

## Raspoređivanje zasnovano na pravilima

Heurističke postupke raspoređivanja možemo uvjetno podijeliti na postupke pretraživanja prostora rješenja i algoritme koji grade raspored u jednom prolazu (ili ograničenom manjem broju prolaza). Jednostavniji heuristički algoritmi su najčešće predstavljeni u obliku koji računa prioritete elemenata sustava te potom element sa najvećim prioritetom uključuje u sustav. Ovakav oblik postupka, koji se naziva i pravilo raspoređivanja, zbog svoje je strukture pogodan za modifikaciju i relativno brzo ispitivanje učinkovitosti metodom pokušaja i pogrešaka.

### Raspoređivanje uz pomoć pravila

U uvjetima raspoređivanja u stvarnom vremenu ili općenito u okruženju u kojemu je potrebno brzo reagirati na promjene stanja sustava, najčešće se za raspoređivanje koristi algoritam koji relativno brzo može doći do potrebnog rješenja. Pri tome potrebno rješenje ne mora biti cjelokupni raspored od početka do (eventualnog) završetka rada sustava, već samo opis sljedećeg stanja sustava, po modelu dinamičkog raspoređivanja. Iako je i postupke koji rješenje pronalaze pretraživanjem moguće posebno prilagoditi za rad u takvim uvjetima (npr. izvođenjem algoritma do zadanog vremenskog trenutka i uzimanjem trenutno najboljeg pronađenog rješenja), takav pristup se ne koristi u većini primjena.

Rješavanje problema u opisanim uvjetima uglavnom se rješava primjenom pravila raspoređivanja. Pravilo raspoređivanja u užem smislu zapravo podrazumijeva samo *funkciju* koja, koristeći određene parametre sustava, definira metriku odnosno prioritet elemenata sustava. U složenijim inačicama funkcija može biti zamijenjena nekim algoritmom koji također mora načiniti odabir najboljeg elementa. Elementi sustava koji se međusobno 'natječu' najčešće su aktivnosti koje traže sredstva sustava, no moguća je i obrnuta situacija (sredstva se natječu za preuzimanje aktivnosti), kao i općeniti slučaj gdje se istovremeno može natjecati više aktivnosti i sredstava. Rješenje koje se dobiva primjenom pravila raspoređivanja definira *pridruživanje* koje je potrebno obaviti prilikom sljedeće promjene stanja sustava. Ukoliko sustav posjeduje samo jedno sredstvo, postupak pridruživanja svodi se na određivanje redoslijeda aktivnosti. Primjena pravila se prema tome razlikuje ovisno o okruženju raspoređivanja, no uvijek se koristi ista funkcija kojom je pravilo definirano.

### Vrste pravila raspoređivanja

Podjelu pravila raspoređivanja možemo definirati promatramo li samo funkciju koja određuje prioritete elemenata u sustavu. Podjela se može načiniti po više osnova [Mor 93], od kojih su najvažnije prikazane u nastavku. Pri tome je potrebno naglasiti da je neke od navedenih vrsta pravila moguće primijeniti samo uz određenu vrstu okruženja raspoređivanja.

S obzirom na promjenjivost prioriteta, pravila možemo podijeliti na *statička* i *dinamička*. Statička pravila računaju prioritete samo jednom (na početku raspoređivanja) i te se vrijednosti tijekom rada sustava ne mijenjaju. Drugim riječima, u računanju prioriteta koriste se samo nepromjenjive vrijednosti, kao što su težine poslova, željeno vrijeme završetka itd. S druge strane, dinamička pravila uključuju ponovno izračunavanje prioriteta svaki put kada je potrebno odrediti novo stanje sustava, a zbog korištenja vremenski ovisnih veličina. Dinamička pravila obično postižu bolje rezultate, mada to ne mora vrijediti u svakom slučaju.

Po pitanju odabira podataka, pravila mogu biti *lokalna* i *globalna* ili oboje istovremeno. U lokalnim pravilima koriste se podaci ovisni o lokalnom elementu sustava, kao što je jedno od sredstava (npr. trajanje operacije nekog posla na dotičnom sredstvu). Globalna

pravila koriste podatke koji su globalno neovisni o pojedinom sredstvu, kao što su vremena dolaska i željenog završetka poslova. Naravno, pravila mogu kombinirati i lokalne i globalne podatkovne strukture.

Ovisno o dostupnosti podataka, raspoznavamo *deterministička* i *nedeterministička* pravila. U determinističkim pravilima javljaju se samo one varijable čija je vrijednost sa sigurnošću poznata ili prethodno određena. Nedeterministička pravila mogu koristiti i veličine čija će se točna vrijednost znati tek nakon završetka rada sustava. Najčešći primjer za to je preostalo vrijeme koje neki posao mora provesti u sustavu, gdje znamo koja količina vremena će se utrošiti na obradu preostalog dijela posla, no ne znamo koliko će vremena biti utrošeno na čekanje. U tome se slučaju raspoređivanje može odvijati na nekoliko načina:

- umjesto stvarne vrijednosti, koristi se *procjena* dotične veličine predstavljena nekim izrazom (npr. preostalo vrijeme u sustavu je vrijeme preostale obrade pomnoženo sa konstantom);
- procjena veličine određuje se na temelju posebne heuristike ili na temelju povijesti sustava, odnosno prethodnih slučajeva raspoređivanja;
- raspoređivanje se provodi *iterativno*: u prvoj iteraciji koristimo početnu procjenu i izradimo cjelokupni raspored, a potom na temelju dobivenog rasporeda odredimo novu vrijednost procjene koju iskoristimo u ponovnoj izradi rasporeda. Postupak se ponavlja dok nije zadovoljen neki uvjet zaustavljanja (najveći broj iteracija, broj iteracija bez poboljšanja, vremensko ograničenje). Već nakon malog broja iteracija mogu se dobiti vrlo velika poboljšanja rezultata, no moraju se uzeti u obzir i povećani vremenski troškovi izrade rasporeda.

### **Prednosti i nedostaci pravila raspoređivanja**

Prednosti pravila raspoređivanja mogu se odmah prepoznati, a najvažnije su:

- *brzina izvođenja* – vremenski troškovi raspoređivanja uz većinu pravila raspoređivanja mogu se smatrati zanemarivima;
- *jednostavnost primjene* – princip raspoređivanja uz pomoć pravila vrlo je jednostavan i obično manje skup od ugrađivanja složenijih postupaka;
- *razumljivost rezultata* – način rada i rezultati pravila raspoređivanja lako su razumljivi i slični ljudskom načinu razmišljanja, stoga se dobivena rješenja brzo interpretiraju ili po želji mijenjaju;
- *brzo reagiranje na promjene* – budući se pravila primjenjuju u dinamičkim uvjetima rada, svaka eventualna promjena u sustavu može se odmah uzeti u obzir.

Uporaba pravila raspoređivanja povlači sa sobom i neke nedostatke; često je, naime, kvaliteta dobivenih rješenja lošija od kvalitete rješenja dobivenog nekim složenijim postupkom pretraživanja (iako to ne mora biti istina za sva okruženja), što je i najočigledniji nedostatak. Osim same jednostavnosti postupka, mogu se identificirati neki uzroci takve lošije učinkovitosti, pri čemu je važno napomenuti da ti uzroci nisu prisutni kod svih vrsta pravila i kod svih načina uporabe pravila.

- *Promatranje samo sljedećeg stanja sustava* – najuobičajeniji način primjene pravila uključuje određivanje samo prve sljedeće promjene stanja sustava. Na ovaj se način često mogu zanemariti neka buduća stanja koja bi mogla utjecati na donošenje trenutne odluke. Djelomično ublažavanje ovoga učinka može se postići uporabom varijabli koje opisuju stanje sustava u budućnosti (ako je to poznato, tj. samo za predodređeno raspoređivanje) ili iterativnom primjenom nedeterminističkih pravila.

- *Neprecizna interpretacija pravila* – budući da pravilo u užem smislu podrazumijeva samo funkciju metrike, za različita okruženja nije precizno definiran način primjene pravila (npr. koji elementi sustava se pridružuju i u kojem trenutku), što posebno dolazi do izražaja kod okruženja koja nisu obuhvaćena klasičnom teorijom raspoređivanja.
- *Uporaba neodgovarajućeg pravila* – ovisno o okruženju i mjerilima vrednovanja rasporeda, potrebno je odabrati pravilo koje će postići najbolje rezultate. Iako je za neke kombinacije okruženja i kriterija poznato koja bi pravila trebalo uporabiti, u većini primjena to nije jednostavan odabir [Cha 96]. Iz toga razloga razvijaju se posebne heuristike koje kombiniraju više pravila raspoređivanja za jedan problem [Wal 05, Auy 03].

Upravo se opisani problem odabira pravila nastoji zaobići metodologijom pronalaženja posebnog pravila za dano okruženje i mjerila vrednovanja, koja je uporabljena u ovom projektu.

## Literatura

- [Auy 03] Andy Auyeung, Iker Gondra, H. K. Dai, *Multi-heuristic list scheduling genetic algorithm for task scheduling*, Symposium on Applied Computing, Proceedings of the 2003 ACM symposium on Applied computing, Melbourne, Florida, Pages: 721 - 724, <http://portal.acm.org/citation.cfm?doid=952532.952673>
- [Cha 96] Yih-Long Chang, Toshiyuki Sueyoshi, Robert Sullivan, *Ranking dispatching rules by data envelopment analysis in a job shop environment*, IIE Transactions, 28(8):631-642, 1996
- [Mor 93] Thomas E. Morton, David W. Pentico, *Heuristic Scheduling Systems*, John Wiley & Sons, Inc., 1993.
- [Wal 05] Scott S. Walker, Robert W. Brennan, Douglas H. Norrie, *Holonic Job Shop Scheduling Using a Multiagent System*, IEEE Intelligent Systems, 2/2005, pp. 50-57