επόμενη γενιά** (**ελιτισμός**).
- Ο ΓΑ τελειώνει όταν οι επαναλήψεις δεν επιτυγχάνουν βελτίωση των λύσεων.
- **<u>Προσοχή</u>**: δεν υπάρχει καμία σιγουριά ότι η τελική λύση που βρήκε ο ΓΑ είναι η βέλτιστη.

Παράδειγμα: Υδατικοί Πόροι

- Κατανομή νερού σε 3 χρήστες
- Μέγιστη ποσότητα νερού σε κάθε χρήστη να μην υπερβαίνει το 5
- Το άθροισμα όλων των χρηστών να μην υπερβαίνει το 6

$$0 \le x_i \le 5 \quad \text{for } i = 1, 2, \text{ and } 3.$$

$$x_1 + x_2 + x_3 \le 6$$



$$\text{Maximize } B(X) = (6x_1 - x_1^2) + (7x_2 - 1.5x_2^2) + (8x_3 - 0.5x_3^2)$$

Διαδικασία επίλυσης

- Δημιουργούμε (τυχαία) ένα πληθυσμό εφικτών λύσεων
- Βασικοί παράμετροι ενός ΓΑ είναι ο αριθμός των λύσεων που εξετάζουμε και παράμετροι που καθορίζουν την επιλογή, διασταύρωση και μετάλλαξη
- Οι παράμετροι αυτοί καθορίζονται με δοκιμή και πλάνη (trial and error).
- Μια πιθανή λύση μπορεί να είναι η **[3,1,2]** που καθορίζει κατανομή $x1=3$, $x2=1$, and $x3=2$.
- Μια άλλη λύση μπορεί να είναι η **[1,0,1]**
- Οι δύο αυτές λύσεις μπορούν να συνδυαστούν και να δώσουν 2 «απογόνους».

Διαδικασία επίλυσης

- Τα γονίδια των απογόνων καθορίζονται από την διασταύρωση και τη μετάλλαξη
- Ας υποθέσουμε ότι το σημείο διασταύρωσης καθορίζεται τυχαία και είναι μετά το πρώτο στοιχείο

3 | 1 2

1 | 0 1

?

Διαδικασία επίλυσης: διασταύρωση

- Ένας άλλος τρόπος (ομοιόμορφης) διασταύρωσης είναι να προσδιορίσουμε **για κάθε ζεύγος γονιδίων** αν θα διασταυρωθούν ή όχι.
- Μπορούμε για παράδειγμα να θέσουμε τη πιθανότητα διασταύρωσης σε 0.30.
- Άρα **κάθε ζεύγος γονιδίων** έχει 30% πιθανότητα να διασταυρωθεί.
- Για παράδειγμα θα μπορούσε να καταλήξει μόνο το μεσαίο από τα ζεύγη των λύσεων 312 και 101 να διασταυρωθεί: 3**0**2 and 1**1**1.

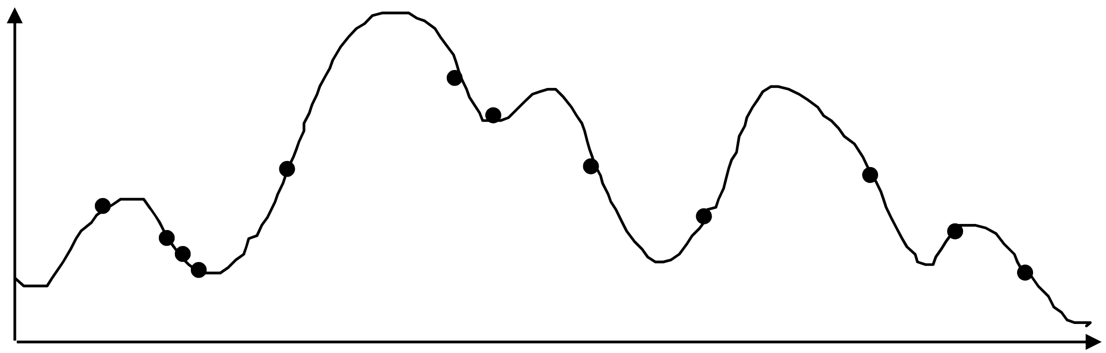
Διαδικασία επίλυσης: μετάλλαξη

- Η μετάλλαξη καθορίζεται σε κάθε γονίδιο και συνεπάγεται την αλλαγή της τιμής του γονιδίου (με κάποιο προκαθορισμένο τρόπο – εκτός αν είναι δυαδικό οπότε απλά το 0 γίνεται 1 και αντίστροφα).
- Πχ, ας αποφασίσουμε ότι μετάλλαξη σημαίνει -1 (εκτός αν είναι μη εφικτό οπότε το γονίδιο γίνεται το μέγιστο) πχ το 0 γίνεται 5, ενώ το 2 γίνεται 1).
- Ας υποθέσουμε ότι μεταλλάσουμε το μεσαίο ψηφίο του **112** και άρα έχουμε **102**.
- Η διαδικασία μας έδωσε από τα **312** και **101** → **301** και **102.**
- Πρέπει να υπολογίσουμε το όφελος για τις νέες λύσεις (301→16.5 & 102→19.0)

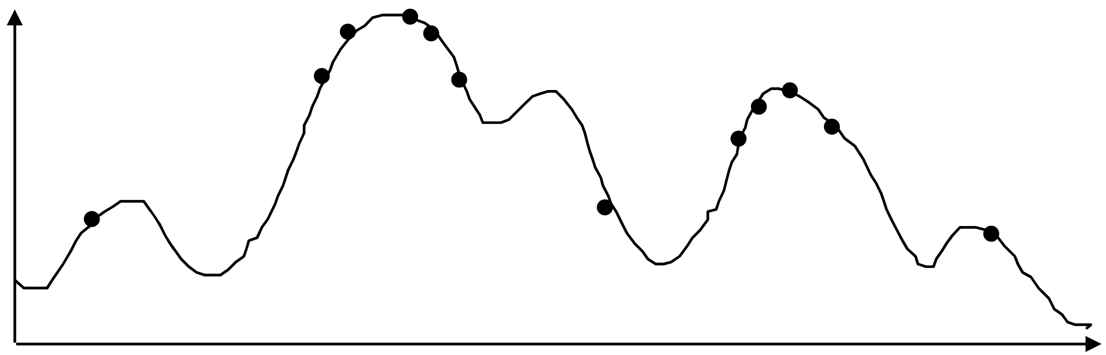
Διαδικασία επίλυσης: επιλογή νέου πληθυσμού

- Το άθροισμα των δύο οφελών είναι 35.5.
- Ορίζουμε ως πιθανότητα επιλογής της λύσης 301 το 16.5/35.5= 0.47 και της λύσης 102 το 19/35.5=0.53
- Πρακτικά επιλέγοντας ένα τυχαίο αριθμό από μια κανονική κατανομή [0 to 1], αν ο αριθμός αυτός είναι μεταξύ 0 to 0.47 τότε επιλέγουμε το 301. Αν είναι μεγαλύτερος επιλέγουμε το 102.
- Και η διαδικασία συνεχίζεται για μια ακόμα γενιά.

Εξέλιξη λύσεων - αναζήτηση



Λύσεις στη γενιά 0



Λύσεις στη γενιά N

Ανοιχτά θέματα

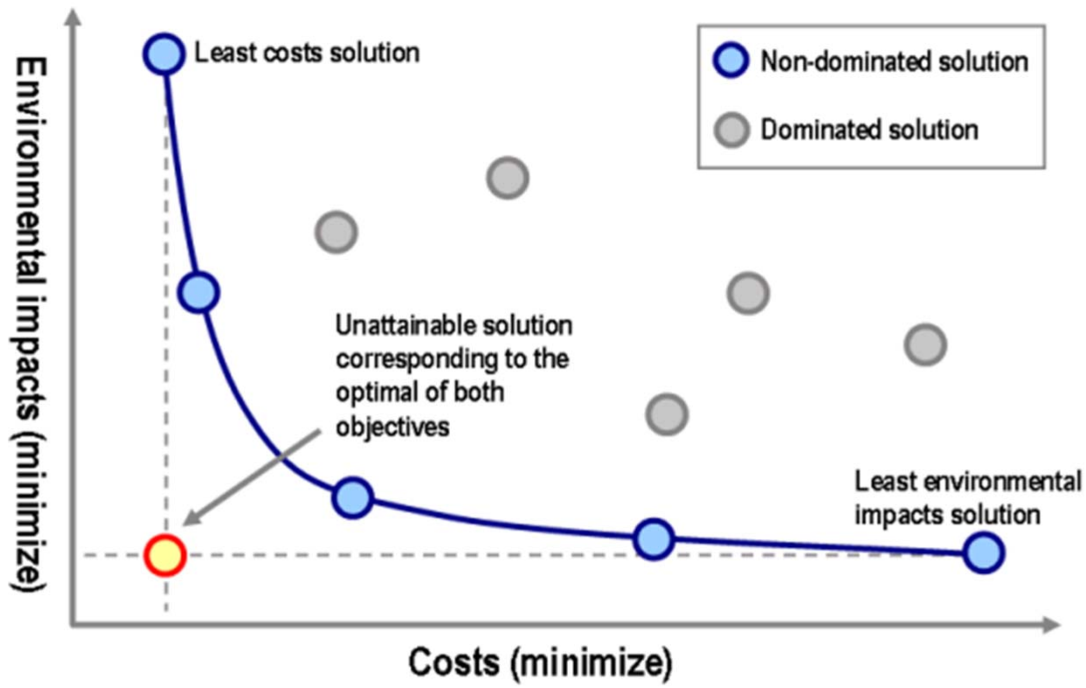
- Επιλογή βασικών παραμέτρων:
  - Μορφή του ατόμου/χρωμοσώματος/λύσης
  - Μέγεθος πληθυσμού, πιθανότητα μετάλλαξης
  - Πολιτική επιλογής
  - Επιλογή τελεστών μετάλλαξης και διασταύρωσης..
- Κριτήρια Τερματισμού
- Επίδοση, σύγκλιση, ταχύτητα
- Προφανώς ή αξιοπιστία της «βέλτιστης» λύσης εξαρτάται (όπως πάντα) από την αντικειμενική συνάρτηση …

Μέγεθος ενός πληθυσμού..

- Το μέγεθος του πληθυσμού εκφράζει πόσα χρωμοσώματα αποτελούν τον πληθυσμό (σε κάθε γενιά).
- Το μέγεθος του πληθυσμού είναι κρίσιμο για την απόδοση του Γενετικού Αλγορίθμου.
- Αν είναι μόνο μερικά χρωμοσώματα, ο ΓΑ έχει μόνο μερικές πιθανότητες να εκτελέσει διασταύρωση και ένα μικρό μέρος του χώρου καταστάσεων θα εξεταστεί.
- Οι μικροί πληθυσμοί βρίσκουν γενικά, τις καλές λύσεις γρήγορα, αλλά είναι συχνά κολλημένοι στα τοπικά βέλτιστα.
- Από την άλλη πλευρά, αν υπάρχουν πάρα πολλά χρωμοσώματα, ο ΓΑ θα μειώσει την ταχύτητά του.
- Οι μεγαλύτεροι πληθυσμοί είναι λιγότερο πιθανό να πιαστούν από τα τοπικά βέλτιστα, αλλά γενικά παίρνουν περισσότερο χρόνο για να βρουν τις καλές λύσεις.
- Ίσως να προκαλεί έκπληξη το ότι μεγάλο μέγεθος πληθυσμού συνήθως δεν βελτιώνει την απόδοση του ΓΑ (με την έννοια της ταχύτητας εύρεσης της λύσης).
- Ένας καλός αριθμός χρωμοσωμάτων σε κάθε πληθυσμό είναι περίπου 20-30, όμως, κάποιες φορές μεγέθη 50-100 θεωρούνται καλύτερα.
- Κάποιες μελέτες επίσης, δείχνουν ότι το μέγεθος του πληθυσμού εξαρτάται από την αναπαράσταση και κυρίως στο μέγεθος του string.

Πολυκριτηριακοί ΓΑ

- Ή καλύτερα πολύ-στοχικοί
- Βελτιστοποίηση κατά Pareto



Πολύ (!) πολυκριτηριακοί Γενετικοί Αλγόριθμοι

VEGA (Vector Evaluated Genetic Algorithms) David Schaffer (1984)

VOES(Vector Optimized Evolution Strategy) Frank Kursawe (1990).

MOGA (Multi-objective GA) Fonseca and Fleming (1993).

NSGA (Non-dominated Sorting GA) Srinivas and Deb (1994)

NPGA (Niched-Pareto Genetic Algorithm) Horn et al. (1994)

PPES (Predator-Prey Evolution Strategy) Laumanns et al. (1998)

DSGA (Distributed Sharing GA) Hiroyasgu et al. (1999)

DRLA (Distributed Reinforcement Learning Approach) Mariano and Morales (2000).

Nash GA, (Game Theoretic) Sefrioui and Periaux (2000)

REMOEA (Rudolph's Elitist MOEA) Rudolph (2001)

NSGA-II (Non-dominated Sorting GA) Deb et a. (2000).

Ορισμός Pareto

- Μια λύση είναι βέλτιστη κατά Pareto αν και μόνο αν:
  - Είναι τουλάχιστον τόσο καλή όσο οι άλλες λύσεις για όλους τους στόχους/κριτήρια και
  - Είναι καλύτερη από όλες τις άλλες λύσεις σε τουλάχιστον ένα στόχο/κριτήριο.

Κυριαρχία (dominance)

$$f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \quad \forall i \in \{1, \ldots, k\}$$

$$\kappa\alpha\iota$$

$$\exists i \in \{1, \ldots, k\} : f_i(\vec{x}_1) < f_i(\vec{x}_2)$$

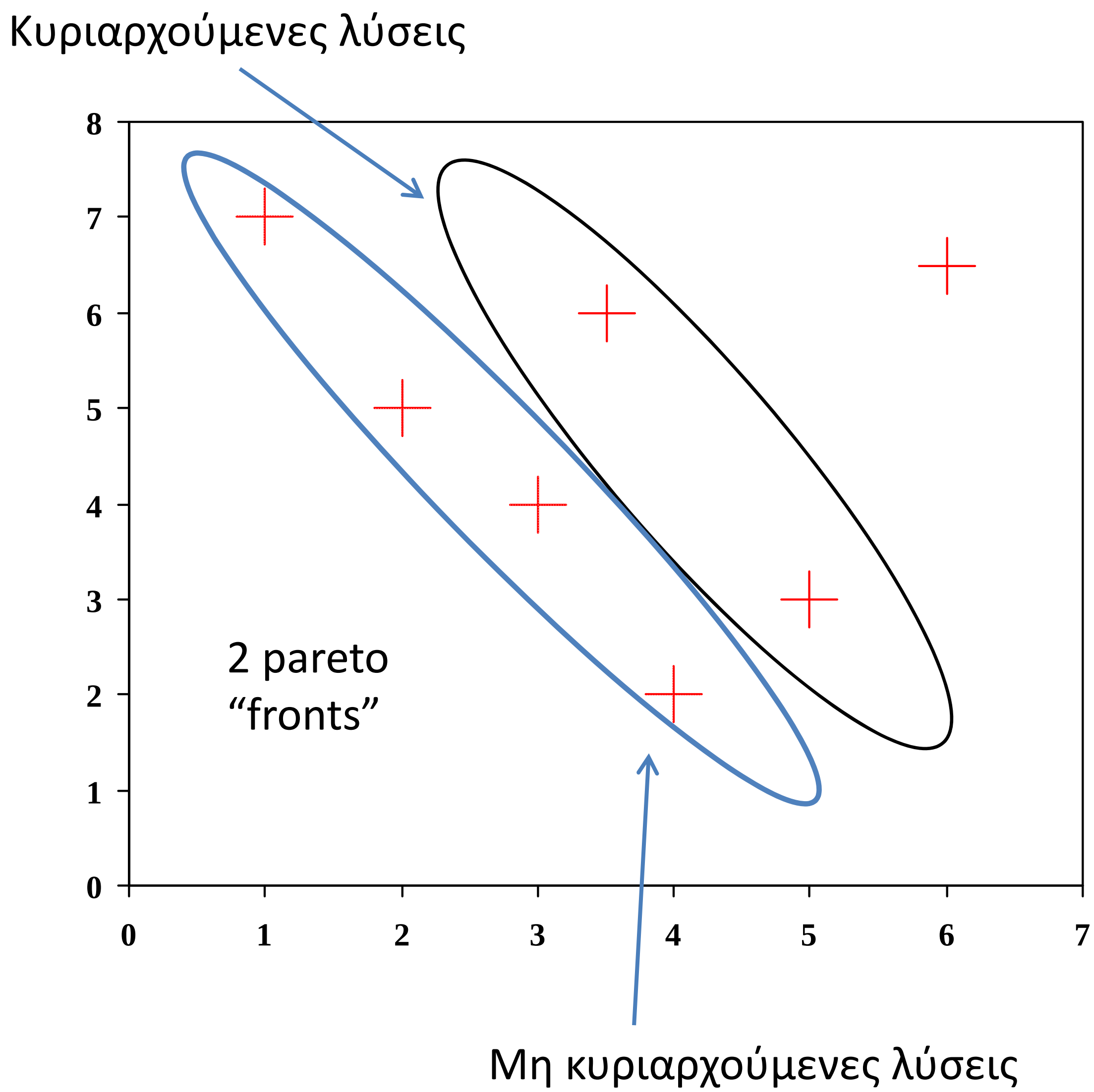
Για πρόβλημα ελαχιστοποίησης..

Μη κυριαρχούμενες λύσεις (non-dominated)

- Μπορούμε να έχουμε μόνο μία κυρίαρχη λύση
- Αλλά μπορούμε να έχουμε αρκετές μη-κυριαρχούμενες λύσεις (pareto set)

$$P := \left\{ x \in F \mid \left[\neg \exists x' \in F \right] \ni \left(\vec{f}(x') \preceq \vec{f}(x) \right) \right\}$$

Παράδειγμα



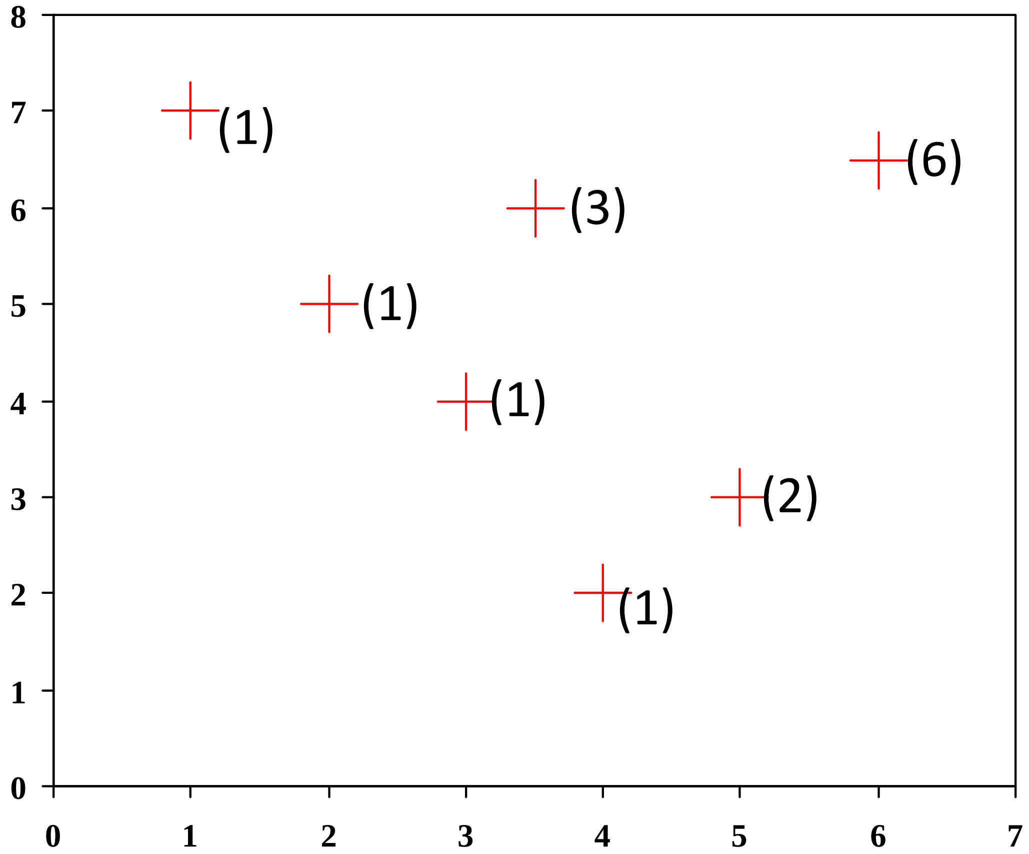
Μια και οι ΓΑ από τη φύση τους δημιουργούν πολλαπλές λύσεις (πληθυσμούς) είναι εύκολο να δημιουργήσεις pareto: είναι πολυκριτηριακοί από φυσικού τους!

(Fonseca and Fleming 1993)

Παράδειγμα (ελαχιστοποίησης)

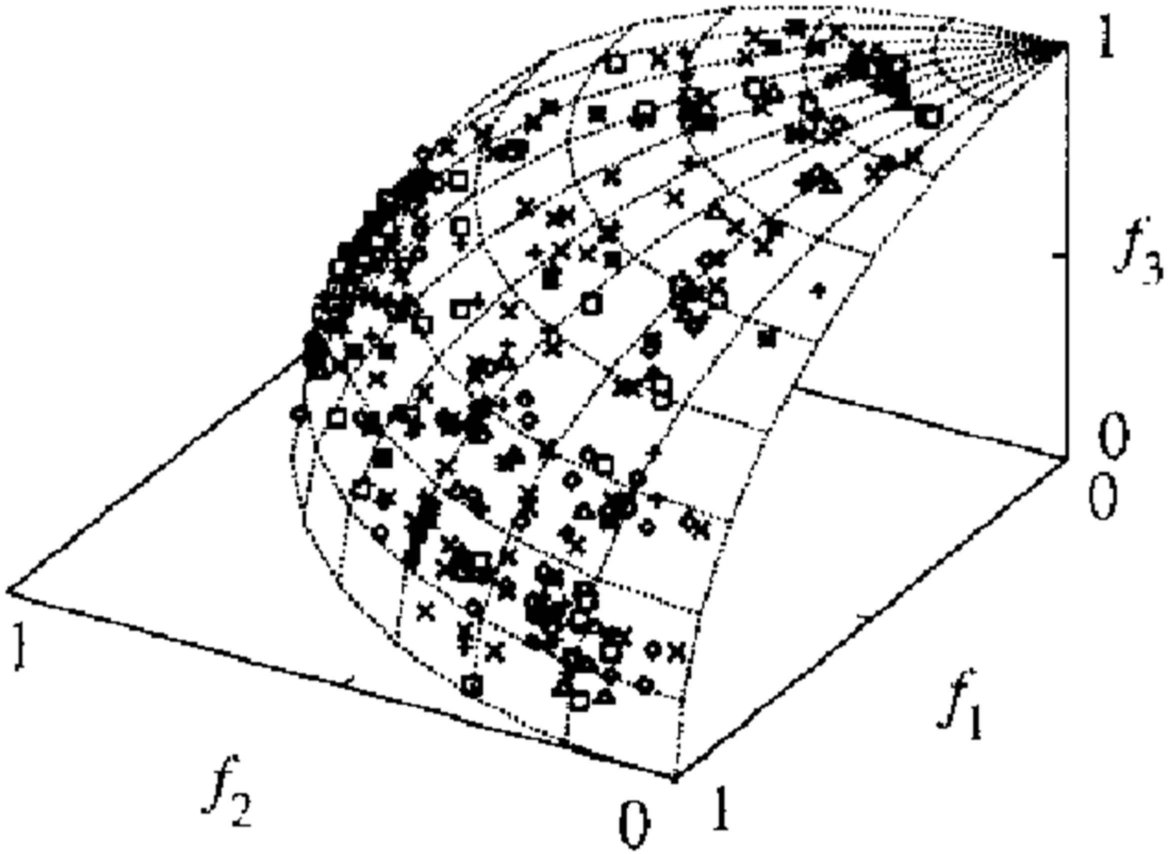
Candidate	f_1	f_2	f_3	f_4
1 (dominated by: 2,4,5)	5	6	3	10
2 (dominated by: 5)	4	6	3	10
3 (non-dominated)	5	5	2	11
4 (non-dominated)	5	6	2	10
5 (non-dominated)	4	5	3	9

Διαχωρίζοντας τις επιφάνειες Pareto (pareto fronts)



(Fonseca and Fleming 1993)

Pareto Front



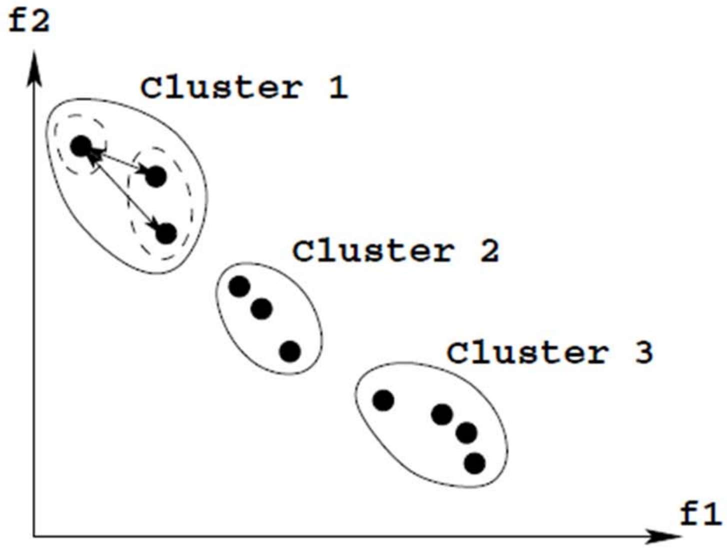
(Tamaki *et al.* 1996)

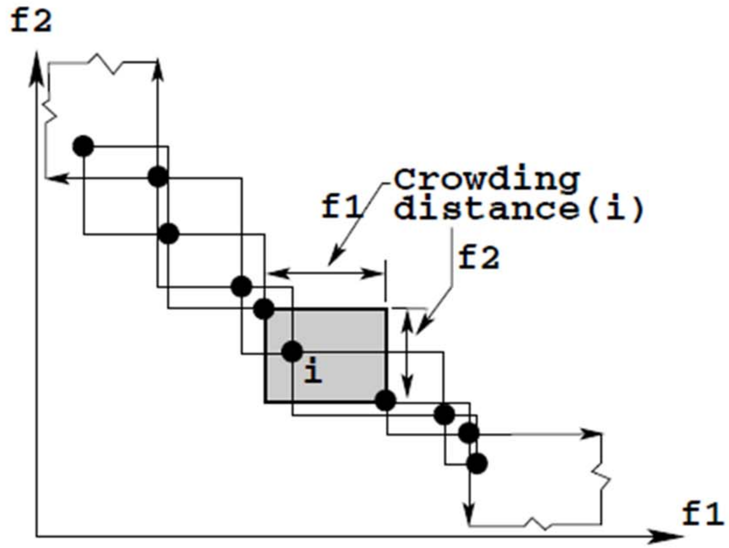
Γενετικοί τελεστές με πληθυσμούς Pareto: Επιλογή

- Pareto Ranking (Pareto Sorting)
- Tournament Selection with Dominance
  - pair wise comparison against a comparison set based on dominance
- Pareto Reservation (Elitism)
  - carry non-dominated candidates forward from previous generation
  - use additional selection method to regulate population size

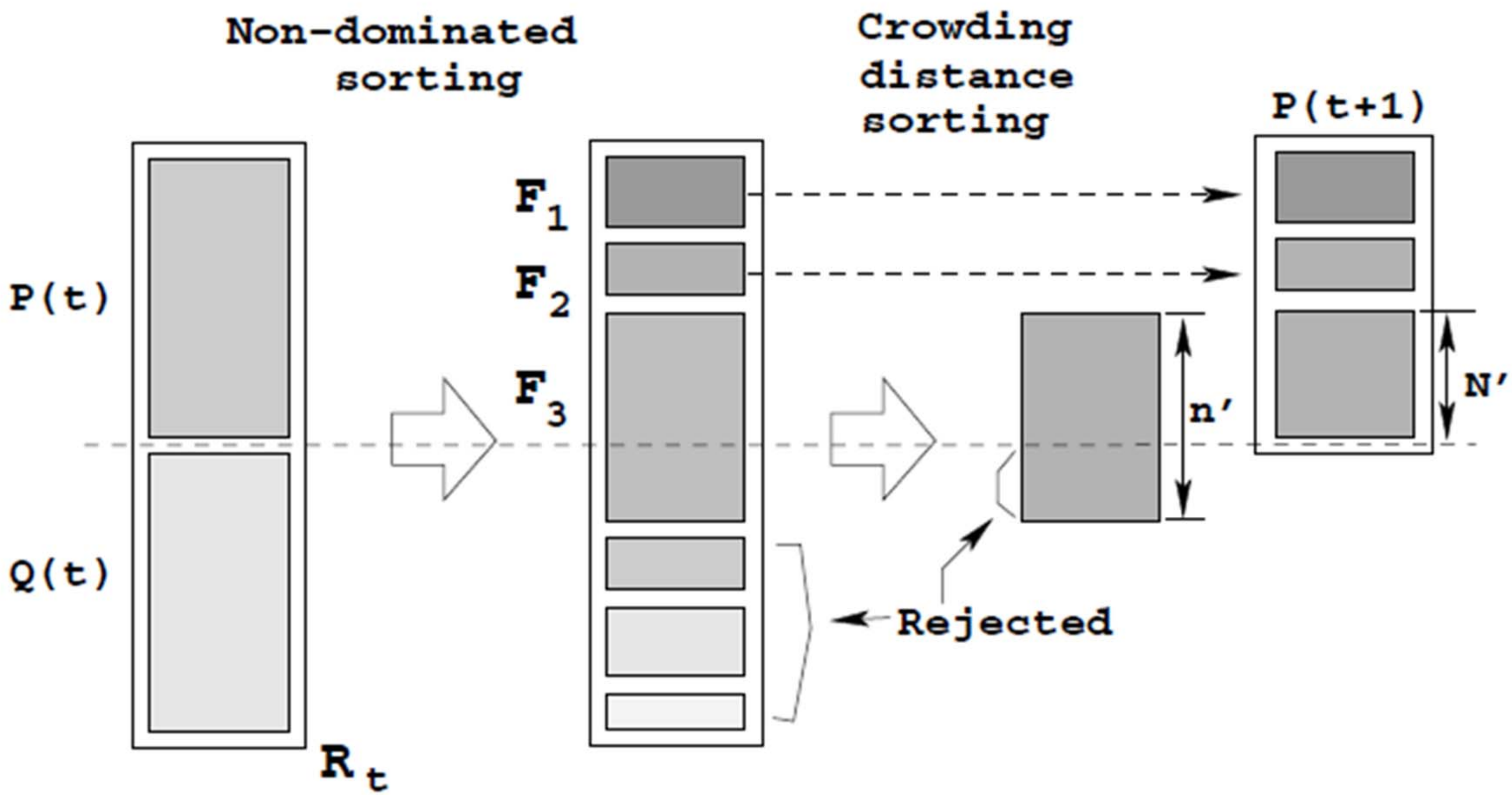
Το πρόβλημα της διασποράς (genetic diversity)

- Οι λύσεις εκτός από μη-κυριαρχούμενες πρέπει να μπορούν να απεικονίζουν ικανοποιητικά το Pareto front
- Άρα δεν έχουν νόημα 100 λύσεις στο ίδιο σημείο του χώρου (clustered) – είναι καλύτερα διασπαρμένες σε όλο το χώρο..





NSGAII

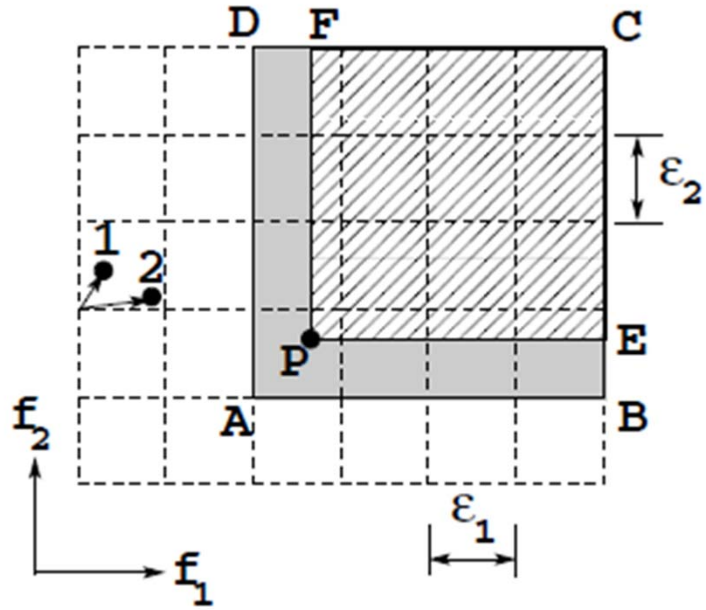


Non-dominated sorting
Crowding distance sorting
P(t+1)
P(t)
Q(t)
R_t
F_1
F_2
F_3
n'
N'
Rejected

Epsilon Domination

(Laumanns , Deb et al., 2002)

- Διαφορετικός τρόπος να πετύχεις διασπορά λύσεων.
- Μόνο μια λύση (τυχαία ή την καλύτερη) σε κάθε κελί
- Προφανώς χάνονται κάποιες καλές λύσεις
- Έχεις όμως καλύτερη διασπορά
- Epsilon = μικρό…



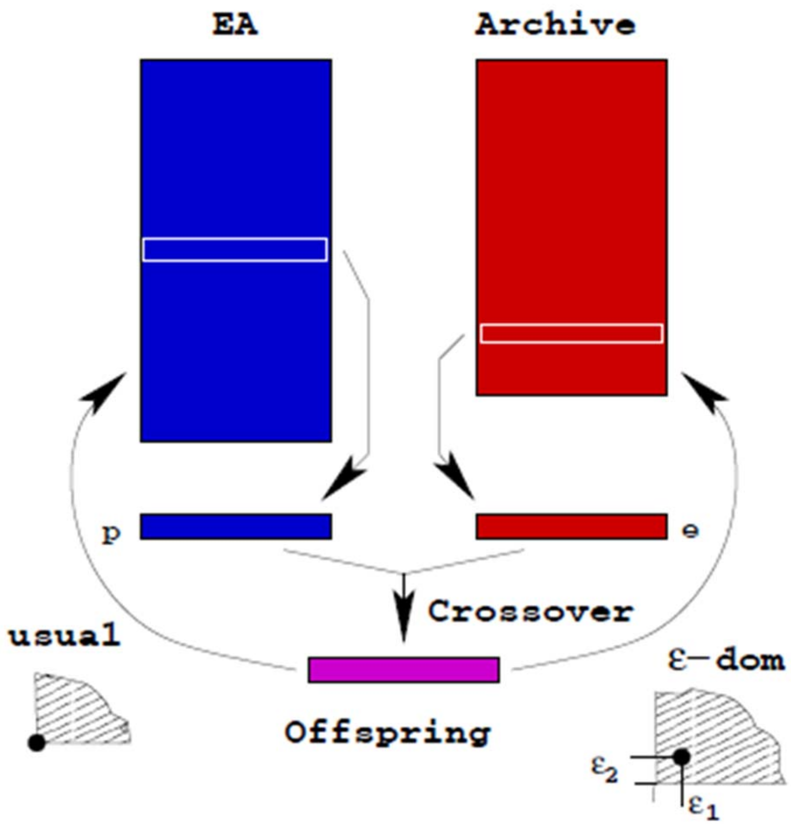
Κατανομή της ευρωστίας: Fitness Sharing

- Περιορίζει τον αριθμό ατόμων εντος της ίδιας περιοχής (niche), κατανέμοντας την ευρωστία τους
- Κατανομή ατόμων σε περιοχές ανάλογα με την ευρωστία της περιοχής.
- Πρέπει να θέσουμε υποκειμενικά το μέγεθος της περιοχής σ_{share} είτε στον χώρο του γενότυπου είτε στου φαινότυπου
- Μετά τρέχουμε κανονικά τον ΓΑ

$$f'(i) = \frac{f(i)}{\sum_{j=1}^{\mu} sh(d(i,j))} \qquad sh(d) = \begin{cases} 1 - d/\sigma & d < \sigma \\ 0 & otherwise \end{cases}$$

Ελιτισμός και Archiving

- Τα καλύτερα άτομα τα κρατάμε σε χωριστό αρχείο (ελιτισμός)
- Δεν χρειάζεται να ξανατρέξουμε την αντικειμενική (archiving)
- Ελέγχουμε τη διασπορά μόνο στις καλύτερες (και τελικές) τιμές



Matlab Toolbox

[x fval] = ga(fitnessfun, nvars, options) where

fitnessfun is the fitness function.

nvars is the number of independent variables for the fitness function.

options is a structure containing options for the genetic algorithm. If you do not pass in this argument, ga uses its default options.

Αποτελέσματα:

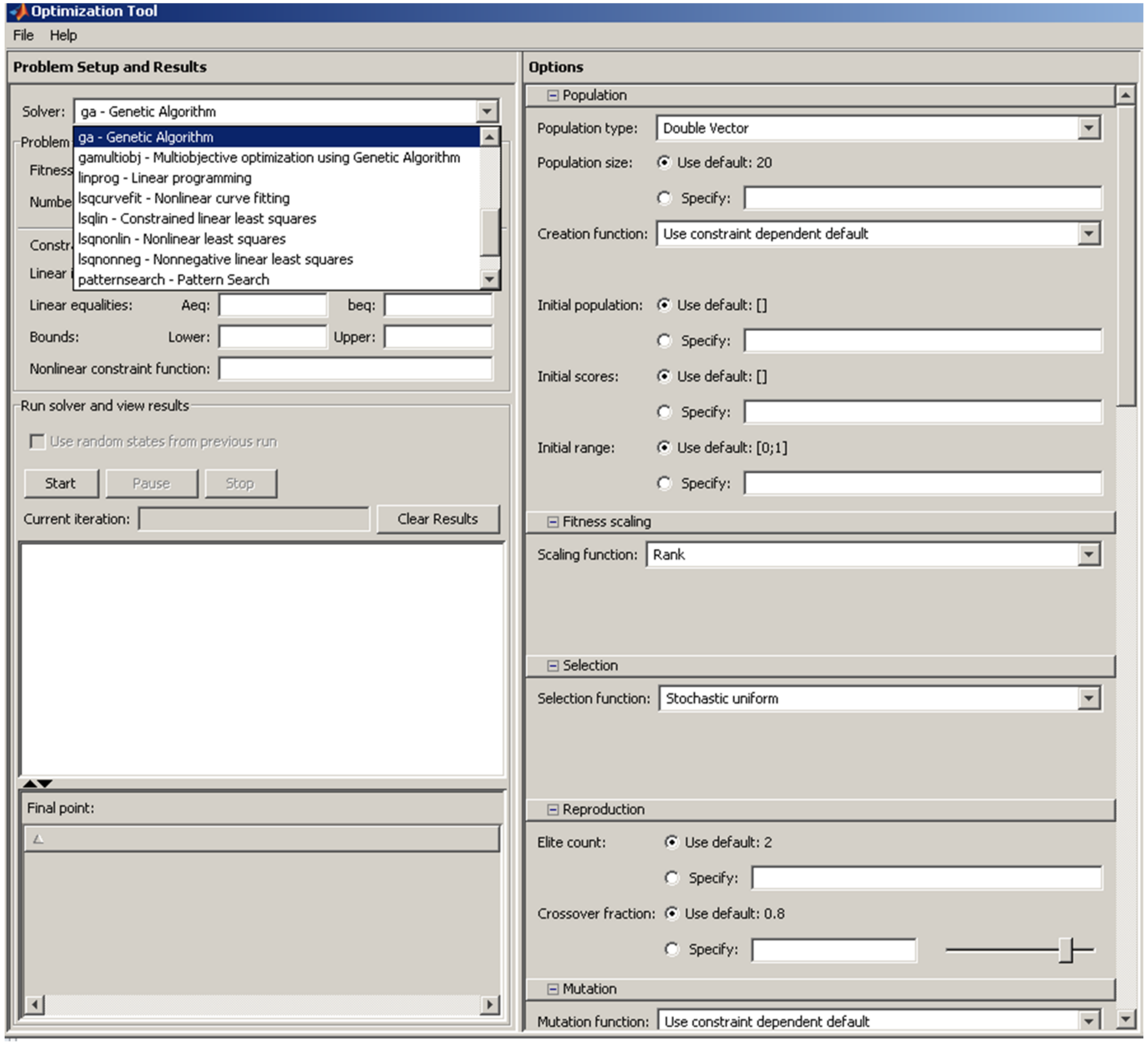
x — Point at which the final value is attained.

fval — Final value of the fitness function.

Ή μέσω toolbox: **optimtool**
(σε παλιότερες version: gatool)

Fitness function — The objective function you want to minimize. Enter the fitness function as fitnessfun.m ie. an M-file that computes the fitness function.

Number of Variables – The number of variables in the given fitness function should be given.



POPULATION

In this case <u>population type, population size and creation function</u> may be selected.

The initial population and initial score may be specified, if not, the 'Ga tool' creates them. The initial range should be given.

FITNESS SCALING

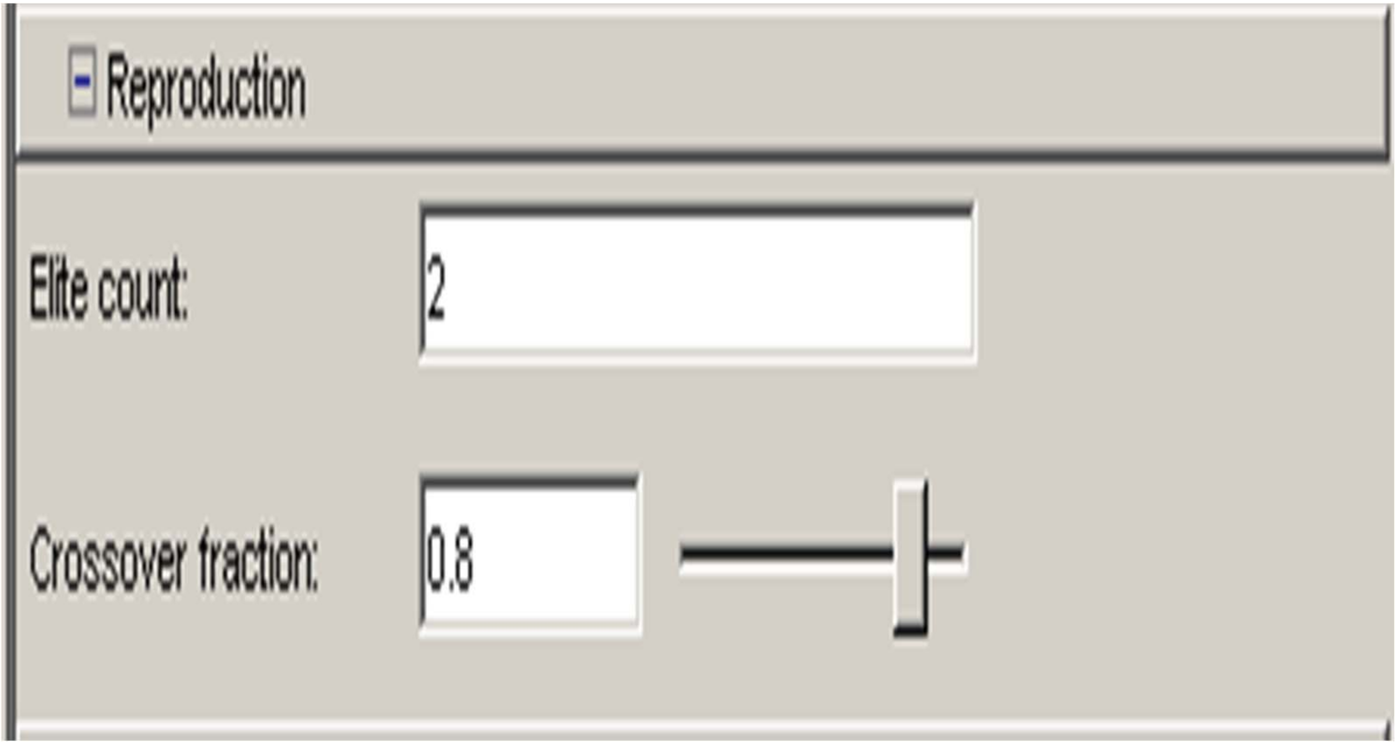
The fitness scaling should be any of the following

a. **Rank**
b. **Proportional**
c. **Top**
d. **Shift Linear**
e. **Custom**

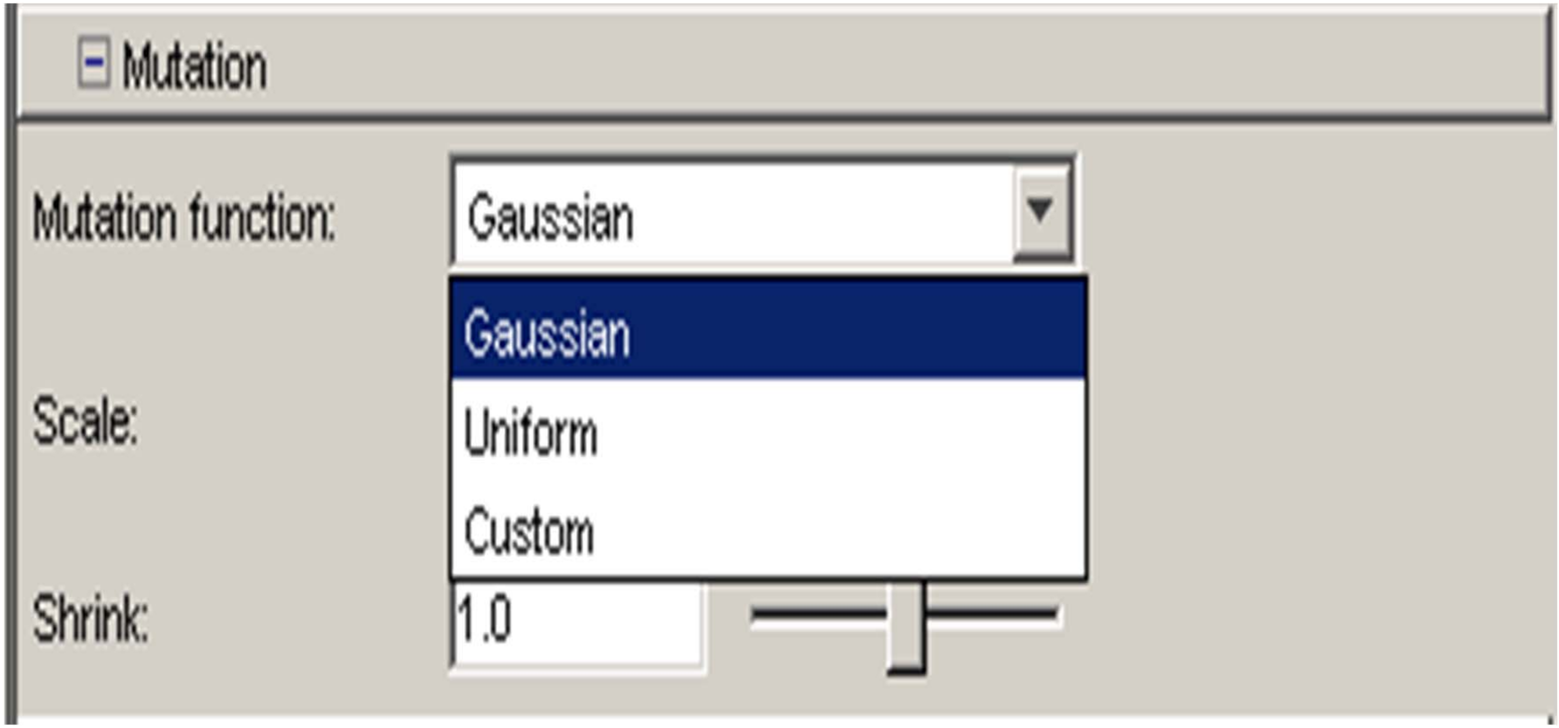
SELECTION


Selection
Selection function:
Stochastic uniform
Stochastic uniform
Remainder
Uniform
Roulette
Tournament
Custom
Reproduction
Mutation

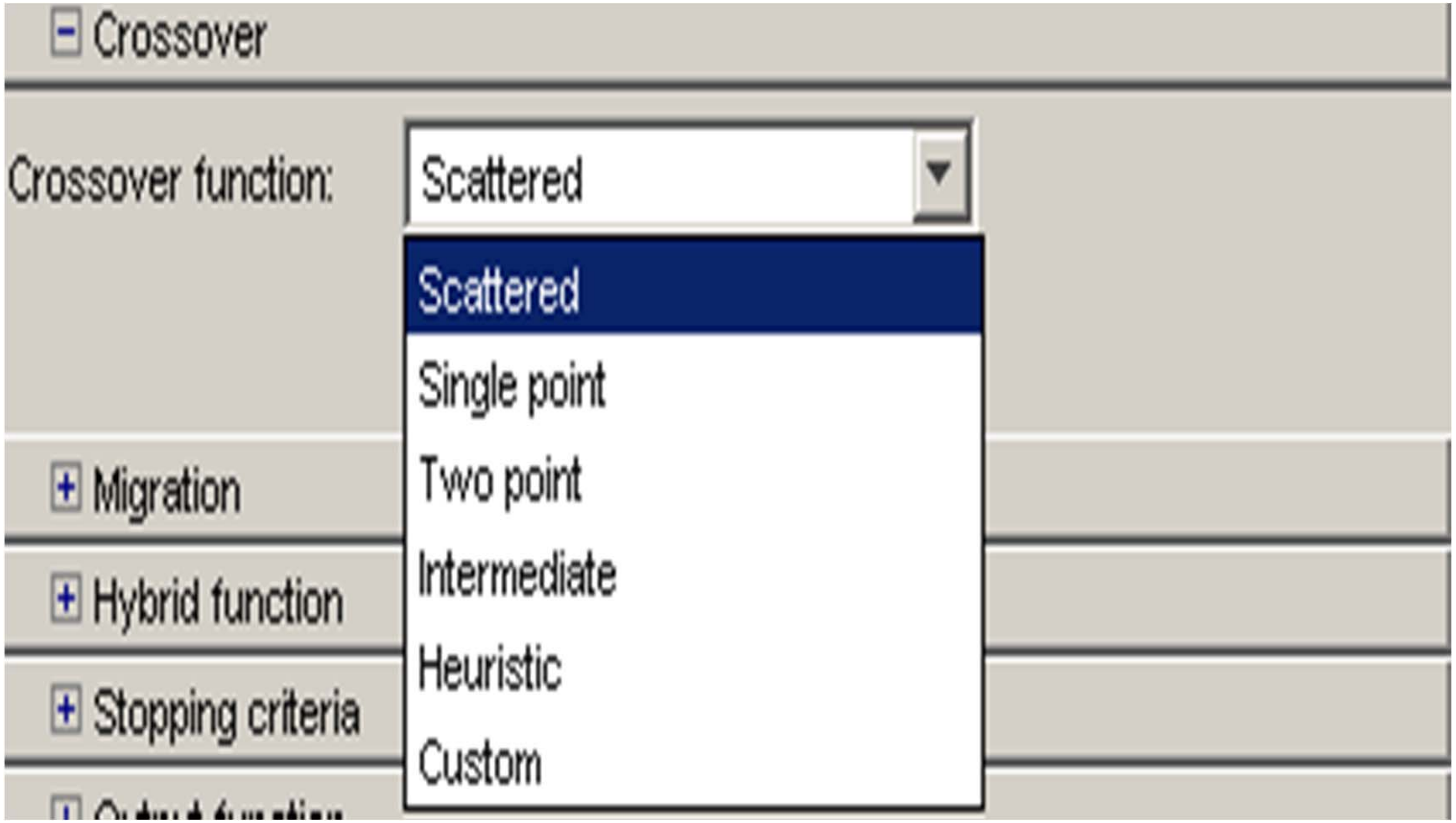
REPRODUCTION



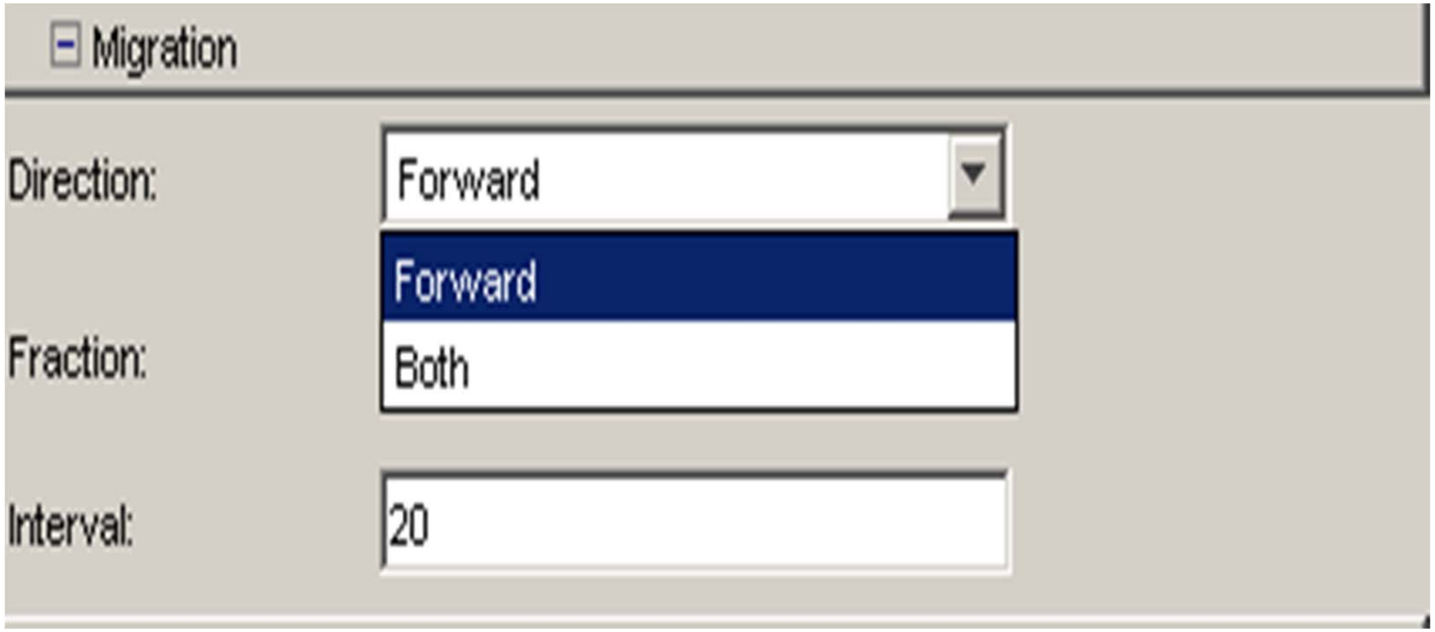
MUTATION


Mutation
Mutation function:
Gaussian
Gaussian
Uniform
Custom
Scale:
Shrink:
1.0

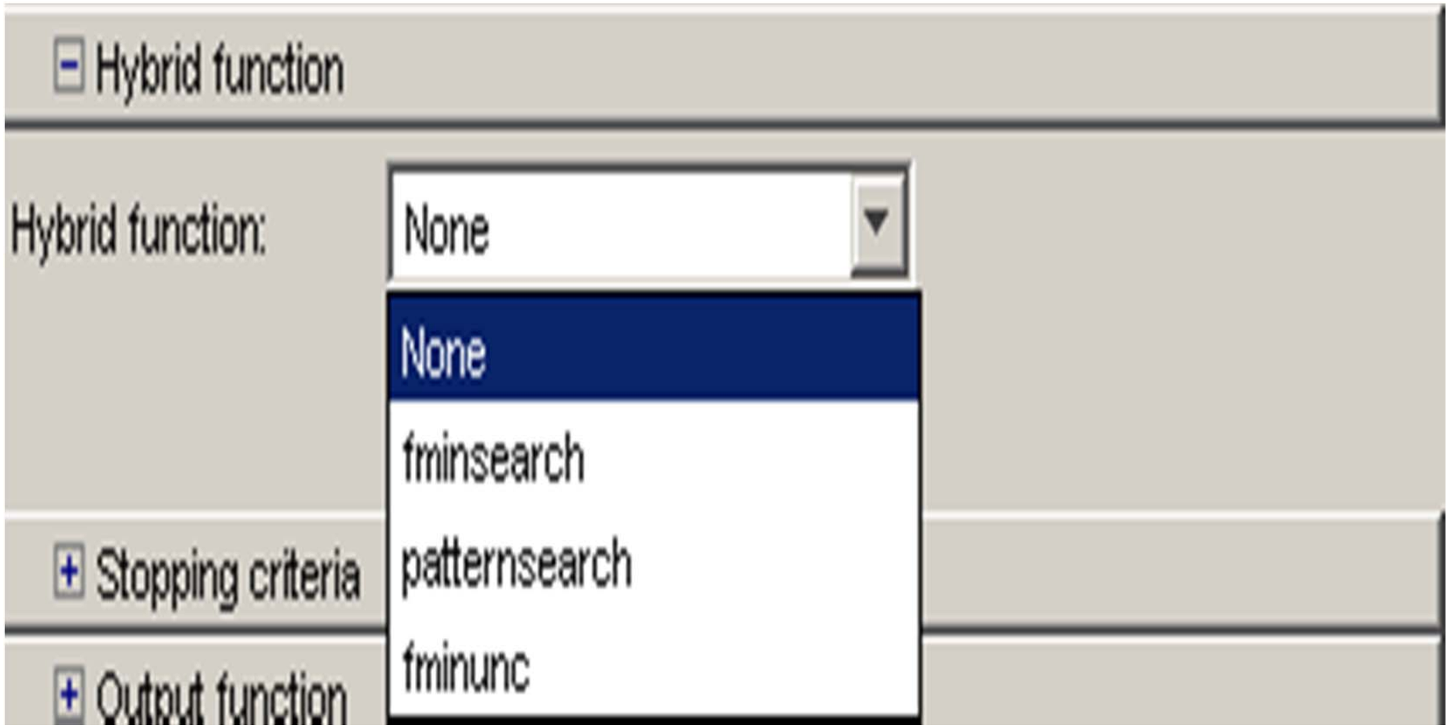
CROSSOVER


Crossover
Crossover function:
Scattered
Scattered
Single point
Two point
Intermediate
Heuristic
Custom
Migration
Hybrid function
Stopping criteria

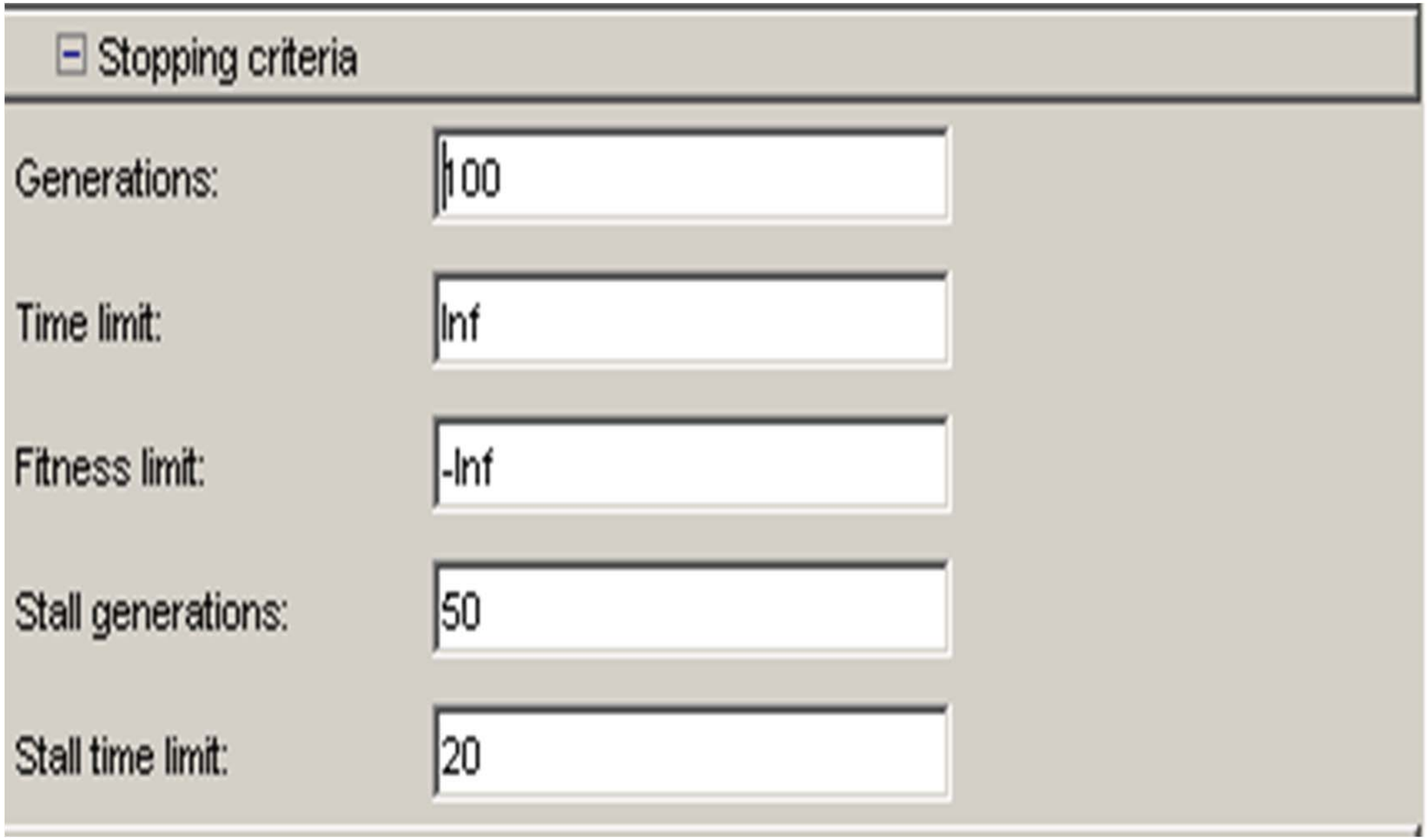
MIGRATION


Migration
Direction:
Forward
Forward
Both
Fraction:
Interval:
20

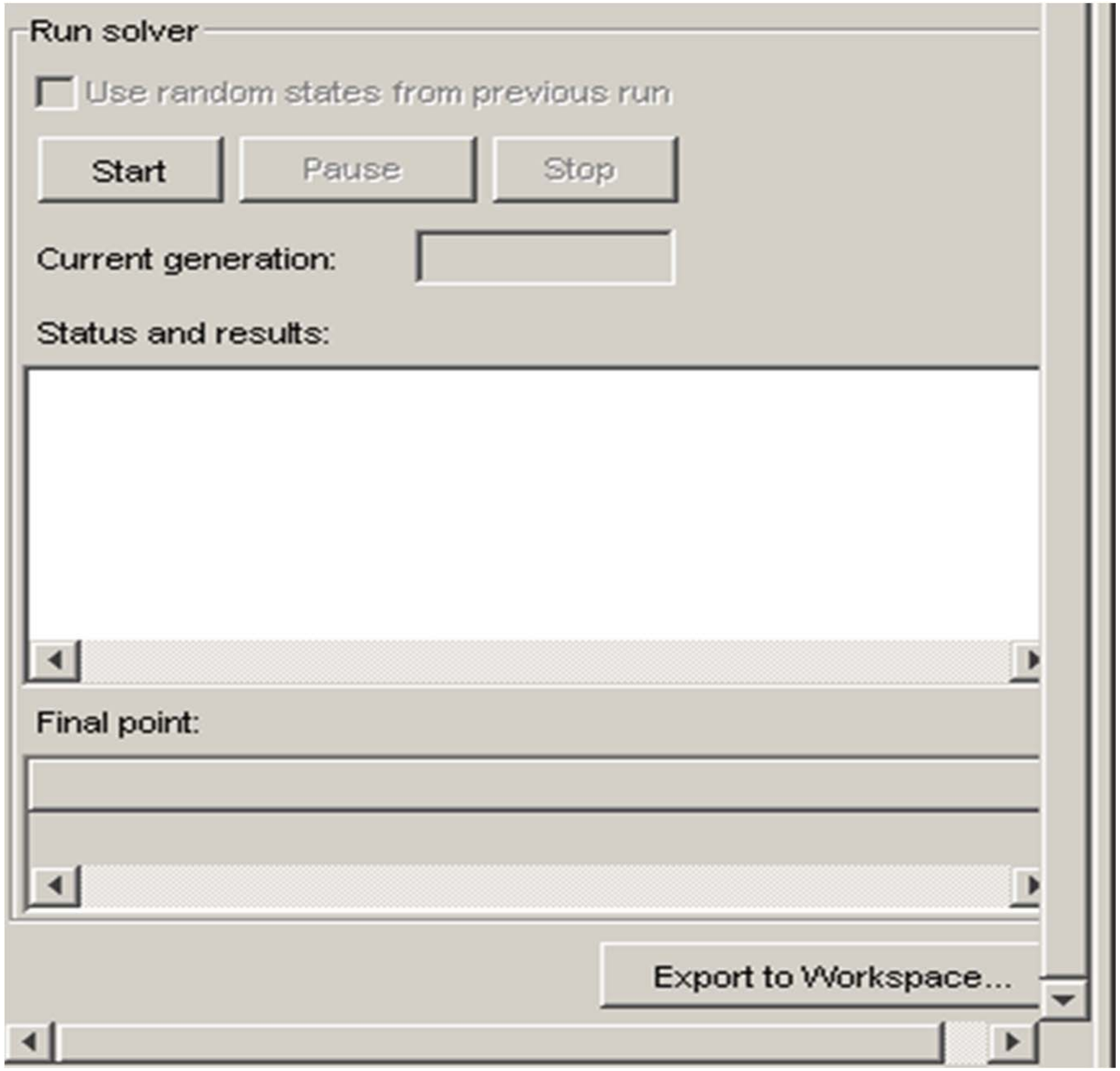
HYBRID FUNCTION



STOPPING CONDITION

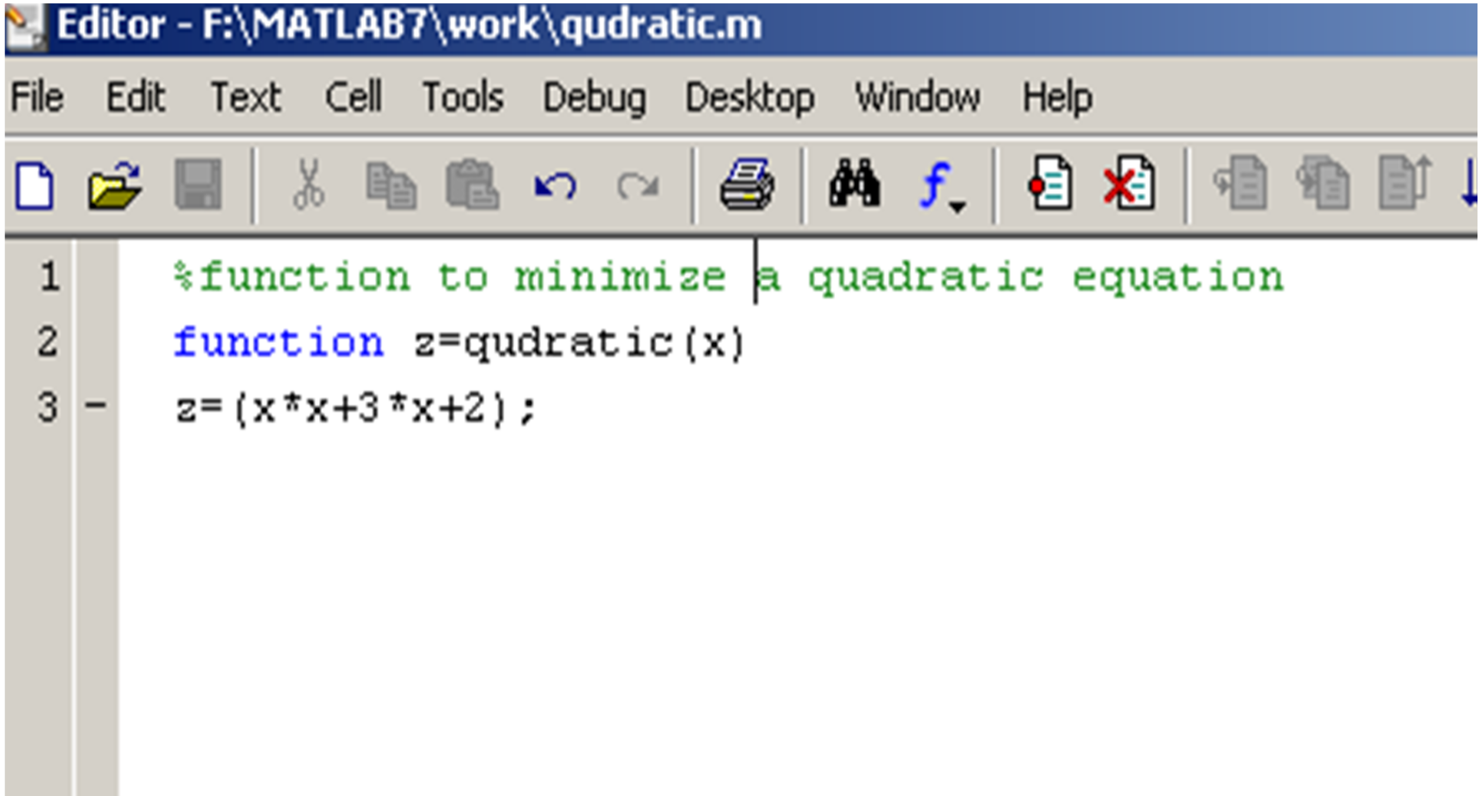


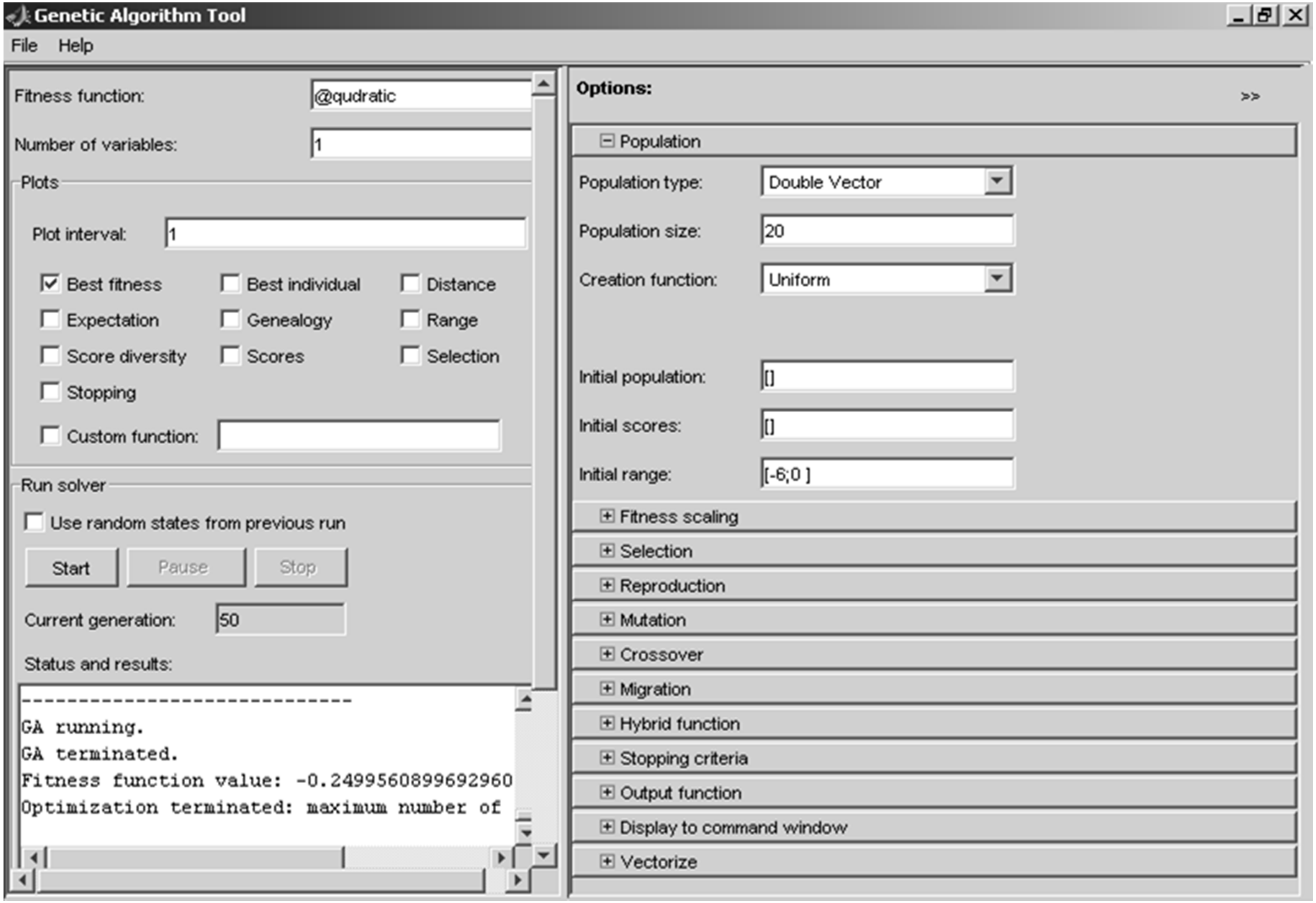
RUNNING AND SIMULATION

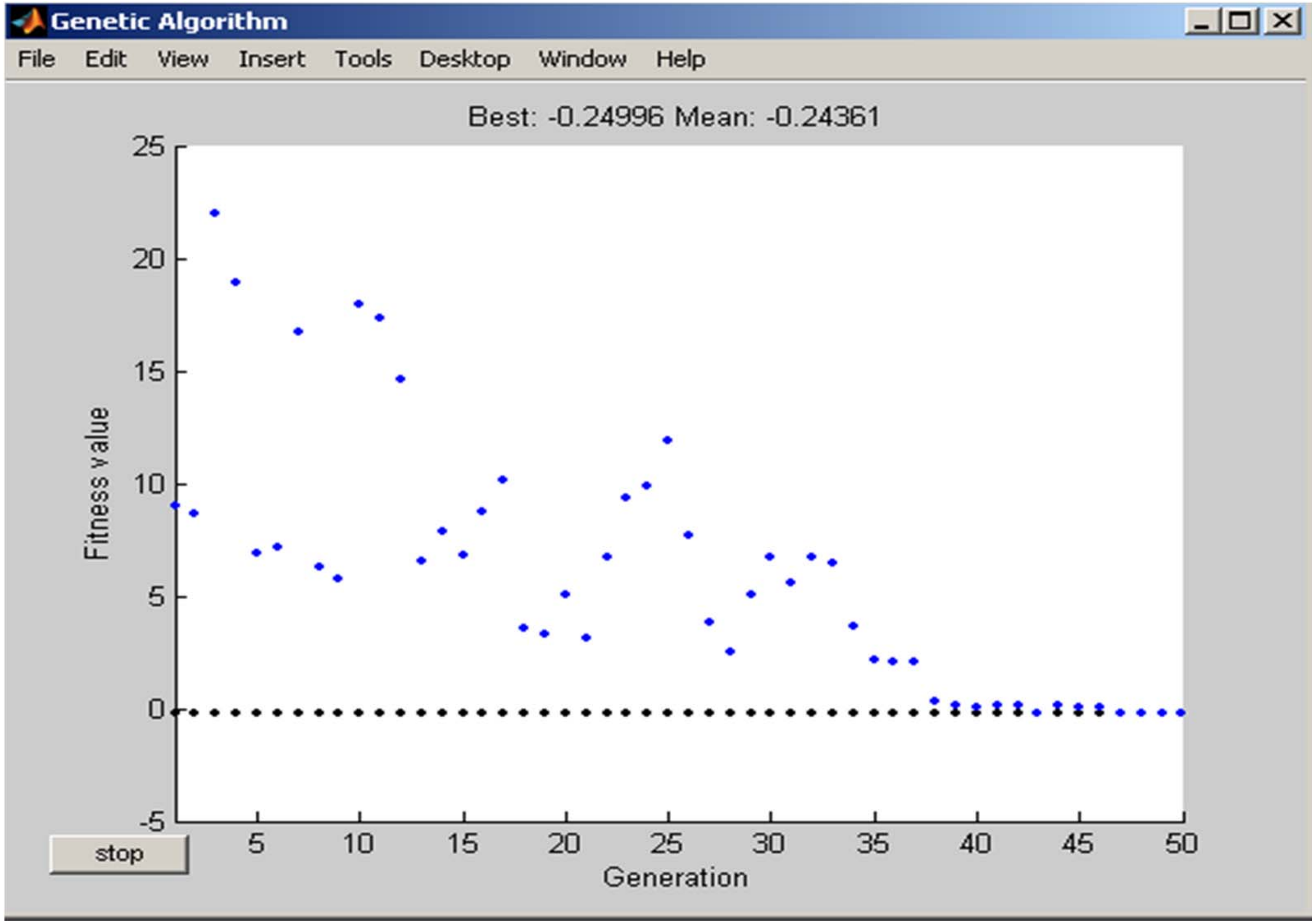


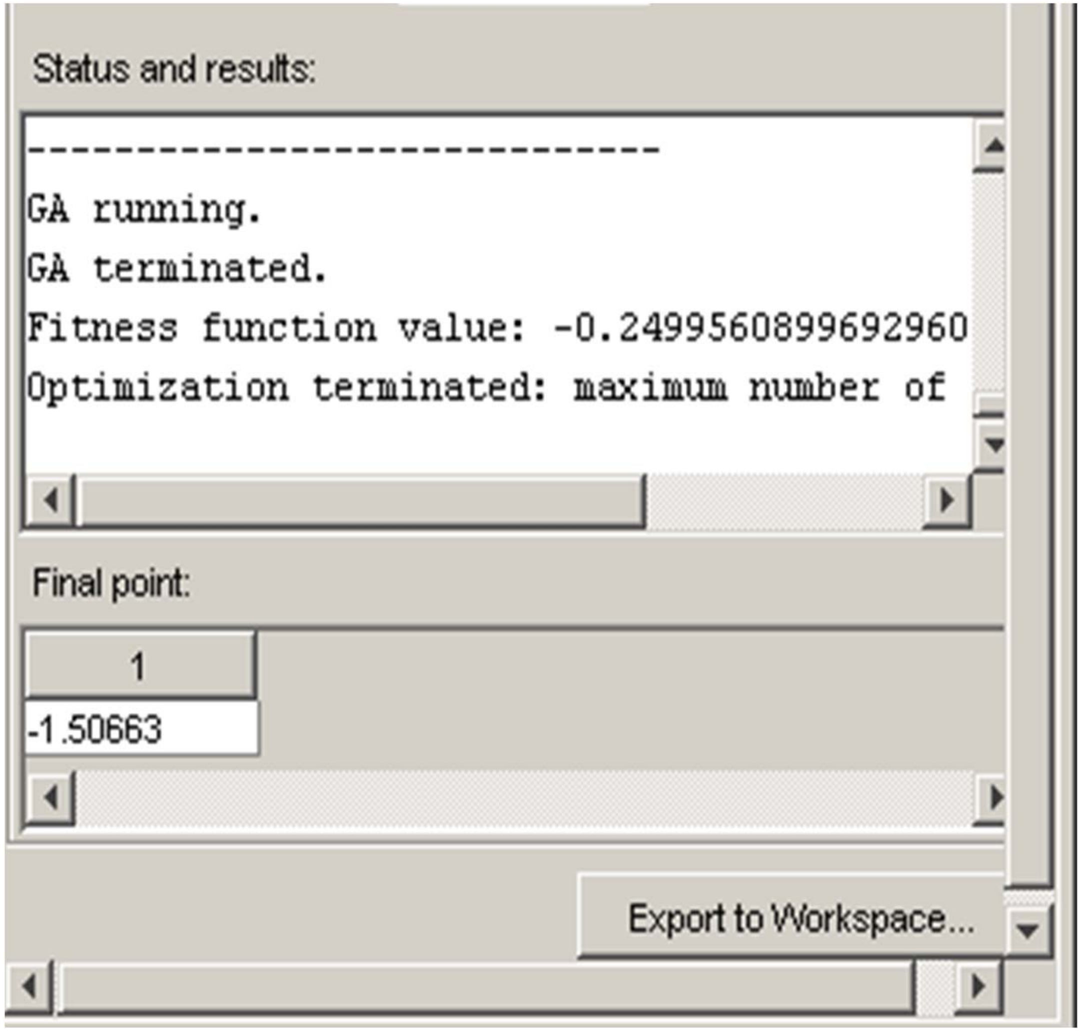
Παράδειγμα

$f(x) = x^2+3x+2$ ως function σε m-file




Genetic Algorithm Tool
File
Help
Fitness function:
@qudratic
Number of variables:
1
Plots
Plot interval:
1
Best fitness
Best individual
Distance
Expectation
Genealogy
Range
Score diversity
Scores
Selection
Stopping
Custom function:
Run solver
Use random states from previous run
Start
Pause
Stop
Current generation:
50
Status and results:
GA running.
GA terminated.
Fitness function value: -0.2499560899692960
Optimization terminated: maximum number of
Options:
Population
Population type:
Double Vector
Population size:
20
Creation function:
Uniform
Initial population:
[]
Initial scores:
[]
Initial range:
[-6;0]
Fitness scaling
Selection
Reproduction
Mutation
Crossover
Migration
Hybrid function
Stopping criteria
Output function
Display to command window
Vectorize

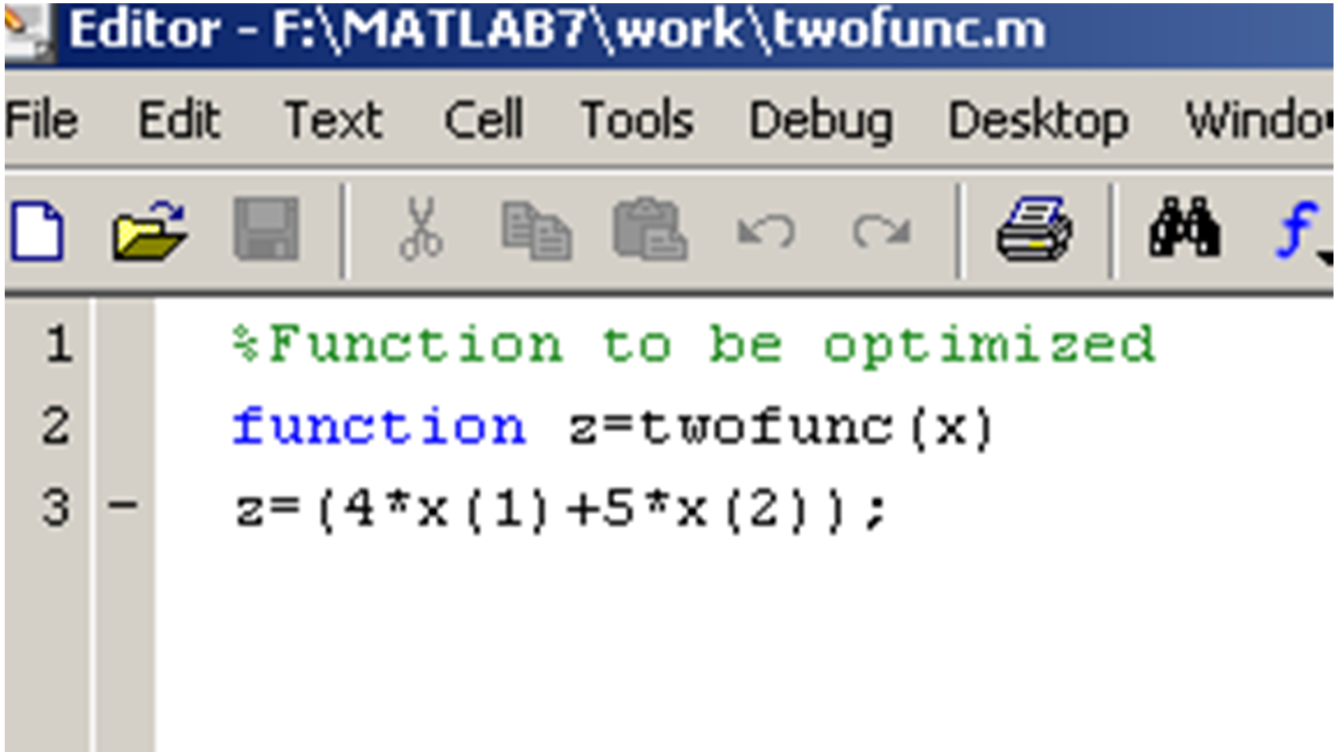

Genetic Algorithm
File Edit View Insert Tools Desktop Window Help
Best: -0.24996 Mean: -0.24361
Fitness value
Generation
stop
25
20
15
10
5
0
-5
5 10 15 20 25 30 35 40 45 50

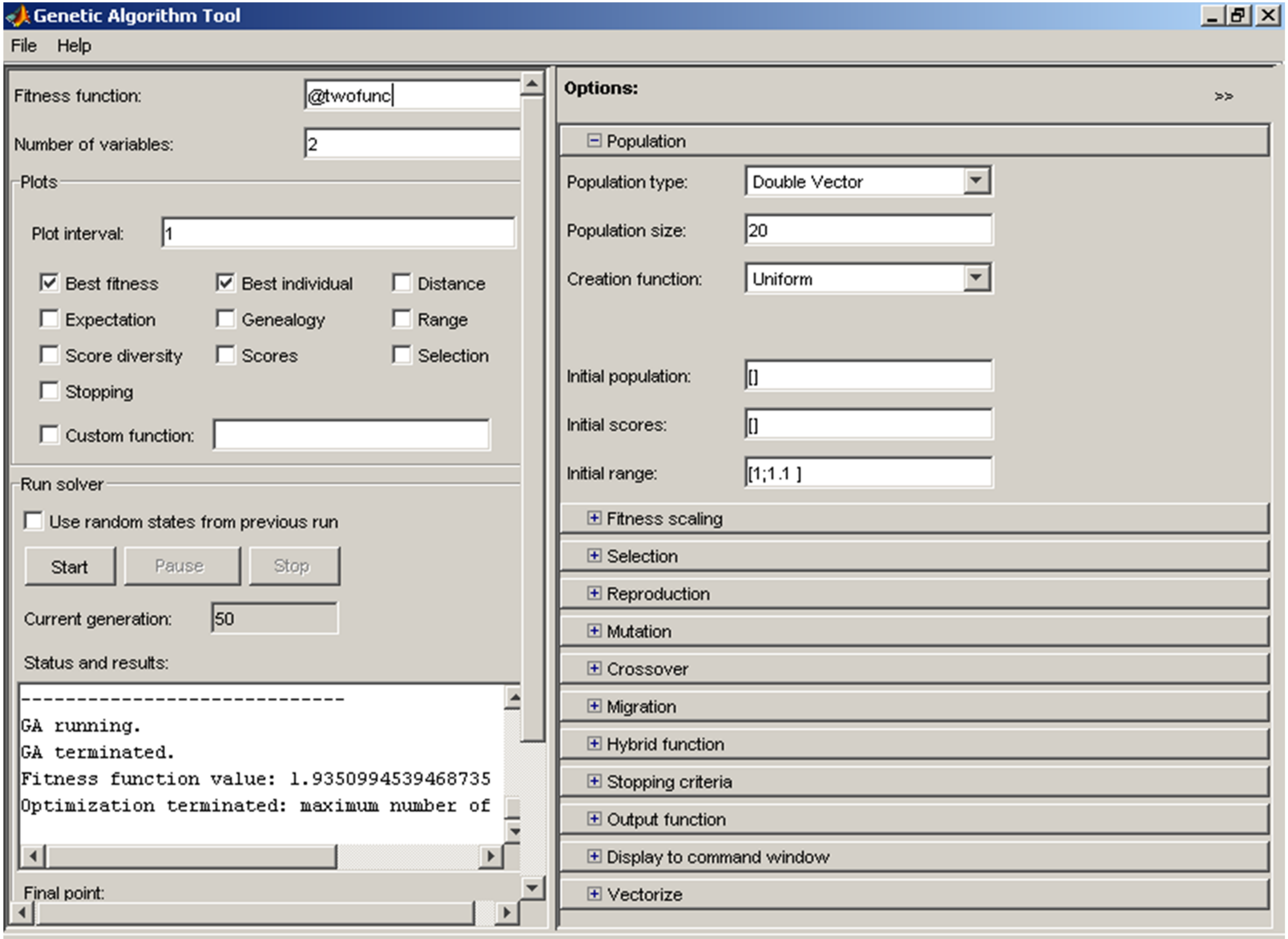

Status and results:

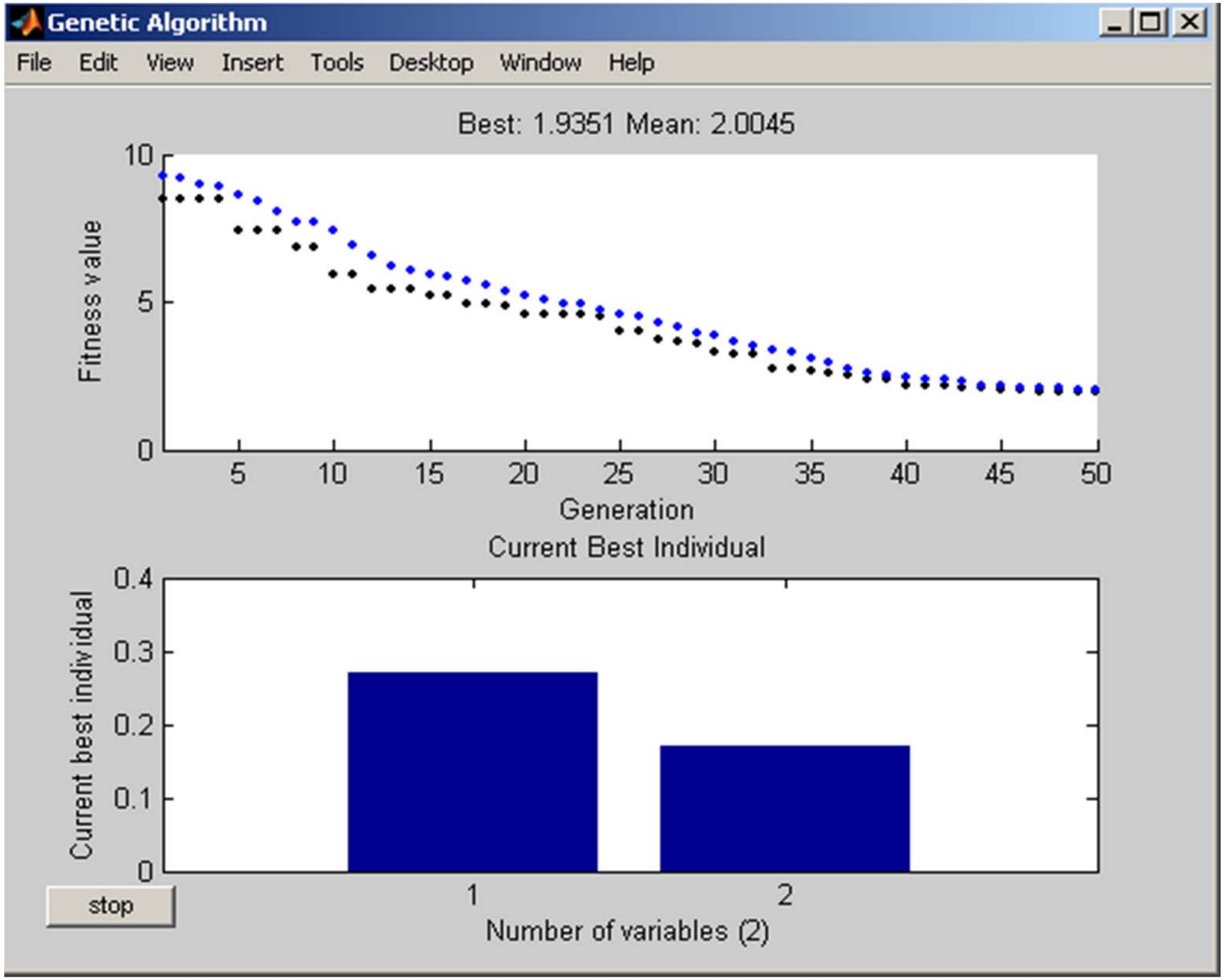
GA running.
GA terminated.
Fitness function value: -0.2499560899692960
Optimization terminated: maximum number of
Final point:
1
-1.50663
Export to Workspace...

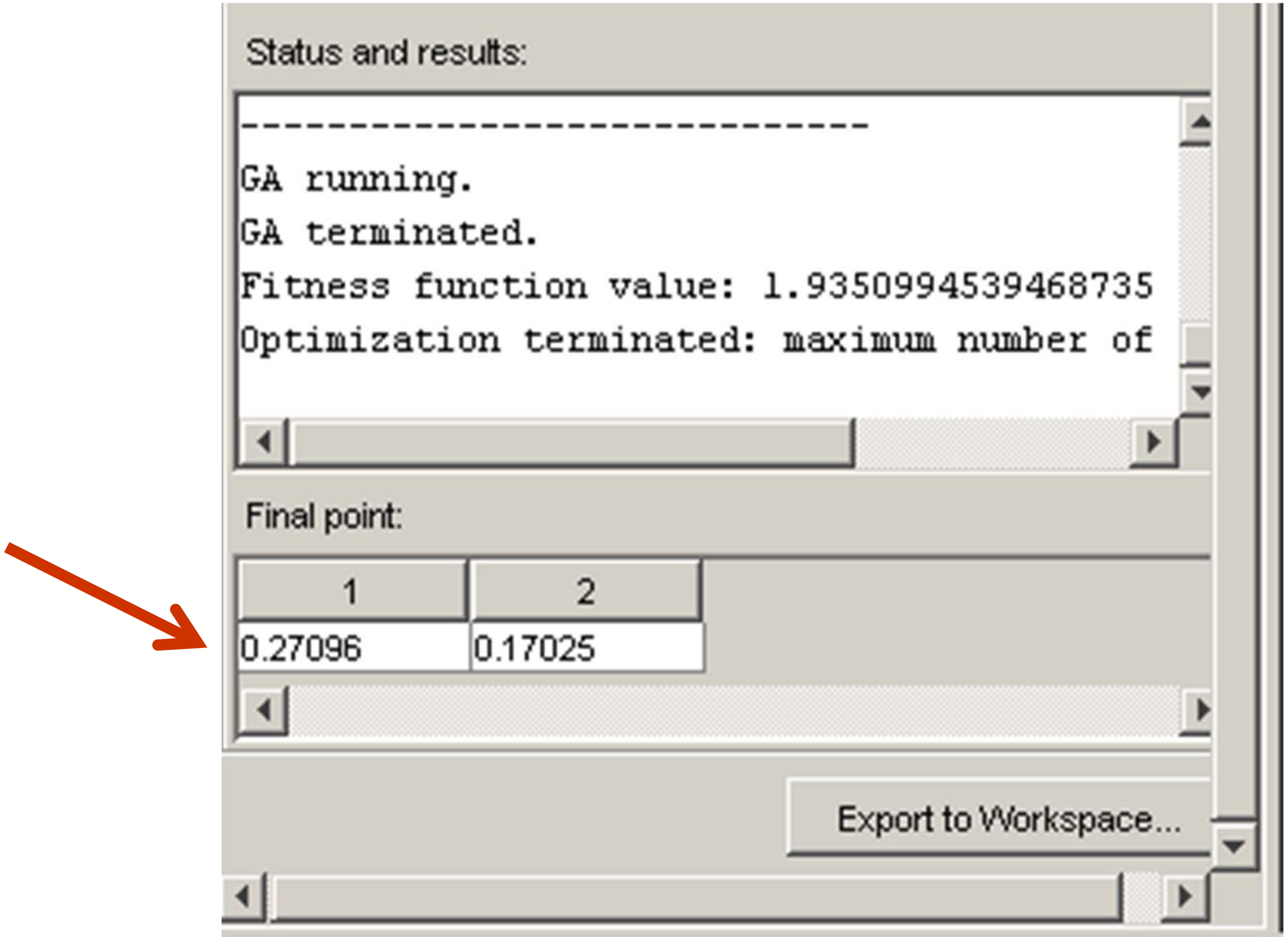
Παράδειγμα

$f(x1,x2)=4x_1+5x_2$ σε m-file σαν function




Genetic Algorithm Tool
File
Help
Fitness function:
@twofunc
Number of variables:
2
Plots
Plot interval:
1
Best fitness
Best individual
Distance
Expectation
Genealogy
Range
Score diversity
Scores
Selection
Stopping
Custom function:
Run solver
Use random states from previous run
Start
Pause
Stop
Current generation:
50
Status and results:
GA running.
GA terminated.
Fitness function value: 1.9350994539468735
Optimization terminated: maximum number of
Final point:
Options:
Population
Population type:
Double Vector
Population size:
20
Creation function:
Uniform
Initial population:
[]
Initial scores:
[]
Initial range:
[1;1.1]
Fitness scaling
Selection
Reproduction
Mutation
Crossover
Migration
Hybrid function
Stopping criteria
Output function
Display to command window
Vectorize


Genetic Algorithm
File Edit View Insert Tools Desktop Window Help
Best: 1.9351 Mean: 2.0045
Fitness value
Generation
Current Best Individual
Current best individual
stop
Number of variables (2)

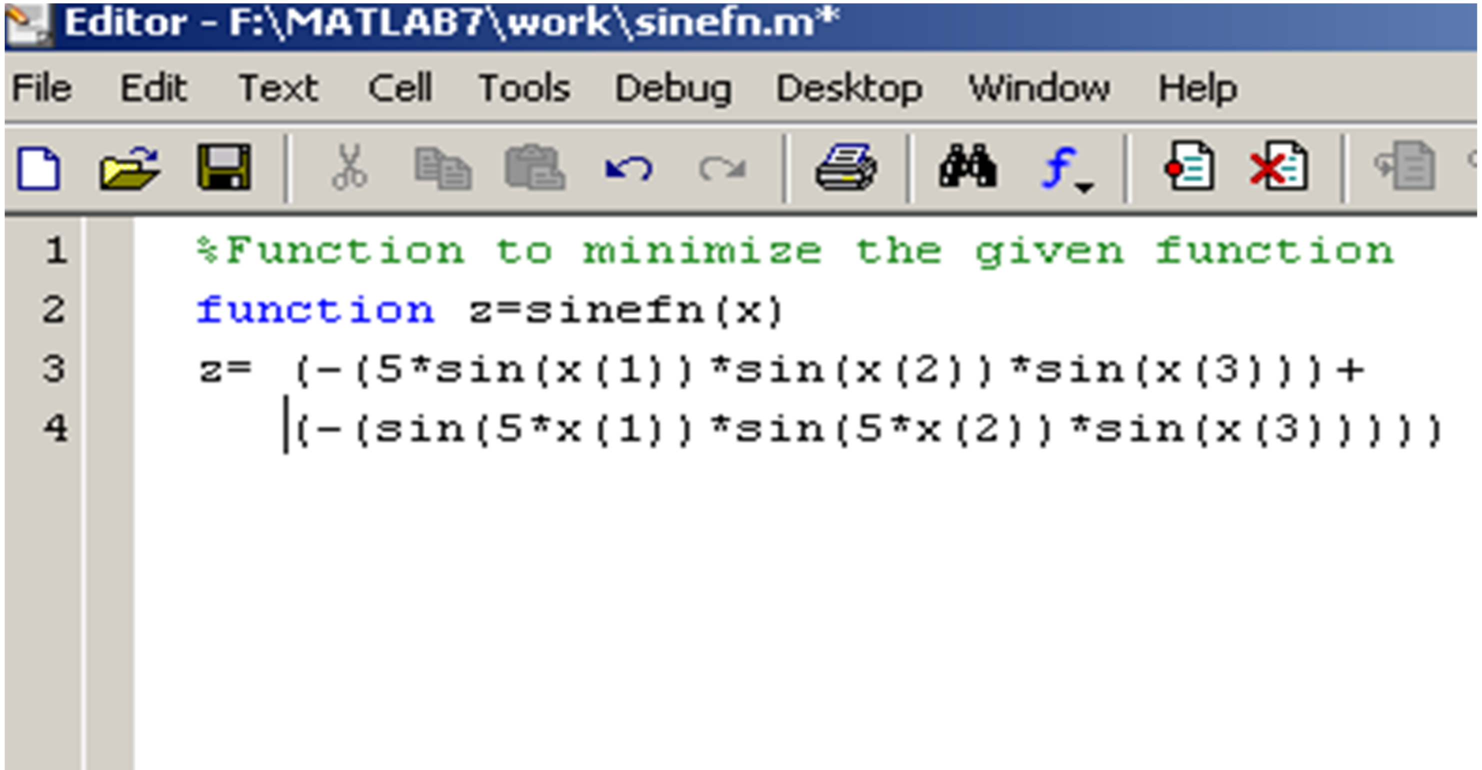

Status and results:

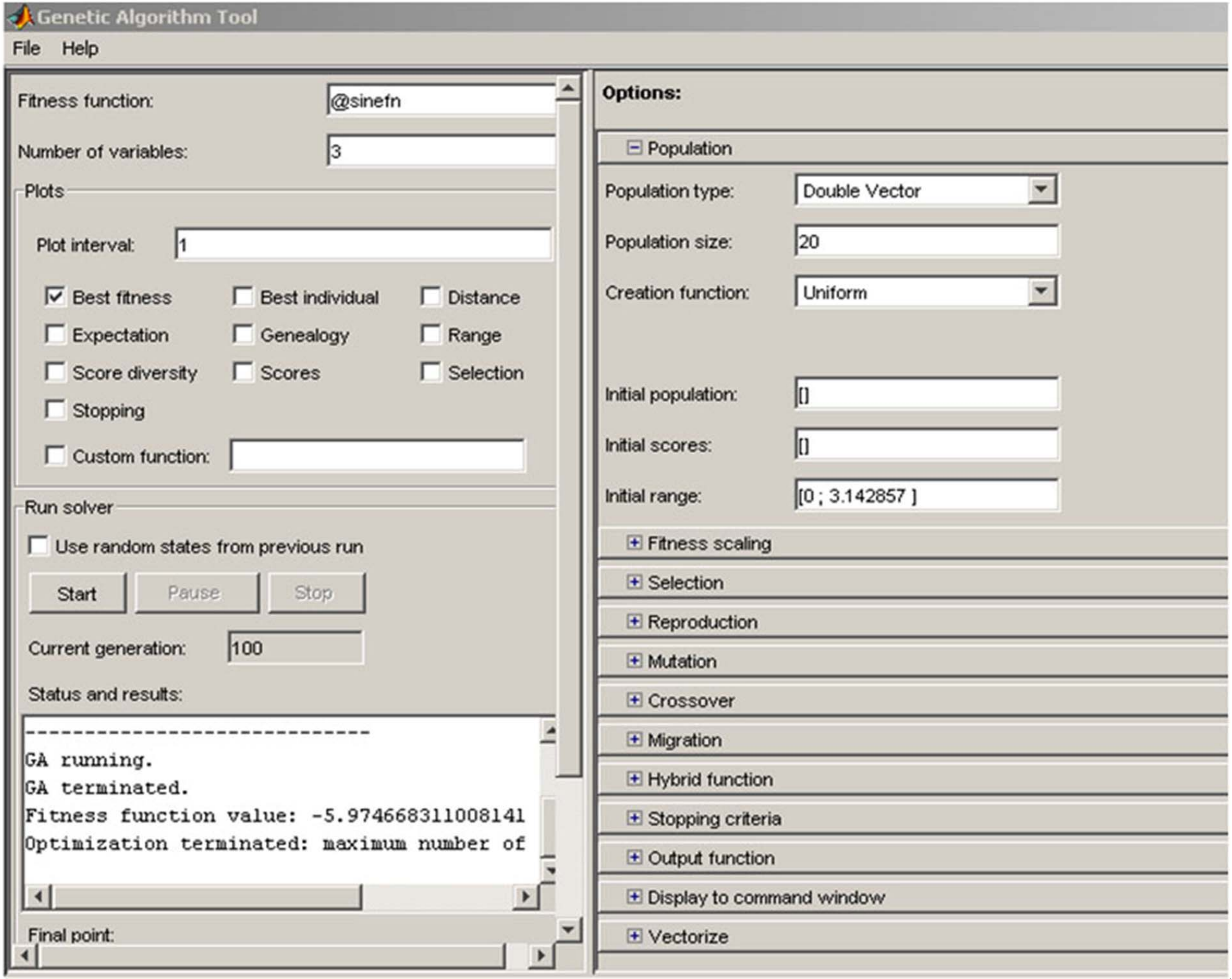
GA running.
GA terminated.
Fitness function value: 1.9350994539468735
Optimization terminated: maximum number of
Final point:
1
2
0.27096
0.17025
Export to Workspace...

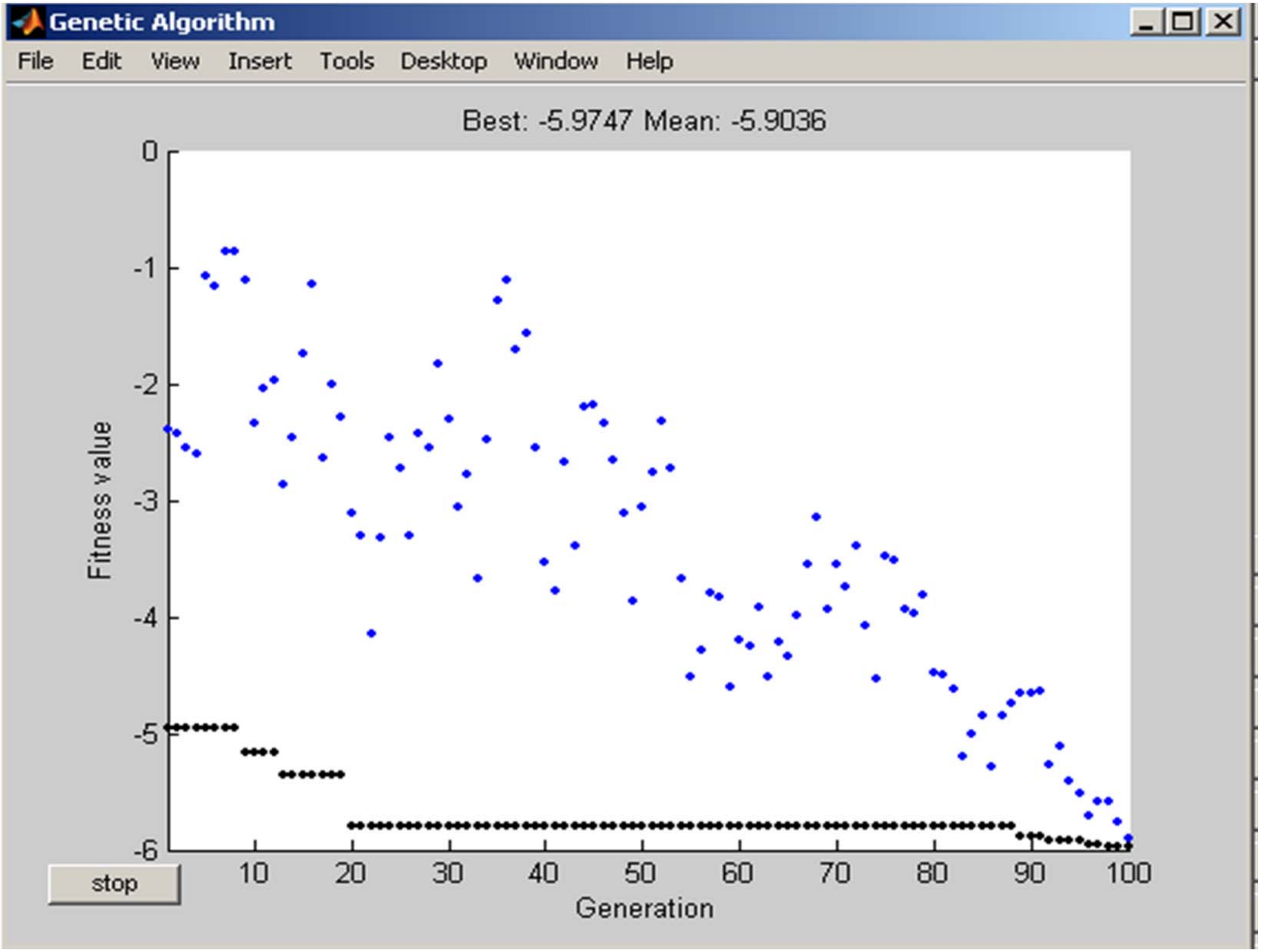
Παράδειγμα

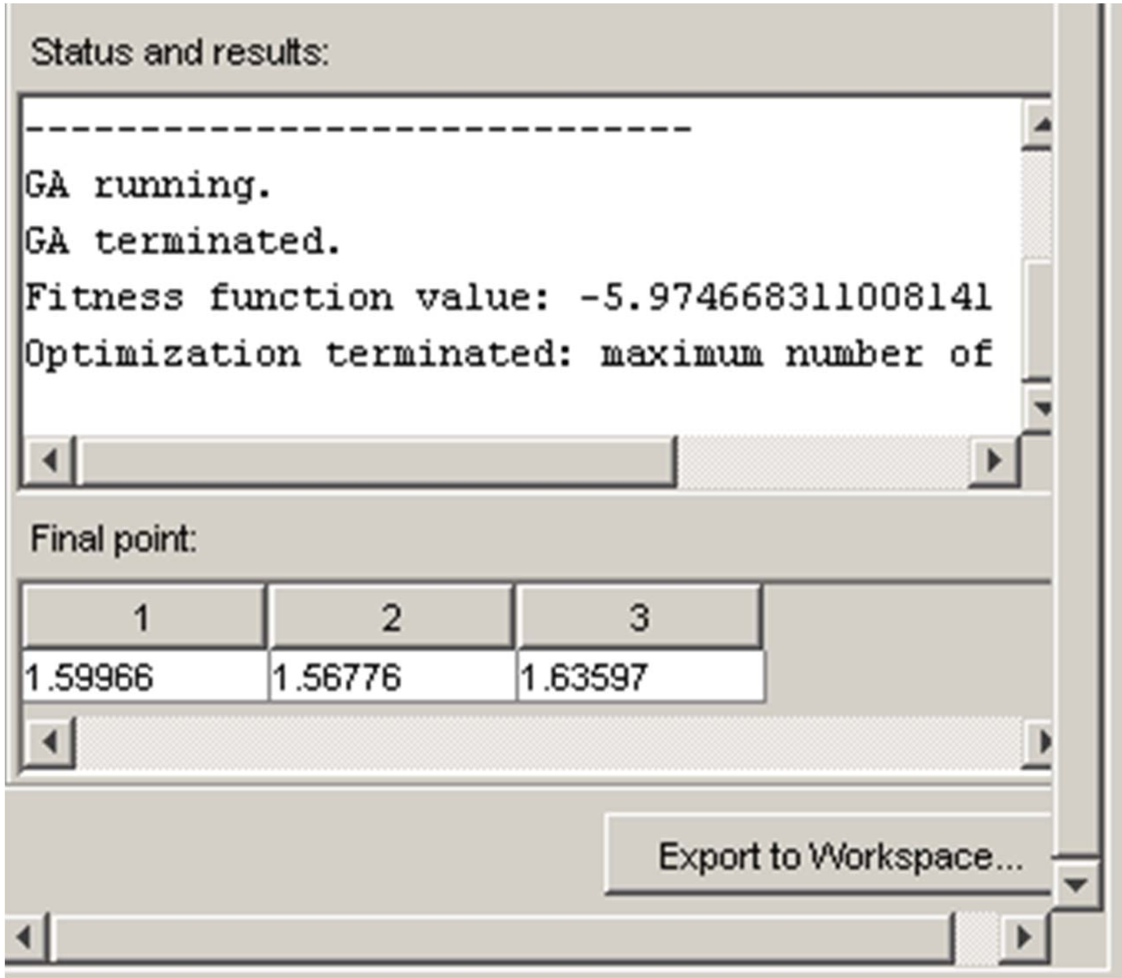
f (x1, x2, x3) = -5sin(x1)sin(x2)sin(x3)+-sin(5x1)sin(5x2)sin(x3)

με 0 <= xi <= pi, for 1<= i <= 3.




Genetic Algorithm Tool
File
Help
Fitness function:
@sinefn
Number of variables:
3
Plots
Plot interval:
1
Best fitness
Best individual
Distance
Expectation
Genealogy
Range
Score diversity
Scores
Selection
Stopping
Custom function:
Run solver
Use random states from previous run
Start
Pause
Stop
Current generation:
100
Status and results:
GA running.
GA terminated.
Fitness function value: -5.974668311008141
Optimization terminated: maximum number of
Final point:
Options:
Population
Population type:
Double Vector
Population size:
20
Creation function:
Uniform
Initial population:
[]
Initial scores:
[]
Initial range:
[0 ; 3.142857]
Fitness scaling
Selection
Reproduction
Mutation
Crossover
Migration
Hybrid function
Stopping criteria
Output function
Display to command window
Vectorize


Genetic Algorithm
File Edit View Insert Tools Desktop Window Help
Best: -5.9747 Mean: -5.9036
0
-1
-2
-3
-4
-5
-6
Fitness value
stop
10 20 30 40 50 60 70 80 90 100
Generation


Status and results:

GA running.
GA terminated.
Fitness function value: -5.974668311008141
Optimization terminated: maximum number of
Final point:
1
2
3
1.59966
1.56776
1.63597
Export to Workspace...

Matlab (Πολυκριτηριακή Βελτιστοποίηση)

- min F(x) = [objective1(x); objective2(x)]
- where
  - objective1(x) = (x+2)^2 - 10
  - objective2(x) = (x-2)^2 + 20

  - And -8<x<8

Δημιουργούμε την αντικειμενική

```
function y = simple_multiobjective(x)
y(1) = (x+2)^2 - 10;
y(2) = (x-2)^2 + 20;
```

Command Line:

```
FitnessFunction = @simple_multiobjective;
numberOfVariables = 1; [x,fval] =
gamultiobj(FitnessFunction,numberOfVariables);
```

Παράδειγμα

```
function y=test(x)
y1=x^2+3*x+2;
y2=x^3-5;
y=[y1,y2];
```

Βιβλιογραφία

- Wendy Williams: Genetic Algorithms: A Tutorial
- Λυκοθανάσης, Σ. Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Παν/μιο Πάτρας (http://edu.eap.gr/pli/pli31/docs/GAs_introduction.pdf)
- Z. *Michalewitz*. *Genetic* Algorithms + Data Structures = Evolution Programs, Springer, Berlin, Germany (1992)
- Coello Coello, C.A. 2002. "Introduction to Evolutionary Multiobjective Optimization." www.cs.cinvestuv.mx/~EVOCINV/download/class1-emoo-eng.pdf
- Fonseca, C.M. and P.J. Fleming. 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Genetic Algorithms: Proceedings of the Fifth International Conference*. S. Forrest, ed. San Mateo, CA, July 1993.