REZOLVAREA UNOR PROBLEME DE OPTIMIZARE MULTI-OBIECTIV BAZATĂ PE ALGORITMI EVOLUTIVI

Iulia Cristina Rădulescu

Universitatea POLITEHNICA din Bucureşti - Facultatea de Automatică şi Calculatoare

iulia.radulescu1702@yahoo.com

Rezumat: Cele mai multe probleme de optimizare care apar în practică au un caracter multi-obiectiv deoarece în cadrul lor este necesar să fie optimizate simultan mai multe funcţii obiectiv.

În acest articol se realizează o analiză a modului de rezolvare a problemelor de optimizare multi-obiectiv, cu accent pe rezolvarea bazată pe algoritmi evolutivi. Se prezintă o clasificare a tehnicilor de optimizare în enumerative, deterministe şi stocastice. Din cadrul tehnicilor de optimizare stocastice sunt detaliaţi algoritmii de calcul evolutiv (Evolutionary Algorithms – EA). Aceşti algoritmi sunt folosiţi cu succes pentru rezolvarea problemelor de optimizare multi-obiectiv. Se prezintă concepte de bază utilizate în cadrul algoritmilor evolutivi şi se trec în revistă mai mulţi algoritmi evolutivi pentru rezolvarea problemelor de optimizare multi-obiectiv. În final, se rezolvă patru probleme de optimizare multi-obiectiv, cu ajutorul unui cunoscut algoritm evolutiv: algoritmul genetic cu sortare ne-dominată (Non-Dominated Sorting in Genetic Algorithms - NSGA-II). Programul Matlab ce implementează algoritmul NSGA-II calculează frontiera Pareto a celor patru probleme considerate.

Abstract: Most optimization problems that arise in practice have multiple objectives because they suppose to optimize simultaneously several objective functions.

In this article we analyze several approaches for solving multi-objective optimization problems, focusing on the approaches based on evolutionary algorithms. We present a classification of optimization techniques in enumerative, deterministic and stochastic. From the stochastic optimization techniques are detailed Evolutionary Algorithms - EA. These algorithms are successfully used for solving multi-objective optimization. We present the basic concepts used in evolutionary algorithms and an overview of several evolutionary algorithms for solving multi-objective optimization problems. Finally, we solve four multi-objective optimization problems using a well-known evolutionary algorithm called the Non-Dominated Sorting Genetic Algorithms - NSGA-II. The Matlab program, that implements the algorithm NSGA-II, computes the Pareto frontier of the four multi-objective optimization problems considered.

Key Words: Multi-objective optimization, Evolutionary Algorithms, Non-Dominated Sorting Genetic Algorithms, Pareto frontier

1. Introducere

Cele mai multe probleme de optimizare care apar în practică au un caracter multi-obiectiv deoarece în cadrul lor este necesar să fie optimizate simultan mai multe funcţii obiectiv. Problemele de optimizare multi-obiectiv joaca un rol important în inginerie, management şi în multe alte domenii. Funcţiile obiectiv în cadrul unei probleme de optimizare multi-obiectiv pot să fie, în cele mai multe cazuri, în conflict. De exemplu în acelaşi timp unele dintre funcţiile obiectiv trebuie maximizate iar altele minimizate. De asemenea optimul realizat pentru anumite funcţii obiectiv nu coincide cu optimul realizat de alte funcţii obiectiv. Găsirea unor soluţii care să realizeze un compromis, cu o bază raţională, a reprezentat o provocare pentru cercetători de-a lungul timpului.

O problemă de optimizare multi-obiectiv poate fi definită ca problema găsirii unui vector (componentele acestuia având semnificaţia de variabile de decizie) care satisface anumite restricţii şi optimizează o funcţie vectorială ale cărei componente reprezintă funcţiile obiectiv [18]. Aceste funcţii descriu din punct de vedere matematic criterii de performanţă care de cele mai multe ori sunt în conflict una cu alta. Termenul de “optimizare” înseamnă găsirea unei soluţii care furnizează valori pentru funcţiile obiectiv acceptabile de către decident.

Articolul este organizat astfel: se prezintă o clasificare a tehnicilor de optimizare în enumerative, deterministe şi stocastice şi se dau exemple de tehnici de optimizare din cadrul fiecărui tip. Sunt detaliate câteva puncte tari şi puncte slabe ale acestor tehnici de optimizare. Se prezintă apoi algoritmii bazaţi pe tehnici de inteligenţă artificială. Se pune accentul pe algoritmii de calcul evolutiv (Evolutionary Algorithms – EA) care pot fi utilizaţi cu succes pentru rezolvarea

problemelor de optimizare multi-obiectiv. Se prezintă concepte de bază utilizate în cadrul algoritmilor evolutivi şi se trec în revistă mai mulţi algoritmi evolutivi pentru rezolvarea problemelor de optimizare multi-obiectiv.

În final, se rezolvă patru probleme de optimizare multi-obiectiv cu ajutorul unui cunoscut algoritm evolutiv: algoritmul genetic cu sortare ne-dominată (Non-Dominated Sorting in Genetic Algorithms - NSGA-II). Programul Matlab ce implementează algoritmul NSGA-II calculează frontiera Pareto a celor patru probleme.

2. Tehnici de optimizare

În general tehnicile de optimizare sunt clasificate în trei categorii: enumerative, deterministe şi stocastice. Figura 1 ilustrează exemple din fiecare tip [2].

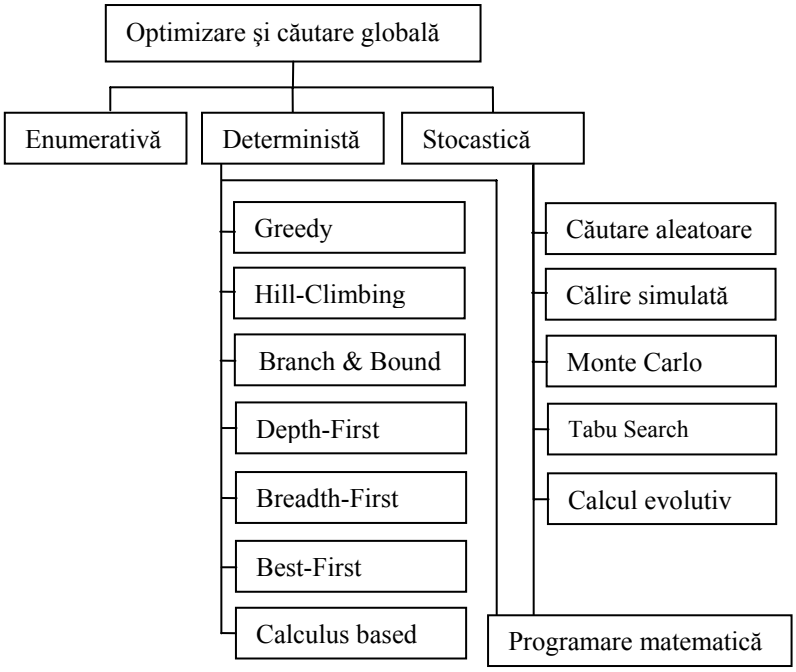

Figura 1. Abordări privind optimizarea globală

Câteva puncte tari şi puncte slabe ale acestora sunt detaliate în continuare [2].

Cele mai simple tehnici de optimizare sunt cele enumerative. În cadrul acestor tehnici este evaluată fiecare posibilă soluţie în spaţiul de căutare care are un număr finit de soluţii admisibile. Totuşi se poate observa cu uşurinţă că această tehnică este ineficientă sau chiar infezabilă atunci când spaţiul de căutare are un număr mare de elemente.

Cum majoritatea problemelor din viaţa reală sunt intensiv computaţionale trebuie implementate reguli de limitare a spaţiului de căutare pentru a obţine "soluţii acceptabile" într-un timp "acceptabil". Algoritmii determinişti încearcă acest lucru prin încorporarea de cunoştinţe privind domeniul problemei. Algoritmii "greedy" aleg optime locale, presupunând că sub-soluţiile optimale sunt totdeauna parte a soluţiei optime globale. Algoritmii "Hill-climbing" caută în direcţia de ascensiune cea mai abruptă de la o poziţie curentă. Aceşti algoritmi lucrează bine pe funcţii unimodale, dar prezenţa optimului local, a platourilor şi a crestelor reduc eficienţa algoritmilor. Algoritmii "Greedy" şi "Hill-climbing", extind un nod în mod repetat, examinează toţi posibili succesori (pentru extindere către nodul cel mai bun) şi nu păstrează nici o înregistrare a nodurilor extinse pe care le-a urmărit.

Tehnicile de căutare "branch and bound" necesită algoritmi specifici (euristici sau de decizie) pentru a limita spaţiul de căutare. Aceştia calculează legături la un nod dat care determină când un

nod este "promițător". Apoi legăturile nodurilor sunt comparate şi algoritmul trece către nodul "cel mai promițător". În căutarea "depth-first" ordinea de căutare este independentă de localizarea soluției (cu excepția căutării de finalizare). Ea extinde un nod, generează toți succesorii, extinde un succesor şi aşa mai departe. Atunci când căutarea este blocată (adică de ajunge la nivelul cel mai de jos al arborelui) se rezumă la cel mai adânc nod din stânga. Căutarea "breadth-first" diferă de căutarea "depth-first" prin acțiunile pe care le ia după extinderea unui nod, când explorează progresiv arborele, un nivel la un moment dat. Căutarea "best-first" foloseşte informație euristică pentru a plasa valori numerice într-un nod "promițător". În final căutarea "calculus-based" necesită continuitate în domeniul unor variabile pentru ca o valoare optimă să se poată obține.

Algoritmii determinişti sunt utilizați într-o gamă largă de probleme. Totuşi multe probleme de optimizare multi-obiectiv sunt de dimensiune mare, discontinue, multimodale şi/sau NP complete. Metodele deterministe sunt adesea ineficiente când se aplică la probleme NP complete sau când se aplică la probleme de dimensiune mare din cauză că ele depind de cerințe privind domeniul problemei pentru a direcționa sau a limita căutarea în spațiul mare de căutare. Aceste probleme sunt numite *neregulate*. Din cauză că multe probleme de optimizare multi-obiectiv sunt neregulate tehnicile de căutare enumerative şi deterministe au un grad de eficiență scăzut. Ca abordări alternative pentru rezolvarea acestor probleme "neregulate" au fost dezvoltate alte tehnici de căutare stocastică şi abordări de optimizare cum ar fi călirea simulată ("Simulated Annealing"), metodele Monte Carlo, căutarea "tabu search" şi algoritmi de calcul evolutiv ("Evolutionary Algorithms" – EA). Acestea nu pot garanta că soluția găsită este cea optimă. Ele găsesc în general soluții bune pentru o gamă largă de probleme de optimizare în care metodele tradiționale deterministe întâmpină greutăți.

3. Algoritmi de calcul evolutiv

Dintre algoritmii bazați pe inteligență artificială avansată fac parte algoritmii de calcul evolutiv ("Evolutionary Algorithms" – EA). Algoritm evolutiv este un termen generic pentru mai multe metode de căutare stochastice care simulează computațional procesul de evoluție naturală. Ca domeniu de cercetare domeniul algoritmilor evolutivi este un domeniu tânăr, deşi tehnici asociate lui există de aproape 50 de ani. Algoritmii evolutivi cuprind algoritmii genetici (Genetic Algorithms - GA) strategii de evoluție (Evolution Strategies - ES) şi programarea evolutivă (Evolutionary Programming - EP). Aceste tehnici sunt bazate pe conceptul de evoluție din natură şi pe conceptul lui Darwin de supraviețuire a celui mai bine adaptat. Ceea ce este comun acestora sunt competiția, variația aleatoare, reproducerea şi selecția de indivizi dintr-o populație. În general algoritmii evolutivi constau dintr-o populație de soluții (indivizi) manipulată printr-un set de operatori şi evaluați printr-o funcție de potrivire "fitness". Fiecare valoare de potrivire asociată soluției determină cine va supraviețui într-o nouă generație.

3.1. Concepte generale privind algoritmii evolutivi

O structură sau *individ* este o soluție codificată a unei probleme. În mod obişnuit un individ este reprezentat de un şir (sau şir de şiruri) ce corespunde la un *genotip* biologic. Acest genotip este compus din unul sau mai mulți *cromozomi,* unde fiecare cromozom este compus din *gene* care iau diverse valori (*alele*) dintr-un alfabet genetic. Un *loc (poziție)* identifică o poziție a genei în cadrul unui cromozom. În final o mulțime de cromozomi este numită *populație.* Un exemplu privind aceste concepte este ilustrat în figura 2 (pentru cromozomi cu valori binare).

În matricea din figura 2, pe linii se află cromozomii, care reprezintă şiruri finite de valori binare. Mulțimea liniilor matricei reprezintă populația asupra căreia acționează operatorii evolutivi.

Ca şi în natură operatorii evolutivi operează pe o populație încercând să genereze soluții cu o cât mai mare potrivire (fitness). Operatorii principali sunt operatorii de *mutație, recombinare şi selecție*.

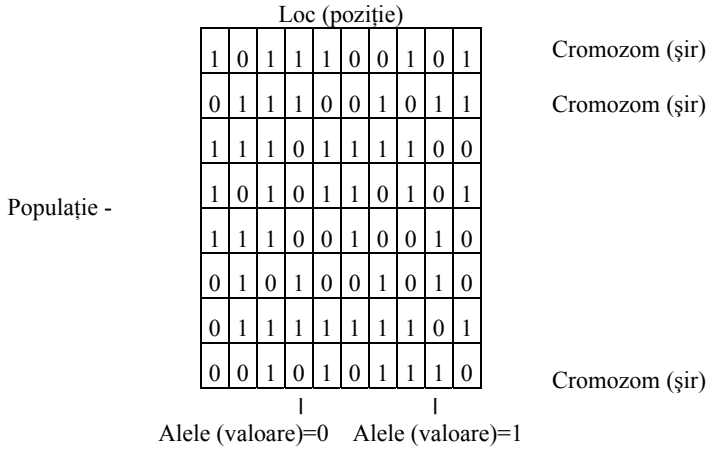

Figura 2. Structura de date şi terminologia algoritmilor evolutivi

Componentele unui algoritm evolutiv sunt:

- populația - mulțime de indivizi (soluții);
- părinții - membrii generației curente;
- copiii - membrii generației următoare;
- generația –o populație creată în cadrul unei iterații.

Structura de date este compusă din:

- cromozomi - formă de a codifica soluția: vector (şir) ce constă din gene cu alele asignate
- potrivire (fitness) - număr asignat unei soluții (ceea ce se doreşte).

Cromozomii cu valori reale folosesc aceleaşi operații de *mutație, recombinare şi selecție* deşi sunt implementate diferit.

Toți algoritmii evolutivi folosesc nişte submulțimi de variație a acestor operații. Există multe variații ale operatorilor de bază. Acestea sunt dependente de domeniul problemei, de restricții şi afectează structura cromozomială şi a atelelor. Un algoritm evolutiv are atât o funcție de fitness cât şi una obiectiv care sunt fundamental diferite. Funcția obiectiv defineşte condiția de optimalitate a algoritmilor evolutivi (şi este o caracteristică a domeniului problemei) în timp ce funcția de fitness (în domeniul algoritmului) este o măsură a cât de bine o soluție particulară satisface condiția de optimalitate şi asociază o valoare reală corespunzătoare acelei soluții.

Multe alte tehnici de selecție sunt implementate de către algoritmii evolutivi. De exemplu "turneu" şi "ordonare". Operatorii de selecție “turneu” operează prin alegerea aleatoare a unui număr q de indivizi din populația generată şi selectează pe cel mai bun pentru a supraviețui în generația următoare. Turneele binare (q=2) sunt probabil cele mai cunoscute. "Ordonarea" asignează probabilități de selecție numai pe baza rangului individului ignorând valorile fitness. Două alte tehnici de selecție sunt strategiile de selecție (μ+λ) şi (μ, λ) unde μ reprezintă numărul de soluții părinți şi λ numărul de copii. Primul selectează μ cei mai buni indivizi extraşi atât dintre părinți cât şi dintre copii, cel de-al doilea selectează μ indivizi numai din populația de copii.

De ce este atât de importantă alegerea tehnicii de selecție a algoritmului evolutiv ? Două obiective conflictuale sunt comune tuturor algoritmilor evolutivi de căutare: explorarea şi exploatarea. Un element esențial al selecției unui algoritm evolutiv este mecanismul de control care determină tipul de căutare executat. O clasificare generală a tehnicilor de selecție conține: tehnici proporționale, liniare, tehnici de ordonare, turnee şi selecție (μ, λ) [1].

În final, momentul în care se opreşte execuţia unui algoritm evolutiv, este determinat de o funcţie de decizie.

Diferenţele majore între trei instanţieri ale calculului evolutiv (algoritmii genetici - GA, strategii evolutive - ES şi programarea evolutivă - EP), din punct de vedere al operaţiilor de mutaţie, recombinare şi selecţie, sunt prezentate în Tabelul 1.

Tabelul 1. Diferenţe majore între EP, ES şi GA

Tip de algoritm evolutiv	**Reprezentare**	**Operatori evolutivi**
Programarea evolutivă	Valori reale	Mutaţie şi selecţie (μ+λ)
Strategii de evoluţie	Valori reale şi parametri strategici	Mutaţie, recombinare şi selecţie (μ+λ) sau selecţie (μ, λ)
Algoritmii genetici	Valori binare istorice; valori reale	Mutaţie, recombinare şi selecţie

3.2. Algoritmi evolutivi pentru rezolvarea problemelor de optimizare multi-obiectiv

Potenţialul algoritmilor evolutivi pentru rezolvarea problemelor de optimizare multi-obiectiv fost demonstrat în anii 1960 de către Rosenberg în teza lui de doctorat [22]. Studiul lui Rosenberg conţine o sugestie care a condus către optimizarea multi-obiectiv. Sugestia a fost folosirea de "proprietăţi" în simularea genetică şi chimică a unei populaţii de organisme cu o singură celulă. Prima implementare a ceea ce numim astăzi ca algoritm evolutiv pentru rezolvarea problemelor de optimizare multi-obiectiv (Multi-Objective Evolutionary Algorithm - MOEA) este dat de David Schaffer, care propune algoritmul genetic de evaluare vectorială (Vector Evaluation Genetic Algorithm- VEGA) în 1984 (în teza lui de doctorat [23]). Lucrarea lui Schaffer a fost prezentată la conferinţa *First International Conference on Genetic Algorithms* [24]. Este interesant de observat că problema fără restricţii cu două funcţii obiectiv propusă de acesta a devenit o problemă uzuală de test pentru a valida mai multe tehnici de optimizare multi-obiectiv evolutive în următorii ani.

Algoritmii evolutivi pentru rezolvarea problemelor de optimizare multi-obiectiv sunt cunoscuţi în literatura de specialitate ca algoritmi MOEA (Multi-Objective Evolutionary Algorithm – MOEA).

O listă cu algoritmi MOEA este furnizată în continuare:

- algoritm genetic pentru evaluare vectorială (Vector Evaluated GA - VEGA) [17], [23], [24]
- algoritm genetic - Ordonare lexicografică (Lexicographic Ordering GA) [9]
- Strategie evolutivă de optimizare vectorială (Vector Optimized Evolutions Strategy - Voes) [15]
- algoritmi genetici bazaţi pe ponderare (Weight-Based GA - WBGA) [10]
- algoritm genetic multi-obiectiv (Multiple Objective GA - MOGA) [8]
- Niched Pareto GA (NPGA, NPGA 2) [7], [11]
- algoritm genetic prin sortare ne-dominată (Nondominated Sorting GA - NSGA, NSGA-II) [5], [6], [25]
- algoritm genetic Pareto bazat pe distanţă (Distance-based Pareto GA - DPGA) [19], [20]
- algoritm genetic termodinamic (Thermodynamical GA - TDGA) [12]

- algoritm evolutiv puternic Pareto (Strength Pareto Evolutionary Algorithm - SPEA, SPEA2) [27].[28]
- algoritm genetic multi-obiectiv dezordonat (Multi-Objective Messy GA - MOMGA-I,II,III) [3], [4], [26], [29], [30]
- algoritm evolutiv arhivat Pareto (Pareto Archived ES - PAES) [13], [14]
- algoritm de optimizare multi-obiectiv Bayesian (Multi-Objective Bayesian Optimization Algorithm - mBOA) [16], [21]

4. Studiu de caz

Algoritmii clasici de optimizare au dificultăți adesea în găsirea optimului global sau în cazul optimizării multi-obiectiv în determinarea frontierei Pareto. Pentru a rezolva problemele de optimizare multi-obiectiv, cercetarea a fost direcționată către algoritmi evolutivi.

Studiul de caz din lucrarea de față se referă la patru probleme de optimizare multi-obiectiv numite MOP1, MOP2, MOP3 şi MOP4, fiecare cu câte două funcții obiectiv şi restricții. Aceste probleme sunt rezolvate cu un cunoscut algoritm evolutiv - algoritmul genetic cu sortare ne-dominată (Non-Dominated Sorting in Genetic Algorithms - NSGA-II). Algoritmul genetic cu sortare ne-dominată (Non-Dominated Sorting in Genetic Algorithms - NSGA) [25] este un algoritm popular bazat pe non-dominare pentru optimizare multi-obiectiv. Este un algoritm eficient dar a fost, în general criticat pentru complexitatea de calcul, lipsa de elitism şi pentru alegerea apriori a unui parametru optim pentru partajare. A fost dezvoltată apoi o versiune modificată [6] care are un algoritm de sortare mai bun, încorporează elitism şi nu este necesar să se aleagă apriori un parametru pentru partajare.

Programul realizat, ce are la bază algoritmul NSGA-II, este o adaptate a programului „open source" a lui Aravind Seshadri din biblioteca MatLab Central. Programul are 11 funcții si este realizat în MatLab. Cu ajutorul acestui program au fost rezolvate problemele de optimizare multi-obiectiv MOP1 - MOP4. Problema constă în a găsi frontiera Pareto globală pentru funcția vectorială $\mathbf{f}=(f_1,f_2)$ pentru fiecare dintre problemele MOP1 - MOP4.

Prima problemă numită MOP1 este:

$$f_1(x) = x_1$$

$$f_2(x,g) = g(x) \times \left(1 - \left(\frac{f_1(x)}{g(x)}\right)^{1/2}\right)$$

$$g(x) = 1 + \frac{9}{n-1} \times \left(\sum_{i=2}^{n} x_i\right)$$

Cu restricțiile $0 \le x_i \le 1$, i=1,2,...,6.

Frontiera Pareto globală este ilustrată în figura 3.

A doua problemă numită MOP2 este:

$$f_1(x) = x_1$$

$$f_2(x,g) = g(x) \times \left(1 - \left(\frac{f_1(x)}{g(x)}\right)^{2}\right)$$

$$g(x) = 1 + \frac{9}{n-1} \times \left(\sum_{i=2}^{n} x_i\right)$$

Cu restricțiile $0 \le x_i \le 1$, i=1,2,...,6.

Frontiera Pareto globală este ilustrată în figura 4.

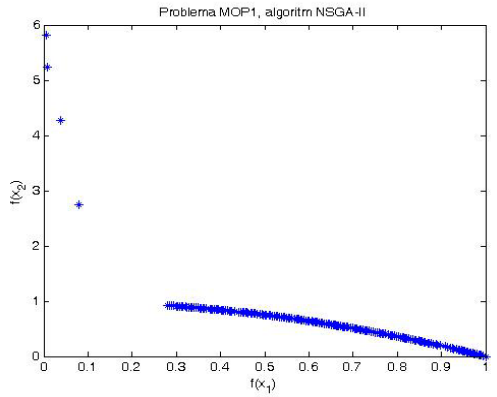

Figura 3. Frontiera Pareto pt. problema MOP1

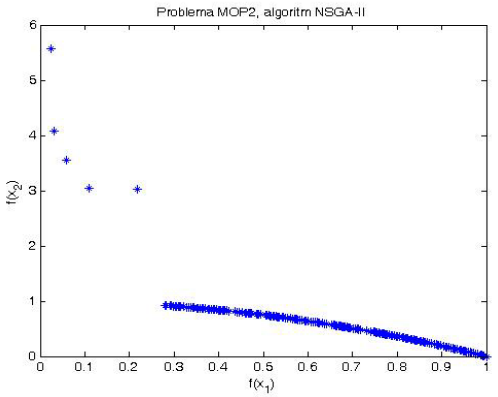

Figura 4. Frontiera Pareto pt. problema MOP2

A treia problemă numită MOP3 este:

$$f_1(x) = x_1$$

$$f_2(x,g) = g(x) \times \left(1 - \left(\frac{f_1(x)}{g(x)}\right)^{1/2} - \frac{f_1(x)}{g(x)} \times \sin(10\pi f_1(x))\right)$$

$$g(x) = 1 + \frac{9}{n-1} \times \left(\sum_{i=2}^{n} x_i\right)$$

Cu restricţiile $0 \le x_i \le 1$, i=1,2,...,6.

Frontiera Pareto globală este ilustrată în figura 5.

A patra problemă numită MOP4 este:

$$f_1(x) = 1 - e^{-4x} \sin^6(6\pi x_1)$$

$$f_2(x,g) = g(x) \times \left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$$

$$g(x) = 1 + 9 \times \left(\sum_{i=1}^{n} \frac{x_i}{9}\right)^{0.25}$$

Cu restricţiile $0 \le x_i \le 1$, i=1,2,...,6.

Frontiera Pareto globală este ilustrată în figura 6.

S-a considerat că numărul de indivizi în populaţie = 200 şi numărul de generaţii = 1000.

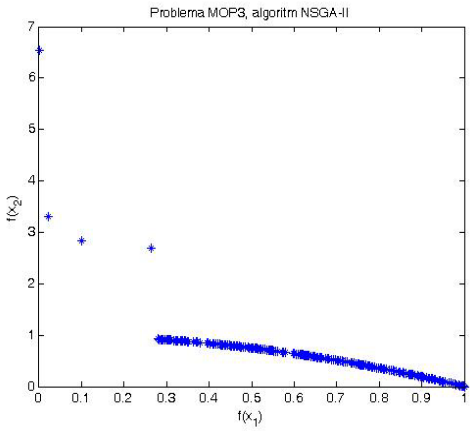

Figura 5. Frontiera Pareto pt. problema MOP3

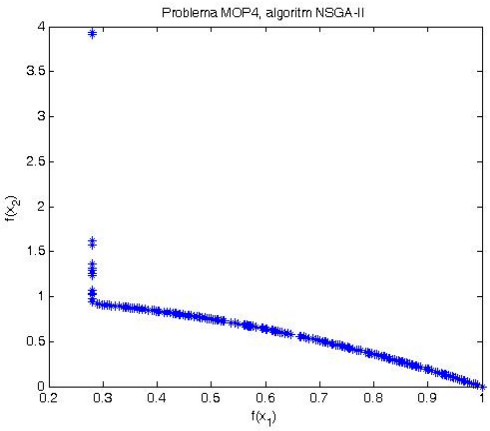

Figura 6. Frontiera Pareto pt. problema MOP4

5. Concluzii

În lucrare s-a realizat o analiză a modurilor de rezolvarea a problemelor de optimizare multi-obiectiv prin algoritmi clasici şi prin algoritmi bazaţi pe Inteligentă Artificială avansată. Domeniul Optimizărilor Multi-obiectiv este extrem de complex şi dificil din punct matematic. El posedă multe sub-domenii puţin cercetate şi probleme remarcabile. Algoritmii bazaţi pe inteligenţă artificială folosiţi pentru rezolvarea problemelor de optimizare multi-obiectiv sunt algoritmii evolutivi.

Algoritmii de inteligenţă artificială au un caracter euristic. Ei sunt aplicaţi de cele mai multe ori la probleme în care abordările cu algoritmi clasici de optimizare au performanţe slabe sau foarte slabe. Aceşti algoritmi sunt eficienţi în sensul că produc soluţii suboptimale într-un interval rezonabil de timp. De multe ori au fost elaborate metode euristice de optimizare care se aplica unor clase largi de probleme de optimizare. Astfel de metode au fost numite meta-euristici. De exemplu algoritmii evoluţionişti sau cei de tip Simulated Annealing, metodele Monte Carlo, căutarea "tabu search" sunt exemple de meta-euristici.

În concluzie pentru a se obţine soluţii bune la problemele de optimizare (multi-obiectiv) este bine să se utilizeze atât algoritmi determinişti clasici cât şi algoritmi bazaţi pe inteligenţă artificială. Există probleme la care cea mai bună alegere pentru rezolvarea lor o reprezintă algoritmii clasici cunoscuţi, la fel cum, există probleme care pot fi rezolvate numai prin aplicarea algoritmilor bazaţi pe inteligenţă artificială.

BIBLIOGRAFIE

1. **BÄCK, T.:** Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.
2. **COELLO, C.; LAMONT, G.; VELDHUIZEN, D.:** Evolutionary Algorithms for solving Multi-Objective Problems. Springer, 2007.
3. **DAY, O.; KLEEMAN, M. P.; LAMONT, G.B.:** Multi-Objective fast messy Genetic Algorithm Solving Deception Problems. In 2004 Congress on Evolutionary Computation (CEC'2004), volume 2, Portland, Oregon, USA, IEEE Service Center, June 2004, pp. 1502-1509.
4. **DAY, R. O.; LAMONT, G. B.:** Multiobjective Quadratic Assignment Problem Solved by an Explicit Building Block Search Algorithm – MOMGA-IIa. In G.R. Raidl and J. Gottlieb, editors, Evolutionary Computation in Combinatorial Optimization. 5th European Conference, EvoCOP 2005, , Lausanne, Switzerland,. Springer, Lecture Notes in Computer Science Vol. 3448, March/April 2005, pp 91-100.
5. **DEB, K.; AGRAVAL, S.; PRATAB, A.; MEYARIVAN, T.:** A Fast Elitist Non-Dominated Sorting Genetics Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K.Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, Proceedings of the Parallel Problem Solving from Nature VI Conference, Lecture Notes in Computer Science No. 1917, Springer, Paris, France, 2000, pp. 849-858.
6. **DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T.:** A fast and Elitist multi-objective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182-197, April 2002.
7. **ERICKSON, M.; MAYER, A.; HORN, J.:** The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001, pp. 681-695.
8. **FONSECA, C. M.; FLEMING, P. J.:** Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, Proceedings of the Fifth

International Conference on Genetic Algorithms, San Mateo, California. University of Illinois at Urbana-Champaign, Morgan Kaufman Publishers, 1993, pp. 416-423.

9. **FOURMAN, M. P.:** Compaction of Symbolic Layout using Genetic Algorithms. In J. J. Grefenstette, editor, Genetic Algorithms and their applications: Proceedings of the First International Conference on Genetic Algorithms,. Lawrence Erlbaum, Hillsdale, New Jersey, 1985, pp. 141-153.
10. **HAJELA, P.; LIN, C. Y.:** Genetic search strategies in multicriterion optimal design. Structural Optimization, 4:99-107, 1992.
11. **HORN, J.; NAFPLIOTIS, N.:** Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAI Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.
12. **KITA, H.; YABUMOTO, Y.; MORI, N.; NISHIKAWA, Y.:** Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In H.-M. Voight, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, Parallel problem Solving from Nature – PPSN IV. Springer-Verlag. Lecture Notes in Computer Science No. 1141, berlin, Germany, September, 1996, pp. 504-512.
13. **KNOWLES, J. D.; CORNE, D. W.:** The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. In 1999 Congress on Evolutionary Computation, Washington, D.C., IEEE Service Center, July 1999, pp. 98-105.
14. **KNOWLES, J. D.; CORNE. D. W.:** Approximating the Nondominated Front Using the Pareto Archieved Evolution Strategy. Evolutionary Computation, 8(2):149-172, 2000.
15. **KURSAWE, F. A:** Variant of Evolution Strategies for Vector Optimization. In H.-P.Schewefel and R. Manner, editors, Parallel Problem Solving from Nature. 1st Workshop, PPSN I, ,.. Lecture Notes in Computer Science No.496, Springer-Verlag, Dortmund, Germany, October 1991, pp. 193-197.
16. **LAUMANNS, M.; OCENASEK, J.:** Bayesian Optimization Algorithms for Multi-objective Optimization. In J.J. Merelo Guervos, P. Adamidis, H.-G. Beyer, J.-L. F.V. nas, and H.-P. Schwefel, editors, Parallel Problem Solving from Nature – PPSN VII,. Lecture Notes in Computer Science No. 2439, Springer-Verlag, Granada, Spain, September 2002, pp. 298-307.
17. **MERKLE, L. D.; LAMONT, G. B.:** A Random Function Based Framework for Evolutionary Algorithms. In T. Bäck, editor, Proceedings of the Seventh International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, California, July 1997, pp. 105–112.
18. **OSYCZKA, A.:** Multicriteria optimization for engineering design, Optimization (J. S. Gero, ed.), Academic Press, 1985, pp. 193-227.
19. **OSYCZKA, A.; KUNDU, S.:** A new method to solve generalized multicriteria optimization problems uing the simple genetic algorithm. Structural Optimization, 10:94-99, 1995.
20. **OSYCZKA, A.; KUNDU, S.:** A Genetic Algorithm-Based Multicriteria Optimization Method. In Proceedings of First World Congress of Structural and Multidisciplinary Optimization, Goslar, Elsevier Science, Germany, May 1995, pp. 909-914.
21. **PELIKAN, M.; SASTRY, K.; GOLDBERG, D. E.:** Multiobjective BOA, Clustering, and Scalability. In H.-G. B.et al., editor, 2005 Genetic and Evolutionary Computation Conference (GECCO'2005), volume 1, New York, USA, ACM Press, June 2005, pp. 663-670.
22. **ROSENBERG, R. S.:** Simulation of genetic populations with biochemical properties. PhD thesis, University of Michigan, Ann Arbor, Michigan, USA, 1967.
23. **SCHAFFER, J. D.:** Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. PhD thesis, Vanderbilt University, Nashville, Tennessee, 1984.

24. **SCHAFFER, J. D.:** Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, , Hillsdale, New Jersey, 1985, pp. 93–100.

25. **SRINIVAS, N.; DEB, K.:** Multiobjective Optimization using NonDominated Sorting in Genetic Algorithms. Evolutionary Computation, 2(3):221-248, fall 1994.

26. **VAN VELDHUIZEN, D. A.; LAMONT, G. B.:** Genetic Algorithms, Building Bloks, and Multiobjective Optimization. In A. S. Wu, editor, Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program, Orlando, Florida, July 1999, pp. 125-126.

27. **ZITZLER, E.; THIELE, L.:** Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation, 3(4):257-271, November 1999.

28. **ZITZLER, E.; LAUMANNS, M.; THIELE, L.:** SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, EURIGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial problems, Athens, Greece, 2001, pp. 95-100.

29. **ZYDALLIS, J. B.; VAN VELDHUIZEN, D. A.; LAMONT, G. B.:** A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA-II. În E. Zitzler, K. Deb, L. Thiele, C.A.C. Coello, and D. Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001, pp. 226-240.

30. **ZYDALLIS, J. B.:** Explicit Building-Block Multiobjective Genetic Algorithms: Theory, Analysis, and Development. PhD thesis, Air Force Institute of Technology, Department of the Air Force, Air University, Wtight-Patterson, Air-forceBase, Ohio, USA, 2003.