Επίλυση Προβλημάτων Μεικτού Μη Γραμμικού και Ακέραιου Προγραμματισμού με Χρήση Γενετικών Αλγόριθμων

Α. Νικολακόπουλος, Α. Αλεξανδρίδης. Α. Σπυρόπουλος, Χ Σαρίμβεης (Ε.Υ.)

Σχολή Χημικών Μηχανικών ΕΜΠ

Σκοπός εργασίας – Κίνητρο

Μία μεγάλη γκάμα προβλημάτων της βιομηχανικής πρακτικής προτυποποιούνται με μαθηματικά μοντέλα Μικτού Ακέραιου Μη Γραμμικού Προγραμματισμού. Σε πολλές περιπτώσεις τα προβλήματα μαθηματικής βελτιστοποίησης που διαμορφώνονται είναι μη κυρτά (non convex) και η επίλυσή τους με ντετερμινιστικές μεθόδους δεν αποδίδει ικανοποιητικά αποτελέσματα. Μεγάλο μέρος της ερευνητικής δραστηριότητας έχει στραφεί λοιπόν στην εφαρμογή στοχαστικών μεθόδων όπως είναι οι Γενετικοί Αλγόριθμοι.

Μαθηματικά μοντέλα προβλημάτων

Μοντέλα μεικτού ακέραιου και μη γραμμικού προγραμματισμού,που περιέχουν ανισωτικούς ή/και εξισωτικούς περιορισμούς (MINLP)

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x},\mathbf{z}) \quad h_m(\mathbf{x},\mathbf{z})=0,\ l=1,\dots,M \quad g_k(\mathbf{x},\mathbf{z})\ge 0,\ k=1,\dots,K$$

όπου **x** είναι το διάνυσμα των συνεχών μεταβλητών και **z** είναι το διάνυσμα των ακέραιων μεταβλητών. Τουλάχιστον μια από τις συναρτήσεις *f*, *g*, *h* είναι μη γραμμική.

Μοντέλα μη γραμμικού προγραμματισμού,που περιέχουν ανισωτικούς ή/και εξισωτικούς περιορισμούς (NLP)

$$\min_{\mathbf{x}} f(\mathbf{x}),\quad h_m(\mathbf{x})=0,\ l=1,\dots,M,\quad g_k(\mathbf{x})\ge 0,\ k=1,\dots,K$$

όπου **x** είναι το διάνυσμα των συνεχών μεταβλητών. Τουλάχιστον μια από τις συναρτήσεις *f*, *g*, *h* είναι μη γραμμική.

Επίλυση MINLP: Αλγόριθμος 1

Προκαταρκτικά βήματα:

1. Δημιουργία διανύσματος κατωφλίου. Επιλέγονται τυχαία $N+1$ διανύσματα τιμών **z** ($\mathbf{z}_i$ $i=1,2,\dots,N+1$) και επιλύονται τα $N+1$ αντίστοιχα NLP προβλήματα. Έστω f_i $i=1,2,\dots,N+1$ οι αντίστοιχες βέλτιστες τιμές της αντικειμενικής συνάρτησης. Το $\mathbf{z}_o$ αντιστοιχεί στην ελάχιστη τιμή της αντικειμενικής συνάρτησης f_o και για τα υπόλοιπα υπολογίζουμε την ποσότητα: $a_i = \frac{f_i - f_o}{f_o} \ge 0$

Δημιουργούμε το διάνυσμα κατωφλίου **a** που αποτελείται από τα *N* διαφορετικά στοιχεία a_i διατεταγμένα κατ' αύξουσα σειρά. Τα διανύσματα $\mathbf{z}_i$ $i=1,2,\dots,N+1$ αποτελούν τις πρώτες γραμμές του πίνακα απαγορευμένης έρευνας **TL.**

2. Επιλέγουμε τις παραμέτρους τερματισμού *w* και *Imin*. Η σημασία και η χρήση αυτών των παραμέτρων θα εξηγηθεί στη συνέχεια.

Η κύρια επαναληπτική διαδικασία είναι η εξής:

1. Θέτουμε $\mathbf{z}=\mathbf{z}_o$ ως τρέχουσα λύση με τιμή αντικειμενικής συνάρτησης $f=f_o$. Θέτουμε αύξοντα αριθμό επανάληψης $I=1$ και αριθμό βελτιωμένων λύσεων $IS=0$.

2. Μεταλλάσσουμε το διάνυσμα **z** σε **z**΄.

3. Αν **z**΄ ανήκει στον πίνακα **TL** επιστρέφουμε στο βήμα 2.

4. Υπολογίζουμε την τιμή της αντικειμενικής συνάρτησης *f* ΄ που αντιστοιχεί στο διάνυσμα **z**΄. Εισάγουμε την **z**΄ στο **TL** και αυξάνουμε τον αριθμό των επαναλήψεων κατά 1, $I = I+1$.

5. Αν $I>=Imin$ τότε υπολογίζουμε το λόγο *IS*/*I* και τον συγκρίνουμε με την παράμετρο τερματισμού w. Αν *IS*/I <= *w* ο αλγόριθμος τερματίζεται και το αποτέλεσμα που προκύπτει είναι το διάνυσμα **z** που αντιστοιχεί στην ελάχιστη τιμή της αντικειμενικής συνάρτησης καθ'όλη τη διάρκεια της διαδικασίας.

6. Υπολογίζουμε το λόγο $T = \frac{f' - f}{f'}$ (3)
Αν $T >\max(\mathbf{a})$ επιστρέφουμε στο βήμα 2. Αν $0<=T<=\max(\mathbf{a})$ τότε το *T* αντικαθιστά το στοιχείο με τη μέγιστη τιμή στο διάνυσμα κατωφλίου **a**, του οποίου τα στοιχεία ταξινομούνται κατά αύξουσα σειρά. Αν T<0 τότε IS = IS + 1.

7. Θέτουμε **z = z**΄ **ως** τρέχουσα λύση με τιμή αντικειμενικής συνάρτησης $f=f'$. Επιστρέφουμε στο βήμα 2.

Επίλυση NLP: Αλγόριθμος 2

Προκαταρκτικά βήματα

Οι περιορισμοί συγχωνεύονται με την αντικειμενική συνάρτηση :

$$F(x,a,b,p,q) = f(x) + \sum_{m=1}^{M} a_m\left[(h_m(x)+p_m)^2 - p_m^2\right] + \sum_{k=1}^{K} b_k\left[\langle g_k(x)+q_k\rangle_+^2 - q_k^2\right]$$

όπου τα **a**,**b** είναι πολλαπλασιαστές ποινής και τα **p**,**q** είναι πολλαπλασιαστές Lagrange

B1- Θέτουμε αρχικές τιμές για τα **x**, **a**,**b**,**p** και **q**

B2- Με Εξελικτική Στρατηγική επιλύουμε το πρόβλημα της βελτιστοποίησης για συγκεκριμένο αριθμό επαναλήψεων, με σταθερά τα **a**,**b**,**p** και **q**.

B3- Αν συμπληρωθεί ο μέγιστος αριθμός επαναλήψεων ο αλγόριθμος τερματίζεται. Διαφορετικά αναπροσαρμόζουμε τις τιμές των **a**,**b**,**p** και **q** και επιστρέφουμε στο βήμα B2.

Εξελικτική Στρατηγική

Θέτουμε *iter* = 0

B1- Παράγουμε πληθυσμό με L τυχαίες λύσεις $\mathbf{x}_i$, $i = 1,\dots,L$

B2- $iter = iter + 1$. Αν $iter = maxiter$ ο αλγόριθμος τερματίζεται.

B3- Υπολογίζουμε την $f(x_i)$, $i = 1,\dots,L$

B4- Κατατάσσουμε τις λύσεις κατά φθίνουσα f(x): $x_1, \mathbf{x}_2, x_3,\dots,\mathbf{x}_L$ όπου το $\mathbf{x}_i$ προηγείται του $\mathbf{x}_j$ αν $f(\mathbf{x}_i) > f(\mathbf{x}_j)$

B5- Εφαρμόζουμε τελεστή διασταύρωσης στις *L*-1 πρώτες λύσεις $\mathbf{x}_{j,new} = x_j + r(x_{j+1} - x_j)$ *r* τυχαίος αριθμός στο [0,1]

Αν $f(\mathbf{x}_{j,new}) < f(x_{j+1})$ τότε $\mathbf{x}_j = \mathbf{x}_{j,new}$

B6- Κατατάσσουμε τις λύσεις όπως στο B4.

B7- Εφαρμόζουμε ανομοιόμορφη μετάλλαξη στις λύσεις $i = 1,\dots,L$ με πιθανότητα :
$p_{m,I} = (L - i + 1)/L$

B8- **Αντικαθιστούμε** την χειρότερη με την καλύτερη λύση, Επιστρέφουμε στο **B2**

Ρουτίνα αναπροσαρμογής των a,*b*,*p* και *q*

Θέτουμε *jter* = 0. Επιλέγουμε τιμές για τις παραμέτρους w_1, w_2

B1- $jter = jter + 1$. Αν $jter = mxjter$ ο αλγόριθμος **τερματίζεται**

B2- Με την **Εξελικτική Στρατηγική** επιλύουμε το πρόβλημα βελτιστοποίησης της εξίσωσης $F(\mathbf{x},\mathbf{a},\mathbf{b},\mathbf{p},\mathbf{q})$.

B3- Προσδιορίζουμε **την** μέγιστη παραβίαση των περιορισμών : $cv_{max} = \max\left(\max_m |h_m|, \max_k \langle g_k \rangle_+\right)$

Και τους περιορισμούς που δεν βελτιώνεται η παραβίασή τους από τον παράγοντα w_1:

$S_E = \{l : |h_m| > cv / w_1\}$ και $S_I = \{k : \langle g_k \rangle_+ > cv / w_1\}$

B4- Αν $cv_{max} >= cv$: $a_m = a_m w_2$, $p_m = p_m/w_2$ για όλα τα *m* του S_E και $b_k = b_k w_2$, $q_k = q_k/w_2$ για όλα τα *k* του S_I

B5- Αν $cv_{max} =< cv$: $cv = cv_{max}$, $p_m = h_m(\mathbf{x}_b) + p_m$, $q_k = \max\{0, g_k(\mathbf{x}_b) + q_k\}$ $\mathbf{x}_b$ είναι η καλύτερη λύση που έχει βρεθεί

B6- Αν $cv_{max} =< cv/w_1$: $a_m = a_m/w_2$, $p_m = p_m w_2$ για $m = 1,\dots,M$ και $b_k = b_k/w_2$, $q_k = q_k w_2$ για $k = 1,\dots,K$
Επιστρέφουμε στο B1.

Αποτελέσματα στοχαστικού Αλγόριθμου 1 σε MINLP προβλήματα

Ο Αλγόριθμος 1 βρίσκει πάντα το απόλυτο βέλτιστο και σε 100 επαναλήψεις του αλγόριθμου, χρειάστηκαν κατά μέσο όρο 400 επαναλήψεις που αντιστοιχούν στο 9,7% του συνόλου των πιθανών λύσεων. Ο προγραμματισμός έγινε στο Matlab και ο μέσος υπολογιστικός χρόνος για την επίλυση του προβλήματος σε επεξεργαστή Pentium III 600 Mhz ήταν 35s. Σε μεγαλύτερα προβλήματα η μέθοδος οδηγεί σε πολύ καλή λύση (κοντά στο ολικό βέλτιστο) σε πολύ μικρό υπολογιστικό χρόνο.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Ο Αλγόριθμος 1 χρησιμοποιείται για την επιλογή των τιμών των ακέραιων μεταβλητών σε προβλήματα Μικτού Ακέραιου και μη Γραμμικού Προγραμματισμού με σκοπό την ελαχιστοποίηση των συνδυασμών που πρέπει να ερευνηθούν και επομένως του απαραίτητου υπολογιστικού χρόνου.

Αποτελέσματα στοχαστικού Αλγόριθμου 2 σε NLP προβλήματα

	Μέγιστος αριθμός επαναλήψεων	150	200	250	300
Ex 1	Μεγαλύτερο	38.8503	38.8503	38.8503	38.8503
	Μικρότερο	38.1995	38.3199	38.4376	38.5873
	Μέση τιμή	38.7447	38.7847	38.8167	38.8185
	Count	38	56	70	83
	STD	0.13964	0.10821	0.0692	0.07369
Ex 2	Μεγαλύτερο	-176.46	-181.25	-182.23	-180.21
	Μικρότερο	-186.73	-186.73	-186.73	-186.73
	Μέση τιμή	-186.00	-186.46	-186.45	-186.56
	Count	65	76	81	89
	STD	1.6099	0.8248	0.7102	0.6829

Ex.#		Ακριβής λύση	Αλγόριθμος 2	HHE-MUM-APP	TPEP	Hybrid EP	EP alone
	[illegible]	[illegible]	0.25000	0.25000	0.25000	0.83636	0.24510
3	[illegible]	[illegible]	0.0	0.0	0.0	Diverged	0.0049
	N	-	2250*	796*	4658	7667	5014
	fmin	-5.50801	-5.50801	-5.50801	-5.50801	-5.42623	-5.51096
4	*Er*	0	0.0		0.0	0.08178	diverged
	N	-	41400*	52615*	3659	8081	4392
	fmin	-6961.81388	-6961.81388	-6961.81388	-6961.81388	[illegible]	1033.13591
5	*Er*	0	0.0		0.0	Diverged	[illegible]
	N	-	78450*	223455*	21739	Diverged	14049
	fmin	5.00000	5.00000	5.00000	[illegible]	[illegible]	4.99800
6	*Er*	0		0.0	0.0	Diverged	[illegible]
	N	-	31200*	32343*	19139	13130	[illegible]
	fmin	1.00000	1.00000	1.00000	1.00000	[illegible]	[illegible]
7	*Er*	0	0.0	0.0	0.0	[illegible]	[illegible]
	N	-	48150*	123191*	3730	Diverged	4251
	fmin	0.05400	0.05400	0.05395	0.05400	Diverged	0.43321
8	*Er*	0				Diverged	diverged
	N	-	125700*	271023*	11210	Diverged	1448

Ex. 9	Αλγόριθμος 2	Static penalties	Dynamic penalties	Genocop II	Superiority of annealing point	Death penalty
Μεγαλύτερο	24.324	36.060	42.358	44.302	48.866	32.477
Μικρότερο	24.306	24.690	25.486	18.917	17.388	25.653
Μέση τιμή	24.310	29.258	26.905	24.418	22.932	27.116
		Παραβίαση περιορισμών		Παραβίαση περιορισμών	Παραβίαση περιορισμών	

		Ex. 10	Ex. 11	Ex. 12
Ακριβής λύση		7.16129	6.613086	1600039.0
Αλγόριθμος 2	Largest	7.16129	6.613086	1600039.0
	Smallest	7.15286	6.608900	1599264.9
	Average	7.15974	6.611581	1599828.8
HGA	Largest	7.16085	6.61305	1600039.0
	Smallest	7.14131	6.60297	1598842.2
	Average	7.14926	6.60727	1599749.4

		Ex. 13	Ex.14	Ex.15
Exact Solution		44.0000	-1.80000	40.56241
Proposed Algorithm	Largest	43.99817	-1.80000	40.56241
	Smallest	43.96888	-1.80000	40.27448
	Average	43.99217	-1.80000	40.53915
EHGA	Largest	43.98178	-1.80028	31.8508
	Smallest	35.8908	-2.38214	20.76224
	Average	41.5932	-2.07616	29.18656

Στους παραπάνω πίνακες φαίνονται τα αποτελέσματα του Αλγόριθμου 2 σε σύγκριση με αποτελέσματα άλλων αλγόριθμων της βιβλιογραφίας για όλα τα παραδείγματα προβλημάτων που παρουσιάζονται αριστερά. Όπως φαίνεται ο αλγόριθμος 2 βρίσκει τη βέλτιστη τιμή στα περισσότερα προβλήματα ενώ πολλοί από τους άλλους αλγόριθμους δεν εντοπίζουν τις βέλτιστες λύσεις ή ακόμα παραβιάζουν τους περιορισμούς. Επιπλέον τα παραγόμενα αποτελέσματα είναι διαθέσιμα σε λιγότερες, κατά κανόνα, αξιολογήσεις της αντικειμενικής συνάρτησης, άρα σε μικρότερους υπολογιστικούς χρόνους. Οι ποικιλία των προβλημάτων αποδεικνύει ότι ο αλγόριθμος 2 έχει σθεναρή συμπεριφορά και είναι αξιόπιστος.

MINLP προβλήματα

Στα πλαίσια της εργασίας, η προτεινόμενη μέθοδος εφαρμόστηκε στην επίλυση προβλημάτων αριστοποίησης παραγωγικών συστημάτων που αποτελούνται από περισσότερα του ενός στάδια παραγωγής και όπου παράγονται περισσότερα του ενός προϊόντα. Οι ακέραιες μεταβλητές δίνουν τον αριθμό των μονάδων διαλείποντος έργου σε κάθε στάδιο της παραγωγής. Οι υπόλοιπες μεταβλητές του προβλήματος είναι συνεχείς και θετικές. Ο αλγόριθμος εφαρμόστηκε σε πρόβλημα αναφοράς μεσαίας κλίμακας με έξι στάδια επεξεργασίας και με δυνατότητα επιλογής μιας έως τεσσάρων μονάδων για κάθε στάδιο. Το σύνολο των λύσεων αυτού του προβλήματος είναι $4^6 = 4096$.

NLP προβλήματα

Ex1. $\max f(\mathbf{x}) = 21.5 + x_1\sin(4\pi x_1) + x_2\sin(20\pi x_2)$
s.t. $-3.0 \le x_1 \le 12.1$, [illegible]

Ex2. $\min f(\mathbf{x}) = \sum_{i=1}^{5} i\cos[(i+1)x_1+i]\cdot\sum_{i=1}^{5} i\cos[(i+1)x_2+i]$
s.t. $-10.0 \le x_1 \le 10.0$, $-10.0 \le x_2 \le 10.0$

Ex3. [illegible]
s.t. $x_1 + x_2^2 \ge 0$, [illegible] $-0.5 \le x_1 \le 0.5$, $x_2 \le 1.0$

Ex4. [illegible] s.t. $4x_1^4 - 32x_1^3 + 88x_1^2 - 96x_1 - x_2 + 36 \ge 0$, $2x_1^4 - 8x_1^3 + 8x_1^2 - x_2 + 2 \ge 0$, $0 \le x_1 \le 3$, $0 \le x_2 \le 4$

Ex5. $\min f(\mathbf{x}) = (x_1-10)^3 + (x_2-20)^3$
s.t. $(\xi_1-5)^2 + (\xi_2-5)^2 - 100 \ge 0$, $-(x_1-6)^2 - (x_2-5)^2 + 82.81 \ge 0$, $13 \le x_1 \le 100$, $0 \le x_2 \le 100$

Ex6. $\min f(\mathbf{x}) = 0.01x_1^2 + x_2^2$
s.t. $2 \le x_1 \le 50$, $x_1x_2 - 25 \ge 0$, $x_1^2 + x_2^2 - 25 \ge 0$, $0 \le \xi_2 \le 50$

Ex7. $\min f(\mathbf{x}) = (x_1-2)^2 + (x_2-1)^2$
s.t. $-x_1^2 + x_2 \ge 0$, [illegible]

Ex8. $\min f(\mathbf{x}) = \exp(x_1x_2x_3x_4x_5)$
s.t. $x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$, $x_1^3 + x_2^3 + 1 = 0$, $x_2x_3 - 5x_4x_5 = 0$, [illegible]

Ex9. $\min f(\mathbf{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3-10)^2 + 4(x_4-5)^2 + (x_5-3)^2 + 2(x_6-1)^2 + 5x_7^2 + 7(x_8-11)^2 + 2(x_9-10)^2 + (x_{10}-7)^2 + 45$
s.t. $105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \ge 0$, $-3(x_1-2)^2 - 4(x_2-3)^2 - 2x_3^2 + 7x_4 + 120 \ge 0$, [illegible], $-x_1^2 - 2(x_2-2)^2 + 2x_1x_2 - 14x_5 + 6x_6 \ge 0$, [illegible], $-5x_1^2 - 8x_2 - (x_3-6)^2 + 2x_4 + 40 \ge 0$, [illegible], $-0.5(x_1-8)^2 - 2(x_2-4)^2 - 3x_5^2 + x_6 + 30 \ge 0$, $-10.0 \le x_i \le 10.0$, $i = 1,\dots,10$

Ex10. $\max f(\xi) = -2x_1^2 + 2x_1x_2 - 2x_2^2 + 4x_1 + 6x_2$
s.t. $-2x_1^2 + x_2 \ge 0$, $5 - x_1 - 5x_2 \ge 0$, $x_1, x_2 \ge 0$

Ex11. $\max f(\mathbf{x}) = -2x_1^2 + 2x_1x_2 - 2x_2^2 + 4x_1 + 6x_2$
s.t. [illegible], $5 - x_1 - 5x_2 \ge 0$, $x_1, x_2 \ge 0$

Ex12. [illegible]
s.t. $x_1^2 + x_2^2 - 16 \ge 0$, [illegible], $0 \le x_i \le 100, i = 1,2$

Ex13. [illegible]
s.t. [illegible], $-x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \ge 0$, [illegible], $-100 \le x_i \le 100, i = 1,\dots,5$

Ex14. [illegible]
s.t. $x_1^2 + x_2^2 - 5 \ge 0$, [illegible]

Ex15. $\max f(\mathbf{x}) = 4x_1^3 - 3x_1 + 6x_2$
s.t. [illegible], $-x_1 + 4x_2 \ge 0$, $2x_1 + x_2 - 5 = 0$, $0 \le x_i \le 100, i = 1,2$

Επίλυση προβλημάτων μεγάλης κλίμακας

Ο αλγόριθμος 2 μπορεί να εφαρμοστεί για την επίλυση σύνθετων προβλημάτων μεγάλης κλίμακας. Μια τέτοια περίπτωση που παρουσιάζει ιδιαίτερο ενδιαφέρον είναι η επίλυση του προβλήματος βελτιστοποίησης που εμφανίζεται σε συστήματα ρύθμισης προβλεπτικού μοντέλου (Model Predictive Controllers, MPC).

Οι τεχνικές MPC βασίζονται στον παρακάτω συλλογισμό: Αν είναι διαθέσιμο ένα δυναμικό μοντέλο διακριτού χρόνου του συστήματος που συσχετίζει τις μεταβλητές εκ χειρισμού με τις ρυθμιζόμενες μεταβλητές, τότε μπορεί να διαμορφωθεί με βάση αυτό το μοντέλο ένα πρόβλημα βελτιστοποίησης σε πραγματικό χρόνο που να υπολογίζει τις βέλτιστες ρυθμιστικές κινήσεις, ελαχιστοποιώντας μια συνάρτηση κόστους. Από τις ρυθμιστικές αυτές κινήσεις υλοποιείται μόνο η πρώτη και στην επόμενη διακριτή χρονική στιγμή το πρόβλημα σχηματίζεται από την αρχή και επιλύεται ξανά.

Ωστόσο τα προβλήματα βελτιστοποίησης που εμφανίζονται στις τεχνικές MPC είναι πολλές φορές πολύπλοκα, μη γραμμικά (σε περίπτωση που χρησιμοποιούμε μη γραμμικό μοντέλο) και μη κυρτά, γεγονός που κάνει ιδιαίτερα δύσκολη την επίλυσή τους. Στη συγκεκριμένη περίπτωση που αντιμετωπίσαμε, η πολυπλοκότητα του προβλήματος αυξήθηκε, θεωρώντας ότι το πρόβλημα βελτιστοποίησης είναι πολυκριτηριακό.

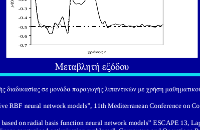
Συνοψίζοντας λοιπόν, ο αλγόριθμος 2 εφαρμόστηκε για την επίλυση ενός πολυκριτηριακού μη γραμμικού μη κυρτού προβλήματος βελτιστοποίησης που διαμορφώνεται εκ νέου σε κάθε χρονική στιγμή. Με τον τρόπο αυτό προέκυψε μια νέα μεθοδολογία MPC που δοκιμάσθηκε με επιτυχία σε συστήματα όπου παρατηρούνται απότομες μεταβολές στη δυναμική συμπεριφορά αλλά και σε συστήματα που περιέχουν πολλαπλές μεταβλητές εισόδου και εξόδου.

Παράδειγμα I: Ρύθμιση συστήματος μιας εισόδου – μιας εξόδου (SISO) με μεταβολή στη δυναμική

Στο παράδειγμα αυτό στόχος είναι η ρύθμιση ενός συστήματος μιας εισόδου - μιας εξόδου, στην περίπτωση που η δυναμική του συστήματος μεταβάλλεται με το χρόνο. Ως μέθοδος ρύθμισης θα χρησιμοποιηθεί μια μεθοδολογία MPC στα πλαίσια της οποίας ο αλγόριθμος 2 καλείται να επιλύσει ένα μη γραμμικό μη κυρτό πρόβλημα βελτιστοποίησης.

Αρχικά δίνουμε μία αλλαγή στην επιθυμητή τιμή της μεταβλητής εξόδου, ενώ στην συνέχεια κατά την χρονική στιγμή 15 μεταβάλλεται η δυναμική του συστήματος και ο αλγόριθμος καλείται να διατηρήσει την έξοδο όσο το δυνατόν κοντά στην επιθυμητή τιμή. Το πρώτο κριτήριο που ικανοποιείται είναι η απαίτηση για διαρκή διέγερση και στην συνέχεια ελαχιστοποιείται η συνάρτηση που αφορά τη μεταβλητή εξόδου.

Μεταβλητή εισόδου



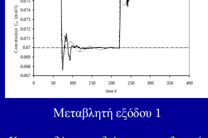
Μεταβλητή εξόδου

ΑΝΑΚΟΙΝΩΣΕΙΣ-ΔΗΜΟΣΙΕΥΣΕΙΣ

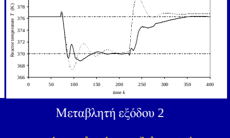
1. Δογάνης Φ, Χ. Σαρίμβεης, Γ. Μπάφας και Δ. Κουφός, «Βελτιστοποίηση της παραγωγικής διαδικασίας σε μονάδα παραγωγής λιπαντικών με χρήση μαθηματικού προγραμματισμού», 4ο Πανελλήνιο συνέδριο Χημικής Μηχανικής, 2003, Πάτρα.
2. Alexandridis A., H. Sarimveis, G. Bafas, "Modeling of continuous digesters using adaptive RBF neural network models", 11th Mediterranean Conference on Control and Automation MED'03, Rhodes, Greece, 2003.
3. Alexandridis A., H. Sarimveis, G. Bafas, "Adaptive control of continuous pulp digesters based on radial basis function neural network models" ESCAPE 13, Lappeenranta, Finland,2003.
4. Sarimveis H. and A. Nikolakopoulos, "A line up evolutionary algorithm for solving nonlinear constrained optimization problems", Computers and Operations Research, 32(6), 2005, 1499-1514.
5. Νικολακόπουλος Α, Χ. Σαρίμβεης και Γ. Μπάφας, «Ανάπτυξη ευρεστικού αλγόριθμου επίλυσης προβλημάτων μικτού ακέραιου και μη γραμμικού προγραμματισμού με εφαρμογή στην αριστοποίηση συστημάτων χημικής μηχανικής», 4ο Πανελλήνιο συνέδριο Χημικής Μηχανικής, 2003, Πάτρα.

Παράδειγμα II: Ρύθμιση συστήματος πολλαπλών εισόδων – πολλαπλών εξόδων (MIMO)

Στο παράδειγμα αυτό έχουμε να κάνουμε με ένα πολυμεταβλητό σύστημα και συγκεκριμένα ένα σύστημα δυο εισόδων – δυο εξόδων. Ο ρυθμιστής εδώ καλείται να οδηγήσει ταυτόχρονα και τις δύο μεταβλητές εξόδου στην αντίστοιχη επιθυμητή τιμή.



Μεταβλητή εξόδου 1



Μεταβλητή εξόδου 2

Και στα δύο παραδείγματα αποδεικνύεται σωστή η επιλογή του εξελικτικού αλγόριθμου ως μεθόδου βελτιστοποίησης καθώς καταφέρνει να εντοπίσει τις τιμές εκείνες των μεταβλητών απόφασης οδηγούν το σύστημα στην επιθυμητή κατάσταση ισορροπίας. Είναι χαρακτηριστικό ότι στα παραπάνω προβλήματα οι κλασικές μεθοδολογίες βελτιστοποίησης που δοκιμάστηκαν δεν παρήγαγαν ικανοποιητικά αποτελέσματα σε αποδεκτό χρόνο.