

Pregledni rad

Prihvaćeno 8. 12. 2003.

JOSIP DVORNIK

Metode rješavanja problema pomoću računala

Computer Methods for Problem Solving

ABSTRACT

There are a lot of computer methods for solving mathematical and other problems. Generally, they can be classified into numerical, symbolic and analytic, as well as heuristic methods which may include "artificial intelligence". For different problems, different approaches are needed. In recent years the development of all the three groups of methods has been very fast. Owing to an increasing number of hybrid methods which combine analytic, numerical and heuristic approach, some very difficult problems have been solved successfully.

Key words: artificial intelligence, heuristic methods, numerical methods, problem solving, symbolic method

MSC 2000: 68T20

Metode rješavanja problema pomoću računala

SAŽETAK

Postoji mnogo metoda za rješavanje matematičkih i ostalih problema uz pomoć računala. Mogu se uglavnom podijeliti na numeričke, simboličke i analitičke te heurističke metode u koje se može uključiti i "umjetna inteligencija". Različiti problemi zahtijevaju i različit pristup rješavanju. U sve tri grupe metoda vidi se velik napredak iz godine u godinu. Postoji i sve više hibridnih metoda, koje kombiniraju analitički, numerički i heuristički pristup i kojima su već uspješno riješeni i neki vrlo teški problemi.

Ključne riječi: heurističke metode, numeričke metode, rješavanje problema, simboličke metode, umjetna inteligencija

nego ikada ranije. Omogućena je suradnja među znanstvenicima koji žive na udaljenim djelovima svijeta a osobno se ni ne poznaju. Dogodile su se velike promjene u pisanju, organiziranju i prelamanju tekstova. Moguća su automatska mjerenja različitih pojava uz gotovo trenutnu obradu rezultata i prikaz na ekranu u dalekom gradu. Grafičke mogućnosti računala su velike a grafički hardver i softver se i dalje brzo razvija, a to je čitaocima ovog časopisa dobro poznato.

U ovom se pregledu ne namjeravamo baviti svim navedenim i nenavedenim primjenama računala u istraživanjima nego ćemo se ustredotočiti na upotrebu računala pri rješavanju matematičkih i ostalih problema te njihove primjene.

Problemi koje rješavamo su vrlo raznovrsni. Neki su od njih dobro definirani i imaju jedno ili nekoliko diskretnih rješenja. U nekima se rješenje može odrediti u prihvatljivom vremenu. Kod nekih drugih problema vrijeme traženja točnog rješenja bi i na najbržem računalu bilo tako dugo da se moramo zadovoljiti najboljim dohvatljivim približnim rješenjem. Ima i problema kod kojih je skup rješenja kontinuiran pa pokušavamo naći najbolje. Ima čak i takvih problema kod kojih je granica između rješenja i ne-rješenja nejasna, pa ih prije svega treba preciznije formulirati.

Navest će se tri grupe metoda koje imaju primjenu u rješavanju problema pomoću računala. To su:

- Numeričke metode
- Simboličke i analitičke metode
- Heurističke metode i "umjetna inteligencija"

Postoje i druge metode, primjerice eksperimentalne, koje se neće spominjati u ovom pregledu.

1 Uvod

Prve su vijesti o elektroničkim računalima bile pune dezinformacija i doživljavale su se kao znanstvena fantastika. No u manje od pola stoljeća računala su postala nezaobilazno pomagalo. Njihovim se mogućnostima još i danas često začudimo. U znanstvenom se radu i tehnici upotrebljavaju na mnogo načina. Informacije su brže dostupne

2 Numeričke metode

Danas se najveća primjena računala u matematici i tehnici odnosi na numeričke metode. One potiču već od početaka ljudskih civilizacija. Računanjem su se bavili matematičari starih naroda: Egipćani, Babilonci, Indijci, Feničani, Kinezi, Azteci, Maje itd. No njihovo računanje se još ne može nazvati numeričkom metodom, jer su se upotrebljavali nedokazani empirijski postupci. Sustavni početak

numeričkih metoda je započeo u antičkoj Grčkoj. Grčki matematičari su uveli pojam algoritma. Od mnogih njihovih algoritama spomenut će se Euklidov algoritam za određivanje najveće zajedničke mjere više brojeva i *Eratostenovo sito*- algoritam za određivanje niza prim brojeva. Arhimed je shvatio pojam limesa i pojam integrala premda ih nije strogo formalizirao - primjerice interpretirao je površinu kruga kao graničnu vrijednost niza površina upisanih i opisanih pravilnih poligona.

Na žalost, Grčki su matematičari (isključujući neke, primjerice Arhimeda koji je ujedno bio i inženjer) imali elitistički stav prema matematici. Primjenu na rješavanje praktičkih problema smatrali su srozavanjem uzvišene znanosti, a numeričke metode manje vrijednim "nužnim zlom". Taj negativan odnos prema numeričkim metodama ni do danas nije posve nestao.

Kasnije je ipak glavni motiv razvoja numeričkih metoda bila primjena na mnoga područja znanosti i tehnike. Od matematičara i fizičara koji su se uz ostalo bavili i tim metodama navodimo nekoliko: Fibonacci, Newton, Fermat, Descartes, Gauss, Euler, Pascal, Laplace, Lagrange, Fourier, Rayleigh, Poincaré, Ljapunov, Courant... Navodimo i neke nematematičare od mnogih koji su dali veliki doprinos: Bairstow (aerodinamičar), Seidel (astronom), Richardson i Lorenz, (meteorolozi), Aitken (statističar), Pareto (ekonomist) ... Njihove su ideje matematičari preuzeli te ih strože formulirali, dokazali i poopćili. Izvorni autori ne bi više prepoznali svoje ideje.

Računala su dala numeričkim metodama novi impuls. Mnoge metode poznate od ranije, znatno su se usavršile i upotrijebile za rješavanje vrlo složenih problema.

Primjerice, iako je Gaussov algoritam eliminacije za rješavanje sustava linearnih jednadžbi bio dobro poznat među matematičarima i inženjerima, za čovjeka oboružanog papirom i olovkom bilo je vrlo teško riješiti sustav od desetak linearnih jednadžbi, a rješenje sustava od dvadeset jednadžbi je bio pravi podvig.

Uz pomoć suvremenih računala ni sustavi od nekoliko milijuna nelinearnih jednadžbi ne predstavljaju nesavladivu prepreku (Ipak nije svejedno o kakvom se tipu nelinearnosti radi. Za razne tipove se upotrebljavaju različite metode.) NASA je pred tridesetak godina najavila do kraja dvadesetog stoljeća numeričko rješenje sustava od oko milijardu jako nelinearnih jednadžbi, nastalih modeliranjem strujanja zraka oko avionskog krila. Nepoznanice su diskretizirane veličine koje opisuju turbulentno strujanje zraka, ali i veličine koje opisuju pomake krila od dinamičkog opterećenja turbulentnim strujanjem. Oscilacije krila i strujanje zraka su međusobno zavisni, pa jednadžbe treba rješavati kao cjeloviti sustav. Ne znamo je li NASA-ino predviđanje ostvareno, ali ako i nije to će se bez sumnje uskoro dogoditi.

Bilo je metoda koje su se razvijale u teoriji u predkompjutorsko vrijeme ali su bile zbog opsežnih proračuna jedva primjenjive. Tek s računalima su mnoge od njih uspješno realizirane. Osobito je veliku primjenu u inženjerstvu doživjela metoda konačnih elemenata (MKE). Ideja je, koliko je poznato, potekla od Couranta, ali se u to vrijeme

mogla primijeniti samo na jednostavne i male školske primjere. Danas se uspješno primjenjuje na sve linearne i nelinearne probleme matematičke fizike s primjenom u tehnici. Teorijskim istraživanjima, numeričkim eksperimentima i inženjerskim iskustvom je utvrđeno da MKE usprkos velikih prednosti ima nedostataka, pa se osim poboljšanja MKE, u novije vrijeme razvijaju i mnoge alternativne metode.

3 Simboličke i analitičke metode

Računala su potakla i renesansu analitičkih i simboličkih metoda. Idejama automatske primjene tih metoda su se bavili već Babbage i Turing prije pojave elektroničkih računala. Babbage je proizveo mehanički kompjutor s kojim se nije moglo baš mnogo računati zbog neprekidnog kvarenja, ali je to ipak bio važan početak. Turing je kasnije izumio matematički model računala - *Turingov stroj*. Iako autor nije ni zamislio da se taj stroj "materijalizira", uz njegovu su pomoć formulirana načela rada računala koja u velikoj mjeri vrijede i danas.

Brzo nakon pojave računala pojavio se programski jezik LISP s kojim su se mogli rješavati simbolički problemi. Ubrzo zatim se pojavio i logički jezik PROLOG, a nakon toga još veliki broj drugih simboličkih i logičkih jezika. LISP je danas još uvijek u intenzivnoj upotrebi, dok je PROLOG u najnovije vrijeme zamijenjen novim još jačim, ali i još apstraktnijim i složenijim logičkim programskim jezikom *Gödel*.

Neće se ovdje nabrajati svi programski jezici kojima se mogu rješavati simbolički problemi, ali treba navesti da danas postoje i vrlo razvijeni matematički paketi, od kojih se ističu *Mathematica*, *Maple* i *MACSYMA*. Ti paketi sadrže veliki broj matematičkih funkcija, algoritama i transformacija. Uz pomoć tih paketa mogu se pojednostaviti matematički izrazi, analitički derivirati, rješavati određeni i neodređeni integrali te obične i parcijalne diferencijalne jednadžbe itd. Ako se neki problem ne može riješiti analitički, bilo zbog toga što je to načelno nemoguće bilo zbog toga što neki postupci još nisu implementirani u programski paket, može se odmah preći na numerički proračun. Prije pojave tih paketa često je važni vremenski zahtjevni dio posla kod izrade magistarskih radova i doktorata iz matematike, prirodnih i tehničkih znanosti sadržavao međusobno množenje ili potenciranje polinoma, supstituciju matematičkih izraza umjesto varijabli u drugim izrazima, deriviranje i integriranje složenih funkcija i slično. Isti je postupak trebalo ponavljati i provjeravati nekoliko puta zbog grešaka. Upotrebom tih paketa to su postale automatske operacije koje se mogu napraviti brzo i bez pogrešaka. Paket *Mathematica* se već dugo upotrebljava u Hrvatskoj.

Iako ne postoji dokaz da nova verzija paketa *Mathematica* može riješiti sve analitički riješive integrale, nije do sada nađen ni jedan protuprimjer, premda su prema navodima autora paketa - firme Wolfram Research, ispitani svi primjeri iz svih poznatih i dostupnih zbirki integrala i

još mnogo integrala - generiranih na računalu deriviranjem složenih izraza i različitim transformacijama, pa im je analitičko rješenje unaprijed poznato. Naravno, kako znamo, za mnoge se integrale može dokazati nemogućnost analitičkog rješenja elementarnim funkcijama. Njih ne može analitički riješiti ni *Mathematica* ni bilo koji drugi paket.

Navedenim se paketima mogu rješavati obične i parcijalne diferencijalne jednadžbe, za koje do sad nije bilo poznato rješenje. Primjerice, u zadnjih desetak godina je otkriveno najmanje dvadesetak do nedavno nepoznatih analitičkih rješenja Plateauovog problema minimalne plohe. Minimalna ploha je oblik koji bi poprimila opna od sapunice razapeta na žicu oblika zadane prostorne krivulje. Rješenja zadovoljavaju Lagrangeovu nelinearnu parcijalnu diferencijalnu jednadžbu. Nije se ni slutilo da postoji toliki broj analitičkih rješenja. Bez računala taj bi posao bio mnogo teži.

Također se u posljednje vrijeme unutar spomenutih paketa razvijaju programi za automatsko dokazivanje matematičkih teorema i postignuti su već mnogi zanimljivi rezultati. Na početku se upisuju aksiomi koje program smije upotrijebiti te hipoteza koju treba dokazati ili oboriti. Korake dokaza računalo ispisuje u jeziku razumljivom matematičarima, pa ih specijalist za konkretno područje može provjeriti. Naravno, događa se da neku tvrdnju program ne može ni dokazati ni oboriti. Iako će napretkom softvera i napretkom matematike kao znanosti takvih "neodlučenih" slučajeva biti sve manje, posve je sigurno da ih neće nikad posve nestati. Gödel je dokazao da postoje neodlučive tvrdnje koje se u načelu ne mogu (i nikada neće moći) matematičkim metodama ni dokazati ni oboriti. Najteže dokaze ipak će još dugo (možda i uvijek) morati provoditi ljudi sa svježim idejama, ali će im računala višestruko ubrzati i olakšati rad.

Do sad je ponovljeno mnogo poznatih dokaza, primjerice teorema iz euklidske planimetrije i stereometrije, a već su u "suradnji" čovjeka i računala dokazani neki do nedavno nedokazanih teških matematičkih teorema.

Mathematica sadrži vrlo djelotvorni programski jezik s funkcijskim, simboličkim, numeričkim, logičkim, grafičkim i ostalim naredbama. Sadrži i naredbe visoke razine, tako da je moguće pisanje vrlo kratkih programa. Treba reći da je za učinkovito programiranje u jeziku *Mathematica* potrebno više znanja iz teorije koja se upotrebljava i više vještine programiranja nego za primjenu proceduralnih kompjutorskih jezika. No trud uloženi u učenje se višestruko isplati.

4 Heurističke metode i "umjetna inteligencija"

Kao treća "grana" razvijaju se heurističke metode i metode tzv. "umjetne inteligencije", koje već sada imaju zanimljivu i vrlo uspješnu primjenu u mnogim područjima tehnike, medicine, ekonomije, biologije, ratnih operacija itd.

Heurističke se metode redovito upotrebljavaju i u svakodnevnom životu. Kad idemo na posao nastojimo odabrati najbolju varijantu: hoćemo li ići tramvajem, automobilom ili pješice? Odlučujemo i kojim je putem najbolje proći. Odluke ćemo donjeti bez proračuna i mjerenja situacije na terenu. Ponekad imamo djelomične podatke. (Primjerice, saznali smo preko radija da je na glavnoj ulici zastoj zbog prometne nesreće.) Pomoću podataka i iskustva od prije nastojat ćemo "od oka" procijeniti optimalno rješenje.

Kad nešto kupujemo odlučit ćemo prema (vizualno procijenjenoj) kvaliteti, cijeni, imenu proizvođača, vlastitom estetskom kriteriju, itd.

Odluke donosimo u svakodnevnom životu pomoću grubih procjena koje ne možemo nazvati "metodama". Ipak ako slični pristup želimo automatizirati za rad na računalu, moramo definirati sustavni postupak koji tada ipak postaje metoda.

Programi ipak još daleko zaostaju za ljudskim mozgom jer moraju raditi po unaprijed definiranim pravilima. Istina je da postoje i programi koji mogu mijenjati svoja pravila, ali samo prema zadanim još složenijim "meta-pravilima" - pravilima za promjenu pravila koja su ipak napisali programeri. Možda čak postoje ili će uskoro postojati i programi koji mogu mijenjati i "meta-pravila" prema zadanim "meta-meta-pravilima". Ne bismo se usudili proricati budućnost i tvrditi da se jednog dana, možda i prije nego što itko očekuje, neće pojaviti doista inteligentni strojevi. A možda inteligencija i nije ništa drugo nego veliki broj hijerarhijskih sve složenijih "meta-meta-meta...pravila" različitih razina? Prema mišljenju Douglasa R. Hofstadtera - koje se čini uvjerljivim, prava umjetna inteligencija bi trebala započeti modeliranjem podsvjesti i definiranjem "najnižeg" programskog jezika na toj razini. U tom bi jeziku trebalo izprogramirati slijedeći "sloj" u kojemu bi se modelirao malo viši stupanj svijesti i definirati viši i apstraktniji programski jezik. Takva koncepcija bi se rekurzivno nastavljala. Svaki bi se "sloj" naslanjao na prethodni, a u njemu bi bio sadržan programski jezik viši od prethodnoga. Nakon desetak slojeva moglo bi se možda već govoriti o pravoj inteligenciji.

Mi ipak nećemo više teoretizirati o modeliranju inteligencije nego o metodama koje se već danas upotrebljavaju. Dat će se pregled tih metoda bez pretenzije na potpunost.

Primjene su npr. automatsko prevodenje, razumijevanje govora i pisma, prepoznavanje slika, orijentacija u prostoru, automatsko upravljanje vozilima i strojevima, medicinska dijagnostika, nalaženje rudnih ležišta iz morfoloških podataka o terenu, optimalizacija raznih sustava, čak i komponiranje glazbe itd.

Općenito, heurističke metode i metode umjetne inteligencije su osobito prikladne za rješavanje "mutnih" problema, koji se ne mogu dobro matematički formulirati, kao što je primjerice već spomenuta medicinska dijagnostika i nalaženje lokacija potencijalnih rudnih ležišta iz podataka o konfiguraciji terena. Upotrebljavaju se i za dobro definirane probleme koji bi se u teoriji mogli rješavati matematičkim metodama, ali se od toga odustaje zbog nedostataka ili nepouzdanosti podataka.

Druga važna klasa problema za koje se intenzivno primjenjuju metode umjetne inteligencije su problemi opterećeni tzv. "kombinatoričkom eksplozijom". To su problemi za koje je poznat egzaktan matematički algoritam, ponekad je čak i jednostavan, ali bi njegova primjena kad je broj nepoznanica velik, zahtijevala neostvarivo mnogo vremena. Poznati je primjer kombinatoričke eksplozije igranje šaha. Nije posebno teško napisati na nekom programskom jeziku algoritam koji bi pretraživanjem svih varijanata do kraja partije egzaktano odredio najbolji potez, ali realizacija takvog algoritma nije moguća u stvarnosti, zbog ogromnog broja varijanata koje bi trebalo istražiti, koje mnogostruko nadmašuju mogućnosti bilo kojeg računala. A treba priznati da šah po broju podataka i mogućih varijanata ne predstavlja naročito veliki problem - postoji samo 64 polja i 32 figure. Problemi koji se pojavljuju u drugim djelatnostima, npr. u automatskom projektiranju, često su veći za mnogo redova veličine, jer imaju neusporedivo više varijabli i njihovih mogućih vrijednosti. U literaturi se navode primjeri koji po formulaciji izgledaju prilično bezazleno, ali bi za njihovo egzaktano rješavanje na nekom budućem računalu mnogo djelotvornijem od današnjih, trebalo mnogostruko više vremena od sadašnje starosti svemira. Naravno da se mora odustati od rješavanja takvog problema pretraživanjem svih mogućnosti.

Metodama umjetne inteligencije nastoji se eliminirati pretraživanja za koja se približnim rezoniranjem može zaključiti da vjerojatno ne sadrže optimum. Tako se dobivaju rješenja u prihvatljivom vremenu, ali se ne može dokazati da su "apsolutno" najbolja. Samo se može tvrditi da su tako dobivena rješenja s vrlo velikom vjerojatnošću mnogo bolja od rješenja koja bismo mogli postići bez primjene tih metoda. Ako opet uzmemo primjer iz šaha, program će odrediti potez koji samo slučajno može biti egzaktano najbolji, ali je vjerojatno najbolji koji se može odrediti na raspoloživom kompjutoru odabranim programom u raspoloživom vremenu. Ni šahovski velemaistor ne može jamčiti da je njegov potez apsolutno najbolji - osim u slučajevima kad je rješenje jednostavno - primjerice kad je moguć forsirani matni napad ili pat te često u završnici kad je jako reduciran broj figura na ploči, pa se ipak mogu analizirati sve mogućnosti.

Treća se klasa odnosi na pojednostavljeno i ubrzano rješavanje vrlo složenih problema. Neki problem bi se mogao jedamput točno riješiti u prihvatljivom, ali dosta dugom vremenu. U procesima optimalizacije takav bi problem trebalo rješavati mnogo puta pa trajanje proračuna postaje ipak neprihvatljivo dugo. Zato prihvaćamo aproksimacije dobivene pojednostavljenim metodama. Zbog složenosti problema i pisanje programa bi dugo trajalo uz mnoge pogreške.

Navest će se nekoliko heurističkih metoda i metoda umjetne inteligencije upotrebljivih za široku klasu problema optimalizacije, bez pretenzije na potpunost:

4.1 Metode Monte Carlo

Umjesto sustavnog pretraživanja čitavog područja definicije problema, pretražuju se samo "slučajno" odabrane

točke u tom području, pa se traži optimum među tim točkama. ("Slučajno" na računalu najčešće znači kvazi-slučajno što se postiže pomoću determinističkih algoritama "generatora slučajnih brojeva". Postoje doduše i hardverski generatori pravih slučajnih brojeva zasnovani na radioaktivnom raspadu nekih materijala ili na električnom iskrenju itd. Slučajni brojevi asociraju na kockarnicu - odatle je i naziv *Monte Carlo*). Često se tako odabrane točke upotrebljavaju kao početni uvjeti za lokalnu matematičku optimalizaciju. Postoji mnogo varijanata metoda *Monte Carlo*. Neke od njih se zasnivaju na slučajnim perturbacijama postignutih lokalnih optimuma u tijeku prethodnog pretraživanja.

Zapravo veliki broj metoda umjetne inteligencije upotrebljava metodu *Monte Carlo* kao jednu od svojih komponenta u trenutku donošenja odluka.

4.2 Ekspertni sustavi

Te se metode služe pravilima kojima oponašaju odluke živog eksperta. Pravila se određuju u suradnji s više živih stručnjaka, a mogu se i automatski određivati na primjerima. Ekspertni sustavi su se pokazali djelotvornima u medicinskoj dijagnostici, geološkim prognozama, generiranju osnovnih varijanata projekata izborom postojećih rješenja iz kataloga i mnogim drugim primjenama.

4.3 Neuralne mreže

Sastoje se od sustava čvorova i veza između njih čime se nastoji simulirati osnovno funkcioniranje mozga - ljudskog ili životinjskog. U većini slučajeva do sada nisu realizirani hardverski nego softverski. To su posebni programski paketi. Razvijaju se i posebna hardverska neuralna računala koja će kad dođu do praktične primjene moći raditi mnogo brže. Neuralne mreže se ne programiraju nego uče na primjerima. "Pokazuju" im se primjeri i rješenja tih primjera, pa neuralna mreža može automatski "zaključiti" po kojim se pravilima mogu odrediti ispravna rješenja. Slijedeće primjere program rješava samostalno pri čemu upotrebljava tako naučena pravila. Područje djelotvorne primjene neuralnih mreža je približno jednako području primjene ekspertnih sustava: medicinska dijagnostika, geološke prognoze, prepoznavanje oblika itd.

4.4 Fuzzy logika

Postoje sustavi s tzv. "mutnom" logikom. Po klasičnoj logici neka smisljena tvrdnja može biti ili istinita ili lažna. Prema mutnoj logici tvrdnja nije posve istinita ni pove lažna, nego joj se može pripisati "stupanj istinitosti". Npr. neki predmet nije ni potpuno svjetao ni potpuno taman nego je 40% svjetao i 60% taman. Uvedena su pravila za baratanje s tom logikom, koja pretstavljaju poopćenje normalne Aristotelove logike i Booleove algebre. Neki se problemi mnogo brže i lakše približno rješavaju pomoću takve formulacije nego pomoću klasične logike. Ta se metoda

pokazala kao vrlo uspješna, posebno za probleme automatskog upravljanja u realnom vremenu, kad je potrebno vrlo brzo odrediti približno rješenje. Primjeri su automatsko određivanje potrebne ekspozicije filma na foto aparatu kad je jedan dio slike u mraku a drugi dobro osvijetljen, ili prilagodavanje rada semafora trenutnoj situaciji u prometu. Kad se primjenjuje na optimalizaciju mutna se logika često primjenjuje u kombinaciji s drugim determinističkim i probabilističkim algoritmima.

4.5 Genetički algoritmi

Genetički algoritmi su inspirirani Darwinovom teorijom prirodne selekcije: U "nultom koraku" se pomoću generatora slučajnih brojeva generira "populacija" potencijalnih rješenja koja zadovoljavaju propisana ograničenja i koja se nazivaju "genomi". Između njih se odabere podskup odabranog broja najboljih. Najbolja rješenja su ona koja imaju najmanju vrijednost funkcije cilja. Svako od tih rješenja je definirano odabranim brojem "gena". U sljedećim koracima se generiraju "potomci" koji od svakog roditelja nasljeđuju dio genetskog koda. Kod toga ne treba doslovno kopirati prirodu pa svako "dijete" može imati i više od dva "roditelja". Također za razliku od prirode, rješenja koja su dovoljno dobra ne stare i ne umiru, nego mogu "živjeti vječno". Unija skupa "roditelja" i skupa "potomaka" čini novi zajednički skup iz kojega se opet odabire podskup najboljih. Osim "križanja" predviđene su i "mutacije" - slučajne promjene vrijednosti nekog gena. To je uvedeno zbog toga da se u konačnom rješenju omogućiti pojava i onih gena koje nema ni jedan od roditelja, a koji mogu dovesti do boljeg rješenja. Radi još veće raznolikosti mogu se uključiti i novi slučajni genomi koji nisu dobiveni modifikacijom starih. Oni se zovu "imigranti". Algoritam nema definiran kraj, nego se postupak može uvijek nastaviti. Ipak nakon izvjesnog broja koraka nova poboljšanja postaju vrlo rijetka i numerički gledano malena, pa se postupak obično prekida po nekom kriteriju. Genetički algoritmi se s velikim uspjehom primjenjuju za mnoge vrste problema, a osobito kombinatoričke optimalizacije, a čak i za probleme diskretizirane kontinuirane optimalizacije.

Kao i u ostalim metodama umjetne inteligencije, još brže se dobivaju dobri rezultati "nečistim" postupkom u kojem se genetski algoritam kombinira s nekom od klasičnih tehnika optimalizacije: Svako od rješenja dobiveno u tijeku genetskog algoritma postaje početni uvjet za lokalni postupak klasične matematičke optimalizacije, primjerice gradijentnom metodom.

Lokalni optimumi dobiveni tim postupkom ponovo ulaze u novi korak genetskog algoritma. Dakle "djeca" se generiraju "križanjem roditeljskih genoma", mutacijama i matematičkom optimalizacijom.

4.6 Evolucijsko programiranje

Zanimljivo je da se postupak sličan genetskom algoritmu može primijeniti i na izradu kompjutorskih programa. "Geni" u tom programu su različite programske naredbe.

Na početku se generiraju slučajni nizovi naredbi uz provjeru sintakse - programi. Program koji za više zadanih setova ulaznih podataka dobiva više ispravnih rezultata ocjenjuje se kao bolji. Među programima se provode postupci "križanja" i "mutacije" te se odabire skup najboljih koji preživljavaju.

Istovremeno se provodi slična simulacija prirodne selekcije među setovima ulaznih podataka. Najbolji set ulaznih podataka je onaj koji je najkritičniji, odnosno onaj na kojemu najviše programa daje pogrešne rezultate.

Ako neki program zadovoljava sve takve evoluirane setove podataka, znači da prolazi test.

Tako u toku "evolucije" programi zadovoljavaju sve više sve strožih testova. Na kraju procesa preostaju samo programi koji su ispravno riješili sve testove, pa se konačni pobjednik određuje između njih po nekom drugom kriteriju, npr. brzini. Naravno, nikada nije posve sigurno da ne postoji neki neotkriveni test na kojemu bi i taj pobjednik "pao na ispitu". No i programima koje su napisali ljudi se ponekad i nakon više godina ispravnog rada pojavi pogrešno rješenje za neki skup ulaznih podataka.

Vrijeme evolucijskog programiranja je za neke tipove problema mnogo kraće nego pisanje konvencionalnog programa koji rade programeri, ali su programi gotovo nečitljivi za ljude.

4.7 Simulirano kaljenje

Taj je algoritam inspiriran poboljšanjem svojstava metala preslaganjem atoma na visokoj temperaturi za vrijeme kaljenja. Počevši od slučajnog ili drugim metodama nađenog inicijalnog rješenja pokušava se slučajnim varijacijama tog rješenja naći bolje. Pri tome se dopušta i pogoršanje u pojedinim koracima, da bi se izišlo iz lokalnog minimuma. U tijeku postupka se postepeno snižuje "temperatura" (što znači da se slučajne varijacije smanjuju). Matematičkim riječnikom simulirano kaljenje se može opisati kao nestacionarni Markovljevi lanac u diskretnom vremenu. Najnovije varijante algoritma automatski popravljaju parametre optimalizacije za vrijeme proračuna prema "iskustvu" iz dosadašnjeg tijeka procesa.

4.8 Tabu algoritam

Jednostavni algoritmi probabilističkog i heurističkog pretraživanja dopustivog područja (Monte Carlo, simulirano kaljenje itd.) često ne konvergiraju ili sporo konvergiraju zbog višestrukog pretraživanja već pretraženih dijelova dopustivog područja te se pojavljuju beskonačni ciklusi. Tabu algoritam sprema podatke o povjesti već obavljenog pretraživanja, pa ne dopušta ponovno posjećivanje istih dijelova područja. Na taj se način istražuje uvijek novi dio područja, zbog čega raste vjerojatnost nalaženja globalnog optimuma. Taj se algoritam najčešće kombinira s drugim metodama.

4.9 Mravlja kolonija

Mravi, termiti, pčele i ose imaju svojstvo tzv. "kolektivne inteligencije". Zadržimo se na primjeru mrava. Mravi svakog dana izlaze iz mravinjaka u potrazi za hranom. Prva skupina mrava su "izviđači". Oni lutaju nasumce i ostavljaju kemijski "feromonski trag" po kojemu ih ostali mogu slijediti. Onaj mrav koji prvi slučajno nađe hranu vraća se u mravinjak udvostručujući svoj trag. Ostali mravi kreću po tom tragu i dodatno ga pojačavaju svojim tragom. Oni mravi iz prve skupine koji nisu našli hranu ili su ju našli daleko od mravinjaka vraćaju se kasnije. Njihov je trag slabiji jer po njemu još nisu išli drugi mravi. S vremenom sve više mrava ide po najjačem tragu do najboljeg nalazišta hrane. Neki od njih u međuvremenu otkriju kraći put do cilja koji po istom algoritmu prihvaćaju i ostali. Tragovi s vremenom isparuju, pa se lošiji putevi kojima ide manje mrava postepeno "zaboravljaju". Tako kolonija mrava zajednički nalazi optimalni put.

Ovaj algoritam je najprije vrlo uspješno simuliran na računalu. Kasnije je i usavršen napuštajući analogiju s mravljom kolonijom, jer kopiranje mrava nije bilo cilj. U najnovije se vrijeme poboljšava u kombinaciji s genetskim algoritmom i sa strogim matematičkim metodama.

4.10 Celularni automati i metoda perkolacije

Često se različite pojave na kontinuumu ili na nepravilnoj mreži mogu dobro aproksimirati idealiziranim diskretnim pojavama na pravilnoj mreži. Mreža može biti kvadratna, ali i općenito romboidna, trokutna i šesterokutna. Pojava se opisuje u diskretnim vremenskim koracima, a prelaz s jednog koraka na drugi je definiran jednostavnim pravilima, koja mogu biti deterministička ili probabilistička. Na taj se način može modelirati razmnožavanje bakterija na hranjivoj podlozi, pritisci u zrnatom materijalu, širenje šumskih požara, širenje epidemija raznih bolesti, procjeđivanje vode kroz sustav pukotina, difuzija itd.

4.11 Čovjek u petlji

Mnoge od navedenih i drugih metoda optimalizacije se mogu još više unaprijediti i ubrzati, ako se omogući da čovjek donosi odluku na kritičnim mjestima u algoritmu. Stručnjak može iskoristiti svoju (prirodnu) inteligenciju, znanje, iskustvo i intuiciju da spriječi pretraživanje onih djelova područja u kojima je mala vjerojatnost postizanja pravog rješenja. Ljudska odluka najčešće mnogostruko ubrzava konvergenciju pri određivanju približnog optimuma. Neki programi u kritičnim trenucima postavljaju pitanja stručnjaku, a njegov odgovor usmjeruje nastavak pretraživanja.

Treba na kraju istaknuti da se metode umjetne inteligencije nažalost vrlo često (zlo)upotrebljavaju za probleme koje bi bilo mnogo lakše riješiti bez njih. To se često radi čak i (ne)namjerno. Neki zadatak koji se može lako i brzo riješiti poznatim determinističkim matematičkim metodama postaje mnogo nerazumljiviji i "znanstveniji" ako se rješava

pomoću "umjetne inteligencije". Nije čak uopće važno jesu li rezultati ispravni. Često se to radi pomoću nekog softverskog paketa za onu metodu umjetne inteligencije koja je trenutno u modi, a koju korisnik najvjerojatnije ni ne razumije i tako se proizvede "izvorni znanstveni rad". Ima recenzenata koji ne žele priznati da nikad nisu ni čuli za novu metodu, pa im je najjednostavnije prihvatiti takav rad. Neki su autori čak metodama umjetne inteligencije rješavali sustave algebarskih jednadžbi, što je jedan od problema najprikladnijih za standardne determinističke matematičke postupke a ujedno i izrazito neprikladnih za primjenu umjetne inteligencije.

5 Zaključak

U različitim primjenama ima mnogo bitno različitih tipova problema. Metode rješavanja su vrlo raznovrsne. Za svaki je tip problema prikladna različita metoda. Ako mislimo na rješavanje na računalu metode možemo podijeliti na numeričke, simboličke-analitičke te heurističke i metode umjetne inteligencije. Unutar svake od spomenutih klasa je veliki broj metoda, koje se nisu mogle pojedinačno analizirati. Sadšnji je trend kombiniranje više tipova metoda, jer se time najdjelotvornije pronalaze dobra rješenja.

Literatura

- [1] ARANTES E OLIVEIRA, E.R., BENTO, J., "The Sense of Progress in Structural Engineering", Development of Knowledge-Based Systems for Engineering, Springer-Velag, New York, Wien, 1998.
- [2] DVORNIK, J., "Razvoj matematičkih modela inženjerskih problema", GRAĐEVINAR 42(1990)12,507-516
- [3] DVORNIK, J., "Numeričke, simboličke i heurističke metode", GRAĐEVINAR 55(2003)10, 575-582 37-58
- [4] HOFSTADTER, D. R., *Gödel, Esher, Bach: An eternal golden braid*, Penguin Books, 1979.
- [5] MOREY, C., SCALES, J., VAN VLECK, E. S., "A Feedback Algorithm for Determining Search Parameters for Monte Carlo Optimization", Journal of Computational Physics 146(1998), 263-281
- [6] O'SHEA, T., EISENSTADT, M., *Artificial Intelligence*, Harper & Row, 1984.

Josip Dvornik

Građevinski fakultet Sveučilišta u Zagrebu

Kačićeva 26, 10000 Zagreb

e-mail: dvornik@grad.hr