SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Izrada rasporeda nastave za osnovne i srednje škole

Siniša Pribil

Mentor: Doc.dr.sc. Marin Golub

Voditelj: Mr.sc. Marko Čupić

Zagreb, lipanj 2011.

SADRŽAJ

1. Uvod

Veliki broj događaja u našoj svakodnevici je određen različitim oblicima rasporeda (školske i fakultetske obveze, poslovni sastanci, kino projekcije i slično). Tijekom prošlosti svi oblici rasporeda su se izrađivali ručno. Tada to nije bilo jednostavno. Danas je problem izrade rasporeda još složeniji proces jer skup elemenata koje je potrebno rasporediti postaje sve veći, a vremenski okvir unutar kojeg ih je moguće rasporediti sve manji. Iz tih razloga, nije teško objasniti težnju da taj proces bude što je više moguće automatiziran.

Obrazovne ustanove su najmanje jednom godišnje suočene s vrlo složenim zadatkom. Potrebno je za desetke grupa ili razreda sa stotinama učenika ili studenata organizirati predavanja i druge događaje koje će svi moći pohađati. Također je potrebno obratiti pozornost na raspoloživost nastavnika i dvorana odgovarajućih veličina u zadanom vremenu. U skladu s tim potrebama postoje tri oblika rasporeda koji se redovito izrađuju u sustavu obrazovanja: osnovno- i srednje-školski, fakultetski i rasporedi fakultetskih ispita.

Izrada svakog od navedenih rasporeda počinje s drugačijim pretpostavkama i ograničenjima, iako u osnovi moraju zadovoljavati jednake zahtjeve (učenik/student i nastavnik ne smiju imati zadana dva događaja u isto vrijeme, dvorana za svaki događaj mora biti odgovarajućeg kapaciteta i slično). Tako se, na primjer, prilikom izrade rasporeda fakultetskih ispita može ili ne mora pretpostaviti da su svi studenti (osim obaveza izazvanih održavanjem drugih ispita koje polažu) slobodni u vrijeme za koje se raspored izrađuje. U osnovno-školskom i srednje-školskom rasporedu je u pravilu poželjno da su sva predavanja u jednom bloku[1], dok u fakultetskom rasporedu to nije nužno. Dakako, spomenuta podjela na tri oblika rasporeda u obrazovnom sustavu nije čvrsta, pa tako, na primjer, postoje škole koje svojim učenicima nude veću slobodu prilikom odabira predmeta koje žele slušati. Takvi rasporedi tada imaju određene sličnosti s fakultetskim.

[1] blok predavanja – jedno ili više predavanja iz jednog ili više predmeta koja se održavaju za redom, bez pauze (osim odmora)

Općenito je izrada školskih rasporeda posebno zanimljiva zbog velikog broja pedagoških i organizacijskih ograničenja koja je potrebno zadovoljiti. Većina škola u Republici Hrvatskoj radi u više smjena i ima manjak slobodnog prostora. S obzirom na to da trenutno ne postoji sustav za automatiziranu izradu rasporeda koji bi u potpunosti zadovoljio njihove potrebe, taj iznimno težak i dugotrajan posao svake godine odrađuju školski satničari, koji ponekad nisu u mogućnosti izraditi zadovoljavajući raspored.

U ovom se radu opisuje problem automatske izrade rasporeda nastave za osnovne i srednje škole, kao i načini njegova rješavanja. Cilj rada je razmotriti potrebe škola u Republici Hrvatskoj za oblikovanjem sustava za automatsku izradu rasporeda. U sklopu rada obavljeni su razgovori s predstavnicima dvije osnovne (OŠ Voltino i OŠ Sveta Nedelja) i tri srednje škole (IV. gimnazija, V. gimnazija i Prva ekonomska škola) u gradu Zagrebu i okolici, kako bi bile prikupljene informacije o specifičnim zahtjevima koje je pri izradi njihovih rasporeda potrebno zadovoljiti. Iz tog razloga su, osim formalnog opisa standardnog problema izrade rasporeda za osnovne i srednje škole, u radu navedena i dodatna ograničenja koja je potrebno uzeti u obzir prilikom izrade rasporeda za zagrebačke škole.

Formalan opis problema i dodatnih ograničenja pri izradi rasporeda opisani su u poglavlju 2. U poglavlju 3 navode se i opisuju oblici pristupa problemu, složenost, kao i algoritmi za njegovo rješavanje. Poglavlje 4 iznosi zaključak rada.

2. Opis problema

U ovom je poglavlju navedena formalna definicija problema izrade rasporeda nastave u dva oblika. Prvi je oblik sastavljen od najmanjeg mogućeg broja zahtjeva koje bi trebao zadovoljavati svaki važeči raspored. Drugi problem se nadograđuje na prvi uzimanjem u obzir zahtjeve koji se vrlo često pojavljuju prilikom izrade rasporeda u stvarnom životu. U drugom odjeljku opisani su specifični zahtjevi koji se pojavljuju prilikom izrade rasporeda u osnovnim i srednjim školama u Zagrebu.

2.1. Formalna definicija

Prije iskazivanja formalne definicije problema, potrebno je zadati oznake osnovnih pojmova koji su potrebni prilikom izrade rasporeda. Neka oznake: $c_1, \ldots, c_m$ označavaju m razreda, oznake $t_1, \ldots, t_n$ označavaju n nastavnika i neka brojevi $1, \ldots, p$ predstavljaju p termina (sati kroz određeni broj dana za koje se raspored izrađuje). Također, neka $R_{m \times n}$ predstavlja matricu predavanja gdje vrijednost r_{ij} označava broj predavanja koje nastavnik t_j održava razredu c_i u zadanom vremenskom periodu (unutar p termina).

2.1.1. Pojednostavljen problem

Najjednostavniji problem izrade rasporeda može se iskazati kao problem raspoređivanja zadanih predavanja u previđene termine (sate) na način da niti jedan nastavnik ili razred ne budu raspoređeni u više od jednog predavanja u istom terminu. De Werra (1985) je taj zadatak formalno iskazao kao:

$$\text{pronađi} \quad X_{m \times n \times p}$$

$$\text{t. d.} \quad \forall (i,j) \quad \sum_{k=1}^{p} x_{ijk} = r_{ij}, \tag{2.1}$$

$$\forall (i,k) \quad \sum_{j=1}^{n} x_{ijk} \leq 1, \tag{2.2}$$

$$\forall(j,k) \quad \sum_{i=1}^{m} x_{ijk} \leq 1 \text{ i} \tag{2.3}$$
$$\forall(i,j,k) \quad x_{ijk} = 0 \text{ ili } 1,$$

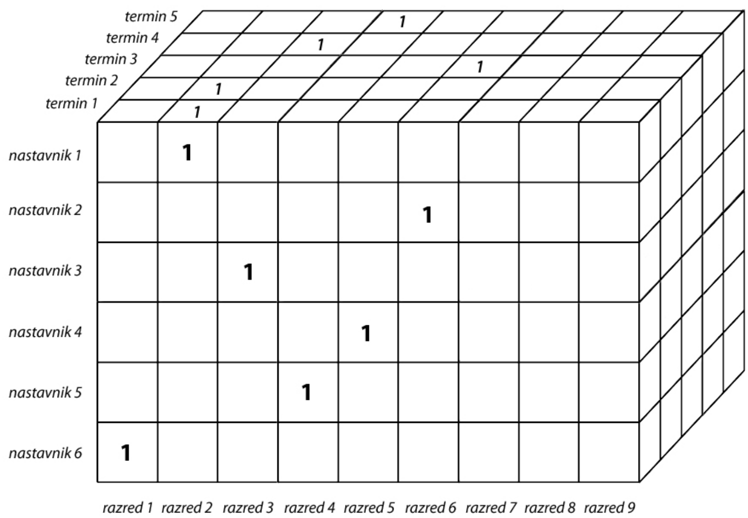
pri čemu je $x_{ijk} = 1$ ako k-ti sat nastavnik t_j predaje razredu c_i, odnosno $x_{ijk} = 0$ inače.

Ograničenjem iskazanim izrazom 2.1 osigurava se da svaki nastavnik održi točno onoliki broj predavanja koliko je zadano. Ograničenja 2.2 i 2.3 osiguravaju da niti jedan razred ili nastavnik nemaju dodijeljena dva ili više predavanja u isto vrijeme.

Dokazano je (Even et al., 1975) kako uvijek postoji rješenje za navedeni problem, uz poštivanje uvjeta da niti jedan razred i niti jedan nastavnik nemaju zadano više od p predavanja u koje ih je potrebno rasporediti. Formalnije zapisano, navedeni problem uvijek ima rješenje ako vrijedi:

$$\forall(i) \quad \sum_{j=1}^{n} r_{ij} \leq p \text{ i}$$
$$\forall(j) \quad \sum_{i=1}^{m} r_{ij} \leq p.$$

Potrebno je naglasiti kako je prilikom rješavanja ovoga problema jedino važno osigurati da svi nastavnici i svi razredi mogu biti prisutni na svim zadanim predavanjima. Bilo kakvo rješenje koje zadovoljava navedeni uvjet je prihvatljivo. Grafički prikaz rasporeda na temelju pojednostavljenog problema prikazan je slikom 2.1.



Slika 2.1: Grafički prikaz rasporeda na temelju pojednostavljenog problema

2.1.2. Osnovni problem

Unatoč činjenici da je raspored dobiven rješavanjem problema opisanog u prethodnom odjeljku ispravan i teoretski izvediv, veoma je mali broj slučajeva u kojima će takav raspored zadovoljiti realne potrebe prosječne škole. Trebalo bi, na primjer, uzeti u obzir da postoje sati kada određeni razred ili nastavnik nisu u mogućnosti biti prisutni na predavanju, odnosno da te sate nisu raspoloživi. To značajno otežava problem, ali i čini dobiveno rješenje prihvatljivijim.

Neka su $C_{m\times p}$ i $T_{n\times p}$ dvije binarne matrice takve da vrijedi $c_{ik} = 1$ (odnosno $t_{jk} = 1$) ukoliko je razred c_i (odnosno nastavnik t_j) raspoloživ u zadano vrijeme k, a $c_{ik} = 0$ (odnosno $t_{jk} = 0$) inače. Tada formalan zapis problema, koji uzima u obzir raspoloživosti razreda i nastavnika, glasi:

$$
\begin{aligned}
&\text{pronađi} \quad X_{m\times n\times p} \\
&\text{t. d.} \quad \forall(i,j) \quad \sum_{k=1}^{p} x_{ijk} = r_{ij}, \\
&\forall(i,k) \quad \sum_{j=1}^{n} x_{ijk} \le c_{ik}, \qquad (2.4)\\
&\forall(j,k) \quad \sum_{i=1}^{m} x_{ijk} \le t_{jk} \text{ i} \qquad (2.5)\\
&\forall(i,j,k) \quad x_{ijk} = 0 \text{ ili } 1.
\end{aligned}
$$

Važno je primijetiti kako izrazi 2.4 i 2.5 zamjenjuju i ujedno postrožuju izraze 2.2 i 2.3 iz pojednostavljenog problema. Naime, tim izrazima je osigurano da niti jednom nastavniku (odnosno razredu) ne može biti dodijeljena obveza onaj sat kada nije raspoloživ.

De Werra (1985) uzima u obzir i unaprijed raspoređena predavanja. To su predavanja kojima je termin određen prije početka izrade rasporeda i ta predavanja se u postupku izrade ne mogu premjestiti. Takav oblik ograničenja koristan je ukoliko za neka predavanja postoje vanjski utjecaji zbog kojih su ona moguća samo u određenim terminima. Unaprijed raspoređena predavanja mogu se formalno iskazati na sljedeći način:

$$\forall(i,j,k) \quad x_{ijk} \ge p_{ijk},$$

gdje je $p_{ijk} = 0$ ukoliko za nastavnika t_j i razred c_i ne postoji unaprijed raspoređeno predavanje u satu k, odnosno $p_{ijk} = 1$ inače. Pokazano je kako je unaprijed raspoređena predavanja moguće ostvariti mehanizmom raspoloživosti koristeći pomoćne izmišljene nastavnike ili razrede (na primjer, izmišljeni nastavnik je raspoloživ u samo jednom terminu).

2.2. Dodatna ograničenja

Ovaj odjeljak sadrži opise dodatnih ograničenja kojima se raspored može bolje prilagoditi potrebama pojedine škole. U odjeljku su ujedno opisani i zanimljivi zahtjevi koje prema svojim rasporedima postavlja pet zagrebačkih osnovnih i srednjih škola. Važno je međutim napomenuti kako svako dodatno ograničenje značajno smanjuje broj mogućih rješenja i samim time otežava zadatak izrade valjanog rasporeda. Dodatak A sadrži popis svih prikupljenih zahtjeva.

2.2.1. Neprekidan raspored

U gotovo svim osnovnim i većini srednjih škola, neprekidnost rasporeda za razrede je vrlo važno ograničenje. Neprekidan raspored znači da razred ne smije imati niti jednu pauzu između predavanja unutar istog dana (kako, na primjer, učenici ne bi lutali školom bez nadzora). Do takve situacije, naravno, neće doći ukoliko je broj predavanja u kojima razred mora sudjelovati jednak ukupnom broju raspoloživih sati p. Ipak, najčešće do takve situacije ne dolazi i u rasporedu postoje sati kada određeni razred nema predavanje. Takvi sati onda obavezno moraju biti prije ili nakon nastave, a nikako između dva predavanja u istom danu. Schaerf (1996) to ograničenje uzima u obzir na način da za svaki razred zadaje sate koji obavezno moraju biti popunjeni predavanjima (u sredini dana), dok je ostale moguće popuniti po potrebi.

Istovremeno, većina škola ne ustraje na potpunim neprekidnim rasporedima za nastavnike, ali zahtijeva da oni budu koliko-toliko uređeni. Tako se obično dopuštaju pauze od jedan sat dnevno do najviše dva do tri sata tjedno. Pauze se, dodatno, obavezno dodjeljuju nastavniku u danu kada ima više od unaprijed određenog broja predavanja. Također, neke škole zahtijevaju da svaki (ili uglavnom svaki) sat u tjednu po jedan nastavnik ima pauzu, kako bi u slučaju potrebe mogao održati zamjensko predavanje.

2.2.2. Raspoređivanje učionica

U osnovnom problemu nije navedeno raspoređivanje predavanja u učionice zato što on pretpostavlja da svaki razred u svakom trenutku ima osiguranu vlastitu učionicu. Koliko god se čini da je ta pretpostavka osnova za normalan rad svake škole, praksa je nažalost pokazala kako ne raspolažu sve škole s takvim resursima. Neka predavanja pak zahtijevaju posebnu opremu koja nije prisutna u svim učionicama (laboratoriji za fiziku, kemiju, učionice za likovni i glazbeni odgoj i slično) ili čak i same učionice nisu raspoložive cijelo vrijeme (na primjer, dvorana za tjelesnu i zdravstvenu kulturu koja se dijeli s drugom školom ili laboratorij koji u određeno vrijeme koriste fakultativne grupe). U svakom slučaju, učionice je često potrebno uključiti u problem izrade rasporeda.

Učionica u kojoj se održava predavanje nova je varijabla kojom je potrebno proširiti prostor u koji se smještaju predavanja prilikom izrade rasporeda. Označimo s $h_1, \ldots, h_o$ o različitih dvorana s kojima škola raspolaže i zadajmo binarnu matricu $H_{o \times p}$ tako da vrijedi $h_{lk} = 1$ ako je dvorana h_l raspoloživa u zadano vrijeme k, odnosno $h_{lk} = 0$ inače. Sada je moguće proširiti i matricu raspoređenih predavanja (iz osnovnog problema) na dimenziju koja uključuje učionice, tako da glavna zadaća problema glasi:

$$\text{pronađi} \quad X_{m \times n \times p \times o}.$$

Osim toga, svakoj se dvorani h_l pridružuje i skup njezinih značajki F_l, odnosno opreme kojom raspolaže. Jednako tako se i svakom predavanju r_{ij} koje je potrebno rasporediti pridružuje skup značajki E_{ij}. Značajkama učionice se može riješiti i slučaj kada učionice nemaju isti kapacitet, a potrebno je paziti da se za određeno predavanje osigura učionica koja će biti dovoljne veličine za zadani razred. Sada se osnovni problem može nadograditi dvama dodatnim ograničenjima:

$$\forall(i,j,k) \quad \sum_{l=1}^{o} x_{ijkl} \leq h_{lk} \text{ i} \tag{2.6}$$

$$\forall(x_{ijkl}) \quad \text{t. d. } x_{ijkl} = 1 \rightarrow E_{ij} \subseteq F_l. \tag{2.7}$$

Izrazom 2.6 se osigurava da se predavanja u određenoj učionici neće održavati one sate kada navedena učionica nije raspoloživa. Izraz 2.7 ograničava raspoređivanje predavanja u samo one učionice koje zadovoljavaju tražene kriterije opremljenosti. Ostali izrazi iz 2.1.2 jednostavno se proširuju za dimenziju učionica i istovjetno moraju vrijediti i u ovom slučaju.

2.2.3. Dijeljena, spojena i paralelna predavanja

Programi u osnovnim i srednjim školama u pravilu su u mnogome slični, a ako postoje značajnije razlike u broju predavanja i predmetima koji se predaju u sklopu više programa, škole najčešće uspješno uspiju organizirati razrede koji se sastoje od učenika istog programa. Unatoč uspješnom organiziranju razreda po programima, učenici često moraju birati između pohađanja, na primjer, vjeronauka i etike ili jednog od više ponuđenih stranih jezika (ponekad i početne i napredne grupe za svakog od njih). Time se dolazi do neizbježne situacije da će se na pojedinim satovima razred morati podijeliti na dvije ili više grupa. Takve podijele obično slijede spajanja manjih grupa iz više razreda koje zajedno slušaju isti predmet. Na primjer, a i b razredi se spajaju prilikom pohađanja etike i vjeronauka.

Važno je uočiti kakva ograničenja na raspored postavlja zahtjev naveden u prethodnom primjeru. Ukoliko se razredi a i b spajaju prilikom pohađanja etike i vjeronauka, predavanja iz navedenih predmeta za oba razreda moraju biti raspoređena u isto vrijeme. Neovisno o tome što se razred dijeli na dvije disjunktne cjeline, ukoliko je zadan zahtjev za neprekidnost rasporeda, svaki učenik navedenog razreda mora imati neprekidan raspored. Postoje dvije mogućnosti za raspoređivanje navedenih predavanja uz istovremeno poštovanje neprekidnosti rasporeda:

- *predavanja iz oba predmeta moraju se održavati u istodobno u različitim učionicama, ili se*
- *predavanja mogu održavati u različitim terminima, ali oni moraju biti na početku ili kraju nastave*.

Prva mogućnost je jednostavnija, ali zahtijeva da nastavnici oba predmeta na koje se razred dijeli, pa i sam razred, budu raspoloživi u isto vrijeme. U drugom slučaju takvog zahtjeva nema, ali u obzir dolaze samo prvi i zadnji satovi zbog očuvanja neprekidnosti rasporeda za drugu polovicu razreda koja ne pohađa navedeni predmet. Lako je primijetiti kako se spomenuti primjer jednostavno može poopćiti za slučaj dijeljena razreda na više od dvije disjunktne cjeline.

Nešto drugačiji zahtjev je ostvarenje usporednih predavanja unutar jednog razreda iz dva ili više predmeta, na primjer, vježbe iz informatike i fizike. Polovica razreda naizmjenično svaki tjedan pohađa određen broj sati vježbi iz jednog ili drugog predmeta. Ovo je također slučaj u kojem se razred dijeli na predavanja dva predmeta, ali se ovom prilikom ne spaja niti s jednim drugim razredom. Mogućnosti za razrješavanje ovog slučaja su jednake kao i kod razdvajanja i spajanja: potrebno je naći termin koji odgovara svim nastavnicima na čija predavanja se razred dijeli ili se navedena predavanja moraju rasporediti na početke i krajeve dana.

2.2.4. Upravljanje opterećenjima

Osim neprekidnosti rasporeda, prilikom izrade rasporeda za škole često se obraća pozornost i na opterećenja učenika i nastavnog osoblja. Tako se, na primjer, prednost daje rasporedima koji imaju ujednačen broj sati kroz dane tijekom tjedna, kako za razrede, tako i za nastavnike (Bufé et al., 2001). Ovisno o školi i njezinim potrebama, moguće je tražiti ili izbjegavati rasporede s više dužih predavanja (duljine veće od jednog sata) istog predmeta unutar istog dana. Mnogi školski satničari trude se što je više moguće olakšati raspored svojim učenicima na način da pokušaju predavanja zahtjevnijih predmeta razmjestiti po različitim danima umjesto u isti dan. Po pitanju broja sati, zadaje se i pokušava poštivati ograničenje o najvećem i najmanjem broju sati nastave dnevno za svaki razred.

Nastavnici obično imaju mogućnost izražavanja želje da u istom danu predaju istoj generaciji (zbog istog gradiva) ili da, ukoliko predaju više od jednog predmeta, u istom danu predaju isti predmet. Takvi se zahtjevi zadovoljavaju ukoliko je to moguće i ne stvara previše problema satničaru.

Dodatno, tijekom razgovora s predstavnicima škola, mnogi su izrazili želju za uvažavanjem poželjnijih raspoloživih termina za nastavnike. Naime, mnogim nastavnicima, zbog obiteljskih ili nekih drugih okolnosti, neki od termina, u kojima su općenito raspoloživi, odgovaraju više od drugih (na primjer predavanja od prvih sati ujutro ili nešto kasnije). Stoga se u mnogim školama, ukoliko to ne predstavlja veliki problem, pokušava udovoljiti takvim željama. Ovakav zahtjev također nije potrebno poštivati u potpunosti, ali može biti dobra mjera pri detaljnijoj međusobnoj usporedbi dvaju rješenja.

3. Metode izrade rasporeda

Ovo poglavlje sastoji se od dva odjeljka. U prvom odjeljku opisana su dva načina na koje se može pristupiti rješavanju problema izrade rasporeda i njihova primjena na formalno opisan problem u poglavlju 2. U istom je odjeljku opisana i složenost algoritma, kao i pristupi problemu s obzirom na interakciju prema korisniku. Drugi odjeljak sadrži opise algoritamskih heurističkih metoda koje se u današnje vrijeme najčešće koriste za izradu rasporeda osnovnih i srednjih škola.

3.1. Oblici pristupa i složenost

U kontekstu interakcije sustava s korisnikom, mnogi autori smatraju da je problem izrade rasporeda nemoguće u potpunosti automatizirati. Kako bi poduprli takvo stajalište, najčešće iznose dva argumenta. Prvi je postojanje faktora koji ljudima jasno razlučuju dva rješenja na bolje i lošije, ali ih je vrlo teško ili čak nemoguće izraziti u *automatiziranom* (engl. *automated*) sustavu. Kao drugi argument iznose činjenicu da je prostor mogućih rješenja iznimno velik. U takvoj situaciji, ljudski faktor može biti presudan za uspješno usmjeravanje pretraživanja prema kvalitetnijim rješenjima na koje se automatski sustav možda ne bi odlučio. Zbog oba navedena razloga, gotovo svi danas raspoloživi sustavi korisniku nude mogućnost izmjene konačnog rješenja (opis i analiza danas značajnijih raspoloživih sustava nalazi se u (Tus, 2011)). Neki sustavi od korisnika zahtijevaju puno veći stupanj uključenosti u proces pa se zovu *interaktivni* (engl. *interactive*) sustavi za izradu rasporeda, ali se ovaj rad bavi tematikom automatiziranih sustava.

Kao što je detaljno opisano u poglavlju 2, problem izrade rasporeda nastave čini organizacija unaprijed zadanog broja predavanja unutar zadanog vremenskog raspona (tipično jedan ili dva tjedna), pritom ujedno zadovoljavajući veći broj ograničenja. Ponekad je dovoljno pronaći raspored koji jednostavno zadovoljava sva zadana ograničenja. Tada se problem izrade rasporeda naziva *problem traženja* (engl. *search problem*).

U drugim slučajevima rješavanju problema pristupa se kao traženju najboljeg (ili gotovo najboljeg) rješenja pa se takav problem naziva *problem optimizacije* (engl. *optimization problem*). U takvim se slučajevima od svakog rasporeda traži da zadovolji sva *čvrsta* ograničenja[1], a dodatni zadatak je optimiranje (maksimalizacija ili minimalizacija) funkcije kvalitete koja ovisi o uspješnosti zadovoljavanja *mekih* ograničenja[2]. Tako je Junginger (1986), za rješavanje osnovnog problema optimizacijom, predložio sljedeću funkciju kvalitete:

$$min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} d_{ijk} x_{ijk},$$

gdje se određena vrijednost d_{ijk} dodjeljuje satima k u kojima je nastavniku t_j manje poželjno održavati predavanje razredu c_i.

Colorni et al. (1992) iznose nešto složeniju funkciju kvalitete rješenja, koja uključuje nekoliko čimbenika kvalitete rasporeda. Tako autori predlažu praćenje:

– *didaktičke* (npr. raspodijeljenost predavanja preko cijelog tjedna i podjednak broj predavanja u danu),

– *organizacijske* (npr. pauza za barem jednog nastavnika svaki sat kako bi mogao održati zamjenu) i

– *kvalitete osobnh rasporeda* (npr. osiguranje slobodnog dana za nastavnika koji predaje u dvije škole).

Funkcije kvalitete koje u obzir uzimaju više čimbenika, obično i bolje razlučuju dva moguća rješenja i samim time daju bolje rezultate.

Probleme pretraživanja lako je pretvoriti u probleme optimizacije. Prema Schaerf (1999), potrebno je samo funkciju kvalitete zadati na način da iskazuje udaljenost trenutnog rješenja od onog "zadovoljavajućeg" (koje u potpunosti zadovoljava sva ograničenja). Lako je uočiti kako je moguć i obrnut proces.

U oba slučaja (pretraživanje i optimizacija) moguće je odrediti temeljni problem. Temeljni problem je problem pronalaska odgovora na pitanje postoji li rješenje (u slučaju pretraživanja) ili postoji li rješenje sa zadanom vrijednošću funkcije kvalitete (u slučaju optimizacije). Kada se spominje složenost određenog problema, zapravo se govori o složenosti njegovog temeljnog problema. Pokazalo se kako je problem izrade rasporeda nastave NP-potpun pri gotovo svim načinima rješavanja. To znači da je najbolje rješenje iscrpnom pretragom moguće pronaći tek za vrlo male slučajeve, dok je

[1] čvrsta ograničenja — ograničenja čijim nepoštivanjem raspored postaje neupotrebljiv

[2] meka ograničenja — ograničenja koja nije potrebno u potpunosti zadovoljiti, ali njihovo zadovoljavanje pridonosi kvaliteti rješenja

za sve ostale potrebno koristiti heurističke metode koje ne jamče da će uvijek pronaći najbolje rješenje.

Pojednostavljeni problem, opisan u poglavlju 2, moguće je riješiti brže, svođenjem na rješavanje problema bipartitnog multigrafa. Razrede i nastavnike moguće je predstaviti vrhovima grafa, dok su predavanja predstavljena bridovima, odnosno razred c_i je povezan s nastavnikom t_j dvama paralelnim bridovima. Even et al. (1975) opisuje metodu rješavanja problema temeljenu na pronalasku niza najvećih skupova bridova koji nemaju zajedničkih vrhova. Hopcroft i Karp (1973) su dokazali kako je niz najvećih skupova, a time i odgovarajući raspored, moguće pronaći u polinomijalnom vremenu.

Pokazalo se još kako je problem moguće svesti i na problem bojanja grafa. Imajući na raspolaganju p boja (svaki nastavni sat predstavljen je jednom bojom), potrebno je obojiti bridove grafa na način da niti jedan par susjednih bridova nije obojan istom bojom. Prema tome, x_{ijk} poprima vrijednost 1 ukoliko je neki brid između vrhova c_i i t_j obojan bojom k. Takav oblik problema također ima polinomijalnu složenost.

3.2. Metode i njihova primjena

Pojednostavljeni problem iz poglavlja 2 ujedno je i jedini oblik problema izrade rasporeda kojeg je moguće riješiti determinističkim metodama s zadovoljavajućim vremenom izvršavanja. Bilo kakav dodatan početni zahtjev svrstava problem u već spomenutu klasu NP-potpunih problema. U ovom odjeljku opisane su heuristike kojima se danas najčešće rješava problem izrade rasporeda nastave za osnovne i srednje škole. Na samom početku opisana je izravna heuristika koja oponaša "ručnu" izradu rasporeda. Ostale navedene metode predstavljaju metaheuristike koje nisu usko vezane uz problem koji se rješava.

3.2.1. Izravne heuristike

Izravne heuristike su metode koje u potpunosti oponašaju ljudski način rješavanja problema. Njima se "ručni" posao automatizira računalom. U slučaju izrade rasporeda nastave, uglavnom se sve metode ručne izrade svode na raspoređivanje jednog po jednog predavanja, sve dok je takvo smještanje moguće. Kada više nije moguće pronaći odgovarajući termin za sljedeće predavanje, pristupa se zamjeni termina dvaju ili više već raspoređenih predavanja kako bi se takav termin oslobodio.

Jedna od takvih metoda opisana je u (Junginger, 1986). Opći postupak moguće je opisati provodeći sljedeća tri postupka:

A: rasporedi *najsloženija* predavanja u termine koji su za njih *najpoželjniji*,

B: kada se u pojedini termin može rasporediti samo jedno predavanje, smjesti to predavanje u navedeni termin i

C: premjesti već raspoređeno predavanje u drugi slobodan termin kako bi se oslobodio termin za novo, neraspoređeno predavanje.

Predavanje je "složeno" kada se na njega odnosi velik broj ograničenja, čime ga postaje teže ispravno rasporediti. Na primjer, to može biti slučaj kada ga predaje nastavnik koji nema puno raspoloživih termina, a istovremeno ima mnogo predavanja koja mora održati. S druge strane, termin je "poželjan" kada se, s obzirom na raspoloživosti ostalih razreda i nastavnika, mali broj ostalih predavanja može smjestiti u njega.

Sustav opisan u (Junginger, 1986) raspoređuje predavanja u termine primjenjujući postupke A i B koliko god je to moguće. Kada više nije moguće niti jedno predavanje rasporediti koristeći navedene postupke, dalje se raspoređuje koristeći postupak C. Postupak A osnova je sustava i korišten je u gotovo svim sustavima koji se temelje na izravnim heuristikama. Razlike su moguće jedino u određivanju složenosti predavanja i poželjnosti termina. Postupak B se koristi kao osiguranje da sustav prilikom primjene postupka A ne dođe do stanja iz kojeg ne može nastaviti s izradom rasporeda. Postupak C najčešće uključuje i neke oblike povratnog praćenja kako bi se mogao ispraviti pogreške do kojih je došlo korištenjem postupka A.

3.2.2. Simulirano kaljenje

Simulirano kaljenje je metaheuristička metoda za rješavanje kombinatoričkih optimizacijskih problema. Inspirirana je postupkom kaljenja metala, a oponaša hlađenje skupine vrućih atoma. Pri visokim temperaturama, atomi imaju visoku energiju i vrlo lako se mogu kretati u nasumičnim smjerovima (na primjer kod rastaljenog metala). Kako se temperatura materijala smanjuje, smanjuje se i mogućnost slobodnog kretanja atoma. Naposljetku, kada je materijal u potpunosti ohlađen (odnosno temperatura padne ispod kritične), atomi poprimaju svoj trajni položaj i više se nisu u mogućnosti micati. Prilikom kaljenja metala, želimo materijal polako hladiti i istovremeno oblikovati željeni objekt. Grubi oblik objekta određujemo pri višim temperaturama, kada je materijal lakše oblikovati. Sa smanjenjem temperature oblikuje se sve više i više detalja i vrši veći broj manjih izmjena dok ne dobijemo konačni oblik.

Isti princip slijedi i računalna simulacija kaljenja. Tijekom cijelog postupka unaprjeđuje se jedno rješenje, a konačan oblik je najbolje (ili skoro najbolje) rješenje problema do kojeg želimo doći. Ono ujedno ima i najnižu energiju, odnosno fizikalno je najstabilnije. Prilikom kaljenja, novo stanje (odnosno rješenje) se prihvaća ukoliko je njegova energija (vrijednost funkcije kvalitete) niža od one koju ima trenutno stanje. Također (u cilju izbjegavanja zaglavljivanja u *lokalnom optimumu*), moguće je prihvatiti i stanje više energije, ali tada je prihvaćanje ovisno o vjerojatnosti prihvaćanja zadane promjene na trenutnoj temperaturi. Drugim riječima, bolje rješenje se uvijek prihvaća, dok se lošije prihvaća s određenom vjerojatnošću. Vjerojatnosti prihvaćanja rješenja mogu se formalno izraziti u obliku:

$$p_{prihvati} = \begin{cases} 1 & \text{ako } f(s') \leq f(s) \\ e^{-\frac{f(s')-f(s)}{T}} & \text{inače} \end{cases},$$

gdje T označava trenutnu temperaturu sustava, s je trenutno rješenje, s' je novo rješenje, a $f(s)$ je funkcija kvalitete rješenja koja se minimizira. Općenit postupak rada metode simuliranog kaljenja prikazan je algoritmom 1.

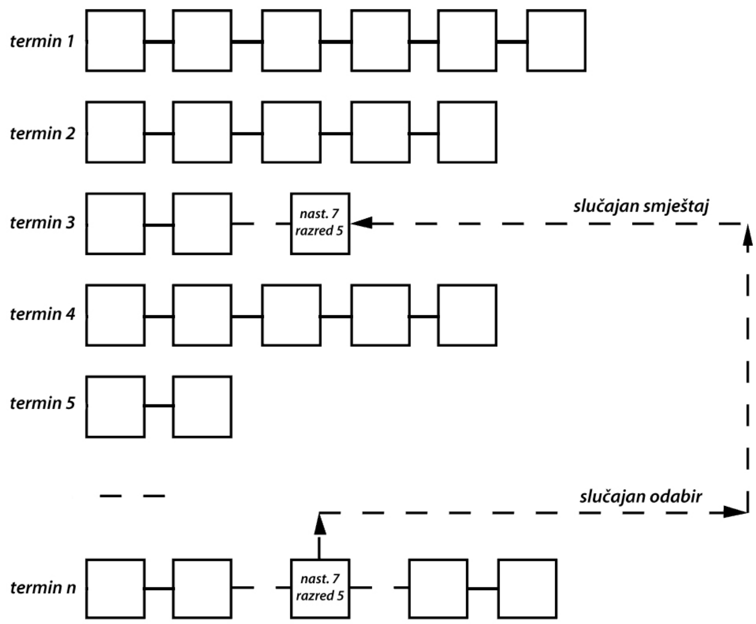
Algoritam 1 Simulirano kaljenje

$s \leftarrow$ nasumično početno rješenje
$T \leftarrow T_0$
dok $(!uvjetZavrsetka)$ **radi**
  $azurirajTemperaturu(T)$
  $s' \leftarrow susjednoRjesenjeOd(s)$
  ako $f(s') < f(s)$ **tada**
    $s \leftarrow s'$
  inače
    $s \leftarrow s'$ s vjerojatnošću $p(T, s, s') = e^{-\frac{f(s')-f(s)}{T}}$
  završi ako
završi dok
vrati s

Postupak se ponavlja dok nisu zadovoljeni uvjeti završetka, a njih može biti više. U svakom slučaju, algoritam može završiti kada se pronađe najbolje rješenje. Ipak, u nekim slučajevima nije moguće unaprijed odrediti svojstva takvog rješenja, pa se najčešće uvode i drugi uvjeti. Neki od njih su i pad temperature sustava ispod određene razine ili vremensko ograničenje, koje se često upotrebljava i prilikom ostvarenja drugih metaheuristika.

Postoje četiri faktora kojima se navedeni općeniti algoritam simuliranog kaljenja može prilagođavati problemu koji se rješava. To su: *odabir susjednog rješenja*, *početna temperatura*, *stupanj hlađenja* i *trajanje trenutne temperature*.

Odabir susjednog rješenja jedan je od najvažnijih faktora kojime se prilagođava algoritam. Naime, kako bi algoritam uspješno težio prema najboljem rješenju, potrebno je izgraditi funkciju koja će iz trenutnog rješenja izgraditi novo. Novo rješenje treba biti drugačije od izvornog, ali i sadržavati poneke dodirne točke kako bi se ostvarila konvergencija prema najboljem. Na taj se način odabiru i istražuju „*susjedi*"[3] trenutnog rješenja. U smislu rješavanja problema izrade rasporeda nastave, susjedno rješenje moglo bi se dobiti premještanje ili zamjena jednog ili više predavanja unutar trenutnog rješenja, kao što je prikazano slikom 3.1. Time se dobiva nešto drugačije, ali slično rješenje. Odabir susjednog rješenja ujedno je i jedini od četiri faktora koji je usko vezan uz problem koji se rješava.



Slika 3.1: Mutacija rasporeda

Početna temperatura jedan je od faktora koji utječe na vrijeme izvođenja algoritma. Što je početna temperatura veća, duže će se sustav hladiti, a samim time se i povećava vjerojatnost da se u početku prihvaćaju rješenja lošija od trenutnog. Time se smanjuje utjecaj početnog nasumično odabranog rješenja na konačno rješenje. Rossi-Doria et al. (2003) predlažu dvije metode za određivanje početne temperature. U prvom se slučaju kao početna temperatura uzima ona koja osigurava vjerojatnost od $1/e$ da će se prihvatiti rješenje koje je 2% lošije od onog nasumično dobivenog. Formalnije zapisano,

[3] susjedno rješenje — izvorno rješenje uz uvedene male promjene

odabire se T takav da je:

$$p = \frac{1}{e} = e^{-\frac{0.02*f(s_r)}{T}},$$

odnosno $T = 0.02 * f(s_r)$. Drugi način koji autori predlažu uključuje stvaranje početne populacije, računanje prosječne vrijednosti varijacije u funkciji kvalitete među članovima stvorene populacije i množenje dobivene vrijednosti s unaprijed zadanim faktorom.

Stupanj hlađenja ima sličan utjecaj kao i početna temperatura: što je stupanj hlađenja veći, to će se sustav sporije hladiti i time duže biti u mogućnosti prihvaćati različitija rješenja. Formalan zapis hlađenja sustava izražen je kao:

$$T_{n+1} = \alpha \times T_n, \quad 0 < \alpha < 1,$$

gdje T_n označava temperaturu u trenutnom koraku, T_{n+1} temperaturu u sljedećem koraku, a α je stupanj hlađenja. Također, postoje i ostvarenja upravljanja temperaturom koja djelomično ponovno zagrijavaju sustav u slučajevima kada zastane u lokalnom optimumu. Mnogi autori takav postupak ocjenjuju vrlo uspješnim.

Naposljetku, trajanje trenutne temperature je čimbenik koji određuje broj ponavljanja algoritma (traženja i ispitivanja susjeda trenutnog rješenja) prije nego se temperatura snizi za faktor stupnja hlađenja. Abramson (1991) predlaže vođenje statistike o ukupnom broju istraženih susjeda, kao i o broju susjeda koje je sustav prihvatio. Autor zatim zadaje brojeve maksimalno dopuštenih istraživanja i prihvaćanja pri istoj temperaturi, a temperaturu sustava zatim smanjuje kada se dostigne bilo koji od dva ograničenja.

3.2.3. Tabu pretraga

Tabu pretraga je također metaheuristika koja unaprjeđuje jedno rješenje u potrazi za najboljim. Slično kao i simulirano kaljenje, postupak započinje odabirom nasumičnog rješenja i nastavlja istražujući njegove susjede. Postupak određivanja susjeda trenutnog rješenja identičan je onome kod simuliranog kaljenja. Za razliku od simuliranog kaljenja koje koristi čimbenike poput temperature sustava i vjerojatnosti prihvaćanja lošijeg rješenja u cilju izbjegavanja lokalnih optimuma i konvergiranja prema *globalnom optimumu*, tabu pretraga koristi memorijske strukture u koje bilježi već učinjene korake. Takve memorijske strukture nazivaju se tabu liste. Tabu pretraga pokazala se kao uspješna metoda za rješavanje raznolikih kombinatoričkih problema iz stvarnoga života.

Osnova sustava tabu pretrage je tabu lista. U nju se pohranjuju rješenja koja je sustav istražio u prethodnih l koraka (prema tome, i sama lista je duljine l). Rješenja koja se nalaze u tabu listi ne uzimaju se u obzir prilikom razmatranja sljedećeg koraka. Time se sprječava stvaranje ciklusa ili zaglavljivanje u lokalnom optimumu. To znači da je moguće da sustav za napredovanje odabere rješenje koje je lošije od trenutnog, ukoliko ne postoji susjed koji ima veću kvalitetu ili se takav nalazi u tabu listi.

Međutim, u nekim se slučajevima dozvoljava tabu korak. To su slučajevi u kojima tabu korak značajnije doprinosi poboljšanju trenutnog rješenja, odnosno zadovoljava kriterij težnje. Kriterijem težnje mogu se, na primjer, spriječiti slučajevi u kojima zbog nedostatka kvalitetnih, ne-tabu izbora algoritam prelazi i nepovoljno područje prostora stanja, a samim time i u mogući lokalni optimum. Pseudokod rada algoritma tabu pretrage prikazan je algoritmom 2.

Algoritam 2 Tabu pretraga

$s \leftarrow$ nasumično početno rješenje
$L \leftarrow \emptyset$
dok $(!vremenskoOgranicenje)$ **radi**
  za n susjeda od s **radi**
    $s_i \leftarrow$ i-ti susjed od s
  završi za
  ako $\exists s_j \mid (zadovoljavaTeznju(s_j)$ && $\forall s_i(f(s_j) \leq f(s_i)))$ **tada**
    $s \leftarrow s_j$
  inače
    $s \leftarrow$ najbolje ne-tabu rješenje od svih s_i
  završi ako
  $L \leftarrow L \bigcup s$
  $L \leftarrow L \setminus$ {najstariji element iz L}
završi dok
vrati s

Ovisno o tome što se točno pohranjuje u tabu listu, postoji više inačica algoritma tabu pretrage. Tako, na primjer, Rossi-Doria et al. (2003) i Colorni et al. (1998) u tabu listu pohranjuju vrste promjena koje su vršili nad trenutnim rješenjem kako bi izgradili novo rješenje. Na takav način u sljedećim koracima neće izravno biti zabranjeno prvotno rješenje, već promjene suprotne od onih koje se nalaze u tabu listi. Autori također mijenjaju i kriterij težnje na način da prihvaćaju tabu promjene ako one dovode do najboljeg dotad pronađenog rješenja. Neke druge inačice tabu liste uključuju zabranjene "atribute" rješenja i slično.

Duljina liste l važna je jer određuje broj koraka algoritma u kojima će određeno rješenje ili promjena biti "tabu". Kako tabu lista izravno utječe na smjer u kojem algoritam napreduje, potrebno je pažljivo odabrati ispravnu vrijednost njezine duljine (mala lista će omogućiti algoritmu stvaranje ciklusa u pretraživanju, dok bi prevelika mogla izazvati izbjegavanje najboljeg rješenja). Rossi-Doria et al. (2003) predlažu zadavanje duljine tabu liste u iznosu određenog postotka broja predavanja koje treba rasporediti (oni odabiru 1%). U cilju smanjenja vjerojatnosti stvaranja ciklusa u pretraživanju i smanjenja broja susjeda rješenja s, isti autori predlažu i prihvaćanje određene promjene s vjerojatnošću 0.1 kako bi cjelokupno susjedstvo rješenja s sveli na svega 10% svih mogućih susjeda.

3.2.4. Genetski algoritam

Genetski algoritam je često korištena metaheuristika inspirirana procesom prirodne evolucije. Sve jedinke koje se spolno razmnožavaju, rađaju se s različitim naslijeđenim svojstvima svojih roditelja. Ta su svojstva "zapisana" u genima koji čine kromosome, a svaka je jedinka jedinstveno opisana svojim genskim zapisom. Tijekom evolucije, genski zapis jedinki može biti promijenjen pod utjecajem vanjskih čimbenika i ne mora imati nikakve povezanosti s genima roditelja. Takve pojedinačne promjene mogu biti pozitivne ili negativne, a nazivaju se mutacijama. Mutirani genski zapis jedinka zatim prenosi na svoje potomke. Ako kombinacija naslijeđenih svojstava roditelja čine potomka bolje prilagođenim za život u okolini u kojoj se nalazi, tada potomak ima i veću vjerojatnost da će u takvoj okolini preživjeti i razmnožavati se. S druge strane, ako nova jedinka nije dobro prilagođena svojoj okolini, manja je vjerojatnost da će preživjeti i moći se razmnožavati. Kao posljedica navedenoga, na nove generacije se prenose uglavnom bolje značajke, čineći ih prosječno sve kvalitetnijim. Uz nasljeđivanje, proces mutacije kao drugi važan faktor pridonosi raznolikosti populacije i omogućava stvaranje novih i boljih obilježja.

Genetski algoritam oponaša proces prirodne selekcije. Koristi se kao tehnika rješavanja složenih kombinatoričkih problema s velikim prostorom pretrage. Za razliku od prethodno navedenih metaheuristika koje unaprjeđuju jedno rješenje, genetski algoritam u svakom trenutku raspolaže s većim brojem pojedinačnih rješenja koja čine populaciju. Primjenjujući operator križanja u populaciji nastaju nova rješenja, a operatorom mutacije unosi se raznolikost u populaciju. Pritom se operacije križanja i mutacije ne vrše uvijek nad svim jedinkama, nego ovisno o vjerojatnostima α i β. Naposljetku se simulacijom procesa prirodne selekcije zadržavaju samo ona najbolja, čime algoritam napreduje prema najboljem rješenju.

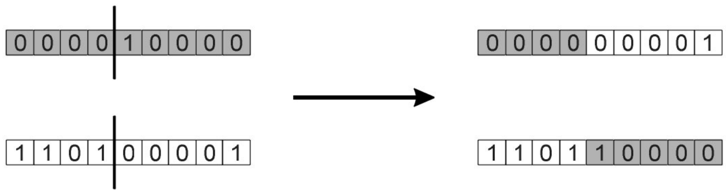
Prema načinu dodavanja novih rješenja u populaciju, postoje dvije vrste genetskih algoritama: generacijski i algoritam stacionarnog stanja. Kod generacijskog genetskog algoritma, cijela stara populacija zamjenjuje se novom, jednako velikom populacijom potomaka. U algoritmu stacionarnog stanja nova jedinka zamjenjuje trenutno najlošiju u populaciji. Važno je naglasiti kako je veličina populacije kod obje vrste u svakom trenutku jednaka.

Trajanje izvođenja algoritma, kao i kod drugih metaheuristika, može biti određeno s više čimbenika. Osim isteka zadanog vremena, uvjeti za kraj izvođenja genetskog algoritma mogu biti i: broj izmjena generacija (ili broj stvorenih jedinki), postizanje željene kvalitete rješenja, duža stagnacija kvalitete rješenja i slično. Opći primjer generacijskog genetskog algoritma prikazan je algoritmom 3.

Algoritam 3 Generacijski genetski algoritam

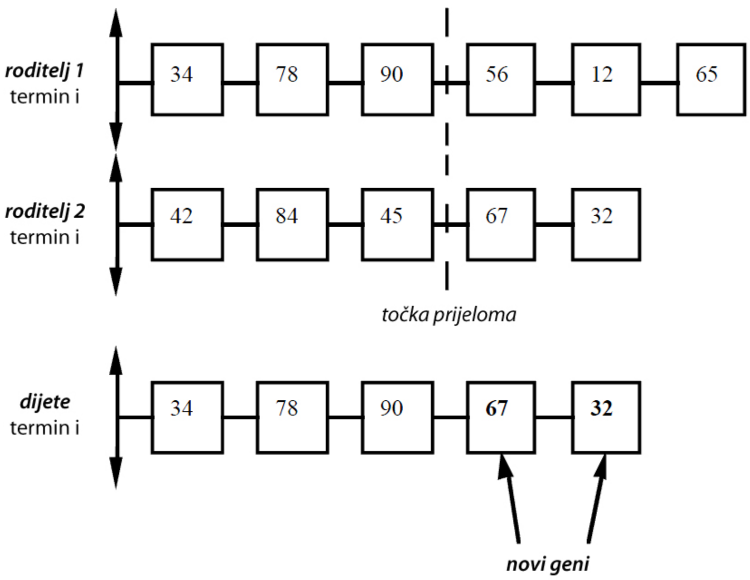
$P \leftarrow$ populacija n nasumičnih početnih rješenja
dok $(!uvjetZavrsetka)$ **radi**
  $P_n \leftarrow \emptyset$
  za $i = 1$ do n **radi**
    $r_{i1}, r_{i2} \leftarrow$ odabir dva roditelja proporcionalno njihovoj kvaliteti (bolje jedinke imaju veće šanse za stvaranje potomstva)
    $d_i \leftarrow krizanje(r_{i1}, r_{i2}, \alpha)$
    $d_i \leftarrow mutacija(d_i, \beta)$
    $P_n \leftarrow P_n \bigcup d_i$
  završi za
  $P \leftarrow P_n$
završi dok
vrati najbolju jedinku iz P

S obzirom na to da genetski algoritam zahtijeva da novo rješenje čini kombinacija značajki dva ili više rješenja-roditelja, potrebno je strukturu rješenja prilagoditi takvim okolnostima. Ako rješenja problema iskažemo kao niz bitova, operator križanja vrlo je lako ostvariti. Moguće je, na primjer, odabrati jednu ili više točaka oko kojih će se vršiti zamjena nizova bitova roditelja. Postupak križanja oko jedne točke prekida prikazan je na slici 3.2. Takvim se postupkom od dvije jedinke roditelja dobivaju dvije jedinke potomaka. Međutim, prilikom rješavanja problema čija su rješenja složena, zapis kromosoma u obliku niza bitova postaje neučinkovit. Velika je vjerojatnost da će križanje na razini bitova u konačnici dovesti do nevažećeg rješenja. Zato se prilikom rješavanja složenijih problema češće koriste hibridni zapisi kromosoma.



Slika 3.2: Križanje oko jedne točke binarnog kromosoma

Hibridni kromosom je, za razliku od binarnog kromosoma, sastavljen od složenih struktura podataka. Time se osigurava bolja kontrola nad postupcima križanja i mutacije. Takav zapis rješenja može biti sličan onome koji se koristi u drugim metaheuristikama, ali mora osiguravati jednostavno upravljanje značajkama. Abramson et al. (1993), za zapis školskog rasporeda, predlaže oblikovanje strukture predavanja organizirane po terminima na način da se sva predavanja koja se održavaju u istom terminu nalaze u istoj listi. Križanje se tada vrlo lako može provesti istovjetno onome nad binarnim kromosomom, zadajući točku prekida. Mutacija se kod hibridnog kromosoma vrši jednako kao i traženje susjednog rješenja kod simuliranog kaljenja ili tabu pretrage: manjim promjenama nad trenutnom jedinkom. Operator križanja nad hibridnim kromosomom rasporeda nastave prikazan je slikom 3.3.



Slika 3.3: Križanje dvaju rasporeda

4. Zaključak

Gotovo svi događaji u našem životu određeni su nekim oblikom rasporeda. Unatoč tome, izrada rasporeda je još danas iznimno težak zadatak i njegovo rješavanje nije jednostavno automatizirati. Većina osnovnih i srednjih škola iz tog razloga svake godine izdvaja značajna sredstva za izradu rasporeda nastave.

U ovom je radu detaljno opisan problem izrade rasporeda nastave za osnovne i srednje škole. Problem je predstavljen u više oblika, počevši od najjednostavnijeg i nadograđujući ga dodatnim zahtjevima. U sklopu rada provedeni su razgovori s predstavnicima pet osnovnih i srednjih škola u Zagrebu i okolici s ciljem prikupljanja detaljnih informacija o potrebama i načinima izrade rasporeda. Neki od prikupljenih zahtjeva detaljno su opisani u ovom radu, dok je cjelokupni detaljan opis iznesen u (Tus, 2011).

U radu su izneseni i oblici pristupa problemu izrade rasporeda, kao i opis njegove složenosti. Dodatno, detaljno je opisana jedna heuristika i tri opće metode kojima se proces može automatizirati. Za svaku od općih metoda opisan je općeniti princip rada i prilagodbe koje je potrebno provesti kako bi se mogao riješiti navedeni problem.

Nakon razmatranja problema i trenutno postojećih metoda, vidljivo je da je izradu rasporeda nastave moguće učinkovito automatizirati. Međutim, kako opisuje Tus (2011), postojeći sustavi koji obavljaju taj posao nisu u mogućnosti zadovoljiti potrebe zagrebačkih osnovnih i srednjih škola. Stoga se nameće zaključak kako je potrebno izgraditi novi sustav koji bi bio više prilagođen njihovim potrebama.

5. Literatura

D. Abramson. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management Science*, 37(1):98–113, 1991. ISSN 0025-1909.

D. Abramson, J. Abela, i CSIRO (Australia). Division of Information Technology. *A parallel genetic algorithm for solving the school timetabling problem*. Citeseer, 1993. ISBN 0868575364.

M. Bufé, T. Fischer, H. Gubbels, C. Häcker, O. Hasprich, C. Scheibel, K. Weicker, N. Weicker, M. Wenig, i C. Wolfangel. Automated solution of a highly constrained school timetabling problem-Preliminary Results. *Applications of Evolutionary Computing*, stranice 431–440, 2001.

A. Colorni, M. Dorigo, i V. Maniezzo. A genetic algorithm to solve the timetable problem. *Politecnico di Milano, Milan, Italy TR*, stranice 90–060, 1992.

A. Colorni, M. Dorigo, i V. Maniezzo. Metaheuristics for high school timetabling. *Computational Optimization and Applications*, 9(3):275–298, 1998. ISSN 0926-6003.

D. de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, 1985. ISSN 0377-2217.

S. Even, A. Itai, i A. Shamir. On the complexity of time table and multi-commodity flow problems. U *16th Annual Symposium on Foundations of Computer Science*, stranice 184–193. IEEE, 1975.

J.E. Hopcroft i R.M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225, 1973.

W. Junginger. Timetabling in Germany: a survey. *Interfaces*, stranice 66–74, 1986. ISSN 0092-2102.

O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, et al. A comparison of the performance of different metaheuristics on the timetabling problem. *Practice and Theory of AutomatedTimetabling IV*, stranice 329–351, 2003.

A. Schaerf. *Tabu search techniques for large high-school timetabling problems*. Computer Science, Department of Interactive Systems, CWI, 1996.

A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2): 87–127, 1999. ISSN 0269-2821.

A. Tus. *Analiza zahtjeva i postojećih rješenja u izradi rasporeda sati za škole, seminarski rad u sklopu predmeta Seminar*. Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave, FER, 2011.

6. Sažetak

Izrada rasporeda nastave za osnovne i srednje škole složen je kombinatorički problem. Danas postoje mnoga računalna rješenja za automatiziranu izradu rasporeda. Međutim, ona ne zadovoljavaju potrebe većine škola u Republici Hrvatskoj. Ovaj rad opisuje problem izrade rasporeda nastave za osnovne i srednje škole. Opisuju se oblici i složenost problema, kao i metode koje se mogu koristiti pri njegovom rješavanju. Također, razmatraju se neki od problema s kojima se suočavaju satničari pet zagrebačkih škola prilikom izrade rasporeda.

Dodatak A
Popis zahtjeva i ostalih ulaznih parametara za izradu rasporeda

Dodatak A sadrži popis svih zahtjeva koji su prikupljeni u razgovorima s predstavnicima pet zagrebačkih osnovnih i srednjih škola, kao i pregledom opcija trenutno postojećih računalnih rješenja za izradu rasporeda nastave. Zahtjevi su, zbog preglednosti, navedeni kao popis ulaznih parametara sustava za izradu rasporeda i podijeljeni u sedam glavnih cjelina na koje se odnose. Detaljan opis svih navedenih parametara s pregledom postojećih aplikacija iznesen je u (Tus, 2011).

1. **Raspored**
   (a) tip rasporeda (jedan, dva, više tjedana)
      i. broj dana u tjednu
   (b) broj i termini početaka satova
      i. "granice" turnusa (na primjer, za $2.d$ i $3.c$)
      ii. sati od kojih raspored može početi ("dozvole" za dolazak na 2., 3. sat...)
   (c) zastavica za izbjegavanje dva "duža" predavanja u danu
      i. trajanje "dužeg" predavanja
   (d) zastavica kontrole kapaciteta dvorana i veličina grupa
   (e) broj dvorana (za slučaj da nisu sve unesene)
2. **Nastavnik**
   (a) raspoloživost u školskim satima
      i. poželjni sati

    ii. poženjni dani u tjednu
    iii. mogućnost zadavanja broja slobodnih dana (bilo kojih)
    iv. mogućnost zadavanja broja dana koje mora raditi (bilo kojih)

  (b) zastavica zahtijevanja predavanja istoj generaciji u istom danu

  (c) ukoliko predaje više predmeta, zastavica predavanja istog predmeta u istom danu

  (d) ukoliko radi u oba turnusa, zastavica izbjegavanja dva dana za redom u istom turnusu

  (e) najveći i najmanji broj sati u danu

  (f) pauza ukoliko je broj sati u jednom danu veći od određenog broja

  (g) najveći i najmanji broj rupa u danu

  (h) najveći i najmanji broj rupa u tjednu

  (i) najveća veličina rupe

3. **Predmet**

  (a) raspoloživost u školskim satima
    i. poželjni sati
    ii. poželjni dani u tjednu

  (b) faktor zahtjevnosti predmeta

  (c) "položaj" predmeta u rasporedu (unutar nastave, međuturnus, suprotan turnus)

4. **Razred**

  (a) podjele (grupe vezane uz predmete, npr.: eng-njem, tjM-tjZ, fiz-inf...)
    i. ovisno o $1.d$ mogu ili ne moraju sadržavati brojeve učenika

  (b) raspoloživost u školskim satima
    i. poželjni sati
    ii. poželjni dani u tjednu

  (c) zastavica obaveznog neprekidnog rasporeda

  (d) najveći broj sati u danu

  (e) zadana učionica

(f) pomoćne učionice

(g) najveće dnevno opterećenje

5. **Dvorana**

6. raspoloživost u školskim satima

   (a) poželjni sati

   (b) poželjni dani u tjednu

7. kapacitet dvorane - ovisno o $1.d$ može ili ne mora biti navedeno

8. **Predavanje**

   (a) predmet (jedan)

   (b) nastavnici (jedan ili više)

   (c) razredi/grupe (jedan ili više)

      i. izravno zadani razredi/grupe kojima se predaje

   (d) raspoložive učionice (jedna ili više)

      i. izravno zadane učionice

      ii. broj potrebnih učionica za odabrati

      iii. zastavica "ne prati učionice" (tada se samo pazi na ukupan broj, ne i razmještaj predavanja - onda nije potrebno navesti $6.d.i$)

   (e) trajanje predavanja

      i. indikator dozvole da algoritam može sam "razlomiti" predavanje u više manjih blokova (u tom slučaju preslikavanje nije $1:1$ - jedno zadano predavanje se preslikava u više predavanja u rasporedu)

         – najveća i najmanja veličina tako nastalih manjih blokova

   (f) tjedan u kojem se održava (ako je raspored višetjedni)

      i. mogućnost odabira: "jednom u rasporedu" (znači jednom u dva/više tjedana) - za 1.5 (i slično) sati tjedno

   (g) čvrsto zadan/i termin/i održavanja predavanja (ima ih najviše onoliko za koliko se tjedana radi raspored) - isključuje $6.c.ii$ i $6.e.i$ jer se odnosi na točno zadani termin

9. **Veze**

   (a) izravno zadano dva ili više predavanja, ili

   (b) predmeti i razredi na koje se odnosi pravilo

   (c) pravila:

      i. moraju/ne smiju biti u istom danu

      ii. moraju/ne smiju biti u isto vrijeme

      iii. moraju/ne smiju biti uzastopno

         – ako moraju biti uzastopno, zastavica je li važan poredak

      iv. moraju/ne smiju biti u istom tjednu (samo za $6.f.i$)