1 Pradiniai duomenys

Profiliuotų mokyklų 11-12 klases pagrindinai mokosi grupėmis, pagal mokinių pasirinkima, 1-10 klases taip pat turi darbo grupes, tik mažiau, pavyzdžiui, anglų-vokiecių kalbos, tikyba-etika, mergaičių-berniukų darbai.
Kai kurioms paskaitoms, kelios grupes gali buti sulietos i viena jungtine grupe ir atvirksciai. Jungtines grupes charakteringos 1-10 klasese, nes cia skirstymas i grupes vykdomas tik kai kurioms disciplinoms: uzsienio kalbos, tikyba-etika, darbai.

Tokiu atveju, grupės atlieka klasių roles, tvarkaraščio požiūriu. Tada, tradicinio tvarkaraščio Fig. 1 užrašyti klasių kodai keičiami į grupių kodus, pavyzdžiui, vietoj

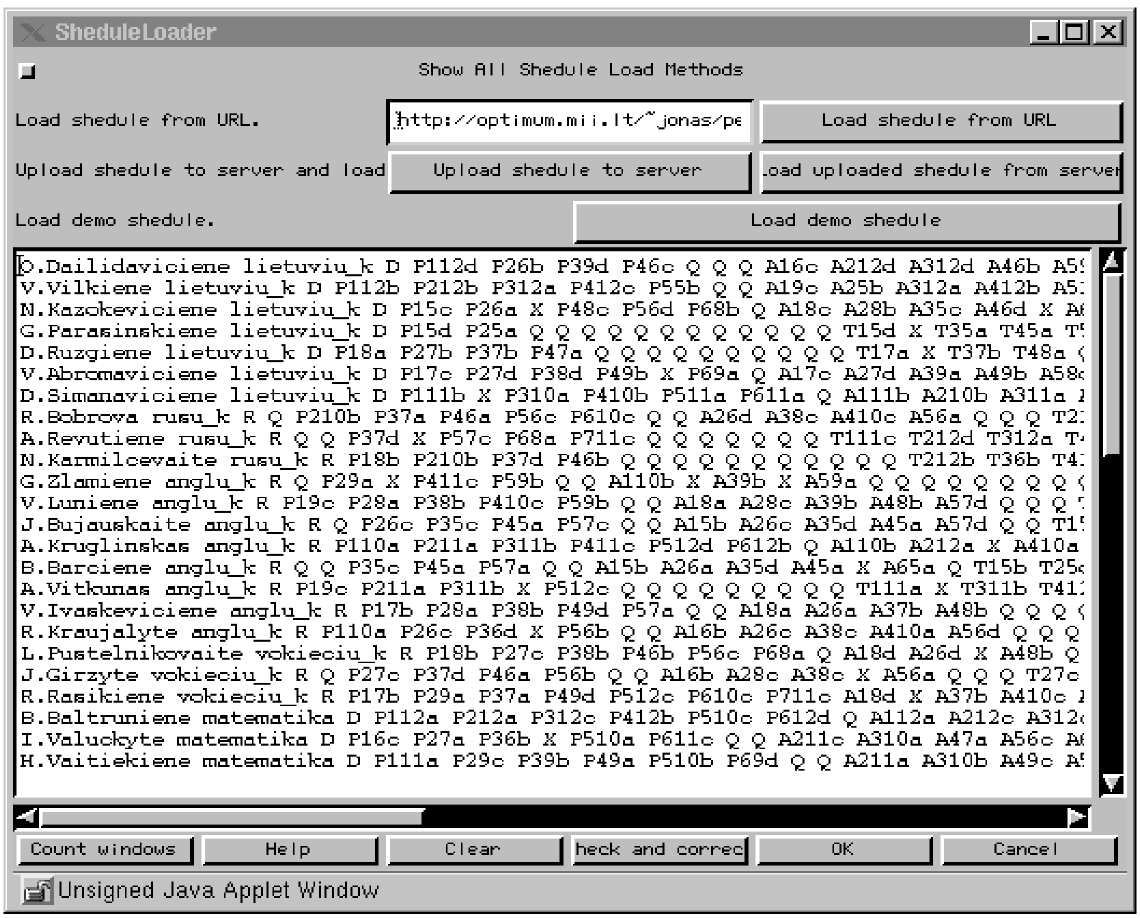

Figure 1: Tradicinio tvarkaraščio fragmentas.

klasės "12.a" rašom grupę "12.54", nes skirtingų 12-tos klasės grupių gali būti beveik numeriai, kad mokytojai ir mokiniai žinotų kur eiti. Taip papildytos lentelės eilutės $i = 1, ..., M$ nurodo mokytojus ir jų dėstomus dalykus , stulpeliai $j = 1, , , .V$ nurodo valandas (susikirstytas pagal savaitės dienas). Langeliuose (i, j), $i = 1., , , .M$, $j = 1, ..., V$ nurodomi grupių kodai ir kabinetų numeriai. Šie duomenys tinka ir mokiniams ir kompiuteriui. Todėl būtent toks tvarkaraštis turėtų būti įvedamas ir išvedamas, kad išvengti įvedimo/išvedimo klaidų. Optimizavimui gali būti patogesni kiti tvarkaraščio formatai.

2 Duomenų atvaizdavimas optimizuojant

Norint vaizdžiai aprašyti optimizavimo algoritmus
gali būti patogu naudoti binarinius masyvus,
kur 1-taip, 0-ne.
Tačiau, tam reikėtų keturmačio masyvo
$tvarka[M][V][G][K]$,
kur
G yra grupių skaičius,
K kabinetų skaičius,
$tvarka[i][j][k][l] = 1$
reiškia, kad mokytojas i valandą j grupę k turės pamoką kabinete l,
$tvarka[i][j][k][l] = 0$,
reiškia kad pamokos nebus.

Skaitom, kad mokytojai sėsto tik po vieną dalykaą. Tačiau, vieną dalyką gali dėstyti keli mokytojai. Tai suskaido mokytojų aibę į dalykų poaibius žymimus indeksu $n = 1, ..., D$, kur $D \leq M$.

Tai labai didelis masyvas, todel jis vargai tiktų realios mokyklos tvarkaraščio optimizavimui. Toliau jis patogus aiškinant naujus algoritmams bei tiriant jų efektyvumą..

Realiam tvarkaraščiui optimizuoti ekonomiškiau (atminties prasme) būtų naudoti dvimatį masyvą, pavyzdžiui, tokį pat kaip naudojam įvedimui/išvedimui.
Skirtumas tik tas, kad mokytojai, dalykai, grupes ir kabinetai būtų pažymeti tiesiog eilės numeriais.

3 Tvarkaraščio įvertinimas

Vertinant bendrais žodžiais, tvarkaraštis turi tenkinti fizinius bei norminius ribojimus bei būti pats patogiausias mokytojams ir mokiniams. Formaliai, tai diskretus vektorinės optimizacijos uždavinys su ribojimais.

3.1 Fiziniai ribojimai

1. mokinys vienu metu gali būti tik viename kabinete vietoje

3. mokiniai ir mokytojai negali dirbti daugiau negu 168 val. per savaitę.

Aibę tvarkarasčių t tenkinačių fizinius ribojimus žymėsim Φ

Kitus ribojimus galima skaityti norminiais, nes fiziškai juos įmanoma pažeisti.

3.2 Norminiai ribojimai

Normos uždoda šias sąlygas:

1. savaitinių valandų skaičius grupėms $v_{gs} \leq V_{gs} < 168$
2. valandų skaicius per dieną grupėms $v_{gd} \leq V_{gd} < 24$,
   11-12 klasėms, leistinas savaitinių valandų skaičius $V_{gs} < 5V_{gd}$ yra mažesnis negu leistinų dienos valandų sandauga $V_{gs} < 5V_{gd}$,
3. kiekvienam mokytojui i duotas savaitinių valandų skaičius V_i
4. kiekvienai grupei k duotas jai destomų dalykų valandų skaičiai V_{kn}, kur k grupės numeris, n dalyko numeris
5. viename kabinete, vienu metu gali būti tik viena pamoka (tai norminis ribojimas, nes fiziǩai jį galima pažeisti)
6. kai kuriems dalykams bei grupėms reikalingos dvigubos (dvi is eilės) pamokos, kai kurioms jos draudžiamos
7. 1-10 klasėms grupių "langai" (ilgi tarpai tarp pamokų) draudžiami

3.3 Patogumo įvertinimas

3.3.1 Nepatogumo faktoriai

Skaitysim, kad patogesnis tas tvakaraštis, kuris sudaro mažiau nepatogumų mokytojams ir mokiniams. Nepatogumo faktoriai šie:

1. mokytojų "langai"
2. grupių langai
3. nepatogios valandos mokytojams
4. nepatogios pamokų sekos

Apskritai, nepatogumas yra subjektyvus, todėl įvairius nepatogumo faktorius vertinsim baudomis. Kadangi baudos subjektyvios, jos nustatomos kaip pradiniai optimizavimo uždavinio parametrai priklausantys ir nuo mokyklos ir nuo laikotarpio. Baudų sarašas:

1. $c_i \geq 0$ - bauda už vieną mokytojo i langą
2. $c_k \geq 0$ - bauda už vieną grupės k langą
3. c_j - bauda už nepatogią valandą j
4. c_s - bauda už nepatogią pamokų seką s

Langų baudos c_i ir c_k gali priklausyti ir nuo atstumo iki "krašto", t.y. pamokų pradžios arba pabaigos, nes kuo arciau krašto, tuo lengviau tokiu langu "atsikratyti" atliekant tvarkaraš čio perstatymus.

Tvarkaraščio t nepatogumo baudų suma:

$$C_t^0 = \sum_i c_i L_i + \sum_k c_k L_k + \sum_i \sum_j N_{ij} + \sum_k \sum_j N_{kj} c_j + \sum_s N_s c_s \qquad (1)$$

kur
L_i tai skaičius mokytojo i langų
L_k tai skaičius grupės k langų
N_{ij} tai skaičius mokytojo i nepatogių valandų j
N_{kj} tai skaičius grupės k nepatogių valandų j

4 Baudos už normų pažeidimus

Formaliai, normos negali būti pažeidinėjamos. Realiai, galima pateisinti nežymius nukrypimus nuo normų jei tai labai padidina patogumą. Todel, įvesim "normų baudas"

$$C_t^1 = \sum_r c_r N_r \qquad (2)$$

kur
c_r tai bauda už nukrypimą nuo normos $r = 1, ..., 7$ N_r tai skaičius normos r pažeidimų. bauda c_r priklauso nuo pažeidimo dydžio, paprastai tiesiog proporcinga jam.

5 Optimizavimo uždavinys

Sumą visu tvarkaraščio t baudų žymesim $C_t = C_T^0 + C_T^1$ Kadangi įvertinimas C_t priklauso ne tik nuo objektyvių bet ir nuo subjektyvių faktorių, toliau vadinsim ji

$$\min_t C_t \tag{3}$$

$$t \in \Phi \tag{4}$$

kur Φ yra aibė tavrkaraščių tenkkinančių fizinius ribojimus.

5.1 Optimizavimas perstatant

Norint atsižvelgti į mokytojų patyrima naudotinas pradinio tvarkaraščio perstatymų metodas. Perstatymų metodas susideda is

1. perstatymo
2. sprendimo ar pasilikti ar pajudeti
3. geriausio tvarkaraščio saugojimo

5.1.1 Perstatymai

Leistinas perstatymas kai leistinas pradinis tvarkaraštis perstatinėjamas nepažeidžiant leistinumo, sunku tai, kad perstatyto tvarkarašçio leistinumo užtikrinimas dažnai prieštarauja konvergavimo sąlygoms reikalaujančioms pakankamai "laisvo judėjimo" tikslu garantuoti kelią globalaus optimumo link, pavyzdžiui, School COMPLETE aprašytas leistino vieno lango perstatymo algoritmas nekonverguoja, ir gana greit "istringa", jei pradinis tvarkaraštis gerai padarytas.

Perstatymas su baudomis kai išlaikant fizinius ribojimus nagrinejami ir neleistini pagal normas perstatymai numatant baudas, kurios priklauso nuo normų pažeidimo laipsnio, kiekvieną tvarkarastį t įvertinsim heuristika

$$C_t = C_t^0 + C_t^1 \tag{5}$$

kur C_t tai visu baudų suma

Šis budas patogesnis konvergavimui, tačiau sunku pasiekti leistinuma. Leistinuma galima "skatinti" pakartojant optimizavimą su padidintomis norminėmis baudomis, t.y

$$C_t(b) = bC_t^0 + c_t^1 \tag{6}$$

kur $b > 1$ didelis Jei tai nepadeda, tada reikalingi papildomi koregavimo algoritmai pašalinantys likusiias neleistinas vietas, čia galetų tikti automatinis koregavimas aprašytas. School COMPLETE.

Optimizavimo greitis ir tikslumas, bei tvarkaraščio leistinumas priklauso nuo baudu, todėl, norint efektyviai dirbti, visos baudos užduodamos monitoriuje, Siekdami konvergavimo, naudosim perstatymą su baudomis bei populiaraus ”simulated annealing” algoritmo modifikaciją:

$$p_{t+1} = 1, \ C_{t+1} < C_t, \quad (7)$$
$$p_{t+1} = exp(C_t - C_{t+1})x, \ C_{t+1} > C_t \quad (8)$$

kur
$t+1$ - tvarkaraštis po perstatymo,
t- tvarkaraštis pries perstatymą,
p_{t+1}- tikimybė pereiti i naują tvarkaraštį ,
čia parametras x optimizuojamas minimizuojant funkciją $f_K(x)$,
kur $f_K(x)$ mažiausia C_t reikšmė,
pasiekta atlikus K tokių iteracijų.

Veliau reikės pagalvoti ir apie analogiška kai kurių baudų optimizavimą, pavyzdžiui koeficioento b, nes nuo jo priklauso konvergavimo greitis ir tikslumas.

5.3 Geriausio išaugojimas

Geriausio ,tvarkaraščio išsaugojimas vykdomas sulyginant naujo ir seno įvertinimus C_t, senas naikinamas, jei naujas geresnis, tai vyksta visą laiką, kai tik paskaičiujamas naujo tvarkaraščio C_t, kartu registruojami atitinkami C_t^0 ir C_t^1

6 Pradinis algoritmo tyrimas

Konkretau perstatymą, kol kas, sunku pasiūlyti. Čia labai svarbi ir mokytojų ir programuotojų nuomonė, butent cia ir pasireikš mokytojų patyrimas bei programuotojų intuicija, kuri, aišku, ivertins ir programinio realizavimo patogumą

Tyrimus pradeti galima ir nuo paprastesnio neprofiliuotų mokyklų tvarkarašçio, nes profiliuotoms mokykloms viskas daug sudetingiau.

Apskritai, šis darbas butu idomus programuotojų ir mokytojų informatikų bendradarbiavimo pavizdys namų darbų rėmuose.

Taciau gyvenimiškų uždavinių sprendimui, kurie reikalingas didelis darbas, magistro ar net daktaro teziu lygio. Tikiuosi, kad atsiras kas tam pasiryš.

Uždavinys butent taip suformuluotas todėl, kad elementarus vieno lango perstatymas nekonverguoja ir, pagerinimas labai nežymus, vos du langai, jei pradinias tvarkaraštis padarytas gerai. Taip gaunasi todėl, kad patys mokytojai tokius akyvaizdžius pagerinimus pastebi ir juos realizuoja be jokių kompiuterių.

Žinoma globalaus optimizavimo taisyklė yra tai, kad pereiti iš lokalaus optimumo į globalų, paprastai, galima tik laikino pablogėjimo saskaita. Mūsų atveju tai reikštų, kad kelias nuo duoto leistino tvarkaraščio į geresnį eina per neleistinus tvarkaraščius iššaukiančius normų pažeidimo baudas.

.Jei sugalvotas algoritmas dirbs patenkinamai, tai bus galima apibendrinti profiliuotoms mokykloms.

7 Pradinio tvarkaraščio sudarymas. "Greedy" heuristikos

Pagrindiniių mokyklų pradinis tvarkaraštuis paprastai žinomas, tai įprastas mokytojų darbas. Profiliuotoms mokykloms patyrimo dar nedau, be to, čia viskas daug sudėtingiau. Todėl, daugeliu atvėjų, pradinį tvarkaraštį reikėtgeneruoti pavyzdžiui, naudojant "greedy" heuristikas.

Tvarkaraštį sudarinėjam nuosekliai papildant jau sudarytą tvarkaraščio dalį. Kiekviename žingsnyje minimizuojam C_t. Skirtingai nuo perstatymo algoritmo, čia C_t nurodo ne viso tvarkarasčio o tik jo dalies tikslo funkciją (įskaitant papildymą). Pavyzdžiui, pradedam nuo "nulio", t.y. pradinis tvarkaraštis
$t^0 = tvarka[M][V][G][K] = 0$.
Nepatogumo baudų čia nebus $C_t^0 = 0$,
tačiau bus didelės normų pažeidimo baudos C_t^1,
nes jokių pamokų nebus todel bus pažeisti skyrelio 3.2 ribojimai 3 ir 4, krie nurodo pamokc u valandas mokinims ir mokytojams.
Tada pirmi trys žingsniai tokie

1. $t^1 = tvarka[1][1][1][1] = 1$, pereinam į t^1 nes čia $C_{t^1} < C_t$,
   kadangi $C_{t^1}^1 < C_t^1$ ir $C_{t^1}^0 = C_t^0 = 0$
2. $t^2 = tvarka[1][1][1][1] = 1,\ tvarka[1][2][2][1] = 1$, pereinam į t^2 nes $C_{t^2} < C_{t^1}$
3. $t^3 = tvarka[1][1][1][1] = 1,\ tvarka[1][2][2][1] = 1,\ tvarka[1][3][3][1] = 1$, čia taip pat $C_{t^3} < C_{t^2}$ todėl.perteinam į t^3
4. kiti žingsniai daromi panašiai. Skirtumas bus tas, kad užpildant tvarkaraštį palaipsniui prijungi.ejant naujus elementus, susidarys situacija kai teks mokėti ne tik norminės bet ir nepatogumo baudas. Tada gali buti, kad $C_{t^{n+1}} > C_{t^n}$ ir sprendžinat klausimąr pereiti į blogesnį t^{n+1} teks naudoti Simulated Annealing ar ka nors panašaus.

Aišku, kad rezultatas priklauso nuo eilės tvarkos, todėl ją nustatant reikėtų tartis su ekspertais.

Toks pradinio tvarkaraščio sudarymas, matomai, bt̄ų naturalus pirmas žingsnis optimizuojant profiliuotų mokyklų tvarkaraščius.

8 Paprastesnis formulavimas

Algoritmo tyrimui bt̄ų patogu sutapatinti mokytojus, dalykus ir kabinetus, tada užtektų trimačio masyvo $tvarka[M][V][G]$,

G yra grupių skaičius,
V valandų skaičius,
M mokytojų-dalykų-kabinetų skaičius.

Be to, tai yra bendresnis tvarkaraščio uždavinio formulavimas, kur G - darbų skaičius
V - laiko intervalas (laiko vienetais),
M - mašinų skaičius.
Čia mašinų eilės tvarka laisva,
reiškia turim tradicinį kalendorinio planavimo uždavinį vadinamaą "Job-Shop".

Šis uždavinys gali buti užduotas fiksuojant darbų k eilės tvarką g bei masinų i eilės tvarkas m_i kiekvienam darbui , kur, pavyzdžiui,
$g = (1,3,2,5,6,4)$, $m_1 = (2,1,4,5,3)$, $m_2 = (1,2,3,5,4), \dots$.
Čia darbų skaičius $G = 6$, mašinų skaičius $M = 5$, darbų vykdymo eilės tvarka $g = (1,3,2,5,6,4)$, mašinų eilės tvarka vykdant pirmą darbą $m_1 = (2,1,4,5,3)$ ir .t.t. Pirmasis fizinis ribojimas (ži-ūr. skyrelis 3.1) yra tai, kad vienu laiku viena darba galima daryti tik viena mašina. Formaliai

$$\sum_{i,j} tvarka[i][j][k] = 1,\ j = 1, ..., L,\ k = 1, ..., G \tag{9}$$

Antras fizinis ribojimas atpuola sutapatinus mokytojų ir kabinetų aibes, t.y. $K = M$. Trečias fizinis ribojimas riboja mašinų darbo ir darbų vykdymo laikus, pavyzdžiui,

$$\sum_{j} tvarka[i][j][k] \leq 168,\ i = 1, ..., M,\ k = 1, ..., G \tag{10}$$

$$\tag{11}$$

Fizinius ribojimų (9) ir (10) reikiėtų griežtai laikytis atliekant bet kuriuos masyvo $tvarka[i][j][k]$ perstatymus.
Norminius ribojimus (žiūr. skyrelis 3.2) įvertinam baudos funkcijoms, nes tai mums parodo "atstumą" iki leistino norminio sprendinio. Tada, spresdami uždavinį 3 artėsim prie leistino sprendinio.