APLICAȚII ALE ALGORITMILOR GENETICI ÎN TEORIA TRANSMISIUNII INFORMAȚIEI, PRELUCRAREA IMAGINILOR ȘI ÎN ERGONOMIE

Prof. dr. ing. Vasile Buzuloiu,
Prep. ing. Constantin Vertan

Universitatea "Politehnica" București

Rezumat: Deși conceptul de algoritm genetic a fost introdus la mijlocul anilor '70 de Holland, multă vreme acesta a trecut neobservat. Extinderea interesului cercetărilor în domeniu s-a manifestat cu precădere în ultimii ani, impulsionat mai ales de spectrul larg de probleme practice, rezolvate folosind algoritmi genetici. Articolul expune câteva aplicații ale algoritmilor genetici în teoria transmisiunii informației (criptografie cu chei publice), prelucrarea imaginilor și ergonomie (optimizarea unor segmente ale unui "loc de muncă"). Prezentarea se va axa doar pe sublinierea unor aspecte strict specifice ale implementării, fără a insista asupra aspectelor generale legate de algoritmii genetici.

Cuvinte cheie: algoritmi genetici, optimizare stocastică, sisteme criptografice cu chei publice, prelucrarea imaginilor, ergonomie.

1. Introducere

Printre cărțile dedicate algoritmilor genetici, apărute în ultimii ani, cea a lui Zbigniew Michalewicz "Genetic Algorithms + Data Structures = Evolutionary Programming" [1] este plină de citate. Ne vom permite să îl citâm la rândul nostru, măcar pentru introducerea în subiect:

"Există o clasă largă de probleme de interes practic pentru care nu avem algoritmi suficient de rapizi. Multe dintre ele sunt probleme de optimizare ce apar frecvent în aplicații. Adesea este posibil, în schimb, să găsim un algoritm eficient ce ne duce doar la o soluție aproximativă. Pentru unele probleme grele de optimizare putem folosi algoritmi probabilistici - care nu garantează o valoare optimă, dar care, alegând aleator suficient de mulți "martori", fac ca probabilitatea de eroare să fie cât de mică dorim... In general, orice sarcină de îndeplinit poate fi gândită ca rezolvarea unei probleme, iar aceasta ca un proces de căutare într-un spațiu al soluțiilor potențiale. Și fiindcă o căutăm pe cea mai bună, avem de a face cu un proces de optimizare. Uneori sunt suficiente metode clasice de căutare, exhaustive; dacă spațiile de căutare sunt mari, trebuie folosite tehnici speciale, ce se înglobează inteligenței artificiale. Algoritmii genetici sunt printre acestea. Ei sunt algoritmi stocastici ale căror metode de căutare modelează fenomenul natural al moștenirii genetice și al luptei darwiniene pentru supraviețuire."

Un algoritm genetic pentru o problemă particulară trebuie să aibă următoarele cinci componente:

- o reprezentare genetică pentru soluțiile potențiale ale problemei
- o modalitate de a crea o populație inițială de soluții potențiale
- o funcție de evaluare, ce joacă rolul mediului, care apreciază soluțiile în termeni de adecvare
- operatori genetici care alterează compoziția descendenților în timpul reproducerii
- valori pentru diferiți parametri pe care îi folosește algoritmul genetic (dimensiunea populației, probabilitățile de aplicare a operatorilor genetici etc.)

După cum se remarcă, aproape orice problemă poate fi formulată în așa fel încât pentru rezolvare să se utilizeze un algoritm genetic. Chestiunea este de a-l folosi când este cu adevărat necesar. Și aceasta vine din experiență, cu timpul.

În cele ce urmează, se vor prezenta câteva aplicații practice ale algoritmilor genetici, în unele domenii de mare actualitate, așa cum ar fi prelucrarea imaginilor, teoria transmisiunii informației (criptanaliza unor cifruri) și optimizarea interfețelor grafice, robotică și ergonomie. Pentru aplicațiile descrise nu se vor trece în revistă decât aspecte cu totul particulare ale modului de implementare a algoritmilor genetici (codarea cromozomială, funcțiile de producere și funcțiile obiectiv), încercând să se realizeze o prezentare mai detaliată a problemei.

Rezolvarea oricărei probleme de extrem poate fi considerată în două moduri: sub un aspect determinist, înglobând informația specifică problemei și aplicând tehnici de gradient, sau sub forma unei tehnici de aproximare a valorii de extrem căutate, problemă echivalentă unei probleme de optimizare combinatorială, ce poate fi rezolvată prin metode stocastice de căutare locală (așa cum sunt algoritmii genetici) În general, optimizarea stocastică a unei funcții este mai lentă decât metodele deterministe corespunzăt are, mai ales datorită nefolosirii informațiilor de derivare; nu este însă mai puțin adevărat că tocmai acest aspect le face mai eficiente, deoarece într-o căutare locală, sunt acceptate degradări temporare ale funcției obiectiv (funcției de cost), evitând astfel extremele locale.

Procesul de căutare stocastică locală poate fi văzut ca un proces cu reglare printr-o buclă de reacție negativă (a se vedea figura 1).

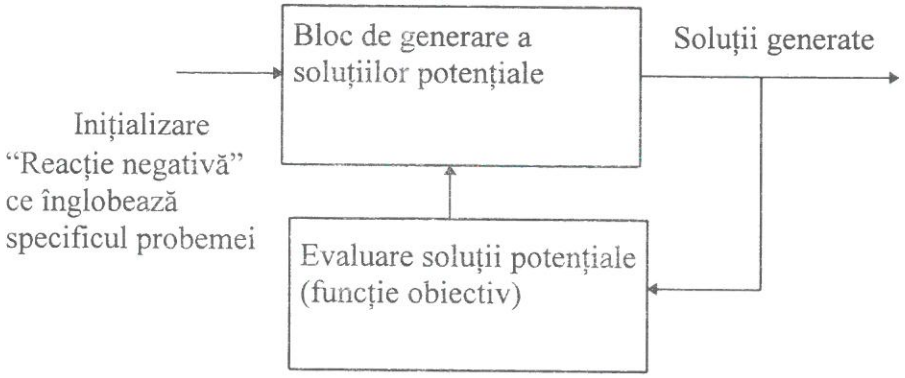


Figura 1.

2.Criptanaliza cifrului rucsac aditiv

Apariția sistemelor criptografice cu chei publice a avut ca scop eliminarea principalului dezavantaj al sistemelor clasice de criptare şi anume necesitatea existenței unei înțelegeri prealabile între corespondenți, prin care să se stabilească cheile specifice de criptare şi decriptare. Pentru transmisia de mesaje criptate între doi utilizatori ai sistemului cu chei publice se utilizează un director public de chei de criptare, asemănător unei cărți de telefon, în care se regăsește cheia de criptare specifică pentru fiecare destinatar (figura 2).

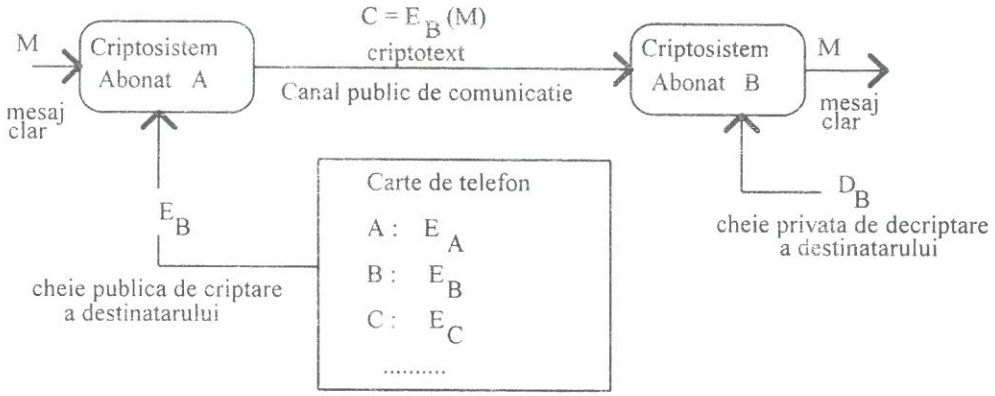


Figura 2. Fluxul mesajelor într-un criptosistem cu chei publice

Cerințele ce se impun transformărilor de criptare/decriptare şi cheilor asociate acestora sunt: decriptarea să fie procesul invers criptării: $D_B(C) = D_B(E_B(M)) = M$ (cu consecința calculului cheii publice de criptare din cheia privată de decriptare); transformările de criptare şi decriptare pentru un utilizator aflat în posesia cheilor corecte să fie simple şi să aibă soluții unice; procesul de decriptare, în absența cheii private de decriptare corespunzătoare, să fie intratabil din punct de vedere al timpului de calcul necesar (cu alte cuvinte, cheia de decriptare privată nu poate fi dedusă uşor din cheia publică de criptare asociată).

Un exemplu clasic şi larg răspândit de sistem de criptare cu chei publice este cifrul rucsac aditiv (numit şi sistemul Merkle-Hellman). Cifrul rucsac aditiv este bazat pe aşa numita problemă a rucsacului, ce se poate formaliza ca găsirea inversului unui vector n-dimensional față de operația de produs scalar: $C = < \mathbf{x}, \mathbf{a} > = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{a}_i$, C (suma rucsacului) şi **a** (vectorul rucsac) cunoscute şi **x** (vectorul de date) necunoscut. Studiul acestei probleme în cadrul teoriei complexității calculului a condus la concluzia că această problemă este de tip NP-complet, nu are soluție analitică şi, în general, soluția nu este unică. Aplicarea problemei rucsac pentru implementarea criptosistemelor cu chei publice s-a făcut prin următoarele particularizări: vectorul de date **x** este un vector cu componente binare $\mathbf{x}_i$; vectorul rucsac **a** este un vector supracrescător, definit prin relația

$\mathbf{a}_k > \sum_{i=1}^{k-1} \mathbf{a}_i$, $k = \overline{2,n}$ şi constituie cheia privată de decriptare. Pentru acest caz, soluţia se calculează simplu cu relaţia : $\mathbf{x}_i = \mathbf{U}(C - \sum_{j=i+1}^{n} \mathbf{x}_j \mathbf{a}_j - \mathbf{a}_i)$, $i = \overline{n,1}$ (U este funcţia treaptă unitate).

În aceste condiţii, cheia publică de criptare se obţine printr-o multiplicare modulară tare $\mathbf{b}_i = \mathbf{a}_i w \bmod M$ (unde M este un număr prim, $M > \sum_{i=1}^{n} \mathbf{a}_i$ şi w este un număr aleator întreg, $w \in [1, \mathbf{a}_n]$), iar criptotextul asociat unui mesaj binar x se obţine prin $C = < \mathbf{b}, \mathbf{x} >$.

Elementele secrete ale cheii private sunt deci vectorul rucsac supracrescător **a**, M şi w. Pe baza acestora, decriptarea unui criptotext C (criptat pe baza cheii publice **b**) se reduce la transformarea sa într-un criptotext echivalent ce se poate decripta cu vectorul rucsac **a** în timp liniar, prin: $C' = Cw^{-1} \bmod M$, cu $ww^{-1} \bmod M = 1$.

Implementările practice ale sistemului Merkle-Hellman se fac cu vectori supracrescători de mai mult de 100 de componente, lungimea fiecărei componente fiind cuprinsă între 100 şi 200 de biţi. Unele variante ale schemei folosesc componente ale vectorului rucsac supracrescător, compuse din părţi aleatoare, părţi deterministe şi părţi ale unui alt vector rucsac supracrescător, aducând o îmbunătăţire a rezistenţei cifrului la atacuri criptografice.

O cale naturală de a aborda problema criptanalizei cifrului rucsac [2] este de a considera căutarea unui vector binar $\tilde{\mathbf{x}}$ care minimizează expresia $|C - \sum_{i=1}^{n} \tilde{\mathbf{x}}_i \mathbf{b}_i|$; vectorul binar găsit poate fi soluţia exactă sau doar o aproximare a acestei soluţii. Aceasta este o problemă de căutare a minimului global a unei funcţii, deci o problemă de optimizare combinatorială.

Algoritmul genetic utilizat este clasic: populaţia are un număr arbitrar de indivizi ($a > 1$), iar spaţiul de căutare este extensia de ordin n a spaţiului binar $\{0,1\}$, $S = \{0,1\}^n$ $(n \in \mathrm{N})$; vecinătatea unui punct s este întreg spaţiul S de căutare. Funcţia de producere înglobează cele două metode de creare a descendenţilor: recombinarea (crossover) şi mutaţia. Acestea sunt operaţii uzuale: prin mutaţie un bit al lui **s** îşi complementează valoarea, iar, prin încrucişare, indivizii **s** şi **t** fac schimb de segmente de biţi. Funcţia de producţie poate fi scrisă compact ca: $f_{prod}(\beta, x) = \begin{cases} cross\_over(s,t), x = \{s,t\} \\ mutatie(s), x = \{s\} \end{cases}$, unde x este mulţimea părinţilor şi β înglobează probabilităţile de mutaţie, respectiv recombinare. Funcţia de reducere (sau selecţie) păstrează indivizii cei mai apropiaţi de soluţie din noua generaţie:

$$f_{red}(y) = \left\{ s_1, \ldots, s_a \in S \middle| \begin{array}{l} \forall i, 1 \le i \le a, \\ \forall s \in y \setminus \{s_1, \ldots, s_a\}, \\ f(s_i) \le f(s) \end{array} \right\}$$

unde funcţia de cost $f(\mathbf{x})$ asociată problemei este $k \in \{1/4, 1/2, 1, 2\}$,

$$f(\mathbf{x}) = |C - \sum_{i=1}^{n} \mathbf{x}_i b_i|^k / (\sum_{i=1}^{n} b_i)^k$$

(se observă normarea funcţiei de cost prin $(\sum_{i=1}^{n} b_i)^k$ (suma rucsacului), pentru a obţine încadrarea în intervalul [0;1]).

Algoritmul genetic poate fi caracterizat de un factor de merit format ca un produs între mărimea populaţiei, numărul de generaţii (iteraţii) până la găsirea soluţiei, gradul de apropiere al acestei soluţii faţă de optim (reprezentat printr-o funcţie de tip $\alpha . f(x), \alpha \in R^*$). În urma simulărilor efectuate asupra unui vector rucsac de 16 componente, convergenţa la soluţie a fost obţinută după 10 - 50 de generaţii, dependentă puternic de funcţia de cost $f(s)$ folosită (figura 3) şi de

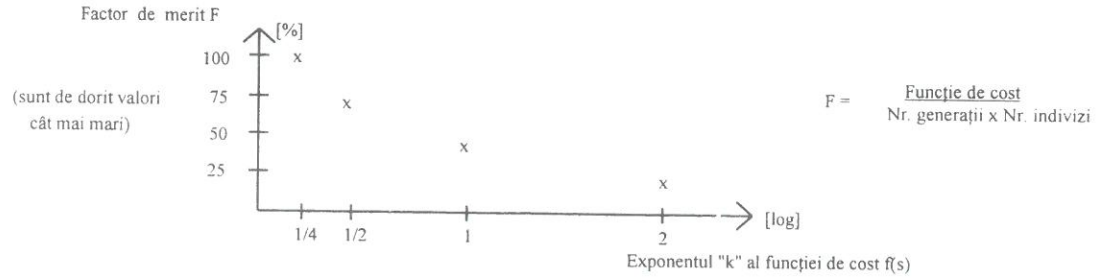


Figura 3. Dependenţa factorului de merit al algoritmului genetic de funncţia de cost

soluția s_0 căutată. Numărul de indivizi ce compun populația a fost variat între 10 și 40; viteza de convergență crește o dată cu creșterea populației, dar pentru populații ce conțin mai mult de 25 de indivizi timpul de calcul total (90% din timp se face evaluarea funcției de cost) a fost mai mare, deși numărul de generații până la convergență a fost mai mic.

3. Aplicații ale algoritmilor genetici în prelucrarea imaginilor

Schema bloc generală a unui sistem de prelucrare digitală a imaginilor este prezentată în figura următoare (figura 4).

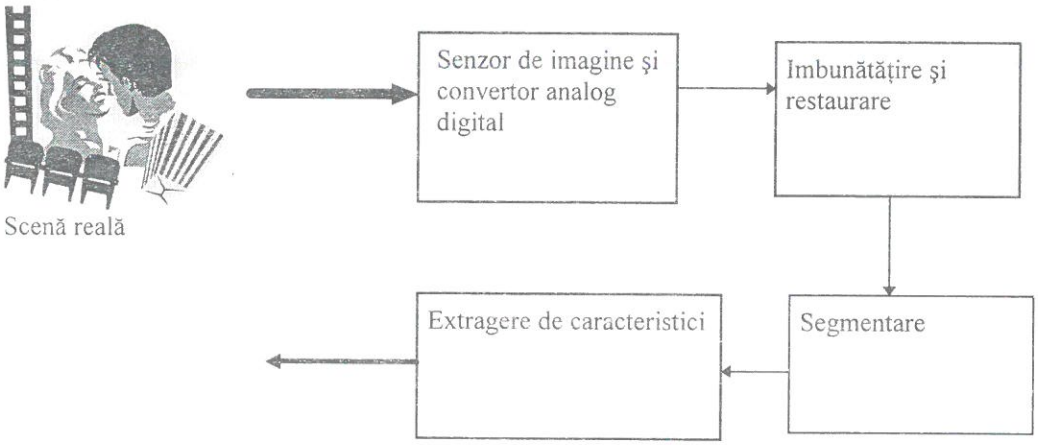


Figura 4.

Vom discuta câte o aplicație tipică a algoritmilor genetici pentru fiecare etapă efectivă de prelucrare a imaginii (îmbunătățirea imaginii, segmentarea imaginii, extragerea de caracteristici).

3.1. Restaurarea imaginii cu algoritmi genetici

Procesul de restaurare a imaginilor presupune refacerea imaginii originale prin îndepărtarea zgomotelor suprapuse de-a lungul lanțului de achiziție și compensarea unor deficiențe de captare a scenei (focalizare, urmărire a obiectelor în mișcare). Unul din modelele generale, folosite pentru restaurare, presupune imaginea degradată, obținută printr-o relație de tipul $g = f * h + n$, unde g este imaginea degradată, f este imaginea originală, h este modelul echivalent de degradare și n este zgomotul adăugat imaginii. Problema este de a obține pe f din g cu o aproximare cât mai bună, aproximare ce se măsoară pe baza unui anumit criteriu (eroare absolută, eroare medie pătratică, entropie maximă etc.).

Un individ al populației (o soluție potențială) este o imagine [3]. Un cromozom este un pixel al imaginii (respectiv nivelul său de intensitate). Este interesant de observat că structura cromozomială a individului este o matrice (și nu un vector, ca în cazul uzual), păstrând astfel corelația între pixeli (atât pe verticală, cât și pe orizontală), caracteristică esențială a unei imagini. Această structură cromozomială se reflectă și în definirea operației de recombinare între indivizi (cross-over). Prin recombinare se vor interschimba blocuri de m X n cromozomi (a căror dimensiune este dimensiunea medie a celui de corelație din imagine), deoarece proprietățile de corelare trebuie să se păstreze prin recombinare.

Funcția obiectiv este entropia imaginii definită pe baza valorilor de intensitate (niveluri de gri) din fiecare punct al imaginii f_i, prin expresia

$$S(f_1, f_2, \ldots f_{N^2}) = -\sum_{i=1}^{N^2} \frac{f_i}{\sum_{j=1}^{N^2} f_j} \log \frac{f_i}{\sum_{j=1}^{N^2} f_j},$$ la

care se adaugă constrângerea

$$Q(f_1, f_2, \ldots f_{N^2}) = \frac{1}{2\sigma^2} \sum_{j=1}^{m} \left(\sum_{i=1}^{N^2} h_{ji} f_i - g_i \right)^2$$ (p

entru un zgomot staționar), unde h_{ij} sunt coeficienți ce definesc modelul de degradare.

3.2. Segmentarea imaginii cu algoritmi genetici

Segmentarea este, în general, prima și cea mai dificilă etapă în orice proces de viziune artificială; etapele următoare de detecție a obiectelor, extragere a parametrilor, recunoaștere și clasificare sunt puternic influențate de calitatea segmentării. Deși au fost propuse numeroase tehnici de

segmentare , nu au fost găsite suficient de generale pentru mai multe tipuri de imagini.

Segmentarea se poate caracteriza printr-un număr mare de parametri de control, ce trebuie ajustați pentru obținerea unei performanțe optime. In aceste sisteme, dimensiunea spațiului de căutare a parametrilor poate deveni prohibitiv de mare (cel puțin dacă nu este parcurs deosebit de eficient). Parametrii majorității algoritmilor de segmentare interacționează în mod complex, adesea neliniar, fapt ce conduce la mari dificultăți în modelarea algoritmică sau bazată pe reguli a comportării parametrilor.

Schema generală a sistemului este:

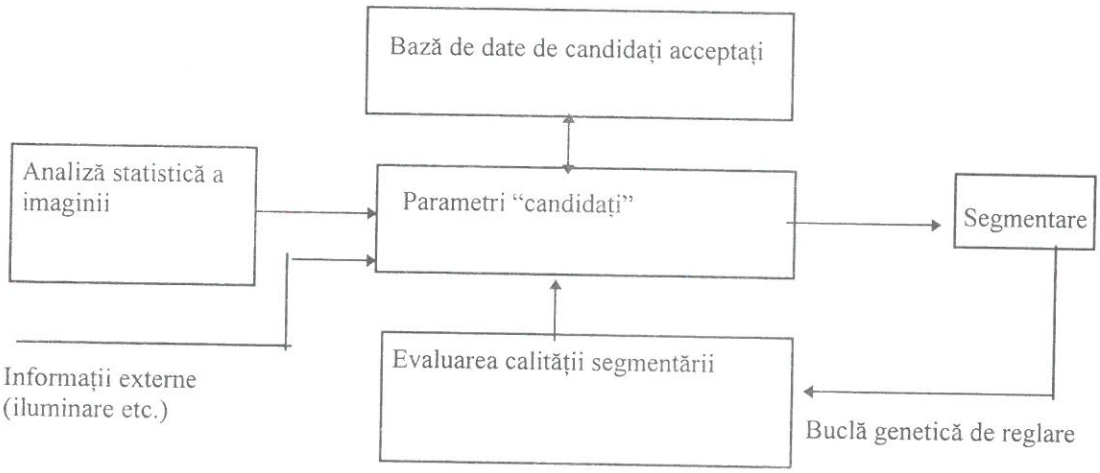


Figura 5.

3.3. Extragerea de primitive geometrice din imagine

După o operație de segmentare, imaginea de prelucrat este transformată într-o colecție de regiuni (obiecte) având proprietăți asemănătoare. Unul din modurile posibile de continuare a procesului de viziune artificială este extragerea de primitive (caracteristici) geometrice: linii, cercuri, elipse (pentru un spațiu bidimensional), plane, sfere, conuri etc.(pentru un spațiu tridimensional).

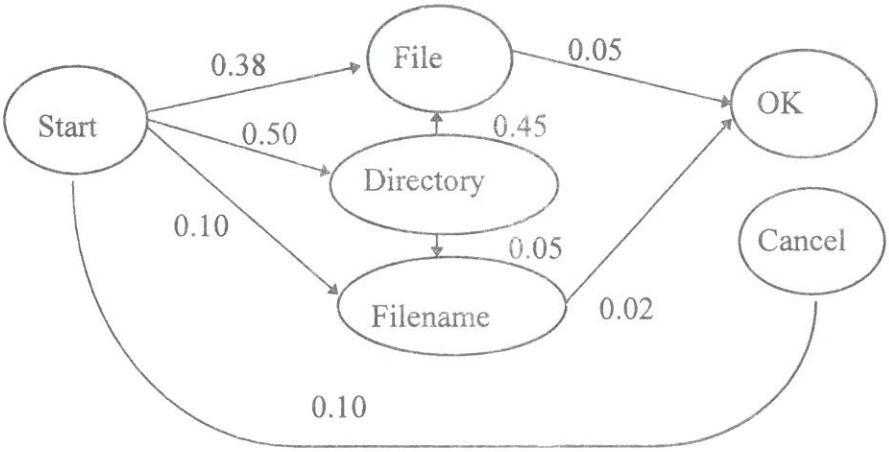
Un mod de extragere a primitivelor geometrice este determinarea unor submulțimi minimale, pe baza cărora se poate calcula vectorul de parametri ai ecuației ce descrie respectiva primitivă, ecuație de forma f(**p**,**a**)=0, unde **p** este vectorul parametrilor și **a** este vectorul coordonatelor punctelor din mulțimea minimală. (De exemplu dreapta în plan este descrisă de ecuația $a_0 + a_1 x + a_2 y = 0$). Particularitățile de implementare ale algoritmului genetic sunt: fiecare individ descrie o submulțime minimală de puncte, cromozomii unui individ reprezintă numărul de ordine a unui punct din lista punctelor de intrare, numărul de cromozomi ai fiecărui individ depinde de tipul de primitivă ce se dorește extrasă (2 pentru o dreaptă plană, 3 pentru un cerc, 4 pentru o elipsă...). Funcția de cost, folosită la evaluarea calității soluțiilor este numărul de puncte din lista de intrare, care se află într-o bandă de lățime Δ în jurul primitivei calculate (număr ce trebuie minimizat).

4. Aplicații ale algoritmilor genetici în ergonomie

Una din problemele ce derivă din ergonomie este și optimizarea aspectului și a conținutului ferestrelor de dialog, folosite la introducerea datelor în aplicații de tip Windows [5], în sensul utilizării unui număr minim de fixări ale ochiului necesare extragerii informației de poziție, distanțele de parcurs între diferitele câmpuri ce trebuie completate, numărul de schimbări de direcție ce se efectuează cu mouse-ul pentru navigarea între câmpuri.

Adecvarea aspectului unei ferestre de dialog se poate deci exprima prin formula generică : $\text{cost} = \sum_i f_i \cdot c_i$, unde f_i este frecvența unei tranziții între două câmpuri și c_i este costul respectivei tranziții.. Frecvențele de tranziție se pot determina fie prin observarea statistică a utilizatorilor ce folosesc sistemul, fie prin identificarea tuturor secvențelor de acțiune posibile și prin atașarea de frecvențe pentru fiecare acțiune. Aceasta este funcția de cost ce se dorește minimizată, iar cromozomii indivizilor vor codifica poziția diferitelor câmpuri ale interfeței.

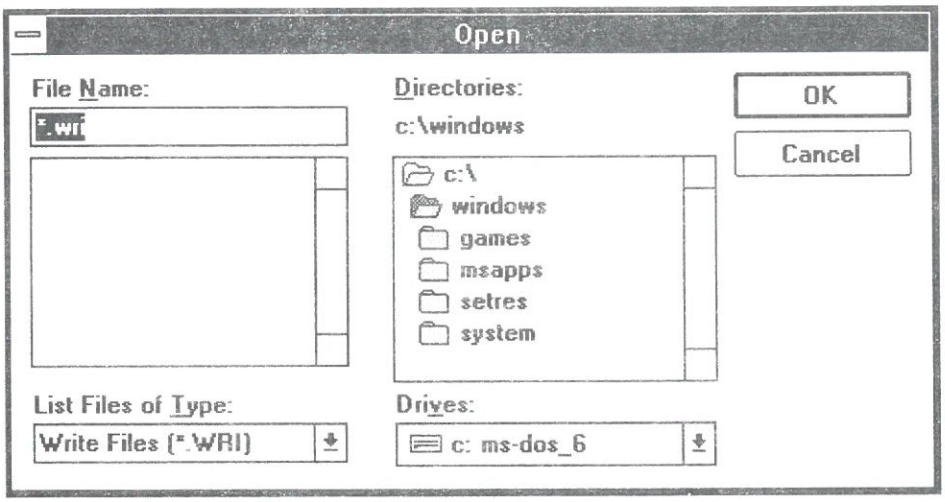
Exemplul următor prezintă cazul unei ferestre standard de dilaog al meniului "File" din utilitarul Write (Windows 3.1), pentru care graful frecvențelor de tranziție este [5]:

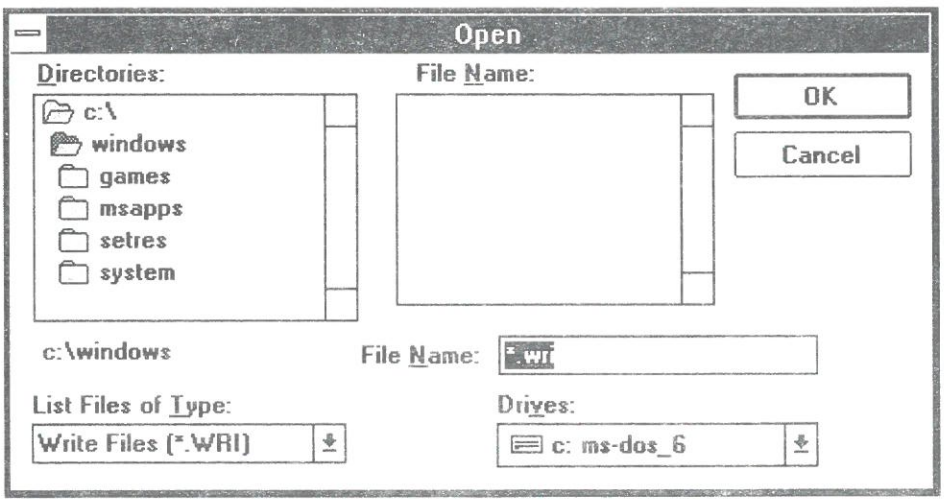


Figurile următoare prezintă forma originală a ferestrei de dialog şi forma optimală obţinută în urma minimizării funcţiei de cost. Metoda poate fi extinsă şi la probleme de organizare a locului de muncă, a halelor de producţie, a amplasării anexelor unui robot industrial [6].

Bibliografie

1. **MICHAILEWICZ, Z.** : Genetic Algorithms + Data Structures = Evolutionary Programming, Springer Verlag, Berlin, 1992.





2. **VERTAN, C., BUZULOIU, V., MALCIU, M., POPESCU, V.**: Criptanaliza cifrurilor rucsac prin metode de căutare stochastică", A XXVIa Sesiune de Comunicări ştiinţifice cu participare internaţională, Academia Tehnică Militară, 16-17 Noiembrie 1995, vol. 4, pp. 131-138.

3. **TOMA, C., DATCU, M.:** Genetic Algorithm for Maximum Entropy Image Restoration. În: Neural & Stochastic Methods in Image and Signal Processing IV Meeting SPIE, San Diego CA, 1994.

4. **BHANU, B., LEE, S., MING, J.** : Self Optimizing Image Segmentation System Using a Genetic Algorithm. În: Proc. ICGA IV, 13-16 July 1991, San Diego CA, pp. 362-369.

5. **SEARS, A.**: Layout Appropriatness: A Metric for Evaluating User Interface Widget Layout. În: IEEE Trans. on Software Engineering, Vol. 19, No. 7, July, 1993, pp. 707-719.

6. **GEANGĂLĂ, C., VERTAN, C.** : Industrial Robot Workspace Optimisation by Relaxation Techniques. În: Proc. of ECCO IX, 1-3 April, 1996, Dublin, Ireland, session TA-1, pp. 81-82.

ICI - Colectivul de Cercetare Independent 1.10.3

BIBLIOTECA DE MODELE DE SIMULARE SI CONTROL PENTRU ECOLOGIE SI PROTECTIA MEDIULUI

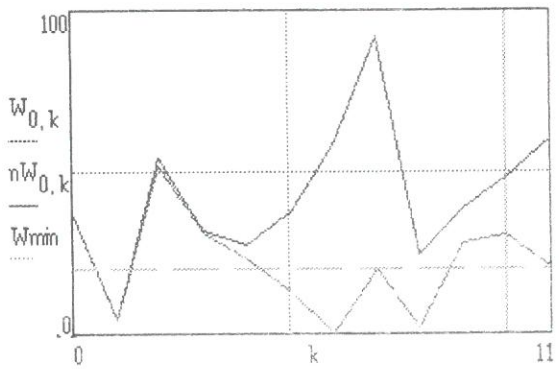
Biblioteca a fost realizata pentru a veni în sprijinul realizatorilor de sisteme informatice si/sau de aplicatii din domeniul ecologiei si protectiei mediului.

***Functiunile* bibliotecii de modele de simulare si control pentru ecologie si protectia mediului sunt urmatoarele:**

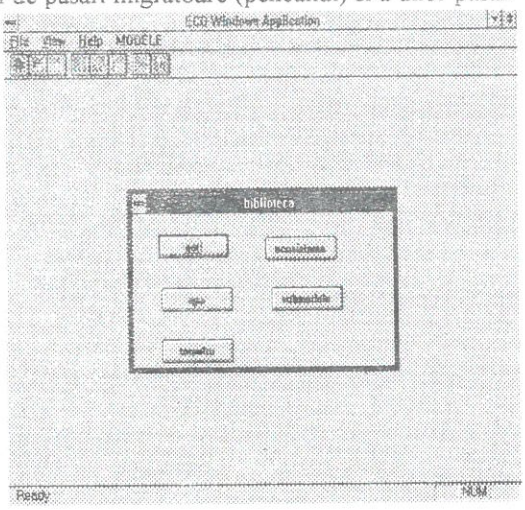
- furnizarea unor modele gata construite, care permit utilizatorului realizarea de experimente de simulare si control a sistemelor ecologice si de protectie a mediului;
- asistarea utilizatorului în construirea de noi modele de simulare si control (cu ajutorul nivelurilor : submodele si formule de calcul);

***Structura* bibliotecii de modele de simulare si control pentru ecologie si protectia mediului este ierarhizata pe mai multe niveluri:**

- nivelul global
- nivelul modele de simulare si control pentru: ecologie, mediu atmosferic, mediu acvatic, mediu terestru
- nivelul submodele
- nivelul formule de calcul



Modelele mediului atmosferic se refera la simularea si controlul difuziei poluantilor chimici industriali în atmosfera urbana. Modelele mediului acvatic se refera la simularea si controlul proceselor hidrologice (HYDRO) si a proceselor complexe hidro-bio-chimice (DELTA). .Modelele mediului terestru se refera la simularea si controlul unui agroecosistem (SOL) si a unui ecosistem forestier (FOREST). Dintre modelele ecologice subliniem modelele de simulare si control al dinamicii unor populatii de pasari migratoare (pelicanul) si a unor pasari nemigratoare (cormoranul).



Selectarea unui model se face automat cu ajutorul mouse-ului si a butoanelor care apar in ferestrele(windows) pe ecran. Rezultatele de simulare si/sau control sunt prezentate grafic.

Cerinte hardware si software.

Modelele de simulare si control sunt realizate cu ajutorul limbajului Mathcad sub Windows, utilizând un calculator PC 486. Butoanele de selectare a modelelor au fost realizate sub Visual C++.

Cui îi este destinat produsul ?

Biblioteca de modele de simulare si control pentru ecologie si protectia mediului este destinata, în principal, urmatorilor utilizatori:

- cercetatori, proiectanti, analisti si programatori din laboratoare si centre de calcul din institutele cu profil de ecologie, biologie, hidrologie, silvicultura, agronomie, calitatea si protectia mediului înconjurator;
- realizatori de sisteme informatice si/sau aplicatii din domeniul ecologiei, biologiei, hidrologiei, silviculturii, agronomiei, calitatii si protectiei mediului înconjurator.

Se ofera: Biblioteca de modele pe discheta de 3½ inches, pentru PC 486.
Documentatia specifica.
Asistenta tehnica la implementare.

Contactati-ne pentru informatii suplimentare si demonstratii :

ICI - Bd. Averescu nr. 8-10 sector 1
Telefon: 6 65 60 60 / interior: 172
Fax: (1) 312 85 39
Email: sflorin@roearn.ici.ro
Persoana de contact: dr. ing. Florin Stanciulescu