

MODELJUĆI SISTEMI U OPTIMIZACIJI

Mr Mohamed Abu Goben, dipl. inž .
Otvoreni univerzitet Al Kude, Gaza, Palestina

Prof. dr Slobodan Krčevinac, dipl. inž.
Fakultet organizacionih nauka, Beograd

Prof. dr Mirko Vujošević, dipl. inž.
Fakultet organizacionih nauka, Beograd

U ovom radu se opisuju modeljući sistemi kao savremena softverska podrška za rešavanje problema operacionih istraživanja. Ovi sistemi obuhvataju: modeljuće jezike, solvere za rešavanje modela, baze podataka, interfejs prema korisniku i interfejs između pojedinih elemenata sistema. Modeljući sistemi su postali deo savremene sajber infrastrukture i na taj način omogućavaju široku primenu metoda operacionih istraživanja. Rešavanje optimizacionih problema biće uskoro mogućnost velikog broja korisnika, a ne samo specijalista iz oblasti matematičkog programiranja i operacionih istraživanja.

Ključne reči: optimizacija, modeliranje, operaciona istraživanja

UVOD

Rešavanje problema planiranja i upravljanja u organizacionim sistemima na način koji je najbolji mogući, danas nije samo tehnološka prednost onih firmi koje su to postigle u svom poslovanju, već sve više i uslov opstanka na globalnom konkurentskom tržištu. Takva iskustva daju snažne podsticaje razvoju savremene računarske podrške procesima odlučivanja u okvirima informacionih tehnologija, kao i razvoju teorije i novih algoritama u okvirima matematike i operacionih istraživanja. Primene metoda operacionih istraživanja za nalaženje optimalnih rešenja u poslovanju, postaju svakodnevna praksa u razvijenim ekonomijama znanja.

Pristup operacionih istraživanja rešavanju problema optimalnog odlučivanja obuhvata, pored ostalih, sledeće osnovne elemente /Krčevinac i dr, 2007/:

- razvoj adekvatnog, po pravilu, matematičkog modela koji opisuje realni sistem i formuliše problem odlučivanja,
- obezbeđenje i korišćenje potrebnih relevantnih podataka,
- razvoj i primenu efikasnog algoritma za nalaženje optimalnog rešenja modela
- prezentacija rezultata modela korisniku na način koji mu odgovara

Za svaki od ovih elemenata razvijaju se pogodne softverske podrške kojima se olakšava posao menadžera u procesima odlučivanja i povećava verovatnoća donošenja dobrih odluka. Značajnu ulogu u tome igra informaciona infrastruktura koja obuhvata Internet u najširem smislu. U ovom tekstu se opisuju postojeći rezultati za navedene korake. U drugoj glavi se daje opis problema modeliranja i definicija savremenih modeljućih sistema. Glava Modeljući jezici. sadrži pregled savremenih modeljućih jezika za potrebe operacionih istraživanja, a u glavi oftvori za rešavanje problema – solveri, se navodi nekoliko savremenih solvera. Problemi interfejsa i povezivanja komponenta modeljućih sistema samo su ukratko naznačeni u glavi Povezivanje i optimizacija na Internetu.

MODELOVANJE I MODELJUĆI SISTEMI

Modelovanje obuhvata konceptualizaciju problema odlučivanja i njegovu apstrakciju u kvantitativnom ili kvalitativnom obliku. U slučaju matematičkih modela, to podrazumeva identifikaciju zavisnih i nezavisnih promenljivih modela i jednačina, odnosno nejednačina, koje opisuju njihove veze. U tom postupku, bitno je očuvati odgovarajuću ravnotežu između stepena složenosti modela i predstavljene realnosti. Jednostavni modeli vode ka prostijem rukovanju i bržem nalaženju rešenja ali su i gora predstava

realnog problema i zbog toga mogu biti neprihvatljivi za praktičnu primenu.

Za konceptualizaciju problema odlučivanja, potrebna nam je dovoljno bogata predstava koja treba da na adekvatnom nivou apstrakcije obuhvati sve objekte i ograničenja problema odlučivanja. Ta predstava mora da ima i svojstvo rešivosti, odnosno mogućnost pronalazjenja optimalnog ili zadovoljavajućeg rešenja.

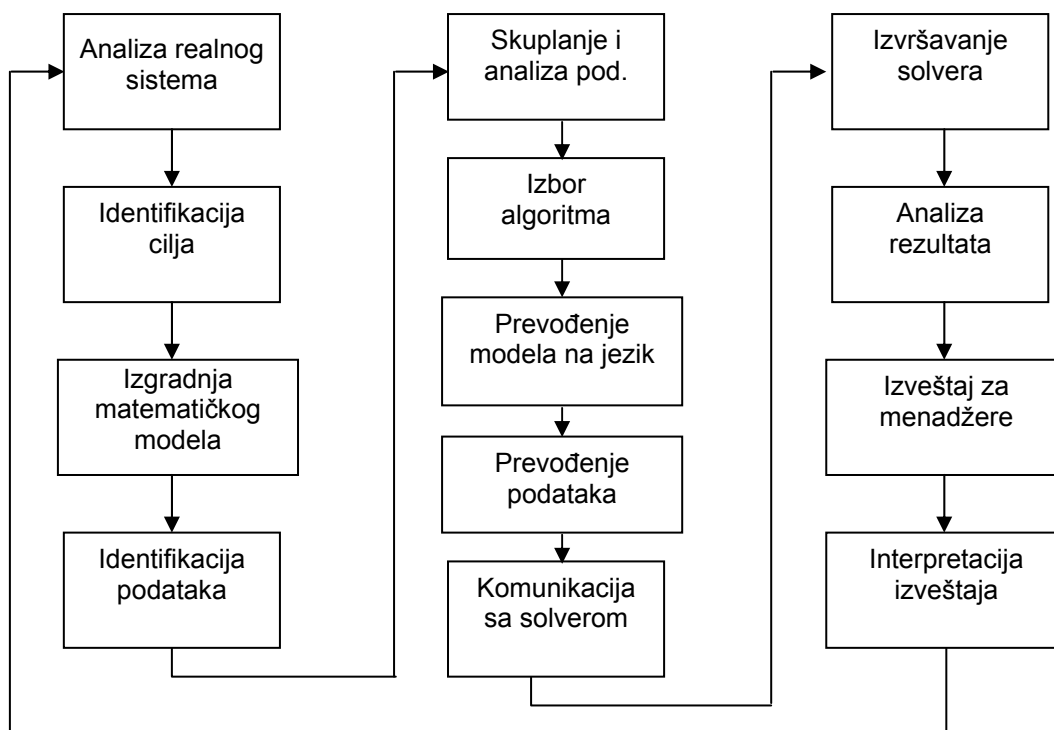
Modelujući sistemi su softverske realizacije koje obuhvataju: modelujuće jezike, solvere za rešavanje modela, baze podataka, interfejs prema korisniku i interfejs između pojedinih elemenata sistema. Savremeni modelujući sistemi omogućavaju efikasnu konceptualizaciju široke klase problema odlučivanja sa kakvim se susreću današnji menadžeri. Štaviše, uključivši i solvere za različite tipove modela, menadžeri su u stanju da brzo i lako dođu do rešenja svojih problema.

Ukoliko se, na primer, menadžer održavanja susretne sa problemom raspoređivanja rad-

nika po smenama, on treba da to prepozna kao pojedinačni slučaj, opštijeg problema raspoređivanja za koji postoje dobro razvijeni algoritmi i njihove softverske imlementacije koje se mogu direktno iskoristiti za rešavanje postojećeg problema odlučivanja. Međutim, za efikasno korišćenje ovih algoritama potrebno je formulisani matematički model prikazati u obliku koji omogućava programsko povezivanje matematičkog modela, podataka potrebnih za njegovo rešavanje i odgovarajućeg solvera koji će se koristiti u rešavanju modela.

Efikasnost modelujućih sistema bitno zavisi od kvaliteta matematičkog modela koji treba da se predstavi u modelujućem jeziku a tek onda uputi solveru na rešavanje. Naime, vreme rada solvera za probleme odlučivanja veće dimenzionalnosti može da bude neprihvatljivo dugačko i tako model postaje praktično neupotrebljiv.

U svetlu korišćenja modelujućih sistema, ciklus modeliranja se sastoji od koraka, prikazanih na slici 1.

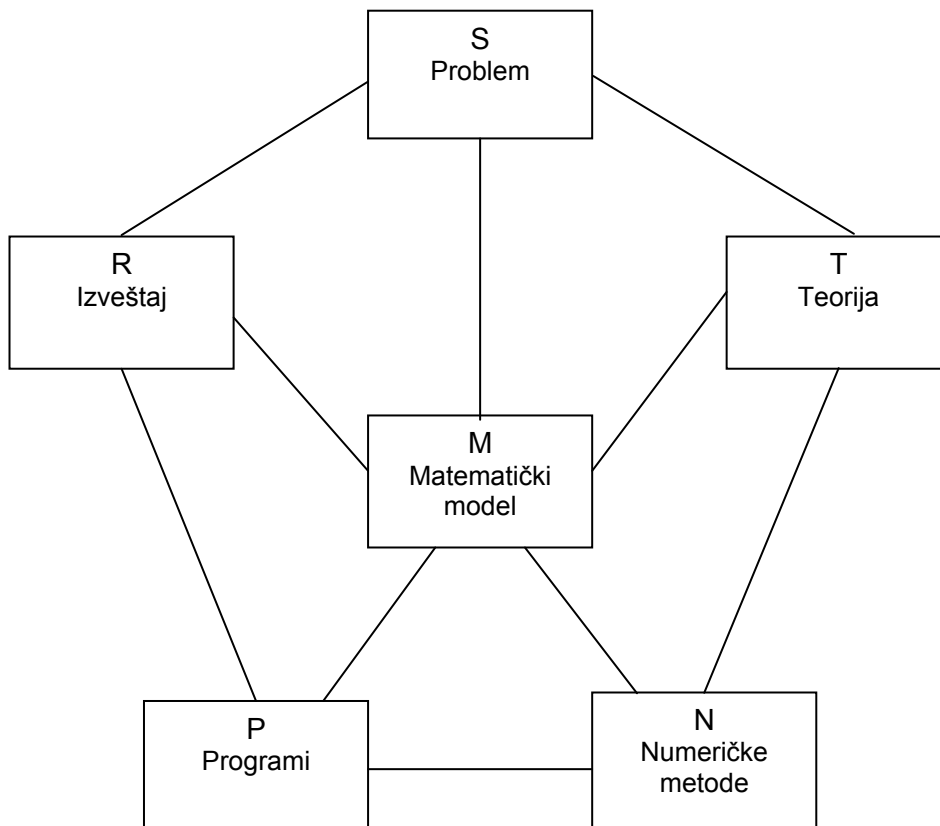


Slika 1. Ciklus modeliranja u okruženju modelujućeg sistema

Između koraka u ciklusu modeliranja postoji mnogo više veza nego što se to može zaključiti iz sekvencijalnog prikaza sa prethodne slike, /Neumaier, 2003/. Bolji uvid u poslove modeliranja se može ostvariti

ilustracijom sa slike 2 koja predstavlja graf u kome čvorovi opisuju informacije koje je potrebno prikupiti, sortirati, analizirati i organizovati. Lukovi grafa prikazuju interakciju i

dvosmerni tok relevantnih informacija između
različitih izvora



Slika 2. Graf informacionih izvora u ciklusu modeliranja

Pretpostavljajući da ulogu modelara problema igra obrazovani operacioni istraživač, semantika čvorova u gornjem grafu je sledeća:

- S. Problem. Problem koji se razmatra dolazi iz realnog sveta i u formulaciji problema potrebna je bliska saradnja sa menadžerima koji su korisnici rešenja za dati problem. Prva prepreka koju je potrebno savladati je ustanovljenje zajedničkog jezika koji omogućava dvosmernu komunikaciju menadžera i operacionog istraživača i minimizira mogućnost nerazumevanja za obe strane. Modelujući jezik bogate semantike je velika pomoć u rešavanju problema komunikacije.
- M. Matematički model. Matematički model je jasna i precizna formulacija problema pisana u modelujućem jeziku. Matematički model obuhvata:

a) upravljačke promenljive, koje predstavljaju različite objekte problema odlučivanja za dati nivo apstrakcije problema,

b) relacije, obično jednačine i nejednačine koje povezuju promenljive, i

c) kriterijumsku funkciju ili funkciju cilja koja svakoj upravljačkoj odluci pridružuje indeks performanse koji treba minimizirati za slučaj da je tipa troškova, ili maksimizirati za slučaj kada su u pitanju koristi.

- T. Teorija. Da bi izgradio koristan model, modelar mora da poznaje teorijsku osnovu problema koji se modelira i, naravno, teoriju matematičkih struktura koje se koriste u izgradnji modela. Izborom prave strukture matematičkog modela može se značajno skratiti vreme rada solvera, odnosno rešavati praktični problemi. To, naročito, važi za modele nelinearnih problema i problema sa celobrojnim promenljivama, jer izborom dobre strategije grananja i ograničavanja, dobrih početnih vrednosti i skaliranjem vrednosti promenljivih može da se znatno utiče na vreme rada solvera. Generalno, dobra struktura modela za mešovito celobrojno programiranje je ona čija relaksacija pomoću linearnog programiranja ima dopustivu oblast koja je konveksna i što bliža najmanjem poliedru koji uključuje sve dopustive tačke polaznog problema. U

praksi to znači, na primer, da su gornje granice promenljivih postavljene što niže.

- N. Numeričke metode. Veliki broj modela od praktičnog interesa nije moguće rešiti analitički, već se moraju tražiti numerička rešenja, odnosno razmotriti raspoloživi numerički algoritmi.
- P. Programi. Softver koji implementira numeričke metode i koji formira takozvani solver, omogućava praktično rešavanje modela na savremenim računarima.
- R. Izveštaji. Sa stanovišta praktične upotrebe modela, veoma je važno da menadžeri dobiju smislene izveštaje koji sadrže informacije koje su neophodne za donošenje odluka i koji su u skladu sa njihovim navikama u strukturiranju informacija. U formiranju izveštaja često je pogodno koristiti različite tehnike vizualizacije podataka. Dobro prezentirani izveštaji o kvalitetnim rezultatima modela omogućavaju sticanje višeg nivoa poverenja menadžera u mogućnosti operacionih istraživanja i njihove primene na realne probleme upravljanja.

MODELJUĆI JEZICI

Razvoj modelujućih jezika je počeo kasnih 70-tih godina pojavom prvog modelujućeg jezika GAMS, /Brooke et al, 1992/. On je omogućio opis problema jezikom koji je blizak matematičkoj notaciji korišćenoj u formiranju modela tako da se ovi jezici nazivaju i algebarski modelujućim jezicima.

Algebarski modelujućim jezicima se mogu smatrati novom paradigmom u programiranju koja kombinuje deklarativnu paradigmu i proceduralnu paradigmu standardnih programskih jezika, /Hurlimann, 1999/. Svi programski jezici (imperativni, funkcionalni i logički) definišu problem na algoritamski, proceduralni način. Drugim rečima, oni ne specificiraju „šta je problem“ već „kako da se reši problem“, zbog čega se i nazivaju algoritamskim ili proceduralnim jezicima. Nasuprot njima, modelujućim jezicima opisuje znanje o problemu, definišu problem putem odgovarajućeg modela i obično ne specificiraju kako se do rešenja dolazi. Zato oni pripadaju klasi takozvanih deklarativnih jezika. Osnovno pitanje, kako opisati šta je problem, u ovim jezicima se rešava formalnom specifikacijom osobina problema.

Za modelujuće jezike koji pripadaju grupi deklarativnih jezika karakteristično je:

- da se problemi predstavljaju na deklarativan način,
- da postoji jasno razdvajanje definicije problema od njegovog rešavanja, i
- da postoji jasno razdvajanje između strukture problema i njegovih podataka.

Opis problema u modelujućem jeziku treba da je nezavisan od solvera koji će se koristiti za rešavanje pošto se za jedan model mogu koristiti različiti solveri. Pored toga, u modelujućim sistemima se sam model i njegovi podaci čuvaju odvojeno u dve različite strukture. Na taj način se izmenom ulaznih podataka može kreirati veći broj instanci jednog modela. Sami podaci se često čuvaju u relacionim bazama podataka ili u unakrsnim tabelama tipa Exel, sa kojima modelujućim jezikom komunicira koristeći ODBC konekciju (Open Database Connectivity).

Formalna specifikacija počinje sa domenom vrednosti X koji je, obično, Dekartov proizvod skupa realnih i skupa celih brojeva. Nad ovim domenom se definiše skup ograničenja putem jednačina, nejednačina i funkcije cilja, koji se zajedno mogu posmatrati kao relacija $R: X \rightarrow \{True, False\}$ koja određuje da li $x \in X$. Kaže se da je x dopustivo rešenje matematičkog modela:

$$M := \{x \in X \mid R(x)\}$$

ukoliko su sva ograničenja zadovoljena, tj. ako je $R(x) = True$. Važno je uočiti da se termin dopustivo rešenje u ovom kontekstu koristi za rešenje koje zadovoljava i ograničenje i obezbeđuje optimalnost funkcije cilja, za razliku od standardne definicije u matematičkom programiranju, gde se termin dopustivo odnosi jedino na zadovoljenje ograničenja. Ovakva specifikacija problema odlučivanja, odnosno matematičkog modela koji nju predstavlja, je osnovna za modelujuće jezike i ima za posledicu odsustvo garancije nalaženja i egzistencije rešenja. Na sreću, postoji dovoljno široka klasa problema od praktičnog interesa (na primer, linearno programiranje) za koje postoje efikasni algoritmi rešavanja realizovani putem solvera – softverskih implementacija algoritama.

Algebarski modelujućim jezicima su projektovani za efikasnu specifikaciju problema optimizacije koji imaju sledeći opšti oblik:

$$\begin{aligned} & \text{Min } f(x) \\ & \text{p.o. } F(x)=0 \\ & G(x) \leq 0 \\ & x \in X \\ & X \subseteq \mathbb{R}^m \times \mathbb{Z}^n \end{aligned}$$

U algebarskom modelujućem jeziku, ovakav model se specificira koristeći sledeće jezičke konstrukcije: skupovi, indeksi, parametri i promenljive. Konceptualno slični entiteti se zajedno grupišu u skup. Pojedinačni entiteti u skupu se referenciraju putem indeksa koji identifikuju elemente skupa. Grupe entiteta (promenljive, ograničenja) se mogu kompaktno predstaviti i koristiti u algebarskim izrazima. Moderni modelujući jezici omogućavaju i specifikaciju nelinearnih modela. Najpoznatiji algebarski modelujući jezici su GAMS, AMPL, LINGO, MPL, MOSEL.

Primer modela u modelujućem jeziku

Ilustrovaćemo primenu modelujućih jezika pomoću jezika AMPL u kome se izraz $\sum x_i$ predstavlja kao sum {i in S} x[i]. Na taj način se dobija formulacija vrlo slična matematičkoj formulaciji i znanje za prevođenje matematičke formulacije u modelujući jezik se svodi na poznavanje sintakse jezika.

Sledeći zadatak koji modelujući sistem treba da reši je prevođenje konkretnog primera, instance problema, u format koji prepoznaje solver odnosno algoritam rešavanja. Ovo se može učiniti razvojem kompaktne notacije indeksiranjem svih skupova i dodavanjem podataka modela. Veoma često, modelujući sistem poseduje i tzv. „presolve“ fazu, koja služi za prethodnu obradu podataka, pre slanja solveru na rešavanje, u cilju što efikasnijeg rada solvera.

Radi ilustracije razmotrimo model kvadratnog programiranja:

$$\begin{aligned} & \min \quad x^T Q x + c^T x \\ & \text{p.o. } Ax \leq b \\ & \quad \|x\| \geq 1 \\ & \quad x \leq [l, u] \end{aligned}$$

sa $N \times N$ matricom Q i $M \times N$ matricom A .

Razvijajući gornji model iz kompaktnog u oblik sa indeksima dobijamo:

$$\min \sum_{i=1}^N \left(\sum_{j=1}^N Q_{ij} x_j + c_i \right) x_i$$

$$\text{p.o. } \sum_{j=1}^n A_{ij} x_j \leq b_i, i = 1, \dots, M$$

$$\sqrt{\sum_{i=1}^N x_i^2} \geq 1$$

$$x_j \in [l_j, u_j], j = 1, \dots, N$$

Poznavajući sintaksu AMPL jezika, ovaj model se može predstaviti kao:

```

### PARAMETERS ###
param N>0 integer;
param M>0 integer;
param c {1..N};
param b {1..M};
param Q {1..N,1..N};
param A {1..M,1..N};
param l {1..N};
param u {1..N};

### VARIABLES ###
var x {1..N};

### OBJECTIVE ###
minimize goal_function:
sum {i in 1..N} (sum {j in 1..N} Q[i,j]*x[j] + c[i]) *
x[i];

### CONSTRAINTS ###
subject to linear_constraints {j in 1..M}:
sum {i in 1..N} A[j,i]*x[i] <= b[j];
norm_constraint: sqrt(sum {j in 1..N} x[j]^2) >= 1;
box_constraints {j in 1..N}: l[j]<=x[j]<=u[j];

##### DATA
#####
data sample.dat;
#####
###
solve; display x;

```

Čitajući gornji listing, mogu se lako identifikovati različiti delovi razvijenog modela, kome su dodati delovi deklaracije

parametara i promenljivih, linija kojom se poziva datoteka sample.dat sa podacima i poslednja linija, koja sadrži jedinu proceduralnu naredbu solve, kojom se poziva solver, i naredba display x kojom se štampa rešenje.

Ostali modelujući jezici u operacionim istraživanjima

Algebarski modelujući jezici nisu pogodni u nekim oblastima primene u kojima je simulacija, a ne optimizacija, osnovna metodologija pristupa problemima. Tako, na primer, modeliranje tehnoloških procesa i njihova simulacija zahteva i drugačiju vrstu koncepata prisutnih u modelujućem jeziku. Zbog toga, je razvijena i grupa modelujućih jezika sa bogatim jezičkim mogućnostima za simulaciju procesa kao što su gPROMS i ASCEND u oblasti hemijskog inženjerstva i EXTEND za proizvodno inženjerstvo, /gPROMS, 2007/, /ASCEND, 2007/, /EXTEND, 2007/.

Proširenje paradigme modelujućih jezika čini i objektno-orijentisan modelujući jezik Modelica, projektovan za inženjersko modeliranje složenih fizičkih sistema koji sadrže mehaničke, električne, elektronske, hidrauličke, termotehničke i slične komponente, /Modelica, 2007/.

Posebno važno mesto, sa stanovišta primene metoda operacionih istraživanja, je grupa modelujućih jezika u klasi logičkog programiranja sa ograničenjima (Constraint Logic Programming – CLP). Ova klasa jezika je namenjena modeliranju teških kombinatornih problema, gde je neophodno da uz deklarativni deo definicije problema postoji i deo koji sadrži algoritamska znanja. Većina algebarskih modelujućih jezika nema mogućnost predstave algoritamskih znanja, koja su isključivo smeštena u solveru. U zavisnosti od korišćenog solvera, modelujući jezik mora da obezbedi i dodatni opis o izvođenju algoritama. Ovaj opis nije deo niti modela niti podataka i različiti solveri zahtevaju različite informacije. Ove informacije su, npr. strategija izbora i pretraživanja, definisanje scenarija u slučaju stohastičkih programa, inicijalna rešenja problema lokalne optimizacije i slično. Kombinatorni problemi često zahtevaju modeliranje vrlo specijalizovanih ograničenja (npr. ograničenje tipa all diff označava da u skupu promenljivih svaka od njih uzima različitu vrednost, ili ograničenje cardinality N koje određuje da tačno N

promenljivih u datom skupu uzima vrednost True). Ovakav tip ograničenja ne postoji u konstrukciji algebarskih modelujućih jezika pošto se uvođenjem specijalnih mešovitih i celobrojnih ograničenja komplikuje struktura modela i čini rešavanje složenijim. Najpoznatiji jezici ove klase su OPL, LPL, AIMMS.

SOFTVERI ZA REŠAVANJE PROBLEMA – SOLVERI

Solveri su softverski alati koji se koriste za rešavanje optimizacionih problema definisanih odgovarajućim matematičkim modelom. Oni su implementacija različitih algoritama pogodnih za rešavanje pojedinih klasa problema ili tipova modela. S obzirom na različitost tipova modela: linearni, nelinearni celobrojni, mešovito-celobrojni i tako dalje, postoji i veliki broj različitih solvera koji su bazirani na različitim algoritmima.

Skoro svi solveri su na neki način dostupni preko Interneta. Mnogi od njih su besplatni ali to ne znači da su beskorisni ili loši. Najveći broj korisnika personalnih računara ne zna da su uz Exel standardno dobili solver kojim se mogu rešavati optimizacioni problemi manjih dimenzija. Međutim, za rešavanje praktičnih problema velikih dimenzija, kada broj upravljačkih promenljivih može da bude reda milion ili desetina miliona, moraju se koristiti komercijalni solveri velikih softverskih kompanija.

Pokrenut je projekat COIN-OR (COmputational Infrastructure for Operations Research) kao neprofitni konzorcijum istraživača iz privrede i univerziteta, koji ima za cilj da poboljša stanje korišćenja solvera u rešavanju praktičnih problema odlučivanja. Taj cilj se ostvaruje promocijom razvoja i korišćenja softvera otvorenog koda za operaciona istraživanja. Između ostalog, ovaj projekat održava i biblioteku softverskih alata koji se mogu koristiti i u izgradnji optimizacionih programa, kao i gotovih softverskih paketa. Tako, ova inicijativa objedinjuje sledeće solvere:

- COIN-LP (COIN-OR LP Solver, open source)
- CPLEX (ILOG, komercijalni)
- dylp (BonsaiG LP Solver, open source)
- FortMP (OptiRisk Systems, komercijalni)
- GLPK (GNU LP Kit, open source)

- COIN-IPOPT (Interior Point Optimizer, open source)
- Mosek (Mosek ApS, komercijalni)
- OSL (IBM, komercijalni)
- SoPlex (Konrad-Zuse-Zentrum fuer Informationstechnik Berflin, besplatan za akademsku upotrebu)
- Volume (COIN-OR, open source)
- XPRESS (Dash Optimization, komercijalni)

Osim navedenih u praksi se koristi još mnogo drugih solvera kao npr. MINOS, MINTO, IpSolve itd.

Efikasnost optimizacionog softvera povećana je u poslednjih deset godina za oko 15 miliona puta /Bixby, 2007/ što znači da se do rešenja dolazi 15 miliona puta brže, odnosno da se uz pomoć savremenih računara i solvera mogu da rešavaju i problemi koji su pre deset godina bili apsolutno nerešivi zbog ograničenih računarskih resursa neophodnih za rešavanje problema. Ovako velikom napretku značajan doprinos je dao razvoj tehnologije, par hiljada puta, ali nekoliko puta veći je doprinos postignut u oblasti razvoja teorije, odnosno matematike i metoda operacionih istraživanja. U budućnosti se očekuje dalji napredak u oba pravca da bi se efikasno rešavali modeli koji će obuhvatiti veće sisteme, više odgovarati realnim situacijama i neminovno biti složeniji od onih koji se danas efikasno rešavaju.

POVEZIVANJE I OPTIMIZACIJA NA INTERNETU

Efikasna primena modelujućih sistema zahteva rešavanje problema kompatibilnosti, interoperabilnosti i komunikacija između elemenata koji formiraju sistem kao i između više modelujućih sistema koji se mogu jednovremeno ili odvojeno koristiti za rešavanje istog problema. Ovo znači da treba standardizovati podatke koji su neophodni u procesima odlučivanja, i interfejs prema bazama podataka koje se koriste.

Pored omogućavanja razmene podataka, potrebno je rešiti i problem razmene modela između različitih modelujućih sistema. Očigledno je da razvoj tog otvorenog standardnog formata mora da bude zasnovan na XML-u ili MathML-u. To bi omogućilo pogodan interfejs i za prezentaciju modela na vebu i rešilo ostale probleme konzistentnosti interfejsa.

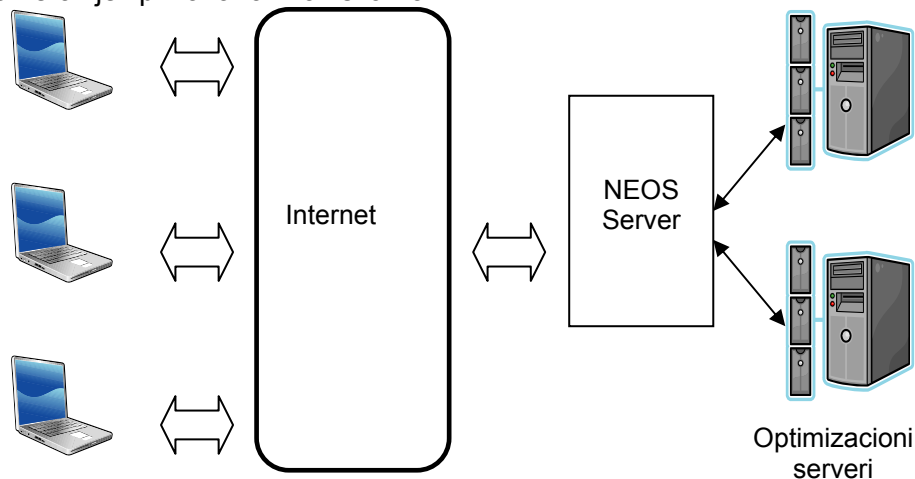
Izuzetno je važan i interfejs prema krajnjem korisniku. Tome se poslednjih godina posvećuje posebna pažnja u okviru takozvane poslovne inteligencije. Ovaj termi označava skup tehnologija koje omogućavaju sakupljanje, sistematizaciju, obradu i prezentaciju podataka potrebnih za efikasno odlučivanje i upravljanje organizacijom /Vujošević, 2006/. Činjenica da interfejs prema krajnjem korisniku predstavlja danas jedno od glavnih interesovanja proizvođača informacionih sistema ukazuje na to da su modelujućim jezici i solveri prilično dobro razvijeni i da su postali dostupan resurs savremenim analitičarima odluka i menadžerima.

Većina modernih modelujućih sistema je raspoloživa u obliku objektna biblioteke bazirane na principima COM (Common Object Model) i CORBA (Common Object Request Broker Architecture), ili u obliku dinamičke ili statičke pozivne biblioteke. Objektna biblioteke su veoma fleksibilan način jer korisnici mogu direktno da pristupaju svim objektima i metodama potrebnim za razvoj njihovih aplikacija. Pored toga, imaju programski pristup internim strukturama korišćenim u modelujućem sistemu, kao što su skupovi, podaci, matrice modela i vektori rešenja. U suštini, korisnici su u stanju da integrišu sve deklarativne funkcije modelujućeg sistema u svoje programe pisane, npr. u C++, Visual Basic i slično, što značajno podiže njihov kvalitet u sistemima za podršku donošenja odluka.

Usluge optimizacije i ostalih metoda operacionih istraživanja se mogu efikasno isporučivati putem veb servisa koji se nalaze na Internet serverima i pružaju korisnicima interfejs kojim oni mogu da pristupe skupu funkcija i metoda tog veb servisa. Skraćenica ASP (Application Service Provider) u ovom kontekstu označava organizacije koje pružaju usluge koristeći Internet infrastrukturu, tj. upravljaju i isporučuju aplikacije većem broju korisnika. Opis servisa je formalno i eksplicitno dat tako da softverski sistemi korisnika mogu sa njim direktno da komuniciraju. Ovo znači da firma korisnik može da iznajmi pristup aplikaciji preko Interneta uz razne modalitete plaćanja za tu uslugu.

Najpoznatiji server za optimizaciju je NEOS, koji omogućava da korisnici svoje probleme podnose koristeći veb formulare, e-mail ili klijentske alate bazirane TCP/IP, [NEOS,

2007]. NEOS raspolaže sa više solvera koji pokrivaju linearno, celobrojno, nelinearno, kvadratno i stohastičko programiranje i podržavaju više modelujućih jezika (GAMS, AMPL,...). Opšta arhitektura NEOS optimizacionog servera je prikazana na slici 3.



Slika 3. Arhitektura NEOS optimizacionog servera

Kada korisnik otkrije servis koji je raspoloživ na Internetu i razume njegov opis i interfejs, on mu može pristupiti, koristeći XML bazirane poruke koje se prosleđuju putem HTTP protokola i na isti način dobiti rezultat. Sledeći jezici i protokoli predstavljaju osnovnu tehnološku bazu veb servisa:

- XML (eXtensible Markup Language) se koristi kao format razmene podataka preko Interneta,
- SOPA (Simple Object Access Protocol) je protokol baziran na XML-u i koristi se za poziv metoda koje obezbeđuje Veb servis,
- WSDL (Web Service Definition Language) je takođe XML baziran jezik, koji se koristi za javnu objavu interfejsa Veb servisa, i
- UDDI (Universal Description, Discovery and Intergration) obezbeđuje globalni registar za oglašavanje, pronalaženje i integraciju Veb servisa.

ZAKLJUČAK

Razvoj metoda operacionih istraživanja, informacionih tehnologija i Interneta su suštinski izmenili pristup rešavanju problema optimizacije, analize i donošenja odluka uz pomoć računara. Danas su donosiocima odluka na raspolaganju potencijali za rešavanje velikih i složenih optimizacionih zadataka koji su do nedavno bili neslućeni i

Ovakva arhitektura omogućava da NEOS može da zadovolji veći broj zahteva distribuirajući ih ka nekom od servera specijalizovanih za pojedine solvere odnosno modelujuće sisteme.

nezamislivi. Da bi se te mogućnosti u praksi iskoristile, potrebno je poznavanje konkretne tehnologije i problema odlučivanja kao i izvesna edukacija u oblasti informatike i operacionih istraživanja. Međutim, ti zahtevi, pogotovo u vezi sa metodama optimizacije, danas su blaži u odnosu na ranije, iako to na prvi pogled može da izgleda protivurečno jer su problemi koji se rešavaju teži a algoritmi optimizacije složeniji. Slično prosečnom savremenom vozaču koji u svom vozilu može da koristi globalni sistem pozicioniranja za navigaciju i određivanje najboljeg puta između dve tačke, a nikada možda nije ni čuo za optimizaciju na grafovima čije rezultate koristi, budući menadžeri će moći da koriste sisteme za podršku odlučivanju koji će im sugerisati najbolje odluke za snalaženje u lavirintima globalnog tržišta. Oni koji to prvi shvate i počnu da koriste imaće najveće komparativne prednosti.

LITERATURA

- /1/ [AIMMS, 2007], Paragon Decision Technology, The Netherlands. www.aimms.com
- /2/ [ASCEND, 2007], <http://ascend.cheme.cmu.edu/>
- /3/ [Brooke et al, 1992], GAMS - A User's Guide (Release 2.25), Boyd & Fraser

- Publishing Company, Danvers, Massachusetts, 1992
- /4/ [Carlisle et al, 2001], Mathematical markup language, version 2.0, 2001. <http://www.w3.org/TR/MathML2/>
- /5/ [COIN-OR, 2007], <http://www.coin-or.org>
- /6/ [Cplex, 2007], Mathematical Programming Optimizer, <http://www.cplex.com>
- /7/ [EXTEND, 2007], <http://www.imaginethatinc.com/>
- /8/ [Foster et al, 2001], "The Anatomy of the Grid: Enabling Scalable virtual Organizations", International Journal of Supercomputer Applications", 15, No.3, 2001
- /9/ [Fourer, 1993], AMPL, A Modeling Language for Mathematical Programming, Danvers, MA: Boyd and Fraser Publishing, 1993
- /10/ [Geoffrion, 2001], „Prospects for Operations Research in the E-Business Era, Interfaces 31, No.2, 2001
- /11/ [GLPK, 2007], GNU Linear Programming Kit , www.gnu.org/software/glpk
- /12/ [gPROMS, 2007], <http://www.psenterprise.com/gproms/index.html>
- /13/ [Hurlimann, 1999], „Mathematical Modeling and Optimization”, An Essay for the Design of Computer-Based Modeling Tools, Volume 31 of Applied Optimization, Kluwer Academic Publishers, Dordrecht, 1999
- /14/ [LPL, 2002] , "A mathematical modeling language, an introduction", Department of Informatics, University of Fribourg, 2002
- /15/ [MINTO, 2007], Mixed INTeger Optimizer, <http://coral.ie.lehigh.edu/minto>
- /16/ [Modelica, 2007], <http://www.modelica.org/>
- /17/ [NEOS, 2007], <http://www.neos.mcs.anl.gov/neos>
- /18/ [NSF, 2003], "Revolutionizing Science and Engineering Through Cyberinfrastructure", <http://www.cise.nsf.gov/>
- /19/ [NSF, 2007], "Cyberinfrastructure Vision for 21st Century Discovery", National Science Foundation Cyberinfrastructure Council, March 2007
- /20/ [Neumaier, 2003], „Mathematical model building" In J. Kallrath, editor, Modeling Languages in Mathematical Optimization, Kluwer, Dordrecht, 2003
- /21/ [OPL, 2007], Optimization Programming Language, <http://www.ilog.com>
- /22/ [OptML, 2000], Halldorsson et al, "A modeling interface to non-linear programming solvers an instance: xMPS, the extended MPS format," Technical report, Carnegie Mellon University and Maximal Software
- /23/ [SNOML, 2001], Lopes et al, "An XML-based Format for Communicating Optimization Problems," <http://senna.iems.northwestern.edu/xml/presentations/LopesFourer>
- /24/ [Ven et al, 2007], "Stimulating Collaborative Development in Operations Research with libOR", Proceedings of the First International Conference on Open Source Systems,
- /25/ Marco Scotto and Giancarlo Succi (Eds.), Genova, 2005
- /26/ [XPress-MP, 2007], Dash Optimization Ltd, England, www.dashoptimization.com
- /27/ [Bixbi, 2007], Robert Bixbi, "From Planning to Operations", Presentation during Closing session of EURO XXII, Prague, July 2007.
- /28/ [Vujošević, 2006], Dušan Vujošević, Upotreba poslovne inteligencije u podizanju performansi organizacije, magistarski rad, FON, Beograd, 2006.

MODELING SYSTEMS AND OPTIMIZATION

The term modeling system, mainly used in the field of optimization, may be considered as the synonym for modern decision support systems. This system consists of modeling

languages, solvers for finding optimal solution of the model, data bases and user interface and communication infrastructure for the elements of the system. The modeling systems enable a wide application of operations research methods and techniques in an easy and friendly way. Solving

optimization problems will be in the future more available for a larger number of users, not the privilege of mathematical programming specialists or operations research professionals.

Key words: optimization, modeling, operations research

EFNMS **Euromaintenance 2008**

Organised by **BEMAS**  in Brussels, April 8th-10th 2008

We are not going to  you to **Belgium** with



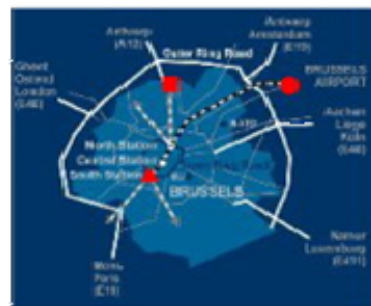
but **convince** you to participate
at an **ambitious** conference on **maintenance**

- Target: at least 500 participants
- 75 speakers – 12 half-day workshops
- Dedicated program per industry / speciality
- 45 minute sessions for efficient knowledge transfer
- Practical case studies presented by leading Maintenance Managers throughout Europe

in the **heart** of Western European industry



The central hub in the Western European pipeline network



Easy access by road, high speed train and airplane

