

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ TECHNOLOGIE
Ústav informatizácie, automatizácie a matematiky

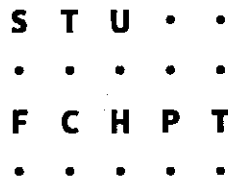
**POROVNANIE PROJEKTOV PRE RIEŠENIE GENEROVANIA
ROZVRHU HODÍN**

BAKALÁRSKA PRÁCA

Jana Kunovská

Študijný program: Automatizácia, informatizácia a manažment v chémii
a potravinárstve
Vedúci bakalárskej práce: prof. Dr. Ing. Miroslav Fikar
Konzultant: RNDr. Štefan Boor

Bratislava 2009



ZADANIE BAKALÁRSKEJ PRÁCE

Študentka: **Jana Kunovská**
ID študenta: 40483
Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve
Kombinácia študijných odborov: 5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo
Vedúci práce: prof. Dr. Ing. Miroslav Fikar
Konzultant: Štefan Boor, RNDr.
Miesto vypracovania: Bratislava

Názov práce: **Porovnanie projektov pre riešenie generovania rozvrhu hodín**

Špecifikácia zadania:

Projekt sa zaoberá literárnym prehľadom dostupných riešení a projektov zaoberajúcich sa tvorbou rozvrhov hodín. Vybrané produkty budú porovnané na vzorke reálnych podkladov pre FCHPT.

1. Literárny prehľad problematiky
2. Optimalizačné metódy riešenia generovania rozvrhu
3. Aplikácia vybraných softvérových riešení na rozvrh hodín na FCHPT.

Zoznam odbornej literatúry:

1. BRUNOVSKÁ, A. *Malá optimalizácia : Metódy, programy, príklady*. Bratislava: Alfa, 1990. 244 s. ISBN 80-05-00770-1.
2. KVASNIČKA, V. – POSPÍCHAL, J. – TIŇO, P. *Evolučné algoritmy*. Bratislava: STU v Bratislave, 2000. 215 s. ISBN 80-227-1377-5.
3. SEKAJ, I. *Evolučné výpočty a ich využitie v praxi*. Bratislava: IRIS, 2005. 157 s. ISBN 80-89018-87-4.

Riešenie zadania práce od: 16. 02. 2009

Dátum odovzdania práce: 22. 05. 2009



Jana Kunovská
študentka

prof. Dr. Ing. Miroslav Fikar
vedúci pracoviska

prof. Dr. Ing. Miroslav Fikar
garant študijného programu

ABSTRAKT

Cieľom tejto práce je oboznámiť sa s problematikou tvorby a generovania rozvrhov a nájsť optimálne riešenie pre našu fakultu spĺňajúce všetky požiadavky oddelení a študijných programov.

Práca je rozdelená na dve časti a to teoretickú a praktickú. Teoretická časť sa zaoberá základnými pojmami súvisiacich s touto problematikou, metódami, ktoré sa využívajú pri generovaní rozvrhov a aplikáciou genetických algoritmov na danú problematiku. V praktickej časti sú opísané podmienky a požiadavky potrebné na vytvorenie rozvrhu, samozrejme aplikácia v konkrétnych programoch a ich vzájomné porovnanie.

ABSTRACT

The aim of this project is to study problems of schedule generation and to find optimal solution for our faculty satisfying requirements of all departments and study programs.

Project is divided into two parts – theoretical and practical. Theoretical part deals with basic concepts, methods that are used for schedule generation and application of genetic algorithm in these problems.

The practical part describes conditions and requirements needed for schedule creation and application in concrete programs with their comparison.

Pod'akovanie

Týmto by som chcela poďakovať vedúcemu bakalárskeho projektu prof. Dr. Ing. Miroslavovi Fikarovi za cenné rady, pripomienky a vedenie, ktoré mi poskytol pri vypracovaní bakalárskej práce.

OBSAH

1	ÚVOD	8
2	TEORETICKÁ ČASŤ	9
2.1	PROGRAMOVANIE S OBMEDZENIAMÍ	9
2.2	HEURISTICKÉ ALGORITMY	10
2.3	TEÓRIA ZLOŽITOSTI A P, NP PROBLÉMY	11
2.4	ALGORITMY	12
2.4.1	<i>Backtracking</i>	12
2.4.2	<i>Genetické algoritmy</i>	14
2.5	APLIKÁCIA GENETICKÝCH ALGORITMOV NA PROBLEMATIKU ROZVRHOV	16
3	PRAKTICKÁ ČASŤ	20
3.1	NAŠE PODMIENKY A DEFINOVANIE PROBLÉMU	20
3.2	ASC	22
3.3	FET	27
3.4	WISE TIMETABLE	32
3.5	VÝSLEDKY	34
4	ZÁVER	36
	POUŽITÁ LITERATÚRA	37
	PRÍLOHA	39
	ZOZNAM KRÚŽKOV	39
	ZOZNAM MIESTNOSTÍ	39
	ZOZNAM PREDMETOV	40
	VÝSLEDKY ASC	42
	VÝSLEDKY FET	44
	ZOZNAM PROGRAMOV	45

1 ÚVOD

V dnešnej dobe je výpočtová technika nevyhnutnosťou a už dávno sa stala súčasťou našich životov. Pomáha nám zjednodušiť a uľahčiť život pri mnohých nudných a zdĺhavých činnostiach. Jednou z nich je aj problematika generovanie rozvrhov, ktorou sa zaoberá táto bakalárska práca.

Každá škola, resp. fakulta rieši problémy súvisiace s generovaním rozvrhov. Niektoré majú zakúpené programy, naprogramované vlastné programy alebo rozvrhy robia ručne, čo je prípad aj našej fakulty. S ručným riešením rozvrhov sú však spojené aj mnohé problémy a ťažkosti – časová náročnosť a vysoká chybovosť. Rozvrh sa vyrába aj niekoľko týždňov, aby splnil všetky požiadavky katedier, pričom veľmi ľahko môže vzniknúť nová chyba. Program tieto problémy eliminuje. Do každého programu, ktorý rieši danú problematiku, stačí zadať základné informácie, správne zadefinovať podmienky a rozvrh je v priebehu pár minút vygenerovaný. Programy sa líšia najmä odlišnými školskými systémami v jednotlivých krajinách. Preto treba nájsť ten optimálny program, ktorý bude vedieť splniť naše požiadavky.

Práca pozostáva z dvoch základných častí - teoretickej a praktickej. V teoretickej časti opisujeme problematiku programovania s obmedzeniami, algoritmy, ktoré sa najčastejšie využívajú pri riešení daných problémov a aplikáciu genetických algoritmov na problematiku rozvrhov. V praktickej časti definujeme naše podmienky potrebné k vygenerovaniu rozvrhov, venujeme sa priamej aplikácii konkrétnych programov na našu fakultu a porovnaniu jednotlivých programov.

2 TEORETICKÁ ČASŤ

2.1 Programovanie s obmedzeniami

Programovanie s obmedzeniami (constraint programming, CSP) sa využíva na hľadanie kombinatorických problémov, kde vzťahy medzi premennými sú vo forme obmedzení. [1] Základnou ideou je, že používateľ stanoví pravidlá a program vyrieši daný problém. Problém danej úlohy, ktorá sa ide riešiť, pozostáva z množiny premenných veličín a množiny vzťahov medzi premennými veličinami.

V našom prípade riešime rozvrh na fakulte, kde premennými sú krúžky, miestnosti a predmety. Obmedzeniami sú kapacity a zariadenie tried, časové obmedzenia, kedy sa určité predmety nemôžu vyučovať a mnohé ďalšie špecifické požiadavky vyučujúcich. Solver vezme tento problém a snaží sa nájsť určité riešenie pre všetky premenné, ktoré vyhovujú daným obmedzeniam. Ďalšími funkciami solvera môže byť napríklad nájdenie optimálneho riešenia podľa jedného alebo viacerých kritérií alebo nájdenie všetkých riešení. Solver môže pri riešení prehľadávať stavový priestor systematicky, napr. algoritmom backtracking alebo algoritmom vetiev a hraníc (branch and bound), alebo používa lokálne vyhľadávanie, ktoré môže byť neúplné.

Základnou algoritmicou technikou pre nájdenie uspokojivého riešenia v definovaných obmedzeniach je vyhľadávanie. Vyhľadávací algoritmus pre riešenie uspokojivého riešenia daného obmedzeniami (CSP, Constraint Satisfaction Problem) môže byť kompletný alebo nekompletný. Kompletné riešenie alebo systematický algoritmus garantuje, že ak dané riešenie existuje, tak bude nájdené. Tento systematický prístup sa môže použiť na dôkazy, že CSP nemá riešenie a nájsť pravdepodobne optimálne riešenie. Nekompletné, nesystematické prehľadávanie nemôže byť použité ako dôkaz, že daný CSP nemá riešenie. Avšak tieto nesystematické algoritmy sú často efektívne v hľadaní riešenia, ak existuje a môžu byť použité na nájdenie riešenia, ktoré sa približuje optimálnemu riešeniu. Backtracking hľadací algoritmus je príkladom systematického hľadania, algoritmus lokálneho hľadania je zase príkladom nekompletného algoritmu.

Programovanie s obmedzeniami je veľmi nápomocné pre použitie v priemysle, obchode, výrobe a vede.

2.2 Heuristické algoritmy

Heuristické alebo tiež stochastické algoritmy sa začali používať s príchodom výpočtovej techniky. [2] Informatika ako veda o informáciách a ich spracovaní chápe heuristiku ako postup pri získavaní riešenia problému, ktoré však nie je presné a nemusí byť nájdené v krátkom čase. Slúži najčastejšie ako metóda rýchlo poskytujúca dostatočné a dosť presné riešenie, ktoré sa však nedá všeobecne dokázať. Najčastejšie použitie algoritmu nájdeme v prípadoch, kde nie je možné použiť lepší algoritmus. Mohli by sme povedať, že heuristikou nazývame postup, ktorý, aj keď nepreberie všetky možnosti, je schopný v niektorých prípadoch nájsť výsledok.

Vo všeobecnosti sa heuristické algoritmy využívajú pri optimalizácii zložitých funkcií s mnohými parametrami a zložitým priebehom s mnohými extrémami.

Deterministické algoritmy sú pri riešení zložitých problémov veľmi zle použiteľné, pretože ich náročnosť (hlavne časová) s lineárne rastúcim rozsahom problému rastie exponenciálne. Naopak výhodou oproti heuristickým algoritmom je nájdenie presného, optimálneho riešenia, ktoré sa dá dokázať.

Medzi často používané metódy heuristiky patria:

- genetický algoritmus – založený na princípe prirodzenej selekcie
- backtracking – prehľadávanie s návratom
- metóda lokálneho vyhľadávania – ide o jednu z najjednoduchších metód; je dobre použiteľná na riešenie matematických problémov
- iteratívna metóda – využíva postupné hľadanie riešení v stále sa zužujúcej oblasti riešenia (prepracúva sa z dobrého riešenia k lepšiemu riešeniu)
- hladný algoritmus – heuristika pre prehľadávanie určí, ktoré z možných pokračovaní si vyberieme

Heuristických metód je ďaleko viac, pričom každá má svoje klady a zápory. Medzi klady zaradujeme možnosť nájdenia presného riešenia, medzi zápory čas, ktorý je potrebný k prevedeniu algoritmu, prípadne množstvo upresňujúcich parametrov.

U niektorých metód istotu, či je výsledok presný, môžeme zvýšiť opakovaním heuristického algoritmu.

2.3 Teória zložitosti a P, NP problémy

- **Zložitosť**

Neodmysliteľnou súčasťou modernej algoritmickej teórie grafov je rozbor obťažnosti riešeného problému a stanovenie výpočtovej zložitosti navrhnutého algoritmu. [3,4] Týmito záležitosťami sa zaoberá teória výpočtovej zložitosti. Ukázalo sa rozumné posudzovať algoritmy podľa počtu elementárnych krokov, ktoré sú potrebné na vyriešenie daného problému. Tieto elementárne kroky môžu byť sčítanie, odčítanie, násobenie, delenie, porovnávanie s vetvením atď. Jednotlivé elementárne kroky považujeme za rovnako časovo náročné. Počet krokov algoritmu môže závisieť nielen od množstva vstupných dát, ale aj od ich vzájomnej konfigurácie.

Výpočtová zložitosť algoritmu priamo nesúvisí so zložitou samotného algoritmu z ľudského pohľadu. Algoritmus môže byť veľmi jednoduchý na pochopenie i implementáciu, napriek tomu môže byť jeho výpočtová zložitosť veľká a naopak.

Poznáme niekoľko druhov výpočtovej zložitosti v závislosti od počtu vstupných prvkov n , (a, b, c sú konštanty, t je čas):

- lineárna zložitosť: $t = a \cdot n + b$
- logaritmicko-lineárna: $t = a \cdot n \cdot \log n + b \cdot n + c$; $O(n \cdot \log n)$
- polynomiálna: t je funkciou nejakého polynómu n ; $O(n^2)$
- algoritmy so zložitou väčšou než je polynomiálna (exponenciálne)

Za rýchle algoritmy sa pokladajú prvé tri druhy.

V roku 1965 sa prvýkrát definoval rozdiel medzi polynomiálnymi algoritmi (t.j. algoritmi so zložitou typu $O(n^k)$) a nepolynomiálnymi algoritmi, ktorých počet krokov

nevieme ohraničiť polynómom. Tie prvé sa nazývajú „dobré“. Aj problémy možno rozdeliť na také, pre ktoré existuje polynomiálny algoritmus a tie ostatné – „beznádejné“, pre ktoré takýto algoritmus neexistuje. Túto ideu sa zatiaľ nepodarilo plne realizovať, pretože neexistencia polynomiálneho algoritmu sa pre väčšinu problémov považovaných za „beznádejné“ nepodarila dokázať.

- **P, NP problémy**

Poznáme P a NP problémy. [5,6] P problémy predstavujú deterministické problémy riešené v polynomiálnom čase. NP sú nedeterministické problémy riešené v polynomiálnom čase, pričom algoritmus je nedeterministický, ak pri riešení úlohy dokáže na jeden stav reagovať viac ako jedným spôsobom. Ak navyše predpokladáme, že reaguje vždy tým správnym, môžeme jeho správanie interpretovať ako kreatívnu činnosť.

Momentálne najväčším matematickým problémom je, či sa P a NP problémy rovnajú. Ak sa nerovnajú, potom riešenie NP-problému vyžaduje vyčerpávajúce hľadanie správneho algoritmu.

Problém je zasadený do P (polynomiálneho času), ak existuje aspoň jeden algoritmus na vyriešenie problému, pričom počet krokov algoritmu je závislý od stupňa polynómu.

Problém sa nazýva NP-hard, ak algoritmus na riešenie problému vieme prepísať na už existujúci/známy NP-problém. V podstate to znamená, že NP-hard je aspoň tak ťažký ako NP-problém, pričom môže byť aj ťažší.

2.4 Algoritmy

2.4.1 Backtracking

Backtracking hľadanie (prehľadávanie do hĺbky, spätné prehľadávanie) je algoritmus, ktorý v štruktúre stromu možných riešení hľadá riešenie postupne v jednotlivých uzloch tak, že najprv ide do hĺbky prehľadávaného stromu až potom do šírky. [1,7] Ide vlastne o hľadanie riešenia pomocou metódy pokusov a omylov. Celý proces sa rozkladá na niekoľko čiastkových úloh, ktoré sa často vyjadrujú pomocou rekurzie a spočívajú

v preskúmaní konečného počtu podúloh. Opakovaním pokusov (vyhľadávaní) sa postupne vybuduje a skúma (zmenšuje) strom podúloh. Technika sa volá prehľadávanie s návratom, pretože po každom navštívenom liste stromu sa algoritmus vráti do zásobníka. Tam má uložené miesta, odkiaľ sa vnoril. Vyberie ďalšiu vetvu a znova sa vnorí. Aby sa problém dal riešiť pomocou prehľadávania s návratom, jeho podproblémy sa musia určitým spôsobom podobať na pôvodný problém.

Backtracking je vo všeobecnosti neefektívny prístup, používajúci prístup hrubou silou (brute-force). Algoritmus sa využíva na riešenie veľkého množstva problémov, avšak kvôli jeho časovej zložitosti sa používa len vtedy, keď nepoznáme efektívnejší algoritmus alebo sa požíva na malé dáta, poprípade existuje dobrá heuristika.

Pre zvýšenie výkonnosti backtracking hľadania bolo navrhnutých a urobených niekoľko vylepšení tohto algoritmu. Niektoré z najdôležitejších techník obsahujú šírenie obmedzení (angl. constraint propagation), nogood recording, spätný krok (angl. backjumping), heuristiky pre premiešavanie premenných, náhodnosť a reštart stratégií. Výsledkom najlepších kombinácií z týchto techník je backtracking algoritmus, ktorý môže teraz rutinne riešiť veľké a kombinatoricky náročné prípady rôznej náročnosti.

- **Nogood recording**

Jednou z najúčinnějších techník pre zlepšenie výkonnosti algoritmu backtracking na CSP je pridanie skrytých obmedzení (angl. implied constraint) alebo nogoods. Obmedzenie je skryté, ak množina riešení CSP je rovnaká, ako by bola s alebo bez daného obmedzenia. Nogood je špeciálnym typom skrytých obmedzení, je to rad úloh pre niektoré podmnožiny premenných, ktoré nevedú k riešeniu. Pridaným toho „správneho“ skrytého obmedzenia do CSP môže znamenať, že veľa slepých vetiev riešenia je odstránených z vyhľadávacieho stromu a ďalšie slepé vetvy sú objavené oveľa skôr a s menším úsilím ako by to bolo bez tohto prístupu.

- **Náhodnosť a reštart stratégií**

Bolo všeobecne pozorované, že backtracking algoritmus môže mať problémy pri niektorých prípadoch. Zdanlivo malé zmeny na premenných alebo zmenená heuristika na

určovanie poradia hodnôt môže priniesť obrovské rozdiely v čase potrebnom na riešenie úlohy. Vysvetlením pre tento fenomén je, že heuristika zodpovedná za poradie robí chyby. Riešením môžu byť techniky ako náhodnosť (angl. randomization) a reštart stratégií (angl. restart strategies). Reštart stratégia $S = (t_1, t_2, t_3, \dots)$ je nekonečná sekvencia, kde každé t_i je či už kladné celé číslo alebo nekonečno. Ideou je, že náhodný backtracking algoritmus ide t_1 krokov. Ak sa riešenie nenájde, algoritmus je reštartovaný a beží pre t_2 krokov, pokiaľ sa nenájde riešenie.

2.4.2 Genetické algoritmy

Genetické algoritmy patria do triedy evolučných algoritmov, ktoré okrem nich zahrňujú aj evolučné programovanie, evolučnú stratégiu a genetické programovanie. [8,9,10,11] Sú to vyhľadávacie algoritmy vychádzajúce z Darwinovej evolučnej teórie. Používajú mechanizmus prirodzeného výberu a princíp genetiky. Ide o aplikáciu princípov, ktoré využíva príroda už niekoľko tisícročí. Ich veľkou výhodou je jednoduchosť. Na jednotlivé prvky množiny vhodných riešení pozeráme ako na živé organizmy v umelom životnom prostredí. Samotné hľadanie spočíva vo výbere pôvodnej populácie zloženej z jedincov s rôznymi vlastnosťami. Tieto vlastnosti sú zakódované v ich génoch, ktoré tvoria väčšie celky, chromozómy a tie tvoria danú populáciu. Nasledujú simulácie vývoja pomocou evolučných mechanizmov zahrňujúce: reprodukciu, kríženie a mutáciu. Pri krížení vznikajú noví jedinci, ktorí majú spravidla náhodne časť génov od jedného rodiča a časť génov od druhého rodiča. Vo výnimočných prípadoch môže dôjsť k náhodnej zmene niektorého génu v chromozóme, tzv. mutácii, ktorá môže, ale nemusí byť pre ďalší vývoj druhu pozitívna. Ako sa populácia vyvíja, vznikajú potomkovia, ktorých môžeme definovať ako dobré a zlé riešenia. Zlé riešenia „vymierajú“, dobré sa medzi sebou krížia. Proces výberu sa stále opakuje a v jeho priebehu sa zlepšujú genetické vlastnosti daného druhu.

V teórii umelej inteligencie je genetický algoritmus proces postupného zlepšovania populácie jedincov opakovanou aplikáciou genetického algoritmu, ktorý vedie k evolúcii takých jedincov, ktorí lepšie vyhovujú stanoveným podmienkam než jedinci, z ktorých vznikli. Proces smeruje k situácii, keď je populácia tvorená len tými najlepšimi jedincami. Hlavným princípom je kopírovanie a menenie reťazcov – chromozómov. Chromozóm má pevnú dĺžku,

jednotlivé pozície tvoria gény. Gény reprezentuje často binárna 0 alebo 1, ale všeobecne môžu mať ľubovoľnú hodnotu. Každá populácia sa v procese reprodukcie zdokonaľuje a to na základe ohodnotenia chromozómov pomocou funkcie $f(x)$ nazývanej fitness. V procese riešenia pomocou genetických algoritmov ide o hľadanie globálneho maxima funkcie fitness, teda ide o to nájsť najlepšie ohodnotené riešenie problému v stavovom priestore.

- **Reprodukcia**

Operátor reprodukcie kopíruje chromozómy do novej populácie. Chromozómy s vyššou hodnotou fitness sú do novej populácie kopírované s vyššou pravdepodobnosťou. Pravdepodobnosť kopírovania je daná vzorcom

$$P = \frac{\Phi}{\sum f}$$

P – pravdepodobnosť reprodukcie i -tého chromozómu

Φ – hodnota fitness funkcie i -tého chromozómu

$\sum f$ – súčet všetkých hodnôt fitness funkcie v populácii

- **Kríženie**

Na urýchlenie optimalizačného procesu sa výber rodičovských chromozómov opäť nevykonáva celkom náhodne, ale pomocou heuristického hodnotenia chromozómov v populácii. Navyše pokiaľ by sme v priebehu kríženia prevzali časť plánu z jedného chromozómu a časť z druhého, niektoré položky by sa objavili vo výslednom pláne viackrát a niektoré by chýbali.

Preto je definovaná funkcia fitgén, ktorá zhodnotí, či sa pri krížení určitý gén ešte môže do chromozómu použiť alebo nie. Postup výberu génov zodpovedá heuristickému hodnoteniu ich kvality. Po vyčerpaní všetkých génov z oboch chromozómov je potomok doplnený podľa pravdepodobnostnej funkcie tak ako pri mutácii.

Kríženie prakticky znamená vytváranie nových jedincov podľa nasledujúceho pravidla: Z populácie vyberáme náhodné páry a náhodne zvolíme pozíciu k . Do prvého potomka skopírujeme gény 1 až k prvého rodiča a gény $k + 1$ až n druhého rodiča, kde n je počet génov, a do druhého potomka kopírujeme gény opačne.

- **Mutácia**

Pri aplikácii operátora mutácie vyberieme s malou pravdepodobnosťou náhodný gén v náhodnom chromozóme a zmeníme jeho hodnotu z 0 na 1 alebo naopak.

Hlavné výhody GA:

- nevyžadujú žiadne špeciálne znalosti o cieľovej funkcii
- sú odolné voči skĺznutiu do lokálneho optima
- vykazujú veľmi dobré výsledky pri problémoch s rozsiahlymi množinami prípustných riešení
- môžu byť využité pre najrozmanitejšie optimalizačné problémy

Hlavné nevýhody GA:

- majú problém s nájdením presného optima
- vyžadujú veľké množstvo vyhodnocovania účelovej funkcie
- ich implementácia nie je vždy priamočiara

V praxi sa pomocou genetických algoritmov riešia úlohy optimalizácie, využívajú sa na vyhľadávanie najlepšej topológie, v technológii a výrobe a v priemyselnej automatizácii a ako alternatívne metódy učenia neurónových sietí. Na rozdiel od gradientových metód, ktoré reprezentujú hľadanie lokálneho minima alebo maxima pomocou jedného spresňujúceho sa riešenia, predstavujú genetické algoritmy iný prístup. Systém musí byť zložitejší ako pri jednoduchých úlohách, kedy môžeme využiť lineárne programovanie. Musí obsahovať nelinearity a veľa podmienok a obmedzení, aby algoritmus bol schopný vyhodnocovať svoj vývoj. Tým sa dosahuje to, že populácia jedincov nájde správne riešenie relatívne rýchlo.

2.5 Aplikácia genetických algoritmov na problematiku rozvrhov

Keď robíme rozvrh, musíme brať do úvahy veľa požiadaviek – počet profesorov, študentov, predmetov a miestností, kapacitu a zariadenie tried a veľa ďalších. [12] Tieto požiadavky môžeme rozdeliť do niekoľkých skupín podľa dôležitosti.

Základné požiadavky (ak nebudú splnené, rozvrh nemôžeme vygenerovať):

- krúžok môže byť umiestnený len do voľnej miestnosti
- žiadny profesor alebo krúžok nemôže mať naraz viac ako jeden predmet
- miestnosť musí mať dostatočnú kapacitu
- miestnosť musí spĺňať požiadavky na zariadenie pre daný predmet

Požiadavky, ktoré nemusia byť 100%tne dodržané:

- uprednostňovaný čas výučby
- uprednostňované miestnosti
- časové a priestorové rozdelenie predmetov krúžkom a profesorom

Všetky tieto požiadavky závisia od konkrétnej situácie. V tomto prípade sú implementované len základné požiadavky.

Čo musíme vedieť a poznať?

Profesori:

V prípade, že potrebujeme triedam priradiť aj konkrétneho profesora, potrebujeme zadať jeho meno a zoznam predmetov, ktoré učí.

Krúžky:

Musíme vedieť počet študentov, poprípade aj mená a samozrejme zoznam predmetov, ktoré navštevujú.

Miestnosti:

Musíme poznať kapacitu triedy, zariadenie, prípadne názov triedy nám stačí na identifikovanie predmetov, ktoré sa tam učia.

Predmety:

Predmety potrebujeme priradiť k jednotlivým krúžkom, určíme vhodnú učebňu s nutným zariadením a kapacitou. Ešte môžeme priradiť profesorov, ktorý daný predmet vyučujú.

Chromozómy:

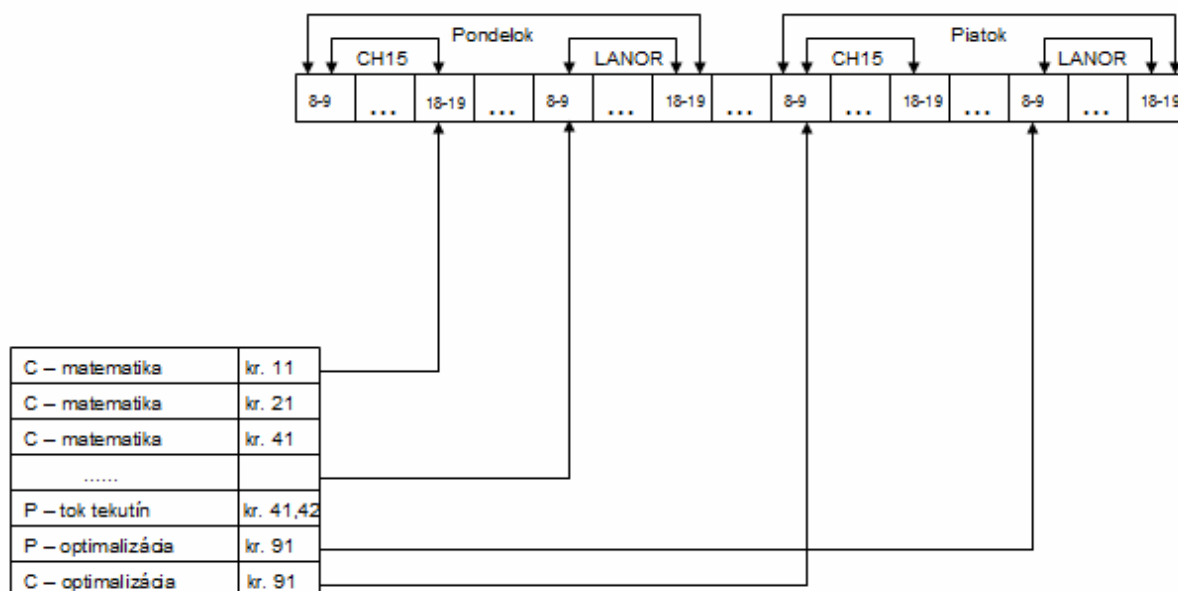
Prvú vec, ktorú musíme zvážiť pri genetických algoritmoch, je ako dostať zadanie do takej formy, aby sme mohli využívať genetické operácie ako kríženie a mutáciu. Tiež musíme

vedieť ako určiť, či je výsledok dobrý alebo zlý. Inými slovami, mali by sme vedieť vypočítať hodnotu fitness funkcie nášho výsledku.

Zápis:

Ako zapísať chromozómy pre rozvrh? Potrebujeme mať počet voľných hodín pre každú miestnosť na každý deň. Dajme tomu, že nemôžeme začínať pred 8 ráno a končíme najneskôr o 19 večer, 5 dní v týždni. Dostaneme vektor $11 \cdot 5 \cdot \text{počet miestností}$ (obr. 1).

Na obrázku vidíme, že máme vektor pozostávajúci z dní, hodín a miestností a zároveň zoznam predmetov, pričom každý predmet má samostatný vstup do vektoru.



Obr. 1: Zápis chromozómu

Fitness

Teraz potrebujeme zapísať hodnotu fitness funkcie chromozómu. Ako bolo už hore spomenuté, iba základné požiadavky sa zvyknú počítať cez fitness funkciu. Robíme to nasledovne:

- každý predmet môže mať od 0 do 5 bodov
- ak predmet používa voľné triedy, dostaneme bod

- ak predmet požaduje počítače a môžeme ho dať do tejto triedy, poprípade počítače nepotrebuje, dostaneme bod
- ak predmet je umiestnený do triedy s postačujúcou kapacitou, máme bod
- ak profesor má len jednu triedu v jednu hodinu, opäť dostane náš predmet bod
- ďalší bod dostaneme, ak krúžky nemajú naraz viac ako jeden predmet
- ak sa poruší nejaké pravidlo, nedostávame bod
- celkový počet rozvrhov je suma bodov všetkých predmetov
- hodnota fitness funkcie sa počíta $\frac{\text{pocet_rozvrhov}}{\text{maximalny_pocet_bodov}}$, pričom $\text{maximalny_pocet_bodov} = \text{pocet_tried} * 5$

Hodnota fitness funkcie je teda v rozsahu od 0 do 1

Kríženie

Kríženie je operácia, ktorá kombinuje informácie z chromozómu od dvoch rodičov a vytvára tak nový chromozóm. Kríženie „roztrhne“ chromozóm oboch rodičov na niekoľko častí rôznych veľkostí. Počet častí je definovaný pomocou počtu krížení plus jedna. Potom kopírujeme časti rodičov do nového chromozómu a vytvoríme tým nový vektor.

Mutácia

Mutácia je veľmi jednoduchá operácia. Náhodne vyberieme predmet a presunieme ho na náhodne vybrané miesto. Počet predmetov, ktoré sa presúvajú v jednej operácii je definovaný veľkosťou mutácie v chromozóme.

Algoritmus

Genetický algoritmus je pomerne jednoduchý. Pre každé generovanie to predstavuje dve základné operácie:

1. Krížením vytvoríme N nových chromozómov z náhodne vybraných N párov rodičov súčasnej populácie.
2. Náhodne vyberieme N chromozómov zo súčasnej populácie a premiestnime ich do novej.

Tieto dve operácie opakujeme, kým nedosiahneme najlepší chromozóm, kedy hodnota fitness funkcie sa rovná 1. Genetický algoritmus garantuje, že najlepšie chromozómy v populácii sa nebudú presúvať.

3 PRAKTICKÁ ČASŤ

3.1 Naše podmienky a definovanie problému

Na začiatku bolo treba určiť cieľ práce a z toho vyplývajúce podmienky pre rozvrh. Aby sme mohli jednotlivé rozvrhy porovnávať, podmienky museli byť rovnaké alebo aspoň veľmi podobné v rámci možností jednotlivých programov.

Cieľom bolo vytvoriť rozvrh pre prvý ročník našej fakulty tak, aby sa vyučovalo od pondelka do piatku. Pondelok až štvrtok od 8:00 do 19:00, piatok iba do 13:00. Po 17:00 bol rozvrh len vo výnimočných prípadoch a to, keď to vyžadoval profesor alebo boli dlhé laboratórne cvičenia. Prednášky mohli byť až od 9:00. Nepretržite za sebou mohlo byť maximálne 5 vyučovacích hodín. Maximálny počet hodín za deň sme obmedzili na 10. A samozrejme nevadil nám voľný deň.

K dispozícii máme 14 krúžkov (viď Príloha – Zoznam krúžkov) zo siedmich programov, pričom v každom sa počíta s 20 študentmi. Počet tried (viď Príloha – Zoznam miestností) sme obmedzili na nevyhnutne potrebné (laboratória), tri prednáškové a 4 cvičiace. Kapacita v laboratóriách nezodpovedá reálnemu stavu. Dôvodom je, že laboratórne cvičenia nezodpovedajú podmienkam, ktoré im určujú oficiálne stanovy. Keďže väčšinou sa vyučuje viac hodín (namiesto troch napr. 5) a semester sa skôr ukončí. Napríklad: Kapacita laboratória anorganickej chémie zodpovedá 30 študentom, čo je v našom prípade jeden a pol krúžku. V reálnej situácii by to riešilo tak, že by sa hľadali krúžky, ktoré majú málo študentov, prípadne by to záviselo od dohody (prvého polsemestra by mala laboratórne cvičenie polovica krúžku, druhý polsemester zvyšok krúžku). Na zjednodušenie situácie sme počítali s kapacitou 40 študentov, čo sa rovná celým dvom krúžkom.

Predmety (viď Príloha – Zoznam predmetov) a ich rozsah máme z oficiálnej knižky. Mnohé z nich majú rôzne kódy a počet kreditov, napriek tomu majú spoločné prednášky (výborným príkladom je predmet Základy všeobecnej ekonomickej teórie, ktorý má 5 kódov, avšak všetky prednášky môžu byť spojené). Tieto informácie sme získali z aktuálneho rozvrhu prvákov.

Požiadavky katedier pre prvý ročník:

- programovanie - prednášky a cvičenia v utorok, stredu alebo štvrtok od 17:00
- fyzika – dve cvičenia naraz
- chemicko-inžinierske výpočty na pc – súbežne
- informačné technológie v potravinárstve – prednáška a cvičenie po sebe
- optimalizácia – prednáška v pondelok od 16:00

Na začiatku sme mali k dispozícii vyše 50 programov (viď Príloha – Zoznam programov) zaoberajúcich sa rôznymi typmi rozvrhov. Postupným vyradovaním nevhodných programov, nám zostalo 13 aplikácií. Dôvodom vylúčenia programov bolo niekoľko:

- nepodarilo sa nám otvoriť stránku
- program sa netýkal školských rozvrhov, ale napr. plánovaním konferencií
- nebola dostupná aspoň anglická verzia
- išlo o čisto komerčné programy, ktoré sa dali len zakúpiť

Programy, ktoré nám ostali, mali k dispozícii demo verzie, ktoré sa dali voľne stiahnuť na internete, jeden program generuje rozvrhy online. Od programu sme vyžadovali:

1. demo verziu, do ktorej sme mohli zadať vlastné dáta v neobmedzenom množstve
2. možnosť zadať miestnosti, krúžky, predmety, prípadne profesorov a definovať vzťahy medzi nimi
3. možnosť spájania tried (dôležité pre zadávanie prednášok)

Tieto podmienky plne splnili len dva programy – ASC a FET, najzaujímavejší bol však WiseTimetable. U niektorých bol problém už s inštaláciou (Open Course Timetabler, Mimosa, Seasia). Tri programy (TTmaker, iMagic Timetable, Visual Classroom) sú z Austrálie a nespĺňajú našu druhú podmienku. Dôvodom je, že zodpovedajú ich školskému systému, čím sa stávajú pre nás nevyužiteľné. Program School-timetable, ktorý pracuje online, nie je

schopný splniť tretiu podmienku. Zvyšné programy ako Orologio, GP-untis, The TimetablerTimetabler majú obmedzené množstvo vstupných dát, resp. sa pracuje s dátami, ktoré sú už preddefinované.

Viackrát budeme v tejto práci používať pojem aktivita. Predstavuje nám konkrétny predmet, ktorý má už priradený krúžok, skupinu miestností, v ktorých sa môže vyučovať, poprípade vyučujúceho. Napr. z predmetu C – fyzika vytvoríme 14 aktivít pre jednotlivé krúžky.

Nižšie si opíšeme a porovnáme tri programy, ktoré najlepšie splnili naše požiadavky: slovenský ASC, rumunský FET a slovinský WiseTimetable.

3.2 ASC

ASC je komerčný celosvetovo rozšírený program, ktorý sa predáva už v 115-tich krajinách sveta a je dostupný v 35 jazykoch. [13, 14] Program využíva algoritmus backtracking vylepšený mnohými heuristickými metódami. Vyhral niekoľko cien na pedagogických výstavách a dodnes si ho objednávajú školy po celom svete. „Školské oddelenie“ firmy má dnes desať spolupracovníkov, ktorí spoločne vytvorili ďalšie úspešné programy aSc Suplovanie, aSc Agenda a aSc EduPage s E-Learningom. Používajú ich dve tisícky škôl na celom Slovensku. Primárne je určený pre základné a stredné školy so špecifikáciami pre jednotlivé krajiny.

DEMO verzia je voľne prístupná na internete a jej jediné obmedzenie spočíva v tom, že vygenerovaný rozvrh sa nedá vytlačiť, čo si myslím, že nie je žiadnou prekážkou.

Keď spustíme program, prakticky nám sám hovorí, čo máme robiť. Nehovoriac o tom, že si môžeme spustiť ukážku programu s komentárom, ktorá je veľmi rozsiahla a miestami aj vtipná.

Učitelia:

Program nevyžaduje zadávanie profesorov, čo nám uľahčuje prácu. Keby ich bolo treba predsa len zadať, musíme zabezpečiť, aby nás to neobmedzovalo. Dalo by sa to

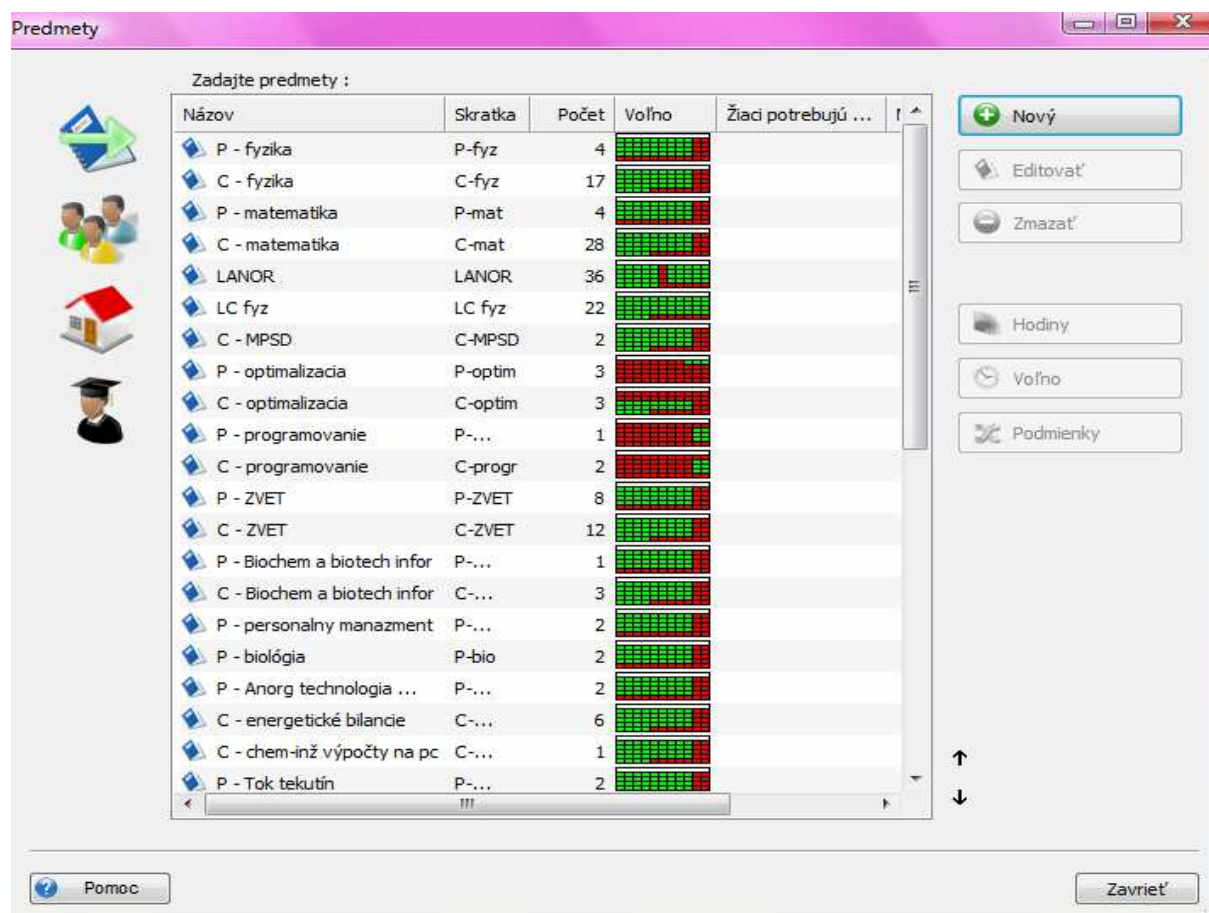
urobiť tak, že nastavíme toľko profesorov, koľko máme aktivít a každej aktivite priradíme iného profesora.

Miestnosti:

Zadefinujeme učebne – názov, skratku. Môžeme vybrať z možností, či je učebňa pre niektorú triedu kmeňová alebo či vyžaduje dozor. Určíme kapacitu miestnosti a hodiny, kedy je obsadená, resp. voľná.

Predmety:

Zadáme meno, skratku. Určíme, kedy sa môže daný predmet vyučovať. Ďalej v možnostiach výberu je, či sa na tento predmet musia alebo nemusia žiaci pripravovať (obr. 2).

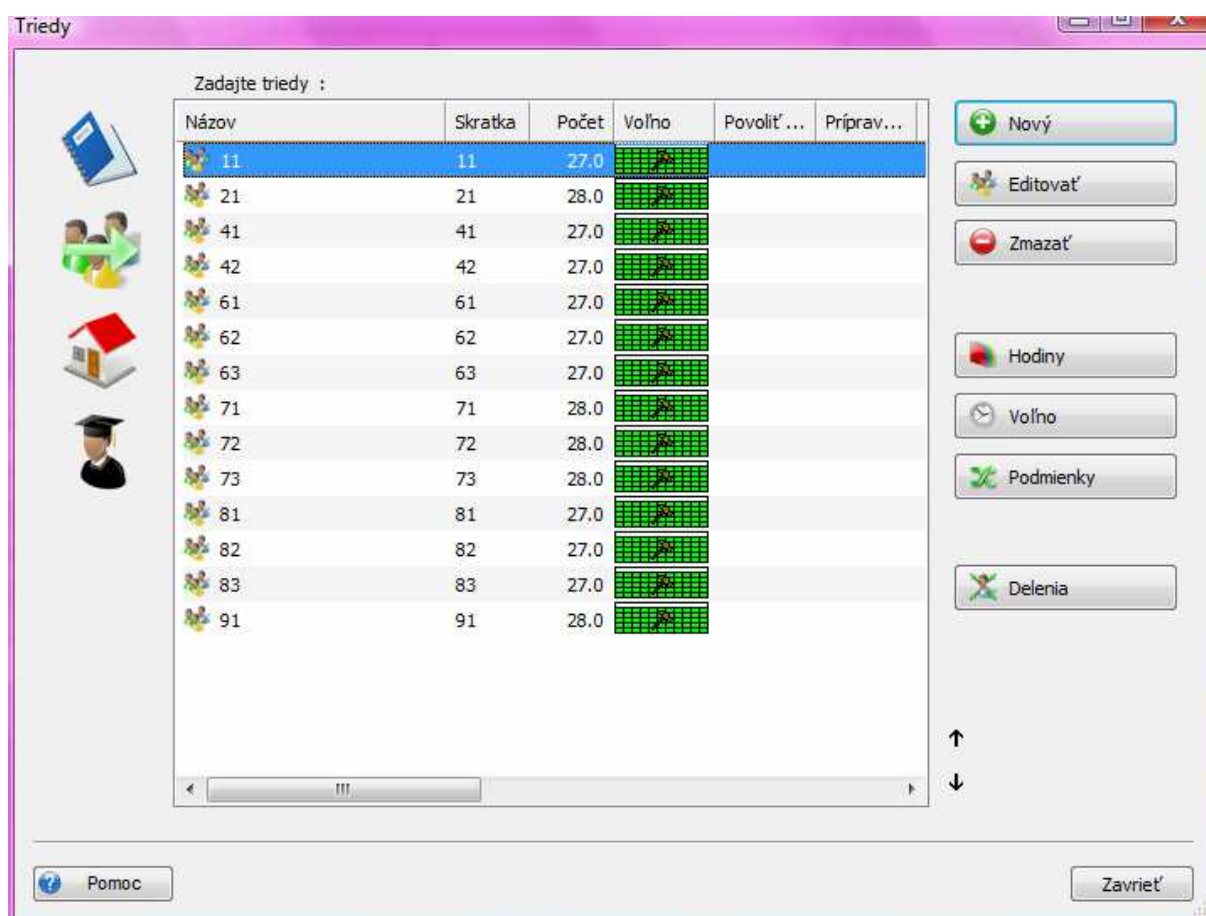


Obr. 2: Zoznam zadaných predmetov

Kružky:

Zadáme meno triedy a skratku (obr. 3). Môžeme určiť, ktorá miestnosť je pre nich kmeňová a kedy nesmú mať vyučovanie (čo má opodstatnenie na základných školách, aby napr. piataci sa neučili siedmu vyučovaciu hodinu).

Kružok môžeme rozdeliť na niekoľko rôznych skupín, či už chceme chlapcov a dievčatá alebo angličtina a nemčina. Generátor bude rozvrh generovať tak, aby sa študentom neprekrývali hodiny.



Obr. 3: Zoznam zadaných krúžkov

Podmienky, ktoré môžeme krúžkom zadať:

- blok vyučovania – nastavuje sa jednotlivým krúžkom a cez možnosť „nastaviť ďalším“, môžeme nastaviť aj ostatným krúžkom

- buď automaticky (prednastavené), čo znamená od prvej, resp. druhej hodiny, bez medzier
- nesúvislý – ignoruje medzery medzi jednotlivými hodinami (naš prípad)
- nastaviť blok vyučovania manuálne – určíme od koľkej do koľkej hodiny, ale vyučuje sa bez medzier

b) minimálny a maximálny počet hodín za deň – v našom prípade minimálny je 0, maximálny 10

c) nastaviť počet dní, kedy má mať trieda vyučovanie – nenastavujeme

d) maximálny počet predmetov vyžadujúcich si prípravu za jeden deň

e) prestávka na obed musí byť v čase od do (my sme si nastavili od 4 do 6 hodiny)

Vytváranie aktivít:

Po stlačení výberu „nová hodina“ sa nám zobrazí okno (obr. 4), v ktorom určíme:

- predmet
- učiteľa/učiteľov, ktorí daný predmet vyučujú
- krúžok, resp. krúžky, ak ide prednášky
- počet hodín za týždeň a koľko hodín naraz sa môže odučiť
- triedy, ktoré sú vhodné pre daný predmet

Takýmto spôsobom vytvoríme všetky aktivity, ktoré už len potrebujú podmienky, aby splnili naše požiadavky.

Obr. 4: Zadávanie aktivít v programe ASC

Vzťahy a podmienky:

Prvým krokom je vybrať predmety a triedy, ktorých sa obmedzenia týkajú a určiť im podmienky, ktoré sa nám ponúkajú. Na dvoch kartách nájdeme všetky možnosti, ktoré tento program poskytuje a to:

- predmety nemôžu byť v jeden deň
- predmety nemôžu byť za sebou v ten istý deň
- dva predmety musia byť v jeden deň
- dva predmety musia nasledovať v špecifickom alebo ľubovoľnom poradí
- prestávka nemôže byť medzi skupinou hodín
- pre skupiny zo zadaných tried musia uvedené predmety začínať naraz

- vybrané predmety musia byť naraz vo všetkých vybraných triedach
- predmet musí byť každý deň v rovnakú hodinu

„Špeciálne“ vzťahy

Špeciálne vzťahy upresňujú požiadavky zadávateľa.

Nájdeme tu:

- minimálny počet vyučovacích hodín za deň vo vybranej oblasti (deň bez vyučovania je v poriadku)
- maximálny počet vyučovacích hodín za deň vo vybranej oblasti
- minimálny počet hodín + okien za deň (deň bez vyučovania je v poriadku)
- maximálny počet hodín + okien za deň
- maximálny počet dier za deň vo vybranej oblasti
- maximálny počet dier za týždeň vo vybranej oblasti
- maximálny počet rôznych učební za deň
- maximálny počet rôznych učební za týždeň
- maximálny počet vyučovacích hodín idúcich po sebe

My sme využili:

- dva predmety musia nasledovať v špecifickom poradí: P – inf. tech. v potravinárstve a C – inf. tech. v potravinárstve
- maximálny počet dier za týždeň vo vybranej oblasti: 3
- maximálny počet vyučovacích hodín idúcich po sebe: 5
- minimálny počet hodín + okien za deň (deň bez vyučovania je v poriadku): 3

Výsledky rozvrhov nájdeme v prílohe – ASC.

3.3 FET

FET je veľmi jednoduchý program s otvoreným kódom založený na bližšie nešpecifikovanom heuristickom algoritme. [15] Najväčšou výhodou je, že sa dá zadarmo stiahnuť na internete a ďalej upravovať. Môžeme ho používať vo viacerých jazykoch vrátane

poľštiny, slovenčina, či čeština zatiaľ bohužiaľ chýba, ale je možné ju tam jednoducho dorobiť.

Zadávať sa základné informácie, vytvoria sa aktivity a dodatočne sa určia všetky potrebné podmienky. Najskôr si zadáme názov školy, počet dní do týždňa a počet hodín za deň.

Učitelia:

Ani v tomto programe nepotrebujeme zadávať profesorov. Keby sme ich predsa len chceli zadávať, stačí napísať meno, požiadavky sa určujú dodatočne.

Miestnosti:

Pri zadávaní miestností určíme len názov a kapacitu.

Označenie aktivity:

Môžeme si nastaviť niekoľko skupín, do ktorých budeme zaeľovať predmety, napr. prednáška, cvičenie, počítače. Výhodné je to vtedy, keď chceme nastaviť rovnaké podmienky všetkým prednáškam (čas vyučovania).

Predmety:

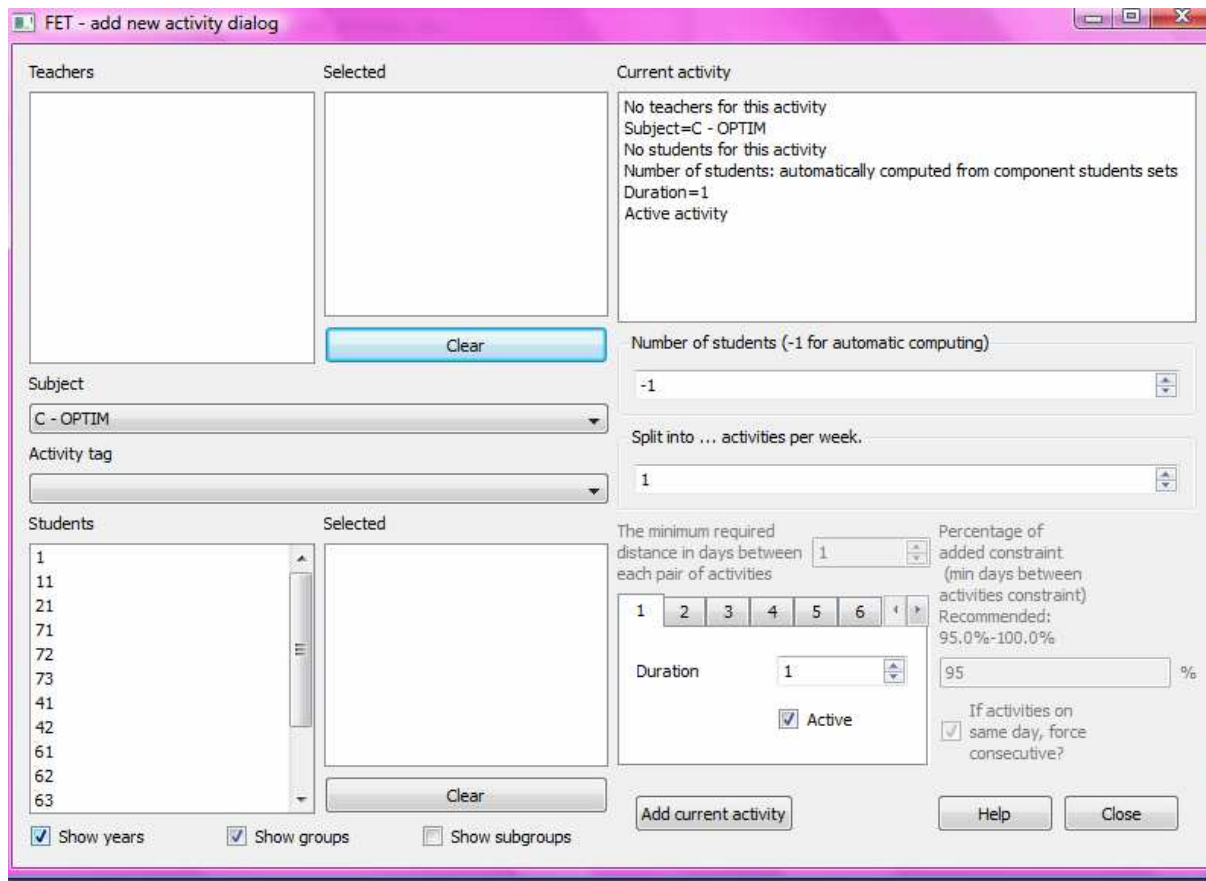
Opäť zadáme len názov predmetu.

Kružky:

Môžeme si určiť ročníky a im priradiť jednotlivé krúžky, poprípade krúžky rozdeliť na skupiny.

Aktivity:

Pri zadávaní aktivít sa nám zobrazí okno (obr. 5), v ktorom vyberieme predmet, označenie aktivity, krúžky a koľkokrát po koľko hodín chceme daný predmet do týždňa.



Obr. 5: Zadávanie aktivít v programe FET

Vzťahy a podmienky:

Na to, aby nám program vygeneroval základný rozvrh bez obmedzení, je potrebné zadať základné časové a priestorové obmedzenia, ktoré hovoria:

1. učitelia nemôžu učiť viac ako jednu hodinu naraz, študenti môžu mať len jeden predmet naraz
2. miestnosti môžu byť využité len jednou aktivitou v jednom čase a zároveň musia zohľadňovať kapacitu tried.

Rozšírené časové obmedzenia môžeme rozdeliť na tie pre profesorov, študentov a aktivity.

a) Keďže sme v rozvrhu nezadávali profesorov, ani obmedzenia v tejto časti sme nevyužívali.

Pre kompletnosť sem patria:

- časy, kedy profesor nie je dostupný
- maximálny počet dní v týždni, ktoré môže profesor odučiť

- max počet dier v rozvrhu za týždeň pre konkrétneho učiteľa/pre všetkých učiteľov
- max počet dier v rozvrhu za deň pre konkrétneho učiteľa/pre všetkých učiteľov
- max počet hodín pre profesora/pre všetkých profesorov na deň
- min počet hodín pre profesora/pre všetkých profesorov na deň
- max počet hodín za sebou pre profesora/pre všetkých profesorov

b) obmedzenia študentov:

- kedy krúžok nie je dostupný
- max počet dier v rozvrhu za týždeň pre krúžok/pre všetkých študentov
- krúžok/všetci študenti začína/začínajú skoro (najneskôr druhú hodinu)
- max počet hodín pre krúžok/pre všetkých študentov na deň
- min počet hodín pre krúžok/pre všetkých študentov na deň
- max počet hodín za sebou pre krúžok/pre všetkých študentov

c) obmedzenia aktivít:

- aktivita má preferovaný čas začiatku
- aktivita/skupina aktivít/skupina subaktivít má preferované vyučovacie hodiny
- aktivita/skupina aktivít/skupina subaktivít má preferovaný čas začiatku
- minimálny počet dní medzi skupinovou aktivít
- aktivita/skupina aktivít končí deň študentov
- skupina aktivít má rovnaký čas začiatku/deň(hocijaká hodina)/hodina(hocijaký deň)
- dve aktivity majú určené poradie
- dve aktivity nasledujú tesne po sebe
- dve aktivity sú spojené
- skupina aktivít sa nemôže prekrývať
- min počet medzier/dní medzi skupinami aktivít

c) priestorové obmedzenia majú viac podskupín a to:

- obmedzenia miestností – kedy sú jednotlivé miestnosti dostupné
- obmedzenia týkajúce sa učiteľov

- profesor vlastnú triedu
- profesor má viac vlastných tried
- max počet budov, ktoré môže profesor zmeniť za deň
- max počet budov, ktoré môže profesor zmeniť za týždeň
- min počet medzier, ktoré potrebuje profesor pri zmene budovy
- obmedzenia týkajúce sa študentov
 - skupina študentov má vlastnú triedu
 - skupina študentov má viac vlastných tried
 - max počet budov, ktoré môže krúžok/všetky krúžky zmeniť za deň
 - max počet budov, ktoré môže krúžok/všetky krúžky zmeniť za týždeň
 - min počet medzier, ktoré potrebuje krúžok/krúžky pri zmene budovy
- obmedzenia týkajúce sa predmetov
 - predmet/skupina predmetov preferuje určitú miestnosť

My sme využili:

- základné časové a priestorové obmedzenia
- cez označenie aktivít sme prednáškam nastavili vyučovanie od pondelka do štvrtku od 9:00 do 17:00, cvičeniam od pondelka do štvrtku od 8:00 do 17:00, v piatok od 8:00 do 13:00
- nastavili sme čas výučby ostatným predmetom, ktoré majú obmedzenia
- nastavili sme dva predmety, ktoré musia nasledovať v špecifickom poradí: P – inf. tech. v potravinárstve a C – inf. tech. v potravinárstve
- max počet hodín za deň – 10
- max počet hodín za sebou – 5
- max počet dier za týždeň – 3
- definovali sme jednotlivým predmetom miestnosti, v ktorých sa môžu vyučovať

Pri spracúvaní materiálov vidieť na akom princípe program pracuje: najskôr si zadelí najobmedzenejšie predmety a postupne pridáva ďalšie. Napr. v našom prípade sme spracovávali okolo 100 aktivít. Pri všetkých zadaných obmedzeniach prvých 80-90 spracoval

bez problémov a potom hľadal optimálne riešenia a vracal sa vo „výpočtoch“ naspäť. Výsledky rozvrhov nájdeme v prílohe FET.

3.4 Wise Timetable

WiseTimetable je ako jediný program určený pre vysoké školy a univerzity. [16] Tomu zodpovedá aj jeho vzhľad a funkčnosť. Keďže je to komerčný program, my sme mali stiahnutú iba demo verziu, ktorá je funkčná 2 dni alebo 10 spustení.

V nastaveniach si nastavíme základné informácie:

- od kedy do kedy je vyučovanie (čas aj týždne v semestri)
- zariadenie, resp. definovanie miestností (počítače, prednášky, cvičenia)
- prednastavená kapacita miestností (20)
- typy kurzov (prednáška, cvičenia, počítače, laboratórne cvičenia)

Profesori:

Profesorom zadávame meno, priezvisko, mail, miestnosti, v ktorých prednostne učí.

Miestnosti:

Zadáваме názov miestnosti, zariadenie, kapacitu, ak je iná ako prednastavená a zariadenie/využitie miestnosti.

Predmety:

V porovnaní s inými programami sa začína zadávaním typu štúdia a dĺžkou (v našich podmienkach bakalárske/inžinierske). My sme si zadali iba bakalárske o dĺžke jeden rok. Ďalej sa jednotlivým ročníkom zadávajú odbory, k odborom krúžky a predmety. Netreba preto zadávať všetky informácie zbytočne viackrát.

Aktivity:

Po vybraní predmetu sa nám zobrazí okno (obr. 6) a môžeme zadávať:

- odbory, ktoré majú daný predmet
- vyberieme daný typ kurzu, napr. cvičenie a zadáme jednu skupinu, ktorá má dvojhodinové cvičenia, pričom vytvoríme podskupiny pre jednotlivé krúžky, zadáme druhú podskupinu, ktorá má iba jednu hodinu cvičení a tiež určíme podskupiny pre jednotlivé krúžky
- pri jednotlivých typoch kurzov (cvičenie, prednáška, ...), definujeme konkrétne požiadavky na miestnosti, poprípade určíme konkrétnu miestnosť, ktorú predmet vyžaduje
- ak chceme, aby predmet bol 2-krát do týždňa po 2 hodiny, zapíšeme to: 2+2
- tiež tu môžeme nastaviť, aby cvičenie bolo až po prednáške
- ďalšiu možnosť, ktorú tento program ponúka, je, že vie nakombinovať blokovanú výučbu s periodickou (klasickou). Ku každému predmetu môžeme teda zadať, o aký predmet sa jedná a v ktorom týždni sa vyučuje.



Obr. 6: Zadávanie aktivít v programe WiseTimetable

Demo verzia nepodporuje generovanie rozvrhu. Napriek tomu môžeme povedať, že program má svoje kvality v tom, že je primárne určený pre vysoké školy. Má presnú štruktúru zadávania informácií, ktorá zodpovedá práve vysokým školám a nevyžaduje také množstvo zadávania údajov ako iné programy.

3.5 Výsledky

Výsledky môžeme porovnať z programov ASC a FET. ASC program je veľmi dobre prepracovaný na úrovni základných a stredných škôl, pričom jeho možnosti a funkcie úplne vyhovujú našim požiadavkám. V porovnaní s programom FET je na oveľa vyššej grafickej úrovni, za čo si však zákazník musí priplatiť. FET je naopak jednoduchý program, voľne dostupný na internete. Obsahuje základné parametre, ktoré musí každá škola prispôbiť svojim podmienkam. Nemá žiadne špecifiká, ktoré sa dajú aplikovať len na určitý okruh škôl, všetko je všeobecné. Dalo by sa povedať, že ASC má už prednastavené podmienky, ktoré môže užívateľ potrebovať, pričom vo FET treba nájsť spôsob ako ich vytvoriť a zadať. Ďalšou výhodou ASC je, že na rozdiel od FET je dostupný slovenčine, čo dokáže výrazne zjednodušiť prácu. Azda najväčším kladom ASC oproti FET je, že sa dajú nastaviť obedy bez toho, aby to výrazne zasahovalo do rozvrhu. Rozvrh vygeneruje tak, aby každý krúžok mal každý deň hodinu voľna v čase od 11:00 do 14:00. Vo FET sa tento problém dá riešiť len zadaním predmetu „obed“, čo naruší časové podmienky a obmedzenia.

Výhodou programu FET je jeho jednoduché ovládanie. V ASC musíme veľa nastavení hľadať, lebo sú vnorené v iných nastaveniach.

Jediná podmienka, ktorú sa nám nepodarilo splniť ani v jednom programe sú dve cvičenia naraz z fyziky. Dalo by sa to spôsobom, že si vyberieme presne dva krúžky, ktoré budú mať fyziku naraz a natvrdo ich nadefinovať. Takisto sme si nadefinovali aj krúžky, ktoré majú spolu prednášky z fyziky alebo matematiky.

Samotné generovanie rozvrhu v ASC začína testovaním, ktoré nám ukáže, či máme správne zadané podmienky, teda či je schopný vygenerovať rozvrh. Samotné generovanie trvá do 5 minút, závisí to samozrejme od náročnosti. V prípade, že sa nám rozvrh nepáči,

môže dať generovať nový. Vo FET sa nedá rozvrh testovať, takže po zadaní podmienok generujeme rozvrh. Môže sa však stať, že rozvrh sa nevygeneruje, lebo naše obmedzenia sú nespĺniteľné. Program však na chybu neupozorní. Po zastavení generovania vidíme, na ktorých aktivitách zastavil, s čím mal problém. Nie vždy musí byť chyba práve tam, preto musíme skontrolovať všetky obmedzenia a skúsiť spustiť generovanie opäť. Samotné generovanie trvá do 10 minút.

Okrem týchto troch programov, ktoré sme opísali a porovnali, vieme o ďalších, ktoré by mohli byť vhodné, avšak nemajú dostupné demo verzie ani ukážky programov. Jedná sa o programy: Celcat, LilaBerg, Novaschem, The timetabler, ClassPlanner.

4 ZÁVER

Cieľom tejto bakalárskej práce bolo poskytnúť prehľad programov zaoberajúcich sa generovaním rozvrhov z dôvodu zjednodušenia práce pri tvorbe rozvrhov na našej fakulte. K tejto téme patrí aj hľadanie vhodných algoritmov, na ktorých sú založené jednotlivé programy. V našej práci sme sa venovali genetickým algoritmom a algoritmu backtracking.

Samotná práca pozostávala z viacerých krokov. Prvým krokom k dosiahnutiu požadovaných výsledkov bolo zhromažďovanie potrebných dát: zoznam predmetov, zoznam krúžkov, zoznam miestností a ich kapacita a zoznam požiadaviek katedier a oddelení. Ďalej sme si našli programy, ktoré sa venujú danej problematike. Z týchto programov sme postupne selektovali tie, ktoré sme mohli reálne využiť. Takýchto programov nám ostalo trinásť. Nasledovne sme ich inštalovali a snažili sa pomocou nich nájsť riešenie problému. Nami zadané kritéria splnili dva programy – ASC a FET. Program WiseTimetable sme odskúšali len na úrovni zadávania dát. Tento ako jediný program bol primárne určený pre vysoké školy a univerzity, čím bol pre nás veľmi zaujímavý. Bohužiaľ jeho demo verzia neumožňovala generovanie rozvrhov.

Dúfame, že táto práca bude užitočná a v blízkej budúcnosti nahradí ručné generovanie rozvrhov niektorým z programov spomenutých vyššie.

POUŽITÁ LITERATÚRA

[1] Constraint programming, Backtracking; [on line]; Dostupné 15.5.2009 z:

<<http://www2.fiit.stuba.sk/~kapustik/ZS/Clanky0809/kohut/index.html#Sekcia13>>

[2] Heuristické algoritmy; [on line]; Dostupné 15.5.2009 z:

<http://cs.wikipedia.org/wiki/Heuristick%C3%A9_algoritmy>

[3] Výpočtová zložitosť; [on line]; Dostupné 15.5.2009 z:

<http://sk.wikipedia.org/wiki/V%C3%BDpo%C4%8Dtov%C3%A1_zlo%C5%BEitos%C5%A5>

[4] Výpočtová zložitosť; [on line]; Dostupné 15.5.2009 z:

<<http://frcatel.fri.uniza.sk/users/paluch/grkap2-nopics.pdf>>

[5] P, NP problémy; [on line]; Dostupné 15.5.2009 z:

<<http://mathworld.wolfram.com>>

[6] P, NP problémy; [on line]; Dostupné 15.5.2009 z:

<http://www.cc.gatech.edu/~howardz/micellaneous/gre_cs_sub/np_complete.htm>

[7] Backtracking; [on line]; Dostupné 15.5.2009 z:

<<http://www.sprite.edi.fmph.uniba.sk/~szorad/Zaklady/Backtracking.html>>

[8] Úvod do genetických algoritmov; [on line]; Dostupné 15.5.2009 z:

<<http://cgg.mff.cuni.cz/~pepca/prg022/luner.html>>

[9] Genetické algoritmy; [on line]; Dostupné 15.5.2009 z:

<<http://programujte.com/index.php?akce=clanek&cl=2005072601-geneticke-algoritmy-a-jejich-aplikace-v-praxi>>

[10] Genetické algoritmy; [on line]; Dostupné 15.5.2009 z:

<<http://cgg.mff.cuni.cz/~pepca/prg022/luner.html>>

[11] Genetické algoritmy; [on line]; Dostupné 15.5.2009 z:

<http://sk.wikipedia.org/wiki/Genetick%C3%BD_algoritmus>

[12] Aplikácia genetických algoritmov; [on line]; Dostupné 15.5.2009 z:

<http://www.codeguru.com/cpp/cpp/algorithms/general/article.php/c14825__1/>

[13] aSc timetables; [on line]; Dostupné 15.5.2009 z:

<<http://www.asctimetables.com>>

[14] aSc timetables; [on line]; Dostupné 15.5.2009 z:

<<http://lesk.cas.sk/cl/100274/245222/Vymysleli-to-chalani>>

[15] FET; [on line]; Dostupné 15.5.2009 z:

<<http://lalescu.ro/liviu/fet>>

[16] WiseTimetable; [on line]; Dostupné 15.5.2009 z:

<<http://www.wisetimetable.com>>

PRÍLOHA

Zoznam krúžkov

Krúžok	Odbor
11	Technológia polymérnych materiálov
21	Chemická technológia
41, 42	Chemické inžinierstvo
61, 62, 63	Potravinárstvo
71, 72, 73	Medicínska chémia
81, 82, 83	Biotechnológia
91	Automatizácia, informatizácia a manažment v chémii a potravinárstve

Zoznam miestností

trieda	kapacita	účel
ch16	176	prednáška
ch15	96	prednáška
ch17	96	prednáška
ch249	20	cvičenie
ch250	20	cvičenie
ch251	20	cvičenie
ch256	20	cvičenie
ch243	20	biochem. a biotech. informácie
ch252	20	počítače
ch211	40	chem.-inž. výpočty na PC
lfyzi	40	laboratórium fyzika
lanor	40	laboratórium anorganická chémia
lmikr	20	laboratórium biológia
linzi	20	laboratórium chemické inžinierstvo

Zoznam predmetov

predmety	krúžky
P – matematika	10, 20, 40, 60, 70, 80, 90
C – matematika	10, 20, 40, 60, 70, 80, 90
P – fyzika	10, 20, 40, 60, 70, 80, 90
C – fyzika	10, 20, 40, 60, 70, 80, 90
LC – fyzika	10, 20, 40, 60, 70, 80, 90
LC – anorganická chémia	10, 20, 40, 60, 70, 80, 90
P - základy všeobecnej ekonomickej teórie	10, 20, 40, 60, 70, 80, 90
C - základy všeobecnej ekonomickej teórie	60, 80
C – metódy počítačového spracovania dát	90
P – optimalizácia	90
C – optimalizácia	90
P – programovanie	90
C – programovanie	90
P – biochemické a biotechnologické informácie	80
C – biochemické a biotechnologické informácie	80
P – personálny manažment	60, 80
P – biológia	60, 80
LC – biológia	60, 80
P – anorganická technológia a materiály	10, 40
C – energetické bilancie	10, 40
C – chemicko-inžinierske výpočty na PC	40
P – tok tekutín	40
C – tok tekutín	40
LC – tok tekutín	40
P – základy práva	20, 40, 70
P – anorganická chémia 2	70
C – anorganická chémia 2	70

P – anorganická technológia	20, 70
C – anorganická technológia	20
P – informačné technológie v potravinárstve	60
C – informačné technológie v potravinárstve	60
P – lignocelulóзовé materiály	10
C – lignocelulóзовé materiály	10
P – makromolekulová chémia	10
C – seminár programu	10
P – chemická informatika	70
C – chemická informatika	70
P – anorganické materiály	20
C – anorganické materiály	20

Výsledky FET

Rozvrh krůžkov

	11										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	C - matematika C ch251		P - lignocel. mat P ch15		---	LANOR lanor				
Tuesday	---	---	---	P - mat P ch16		C - lignocel. mat C ch249	P - ZVET P ch16		---	---	---
Wednesday	---	P - fyz P ch16		C - energet bilancie C ch249		---	C - sem programu C ch256	P - makromol chem P ch17		---	---
Thursday	LC fyz Lfyz			---	P - anorg techn a mat P ch17		---	---	---	---	---
Friday	C - fyz C ch250	---	---	---	---	---	---	---	---	---	---

	21											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	---	---	P - fyz P ch16		C - fyz C ch250	---	LANOR lanor					
Tuesday	---	P - anorg technologia P ch16		P - mat P ch16		---	P - ZVET P ch16		---	---	---	
Wednesday	---	---	---	P - zaklady prava P ch16		C - anorg mat C ch249		---	---	---	---	
Thursday	LC fyz Lfyz			---	C - matematika C ch249		P - anorg mat P ch16		---	---	---	
Friday	---	---	C - anorg. tech C ch256			---	---	---	---	---	---	

	41											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	---	---	P - fyz P ch16		LC fyz Lfyz			---	---	---	---	
Tuesday	LC tok tekutin linz	C - chem-inz. vypoety na pc C ch211		C - fyz C ch251	P - mat P ch16		---	P - ZVET P ch16		LANOR lanor		
Wednesday	---	C - energet bilancie C ch249			P - zaklady prava P ch16		---	---	---	---	---	
Thursday	---	P - tok tekutin P ch16			---	P - anorg techn a mat P ch17		---	C - tok tekutin C ch256		---	
Friday	---	---	---	C - matematika C ch250			---	---	---	---	---	

	42										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	LC tok tekutin linz	---	P - fyz P ch16		LC fyz Lfyz		---	---	---	---	---
Tuesday	C - matematika C ch249		---	P - mat P ch16		---	P - ZVET P ch16	LANOR lanor			
Wednesday	C - fyz C ch251	C - tok tekutin C ch256	P - zaklady prava P ch16			---	---	---	---	---	---
Thursday	---	P - tok tekutin P ch16	C - chem-inz. vypocty na pc C ch211	P - anorg techn a mat P ch17		---	---	---	---	---	---
Friday	C - energet bilancie C ch256		---	---	---	---	---	---	---	---	---

	61										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	---	P - fyz P ch16		---	P - pers. manazment P ch16	P - biologia P ch16		---	---	---
Tuesday	---	---	---	LC fyz Lfyz			---	---	---	---	---
Wednesday	---	LC bio Lmikr	P - inf techn v potr P ch15	C - inf techn. v potr C ch15	---	P - ZVET P ch16		---	---	---	---
Thursday	C - matematika C ch251	C - fyz C ch250	P - mat P ch16		---	LANOR lanor					
Friday	---	C - ZVET C ch250	---	---	---	---	---	---	---	---	---

	62										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	LC bio Lmikr	P - fyz P ch16		---	P - pers. manazment P ch16		P - biologia P ch16		---	---	---
Tuesday	C - ZVET C ch250	---	LC fyz Lfyz			C - matematika C ch251		---	---	---	---
Wednesday	---	---	P - inf techn v potr P ch15	C - inf techn. v potr C ch15	C - fyz C ch250	P - ZVET P ch16		---	---	---	---
Thursday	---	---	---	P - mat P ch16		---	LANOR lanor				
Friday	---	---	---	---	---	---	---	---	---	---	---

	63											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	---	---	P - fyz P ch16		---	P - pers. manazment P ch16		P - biologia P ch16		---	---	
Tuesday	LANOR lanor					---	LC bio Lmikr	C - fyz C ch249		---	---	
Wednesday	---	---	---	P - inf techn v potr P ch15	C - inf techn. v potr C ch15		---	P - ZVET P ch16		---	---	
Thursday	---	C - ZVET C ch249		P - mat P ch16			---	---	---	---	---	
Friday	C - matematika C ch249		LC fyz Lfyz			---	---	---	---	---	---	

	71											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	C - matematika C ch250		C - fyz C ch256		---	C - chem infor C ch256		LC fyz Lfyz		---	---	
Tuesday	---	P - anorg technologia P ch16		P - mat P ch16		---	P - ZVET P ch16		---	---	---	
Wednesday	---	P - fyz P ch16		P - zaklady prava P ch16			---	LANOR lanor				
Thursday	---	---	P - anorg chemia P ch17		P - chem infor. P ch15		C - anorg. chemia C ch250		---	---	---	
Friday	---	---	---	---	---	---	---	---	---	---	---	

	72											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	C - chem infor C ch256		C - matematika C ch250		---	C - fyz C ch250		LC fyz Lfyz		---	---	
Tuesday	---	P - anorg technologia P ch16		P - mat P ch16		---	P - ZVET P ch16		---	---	---	
Wednesday	---	P - fyz P ch16		P - zaklady prava P ch16			---	LANOR lanor				
Thursday	C - anorg. chemia C ch250		P - anorg chemia P ch17		P - chem infor. P ch15		---	---	---	---	---	
Friday	---	---	---	---	---	---	---	---	---	---	---	

	73										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	LANOR lanor					---	C - matematika C ch251	---	---	---	---
Tuesday	---	P - anorg technologia P ch16	P - mat P ch16			---	P - ZVET P ch16	---	---	---	
Wednesday	---	P - fyz P ch16		P - zaklady prava P ch16		---	LC fyz Lfyz	---	---	---	
Thursday	---	---	P - anorg chemia P ch17	P - chem infor. P ch15	C - fyz C ch256		---	---	---	---	
Friday	---	C - chem infor C ch251		C - anorg. chemia C ch251		---	---	---	---	---	

	81										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	LC fyz Lfyz			---	P - pers. manazment P ch16			P - biologia P ch16	---	---
Tuesday	LANOR lanor					---	---	---	---	---	
Wednesday	---	P - fyz P ch16	C - biochem a biotech info C ch243		---	P - biochem a biotech infot P ch16	P - ZVET P ch16	C - fyz C ch251	---	---	
Thursday	---	LC bio Lmikr	P - mat P ch16			---	C - ZVET C ch249		---	---	
Friday	---	---	---	C - matematika C ch249		---	---	---	---	---	

	82										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	LC fyz Lfyz			---	P - pers. manazment P ch16			P - biologia P ch16	---	---
Tuesday	---	LC bio Lmikr	C - matematika C ch249		---	---	---	---	---	---	
Wednesday	---	P - fyz P ch16	---	C - fyz C ch250	P - biochem a biotech infot P ch16		P - ZVET P ch16	---	---	---	
Thursday	---	---	---	P - mat P ch16	C - biochem a biotech info C ch243		C - ZVET C ch251	---	---	---	
Friday	LANOR lanor					---	---	---	---	---	

	83											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	---	---	---	---	---	P - pers. management P ch16		P - biologia P ch16	---	---	---	
Tuesday	LC fyz Lfyz		C - matematika C ch251		---	---	---	---	---	---	---	
Wednesday	---	P - fyz P ch16	C - fyz C ch256	---	P - biochem a biotech infot P ch16		P - ZVET P ch16		---	---	---	
Thursday	---	C - ZVET C ch256	P - mat P ch16		---	C - biochem a biotech info C ch243		LC bio Lmikr	---	---	---	
Friday	LANOR lanor				---	---	---	---	---	---	---	

	91											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	C - matematika C ch249		P - fyz P ch16		---	C - MPSD pc ch252		---	P - optim ch17			
Tuesday	---	---	---	---	---	---	---	P - ZVET P ch17	C - fyz C ch256	P - programovanie ch15	---	
Wednesday	---	---	---	---	---	---	---	P - ZVET P ch15		C - programovanie pc ch252		
Thursday	LANOR lanor			P - mat P ch16		---	C - optim pc ch252		---	---	---	
Friday	---	---	LC fyz Lfyz		---	---	---	---	---	---	---	

Rozvrh miestností

	ch15											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	---	---	---	11 P - lignocel. mat P		---	---	---	---	---	---	
Tuesday	---	---	---	---	---	---	---	---	---	91 P - programovanie	---	
Wednesday	---	---	---	61, 62, 63 P - inf techn v potr P	61, 62, 63 C - inf techn. v potr C	---	---	91 P - ZVET P	---	---	---	
Thursday	---	---	---	---	71, 72, 73 P - chem infor. P	---	---	---	---	---	---	
Friday	---	---	---	---	---	---	---	---	---	---	---	

	ch16										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	---	91, 62, 61, 63, 21, 41, 42 P - fyz P	---	---	61, 62, 63, 81, 82, 83 P - pers. manazment P	---	61, 62, 63, 81, 82, 83 P - biologia P	---	---	---
Tuesday	---	21, 71, 72, 73 P - anorg technologia P	---	11, 21, 71, 72, 73, 41, 42 P - mat P	---	---	11, 21, 41, 42, 71, 72, 73 P - ZVET P	---	---	---	
Wednesday	---	71, 72, 73, 81, 82, 83, 11 P - fyz P	---	21, 71, 72, 73, 41, 42 P - zaklady prava P	---	81, 82, 83 P - biochem a biotech infot P	---	61, 62, 63, 81, 82, 83 P - ZVET P	---	---	
Thursday	---	41, 42 P - tok tekutin P	---	61, 62, 63, 81, 83, 82, 91 P - mat P	---	---	---	21 P - anorg mat P	---	---	
Friday	---	---	---	---	---	---	---	---	---	---	

	ch17										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	---	---	---	---	---	---	---	91 P - optim	---	---
Tuesday	---	---	---	---	---	---	---	91 P - ZVET P	---	---	---
Wednesday	---	---	---	---	---	---	---	11 P - makromol chem P	---	---	---
Thursday	---	---	71, 72, 73 P - anorg chemia P	---	11, 41, 42 P - anorg techn a mat P	---	---	---	---	---	---
Friday	---	---	---	---	---	---	---	---	---	---	---

	ch249										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	91 C - matematika C	---	---	---	---	---	---	---	---	---	---
Tuesday	42 C - matematika C	---	---	82 C - matematika C	---	11 C - lignocel. mat C	---	---	63 C - fyz C	---	---
Wednesday	---	41 C - energet bilancie C	---	11 C - energet bilancie C	---	21 C - anorg mat C	---	---	---	---	---
Thursday	---	63 C - ZVET C	---	---	---	21 C - matematika C	---	81 C - ZVET C	---	---	---
Friday	63 C - matematika C	---	---	81 C - matematika C	---	---	---	---	---	---	---

	ch250										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	71 C - matematika C		72 C - matematika C		21 C - fyz C	72 C - fyz C		---	---	---	---
Tuesday	62 C - ZVET C		---	---	---	---	---	---	---	---	---
Wednesday	---	---	---	---	82 C - fyz C	62 C - fyz C	---	---	---	---	---
Thursday	72 C - anorg. chemia C		61 C - fyz C	---	---	71 C - anorg. chemia C		---	---	---	---
Friday	11 C - fyz C	61 C - ZVET C		41 C - matematika C		---	---	---	---	---	---

	ch251										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	11 C - matematika C		---	---	---	73 C - matematika C		---	---	---
Tuesday	---	---	41 C - fyz C	83 C - matematika C		---	62 C - matematika C		---	---	---
Wednesday	42 C - fyz C	---	---	---	---	---	---	---	81 C - fyz C	---	---
Thursday	61 C - matematika C		---	---	---	---	82 C - ZVET C		---	---	---
Friday	---	73 C - chem infor C		73 C - anorg. chemia C		---	---	---	---	---	---

	ch256										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	72 C - chem infor C		71 C - fyz C		---	71 C - chem infor C		---	---	---	---
Tuesday	---	---	---	---	---	---	---	---	91 C - fyz C	---	---
Wednesday	---	42 C - tok tekutin C		83 C - fyz C	---	---	11 C - sem programu C		---	---	---
Thursday	---	83 C - ZVET C		---	---	73 C - fyz C		41 C - tok tekutin C		---	---
Friday	42 C - energet bilancie C		21 C - anorg. tech C		---	---	---	---	---	---	---

	ch243										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	---	---	---	---	---	---	---	---	---	---
Tuesday	---	---	---	---	---	---	---	---	---	---	---
Wednesday	---	---	---	81 C - biochem a biotech info C	---	---	---	---	---	---	---
Thursday	---	---	---	---	---	82 C - biochem a biotech info C	83 C - biochem a biotech info C	---	---	---	---
Friday	---	---	---	---	---	---	---	---	---	---	---

	ch252											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	---	---	---	---	---	91 C - MP&SD pc	---	---	---	---	---	
Tuesday	---	---	---	---	---	---	---	---	---	---	---	
Wednesday	---	---	---	---	---	---	---	---	---	91 C - programovanie pc	---	
Thursday	---	---	---	---	---	---	91 C - optim pc	---	---	---	---	
Friday	---	---	---	---	---	---	---	---	---	---	---	

	lanor											
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
Monday	73 LANOR					---	11, 21 LANOR					
Tuesday	81, 63 LANOR					---	---	---	41, 42 LANOR			
Wednesday	---	---	---	---	---	---	71, 72 LANOR					
Thursday	91 LANOR				---	---	---	62, 61 LANOR				
Friday	83, 82 LANOR					---	---	---	---	---	---	

	Lfyz										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	81, 82 LC fyz		41, 42 LC fyz			71, 72 LC fyz		---	---	
Tuesday	83 LC fyz			61, 62 LC fyz			---	---	---	---	---
Wednesday	---	---	---	---	---	---	73 LC fyz		---	---	---
Thursday	11, 21 LC fyz			---	---	---	---	---	---	---	---
Friday	---	---	91, 63 LC fyz			---	---	---	---	---	---

	Lmikr										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	62 LC bio		---	---	---	---	---	---	---	---	---
Tuesday	---	82 LC bio		---	---	---	63 LC bio		---	---	---
Wednesday	---	61 LC bio		---	---	---	---	---	---	---	---
Thursday	---	81 LC bio		---	---	---	---	83 LC bio		---	---
Friday	---	---	---	---	---	---	---	---	---	---	---

	linz										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	42 LC tok tekutin		---	---	---	---	---	---	---	---	---
Tuesday	41 LC tok tekutin		---	---	---	---	---	---	---	---	---
Wednesday	---	---	---	---	---	---	---	---	---	---	---
Thursday	---	---	---	---	---	---	---	---	---	---	---
Friday	---	---	---	---	---	---	---	---	---	---	---

	ch211										
	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
Monday	---	---	---	---	---	---	---	---	---	---	---
Tuesday	---	41 C - chem-inz. vypocty na pc C	---	---	---	---	---	---	---	---	---
Wednesday	---	---	---	---	---	---	---	---	---	---	---
Thursday	---	---	---	42 C - chem-inz. vypocty na pc C	---	---	---	---	---	---	---
Friday	---	---	---	---	---	---	---	---	---	---	---

Zoznam programov

Použité programy

aSc Timetables	-	www.asctimetables.com
FET	-	www.lalescu.ro/liviu/fet
Wise Timetable	-	www.wisetimetable.com
gp-Untis	-	www.grupet.at
Orologio	-	www.antinoos.gr/en
Seasia	-	seasiaconsulting.com
online program	-	www.school-timetable.eu
TTMaker	-	www.rootsoftware.com
Visual Classroom	-	www.vvs.com.au
The TimetablerTimetabler	-	www.timetabler.com
iMagic	-	www.imagictimetablessoftware.com
Mimosa	-	www.mimosa.com
Open Course Timetabler	-	www.opencct.org

Ostatné programy

Abacus School Pro	-	www.abacusschoolpro.com
ABC	-	www.abc-scheduling.com
Ad Astra	-	www.aais.com
Allocate+	-	allocateplus.com.au
Assignment	-	www.schoolassignment.com
Automated	-	www.autoscheduling.com
BenchMark Timetable	-	www.benchmarktimetable.com
Celcat	-	www.celcat.com
ClassPlanner	-	www.classplanner.biz
ComQuip, Inc.	-	www.comquip.com
CyberMatrix	-	www.cyber-matrix.com
Edval	-	www.edval.com.au

EMS	-	www.dea.com
eventMap	-	www.evmap.com
ExamPlanner	-	www.examplanner.com
Expression Resource Booking	-	www.expressionsoftware.co.uk
GTI3	-	www.plansoft.co.uk
Human Edge	-	www.humanedge.biz
Infosilem	-	www.infosilem.com
IQ.Session	-	iqsession.com
LBE	-	www.room-booking-software.com
LilaBerg	-	www.lilaberg.com
Meeting Room Manager	-	www.netsimplisity.com
MMS	-	www.cri-mms.com
Modelino	-	www.viosystem.com
Novaschem	-	www.novaschem.com
o!Timetabling	-	www.itsabacus.co.za
ParentEve –	-	www.parenteve.com
Parent/Teacher Conference Scheduler	-	www.cbfssoftware.com
Penalara	-	www.penalara.som
Room Booking System	-	www.roombookingsystem.com
Satyanash	-	satyanash.sourceforge.net
Scheduling Optimizer	-	www.tabbysoft.com
SchoolDays	-	www.myschoolcal.com
Scientia	-	www.scientia.com
Silk	-	www.exlogica.com
Solstar	-	www.mjdata.com/solstar
Synergy	-	www.synergyinfosys.com
Teched	-	www.teched.com.au
Thoughtimus	-	www.thoughtimus.com
TimeDesign	-	www.onyxsa.co.za/timedesign
TTable	-	ttable.com

Untis	-	www.saiposa.ch
Visual Timetabling	-	visual.timetabling.free.fr
ypSched	-	formulus.com/ypshed
online program	-	www.supertime2000.info