OPTIMIZACIJA REDUNDANSE

UVOD

Pouzdan rad i visoka raspoloživost, tj sposobnost uređaja, mašine ili tehničkih sistema kao celine da u proizvoljno izabranom trenutku vremena ili u zadatom vremenskom intervalu dobro radi, su karakteristike kojima se posvećuje velika pažnja. Ove karakteristike nisu podjednako važne za sve sisteme, niti je svejedno kakvu funkciju posmatrani sistem obavlja. Postoje sistemi koji jednostavno ne smeju otkazati, tj. sistemi koji moraju biti apsolutno pouzdani, ali isto tako i oni kojima pouzdanost i raspoloživost nisu najvažnije karakteristike.

Kada je reč o pouzdanosti i raspoloživosti sistema bilo koje prirode, intuitivno i iskustveno se zaključuju dve stvari:

(1) Ove karakteristike postaju naročito važne za složenije sisteme sa većim brojem sastavnih delova - komponenata, jer sa porastom broja komponenata raste verovatnoća da će neka od njih otkazati,a time i potencijalna opasnost da će sistem prestati da radi.

(2) Povećanje pouzdanosti sistema košta, što će reći da pouzdaniji sistem po prirodi stvari mora biti skuplji, nego njemu, po funkciji, ekvivalentni, manje pouzdan sistem.

Karakteristike pouzdanost i raspoloživost sistema dolaze na inženjersku scenu sredinom prošlog veka. Jednostavno iskazano, pouzdanost je verovatnoća uspešnog rada sistema do određenog vremenskog trenutka, dok je raspoloživost verovatnoća da će sistem raditi u proizvoljno odabranom trenutku.

Ne može se sa sigurnišću tvrditi da je pojava složenih elektronskih računskih mašina, sa desetinama hiljada osetljivih sastavnih delova, presudno delovala da od projektanata sistema zahtevaju pozdan rad i visoku raspoloživost, jer su i pre njihove pojave postojali i te kako složeni sistemi u vazduhoplovstvu, pomorstvu, komunikacijama, naoružanju, itd., koji su sa gledišta pouzdanosti morali biti jako dobri. Ipak, ideja o izgradnji pozdanih sistema dolazi od strane Von Neumann-a 1956. godine, a izrečena u kontekstu računarstva i teorije automata: “U konstrisanju složenih računskih sistema nepouzdanost komponenata trebalo bi da se prevazišle ne putem ugradnje pouzdanijih komponenata, već njihovim organizovanjem na takav način da pouzdanost celog računskog sistema bude veća nego pouzdanost njegovih delova“.

U opštem slučaju, povišenje pouzdanosti i raspoloživosti sistema ostvaruje se na tri principijelno različita načina:

(1) U fazi projektovanja sistema treba izbegavati nepotrebno složene konfiguracije, truditi se da se tražene funkcije ostvare jednostavno, a za sastavne delove sistema upotrebiti pouzdane elemente.

(2) U fazi eksploatacije sistema treba organizovati kvalitetno održavanje: (a) preventivno, sa ciljem sprečavanja pojave otkaza izvršavanjem podešavanja ili zamene sumnjivih i dotrajalih komponenata, bilo zbog lošeg stanja u kome se one nalaze ili posle tzv. Iskorišćenog resursa, na primer, posle određenog broja časova rada ili broja startovanja, posle završene misije, itd., (b) korektivno, sa ciljem brze detekcije otkaza u sistemu i otklanjanja neispravnosti efikasnom popravkom ili zamenom otkazalih komponenata sistema novim.

(3) U fazama projektovanja i eksploatacije treba koristiti razne tipove redundanse, dupliranje, tripliranje ili multipliciranje manje pouzdanih komponenata sistema, delova sistema, pa čak i celog sistema.

Sistem u kome je u cilju povećanja pouzdanosti i raspoloživosti upotrebljena redundansa naziva se sistem sa redundansom ili redundantni sistem. Odabrati redundansu za jedan sistem se pojavljuje kao projektantski zadatak ili kao zadatak korisnika sistema u fazi eksploatacije. U oba slučaja može se težiti idejama optimizacije i postaviti cilj: *optimizirati redundansu*.

Redundantni sistem čine osnovni delovi (komponente, podsistemi, sistemi) i njima pridruženi redundantni delovi (komponente, podsistemi, sistemi). Ako su osnovni delovi ispravni, sistem uspešno funkcioniše. Pridruženi ili dodatni delovi koji nisu neophodni za funkcionisanje sistema sve dok su njegovi osnovni delovi ispravni zovu se redundantni.

Redundantni delovi koji se dodaju tako da paralelno izvršavaju istu funkciju u sistemu kao i osnovni delovi, nazivaju se paralelna ili aktivna redundansa. Multipliciranje pojedinih delova sistema tako da paralelno i nezavisno izvšavaju istu funkciju u sistemu nekada se ne može lako ostvariti,a nekada je ekonomski neopravdano. Tada se zamisao povećanja pouzdanosti sistema ostvaruje korišćenjem redundanse na taj način što pored osnovnog dela u sistemu postoji i rezervni koji se u slučaju otkaza osnovnog dela automatski ili ručnom zamenom uključuje ili postavlja na mesto otkazalog i zove se pasivna redundansa ili rezervni deo.

O različitim pokazateljima pouzdanosti bilo redundantnih sistema sa obnavljanjem, ili sistema bez obnavljanja, može se naći u materijalu sa časa ili u knjigama koje ima biblioteka Saobraćajnog fakulteta: Elementi teorije pouzdanosti i teorije obnavljanja tehničkih sistema (S. Vukadinović, D. Teodorović, Privredni pregled, 1979.), Optimizacija redundantnih sistema (R. Petrović, M. Vujošević, D. Petrović, Saobraćajni fakultet, 1993.).

U zadatku optimizacije redundanse, biraju se i matematički formulišu različiti kriterijumi i različite funkcije ograničenja. U razvoju modela koriste se, pored ostalih, sledeće opšte pretpostavke:

(1) Sistem koji se posmatra je koherentan i sastoji se od I podsistema; veze između podsistema ne mogu se menjati (u posmatranom periodu sistem ne menja strukturu).

(2) Podsistem se sastoji od jedne ili više identičnih komponenata.

(3) Ispravan/neispravan su dovoljan opis za svaku komponentu, podsistem ili celi sistem.

(4) Podsistemi su statistički uzajamno nezavisni. Takođe su statistički uzajamno nezavisne i komponente u okviru podsistema. Sve komponente u okviru jednog podsistema imaju isti intenzitet otkaza, bilo da je u pitanju aktivna ili pasivna redundansa

(5) Količina potrebnih resursa za celi sistem je aditivna funkcija po podsistemima.

Zadatak izbora optimalne redundanse jeste određivanje vektora $\overline{x} = (x_1, ..., x_I)$, čiji element x_i predstavlja boj redundantnih komponenata u podsistemu i, i=1, ..., I, tako da izabrani kriterijum dobije ekstremnu vrednost (maksimalnu ili minimalnu u zavisnosti od kriterijuma), uz istovremeno zadovoljenje postojećih ograničenja u odnosu na raspoložive resurse. Slede neki primeri optimizacionih zadataka.

Zadatak 1. Zadatak maksimizacije pouzdanosti sistema koji je serijska veza podsistema, sa redundantnim komponentama, aktivnim ili pasivnim, uz jedno linearno ograničenje na ukupna raspoloživa novčana sredstva (ukupni troškovi)

Odrediti $\overline{x} = (x_1, ..., x_I)$ koje maksimizira pouzdanost sistema R_S

$$R_S = \prod_{i=1}^{I} R_i(x_i)$$

pri ograničenjima

$$\sum_{i=1}^{I} c_i x_i \le C$$

$$x_i \in N_0 = \{0,1,2,...\}$$

gde je:

c_i – jedinični trošak za redundantnu komponentu

C – maksimalno dozvoljeni trošak za ukupnu redundansu.

Zadatak 2. Zadatak maksimizacije pouzdanosti sistema koji je serijska veza podsistema, sa redundantnim komponentama, aktivnim ili pasivnim, uz više linearnih ograničenja.

Odrediti $\overline{x} = (x_1, ..., x_I)$ koje maksimizira pouzdanost sistema R_S

$$R_S = \prod_{i=1}^{I} R_i(x_i)$$

pri ograničenjima

$$\sum_{i=1}^{I} c_i x_i \le C$$
$$\sum_{i=1}^{I} w_i x_i \le W$$
$$\sum_{i=1}^{I} v_i x_i \le V$$

$$x_i \in N_0=\{0,1,2,...\}$$

gde je:
c_i – jedinični trošak za redundantnu komponentu
C – maksimalno dozvoljeni trošak za ukupnu redundansu
w_i – jedinična težina redundantne komponente
W – maksimalna dozvoljena težina ukupne redundanse
V_i – jedinična zapremina redundantne komponente
V - maksimalna dozvoljena zapremina ukupne redundanse.

Zadatak 3. Zadatak minimizacije cene sistema koji je serijska veza podsistema, sa redundantnim komponentama, aktivnim ili pasivnim, uz ograničenje da pouzdanost celog redundantnog sistema ne bude niža od neke zahtevane R_z.

Odrediti $\overline{x} = (x_1,...,x_I)$ koje minimizira ukupnu cenu redundanse C

$$C = \sum_{i=1}^{I} c_i x_i$$

pri ograničenjima

$$R_S = \prod_{i=1}^{I} R_i(x_i) \ge R_z$$
$$x_i \in N_0=\{0,1,2,...\}$$

Ovom zadatku mogu se pridružiti još neka ograničenja.

Slede metode koje se koriste za rešavanje ovako ili slično formulisanih zadataka optimizacije redundanse.

METODA JEDINIČNOG PRIRAŠTAJA KRITERIJUMSKE FUNKCIJE

Aproksimativna metoda izbora redundanse tj. određivanja vektora $\overline{x}=(x_1, x_2, ..., x_I)$, čiji element x_i predstavlja broj redundantnih komponenata u podsistemu i, i=(1, ..., I), tako da izabrani kriterijum optimalnosti dobije ekstremnu vrednost (max ili min u zavisnosti od kriterijuma) uz istovremeno zadovoljenje postojećih ograničenja u odnosu na raspoložive resurse.

Osnovna ideja u metodi je da se povećanjem broja redundantnih komponenata, iterativno, počev od nekog dopustivog rešenja, jedinično povećava (smanjuje) vrednost kriterijumske funkcije birajući pravce najboljih priraštaja dok god su postavljena ograničenja zadovoljena.

Jednostavna je i lako se implementira na računar.

Dva načina za računanje priraštaja pouzdanosti, kao jedne od mogućih kriterijumskih funkcija:

a) $\Delta_i = \dfrac{R_i(x_i+1) - R_i(x_i)}{R_i(x_i)}$

ili

b) $\Delta_i = R_i(x_i+1) - R_i(x_i)$

U narednoj tabeli date su vrednosti priraštaja pozdanosti pod a) i b), za slučaj dodavanja jedne aktivne redundanse

R(0)=R	0.01	0.1	0.3	0.5	0.7	0.9	0.99
$R(1)=1-(1-R)^2$	0.01999	0.19	0.51	0.75	0.91	0.99	0.9999
$\frac{R(1)-R(0)}{R(0)}$	0.999	0.9	0.7	0.5	0.3	0.1	0.01
$R(1)-R(0)$	0.00999	0.09	0.21	0.25	0.21	0.09	0.0099

Iz tabele se vidi da varijanta računanja priraštaja (a) favorizuje krajnje nepouzdane elemente, dok varijanta (b) favorizuje srednje pouzdane elemente.

Algoritam metode jediničnog priraštaja kriterijumske funkcije biće dat na primeru zadatka alokacije redundanse, kada se maksimizira pouzdanost uz jedno linearno ograničenje, na primer na ukupnu raspoloživu količinu novčanih sredstava (ukupne troškove).

Algoritam

1. Odrediti početne vrednosti za x_i, i=1, ..., I, obično se uzima x_i=0
2. Izračunati odnose priraštaja pouzdanosti kada se x_i poveća za jedan i jedinične cene

   $$\delta_i = \frac{\Delta_i}{c_i} \quad , i = 1,\ldots,I$$

3. Pronaći i^* tako da

   $$\delta_{i^*} = \max_i \delta_i$$

   Ako je δ_{i^*}=0, preći na korak 5. Ako je $\delta_{i^*} \neq 0$, povećati x_{i^*} za jedan.
4. Proveriti da li je zadovoljeno ograničenje u vezi sa ukupnom cenom (troškovima). Ako jeste izračunati novu vrednost za δ_{i^*} i vratiti se na korak 3. Ako nije, smanjiti x_{i^*} za jedan, staviti δ_{i^*}=0, i vratiti se na korak 3.
5. Kraj.

Algoritam se lako može prilagoditi rešavanju drugačije izabranih kriterijumskih funkcija i ograničenja.

DINAMIČKO PROGRAMIRANJE

Dinamičko programiranje se zasniva na principu optimalnosti koji je intuitivno razumljiv, a izražava se na više načina. Princip optimalnosti se može shvatiti kao proširenje principa uzročnosti, tj. determinizma klasične fizike. U slučaju optimizacije redundanse on se izražava: “U slučaju ograničenih sredstava, optimalne odluke o broju redundantnih komponenata u I-serijskom sistemu imaju osobinu da bez obzira na odluke o redundansi u prvom, drugom, i tako dalje do i-tog podsistema, optimalne odluke o redundansi za podsisteme i+1, i+2, i tako dalje do poslednjeg I-tog podsistema, isključivo zavise od sredstava preostalih za i+1, i+2,..., I-ti podsistem (tj. ne zavise od eksplicitnih odluka o redundansi za prvih i podsistema)”.

Prevođenjem principa optimalnosti na matematički jezik dobijaju se rekurentne relacije na osnovu kojih se formira algoritam dinamičkog programiranja. Do rekurentnih relacija dolazi se korišćenjem matematičke tehnike invarijantnog uronjavanja. Ona se sastoji u sledećem: umesto konkretnog optimizacionog zadatka koji je potpuno matematički struktuiran, a njegovi parametri utvrđeni i zadati, posmatra se čitava familija sličnih zadataka, koja je formirana na taj način što se neki, inače u našem konkretnom optimizacionom zadatku zadati parametri, smatraju za promenljive. Tako se na primer u Zadatku 1., I i C smatraju promenljivim, a u Zadatku 2., promenljive su I, C, W, V. Zatim se u tako formiranoj familiji zadataka izabere i reši zadatak koji je najlakše rešiv. To je u ovim slučajevima očigledno zadatak kada je I=1. U sledećem koraku potraži se veza između rešenja pojedinih zadataka u familiji, počev od najlakše rešivog. Tako se dolazi do rešenja konkretno postavljenog zadatka. Veze između rešenja pojedinih zadataka u familiji izražavaju se rekurentnim relacijama.

Posmatrajmo Zadatak 1 (slično važi i za ostale optimizacione zadatke). Zadatak optimizacije redundanse u suštini se svodi na optimalnu raspodelu jednog resursa količine C (ili više vrsta resursa C,W,V) na I podsistema. U Zadatku 1, na svaki podsistem se dodeljuje $c_i x_i$, i=1,...,I resursa. Funkcije $R_i(x_i)$, i=1,..., I izražavaju efekte alokacije diskretnih količina resursa c_i, $2c_i$, $3c_i$, ... (tj. x_i=0,1,2,3...) na podsisteme i=1, ..., I. Svrstavamo naš problem u familiju problema, tj. smatramo C i I promenljivim. U toj familiji jedan od problema ima trivijalno rešenje. To je problem za I=1, kada postoji jedan podsistem i ustvari nema raspodele resursa, već se celokupno C dodeljuje prvom podsistemu, tj. $x_1=\left[\overline{C/c_1}\right]$. Nadvučena srednja zagrada označava celobrojnu vrednost broja u zagradi. Sada treba uspostaviti relacije između rešenja problema za I=1 i I=2, zatim za I=2 i I=3, itd., do poslednje relacije između rešenja za I-1 i I. Da bismo do tih relacija došli definišimo jedan niz funkcija $R_I^*(C)$.

Definicija. $$R_I^*(C) = \max_{(x_1,\dots,x_I)} \prod_{i=1}^{I} R_i(x_i), \quad I = 1,\dots,I \qquad (1)$$

Po definiciji $R_I^*(C)$ je optimalna pouzdanost I-serijskog sistema kada je za redundansu utrošena količina resursa C.

Iz (1) za I=1 sledi:

$$R_1^*(C) = R_1(\left[\overline{C/c_1}\right])$$

Veze između $R_1^*(C)$, $R_2^*(C)$, ..., $R_I^*(C)$ nalaze se na sledeći način. Pretpostavimo da je za poslednji, I-ti podsistem odvojeno $c_I x_I$ resursa, a za svih prethodnih I-1 podsistema tada preostaje $C - c_I x_I$. Ako je $C - c_I x_I$ optimalno raspodeljeno na prvih I-1 podsistema, tada je s obzirom na (1) optimalna pouzdanost ovih I-1 podsistema $R_{I-1}^*(C - c_I x_I)$, a pozdanost celog sistema je $R_I(x_I)\cdot R_{I-1}^*(C - c_I x_I)$. Tada, opet na osnovu (1) važi:

$$R_I^*(C) = \max_{x_I}\left[R_I(x_I)\cdot R_{I-1}^*(C - c_I x_I)\right]$$

Ova rekurentna relacija omogućava da se nađe $R_I^*(C)$ kada se zna $R_{I-1}^*(C)$. Iz principa optimalnosti sledi opšta rekurentna relacija za svaki indeks i=2, ...,I:

$$R_i^*(C) = \max_{x_i}\left[R_i(x_i)\cdot R_{i-1}^*(C - c_i x_i)\right] \qquad (2)$$

$$R_1^*(C) = R_1(\left[\overline{C/c_1}\right])$$

U (2) treba sprovoditi operaciju max po samo jednoj promenljivoj. Ova operacija se izvodi numeričkim pretraživanjem. Ovim postupcima je optimizacioni zadatak traženja maksimuma funkcije I promenljivih x_1, ..., x_I transformisan u niz lakših zadataka određivanja maksimuma funkcija samo jedne promenljive x_i, ali za sve C od 0 do zadatog C, sa pogodno odabranim korakom diskretizacije ΔC.

Računarski proces u dinamičkom programiranju ima dve faze: 1) računanje funkcija $R_i^*(C)$, i=1, ..., I, 2) određivanje optimalnih $x_1^*,\dots,x_I^*$ koristeći funkcije $R_i^*(C)$.

U računanju funkcija $R_i^*(C)$, najpre se uniformno diskretizuje C, sa korakom diskretizacije ΔC, tako da se funkcije $R_i^*(C)$ računaju za argumente 0, Δ, 2Δ, ...RΔ=C. Traženje maksimuma u (2) ide preko nenegativnih celobrojnih x_i, počev od nule. Rezultati primene rekurentne relacije (2) arhiviraju se kao u tabeli koja sledi, gde posle svake kolone $R_i^*(C)$, i=1,...,I stoji po jedna kolona $x_i^*(C)$, i=1,...,I koje prema (2) daju maksimum desnoj strani rekurentne relacije.

C	$R_1^*(C)$	$x_1^*(C)$	...	$R_I^*(C)$	$x_I^*(C)$
0			...		
Δ			...		
2Δ			...		
$\vdots$					
RΔ=C			...		

Kada je ova tabela formirana pristupa se drugoj fazi dinamičkog programiranja. Postupak je sledeći. Za zadato C odmah nalazimo optimalno x_I^* - optimalni broj redundantnih komponenata za podsistem I. Za preostalih I-1 podsistema ostaje $C - c_I x_I^*$ resursa. U koloni C nalazimo vrstu sa tom vrednošću i u preseku te vrste sa kolonom x_{I-1}^* nalazimo optimalni broj redundantnih komponenti za podsistem I-1. Ostaje $C - (c_I x_I^* + c_{I-1} x_{I-1}^*)$ resursa za preostalih I-2 podsistema, itd., do prvog podsistema.

Rekurentne relacije u dinamičkom programiranju kada je kriterijumska funkcija zbir funkcija, biće prikazane na primeru zadatka u kome se maksimizira verovatnoću bezotkaznog rada sistema P_S, do zadatog vremena t.

$$P_S(x_1,\dots,x_I) = \sum_{i=1}^{I} P_i(x_i)$$

pod ograničenjem

$$\sum_{i=1}^{I} c_i x_i \leq C$$

$$x_i \in N_0 = \{0,1,2,\dots\}$$

Rekurentne relacije u ovom slučaju izgledaju:

$$P_i^*(C) = \max_{x_i}\left[P_i(x_i) + P_{i-1}^*(C - c_i x_i)\right]$$

$$P_1^*(C) = P_1(\left[\overline{C/c_1}\right])$$

Takođe, u nastavku će biti prikazane rekurentne relacije za slučaj zadatka kada postoji više ograničenja, kao u Zadatku 2. U ovom zadatku postoje 3 ograničenja. Do rekurentnih relacija se dolazi istim rezonovanjem kao u slučaju postojanja jednog ograničenja. Najpre se obrazuje familija problema tako što se I, C, W, V smatraju promenljivim. Tada se definiše niz funkcija $R_I^*(C,W,V)$.

Definicija. $R_I^*(C,W,V) = \max\limits_{(x_1,\dots,x_I)} \prod_{i=1}^{I} R_i(x_i), \quad I = 1,\dots,I$

Zatim se iz principa optimalnosti slede rekurentne relacije:

$$R_i^*(C,W,V) = \max_{x_i}\left[R_i(x_i) \cdot R_{i-1}^*(C - c_i x_i, W - w_i x_i, V - v_i x_i)\right], \quad i=2,..,I$$

$$R_1^*(C) = R_1(\min\{\left[\overline{C/c_1}\right], \left[\overline{W/w_1}\right], \left[\overline{V/v_1}\right]\})$$

Obim računanja raste, a takođe postaje vrlo važno vešto izabrati korake diskretizacije za tri resursa C,W,V.