

Υπολογιστική Νοημοσύνη

Ιωάννης Γ. Τσούλος

Τμήμα Πληροφορικής και τηλεπικοινωνιών
Πανεπιστήμιο Ιωαννίνων

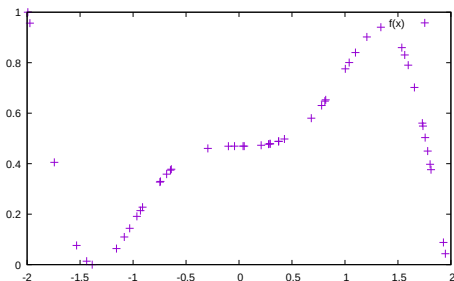
2025

Περίληψη

- 1 Ειδικά θέματα Γραμματικής Εξέλιξης
 - Μάθηση συναρτήσεων
 - Κατασκευή χαρακτηριστικών
 - Χρήση κατασκευής χαρακτηριστικών στο COVID
 - Κατασκευή ΤΝΔ
- 2 Προσομοιούμενη Ανόπτυση
- 3 Διαφορική Εξέλιξη
- 4 Η μέθοδος Βελτιστοποίησης σμήνους σωματιδίων

- 1 Μάθηση συναρτήσεων από δεδομένα
- 2 Χρησιμοποιείται σε πολλά πεδία (ιατρική, προβλέψεις μετοχών κτλ).
- 3 Συνήθως η συνάρτηση προς αναζήτηση δεν υπάρχει και δημιουργείται προσεγγιστική

Γράφημα



- 1 Παρεμβολή με πολυώνυμα
- 2 Νευρωνικά δίκτυα
- 3 RBF δίκτυα

Μάθηση με Γραμματική εξέλιξη

- 1 Χρήση γραμματικής που περιγράφει συναρτήσεις γενικά
- 2 Θα πρέπει να υπάρχει μεγάλη σειρά από συναρτήσεις στις διαθέσιμες για να μπορούμε να μάθουμε την συνάρτηση
- 3 Δημιουργούνται δοκιμαστικές συναρτήσεις
- 4 Το μέσο τετραγωνικό σφάλμα είναι η καταλληλότητα του Γενετικού αλγόριθμου.

- 1 Mean squared error.

$$E_{MSE} = \frac{1}{M} \sum_{j=1}^M (y_j - t_j)^2$$

- 2 Mean absolute error.

$$E_{ABS} = \frac{1}{M} \sum_{j=1}^M |y_j - t_j|$$

- 3 Maximum squared error.

$$E_{MAX} = \max_j (y_j - t_j)^2$$

Γραμματική

$\langle S \rangle ::= \langle \text{expr} \rangle$

$\langle \text{expr} \rangle ::= (\langle \text{expr} \rangle)$
| $\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$
| $\langle \text{func} \rangle (\langle \text{expr} \rangle)$
| $\langle \text{terminal} \rangle$

$\langle \text{op} \rangle ::= + \mid - \mid * \mid / \mid ^$

$\langle \text{func} \rangle ::= \sin \mid \cos \mid \exp \mid \log \mid \text{abs} \mid \text{sqrt}$

- Για το πρόβλημα των επιδόσεων σε αγώνα μπάσκετ (BK)

$$f(x) = \sin(8.59) \frac{\cos(\sqrt{\exp(\sin(x_1))})}{\sqrt{\cos\left(\frac{x_3}{33.1 - 0.245 \cos(x_3 x_2)}\right)}}$$

- Για την συνάρτηση $h(x) = x \sin(x^2)$

$$f(x) = (\sin(x * \text{abs}(1.0)*x) * x)$$

Προβλήματα

- 1 Θα πρέπει οι συναρτήσεις να επαρκούν για την μάθηση του προβλήματος
- 2 Αργή διαδικασία
- 3 Θα πρέπει να χρησιμοποιούνται προστατευόμενες συναρτήσεις
- 4 Πολύ συχνά προβλήματα υπερχείλισης δεκαδικών αριθμών
- 5 Πολλές φορές ο γενετικός εγκλωβίζεται στο να μάθει σειρές από αριθμητικά ψηφία. Λύση: αριθμοί με περιορισμένο πλήθος δεκαδικών τιμών
- 6 Ο γενετικός μπορεί να παράξει σειρά από παρενθέσεις.

Το πρόβλημα

- 1 Σε πολλά προβλήματα τα χαρακτηριστικά είναι πολλά σε σχέση με τα διαθέσιμα πρότυπα
- 2 Κατάρα της διάστασης
- 3 Πολλά χαρακτηριστικά είναι προβληματικά λόγω θορύβου
- 4 Πολλά χαρακτηριστικά δεν παίζουν ρόλο στο σύστημα αναγνώρισης προτύπων
- 5 Απαιτείται μείωση χαρακτηριστικών
- 6 Πολλές φορές χρειάζεται να βρούμε και τις κρυμμένες συσχετίσεις στα χαρακτηριστικά

- 1 Αλγόριθμος PCA
- 2 Multi dimensional scaling
- 3 ISOMAP

Χρήση Γραμματικής εξέλιξης

- 1 Δημιουργία γραμματικής BNF, όπως και στην μάθηση συναρτήσεων
- 2 Τίθεται ένας ζητούμενος αριθμός νέων χαρακτηριστικών, έστω K (πχ $K=2$)
- 3 Κάθε χρωμόσωμα είναι κανόνες για την δημιουργία των νέων χαρακτηριστικών
- 4 Το αρχικό σύνολο δεδομένων μεταμορφώνεται στο καινούριο
- 5 Χρησιμοποιείται ένα μοντέλο μάθησης σαν αξιολογητής
- 6 Το σφάλμα του μοντέλου μάθησης είναι η καταλληλότητα του χρωμοσώματος.

Αποτελέσματα (μάθηση κατηγοριών)

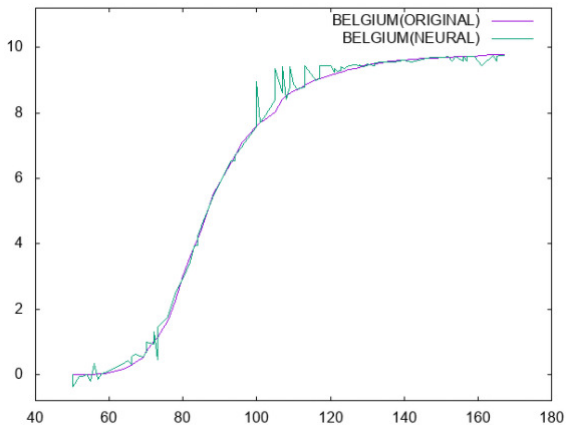
DATASET	MLP GEN	FC MLP	GAIN
BALANCE	8.23%	0.30%	96.35%
DERMATOLOGY	10.01%	4.98%	50.25%
GLASS	58.03%	45.84%	21.00%
HAYES ROTH	35.26%	23.26%	34.03%
HEART	25.46%	17.71%	30.44%
IONOSPHERE	13.67%	8.42%	38.41%
PARKINSONS	17.47%	10.10%	36.46%
PIMA	32.98%	23.76%	27.96%
POPFAILURES	7.66%	4.66%	39.16%
SPIRAL	45.71%	26.53%	41.96%
WINE	20.82%	7.31%	64.89%
WDBC	6.32%	3.47%	45.09%
Z_F_S	9.42%	5.52%	41.40%
Z_O_N_F_S	60.38%	31.20%	48.33%
ZO_NF_S	8.06%	4.00%	50.37%

Αποτελέσματα (μάθηση συναρτήσεων)

DATASET	MLP GEN	FC MLP	GAIN
BK	0.21	0.03	85.71%
BL	0.84	0.005	99.40%
Housing	30.05	10.77	64.16%
Laser	0.003	0.002	33.33%
NT	1.11	0.01	99.10%
Quake	0.07	0.03	57.14%
FA	0.04	0.01	75.00%
PY	0.21	0.02	90.47%

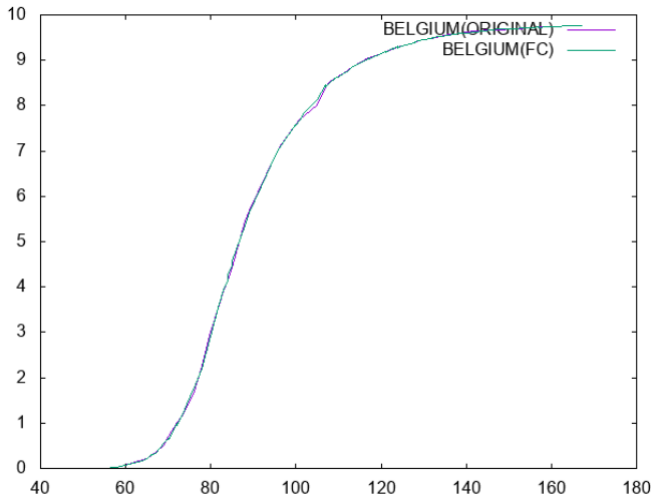
- 1 Πόσα χαρακτηριστικά θα φτιαχτούν (στην πράξη $K = 2, K = 3$ είναι καλή επιλογή)
- 2 Ποιος αξιολογητής θα χρησιμοποιηθεί για τα χαρακτηριστικά \Rightarrow χρειάζεται ένα ελαφρύ μοντέλο όπως το RBF
- 3 Πως τερματίζει ο αλγόριθμος;
- 4 Υπάρχει και εδώ ο κίνδυνος δημιουργίας μεγάλων αριθμητικών σταθερών

Το απλό ΤΝΔ



Πρόβλεψη σε Βέλγιο

Κατασκευή χαρακτηριστικών



Το πρόβλημα

- 1 Πολλές φορές χρησιμοποιούνται εξελικτικές τεχνικές για την μάθηση βαρών στα ΤΝΔ
- 2 Αυτοί οι αλγόριθμοι επιτυγχάνουν πολύ καλά αποτελέσματα
- 3 Το πρόβλημα είναι ότι πολλές φορές δεν έχουν καλές γενικευτικές ικανότητες.
- 4 Χρειάζεται να γίνει μάθηση και της δομής του ΤΝΔ ώστε να είναι αποτελεσματική η μάθηση

- ① Μάθηση βαρών
- ② Εύρεση με γενετικό αλγόριθμο του πλήθους των επιθυμητών κόμβων
- ③ Δημιουργία τοπολογίας με Γενετικό αλγόριθμο
- ④ Γενικά οι περισσότερες υλοποιήσεις αντιμετωπίζουν μόνο μέρος του προβλήματος.

Κατασκευή με Γραμματική Εξέλιξη

- 1 Δημιουργία κατάλληλης γραμματικής
- 2 Κάθε χρωμόσωμα αναπαριστά τους κανόνες για να φτιαχτεί ένα ΤΝΔ
- 3 Το ΤΝΔ δοκιμάζεται στα δεδομένα εκπαίδευσης
- 4 Το μέσο τετραγωνικό σφάλμα είναι η καταλληλότητα του χρωμοσώματος

```

S:=<sigexpr> (0)
<sigexpr>::=<Node> (0)
      | <Node> + <sigexpr> (1)
<Node>::=<number>*sig(<sum>+<number>) (0)
<sum>::= <number>*<xxlist> (0)
      | <sum>+<sum> (1)
<xxlist>::= x1 (0)
      | x2 (1)
      .....
      | xn (n-1)
<number>::= (<digitlist>.<digitlist>) (0)
      | (-<digitlist>.<digitlist>) (1)
<digitlist>::= <digit> (0)
      | <digit><digitlist> (1)
<digit>::= 0 (0)
      | 1 (1)
      .....
      | 9 (9)

```

Παράδειγματα χρήσης

- 1 Μάθηση συναρτήσεων
- 2 Μάθηση δεσμών μεταξύ ατόμων
- 3 Σε Εγκεφαλογραφήματα
- 4 Σε επίλυση διαφορικών εξισώσεων
- 5 Σε δεδομένων από ασθενείς με Πάρκινσον

Πειραματικά αποτελέσματα

PROBLEM	RPROP	BFGS	GENETIC	NNC
Wine	59.93%	48.86%	14.26%	4.44%
Glass	71.37%	53.94%	49.41%	48.01%
Pima	33.19%	36.56%	32.16%	25.00%
Wdbc	32.71%	20.91%	6.87%	4.56%
Circular	44.94%	7.95%	7.84%	6.20%
Spiral	49.06%	45.31%	43.58%	44.90%
Liverdisorder	43.56%	38.86%	35.84%	30.06%
Ionosphere	15.78%	17.08%	16.27%	9.66%



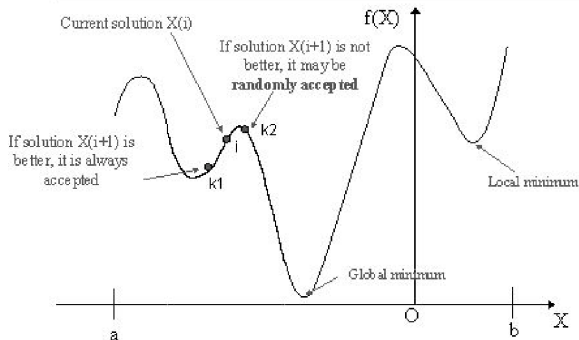
Ορισμοί

- Η μέθοδος Simulated Annealing βασίζεται στην πειραματική διαδικασία της προσομοιούμενης ανόπτησης
- Χρησιμοποιείται σε κάθε πρόβλημα ελαχιστοποίησης, πχ μείωση τους σφάλματος ενός ΤΝΔ
- Κεντρικό ρόλο στον αλγόριθμο παίζει η θερμοκρασία T
- Ενδιάμεσα ο αλγόριθμος μπορεί να δεχτεί και λύσεις που δεν βελτιώνουν την τρέχουσα κατάσταση, υπό κάποια πιθανότητα
- Στην αρχή ο αλγόριθμος κάνει μεγάλα βήματα και στην συνέχεια αυτά μειώνονται όσο περνάει ο χρόνος

- 1 Αρχικοποίηση $k = 0$, $T_0 > 0$, Λήψη δείγματος S_0
- 2 Λήψη δείγματος y
- 3 Θέτουμε $x_{k+1} = y$ με πιθανότητα $P(y) \Rightarrow$ κριτήριο Μετρόπολις
- 4 $P(y) = \min \left(1, \exp \left(-\frac{f(y) - f(x_k)}{T_k} \right) \right)$
- 5 $k = k + 1$
- 6 Μείνουμε την θερμοκρασία με κάποιο προκαθορισμένο ρυθμό πχ. $T_{k+1} = aT_k$, όπου $a < 1$
- 7 Αν ισχύει το κριτήριο τερματισμού, τερματισμός
- 8 Μετάβαση στο βήμα 2.
- 9 Παράδειγμα για τον πλανόδιο πωλητή στο URL:
<https://imgur.com/XAY0tKu>

Σχήμα

Principle of simulated annealing



Τεχνικές μείωσης θερμοκρασίας

- 1 Θεωρούμε πως ξεκινάμε πάντα από μια μεγάλη θερμοκρασία T_0
- 2 k είναι η επανάληψη του αλγορίθμου.
- 3 Εκθετική μείωση:

$$T_k = T_0 a^k, \quad 0.8 \leq a \leq 0.9$$

- 4 Logarithmical multiplicative cooling:

$$T_k = \frac{T_0}{1 + a \log(1 + k)}$$

- 5 Linear multiplicative cooling:

$$T_k = \frac{T_0}{1 + ak}$$

- 6 Quadratic multiplicative cooling:

$$T_k = \frac{T_0}{1 + ak^2}$$

Γενικά

- 1 Αγγλικός όρος: Differential Evolution
- 2 Χρήση σε ελαχιστοποίηση συναρτήσεων
- 3 Μοιάζει αρκετά με τους Γενετικούς Αλγόριθμους
- 4 Είναι πιο απλοί στην εκτέλεση
- 5 Αποκλειστικά για προβλήματα δεκαδικής βελτιστοποίησης

- 1 Κάθε μέλος του πληθυσμού ονομάζεται πράκτορας (agent)
- 2 Κάθε πράκτορας έχει μήκος n (διάσταση του προβλήματος)
- 3 Μέγεθος πληθυσμού NP , με $NP \geq 4$.
- 4 Πιθανότητα διασταύρωσης CR , $CR \in [0, 1]$
- 5 Differential weight F , $F \in [0, 2]$.

Αλγόριθμος

- 1 **Αρχικοποίηση** όλων των πρακτόρων, **Θέσε** $k = 0$
- 2 **Για** κάθε πράκτορα x_i , $i = 1, \dots, NP$ **κάνε**
 - 1 **Επιλογή** τριών πρακτόρων a, b, c (διαφορετικών) από το σύνολο των πρακτόρων.
 - 2 **Επιλογή** ενός τυχαίου ακεραίου $R \in [1, n]$, n είναι διάσταση του προβλήματος.
 - 3 **Υπολογισμός** του νέου πράκτορα $y = (y_1, y_2, \dots, y_n)$ ως εξής:
 - 1 Επιλογή τυχαίου αριθμού $r_i \in (0, 1)$, $i = 1, \dots, n$
 - 2 Αν $r_j < CR$ ή $j = R$, $y_j = a_j + F \times (b_j - c_j)$, αλλιώς $y_j = x_{i,j}$
 - 4 **Αν** $f(y) \leq f(x_i)$ **Θέσε** $x_i = y$
- 3 **Τέλος Επανάληψης**
- 4 **Θέσε** $k = k + 1$
- 5 **Αν** ισχύει το κριτήριο τερματισμού **τερματισμός, αλλιώς** μετάβαση στο βήμα 2.

- Έχουν προταθεί αρκετές τροποποιήσεις από ερευνητές όπως Ali, Storey, Tsoulos et al.
- Σε κάποιες τροποποιήσεις απαιτείται η χρήση του tournament selection:
 - Επιλογή N ατόμων από ένα πληθυσμό με τυχαίο τρόπο
 - Διατήρηση του ατόμου που έχει την καλύτερη συναρτησιακή τιμή στον υποπληθυσμό των N ατόμων

Ο αλγόριθμος DERL

- 1 Αρχικοποίηση όλων των πρακτόρων.
- 2 Θέσε $k = 0$
- 3 Αν $|f_{max} - f_{min}| \leq \epsilon$, τερματισμός.
- 4 Για κάθε πράκτορα x_i κάνε
 - 1 Επιλογή με tournament selection του σημείου x_{tb}
 - 2 Τυχαία επιλογή δύο πρακτόρων a, b
 - 3 $\hat{x}_i = x_{tb} + F \times (a - b)$ (μετάλλαξη)
 - 4 **Υπολογισμός** του νέου πράκτορα $y = (y_1, y_2, \dots, y_n)$ ως εξής:
 - 1 Επιλογή τυχαίου αριθμού $r_j \in (0, 1)$, $j = 1, \dots, n$
 - 2 Αν $r_j < CR$ ή $j = R$, $y_j = x_{i,j} + F \times (x_{i,j} - \hat{x}_{i,j})$, αλλιώς $y_j = x_{i,j}$
 - 5 Αν $f(y) \leq f(x_i)$ Θέσε $x_i = y$
- 5 Τέλος - Επανάληψης
- 6 Θέσε $k = k + 1$
- 7 Μετάβαση στο βήμα 3.

Άλλες προτάσεις για Differential Weight

- Στον αλγόριθμο DERL μπορεί να είναι μεταβαλλόμενος ώστε να μην παράγονται σημεία με την μετάλλαξη που να είναι εκτός ορίων της συνάρτησης.
- Προτεινόμενος μηχανισμός από ALI:

$$F = \begin{cases} \max \left(l_{\min}, 1 - \left| \frac{f_{\max}}{f_{\min}} \right| \right) & , \quad \text{if} \quad \left| \frac{f_{\max}}{f_{\min}} \right| \leq 1 \\ \max \left(l_{\min}, 1 - \left| \frac{f_{\min}}{f_{\max}} \right| \right) & , \quad \text{otherwise} \end{cases} \quad (1)$$

Η μέθοδος GENDE

- Προτάθηκε ο μηχανισμός για το differential weight:

$$F = -\frac{1}{2} + 2 \times R \quad (2)$$

- Προτάθηκε ο εξής κανόνας τερματισμού:
 - Υπολογισμός της ποσότητας

$$\delta^{(t)} = \left| \sum_{i=1}^{N\Pi} |f_i^{(t)}| - \sum_{i=1}^{N\Pi} |f_i^{(t-1)}| \right| \quad (3)$$

- Τερματισμός αν $\delta^{(t)} \leq \epsilon$ για μια σειρά από M επαναλήψεις
πχ $M = 20$

Γενικά

- 1 Αγγλικός όρος: PSO
- 2 Βασίζεται και αυτός σε παρατήρηση της φύσης
- 3 Προσπαθεί να προσομοιώσει την κίνηση ενός σμήνους προς αναζήτηση τροφής
- 4 Εδώ τα χρωμοσώματα ονομάζονται σωματίια
- 5 Κάθε σωματίιο έχει τρέχουσα θέση και ταχύτητα
- 6 Χρησιμοποιείται κυρίως σε προβλήματα δεκαδικής ελαχιστοποίησης
- 7 Προτάθηκε το 1995 από τους Kennedy, Eberhart.

Μεταβλητές

- 1 S τα συνολικά σωματρία
- 2 $f(x)$, η καταλληλότητα κάθε σωματίου x
- 3 x_i κάθε σωματριο του πληθυσμού
- 4 u_i η ταχύτητα κάθε σωματίου
- 5 y_i η καλύτερη θέση κάθε σωματίου στον πληθυσμό
- 6 x^b το συνολικά καλύτερο σωματριο στον πληθυσμό

- 1 Για κάθε σωματίο $i = 1, \dots, S$
 - 1 Αρχικοποίηση της θέσης x_i κάθε σωματίου
 - 2 Αρχικοποίηση της ταχύτητας u_i κάθε σωματίου
 - 3 $x^b = \operatorname{argmin}(f(x))$, $y_i = x_i$
- 2 Μέχρι να ισχύσει ένα κριτήριο τερματισμού
- 3 Για κάθε σωματίο $i = 1, \dots, S$
 - 1 Για κάθε διάσταση $d = 1, \dots, n$
 - 1 Ενημέρωση της ταχύτητας
$$u_{i,d} = \omega u_{i,d} + r_1 (x_{i,d} - y_{i,d}) + r_2 (x_{i,d} - x_d^b)$$
 - 2 Ενημέρωση θέσης $x_{i,d} = x_{i,d} + u_{i,d}$
 - 2 Αν $f(x_i) < f(y_i)$, $y_i = x_i$
 - 3 Αν $f(x_i) < f(x^b)$, $x^b = x_i$

Inertia (ω)

- 1 Είναι ποσότητα που μπορεί να είναι σταθερή ή να μεταβάλλεται
- 2 Συνήθως $\omega < 1$
- 3 Ελέγχει ποιος είναι ο ρόλος της προηγούμενης ταχύτητας στην δημιουργία της καινούριας
- 4 Παράδειγμα μεταβολής σε κάθε γενιά k :

$$\omega = \omega_{MAX} - \frac{k}{k_{MAX}} (\omega_{MAX} - \omega_{MIN})$$

- 1 Προκαλείται όταν η ταχύτητα παίρνει πολύ μεγάλες ή πολύ μικρές τιμές, και το σωματίο βγαίνει εκτός ορίων
- 2 Λύση: όρια στις αλλαγές της ταχύτητας
- 3 Λύση: απόρριψη αλλαγών στην ταχύτητα που θα οδηγήσουν σε έκρηξη
- 4 Λύση: κανονικοποίηση τιμών της ταχύτητας σε συγκεκριμένο διάστημα

Random Inertia



$$\omega_{\text{iter}} = 0.5 + \frac{r}{2} \quad (4)$$

όπου r τυχαίος αριθμός και $r \in [0, 1]$.

Κανόνας τερματισμού ALI

- Σε κάθε επανάληψη υπολογισμός των ποσοτήτων f_{max} και f_{min}
- Τερματισμός αν $|f_{max} - f_{min}| \leq \epsilon$
- Δύσκολο να ικανοποιηθεί καθώς δεν συγκλίνουν πάντοτε μεταξύ τους αυτές οι τιμές.

Κανόνες τερματισμού DoubleBox

- Η διακύμανση $\sigma^{(iter)}$ της ποσότητας f_{min} υπολογίζεται σε κάθε επανάληψη iter.
 - Αν ο αλγόριθμος δεν μπορεί να εντοπίσει μια νέα καλύτερη τιμή για την f_{min} για μια σειρά από επαναλήψεις, τότε η μέθοδος θα πρέπει να τερματιστεί.
 - Τερματισμός αν:





$$\sigma^{(iter)} \leq \frac{\sigma^{(iter_{last})}}{2} \quad (5)$$

όπου $iter_{last}$ είναι η τελευταία επανάληψης στην οποία βρέθηκε μια νέα καλύτερη τιμή για το f_{min}




Σύνοψη

- Παρουσιάστηκαν ειδικά θέματα Γραμματικής Εξέλιξης
- Παρουσιάστηκαν μερικοί εξελικτικοί αλγόριθμοι.

Βιβλιογραφία I

-  ITsoulos, I.G., Lagaris, I.E. Solving differential equations with genetic programming. Genet Program Evolvable Mach 7, 33–54 (2006). <https://doi.org/10.1007/s10710-006-7009-y>
-  I. G. Tsoulos, D. Gavrilis and E. Dermatas, GDF: A tool for function estimation through grammatical evolution, Computer Physics Communications 174, pp. 555-559, 2006.
-  I.G. Tsoulos, D. Gavrilis, E. Glavas, “Neural Network Construction and Training using Grammatical Evolution”, Neurocomputing 72, pp. 269-277, 2008.
-  D. Gavrilis, I.G. Tsoulos, E. Dermatas, “Features Selection and Construction using Grammatical Evolution”, Pattern Recognition Letters 29, pp. 1358-1365, 2008.

Βιβλιογραφία II

-  I.G. Tsoulos, D. Gavrilis, E. Glavas, Solving differential equations with constructed neural networks, *Neuromputing* 72, pp. 2385-2391, 2009.
-  Ioannis G. Tsoulos, Creating classification rules using Grammatical Evolution, accepted for publication in *International Journal of Computational Intelligence Studies*, 2018
-  Ioannis G. Tsoulos, Alexandros Tzallas, Dimitrios Tsalikakis, NNC: A tool based on Grammatical Evolution for data classification and differential equation solving, *SoftwareX* 10, 2019.