

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

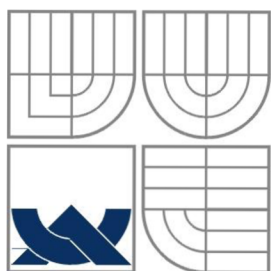
**AUTOMATICKÉ JAZZOVÉ ARANŽMÁ**

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

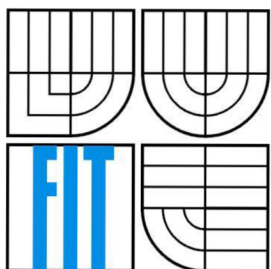
**AUTOR PRÁCE**  
AUTHOR

**Bc. PETR CHADIM**

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# AUTOMATICKÉ JAZZOVÉ ARANŽMÁ

AUTOMATIC JAZZ ARRANGEMENT

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. PETR CHADIM

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. MICHAL FAPŠO

BRNO 2011

# Automatické jazzové aranžmá

## Automatic jazz arrangement

**Vedoucí:**

Fapšo Michal, Ing., UPGM FIT VUT

**Oponent:**

Karafiát Martin, Ing., Ph.D., UPGM FIT VUT

**Student:**

Chadim Petr, Bc.

**Zadání:**

1. Seznamte se s principy a technikami používanými pro aranžování melodií pomocí CSP (Constrained Satisfaction Problem).
2. Navrhněte systém, který k zadané jednoduché melodii vytvoří jazzové aranžmá a výsledek vhodným způsobem zobrazí nebo přehraje.
3. Vytvořte funkční prototyp.
4. Implementujte systém.
5. Prezentujte funkčnost programu na několika melodiích.

**Část požadovaná pro obhajobu SP:**

Body 1, 2, 3.

**Kategorie:**

Umělá inteligence

**Literatura:**

- Luděk Zenkl, ABC hudební nauky, 2003
- Jaroslav Kofroň, Učebnice harmonie, 2002
- Dle pokynů vedoucího

## **Abstrakt**

Práce pojednává o aranžování melodie, která je doplněna jazzovými akordy. Zabývá se vytvořením dalších harmonických hlasů pomocí metody blokové harmonizace. Rozdělení na cílové a přechodové noty je provedeno pomocí techniky programování s omezujícími podmínkami (CSP). Přechodové noty jsou reharmonizovány dominantním septakordem nebo paralelním akordem. Pomocí CSP je také vytvořen basový part. Pro řešení CSP je použita knihovna Gecode. Harmonické hlasy jsou aranžovány čtyřhlasou úzkou harmonií. Výsledek aplikace slouží jako pomůcka pro hudebního aranžéra.

## **Klíčová slova**

Harmonizace melodie, CSP, programování s omezujícími podmínkami, bloková harmonizace, reharmonizace, cílová nota, přechodová nota, aranžování, čtyřhlasá úzká harmonie, Gecode, Lilypond.

## **Abstract**

This Thesis is focused on the arranging of the melody, which is accompanied by jazz chords. It deals with creating a more harmonious voices using Block Voicing method. Distribution to target notes and passing notes is made using techniques of constraint programming (CSP). Passing notes are reharmonized by dominant seventh chord or by parallel chord. Using CSP a bass part is also created. To solve CSP is used Gecode library. The harmonious voices are arranged by Four Part Close Voicing. The application result is a tool for the music arranger.

## **Keywords**

Melody harmonization, Constraint Satisfaction Problem, CSP, Constraint Programming, Block Voicing, Reharmonization, target note, passing note, arranging, Four part close voicing, Gecode, Lilypond.

## **Citace**

Chadim Petr: Automatické jazzové aranžmá. Brno, 2011, Diplomová práce, FIT VUT v Brně



# Automatické jazzové aranžmá

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Michala Fapša. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Petr Chadim  
24.5.2011

## Poděkování

Chtěl bych na tomto místě poděkovat za cenné rady, ochotu a dobré vedení především svému vedoucímu semestrálního projektu Ing. Michalu Fapšovi.

© Petr Chadim, 2011.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah .....	1
1 Úvod.....	2
2 Základy hudební teorie.....	4
2.1 Melodie a harmonie .....	4
2.2 Intervaly .....	5
2.3 Jazzové akordy, jejich tenze a akordové značky .....	5
2.4 Aranžování versus kompozice .....	7
3 Metodika jazzového aranžování.....	9
3.1 Blokovaná harmonizace .....	9
3.2 Aranžování basu .....	14
4 Programování s omezujícími podmínkami .....	16
4.1 Metody řešení CSP .....	16
4.2 Algoritmus prohledávání do šířky - BFS.....	17
4.3 Algoritmus prohledávání do hloubky - DFS.....	17
4.4 Optimalizace metod řešení CSP .....	19
4.5 Knihovna Gecode .....	20
5 Návrh systému .....	21
5.1 Reprezentace hudebních dat .....	21
5.2 Schéma aranžéru.....	27
5.3 Vytvoření basového partu.....	27
5.4 Rozdělování na cílové a přechodové noty a stanovení CSP .....	31
5.5 Harmonizace cílových a reharmonizace přechodových not .....	35
5.6 Uživatelské rozhraní aplikace.....	37
6 Závěr .....	41
6.1 Zhodnocení dosažených výsledků .....	41
6.2 Možnosti pokračování práce.....	42
Literatura .....	43
Seznam příloh .....	44
Příloha 1: Manuál k ovládní aplikace .....	45
Příloha 2: Syntaxe konfiguračního souboru.....	46
Příloha 3: Ukázky výsledných aranžmá.....	48

# 1 Úvod

V dnešním sofistikovaném světě jsou již informační technologie natolik rozvinuty, že nevyhnutelně zasahují do nejrůznějších oblastí lidských činností. V některých již zcela převzali vládu a lidé se již na ně plně spoléhají, nejsou-li na nich plně závislí. V mnoha dalších oblastech teprve informační technologie hledají to správné využití, aby postupnými krůčky časem převzali celou úlohu na sebe. Lidé, jakožto uživatelé vyspělých systémů, zbaveni nutnosti rozumět danému problému, musí umět systém pouze správně ovládat. To je fakt, na který se můžeme dívat ze dvou stran. Bez jakýchkoliv pochybností můžeme říci, že lékařské přístroje zachraňují mnoho lidských životů. Ovšem například v oblastech umění jsou komukoliv, díky chytrému počítačovému softwaru, zpřístupněny schopnosti, kterých by jinak třeba nikdy nedosáhl. Dá se tedy také říci, že informační technologie někdy smazávají rozdíly v lidských talentech a uměleckých schopnostech.

Oblastí, jež v posledních letech začíná na informačních technologiích stavět, je také hudba. Když pomínu využití softwaru zpracovávající zvuk v nahrávacích a zvučících systémech či nástroje používající syntetizovaný nebo jinak upravený zvuk, můžeme zaměřit pozornost na samotný proces tvorby hudební myšlenky.

## Motivace

Pokusím se popsat aspekty, které mě motivovali právě k tomuto tématu mé práce. Řada prací přede mnou, které se zabývaly harmonizací melodie, byla zaměřena na tvorbu harmonie k dané melodii na základě pravidel klasické hudby. Tyto práce určovali harmonii skladby podle tónů melodie. V takovém případě se může stát, že dva takové různé programy zharmonizují danou melodii odlišným způsobem a vzniknou tak v podstatě dvě různé skladby. Kdybychom tak vzali známou melodii, mohl by ji harmonizační program zharmonizovat jinak, než jak ji známe, ač by postupoval správně dle definovaných pravidel. V praxi však, když skladatel vymyslí nějakou melodii, má již zároveň představu o její harmonii.

Mám s kamarády saxofonové seskupení a pokud nechceme, aby náš repertoár byl omezen tím, jaké skladby v úpravě pro 4-5 hlasů se nám podaří sehnat, musíme být schopni skladby aranžovat sami. Pokud vezmeme nějakou známou melodii, je třeba k ní přidat další harmonické hlasy tak, aby byla melodie podpořena a zasazena do širšího kontextu. Musí být zachovány původní harmonické postupy, aby skladba zněla na poslech stejně či podobně jako originál, pouze v jiném „kabátu“. Harmonii skladby určuje sled akordů, z nichž každý přísluší fragmentu melodie. Pokud tedy známe melodii skladby a k ní správný sled akordů, jsme schopni upravit skladbu pro jakékoliv nástrojové obsazení.

Nyní vezmu v úvahu velký orchestr jako je například *jazzový big band*, který je složen z rytmické, trumpetové, pozounové a saxofonové sekce. Pokud pro takovéto složení chceme aranžovat skladbu, princip je stále stejný. Dle melodie a akordů tvoříme harmonii. Buď v každé sekci zvlášť, kdy se mohou jednotlivé sekce střídát ve vedení melodie, nebo mohou všechny hrát v široké harmonii najednou. Případně můžeme vést v každé sekci jinou melodickou linku, které se pak vzájemně doplní.

Je samozřejmě mnoho taktik, jak harmonizovat jednotlivé nástrojové sekce. Navíc se jedná do značné míry o intuitivní záležitost, při níž hrají roli zkušenosti a styl aranžéra a v neposlední řadě i hudební žánr cílové skladby. Můžeme se ovšem zaměřit na jednu z používaných aranžovacích metod a pak poměrně snadno definovat množinu pravidel, jimiž se harmonizace řídí.

Ve své práci se chci zabývat výhradně harmonií jazzovou, která je oproti harmonii klasické pro lidské ucho podstatně složitější. Často není výjimkou lehká disonance a díky tomu najdeme obecných pravidel pro tvorbu jazzové harmonie podstatně méně než v klasické hudbě. Originalita a pestrost skladeb pramení právě v jejich porušování. Je pouze na autorovi či aranžérovi skladby, jak složitou harmonií svou melodii ozdobí.

## Cíle práce

Zadání práce hovoří o automatickém jazzovém aranžmá a já jsem se v předchozím odstavci zmínil o big bandu a o jeho nástrojových sekcích. Jako prvotní cíl své práce jsem si zvolil vytvořit program, který bude aranžovat melodie pro saxofonovou sekci. Na vstupu bude vhodným způsobem zadaná melodie s akordy, definujícími její harmonickou stavbu. Úkolem je na základě pravidel aranžování jazzové hudby ke každé notě melodie přidat další harmonické hlasy.

Takovýto pětihlas ale postrádá basový part, který určuje harmonickou funkci a tím činí dojem z hudby více konkrétním a pochopitelným. V big bandu obstarává bas rytmická sekce. Výsledné pětihlasé aranžmá však může být použito i pro jakékoliv jiné nástrojové seskupení. Doplním ho tedy ještě o basový part.

Součástí zadání je využít techniku CSP aneb programování s omezujícími podmínkami. Aranžování vnitřních harmonických hlasů i vedení basového partu podléhá svým pravidlům, které je možno vyčíst z literatury pro aranžování jazzové hudby. Tyto problémy mohou mít více řešení a lze je popsat konečnou množinou pravidel či omezujících podmínek. Využití CSP se tedy přímo nabízí.

## 2 Základy hudební teorie

V rámci této kapitoly bych chtěl objasnit alespoň některé z pojmů, které ve své práci používám. Není zde ovšem možné zcela zasvětit hudebního laika, ani to není cíl práce. Pokusím se ale vystihnout podstatu, která nás zde bude zajímat. Stručně nastíním význam harmonie a melodie, poté věnuji kapitoly jazzovým akordům a intervalům, z nichž se skládají, a na závěr bych rád nastínil rozdíl mezi aranžováním a kompozicí, z čehož vyplývá účel mojí práce. Informace pro tuto kapitolu byly čerpány z [8], [9], [10] a [11].


### 2.1 Melodie a harmonie

**Melodie** je to, co nám utkví v hlavě při poslechu skladby. Je to ústřední motiv, který je nejvíce patrný. Pokud si sami zpíváme nějakou píseň, zpíváme její melodii.

Můžeme ji definovat jako skladatelem rytmicky organizovanou sekvenci jednotlivých tónů, navazujících na sebe tak, aby byla vyjádřena hudební myšlenka nebo její frakce.

Samotná melodie ovšem působí příliš chudě, pokud zapomeneme na možnost, že je to autorův záměr. Přidáme-li k melodii doprovodné hlasy, dáme jí tím teprve pravý význam. Další přidané hlasy k melodii tvoří **harmonii** skladby.

Definovat harmonii můžeme jako nauku, zabývající se studiem současně znějících tónů či akordů. Bývá označována jako vertikální aspekt hudby, zatímco melodie je aspektem horizontálním. Na obr. 1 je notový zápis části známé písně „Kočka leze dírou“. Melodie je zapsána na 1. řádku notové osnovy společně s akordy určujícími harmonii a psanými ve značkách. Na druhém řádku notové osnovy je vyjádřena harmonie jako doprovod písně. Obsahuje vždy tóny z příslušného akordu psané nad sebou.



The image shows a musical score for the song "Kočka leze dírou". It consists of two staves. The top staff is the melody, written in treble clef with a common time signature (C). The notes are: C4, D4, E4, F4, G4, A4, B4, C5. The bottom staff is the harmony, written in treble clef with a common time signature (C). It shows chords: C, F, C, G, C, G7, C. The lyrics are: Ko-čka le - ze dí - rou pes ok - nem ne - bu - de - li pr - šet ne - zmo - knem.

Obrázek 1: Příklad melodie s harmonií.

Pro tvorbu harmonie existuje spousta pravidel, ovšem variabilita hudby spočívá právě v jejich porušování. Budeme-li se jimi striktně držet, máme jistotu, že skladba bude harmonizována „korektně“ a nebude obsahovat rušivé elementy. Ovšem jen autor ví, jaké pocity chtěl skladbou

vyjádřit, a je čistě na něm jak s obecně přijímanými pravidly naloží. Ve své práci proto ponechám stanovení harmonie na uživateli ve formě sledu akordů. Nutno však podotknout, že se tím nezřikám procesu harmonizace. Chci jen respektovat autorovu představu o tom, jak má být melodie chápána. Stále je několik cest, jak vést další harmonické hlasy, což silně závisí na hudebním žánru. Více o problematice dále v mé práci.

## 2.2 Intervaly

Interval v hudbě vyjadřuje výškovou vzdálenost dvou tónů. Všechny možné intervaly v rámci dvou oktáv jsou pojmenovány. Tyto termíny jsou pak v analýze akordů a harmonie hojně využívány. Výčet intervalů uzpůsobím do technicky srozumitelné podoby, v jaké se pak na ně odkazují dále ve své práci. Vždy uvedu číslo (v tabulce č. 1 sloupec vzd.) značící vzdálenost dvou tónů v půltónech a poté název intervalu, který tuto vzdálenost označuje.

vzd.	název intervalu	vzd.	název intervalu
0	(čistá) prima	12	(čistá) oktáva
1	malá sekunda	13	malá nona
2	(velká) sekunda	14	(velká) nona
3	malá tercie	15	malá decima
4	(velká) tercie	16	(velká) decima
5	(čistá) kvarta	17	(čistá) undecima
6	zvětšená kvarta, zmenšená kvinta	18	zvětšená undecima, zmenšená duodecima
7	(čistá) kvinta	19	(čistá) duodecima
8	zvětšená kvinta, malá sexta	20	zvětšená duodecima, malá terdecima
9	(velká) sexta	21	(velká) terdecima
10	malá septima	22	malá kvartdecima
11	(velká) septima	23	(velká) kvartdecima

**Tabulka 1: Názvy intervalů uváděných v počtu půltónů.**

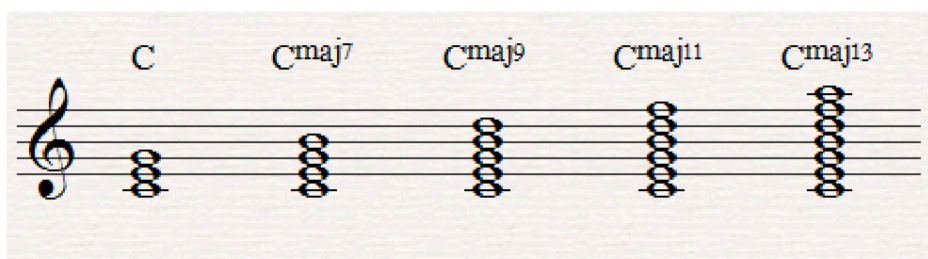
## 2.3 Jazzové akordy, jejich tenze a akordové značky

Akord a jeho co nejpřesnější zachycení pomocí správné akordové značky je v jazzu, co se týká harmonie, jedna z nejdůležitějších věcí.

Obecně můžeme říci, že akord je souzvuk minimálně třech a maximálně sedmi tónů. Podle počtu tónů rozlišujeme:

- kvintakordy
- septakordy
- nonakordy
- undecimové akordy
- terdecimové akordy

Někdy se jim také říká terciové akordy, protože vznikají z tónů skládaných na sebe v terciových intervalech. **Kvintakord**, ze kterého všechny širší akordy vycházejí, je složen z toniky, tercie a kvinty. Přidáváním dalších lichých intervalů (septima, nona, undecima, terdecima) dostáváme ostatní zmíněné druhy (viz obr. 2).

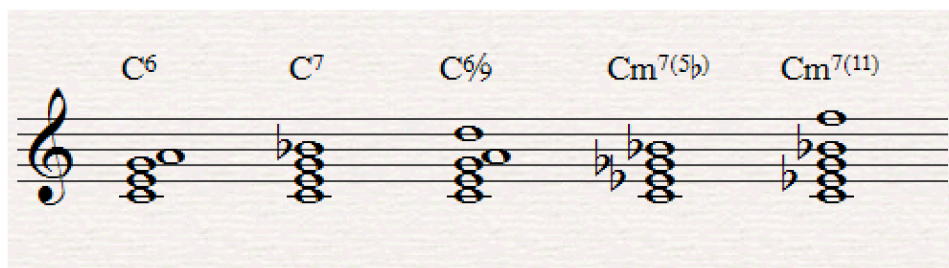


Obrázek 2: Nealterované akordy vycházející z durového kvintakordu.

Kvintakord může být **durový** nebo **molový** v závislosti na tom, jakou obsahuje tercii. Durový obsahuje velkou tercii, molový pak malou. Všechny další intervaly mohou být různě alterované (snížené či zvýšené) a my tak budeme dostávat další akordy.

Základním jazzovým akordem bývá označován septakord, nebo obecně akord tvořený ze čtyř tónů. Další tóny přidávané k septakordu se nazývají **tenze** [8]. Intervaly nona, undecima a terdecima jsou tedy tenze. Někdy bývá uváděna jako tenze i septima, která tvoří septakord z kvintakordu.

Hojně jsou však používány i jiné kombinace kvintakordu a přidaných tenzí. Například durový kvintakord s přidanou terdecimou je vlastně velmi známá „šestka“. Pokud k němu přidáme ještě nonu, dostaneme další často používaný jazzový akord. A tak bychom mohli pokračovat velmi dlouho. Různé příklady jazzových akordů tvořených tenzemi vidíme na obr. 3.



Obrázek 3: Jazzové akordy s tenzemi

Na uvedených obrázcích jsou znázorněny i značky akordů. Popíšu zde jejich základní systém a případné zkratky.

Základní tón akordu je značen velkým písmenkem a udává, od kterého tónu budeme akord stavět. Je-li za ním písmeno „m“ nebo řetězec „mi“, znamená to, že akord je molový a přidáme mu malou tercii. Jinak přidáme tercii velkou. Obsahuje-li akordová značka „-“, „5b“ nebo „5-“, přidáme zmenšenou kvintu. Najdeme-li „+“, „5#“ nebo „5+“, přidáme kvintu zvětšenou. Jinak bude kvinta čistá.

Tím máme postavený kvintakord. Všechny ostatní prvky v akordové značce, které jsme dosud nevyužili, značí přidané tenze. Čísla 7, 9, 11, 13 značí vždy příslušnou tenzi. Pokud jsou uvedeny v závorce, za lomítkem nebo s předponou „add“, jsou přidány pouze tóny, které značí. Jinak jsou automaticky doplněny všechny tenze předchozí (viz obr. 2). Vždy platí, že tenze může být zvýšená znakem „#“ nebo „+“ a snižená znakem „b“ nebo „-“. Septima (7) je použita vždy malá, pokud není ve značce „maj“. Pokud je obsažena pouze terdecima, může se značit číslem 13, ale také 6.

Pak jsou tu speciální akordy bez tercie (tzv. suspendované). Mají ve značce „sus4“ nebo „sus2“ podle toho, zda je tercie nahrazena kvartou nebo sekundou. Také nutno zmínit zmenšený septakord značený „dim“ či „dim7“ poskládaný pouze z malých tercií.

Často jsou používány zkratky ve značení akordů. Z těch nejpoužívanějších například trojúhelník značí „maj“, podtržítko „m“, kroužek „dim“, přeškrtnutý kroužek „m<sup>5b</sup>“.

## 2.4 Aranžování versus kompozice

Každá závažnější hudební myšlenka, zvláště když má být interpretována početnějším seskupením hudebníků, musí být nejen zanesena do not, ale i upravena v závislosti na nástrojovém složení interpretů. Měli bychom rozlišovat pojmy **kompozice** a **aranžování**.

Cílem **kompozice** je hudební myšlenku vytvořit a postavit na ní skladbu či píseň. Skladatel většinou skladbu komponuje pro nějaké konkrétní nástrojové složení. V podstatě tvoří notový zápis pro všechny nástroje. Každý nástroj má svůj notový part neboli hlas a jednotlivé hlasy tvoří dohromady harmonii skladby.



V případě, že chceme skladbu interpretovat hudebním tělesem jiného nástrojového obsazení, musíme ji upravit – **aranžovat**.

**Aranžování** je tedy proces, kdy vezmeme existující skladbu a transformujeme ji do podoby pro jiný počet, případně i jiné druhy, hudebních nástrojů. Přitom se snažíme vystihnout stejnou hudební myšlenku. Laicky řečeno, jde o to, aby interpretace takto upravené skladby jednoznačně připomínala skladbu původní.

Aranžováním ovšem můžeme nazvat i fázi, kdy komponista již má představu o melodii a harmonii své nové skladby, a chce ji nyní napsat pro konkrétní hudební těleso. V tomto případě by moje výsledná aplikace mohla posloužit také. Budu tedy ve své práci hovořit o aplikaci výhradně jako o pomůcce pro aranžéra, jemuž může podstatně urychlit práci.

## 3 Metodika jazzového aranžování

V této kapitole jsou popsány různé možnosti, taktiky a postupy používané při aranžování jazzové hudby pro konkrétní nástrojová složení. Je zaměřena na tvorbu harmonických hlasů dle akordů skladby a na vytváření basového partu. Informace byly čerpány především z [1], [3], [8] a [13].

### 3.1 Bloková harmonizace

Pojem bloková harmonizace je volný překlad z anglického *Block Voicings* a jedná se o často používanou metodu pro harmonizování nástrojových sekcí.

Termín *Voicings* znamená harmonizování melodie několika nástroji či hlasy (angl. *voices*) buď ze stejné nástrojové sekce anebo zkombinováním sekcí. Pokud se jedná o harmonizaci blokovou, znamená to, že vnitřní harmonické hlasy se pohybují stejným směrem jako melodie. Ta je hlasem prvním a hraje ji nejvýše položený či nejhlasitější nástroj sekce, což je v našem případě 1. alt saxofon. Tento typ harmonie funguje dobře právě pro typický jazzový big band, ale i v mnoha dalších stylech popu a rocku, například pro vokální seskupení atd. Nás ovšem v rámci této práce bude zajímat pouze saxofonová sekce big bandu.

#### 3.1.1 Princip blokové harmonizace

Základní princip spočívá v rozdělení tónů melodie na dvě skupiny. Noty **cílové** a noty **přechodové** (přeloženo z angl. *target notes* a *passing notes*). Cílové noty jsou harmonizovány akordem, který jim byl přiřčen, a noty přechodové jsou tzv. reharmonizovány, neboli harmonizovány jiným vhodně zvoleným akordem (podrobněji v podkapitole 2.1.4). Pro samotnou harmonizaci se pak používá několik metod, z nichž některé uvedu v dalších podkapitolách. Jedná se především o čtyřhlasou úzkou nebo širokou harmonii.

Pokusím se blíže vysvětlit rozdělení not na cílové a přechodové. V jakémkoli místě skladby vždy tvoří všechny hlasy dohromady nějaký akord. Jak již bylo zmíněno, v rámci blokové harmonizace se všechny vnitřní hlasy pohybují s melodií, což znamená zharmonizovat samostatně každý tón vhodným akordem.

Ve své práci počítám s tím, že na vstupu je melodie a k ní sled akordů. Každý z těchto akordů většinou přísluší fragmentu melodie, tedy několika tónům. Vezměme například fragment z již jednou zde uvedené známé písně (viz obr. 4).



Obrázek 4: Fragment melodie. Akordické noty.

Pro všechny noty platí akord C-dur, který obsahuje tóny {c, e, g}. Uvedená část melodie však obsahuje i tóny {d, f}, které v akordu C-dur obsaženy nejsou. Pokud má být při jejich harmonizaci zachován pohyb i ve vnitřních hlasech, musíme pro ně použít jiný akord. Jedná se o noty přechodové.

Neplatí však vždy, že přechodové noty jsou ty, které nejsou obsaženy v akordu daném k melodii. Vždy záleží na několika aspektech, jež je možno vyjádřit několika pravidly. Na následujícím příkladu (obr. 5) se pokusím nejzákladnější z nich popsat.

Jedná se již o příklad jazzové hudby. Vyznačuje se rozmanitým rytmem i složitějšími akordy. Dané akordy navíc mohou být vhodně rozšířeny o jejich tenze (tenze viz kapitola 1.3), což nám umožní třeba harmonizovat noty melodie akordem, v němž nejsou obsaženy.

Dále je zde vidět prvek, který váže dvě noty k sobě, zvaný **ligatura** a značí se obloučkem spojujícím dané dvě noty. Takové dvě noty jsou pak chápány jako jedna, která má délku rovnající se součtu obou not.

Ligatura se často využívá k efektu zvanému *anticipace*. Jedná se o situaci, kdy nota, během které se mění akord, začíná již v předchozím akordu. U not, u kterých se během jejich trvání změnil akord v melodii, je použit ten poslední – je jakoby předejmut. To platí na všech místech v obr. 5, kde byla použita ligatura.

Obecně, cílové noty bývají dlouhé či nějak zdůrazněné akordové noty nebo jejich tenze. Naopak, přechodové noty bývají krátké a jakoby pocitově směřující k cílovým notám. Také hraje jistou roli situace, když se za notou mění akord. Pokud je navíc nota krátká, bývá často označena jako přechodová.

Můžeme tedy zapsat tyto pravidla, kdy nota může být cílová:

- Jedná se o tón daného akordu.
- Jedná se o tenzi daného akordu.
- Má větší délku.
- Následující nota má stejný akord.

Pravidla mohou mít různou váhu a mohou platit zároveň. Logicky, čím více pravidel platí, tím spíše se bude jednat o cílovou notu a naopak.



Obrázek 5: Rytmicky složitější melodie s jazzovými akordy.

V obr. 5 jsou znakem „X“ označeny noty cílové a znakem „O“ noty přechodové. Třetí cílová nota {e} od konce sice není obsažena v daném akordu G<sup>7</sup> {g, h, d, f}, nicméně se jedná o jeho tenzi 6 (případně 13) a navíc je oproti ostatním notám poměrně dlouhá. Ostatní cílové noty jsou obsaženy v daných akordech. První přechodová nota {d} by sice mohla být tenzí 9 akordu C<sup>6</sup> {c, e, g, a}, ale je příliš krátká a směřuje k cílové notě {e}. Stejný případ platí pro poslední přechodovou notu {g}. Druhá přechodová nota {f} je navíc poslední nota akordu C<sup>6</sup>, tvoří jakýsi přechod k akordu A<sup>7(b9)</sup>.

### 3.1.2 Čtyřhlasá harmonie

Pro aranžování jazzového vícehlasu, neboli pro tvorbu *voicings*, můžeme najít několik používaných postupů, které popisují různé systémy vybírání či skládání tónů do vnitřních hlasů. Předpokládá se samozřejmě, že máme jasno, jakým akordem chceme harmonizovat notu melodie.

V jazzu se většinou vyskytují akordy o nejméně 4 tónech. Pokud harmonizujeme melodii pro sekci o čtyřech či více nástrojích, provádíme to čtyřhlasou harmonií.

#### Úzká harmonie (Close Voicing)

V případě čtyřhlasé úzké harmonie postupujeme tak, že zapíšeme tón melodie a pod něj naskládáme tři tóny z příslušného akordu pod sebe, jak nejtěsněji to jde, bez vynechání žádného (odtud pojem úzká harmonie). Na obr. 6 můžeme vidět situaci, kdy je tón melodie {e} doplněn třemi tóny akordu C<sup>6</sup> do těsného čtyřhlasu.



Obrázek 6: Ukázka úzké harmonie.

## Otevřená harmonie (Open Voicing)

Tvorba otevřené čtyřhlasé harmonie vychází z harmonie úzké. Nejprve vytvoříme úzkou harmonii k danému tónu melodie a poté provedeme tzv. *drop 2* nebo *drop 3*. Znamená to, že druhý nebo třetí tón od shora přepíšeme o oktávu níže. Na obr. 7 vidíme případ z předchozího obrázku v otevřené harmonii. Zde byl použit *drop 2*.



Obrázek 7: Ukázka otevřené harmonie.

V praxi se úzká a otevřená harmonie často střídají. Celé pasáže můžeme harmonizovat buď otevřeně či úzce, nebo lze obě metody kombinovat. Změna způsobu harmonizování zní dobře například v případech, kdy je v melodii větší skok.

### Pravidla pro čtyřhlasou harmonii

Z odborné literatury (např. [3] nebo [8]) bych zde mohl vypsát řadu pravidel, výjimek a možností, jak čtyřhlasou harmonii vést. Avšak kromě toho, že zde na to není dostatek prostoru, se povětšinou jedná spíše o intuitivní tahy aranžéra jako reakce na aktuální kontext. V následujícím seznamu uvádím proto jen ty, které považuji za nejeftektivnější.

Pravidla čtyřhlasé harmonie:

- Mezi prvním a druhým hlasem by neměl být půltón.
- Akordy rozšiřuj vhodně o tenze 7, 9, 13.
- Pokud je pro akord maj7 v melodii tonika, použij akord 6.
- U pěti a vícetónových akordů můžeme vypustit toniku nebo kvintu, případně nonu, pokud nejsou v melodii.

### 3.1.3 Pětihlasá harmonie

Vychází ze stejného principu jako čtyřhlasá harmonie. Pátý hlas získáme několika způsoby, uvedu zde však jen dva.

První případ je, že vytvoříme čtyřhlasou úzkou harmonii a do pátého hlasu zapíšeme melodii sniženou o jednu oktávu. Druhý způsob využívá čtyřhlasou otevřenou harmonii a pátý hlas plní funkci basu, obsahuje základní tóny akordů (toniky).

Saxofonová sekce big bandu sestává z pěti saxofonů, dva alt saxofony, dva tenor saxofony a jeden baryton saxofon. Jedná se tedy o pět hlasů, které jsou tvořeny následovně. První alt saxofon má první hlas, což je melodie. Stejný hlas, pouze o jednu oktávu níže, je psán pro baryton saxofon. Oba tyto krajní, téměř identické hlasy, zapouzdřují vnitřní tři další harmonické hlasy pro ostatní tři saxofony.

### **3.1.4 Reharmonizace přechodových not**

Úkolem reharmonizace je harmonizovat přechodové noty vhodně zvoleným akordem tak, aby nenarušovali harmonii melodie a jen poskytovali přechod mezi cílovými notami. Řídíme se vždy následující cílovou notou. Podle ní je postaven akord, který bude použit pro harmonizaci přechodové noty. Akord, jenž je výsledkem reharmonizace, může být aplikován na příslušnou notu melodie metodou čtyřhlasé či pěťhlasé harmonie.

Standardní reharmonizační techniky popsané níže jsou chromatická, diatonická, paralelním akordem a dominantní septakordem [1].

#### **Chromatická reharmonizace**

Použije se v případech, kdy je naše přechodová nota vzdálená od následující cílové noty jen o jeden půltón. Jedná se o chromatický postup v melodii. Provádí se tak, že všechny hlasy se posunou o jeden půltón níže či výše, než jsou jejich odpovídající následovníci. Samozřejmě jsou posunuty ve stejném směru jako melodie.

#### **Diatonická reharmonizace**

Lze použít, pokud přechodová nota melodie přísluší tónině skladby, aneb je-li diatonická. Každý hlas je posunut o jeden diatonický krok směrem od svého korespondujícího následovníka příští cílové noty. Diatonický krok je následující tón ve stupnici dané tóninou skladby.

#### **Reharmonizace paralelním akordem**

Tato technika může být použita kdykoliv. Je velmi podobná technice chromatické reharmonizace s tím rozdílem, že aktuální přechodová nota je od následující cílové noty vzdálena o více než jeden půltón. O stejný krok musí být posunuty všechny harmonické hlasy.

#### **Reharmonizace dominantním septakordem**

Dominantní septakord se tvoří tak, že vezmeme základní tón akordu následující cílové noty a přičteme k němu 7 půltónů. Tento tón vezmeme jako základní tón septakordu. Máme-li například akord následující cílové noty C<sup>6</sup>, dominantní septakord přechodové noty bude G<sup>7</sup>. Použití

dominantního septakordu je podmíněno tím, že tón melodie v něm musí být obsažen. Pokud toto není splněno, jsme nuceni použít jinou z uvedených technik.



Obrázek 8: Výsledek blokové harmonizace.

Na obr. 8 vidíme výsledek blokové harmonizace pro melodii s akordy z obr. 5. Vrchní hlas je původní melodie. Aranžmá je psáno o oktávu výš, čistě jen pro tento případ, aby se všechny vnitřní hlasy vešly do notové osnovy. Vidíme tedy použití čtyřhlasé úzké harmonie na všechny cílové noty. První a třetí přechodová nota je reharmonizována dominantním septakordem. Druhá přechodová nota pak paralelním akordem k akordu následujícímu.

## 3.2 Aranžování basu

V cílech práce jsem uvedl, že svůj pěťhlas doplním o basový part. Ten může mít mnoho podob v závislosti na různých žánrech hudby. Zaměříme-li se na jazzový žánr, stále existuje více stylů vedení basu, které postupně vznikaly spolu s vyvíjením jazzu. Principiálně ale můžeme vyzorovat dva styly, které jsou v podstatě základem swingové hudby. [1] [13]

### 3.2.1 Vedení basu stylem *2 in a bar*

Tento styl vznikl někdy ve dvacátých letech dvacátého století v raném swingu. Název stylu přeložený do češtiny znamená „dvě v taktu“, což už nám napovídá, že bude tvořen půlovými notami ve čtyřčtvrtečním taktu.

Používají se především toniky a kvinty, výjimečně tercie. Při změně akordu o kvartu vzhůru nebo o kvintu dolů by měl být v basu postup tonika-tonika, tercie-tonika nebo kvinta-kvinta. Při jiné změně akordu je postup basu tonika-tonika. Tím pravidla pomalu končí. Zbývá jen dodat, že by v basu neměl být skok větší než jedna oktáva, a také, že bas by měl začínat a končit na tonice.

V tomto typu basu se často vyskytují i kratší noty, jak pro efekt, tak pro účelové důvody například při rychlejších akordových postupech. Příklad takového basu lze vidět na obr. 9.



Obrázek 9: Ukázka basu 2 in a bar.

### 3.2.2 Vedení basu stylem *Walking bass*

Jedná se o styl, který vznikl o něco později než styl „2 in a bar“. Pochází zhruba z třicátých let dvacátého století. Anglický název „Walking bass“ znamená „kráčející bas“. Bas psaný tímto stylem je tvořen čtvrtřovými notami ve čtyřčtvrtečním taktu.

Tento typ basu je tvořen rozklady akordů a jejich obraty, ale také dalšími tóny odvozených ze stupnic daných akordů. Platí zde několik následujících pravidel. První tón v akordu by měl být tonika a poslední tón v akordu by měl jakoby vést k následujícímu tónu. Ostatní noty jsou akordické (tonika, tercie, kvinta, sexta, septima) nebo přechodové, což jsou noty většinou o půltón výše či níže než následující nota. Během akordu můžeme tóny opakovat, nikoliv však při změně akordu. Na obr. 10 vidíme příklad takového basu.



Obrázek 10: Ukázka stylu *walking bass*.



# 4 Programování s omezujícími podmínkami

Technika programování s omezujícími podmínkami (**CP – Constraint Programming**) má trochu jiný přístup než klasické programovací techniky. U běžného programování musí programátor danou úlohu umět vyřešit sám. Pak může vytvořit program, který se řídí daným postupem, čili v podstatě naučí počítač řešit úlohu. Technika CP se používá u problémů, kdy programátor neumí, nebo někdy ani nemůže, definovat jednoznačný postup řešení. Definuje tedy vstupní hodnoty, omezení pro výstupní hodnoty a implementuje metodu, která dle jeho názoru nejlépe najde řešení [6].

Aranžování či harmonizace hudby má v podstatě, ač si to aranžér nemusí uvědomovat, základy v matematice. Nejmenší možný krok mezi dvěma tóny je jeden půltón. Vzdálenost mezi jakýmkoli dvěma tóny (interval) můžeme definovat jako počet půltónů mezi nimi (viz kapitola 1.2). Navíc můžeme tónům přiřadit přirozená čísla, podle toho, o kolik půltónů jsou vzdáleny od nejspodnějšího tónu, což může být třeba námi stanovená hranice.

Obecně nám tedy při harmonizaci jde o to, abychom určili pod každý tón melodie hodnotu tónu pro všechny spodní hlasy. Hodnoty, kterých mohou tóny nabývat, jsou vybírány z předem definované množiny. Postup harmonizace lze do jisté míry transformovat do konečného množství pravidel, neboli omezujících podmínek, pomocí kterých budou hodnoty vybírány. Aplikováním pravidel omezujeme vstupní množinu hodnot.

V takovém případě hovoříme o problému splňování podmínek (**CSP – Constraint Satisfaction Problem**), který je formálně definován jako:

- konečná množina proměnných
- každé proměnné je přiřazena konečná doména, tj. množina možných hodnot
- konečná množina podmínek omezujících hodnoty, které mohou proměnné současně nabývat

Hlavními zdroji informací o CP a CSP pro mou práci byli [6], [7] a [12].

## 4.1 Metody řešení CSP

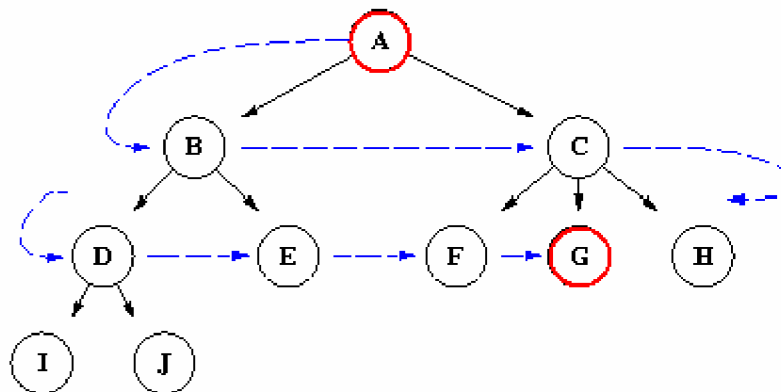
Řešením CSP problému je ohodnocení všech proměnných, tj. vybrání hodnoty proměnné z příslušné domény tak, že všechny podmínky jsou splněny. Někdy stačí nalézt jediné přípustné řešení, jindy se požaduje nalézt všechna řešení a často se setkáme s hledáním optimálního řešení, kde kvalita řešení je určena objektivní funkcí nad proměnnými problému. [6]

Pro nalezení řešení se většinou používají metody prohledávání stavového prostoru. Jejich princip spočívá ve vhodném procházení stavů řešené domény za účelem nalezení požadovaného stavu. Stavový prostor nám vznikne, pokud si řešenou doménu rozdělíme do různých stavů a definujeme, že jeden z těchto stavů je počáteční, některé stavy jsou cílové a mezi různými stavy je možné aplikováním určitých akcí přecházet. Nejčastěji bývá reprezentován orientovaným grafem, jehož uzly jsou stavy a přechody udávají akci, jejímž vykonáním se dostaneme z jednoho stavu do druhého. Nalezení řešení spočívá pak v nalezení cesty v grafu mezi počátečním a cílovým uzlem.

Algoritmy prohledávání stavového prostoru se obecně dělí na metody prohledávání do hloubky (**DFS - Depth First Search**) nebo do šířky (**BFS - Breadth First Search**).

## 4.2 Algoritmus prohledávání do šířky - BFS

Algoritmus BFS prochází nejprve všechny následníky daného uzlu a poté zase všechny následníky těchto následníků, dokud nedojde k cílovému uzlu. Přitom si poznamenává předchůdce jednotlivých uzlů a tím je poté vytvořen strom nejkratších cest k jednotlivým uzlům ze startovního uzlu (viz obr. 11).

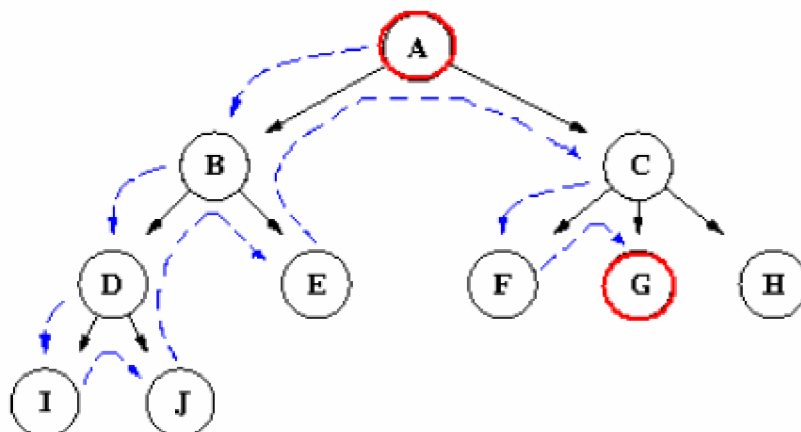


Obrázek 11: Prohledávání stavového prostoru do šířky.

Při tomto způsobu prohledávání máme jistotu, že vždy nalezneme koncový stav, ovšem za cenu projití značného počtu uzlů. Každý uzel je navštíven nejvýše jednou. Pro řešení CSP není příliš efektivní.

## 4.3 Algoritmus prohledávání do hloubky - DFS

Jedná se o metodu pro řešení CSP daleko používanější. Na rozdíl od BFS tento algoritmus prochází grafem do hloubky tak, že vezme prvního následníka, aniž by procházel všechny ostatní (viz obr. 12).



Obrázek 12: Prohledávání stavového prostoru do hloubky.

Tento způsob prohledávání může vést k cíli mnohem rychleji, než u BFS, zvláště když se vydáme správným směrem. Ovšem v jeho základní podobě nemáme zaručeno, že se neocitneme ve slepé větvi, která nevede k cíli. Proto bývá doplněna kontrola, zda se v grafu ubíráme správným směrem. Ta bývá buď dopředná (tj. varianta DFS zvaná **Forward checking**), či zpětná (**Backtracking** DFS algoritmy).

### 4.3.1 Forward checking

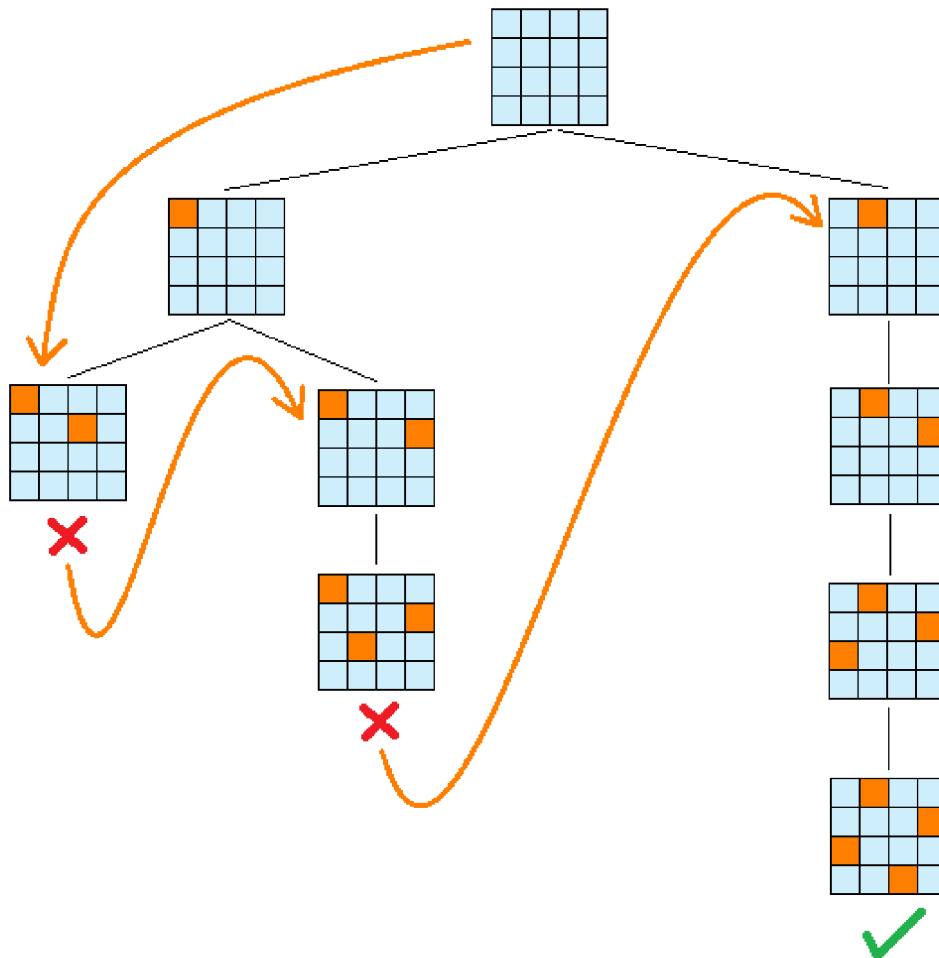
Zkoumá, zda z následníka, do něhož hodlá přejít, ještě může cesta vést k cíli. Pokud zjistí, že ne, totéž vyhodnotí u dalšího následníka aktuálního uzlu. Vyhodnocení provádí tak, že se pokusí aplikovat dosud nepoužitá pravidla (omezující podmínky) na tzv. budoucí proměnné a kontroluje, zda doména některé z těchto proměnných není prázdná.

### 4.3.2 Backtracking

Liší se od předchozího algoritmu tím, že do následníka nejprve přejde a teprve až poté zkoumá, zda si tím neodřízl cestu k cíli. V případě, že ano, vrací se k předchozímu uzlu a vydává se jinou cestou. Pokud i tam již vyčerpal všechny možnosti, jde o další krok zpět. V kontextu domén a proměnných to vypadá tak, že v případě, kdy při přiřazování hodnot do volných proměnných narazí algoritmus na nekonzistenci, vrátí se k poslední ohodnocené proměnné a přiřadí jí jinou hodnotu. Pokud nastane takový případ, že dané proměnné již nelze přiřadit žádnou hodnotu z její domény, vrací se algoritmus k předchozí proměnné.

Typickým, často uváděným, příkladem pro backtracking je problém rozmístění  $N$  dam na šachovnici o rozměrech  $N \times N$  tak, aby se žádné dvě z nich vzájemně neohrožovaly. Ukázkou průchodu

grafem tohoto příkladu je možno vidět na obr. 13. Jsou znázorněny pouze stavy, které vyhovují podmínkám.



Obrázek 13: Použití Backtrackingu na problém N dam na šachovnici  $N \times N$ .

## 4.4 Optimalizace metod řešení CSP

Pro zvýšení efektivity prohledávacích algoritmů se používá několik vylepšení. Stavový prostor může být podstatně zmenšen na základě znalosti podrobností daného problému. Také můžeme ovlivnit pořadí proměnných, v jakém do nich budou přiřazovány hodnoty.

Jedná se například o techniku zvanou **First Fail**. Jejím cílem je na počátku vyřadit možnosti, které nevedou k řešení. Často tedy bývá vhodné začít s proměnnou, která má nejmenší doménu či která má nejvíce omezujících podmínek. Opakem je technika **Succeed First**, která se snaží od začátku vybírat takové hodnoty, které nejméně omezí domény zbývajících proměnných.

## 4.5 Knihovna Gecode

Dnes již existuje plno nástrojů podporujících a usnadňujících řešení CSP. Lze v nich deklarovat vstupní proměnné a jejich domény, omezující podmínky ve formě pravidel. Často obsahují i různé algoritmy prohledávání stavového prostoru. V takovém případě se jedná o nástroj obecně zvaný Constraint Solver.

Po zvážení několika aspektů jsem zvolil Constraint Solver v podobě knihovny Gecode. Je psána v C++ a je volně šiřitelná. Pro vytvoření omezujících podmínek poskytuje tzv. *propagátory*. S jejich pomocí lze realizovat nejrůznější tvary pravidel. Samozřejmostí jsou logické operace nad booleovskými i číselnými hodnotami, lineární matematické operace a řada dalších. Pro nás jsou nejdůležitější operace logické a rovnost.

Gecode nabízí volbu mezi DFS a BFS, jakožto algoritmy pro hledání řešení. Poskytuje také řadu tzv. *brancherů*, kterými lze upřesnit, jakým způsobem budou vybírány proměnné a jejich hodnoty při hledání řešení. To nám bude prospěšné k nalezení rozličných vzorků řešení výsledného aranžmá.

Otevřenost knihovny navíc podporuje implementaci vlastních propagátorů i vyhledávacích strategií.

## 5 Návrh systému

Výsledná aplikace mé práce by měla sloužit jako pomůcka při aranžování skladeb metodou blokové harmonizace (viz kapitola 2). V této kapitole popíšu, jak jsem navrhl celý algoritmus a jednotlivé jeho součásti. Také jsou rozebrány možnosti reprezentace hudebních dat. Dále nastíním, jak bylo využito programování s omezujícími podmínkami a jak byl navrhnut vstup a výstup programu.

### 5.1 Reprezentace hudebních dat

Je třeba se zamyslet nad tím, s jakými hudebními informacemi budeme pracovat, abychom mohli poté navrhnout vhodné datové struktury pro jejich uchování.

Na vstupu aplikace je třeba načíst melodii a uchovat informace nejen o její melodické ale i rytmické stavbě. Melodická stavba je dána výškami tónů a rytmická stavba jejich délkami.

#### 5.1.1 Reprezentace noty a jejích parametrů

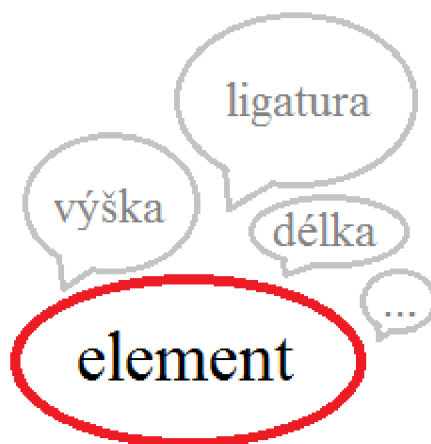
V notovém zápisu existuje kromě výšky a délky spousta dalších parametrů a vlastností, které může nota mít. Jsou to ale většinou různé značky pro hudebníka, které mu sdělují, jakým stylem má notu zahrát, nebo jakým efektem ji ozdobit. Pro naše účely tedy nemají žádný hlubší význam a my budeme používat jen minimum z nich.

Je však více než vhodné jeden z těchto parametrů vyzdvihnout a použít. Jedná se o tzv. *ligaturu* (angl. *tie*), oblouček, spojující délky dvou sousedních not stejné výšky. Hudebník je chápe jako jednu notu o délce rovnající se součtu délek obou těchto not. Ligatura je použita v případech, kdy chceme zapsat notu, která by svou délkou přesáhla do dalšího taktu. Takovou notu vhodně rozdělíme a svážeme ligaturou. Má však opodstatnění i uprostřed taktu, pokud chceme zachovat průzračnost složitého rytmu, který by byl v jiném, méně zvyklém, zápisu hůře čitelný. Oba tyto případy použití ligatury (viz obr. 14) jsou velmi typické pro zápis jazzové hudby, kterou se ve své práci zabývám.



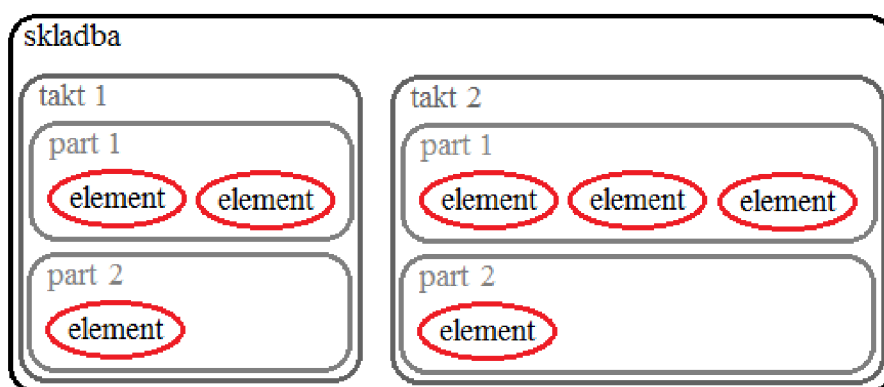
Obrázek 14: Užití ligatury v zápisu hudby.

V zápisu hudby se ale vyskytují také pauzy, místa, kdy hudebník nemá hrát. Pauza může být v našem případě interpretována jako nota, u níž nás bude zajímat pouze délka. Nota (nebo pauza) a všechny její parametry lze zapouzdřit do jednoho tzv. **elementu** (viz obr. 15), který je pak základní jednotkou zapisované hudby.



**Obrázek 15: Nota či pauza a její vlastnosti zapouzdřené v elementu**

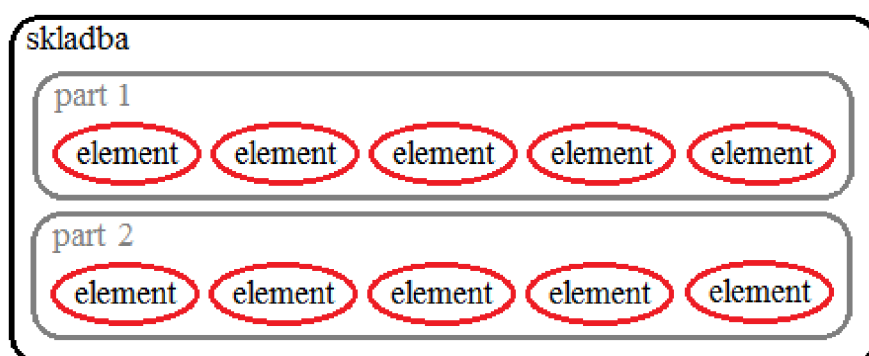
Používá se více přístupů, jak uložit notový zápis, respektive jak skládat jednotlivé elementy a jaké souvislosti mezi nimi tvořit. Záleží vždy na tom, jak budeme data dále zpracovávat a které jejich vlastnosti či souvislosti jsou pro nás důležitější. Jedna z možností je členit je po jednotlivých hlasech (partech), dále pak po taktech (viz obr. 16). Je pak snadné se dotázat na konkrétní takt a zjistit například jaké rozdíly v rytmu jsou mezi konkrétními hlasy.



**Obrázek 16: Reprezentace hudebního zápisu s členěním na takty.**

V našem případě je však situace jednodušší v tom, že při harmonizaci nepotřebujeme řešit rytmickou stavbu jednotlivých vnitřních hlasů. Jednoduše proto, že je stejná jako v melodii. Navíc, potřebujeme jenom být schopni opět zrekonstruovat na výstupu aplikace melodii do její původní podoby. Tam musíme umět zapsat nejen znovu melodii, ale i další 4 harmonické hlasy, které budou mít stejný rytmus jako melodie, ovšem jiné výšky tónů. Basový part bude generován zvlášť, téměř bez závislosti na stavbě melodie.

Nemá tedy smysl vymýšlet složité struktury. Informace, se kterou v průběhu harmonizace budeme pracovat, bude výška a délka tónu, vzdáleného od začátku skladby o stejný počet elementů (obr. 17). Snadno se tedy na ně dotážeme.



Obrázek 17: Reprezentace hudebního zápisu bez členění na takty.

### 5.1.2 Výška noty

Nyní je třeba najít vhodný systém pro určení výšky tónu. Z fyziky víme, že výška tónu je dána frekvencí zvukové vlny, která ho tvoří. Mohli bychom tedy ukládat frekvence. Ovšem také víme, že rozložení frekvencí jednotlivých po sobě jdoucích tónů či půltónů není lineární. Zabředli bychom tedy zbytečně do neprimitivního dopočítávání frekvencí dalších tónů. Navíc vstupem naší aplikace nebude vzorkovaný zvuk, ale přímo hudební zápis inspirovaný hudební teorií. Můžeme tedy tóny snadno reprezentovat číselnými hodnotami. V našem případě se můžeme spolehnout na fakt, že nejmenší vzdálenost mezi dvěma po sobě jdoucími tóny je jeden půltón. To však není zdaleka celosvětově stanoveno. V některých východních národech není výjimkou čtvrttón nebo ještě menší interval. V takovém případě bychom byli nuceni použít pro reprezentaci reálná čísla či zlomky.

My však můžeme tónům přiřadit přirozená čísla, dle nichž bude jednoznačně definována jejich výška. Nejen o který tón jde {c, cis, d, dis, ...}, ale i v jaké je oktávě. Platí pak, že tóny vzdálené o hodnotu 1, jsou od sebe výškově vzdáleny o 1 půltón.



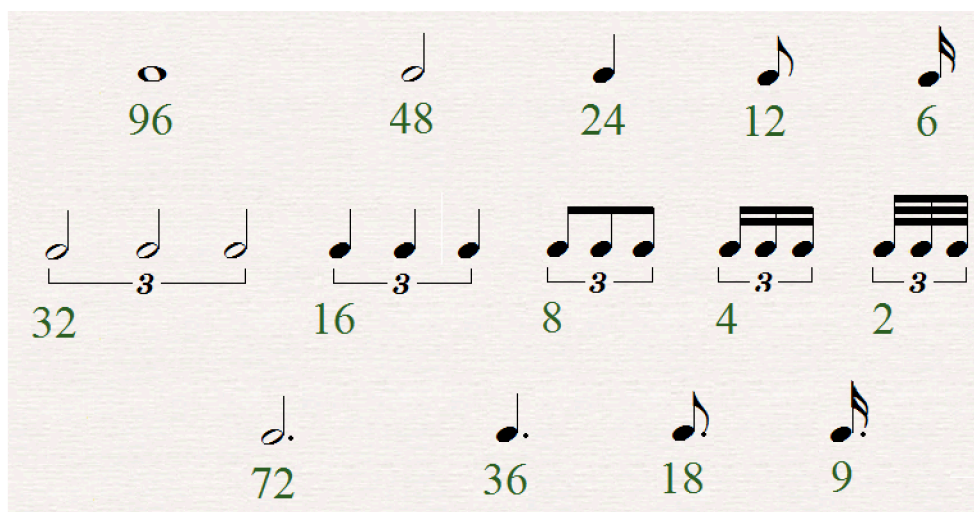
Pro volbu hodnot, jakými osadit tóny v rozsahu, ve kterém se budeme při harmonizaci pohybovat, se přímo nabízí inspirace mezinárodním standardem MIDI. Ten umožňuje zápis hudby v digitální formě. Všechny hudební informace, jako jsou výška a délka noty či nástroj, který má právě hrát, mají své číselné označení. MIDI popisuje rozsah 127 tónů v 11 oktávách.

Jednočárkovaná oktáva začíná na čísle 60 a „komorní a“ má číslo 69. My se budeme pohybovat okolo těchto hodnot. Basový part pak bude čerpat hodnoty z rozsahu zhruba od 24 do 48, což odpovídá poloze basových nástrojů.

### 5.1.3 Délka noty

Potřebujeme ale uchovat nejen výšku tónu, nýbrž i jeho délku. Zde se nabízí dva přístupy. Délku, jako časový interval, můžeme měřit v sekundách. My však nepotřebujeme znát trvání noty v jednotkách času, ale spíše relativně v rámci skladby. Využijeme tedy jiný přístup. Rychlost neboli tempo skladeb je měřeno na tzv. *doby* a v dobách jsou udávány délky not.

Nejdelší nota je nota celá a délky ostatních not lze získat dělením dvěma nebo třemi. Když celé notě přiřkneme hodnotu 96, postačí to, abychom vyjádřili všechny používané délky not přirozeným číslem (viz obr. 18).



Obrázek 18: Hodnoty délek tónů.

Když už víme, jak uložit tóny, umíme tedy uložit melodii, kterou bude mít aplikace na vstupu. Mimo melodie však bude na vstupu také posloupnost akordů, kterými má být melodie harmonizována a které je třeba také nějakým způsobem uložit. Právě akordy vyjadřují harmonii skladby, každý typ akordu má svoji harmonickou funkci. Typ akordu v podstatě definuje skupinu intervalů, které nám určí konkrétní tóny v závislosti na základním tónu akordu.

K jednoznačné identifikaci akordu tedy potřebujeme jeho základní tón (C, Cis, D, ...) a typ (Major, Minor, Maj7, 6, ...). Základní tón může být určen přirozeným číslem v rozsahu 0 až 11, což jsou vlastně hodnoty MIDI standardu v nejnižší oktávě. Oktáva u akordů nehraje roli.

Typy budeme mít někde vhodně nadefinované. Postačí nám mít zaznamenány intervaly (tóny) obsažené v daném typu akordu v rozsahu 0 až 11, kde 0 bereme za první tedy základní tón. Nebude potom problém dopočítat konkrétní tóny každého akordu a snadno se pak na ně dotážeme.

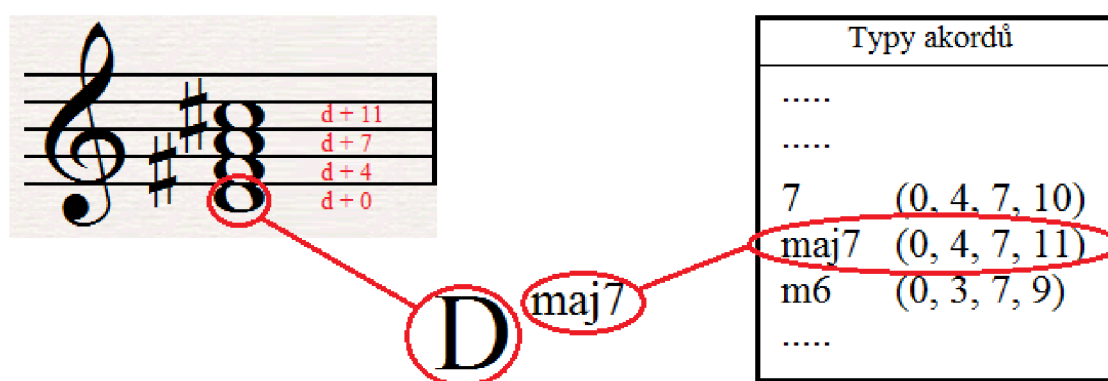
### 5.1.4 Re prezentace akordu a jeho tenzí

Spolu s melodií, kterou už umíme reprezentovat, máme na vstupu také akordy, které musíme nějak uložit. Tenze akordu budeme potřebovat pro další průběh aranžování.

#### Re prezentace akordu

Z kapitoly 1.3 víme, že akord je tvořen několika tóny, přičemž nejspodnější tón je základní neboli basový. Výběr tónů, z kterých se akord skládá, se řídí typem akordu. Přesněji, typ akordu určí tóny relativně k základnímu tónu. Máme-li pak stejný typ akordu ale od jiného základního tónu, snadno jsme schopni odvodit konkrétní tóny.

Typ akordu a jeho základní tón je jednoznačně určen příslušnou akordovou značkou a právě ve formě akordových značek budou vstupní akordy zapsány. Při načítání akordů ze vstupní melodie musíme tedy umět dekódovat akordové značky a rozpoznat základní tón a typ akordu. To znázorňuje obrázek 19.



Obrázek 19: Re prezentace akordu.

Na obrázku 19 je také vidět, jakým způsobem budou v paměti uloženy typy akordů. Tóny daného typu akordu jsou vztaženy relativně k základnímu tónu. Navíc nám stačí ukládat pouze výšky

tónů, jinými slovy přirozená čísla, u nichž hodnotě 1 odpovídá jeden půltón. Jedná se tedy o intervaly (*interval* viz kapitola 1.2) udávané v počtu půltónů od tónu základního.

Jako základní tón použijeme hodnotu 0, jelikož ho ještě neznáme konkrétně. Nulová výška tónu odpovídá v MIDI standardu tónu {c}. Snadno pak při dotazu na nějaký akord dopočítáme všechny jeho tóny. Na předchozím obrázku 19 je také uveden příklad takového výpočtu.

## Reprezentace tenzí

Víme, že tenze jsou další přidávané tóny k akordu, který je jimi harmonicky rozšířen. Jsou v jazzu hojně využívané. Jedná se vlastně o další intervaly vztažené od základního tónu akordu.

Jedná-li se tedy jakoby o další tóny akordu, můžou být tenze reprezentovány naprosto stejně. Ke každému typu akordu bude přiřazena další množina hodnot, představující vzdálenosti jednotlivých tenzí od základního tónu v počtu půltónů.

Jelikož se tenze staví nad původní akord, jsou od základního tónu vzdálené většinou o více než o oktávu (dvanáct půltónů) a nabývají hodnot větších než 12. Z hlediska harmonické funkce akordu rozšířeného tenzemi však není podstatné, v jaké oktávě budou tóny posazeny. Umístíme je proto všechny také do rozsahu 0 – 11.

Na obrázku 20 je obdobný příklad jako na předchozím. Tentokrát chceme zjistit, jak daný akord může být rozšířen o své tenze.

Tenze akordů	
.....	
.....	
7	(1, 2, 9)
maj7	(2, 9)
m6	(2, 11)
.....	

Obrázek 20: Reprezentace tenzí akordu. Ukázka akordu s tenzemi.

V průběhu procesu aranžování metodou blokové harmonizace budeme často potřebovat vědět o tónu melodie, zda se jedná o akordický tón, čili zda je obsažen v příslušném akordu, nebo zda se jedná o jeho případnou tenzi.

## 5.2 Schéma aranžéru

Algoritmus celého automatického aranžování by mohl vypadat nějak takto:

- 1) Načti melodii s akordy.
- 2) Vytvoř rytmický základ basového partu.
- 3) Generuj požadovaný počet řešení basového partu pomocí Gecode.
- 4) Označ cílové/přechodové noty pomocí Gecode. Bude též více řešení.
- 5) Harmonizuj cílové noty.
- 6) Reharmonizuj přechodové noty.
- 7) Zapiš jednotlivá řešení výsledného aranžmá

V následujících podkapitolách budou popsány jednotlivé části algoritmu.

## 5.3 Vytvoření basového partu

Rozhodl jsem se vytvořit basový part ke svému pětihlasu, a to především proto, že výsledné aranžmá bude působit mnohem konkrétnějším a jasnějším dojmem. To pak platí dvojnásob u jazzové hudby, kde je harmonie někdy natolik složitá, že bez basu zní nejistě či ochuzeně. Dovolím si zde použít výstižnou, ač trochu hovorovou frázi, používanou v hudební sféře, která říká, že basa takzvaně „tvrdí muziku“.

### 5.3.1 Rytmický základ basového partu

Při aranžování pětihlasu pomocí blokové harmonizace není nutné se zabývat rytmem, protože všechny hlasy mají rytmus stejný. V případě basu tomu tak už není. Je tedy třeba nejprve navrhnout rytmus basu.

Bas má rytmus svůj a často spíše opačný než melodie. Pokud je v melodii delší nota, bas se v daném místě může pohybovat rychleji a naopak, pokud je v melodii rychlý pohyb aneb kratší noty, bas může mít pomalé změny. Ve výsledku to pak působí kontrastně, protože pokud je něco zajímavého v melodii, bas na sebe příliš neupozorňuje a naopak, když je v melodii nějaký předěl mezi frázemi či pauza, bas na sebe upoutá pozornost zajímavým postupem.

V kapitole 2.2 jsem uvedl dva styly aranžování basu používané v jazzové hudbě. Rozdíl mezi nimi tkví především v rytmu.

## Tvorba rytmu pro bas 2 *in a bar*

Základem pro tento typ basu jsou půlové noty. Nejsnazší by tedy bylo naskládat půlové noty za sebe v celkové délce rovnající se délce vstupní melodie. My ale můžeme bas udělat trochu zajímavější tím, že místo některých půlových not použijeme více kratších not. Jedná se například o místa, kde je v melodii pauza nebo dlouhý tón.

Pak jsou ale také situace, kdy bychom také měli půlovou notu v basu rozbit na kratší noty. Jsou to místa, kde se v melodii změní akord. Pokud během trvání basové půlové noty dojde ke změně akordu, měli bychom ji rozdělit. Změny akordů, tedy harmonické změny ve skladbě by měl bas reflektovat jako první.

Pro tento typ basu budeme tedy používat i noty čtvrté. Mohli bychom samozřejmě užít i noty osminové či ještě kratší. Pro zpestření a reflektování změn v melodickém a akordovém postupu skladby však čtvrté noty postačí, navíc bychom si pak zbytečně komplikovali systém určování výšek basových tónů.

The image shows a musical score for a bass line in 2 in a bar style. It consists of two systems of music. The first system has five measures with chords C, Bm7, E7, Am, and Am7. The second system has seven measures with chords Fmaj7, G7, C, Bm7b5, Am7, D7, F, and C. The bass line uses a mix of half notes and quarter notes to reflect the changes in the melody and harmony.

Obrázek 21: Ukázka použití půlových a čtvrt'ových not v basu 2 *in a bar*.

Můžeme tedy stanovit několik zásad, dle kterých se budeme při tvorbě rytmu basového partu řídit. Na obrázku 21 vidíme příklad návrhu rytmu takového basového partu, řízeného podle změn v melodii a v akordech.

Zásady tvorby rytmu basu 2 *in a bar*:

- Používej především půlové noty.
- Místo půlové noty použij dvě čtvrté noty, pokud dojde ke změně akordu během první poloviny této doby.

- Místo lichých půlových not použij dvě čtvrtě noty, pokud je po celou tuto dobu v melodii dlouhý tón či pauza.

### Tvorba rytmu pro *Walking bass*

Tento typ basu používá především noty čtvrtě. I tady bychom mohli vymyslet několik zásad pro použití kratších not, než jsou čtvrtě, v některých místech. Tyto zásady by se dost podobaly těm, které jsme použili pro typ basu *2 in bar*. Ve své aplikaci jsem však rytmus basu postavil jen a pouze na čtvrtě notách. Vycházel jsem také z předpokladu, že tento typ basu bývá častěji využíván pro rychlejší tempa, je proto o důvod méně používat not osminových a kratších. Do námětů na vylepšování aranžéru do budoucna bych však tuto možnost rozhodně zařadil.

The image shows two systems of musical notation for a walking bass line. Each system consists of a treble clef staff and a bass clef staff. The first system has five measures with chords: C, Bm7, E7, Am, and Am7. The second system has seven measures with chords: Fmaj7, G7, C, Bm7b5, Am7, D7, F, and C. The bass line in both systems is composed of quarter notes, demonstrating the 'walking bass' style.

Obrázek 22: Ukázka použití čtvrtě not ve stylu *walking bass*.

Po celé délce melodie tedy naplníme basový part čtvrtě notami bez ohledu na to, co je v melodii či jaké jsou změny akordů (viz obr. 22).

### 5.3.2 Pravidla pro generování výšek tónů basového partu a stanovení CSP

V předchozí kapitole jsme navrhli postup, jak vytvořit rytmus basového partu a nyní je na řadě podle pravidel aranžování basu v jazzové hudbě určit výšky jednotlivých tónů. K tomu nám dobře poslouží metoda řešení úloh s omezujícími podmínkami a knihovna Gecode.

Výšky tónů basového partu budou proměnné, jimž stanovíme doménu podle toho, jakých hodnot mohou nabývat v závislosti na pravidlech aranžování basu v jazzové hudbě.

## Společné CSP pro styly *2 in a bar* i *walking bass*

Dosud jsem rozlišoval dva typy basu, které chci ve svém aranžéru implementovat. Následující definice problému CSP a návrh pravidel se týká především varianty basu *2 in a bar*. Pro variantu *walking bass* by mělo být definováno CSP a pravidla zvlášť. Z časových důvodů však toto nebylo implementováno. Proto zmíním návrh CSP pouze pro variantu *2 in a bar*.

Ve výsledné aplikaci však varianta *walking bass* bude fungovat. Použijeme pro její vytvoření stejné CSP jako pro variantu *2 in a bar*. Bas složený ze čtvrtových not je tak správně vygenerován a zcela korektní a funkční. Je pouze ochuzen o širší možnosti výběru tónů, které by ho učinily věrnějším.

### Stanovení CSP

- Konečná množina proměnných obsahující všechny tóny basového partu.
- Pro každou proměnnou stanovíme doménu, která bude obsahovat následující intervaly z akordu příslušejícího danému tónu v několika oktávách. {toniky, tercie, kvinty}.
- Konečná množina omezujících podmínek:
  - Vzdálenost mezi sousedními tóny nesmí být větší než jedna oktáva.
  - Dvě sousední čtvrtové noty nesmí mít stejnou výšku.
  - První tón skladby je tonika.
  - Poslední tón skladby je tonika.
  - Při změně akordu o kvartu výše či o kvintu níže použij postup:
    - tonika => tonika
    - tercie => tonika
    - kvinta => kvinta
  - Při jiné akordické změně použij postup tonika => tonika.

Uvedené omezující podmínky CSP reflektují pravidla pro aranžování basu typu *2 in a bar* uvedená v kapitole 2.2.1. Použité termíny *tonika*, *tercie*, *kvarta* a *kvinta* jsou intervaly, které byly vysvětleny v kapitole 1.2.

Intervaly jsou vztaženy vždy k akordu, který danému tónu přísluší. Je-li tedy například akordový postup *C dur – F dur*, jedná se o změnu akordu o kvartu výše (případně o kvintu níže) a vybereme-li si postup *tercie => tonika*, budou basové tóny {e} a {f}.

### Taktika generování řešení

Uživatel si v konfiguračním souboru nastaví, kolik řešení si přeje vyhledat maximálně. Pokud bychom ale generovali několik řešení po sobě, jednotlivá řešení by se lišila jen nepatrně. Dovolím si



předpoklad, že si uživatel bude přát mít řešení co nejvíce odlišná, aby tím získal co nejvíce nápadů na basový part.

Mohli bychom to vyřešit tak, že bychom některá řešení zahazovali a dobrali se tak podstatnějších změn v nich. Ovšem generování zbytečných řešení by bylo časově poměrně náročné. Doba vyhledání jednoho řešení roste nelineárně s délkou vstupní melodie.

Gecode ale poskytuje možnost nastavení prohlédávacího algoritmu přidáním tzv. *brancherů*. Ovlivníme tím pořadí vybírání proměnných pro přiřazování hodnot a také pořadí těchto přiřazovaných hodnot.

Nastavíme tedy několik *brancherů* a pro každý z nich vygenerujeme pouze první řešení. Vhodnou volbou prohlédávacích taktik docílíme slušné variability vyhledaných řešení.

Použil jsem různé kombinace následujících taktik:

- Upřednostnění výběru proměnných s nejméně omezeními.
- Upřednostnění výběru proměnných s nejvíce omezeními.
- Náhodný výběr proměnných.
- Bez upřednostnění výběru proměnných.
- Upřednostnění výběru nejnižších hodnot.
- Upřednostnění výběru nejvyšších hodnot.
- Upřednostnění výběru hodnot rovnajících se mediánu.
- Náhodný výběr hodnot.

Pro prohlédávání stavového prostoru řešení jsem zvolil algoritmus *Depth first search*.

## 5.4 Rozdělování na cílové a přechodové noty a stanovení CSP

V kapitole 2.1.1 jsem zmínil několik pravidel, podle kterých by se mohlo řídit určování, zda se jedná o cílovou či přechodovou notu melodie. V praxi se však aranžér řídí kromě několika všeobecných pravidel především svým citem. Je proto třeba vymyslet nějaký systém, který bude jeho postup co nejvíce připomínat. Pokud se nám navíc podaří stanovit několik přesně definovaných podmínek a omezení, můžeme využít metodu řešení úloh s omezujícími podmínkami a vygenerovat tak více řešení.

Zde je několik pravidel, dle kterých by mělo rozhodování fungovat. Jejich smysl je takový, že čím více je jich splněno, tím spíše se bude jednat o **cílovou** notu.



Základní pravidla vypadají takto:

- Jedná se o akordický tón.
- Jedná se o tenzi akordu.
- Jedná se o tón střední délky v rámci celé melodie.
- Jedná se o tón větší délky v rámci celé melodie.
- Nejedná se o poslední tón aktuálního akordu.

První dvě pravidla se navzájem vylučují. Pokud se jedná o akordický tón, již nemůže jít o tenzi akordu a naopak. Nemusí však jít ani o jedno. Stejně tak je na tom dvojice pravidel týkající se délky tónu. Pravidlo o délce tónu je rozděleno do dvou samostatných proto, že u délek třeba dvojnásobně větších, než je průměrná délka tónu, je pravděpodobnost, že se jedná o cílovou notu, již poměrně velká.

Zmíněná pravidla ve dvojicích mohou tedy platit zvlášť, ale nemohou platit obě najednou. Nemusí ale také platit ani jedno z nich.

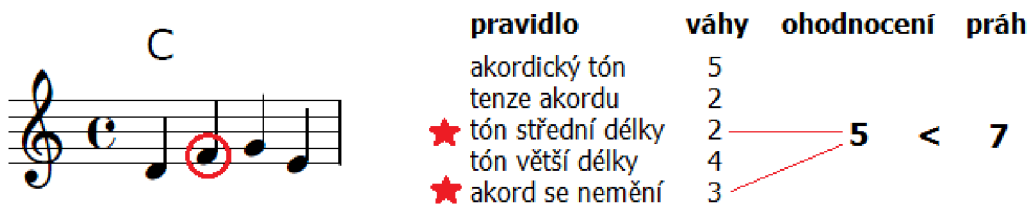
### 5.4.1 Stanovení váhy pravidel a ohodnocení potenciální cílové noty

Aby se moje aplikace alespoň trochu přiblížila reálnému postupu aranžéra, navrhl jsem systém ohodnocování. Každému z pravidel přiřkneme určitou váhu, podle které budeme rozhodovat, jak moc má dané pravidlo ovlivnit výsledek. Váha každého pravidla bude nastavitelná uživatelem, který vhodně zváží, která pravidla upřednostní a která naopak vyzdvihne. To bude činit v závislosti na konkrétním příkladu melodie a na jeho představě o způsobu harmonizace.

Budeme zkoumat vstupní melodii tón po tónu. Postupně budeme procházet jednotlivá pravidla a zjišťovat, jestli jsou pro aktuální tón pravdivá. U toho, které bude platit, si zapamatujeme jeho váhu. Sečteme váhy všech platných pravidel a součet porovnáme s **prahem**, který bude uživatel nastavovat též.

Skutečnost, že se některá pravidla navzájem vylučují, tedy, že nemohou platit najednou, nám nijak nevádí. Je to důležité pouze a především pro uživatele, jenž bude váhy pravidlům nastavovat a který musí vhodně nastavit práh, dle něhož se nota označí za přechodovou či cílovou.

Na následujícím obrázku 23 můžeme vidět názorně postup ohodnocení tónu melodie postupnou aplikací pravidel.



Obrázek 23: Ohodnocení tónu pro rozdělení na cílové a přechodové.

Zkoumáme tón {f} fragmentu melodie, kterému přísluší akord *C dur*. Pro akord tohoto typu máme někde v paměti v relativní podobě uloženy jeho tóny a tenze. Vezmeme základní tón {c} a odvodíme od něho konkrétní tóny {c, e, g} a tenze {d, a, b} akordu *C major*. Zjistíme, že zkoumaný tón {f} není obsažen v akordu *C dur* a není ani jeho tenzí.

Vypočítáme průměrnou délku tónu melodie a porovnáme ji s délkou tónu {f}. Zde vhodným způsobem ještě rozdělíme situaci, kdy je délka tónu větší než průměrná délka, a budeme odlišovat tóny přibližně průměrné délky a tóny podstatně větší délky. Náš tón {f} je v rámci fragmentu melodie středně dlouhý.

Jako poslední nás zajímá, jestli se nejedná o poslední tón fragmentu melodie, které je přiřazen aktuální akord. Pokud by následujícímu tónu {g} už příslušel jiný akord než *C dur*, měli bychom o důvod více pokládat notu {f} spíše za přechodovou. Ovšem, protože tóny {f} a {g} spadají oba pod stejný akord, můžeme poslední pravidlo prohlásit za platné a použít jeho váhu.

Nyní tedy máme sumu, kterou můžeme porovnat s prahem. V našem případě je celkové ohodnocení nižší než uživatelem stanovený práh. Nemůžeme proto označit notu za cílovou.

## 5.4.2 Sporné noty a řešení využitím programováním s omezujícími podmínkami

V předchozí kapitole bylo předvedeno, jak ohodnotíme tón melodie. Bylo by nyní snadné porovnat jej s uživatelem definovaným prahem a říci, že noty se stejným či vyšším ohodnocením budou cílové a noty s nižším ohodnocením přechodové. Bylo by to však příliš unáhlené řešení.

U některých tónů je rozhodování velice sporné a měli by u nich být pro výsledné řešení ponechány obě varianty. Jedná se o tóny s ohodnocením jen o málo nižším, než je stanovený práh. Sporné tóny bývají například akordické tóny krátkého trvání či neakordické tóny delšího trvání. Pomocí programování s omezujícími podmínkami můžeme tento problém elegantně vyřešit.

Vezmeme práh a stanovíme 3 skupiny:

1. Skupina s ohodnocením stejným nebo vyšším než práh – cílové noty.
2. Skupina s ohodnocením o méně než 1/3 nižším než práh – **sporné noty**.
3. Skupina s ohodnocením o více než 1/3 nižším než práh – přechodové noty.

### Stanovení CSP

- Konečná množina proměnných obsahující všechny tóny melodie.
- Pro všechny proměnné platí stejná doména {přechodová nota, cílová nota}.
- Konečná množina omezujících podmínek:
  - Patří-li tón do 1. skupiny, označ ho za cílový.
  - Patří-li tón do 2. skupiny a nejedná se o akordický tón, označ ho za přechodový.
  - Patří-li tón do 2. skupiny a jedná se o akordický tón, pak ho označ za cílový, **právě když** byl předchozí tón označen za přechodový:

$$(\text{tón}[i - 1] = \text{přechodový}) \quad \Leftrightarrow \quad (\text{tón}[i] = \text{cílový})$$

- Patří-li tón do 3. skupiny a jedná se o akordický tón, pak ho označ za cílový, **jestliže** byl předchozí tón označen za přechodový.

$$(\text{tón}[i - 1] = \text{přechodový}) \quad \Rightarrow \quad (\text{tón}[i] = \text{cílový})$$

- Jde-li o poslední tón melodie, označ ho za cílový.

### Taktika generování řešení

Takto stanovené CSP vyřešíme některým z prohledávacích algoritmů knihovny Gecode. Vybral jsem algoritmus *Depth first search*. Kromě prohledávacího algoritmu lze zvolit způsob vybírání proměnných a hodnot z jejich domén. Nastavil jsem vybírání proměnných od těch s nejmenší doménou a vybírání hodnot od té nejmenší v doméně.

Vygenerovaných řešení však může být různý počet v závislosti na délce vstupní melodie. Gecode pak generuje všechny možné kombinace označení not za cílové či přechodové. Nemá smysl pro další zpracování používat všechna tato řešení. Bude stačit, když vybereme jen ta, jež se od sebe více liší rozdělením na přechodové a cílové noty.

Rozmanitosti řešení docílíme tak, že vybereme první a poslední řešení, a poté budeme vybírat vzorky po stejných úsecích z domény řešení. Tyto úseky budou vypočteny podle toho, kolik řešení si uživatel přeje vygenerovat.

Všechna tato řešení budou použita pro další průběh zpracování. První část blokové harmonizace tedy máme za sebou a nyní můžeme každé řešení použít jako základ pro tvorbu dalších hlasů k melodii.

## **5.5 Harmonizace cílových a reharmonizace přechodových not**

V tomto stadiu předpokládáme, že jsou již tóny melodie rozděleny na cílové a přechodové. Jinými slovy, máme označené tóny, které musí být reharmonizovány jiným akordem, než který je jim určen v melodii.

Ze zmíněných technik blokové harmonizace budeme používat pouze čtyřhlasou úzkou harmonii doplněnou do pětilhasu melodií zapsanou o oktávu níže.

### **5.5.1 Harmonizace tónu akordem**

Máme-li akord, kterým má být tón melodie zharmonizován, nezbyvá než dle pravidel čtyřhlasé úzké harmonie doplnit 3 další akordické tóny a jako pátý tón o oktávu snížený výchozí tón melodie.

V kapitole 4.1 byl navrhnout způsob uložení akordů v paměti. Nyní potřebujeme zjistit konkrétní tóny daného akordu.

Při skládání tónů v úzké harmonii se budeme držet několika zásad:

- Akordy skládající se pouze ze třech tónů je třeba rozšířit o čtvrtý tón. Jinými slovy, z kvintakordů udělat alespoň septakordy.
- Občas doplnit akordy o jejich tenze.
- Pohlídat, aby mezi prvním a druhým hlasem byla větší vzdálenost než jeden půltón.
- V případě potřeby lze vypustit některé tóny akordu, ovšem jen pokud by se nejednalo o tón melodie. Vypouštíme v tomto pořadí:
  - 1) Toniku, pokud není v melodii.
  - 2) Tón vzdálen jen o jeden půltón od toniky, pokud ta je v melodii.
  - 3) Kvintu, pokud není v melodii a není snižená či zvýšená (tedy pokud je čistá, čili má vzdálenost sedm půltónů od toniky).

## Celý algoritmus harmonizace tónu melodie přiřazeným akordem

1. Načti tóny příslušného akordu.
2. Pokud tón melodie není v akordu obsažen:
  1. A pokud tam není ani o jeden půltón nižší tón, potom tón melodie přidej.
  2. A pokud je tam o jeden půltón nižší tón, vlož tón melodie místo něho.
3. Jsou-li nyní v akordu pouze 3 tóny, přidej septimu. Pokud je v melodii tonika, přidej sextu.
4. Pokud je v melodii kvinta, přidej tenzi (nonu) tak, že toniku nahrad' sekundou.
5. Máme-li nyní více jak 4 tóny, provedeme následující:
  1. Pokud není v melodii tonika, vypušt' ji.
  2. Pokud tonika v melodii je, vypušt' případný tón vzdálený od toniky o jeden půltón.
  3. Pokud stále máme více jak čtyři tóny, vypušt' čistou kvintu.
  4. Není-li čistá kvinta, vypušt' tón nad tonikou.
6. Vyskládej tóny těsně pod sebe počínajíc tónem melodie.

### 5.5.2 Reharmonizace jiným akordem

V kapitole 2.1.4 jsem uvedl čtyři způsoby reharmonizace přechodové noty. Ve výsledku však používám pouze dva, a to z následujících důvodů.

Pro diatonickou reharmonizaci bychom museli mít v paměti uložených mnoho druhů stupnic a k nim vzájemné souvislosti s akordy a tóninou skladby. Navíc je to technika střídměji používaná.

Co se týče chromatické reharmonizace, je to vlastně reharmonizace paralelním akordem. Rozdíl je pouze v tom, že krok není libovolná vzdálenost, ale přesně jeden půltón. Můžeme říci, že reharmonizace paralelním akordem chromatickou reharmonizaci zahrnuje.

Zbyly nám tedy dvě techniky, které jsem implementoval. Jedná se o reharmonizaci paralelním akordem a dominantním septakordem.

Není však lehké navrhnout strategii rozhodování ani mezi pouhými dvěma metodami reharmonizace. V praxi aranžér volí dle své zkušenosti a hudební představy. Měli bychom tedy buď vygenerovat více řešení, nebo zařídit zpětnou vazbu od uživatele, který by zvolil dle svého uvážení.

Pro naše účely jsem však volbu stanovil na pevnou. Vycházel jsem z jednoduché úvahy. Zatímco paralelní akord můžeme použít vždy, dominantní septakord pouze v případě, že je v něm **tón melodie obsažen**. To se stává spíše zřídka než často, proto dáme dominantnímu septakordu přednost. Pokud ho bude možné použít, vykonáme tak. V ostatních případech budeme reharmonizovat paralelním akordem.

Nicméně, je zde ještě jedna podmínka, která musí být splněna při použití dominantního septakordu. **Následující tón** melodie musí být **cílový**.

### **Reharmonizace dominantním septakordem**

Zde předpokládáme, že následující tón melodie nebude reharmonizován, tedy, že je cílový.

Nejprve vezmeme akord následujícího tónu melodie a vypočítáme k němu dominantní septakord. To provedeme nejjednodušeji tak, že základní tón tohoto akordu navýšíme o sedm půltónů, případně snížíme o pět půltónů, na směru nezáleží.

Poté ověříme, zda aktuální tón melodie je v tomto akordu obsažen. Pokud ne, dominantní septakord zahodíme a použijeme paralelní akord (viz dále). Pokud však ano, máme k tónu melodie stanovený akord, kterým má být harmonizován, a můžeme využít algoritmu harmonizace cílových not uvedeného v kapitole 4.5.1.

Na tomto místě se jen v rychlosti zmíním o reharmonizaci zmenšeným septakordem, která je jednou z nejpoužívanějších. Ta totiž vychází z dominantního septakordu rozšířeného o tenzi malé nony (třináct či jeden půltón nad tonikou) a ochuzeného o toniku. Bývá použit v případech, kdy v melodii není tonika.

### **Reharmonizace paralelním akordem**

Provede jednoduše tak, že zjistíme výškovou vzdálenost mezi přechodovým tónem a následujícím tónem melodie. Do vnitřních hlasů zapíšeme tóny následujícího cílového akordu snížené či zvýšené o zjištěnou hodnotu.

## **5.6 Uživatelské rozhraní aplikace**

Moje aplikace bude mít velice jednoduché textové rozhraní a pro načtení vstupní melodie a pro uložení hotového aranžmá použije externí nástroj pro sazbu not. Vytvoření mého vlastního grafického rozhraní pro sazbu not, a hlavně takového, aby poskytovalo možnosti, které potřebujeme, by mě poměrně dost zaměstnalo a zbytečně odvedlo pozornost od hlavního cíle mé práce. Navíc by svým rozsahem možná vydalo na další takovou práci.

### **5.6.1 Využití notačního nástroje LilyPond a návrh vstupu a výstupu aplikace**

LilyPond je komplexní nástroj pro sazbu not. Umožňuje zapisovat noty ve vysoké kvalitě. Jedná se vlastně svým způsobem o notovou tiskárnu a je součástí GNU Project.

Pro zápis hudby používá svůj programovací jazyk, jehož sémantika vychází z hudební teorie. I když psaní not v něm není būhvíjak svižné, po osvojení jednoduché a intuitivní syntaxe v něm lze zapsat téměř cokoliv.

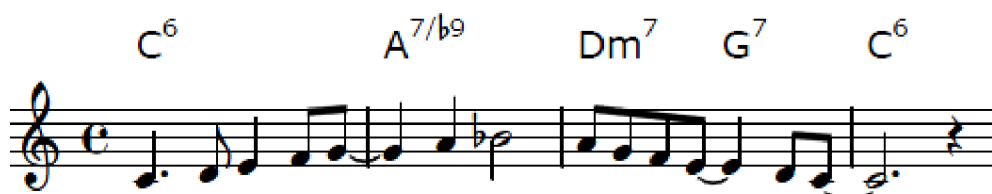
LilyPond umí tisknout noty mimo další formáty především do PDF nebo PostScript, které snadno vytiskneme. Jednou z důležitých vlastností LilyPondu je možnost výstupu do MIDI, což výrazně usnadňuje kontrolu vysázené skladby. K vizuální složce se tak přidává i poslechová. Je ale třeba říct, že oproti sazbě má konverze do MIDI jistá omezení, například jsou ignorovány značky zeslabení, zesílení či staccato, ale i tak je to velmi šikovná pomůcka.

Právě jazyk LilyPondu jsem si zvolil za vstupní i výstupní formu hudebních dat pro moji aplikaci. Uživatel v něm запиše vstupní melodii, kterou si může pro kontrolu vytisknout do PDF, nebo vygenerovat MIDI soubor, který si přehraje. LilyPond umí také sázet k notám akordové značky, což také potřebujeme. Uživatel tedy kromě melodie ještě vloží příslušný sled akordů. Při vygenerování do MIDI se akordy dokonce transformují do zvukové podoby. Jednoduchý příklad zápisu melodie s akordy můžeme vidět na obr. 24.

```
\score
{
  <<
    \chords
    {
      c1:6 a:7.9- d2:m7 g:7 c1:6
    }

    \new Staff
    {
      \clef treble
      \key c \major
      \time 4/4

      c'4. d'8 e'4 f'8 g'8 ~g'4 a'4 bes'2
      a'8 g'8 f'8 e'8 ~e'4 d'8 c'8 ~c'2. r4
    }
  >>
}
```



Obrázek 24: Ukázka sazby not v LilyPondu.

Můj automatický aranžér bude postaven tak, aby byl schopen ze souboru pro LilyPond načíst melodii s akordy a také všechny potřebné informace o skladbě (tempo, tónina, počet dob v taktu atd.). Po vytvoření aranžmá budou tyto informace potřebné pro vytvoření souboru pro LilyPond, kde budou kromě melodie vysázené další čtyři hlasy a basový part. Vytvořený soubor s hotovým aranžmá si uživatel bude moci stejným způsobem převést do PDF či MIDI.

Možnost výstupu do MIDI má kromě možnosti poslechu skladby ještě jednu podstatnou výhodu. Tento standard podporují nejpoužívanější notové editory, jmenovitě např. Finale či Sibelius. Uživatel, bude-li chtít výstup mého aranžéru dále efektivně zpracovat, si prostřednictvím MIDI souboru může přenést aranžmá do svého oblíbeného notačního nástroje.

## **5.6.2 Návrh hromadného aranžování několika vstupních melodií**

Jak bylo již několikrát zmíněno, výsledná aplikace bude schopna posloužit jako obecný pomocník při aranžování skladeb. Přestože je zaměřena na saxofonovou sekci, výsledný vícehlas může být využit pro jakékoli jiné nástrojové složení. Hudebník začátečník si tak může zaranžovat jakoukoli skladbu, pokročilejšímu aranžérovi poslouží jako inspirace. Uživatel se pomocí aranžéru také může učit blokové harmonizaci.

Tyto důvody a mnohé další hovoří pro to, aby aplikace uměla hromadně zpracovávat více souborů najednou. Ponecháme uživateli možnost volby, zda uvede seznam souborů se vstupními melodiemi, které chce zaranžovat, nebo adresář, v němž se zpracují všechny nalezené melodie. Vše nastaví v konfiguračním souboru aplikace.

Aplikace pak pro každou ze vstupních melodií provede celý proces blokové harmonizace a vytvoří basový part. Jednotlivá řešení každé z těchto melodií vhodně pojmenuje a uloží do uživatelem zvoleného adresáře.

Pro možnost hromadného zpracování hovoří také fakt, že efektivně a poměrně rychle jsou zaranžovány jen krátké melodie. Delší melodie celkem dost zaměstnají prohledávací algoritmus Gecodu, jelikož ten je zatížen více proměnnými a více omezujícími podmínkami. Delší melodie je proto vhodné rozdělit do více souborů a ty pak hromadně zaranžovat.

Tento problém by v dalším pokračování na této práci mohl být řešen automatickým rozdělením melodie na menší části a jejich postupným aranžováním. Bylo by ovšem nutné nalézt vhodný bod, kde by bylo možno melodii rozdělit bez újmy na kontinuitě aktuální hudební fráze.



### 5.6.3 Konfigurační soubor

K zadání vstupních parametrů a k řízení celého procesu aranžování jsem navrhnul jednoduchý konfigurační soubor. V kapitolách výše jsem občas zmínil jisté atributy, které byly přizpůsobeny k tomu, aby mohl uživatel nastavováním jejich hodnot více či méně ovlivňovat průběh automatického aranžování.

V konfiguračním souboru lze nastavit:

- Zda je vstupem seznam souborů či adresář se soubory.
- Seznam souborů se vstupními melodiemi.
- Adresář, v němž budou zpracovány všechny soubory s melodiemi.
- Adresář pro uložení souborů s jednotlivými řešeními.
- Basový part – má-li se generovat a případně jaký typ (*2 in a bar/walking bass*).
- Ovlivnění výběru cílových not ohodnocením různých stavů:
  - Jedná se o akordický tón.
  - Jedná se o tenzi akordu.
  - Jedná se o tón střední délky v rámci celé melodie.
  - Jedná se o tón větší délky v rámci celé melodie.
  - Nejedná se o poslední tón aktuálního akordu.
- Minimální práh celkového hodnocení, kdy se ještě jedná o cílovou notu.
- Maximální počet generovaných řešení.
- Zda mají být ve výsledném aranžmá akordové značky.
- Zda mají být označeny přechodové noty, které byli reharmonizovány.
- Zda mají být party transponovány pro příslušné nástroje.
- Zda má aplikace po skončení čekat na ENTER.

Podrobný popis syntaxe konfiguračního souboru je uveden v příloze 2.

## 6 Závěr

Cílem mé práce bylo načíst vstupní melodii, ke které byly zároveň zadány i akordy, tuto zaranžovat do pěti hlasů a k nim přidat ještě basový part. Dále měla být použita technika programování s omezujícími podmínkami.

Byl vytvořen prototyp aplikace, který načte melodii spolu s akordy ze souboru. Vstupní soubory mají textovou podobu a jsou zároveň zdrojovými soubory pro notosazební program Lilypond.

Pomocí CSP byly noty melodie rozděleny na cílové a přechodové a následně harmonizovány čtyřhlasou úzkou harmonií. Přechodové noty byly reharmonizovány dominantním septakordem nebo paralelním akordem. Vzniklo 5 hlasů, které byly uloženy do 5 notových osnov opět do souboru pro Lilypond.

Aplikace byla vytvořena pod platformou Windows, nebyl by ale velký problém ji překlomit do Unixového systému. Knihovna Gecode pro řešení CSP je vyvíjena pro obě platformy a externí nástroj na sazbu not LilyPond taktéž.

### 6.1 Zhodnocení dosažených výsledků

Vzniklá aplikace je schopna celkem dobře aranžovat a opravdu může celkem dobře sloužit jako pomůcka pro hudebního aranžéra. Ač je postavena pro harmonizaci pro konkrétní hudební těleso, není problém vzniklé harmonické hlasy využít i v jiných hudebních formacích.

Byly testovány různé melodie. Nejlepší výsledky byly dosaženy pro typicky jazzové melodie, což byl vlastně cíl práce. Při aranžování melodie jiného hudebního žánru musíme počítat s tím, že tato melodie dostane trochu nečekaný jazzový „kabát“. Je pak již otázkou vkusu posluchače, zda to byl tah správným směrem.

System ohodnocování různých vlastností not pro označení za cílové či přechodové umožňuje do jisté míry ovlivnit nevhodné zaranžování některých not. Ne však vždy je tento systém dostatečně schopný. Zkušený uživatel však může vhodně změněnými akordy v melodii docílit zharmonizování některých not jiným způsobem.

Ke zvýšení použitelnosti přispívá hromadné zpracování několika melodií. Uživatel si může vytvořit několik návrhů či fragmentů melodie a všechny jednoduše aranžérem zpracovat.

Možnost výsledné aranžmá transponovat z C hlasu do ladění příslušných nástrojů považuji také za užitečnou. Ihned totiž výsledné aranžmá můžeme vyzkoušet a přehrát v dané nástrojové sestavě, v našem případě v saxofonové sekci s případným basovým nástrojem.

## 6.2 Možnosti pokračování práce

Na této práci by bylo zcela jistě možno pokračovat dále a námětů by rozhodně nebylo málo. Průběžně v textu této práce jsem se občas zmínil o dalších schopnostech, kterými by mohl být aranžér v budoucnu obdařen. V následujících několika odstavcích se je pokusím shrnout, případně navrhnout nějaké další.

Patrně největší přínos by mělo rozšíření o zpětnou vazbu ze strany uživatele. V průběhu aranžování nastává několik momentů, kdy se program sám snaží rozhodnout, jaké techniky použít a snaží se svá rozhodnutí založit na nejrůznějších pravidlech. Jak jsem již však několikrát zmínil, nejlepší pravidlo je okamžitý záměr aranžéra. Jen on může posoudit, zda se mu výsledky líbí či zda zapadají do jeho představ o výsledném aranžmá. Uživatel by tedy měl mít šanci přímo nebo i zpětně ovlivňovat průběh aranžování.

Další návrh na rozšíření vychází ze široké škály aranžovacích technik v rámci blokové harmonizace. Kromě využití techniky čtyřhlasé úzké harmonie bychom mohli vyjmenovat řadu dalších a neméně používaných. Kromě otevřené harmonie, která byla také zmíněna, jsou to například *klastry*, v nichž se na sebe kupí akordické tóny a všechny možné tenze co nejtěsněji a vznikají tak zajímavé zvukové efekty. Dále pak třeba lomené akordy atd. Můžeme ale opustit i blokovou harmonizaci a tvořit ve vnitřních hlasech i jiný rytmus. Otevrou se nám tak další salvy možností.

K saxofonové sekci můžeme postupně přidat trumpetovou a pozounovou, pro něž se používají stejné techniky aranžování. Sekce mohou hrát najednou či střídavě. Pomocí CSP bychom mohli vygenerovat kromě basu ještě party pro klavír a bicí. Tím bychom získali aranžmá pro celý big band.

Co se týče vstupu a výstupu aranžéra, nabízí se využití formátu MuxicXML. Ten umožňuje notový zápis do souboru XML. Obrovská výhoda tohoto formátu je taková, že je v poslední době hojně využíván pro přenos notových zápisů mezi známými komplexními notačními nástroji. Vstupní melodie i výstupní aranžmá by tak mohly být dále zpracovávány nástroji jako Sibelius či Finale.

# Literatura

- [1] Lowell Dick, Pullig Ken, edited by Michael Gold, *Arranging for Large Jazz Ensemble*, Berklee Press (2003).
- [2] Maňák Ondřej, *Automatická tvorba varhanní přede hry k církevním písním*.  
Diplomová práce (FIT VUT v Brně 2010).  
<https://www.fit.vutbr.cz/study/DP/rpfile.php?id=7881>
- [3] Taming the Saxophone, *Jazz Arranging: Four Part Block Voicing*  
<http://tamingthesaxophone.com/jazz-blockvoicing.html>
- [4] Anders Torsten, *Composing Music by Composing Rules*. (2007)  
<http://strasheela.sourceforge.net/documents/TorstenAnders-PhDThesis.pdf>
- [5] Schulte, Ch., Tack, G., Lagerkvist, M.: *Modelling and Programming with Gecode*. (2009)  
<http://www.gecode.org/doc-latest/MPG.pdf>
- [6] Barták Roman, *Programování s omezujícími podmínkami*. (2000)  
[http://dml.cz/bitstream/handle/10338.dmlcz/141039/PokrokyMFA\\_45-2000-3\\_4.pdf](http://dml.cz/bitstream/handle/10338.dmlcz/141039/PokrokyMFA_45-2000-3_4.pdf)
- [7] Berka, P. *Úvod do umělé inteligence* (2007).  
<http://sorry.vse.cz/~berka/docs/4iz229/s02-LISP-4p.pdf>
- [8] Svoboda Milan, *Praktická jazzová harmonie*, skripta pro potřeby studentů Konzervatoře Jaroslava Ježka v Praze (2002).  
<http://www.milansvoboda.com/skripta.html>
- [9] Silná, I. *Nauka o harmonii*. Konzervatoř P. J. Vejvanovského Kroměříž (1998).
- [10] Zenkl, L. *ABC hudební nauky*. Praha, Editio Bärenreiter (2003).
- [11] Kofroň, J. *Učebnice harmonie*, Praha, Editio Bärenreiter (2006).
- [12] Zbořil, F. st., Zbořil, F. ml. *Základy umělé inteligence*, Studijní opora, VUT, FIT, Brno (2006).
- [13] Thomas, Pete. *Composition & Jazz theory*.

# Seznam příloh

Příloha 1: Manuál k ovládání aplikace.

Příloha 2: Syntaxe konfiguračního souboru.

Příloha 3: Ukázky výsledných aranžmá.

Příloha 4: DVD

# Příloha 1: Manuál k ovládání aplikace

Předpokládejme, že je aplikace zkompileována a sestavena do souboru `arranger.exe`.

Když nyní spustíme `arranger.exe`, aplikace se pokusí najít konfigurační soubor s implicitním názvem `config.txt` v aktuálním adresáři. Pokud konfigurační soubor nenalezne, spustí se s přednastavenými hodnotami.

Druhou možností je zvolit jiný konfigurační soubor, který si uživatel připraví. Ten může mít jakýkoliv název a na jeho umístění nezáleží. Při spouštění aplikace uvedeme název konfiguračního souboru s kompletní cestou k němu jako další parametr v příkazové řádce. V tomto případě nesmí být zadán žádný další parametr.

Příklad spuštění:

```
> arranger dir\my_config_file.txt
```

Konfigurační soubor musí být vždy textový. V případě nenalezení nebo neúspěchu otevření souboru v textovém režimu bude hledán implicitní konfigurační soubor.

## Příloha 2: Syntaxe konfiguračního souboru

Konfigurační soubor musí být **textový** a mít následující **syntaxi**:

```
input_type:           [FILES/DIR]
input_files:          (cesta1\soubor1 cesta2\soubor2 ...)
input_dir:            (cesta\adresar)
output_dir:           (cesta\adresar)
max_solution_count:  [1-5]
bass:                 [NO/2INBAR/WALKING]
show_chords:          [YES/NO]
show_passing_tones:  [YES/NO]
transpose:            [YES/NO]
targetEval_threshold: [0-(max_int)]
targetEval_IsChordTone: [0-(max_int)]
targetEval_IsChordTension: [0-(max_int)]
targetEval_IsMediumLongTone: [0-(max_int)]
targetEval_IsLongTone: [0-(max_int)]
targetEval_IsNotLastChordNote: [0-(max_int)]
pause_when_finish:   [YES/NO]
```

Všechny položky jsou case-sensitive, takže je třeba striktně dodržovat velká a malá písmena. Každý atribut se svou hodnotou musí být na novém řádku tak, jak je uvedeno výše. Pořadí atributů (řádků) je možno libovolně zaměnit. Pokud bude atribut uveden vícekrát, bude platit poslední jeho výskyt. Atributy nemusí být uvedeny všechny. Při absenci atributu bude nastavena implicitní hodnota a vypsáno upozornění na terminál. Tak se stane i v případě chybně zapsané hodnoty atributu.

### Atributy

input_type	- zda je vstup ze souborů nebo celý adresář souborů
input_files	- výčet vstupních souborů s kompletní cestou
input_dir	- vstupní adresář
output_dir	- výstupní adresář
max_solution_count	- počet požadovaných řešení
bass	- jaký typ basu má být vytvořen
show_chords	- zda mají být ve výsledném aranžmá zobrazeny akordy
show_passing_tones	- zda mají být zvýrazněny přechodové noty

transpose - zda má být provedena transpozice pro jednotlivé nástroje  
 pause\_when\_finish - zda má aplikace před skončením čekat na ENTER

/\* váhy pravidel pro určení cílových not \*/

targetEval\_IsChordTone - akordický tón  
 targetEval\_IsChordTension - tenze akordu  
 targetEval\_IsMediumLongTone - středně dlouhý tón  
 targetEval\_IsLongTone - dlouhý tón  
 targetEval\_IsNotLastChordNote - akord se nezmění  
 targetEval\_threshold - práh ohodnocení cílových not

## Sémantika

FILES/DIR - vstup ze souborů / vstup z celého adresáře  
 NO/2INBAR/WALKING - bez basu / bas z půlových not / bas ze čtvrtových not  
 YES/NO - zapnout funkci / vypnout funkci

## Příklad konfiguračního souboru

input\_type: DIR - budeme načítat z adresáře  
 input\_dir: melodies - z tohoto adresáře zpracujeme všechny soubory

output\_dir: arrangements - do tohoto adresáře budou uložena aranžmá  
 max\_solution\_count: 3 - chceme maximálně 3 řešení  
 bass: 2INBAR - chceme bas z půlových not  
 show\_chords: YES - zobraz akordy v aranžmá  
 show\_passing\_tones: YES - vyznač přechodové tóny v aranžmá  
 transpose: NO - ponech všechny nástroje v C hlase

targetEval\_threshold: 7  
 targetEval\_IsChordTone: 5  
 targetEval\_IsChordTension: 2  
 targetEval\_IsMediumLongTone: 2  
 targetEval\_IsLongTone: 4  
 targetEval\_IsNotLastChordNote: 3

pause\_when\_finish: YES - až skončíš, čekej na ENTER



## Příloha 3: Ukázky výsledných aranžmá

Na přiloženém mediu lze nalézt ke všem zde uvedeným aranžmá MIDI ukázky.

[arr\_Cdur7.mid]

Musical score for arr\_Cdur7.mid, featuring six staves: two alto saxophones, two tenor saxophones, one baritone saxophone, and one bass. The score is in common time (C) and C major. The alto saxophones play a melodic line with eighth and quarter notes. The tenor saxophones play a similar melodic line with some chromaticism. The baritone saxophone plays a more rhythmic line with eighth notes. The bass plays a simple harmonic accompaniment with quarter and eighth notes.

[arr\_CharlieParker1.mid], [arr\_CharlieParker2.mid]

Musical score for arr\_CharlieParker1.mid and arr\_CharlieParker2.mid, featuring six staves: two alto saxophones, two tenor saxophones, one baritone saxophone, and one bass. The score is in common time (C) and D major. The alto saxophones play a melodic line with eighth and quarter notes, including a triplet. The tenor saxophones play a similar melodic line with some chromaticism. The baritone saxophone plays a more rhythmic line with eighth notes. The bass plays a simple harmonic accompaniment with quarter and eighth notes.

First system of musical notation, consisting of six staves. The top five staves are in treble clef with a key signature of one sharp (F#). The bottom staff is in bass clef with the same key signature. The music features a complex melodic line with many accidentals and rests, and includes several triplet markings (indicated by a '3' above a bracket) across the upper staves.

Second system of musical notation, consisting of six staves. The top five staves are in treble clef with a key signature of one sharp (F#). The bottom staff is in bass clef with the same key signature. The music continues with complex melodic lines and includes several triplet markings (indicated by a '3' above a bracket) across the upper staves.

Third system of musical notation, consisting of six staves. The top five staves are in treble clef with a key signature of one sharp (F#). The bottom staff is in bass clef with the same key signature. The music continues with complex melodic lines and includes several triplet markings (indicated by a '3' above a bracket) across the upper staves.

A musical score for a piano arrangement in G major, 4/4 time. It consists of six staves. The top five staves are treble clef, and the bottom staff is bass clef. The music features a melodic line with eighth and sixteenth notes, and a bass line with quarter notes. There are three triplet markings in the upper staves.

[arr\_InTheMood1.mid], [arr\_InTheMood2.mid], [arr\_InTheMood3.mid]

A musical score for a saxophone and bass arrangement in G major, 4/4 time. It consists of six staves. The top five staves are treble clef and labeled 'alto sax', 'alto sax', 'tenor sax', 'tenor sax', and 'baritone sax'. The bottom staff is bass clef and labeled 'bass'. The saxophone parts feature melodic lines with eighth and sixteenth notes, and the bass part features a simple quarter-note bass line. There are three triplet markings in the saxophone parts.

alto sax

alto sax

tenor sax

tenor sax

baritone sax

bass

alto sax

alto sax

tenor sax

tenor sax

baritone sax

bass

3