Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

# DIPLOMOVÁ PRÁCE

Petr Sokol

## Optimalizace obsazení turnusů řidičů autobusové dopravy

**Katedra pravděpodobnosti a matematické statistiky**

Vedoucí diplomové práce: Mgr. Bohumír Bartušek

Studijní program: Matematika

Studijní obor: Pravděpodobnost, matematická statistika a ekonometrie

Rád bych na tomto místě poděkoval vedoucímu této diplomové práce panu Mgr. Bohumíru Bartuškovi a konzultantovi panu Doc. RNDr. Petru Lachoutovi CSc, za ochotu při vedení práce, trpělivost a zapůjčení studijních materiálů, dále paní Mgr. Aleně Sokolové za cenné rady a připomínky.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 17.7.2006

Petr Sokol

Petr Sokol

# Obsah

**Název práce:** Optimalizace obsazení turnusů řidičů autobusové dopravy

**Autor:** Petr Sokol

**Katedra:** Katedra pravděpodobnosti a matematické statistiky

**Vedoucí diplomové práce:** Mgr. Bohumír Bartušek

**e-mail vedoucího:** bartusek@svt.cz

**Abstrakt:** Cílem diplomové práce je vytvořit algoritmus pro určování optimálního plánu přidělení řidičů a vozidel na předem dané schéma jízd. Naším cílem při tvorbě plánu je minimalizace přímých nákladů spojených s přepravou. Zároveň se snažíme vytvořit vyvážený plán, s ohledem na vytížení řidičů a vozidel. Struktura problému plánování byla matematicky popsána sestavením rozsáhlého lineárního modelu, jehož součástí jsou algebraické formulace jednotlivých omezení, která je nutné v automobilové dopravě dodržovat. V tomto modelu figurují binární a reálné proměnné, jedná se tedy o model smíšeného celočíselného programování. O řešení celočíselných úloh se dá říci, že je obecně mnohem komplikovanější než řešení úloh bez celočíselných omezení. K praktickému řešení úlohy byl použit optimalizační software GAMS, který pro řešení celočíselných úloh používá algoritmus větvení a mezí. V GAMSu byl, na základě výše zmíněného modelu, sestaven programový kód, který je schopen pro sérii vstupních dat určit optimální plán obsazení jízd. Součástí práce je i několik vyřešených vzorových úloh.

**Klíčová slova:** smíšené celočíselné programování, algoritmus větvení a mezí, lineární model

**Title:** Optimalization in allocation drivers at bus service

**Author:** Petr Sokol

**Department:** Department of Probability and Mathematical Statistics

**Supervisor:** Mgr. Bohumír Bartušek

**Supervisor's e-mail address:** bartusek@svt.cz

**Abstract:** The thesis is focused on suggesting an algorithm for planning of optimal assignment drivers and vehicles to scheme of expeditions. When we make the assignment, we aspire to minimize the total transportation costs. By the same mail we push for well-balanced assignment, with respect to utilization of drivers and vehicles. The structure of the planning problem there is describing by making out of a wide linear model, whose parts are algebraic formulations of restrictions, which one must in traffic abide. In this model we can see binary and real variables that mean we draw up a model of mixed integer programming. We can say, that the integer programming is in general more complicated, then integer programming. We used optimization software named GAMS for solution this problem. The software uses Branch and Bound algorithm for the integer number problems. We draw up a program in GAMS, which is able to set an optimal plan for assignment drivers and vehicles to expeditions. A part of this work is couple of exemplary tasks, with their solution.

**Keywords:** mixed integer programming, Branch & Bound algorithm, linear model

# Úvod

Přestože v poslední době dochází k poměrně rychlému rozvoji možností i kapacity výpočetní techniky, stále existuje velký počet dopravních firem, ve kterých plánovaní jízd musí provádět lidé (tzv. dispečeři) zcela manuálně, případně za minimální podpory výpočetní techniky. Tato činnost je ovšem velice náročná a její náročnost díky několika faktorům stále stoupá. Mezi takovéto faktory se řadí rostoucí počet různých nařízení, která je nutné v dopravě dodržovat nebo stále sílící konkurence, která dopravce nutí k větší flexibilitě a zároveň ke snižování provozních nákladů.

V této práci se pokusíme problém plánování jízd důkladně popsat a navrhnout algoritmus, který bude schopna vyřešit výpočetní technika, bez dalšího zásahu dispečera. Na dispečerovi ponecháme pouze odpovědnost za správné sestavení vstupních souborů.

# 1. Popis problému plánování jízd

## 1.1 Struktura zadání

Do úlohy vstupují tři hlavní oblasti: řidiči, autobusy a úkoly. Hlavním účelem úlohy je přiřadit jednotlivé řidiče a autobusy, případně i jiné typy vozidel, konkrétním úkolům tak, aby bylo možné za minimálních nákladů realizovat všechny požadavky na přepravu, které dopravce eviduje.

Pod jednotlivými úkoly si můžeme představit nejen jízdy, které se musí uskutečnit, ale i další činnosti, které musí být provedeny, např. údržba autobusů, školení či dovolené řidičů. Díky tomu nemusí vždy platit, že je k vykonání úkolu potřebný řidič i autobus. U některých úkolů nemusí být ani pevně určen čas, kdy má dojít ke splnění úkolu, např. zákazník požaduje přepravu nákladu v následujícím týdnu, ale přesné určení dne přepravy ponechá na přepravci. Nebo je například nutné do konce probíhajícího měsíce provést preventivní údržbu vozidla, ale ani v tomto případě není určen přesný datum, kdy má být údržba provedena.

Při plánování jízd je nutné nepřekročit žádné ze závazných omezení daných zákony a vyhláškami. Jedná se především o jednotlivá omezení daná zákoníkem práce a české i mezinárodní směrnice upravující povinnosti přepravců v automobilové dopravě.

Při plánování je dále nutné počítat s dalšími skupinou podmínek – jedná se o technická omezení. Například ne všechna vozidla mohou být použita na konkrétní jízdu nebo někteří řidiči nemají oprávnění na řízení všech typů vozidel.

V neposlední řadě je nutné, aby byl připravený plán jízd vyvážený – všichni řidiči by měli během plánovacího období odpracovat přibližně stejnou dobu a zároveň vozidla by měla být přibližně stejně vytížena.

## 1.2 Zákonem daná omezení a další vyhlášky

Snaha o neustálé zvyšování bezpečnosti silniční dopravy vede k potřebě regulovat doby řízení a doby odpočinků řidičů. I z tohoto důvodu byla v roce 1970 uzavřena mezinárodní dohoda AETR (Evropská dohoda o silniční dopravě), touto dohodou se řídí mezinárodní doprava. Naproti tomu doprava uvnitř Evropského společenství podléhá Nařízení (EHS) č. 3820/85 Rady ze dne 20. prosince 1985 o harmonizaci určitých sociálních předpisů v silniční dopravě a Nařízení (EHS) č. 3821/85 Rady ze dne 20. prosince 1985 o záznamovém zařízení v silniční dopravě. Z hlediska povinností řidiče jsou dohoda AETR a výše uvedená nařízení velmi podobná, proto nebudeme u jednotlivých omezení uvádět, zda jsou obsažena v dohodě AETR nebo v těchto nařízeních.

Znění nařízení nebudeme citovat v celé šíři, ale uvedeme pouze pasáže, které se přímo týkají časových omezení, která musí řidiči i dopravci dodržovat. Kompletní znění je možné nalézt například v [5].

***NAŘÍZENÍ (EHS) č. 3820/85**, ODDÍL IV – Doby řízení*

***Článek 6***

*1. Celková doba řízení mezi dvěma denními odpočinky nebo mezi jedním denním odpočinkem a jedním týdenním odpočinkem, v dalším nazývaná "denní doba řízení", nesmí přesáhnout 9 hodin. Dvakrát za týden[1] může být prodloužena na 10 hodin.*

*Po nejvýše šesti denních dobách řízení musí řidič zařadit týdenní dobu odpočinku, jak je stanoveno v článku 8 odstavci 3.*

*Týdenní doba odpočinku smí být přesunuta na konec šestého dne, jestliže celková doba řízení po dobu šesti dnů nepřesahuje maximum odpovídající šesti denním dobám řízení.*

*V přeshraniční osobní dopravě, kromě pravidelné linkové dopravy, se nahrazují slova "šesti" a "šestého" uváděná v předchozím druhém a třetím pododstavci slovy "dvanácti" a "dvanáctého".*

---

[1] Pod pojmem „týden" rozumíme časové období mezi 0:00 hodinami v pondělí a 24:00 hodinami v neděli.

*Každý členský stát může, kromě pravidelné linkové dopravy, rozšířit působnost předchozího odstavce také na vnitrostátní osobní dopravu na svém území.*

*2. Celková doba řízení nesmí překročit devadesát hodin za období dvou po sobě následujících týdnů.*

***NAŘÍZENÍ (EHS) č. 3820/85**, ODDÍL V – Přestávky a doby odpočinku*

**Článek 7**

*1. Po čtyřech a půl hodinách doby řízení musí řidič zařadit přestávku nejméně 45 minut, pokud nezahájí dobu odpočinku.*

*2. Tato přestávka smí být nahrazena kratšími přestávkami, z nichž každá má nejméně 15 minut; tyto přestávky musí být vloženy do doby řízení nebo zařazeny okamžitě po této době řízení tak, aby se vyhovělo ustanovení odstavce 1.*

*3. V případě vnitrostátní osobní linkové dopravy mohou členské státy stanovit odlišně od odstavce 1 minimální dobu přestávky na nejméně 30 minut po době řízení nejvýše 4 hodiny. Takovéto výjimky lze udělit jen v případech, kdy by přerušení doby řízení na více než 30 minut narušovalo plynulost městské dopravy, a když není možno řidičům do čtyřapůlhodinové doby řízení vložit přestávku o délce 15 minut, která by předcházela tuto třicetiminutovou přestávku.*

*4. Během těchto přestávek nesmí řidič vykonávat žádné jiné práce. Pro účely tohoto článku se doba čekání a doba nevěnovaná řízení strávená v jedoucím vozidle, na trajektu nebo ve vlaku nepovažují za "jiné práce". (To znamená, že přestávky lze trávit – pokud dotyčná osoba neřídí – ve vozidle, které řídí jiný řidič, nebo ve vozidle, které je vezeno na trajektu či po železnici.)*

*5. Přestávky podle tohoto článku nesmí být považovány za denní odpočinek.*

**Článek 8**

*1. V průběhu každých 24 hodin musí mít řidič denní odpočinek. Denní doba odpočinku trvá nejméně 11 za sebou následujících hodin; smí být zkrácena na nejméně 9 hodin za sebou následujících hodin nejvýše třikrát v jednom týdnu za podmínky, že bude náhradou poskytnuta odpovídající doba odpočinku jako vyrovnání před koncem následujícího týdne.*

*Ve dnech, ve kterých není denní odpočinek zkrácen podle prvého odstavce, smí být čerpán ve dvou nebo třech oddělených částech během 24 hodin, přičemž jedna z těchto částí musí trvat nejméně 8 za sebou následujících hodin. V takovémto případě se minimální denní doba odpočinku prodlužuje na 12 hodin.*

*2. Jsou-li ve vozidle nejméně dva řidiči, musí mít každý z nich denní odpočinek nejméně 8 za sebou následujících hodin v průběhu každých 30 hodin.*

*3. V každém týdnu musí být jedna z dob odpočinku uvedených v odstavcích 1 a 2 prodloužena a použita jako týdenní odpočinek. Týdenní doba odpočinku trvá nejméně 45 po sobě následujících hodin. Týdenní doba odpočinku smí být zkrácena na nejméně 36 po sobě následujících hodin, je-li vybírána v místě obvyklého stanoviště vozidla nebo v obvyklém místě pobytu řidiče, nebo na nejméně 24 hodin po sobě následujících, je-li vybírána mimo tato místa. Každé zkrácení týdenní doby odpočinku musí být nahrazeno odpovídající dobou odpočinku vybranou vcelku před koncem třetího týdne následujícího po dotyčném týdnu.*

*4. Týdenní doba odpočinku, která začíná v jednom týdnu a pokračuje do následujícího týdne, smí být připojena k libovolnému z těchto týdnů.*

*5. V případě osobní dopravy, na kterou se vztahuje článek 6 odstavec 1, čtvrtý nebo pátý pododstavec, se týdenní doba odpočinku smí odložit do týdnu následujícího po tom, v němž má být odpočinek čerpán, a smí se připojit k týdennímu odpočinku tohoto týdne.*

*6. Jakákoli doba odpočinku vybraná náhradou za zkrácení denních a/nebo* *týdenních dob odpočinku musí být připojena k jinému odpočinku trvajícímu nejméně osm hodin a musí být poskytnuta na žádost řidiče v místě obvyklého stanoviště vozidla nebo obvyklém místě pobytu řidiče.*

*7. Řidič smí trávit denní odpočinek ve vozidle, pokud je vozidlo vybaveno lehátkem nebo lůžkem a vozidlo stojí.*

## 1.3 Interpretace zákonných omezení

Je vidět, že se jedná o značně složitý systém podmínek, jehož převod do čistě matematického zápisu je poměrně komplikovaný. Naproti tomu stojí potřeba zapsat všechny podmínky v jednoduché, pokud možno lineární podobě, aby nedošlo k příliš velkému zkomplikování optimalizační úlohy a tím i prodloužení doby výpočtu, případně zvýšení nároků na použitý hardware.

Proto je nezbytné u každé podmínky rozhodnout, zda je nutné nechat podmínku přesně v zákonném tvaru nebo zda je možné podmínku zjednodušit, případně zajistit její splnění mimo vlastní optimalizační úlohu.

Mezi podmínky, jejichž splnění se dá zajistit mimo vlastní optimalizační úlohu zařadíme podmínku na dodržování povinných přestávek (přestávky trvající nejméně 45 minut, které musí řidič zařadit vždy po čtyřech a půl hodinách jízdy nebo v jejich průběhu). S tím, že odpovědnost za správné dodržování odpočinků necháme na řidiči. Je pouze nutné, aby dopravce již při zadávání jízd na takovéto přestávky pamatoval a při určování času jízdy nepočítal pouze s dobou nutnou pro ujetí celé vzdálenosti a času dalších činností, které budou nutné v rámci jízdy vykonat, ale i s časem určeným na zákonné přestávky. Konkrétní rozložení přestávek mezi doby řízení si určí řidič sám, například podle dopravní situace nebo dle své únavy.

Obdobná situace nastane u dálkových – vícedenních jízd, kdy je nutné aby řidiči v průběhu takovéto jízdy zařadili nejen dostatečné množství přestávek v řízení, ale i odpovídající množství denních odpočinků. U některých dlouho trvajících jízd může být nutné zařadit i týdenní odpočinky. V této optimalizační úloze budeme tedy rozlišovat dva typy jízd:

- Krátkodobé jízdy – takovéto jízdy se nesmí překrývat s denním odpočinkem. Řidič může na takovouto jízdu nastoupit po skončení svého denního odpočinku a jízda musí skončit před začátkem dalšího řidičova denního odpočinku.
- Dlouhodobé jízdy – součástí takovýchto jízd mohou být i denní odpočinky. Při zadávání takovýchto jízd musí být obdobně jako v předchozím odstavci počítáno i s dobou trvání denních odpočinků řidiče. Za správné zařazení denních odpočinků opět odpovídá řidič. Mezi dlouhodobé jízdy budou patřit i všechny jízdy se dvěma řidiči, proto nebude v úloze nutné počítat se speciální podmínkou na denní odpočinky u dvoučlenných osádek.

Podmínku na správné zařazování denních odpočinků dále zjednodušíme tím, že nebudeme uvažovat možné výjimky od základní jedenáctihodinové denní doby odpočinku. Pokud bychom chtěli umožnit využití všech zákonných výjimek (možné zkrácení doby odpočinku v některých dnech, případně rozdělení denního odpočinku na několik částí), museli bychom do úlohy zavést další sérii proměnných – časy jednotlivých denních odpočinků každého řidiče a další skupinu poměrně složitých podmínek, které by zajistily nepřekročení žádné z podmínek na délku denního odpočinku. Zvláště velké komplikace by přineslo povinné nahrazování času, o který byly jednotlivé denní odpočinky zkráceny. Umožnění těchto výjimek by přineslo další nepříjemný efekt – plán jízd se vždy připravuje na určitou dobu, ale nahrazování časů, o které byly denní odpočinky zkráceny, je nutné zajistit kontinuálně. Pokud bychom tedy na konci jednoho plánovacího období u řidičů využili větší množství možných zkrácení denních dob odpočinků, mohlo by to přinést komplikace v následujícím plánovacím období, kdy bychom museli tato zkrácení nahrazovat.

U týdenních dob odpočinku zavádíme obdobné zjednodušení jako v předchozím případě. Budeme počítat s tím, že každá doba týdenního odpočinku musí trvat nejméně 45 hodin. Zanedbáme tedy zákonem umožněné zkrácení týdenní doby odpočinku na 36 hodin (24 hodin mimo místo obvyklého pobytu řidiče). I v tomto případě nám vypuštění možných výjimek přinese zjednodušení celé úlohy a vyhneme se možnému nechtěnému ovlivnění následného plánovacího období.

S tímto problémem souvisí další zákonná podmínka, která nařizuje zařadit týdenní dobu odpočinku maximálně po šesti denních dobách řízení. I u této podmínky platí, že její plné zahrnutí by model velice zkomplikovalo, proto podmínku zjednodušíme na kontrolu součtu dob řízení mezi jednotlivými týdenními odpočinky. Konkrétně budeme požadovat, aby celková doba řízení mezi dvěmi týdenními odpočinky nepřesáhla dobu odpovídající šesti denním dobám řízení.

Další mírné zjednodušení týkající se týdenní doby odpočinku zavádíme kvůli zvýšení přehlednosti úlohy. Dle tohoto zjednodušení je nutné, aby týdenní odpočinek začínal vždy v týdnu, ke kterému přísluší. Podle vyhlášky může začínat i v předchozím týdnu, ale musí do příslušného týdne alespoň zasahovat. Pokud bychom chtěli nechat původní podmínku v platnosti, čas začátku týdenního odpočinku by nebyl zdola omezen začátkem týdne, ale rozdílem začátku týdne a hodnoty minimálního týdenního odpočinku.

Ostatní podmínky ponecháme bez dalšího zjednodušení a všechny převedeme do systému omezení v optimalizační úloze.

## 1.4 Struktura úkolů

Aby bylo možné popsat všechny přípustné požadavky na přepravu, bylo nutné rozložit popis jednotlivých úkolů do tří úrovní. V nejvyšší úrovni se rozlišují jednotlivé **úkoly**. V podstatě se jedná o jednotlivé zakázky dopravce a další doplňkové akce, které mohou způsobit vytížení řidiče nebo vozidla. Úkolem může být například požadavek na převoz účastníků zájezdu, zajištění nepřetržité kyvadlové dopravy mezi dvěma místy (např. mezi nákupním střediskem a sídlištěm nebo dvěma vlakovými nádražími v případě výluky na vlakové trati). Dále je mezi úkoly nutné zařadit další činnosti související s provozem dopravního podniku, např. údržba vozidel, různé nepřítomnosti řidičů, jako jsou dovolené, nemoci, školení nebo dočasné přidělení na jinou práci.

U některých úkolů nemusí být předem přesně stanoven čas a datum, kdy se má daný úkol vykonat – existuje více alternativ, kdy je možné úkol provést – například preventivní údržba vozidla, která může být provedena buď v prvním nebo druhém týdnu v měsíci. Ani přesné provedení jízdy nemusí být určeno předem – například jízda, kdy je možné odvést celý náklad najednou větším vozidlem nebo jet dvakrát menším vozidlem. V těchto případech nejsme schopni předem říci, která z alternativ bude pro dopravce, s ohledem na náklady a vytížení řidičů a vozidel ostatními úkoly, výhodnější, proto ponecháváme otevřeny všechny možnosti a výběr nejvhodnější alternativy ponecháme na optimalizačním algoritmu.

Kvůli reprezentování různých možností realizace daného úkolu zavádíme další rozpadovou úroveň – jednotlivé úkoly se skládají z jedné nebo více **alternativ**. Ze všech alternativ jednoho úkolu musí být vždy vybrána právě jedna „nejvýhodnější" alternativa, která se uskuteční. Ostatní alternativy uskutečněny nebudou.

Další komplikaci přináší do plánování jeden specifický druh úkolů, jedná se o periodicky se opakující krátké jízdy v městské dopravě. Například linka, spojující nákupní centrum s jinou částí města, která každodenně přepravuje cestující v čase od 6:00 do 22:00 hodin. Přičemž objetí jednoho okruhu trvá 1 hodinu, stejná jízda se tedy opakuje šestnáctkrát během jednoho dne. Na takovéto jízdy je zpravidla využíván jeden autobus, ale jeho nasazení trvá příliš dlouho – déle než trvá maximální denní doba řízení, proto je nutné, aby u takovýchto jízd docházelo v průběhu dne k střídání řidičů. Předem ovšem není dané, kdy má ke střídání dojít, daná je pouze skutečnost, že se řidiči mohou vystřídat pouze ve výchozím bodě.

Kvůli reprezentování tohoto typu požadavků zavádíme poslední rozpadovou úroveň, v této úrovni se alternativy rozpadají na jednotlivé **jízdy**. Alternativy se skládají vždy z jedné nebo více jízd. Jízdy z jedné alternativy následují vždy těsně po sobě (čas konce jedné jízdy se shoduje s časem začátku další jízdy). Dále platí, že na všechny jízdy z jedné alternativy musí být použit stejný autobus, naproti tomu řidiči se mohou mezi jednotlivými jízdami vystřídat. Kvůli souvislosti pracovní doby řidičů je ovšem vhodné, aby ke střídání řidičů nedocházelo příliš často a řidič pokud možno absolvoval celou svoji směnu v souvislém čase.

## 1.5 Technická omezení

Do optimalizační úlohy musíme k omezením, sestavených na základě zákonných norem, přidat i další podstatná omezení na straně dopravce. Tato omezení budeme nazývat „technická omezení". Obecně můžeme říci, že ne všichni řidiči a ne všechna vozidla můžeme přiřadit na každou jízdu. Přitom takováto omezení budeme uvádět ve třech úrovních a to vždy ve vztahu dvou skupin.

Jedná se o vyloučení některých kombinací jízd a řidičů. Tento případ může nastat například u jízd, ve kterých jsou přepravovány osoby. Na takovéto jízdy mohou být přiděleni pouze řidiči s dostatečnou praxí.

Dále je nutné vyloučit některé kombinace jízd a vozidel, například z kapacitních důvodů vozidla. Tento a předchozí typ vyloučení některých kombinací budeme využívat i pro správné naplánování „speciálních" jízd. Je-li takovouto speciální jízdou plánovaná údržba konkrétního vozidla, musíme u všech ostatních vozidel zakázat přiřazení na tuto jízdu.

Poslední vylučovanou kombinací je kombinace některých řidičů a vozidel. V tomto případě se jedná o zamezení stavu, kdy bude řidič přidělen na vozidlo, na které nemá řidičské oprávnění nebo jej nemůže řídit z jiného důvodu.

## 1.6 Účel úlohy

Vlastním účelem optimalizační úlohy je přiřadit plánovaným jízdám konkrétní řidiče a vozidla. U úkolů s více možnými alternativami je nutné v rámci optimalizační úlohy určit, které jízdy se uskuteční a které nikoliv. Přitom je nutné dbát na dodržení všech zákonných i technických omezení, uvedených výše. Již z názvu úlohy je patrné, že se nám jedná o vytvoření v nějakém smyslu optimálního plánu. V této úloze se budeme snažit plán optimalizovat z více ohledů zároveň.

V první řadě se budeme snažit minimalizovat přímé náklady spojené s jízdou. Mezi tyto náklady řadíme především pořizovací cenu spotřebovaného paliva. Dále je do této položky nutné zahrnout další náklady spojené s uskutečněním jízdy, například poplatky za užití některých komunikací, náklady spojené s opotřebením vozidla nebo odměna, která musí být vyplacena řidiči za absolvování dané jízdy. Přesnou výši nákladů, které budou na jízdu vynaloženy nejsme, bohužel, schopni předem určit. Tyto náklady mohou být ovlivněny různými náhodnými vlivy, například počasím nebo dopravní situací. Proto je nutné tyto náklady nějakým způsobem odhadnout. Takovéto odhady budou vždy založeny na předchozích zkušenostech se stejnými nebo podobnými typy jízd. Pro dopravce je proto vhodné, aby si průběžně vytvářel různé statistiky nákladů na jízdy v závislosti na použitých vozidlech a řidičích. Z takovýchto statistik se dá poměrně spolehlivě určit očekávaná výše nákladů na jízdu.

V této úloze budeme náklady dělit do tří vztahových úrovní:

- Náklady, které bude nutné vynaložit na jízdu v závislosti na použitém vozidle. Je zřejmé, že náklady výše nákladů na bude záviset i na typu vozidla, které bude použito. Například vozidlo s vyšší přepravní kapacitou má i vyšší spotřebu. Proto se vždy budeme snažit vybrat na jízdu takové vozidlo, které svou kapacitou a dalšími vlastnostmi nejlépe odpovídá charakteru jízdy. Mohou ovšem nastat i situace, kdy budou všechna k jízdě vhodná vozidla obsazena a my budeme nuceni použít ne zcela optimální vozidlo.
- Navýšení / snížení nákladů na jízdu v závislosti na přiděleném řidiči. Pod tímto typem nákladů si můžeme zejména představit z předchozích jízd vypozorované zvýšení (snížení) spotřeby pohonných hmot, pokud je pro jízdu vybrán konkrétní řidič. Nebo nějaké další náklady spojené s výběrem konkrétního řidiče. Do této skupiny nákladů můžeme zahrnout i položky, které se nedají konkrétně vyčíslit – jedná se o různé preference. Například stálý zákazník je zvyklý, že všechny jím objednané jízdy vykonává tentýž řidič. Pokud je zákazník s tímto řidičem spokojený, preferujeme, aby byl tento řidič přednostně vybírán i nadále. V opačném příkladě mohl být zákazník s řidičem nespokojen, v tom případě preferujeme, aby řidič již na jízdu objednanou zákazníkem vybrán nebyl. Na rozdíl od technických omezení, popsaných výše, se v těchto případech nejedná o striktní omezení, ale pouze o určení, který z možných výběrů nám více vyhovuje. Jediná komplikace spočívá v tom, že musíme být schopni takovéto preferenci přiřadit nějakou „cenu“, abychom ji mohli přičíst k ostatním nákladům.
- Navýšení / snížení nákladů v závislosti na kombinaci řidiče a vozidla. I v tomto případě se především jedná o zanesení našich předchozích

zkušeností do aktuálního plánování. Můžeme mít statisticky zjištěno, že má některý řidič na určitém typu vozidla nižší průměrnou spotřebu než ostatní řidiči.

Další hledisko, které se budeme v této úloze snažit optimalizovat, je přidělování řidičů k jednotlivým vozidlům. V automobilové přepravě je zvykem, že řidič má vždy přiděleno konkrétní vozidlo a většinu jízd vykonává s tímto vozidlem. Tento model je výhodný pro řidiče, kteří si díky tomu nemusí často zvykat na jiná vozidla, i pro dopravce, protože řidič obvykle odpovídá za technický stav přiděleného vozidla. Z toho důvodu se budeme snažit, aby byli řidiči v maximální míře přidělováni na „svá" vozidla. Tuto skutečnost budeme vyjadřovat pomocí hodnoty preference, kterou přiřadíme každé (možné) kombinaci řidiče a vozidla. Výše popsané případy budou mít vždy vysokou preferenci. Naproti tomu nízká preference bude přidělena případu, kdy z nějakého důvodu není vhodné, aby řidič konkrétní vozidlo řídil, například v situaci, kdy řidič nemá s řízením vozidla zkušenosti nebo sám řidič preferuje řízení vozidel jiných typů.

Jak již bylo zmíněno při popisu struktury jízd, dalším z optimalizačních kritérií bude snaha o minimalizaci počtu výměn řidičů mezi jednotlivými jízdami v rámci jedné alternativy. Toto kritérium zavádíme kvůli zamezení situaci, kdy se budou řidiči střídat po příliš krátkých časových úsecích.

Při plánování se dále musíme snažit o rovnoměrné rozložení pracovní doby mezi všechny řidiče. Kvůli zabezpečení tohoto požadavku nejdříve vypočítáme průměrnou pracovní dobu připadající na jednoho řidiče a dále pro každého řidiče zavedeme dvě hodnoty – překročení a nedosažení průměrné pracovní doby. Jedná se o kladnou a zápornou část rozdílu průměrné pracovní doby a pracovní doby konkrétního řidiče. Je tedy zřejmé, že pouze jedna z těchto dvou hodnot může být nenulová. I v tomto případě musí dopravce stanovit nějaký koeficient, kterým se hodnota překročení nebo nedosažení průměrné odpracované doby vynásobí, aby se i tyto hodnoty daly správně zahrnout do optimalizační úlohy. Takto stanovené koeficienty mohou být u řidičů různé, zároveň se u každého řidiče může lišit koeficient pro překročení průměrné odpracované doby od koeficientu pro nedosažení průměrné pracovní doby. Při určování těchto koeficientů může dopravce vycházet například z ochoty konkrétních řidičů pracovat přesčas nebo i z kvality řidičů – lepšího řidiče nechá pracovat déle.

Obdobné kritérium budeme sledovat i u vozidel, u kterých ve většině případů také preferujeme rovnoměrné vytížení. Na rozdíl od řidičů nebudeme u vozidel sledovat odchylku od průměrné doby, ale od průměrné ujeté vzdálenosti připadající na jedno vozidlo. Ostatní zůstává obdobné jako u řidičů.

Poslední optimalizační kritérium by se dalo považovat za samostatnou optimalizační úlohu. Jedná se o přeplánování plánu, který byl stanoven dříve a v aktuální době se podle něho obsazování jízd řídí. K přeplánování může docházet z různých důvodů, například když dojde k onemocnění některého řidiče, když se na některém z vozidel vyskytne porucha, která si vyžádá dlouhodobou opravu nebo při dodatečném přidání původně neplánované jízdy. Při takovémto přeplánování je důležité, aby byl stávající plán v maximálně možné míře zachován. Maximální zachování původního plánu požadujeme zejména u řidičů, k čemuž nás nutí i zákoník práce. Ten požaduje, aby zaměstnavatel vypracoval rozvrh stanovené týdenní pracovní doby a seznámil s ním zaměstnance nejpozději dva týdny předem.[2]

Přeplánování bude v rámci této úlohy řešeno následujícím způsobem: Mezi ostatními vstupními parametry (daty) necháme do úlohy vstoupit i obsazení jízd řidiči a vozidly dle stávajícího plánu. Poté budeme v rámci optimalizační úlohy minimalizovat počet změn mezi stávajícím a nově generovaným plánem.

Tímto jsme uvedli výčet kritérií, která se budeme v této úloze snažit optimalizovat. Popsaný typ optimalizační úlohy se obvykle nazývá vícekriteriální optimalizační úloha. Vícekriteriální optimalizační úlohy se dají řešit dvěma způsoby.

Buď převedením na standardní jednokriteriální optimalizační úlohu. Pro tento převod je nutné mezi optimalizačními kritérii zvolit jedno hlavní kritérium, podle kterého se bude v nové úloze optimalizovat. Ostatní optimalizační kritéria nahradíme přidáním speciálních podmínek (omezení) k původním podmínkám optimalizační úlohy. Pro vytvoření těchto podmínek musíme u každého optimalizačního kritéria určit minimální, resp. maximální hodnotu a podmínky

[2] Zákon č. 65/1965 Sb., zákoník práce, § 85, odst. 2.

zapíšeme ve tvaru, kdy hodnota každého z kritérií musí být alespoň rovna odpovídající minimální hodnotě, případně nižší nebo rovna maximální hodnotě.

Druhý možný způsob spočívá ve spojení všech optimalizačních kritérií do jediného. Toto spojení vytvoříme jako vážený součet všech původních kritérií, kde jednotlivé váhy odpovídají důležitosti jednotlivých kritérií.

Další možnosti vzniknou různými kombinacemi těchto dvou přístupů. Například můžeme minimalizovat váženou sumu všech kritérií a zároveň pro některá kritéria nastavit mezní hodnoty.

V naší úloze budeme používat druhou z výše popsaných metod – budeme optimalizovat vážený součet hodnot jednotlivých optimalizačních kritérií. Váhy ovšem nebudou určeny fixně, ale budou do optimalizace vstupovat spolu s dalšími vstupními daty. Tím bude dopravci umožněno měnit přístup k optimalizaci v různých situacích. Například při přeplánování může být u ostatních kritérií nastavena nulová váha a tím bude dopravce sledovat maximální zachování původního plánu bez ohledu na ostatní kritéria.

## 1.7 Plánovací období

Plán přiřazení konkrétních řidičů a vozidel jednotlivým jízdám bude určován vždy pro konkrétní „plánovací období“. Kvůli struktuře výše uvedených závazných vyhlášek, v nichž se velká část omezení vztahuje k týdnům, budeme v této úloze uvažovat pouze plánovací období sestávající se z jednoho nebo více celých týdnů. Tato skutečnost by neměla dopravci příliš komplikovat žádnou z činností související s plánováním. Uveďme příklad, kdy bude dopravce potřebovat vytvořit plán na kalendářní měsíc, s tím, že plán na předchozí měsíc je již známý. V tomto příkladě se do nového plánovacího období zahrne i týden, ve kterém jeden měsíc končí a druhý začíná a obsazení dané původním plánem se zafixuje pomocí technických omezení. Tím dojde k plynulému přechodu mezi jednotlivými plánovacími obdobími.

Ačkoliv jsou jednotlivá plánovací období od sebe přesně časově oddělena dochází k jejich vzájemnému prolínání. A to v několika ohledech. Za první z nich, můžeme považovat vícedenní jízdy, které začínají v jednom plánovacím období a pokračují v následujícím plánovacím období. Další ovlivnění následujícího plánovacího období mohou způsobit přesahující denní nebo týdenní odpočinky řidičů. Ve všech těchto případech ponecháme na dopravci správné zadání dat (technických omezení) pro následující plánovací období tak, aby nedošlo ke vzájemné kolizi těchto dvou plánovacích období.

Další oblast, ve které se plánovací období přímo ovlivňují je kontrola zákonného omezení na celkovou dobu řízení ve dvou po sobě jdoucích týdnech. Tuto podmínku je nutné dodržet i v období posledního týdne jednoho plánovacího období a prvního týdne následujícího období. Předpokládejme, že před vlastním plánováním víme, kolik hodin budou řidiči řídit během týdne předcházejícího aktuálnímu plánovacímu období[3]. Tato hodnota se tedy musí projevit ve vstupních datech a musí být zohledněna v jedné z podmínek optimalizační úlohy.

Abychom předešli situaci, kdy v posledním týdnu jednoho plánovacího období odřídí některý řidič příliš mnoho hodin a díky tomu jej nebudeme moci využít v prvním týdnu následujícího plánovacího období, zavádíme další pomocné omezení – v posledním týdnu plánovacího období může řidič odřídit maximálně 60% povolené dvoutýdenní doby řízení.

Obdobná situace jako u dvoutýdenní doby řízení nastává i u denní doby řízení. Pokud pro některého řidiče končí jedno plánovací období dobou řízení a následující plánovací období dobou řízení začíná, je nutné tyto dvě doby řízení sečíst a zkontrolovat zda tento součet nepřesahuje povolenou mez. Stejně jako v předchozím případě přidáme údaj o době řízení na konci předchozího plánovacího období mezi vstupní data a budeme s ním počítat v optimalizační úloze.

Tato optimalizační úloha řeší plánování jízd vždy na relativně krátká plánovací období, jejichž délka nebude ve většině případů výrazně přesahovat jeden měsíc. Proto nejsme schopni v rámci této úlohy řešit další zákonný požadavek, tentokrát daný zákoníkem práce. Jedná se o omezení na maximální počet hodin práce přesčas.[4] Dohlížet na dodržování tohoto omezení by měl v průběhu celého roku dopravce a velký počet přesčasových hodin u řidičů řešit zvýšením penalizace za překročení průměrné doby u těchto řidičů. Pokud by došlo

[3] Aktuálním plánovacím obdobím rozumíme období, na které vytváříme plán.

[4] Zákon č. 65/1965 Sb., zákoník práce, § 96, odst. 1.

ke situaci, kdy se velký počet přesčasových hodin objevuje u všech řidičů, by měla být pro dopravce signálem pro přijetí dalších řidičů.

# 2. Sestavení matematického modelu

## 2.1 Popis vstupních dat

Nyní se dostáváme k popisu struktury dat, nad kterými bude probíhat optimalizační výpočet. Data lze rozdělit do několika oblastí – data popisující jednotlivé řidiče, dále data popisující jednotlivá vozidla a data popisující jednotlivé jízdy. U každé z oblastí se jedná o několik určujících parametrů, v matematickém zápisu úlohy můžeme každý z těchto parametrů chápat jako vektor, jehož i-tá složka popisuje i-tý prvek z množiny řidičů (resp. vozidel, jízd). Každý z těchto vektorů budeme pro snazší zápis značit vhodným písmenem.

Další tři oblasti dat popisují vlastnosti, které přinese kombinace buď řidiče a vozidla, řidiče a jízdy nebo vozidla a jízdy. I v tomto případě je každá takováto kombinace popsána několika parametry, které můžeme chápat jako dvourozměrné matice.

Kvůli přeplánování, které je součástí této úlohy, musíme mezi vstupní data zařadit i původní plán obsazení jízd. Do úlohy vstupuje několik dalších informací, které se svým charakterem dají spíše považovat za konstanty, těmi jsou konkrétní mezní hodnoty zákonných omezení a informace o datu začátku a délce plánovacího období.

## 2.2 Podrobný výčet vstupních dat

### 2.2.1 Vozidla

Celkový počet vozidel budeme značit písmenem $B$, pro jednotlivá vozidla budeme používat index $b$. Vozidla budou popsána těmito charakteristikami:

- $\beta_b^+ \ldots$ Koeficient penalizace za překročení průměrného počtu odjetých kilometrů, reálné číslo nabývající hodnotu z intervalu $\langle 0,1 \rangle$.

- $\beta_b^-$ ... Koeficient penalizace za nedosažení průměrného počtu odjetých kilometrů, reálné číslo nabývající hodnotu z intervalu $\langle 0,1 \rangle$.
- $\Omega_b$ ... Počet předem fixovaných kilometrů, kladné reálné číslo. Nenulová hodnota bude vyplněna například v případě, kdy bude nutné v průběhu měsíce přistoupit k přeplánování. V takovém momentě jsou již některé jízdy odjety a je možné, že v těchto jízdách došlo k nerovnoměrnému využití vozidel, což budeme při přeplánování chtít kompenzovat. Nenulová hodnota může být v tomto poli použita i u vozidla, které nechce dopravce z nějakého důvodu využívat, například kvůli jeho vysoké poruchovosti. Pokud je pro vozidlo $b$ hodnota $\Omega_b$ vyšší než průměrný počet kilometrů připadající na jedno vozidlo, bude v účelové funkci penalizováno každé použití vozidla $b$.

### 2.2.2 Řidiči

Celkový počet řidičů budeme značit písmenem $R$, pro označení jednotlivých řidičů budeme používat index $r$. Řidiči budou popsáni následujícími charakteristikami:

- $\rho_r^+$ ... Koeficient penalizace za překročení průměrné odpracované doby, reálné číslo nabývající hodnotu z intervalu $\langle 0,1 \rangle$.
- $\rho_r^-$ ... Koeficient penalizace za nedosažení průměrné odpracované doby, reálné číslo nabývající hodnotu z intervalu $\langle 0,1 \rangle$.
- $\varphi_r$ ... Předem fixovaná odpracovaná doba, kladné reálné číslo. Nenulová hodnota bude vyplněna například u přeplánování – v momentě, kdy jsou již některé jízdy odjety a řidiči nemají odpracován stejný počet hodin. Tento atribut se dá použít i v případě, kdy některý řidič pracuje na částečný úvazek – zatímco ostatní řidiči budou mít na začátku plánovacího období $\varphi_r$ rovné nule, řidič s částečným úvazkem bude mít $\varphi_r$ nenulové.
- $\lambda_r^{tt}$ ... Doba řízení v posledním týdnu před začátkem plánovacího období, kladné reálné číslo. Zavádí se kvůli kontrole překročení maximální dvoutýdenní doby řízení v prvním týdnu aktualizačního období a předchozím týdnu.

- $\lambda_r^d$... Doba řízení po denním odpočinku za poslední den před plánovacím obdobím, kladné reálné číslo. Zavádí se kvůli kontrole překročení maximální denní doby řízení před prvním denním odpočinkem v plánovacím období.
- $\lambda_r^t$... Doba řízení po týdenním odpočinku za poslední týden před plánovacím obdobím, kladné reálné číslo. Zavádí se kvůli kontrole překročení maximální týdenní doby řízení před prvním týdenním odpočinkem v plánovacím období.

### 2.2.3 Jízdy

Celkový počet jízd budeme značit písmenem *J*, pro jednotlivé jízdy budeme používat index *j*. V případech, kdy bude nutné použít dva indexy, bude druhým indexem písmeno *i*. Jízdy budou popsány následujícími charakteristikami:

- $u_j$... Pořadí úkolu, který obsahuje jízdu *j*, přirozené číslo nabývající jedné z hodnot od *1* do *U* (*U*... celkový počet úkolů, $U \leq J$).
- $a_j$... Index alternativy úkolu v rámci úkolu $u_j$, přirozené číslo. Tato hodnota určuje, která alternativa úkolu $u_j$ obsahuje jízdu *j*. Hodnota $a_j$ se tedy bude shodovat u všech jízd, které jsou součástí stejné alternativy jednoho úkolu. Pro přehlednost se u každého úkolu alternativy číslují vždy sekvenčně počínaje *1*.
- $p_j$ ... Pořadí jízdy v alternativě $a_j$, přirozené číslo. Každá alternativa úkolu se skládá z jedné nebo několika jízd. Jízdy, které jsou součástí jedné alternativy, na sebe postupně navazují (čas konce jedné jízdy se vždy shoduje s časem začátku následující jízdy). Jednotlivé jízdy v alternativě jsou číslovány sekvenčně (od *1*) dle času začátku jízdy.
- $q_j$... Koeficient preference výběru alternativy $a_j$. Reálné číslo nabývající hodnoty z intervalu $\left(0,1\right\rangle$.[5] Ne všechny alternativy konkrétního úkolu musí být stejně výhodné, proto je možné zvýhodnit „lepší" alternativy vyšší hodnotou koeficientu preference. Kvůli věcné správnosti modelu je ovšem nutné již ve

---

[5] Nezahrnutí 0 do intervalu možných hodnot pro koeficient preference alternativy je čistě technické omezení – v dalších výpočtech se bude tímto členem dělit.

vstupních datech zajistit, aby všechny jízdy v jedné alternativě měly stejnou hodnotu tohoto koeficientu.

- $t_j^z$... Čas začátku jízdy, reálné číslo, jehož celá část představuje datum a desetinná část představuje čas.[6] Všechny jízdy, bez ohledu na jejich zařazení do jednotlivých úkolů a alternativ, budou seřazeny vzestupně podle časů jejich začátků, musí tedy platit: $t_j^z > t_i^z \Rightarrow j > i$. Toto seřazení bude využito pro zjednodušení matematického zápisu několika podmínek.
- $t_j^k$... Čas konce jízdy, reálné číslo, jehož celá část představuje datum a desetinná část představuje čas. Kvůli umožnění správného výpočtu týdenní doby řízení, je nutné aby každá jízda zasahovala pouze do jednoho týdne – začátek i konec jízdy musí spadat do jednoho týdne. Pokud nějaký úkol zasahuje do dvou (nebo více) týdnů, je potřeba jej rozdělit na více jízd, s tím, že časové předěly mezi těmito jízdami nastávají v pondělí v 0:00.
- $\tau_j^p$... Doba potřebná k příjezdu řidiče na místo začátku jízdy. Pokud se v rámci jedné alternativy uskuteční více jízd, je pravděpodobné, že k přechodu mezi jednotlivými jízdami nebude docházet v obvyklém stanovišti vozidel, ale na některém jiném místě. V případě, že při takovémto přechodu dojde ke změně řidiče, je nutné aby se nově nastupující řidič na takovéto místo dopravil. Hodnota $\tau_j^p$ odpovídá době, kterou zabere cesta řidiče na místo začátku jízdy. Zadaní nenulové hodnoty parametru $\tau_j^p$ může dopravce použít ještě v jedné specifické situaci. Tato situace nastane v případě jízd, kterým musí předcházet řidičův denní (týdenní odpočinek). Nastavením kladné hodnoty parametru $\tau_j^p$ zajistíme, že řidič nebude v příslušné době přidělen na jinou jízdu, ale vozidlo přiděleno být může.
- $\tau_j^o$... Doba potřebná k odjezdu řidiče z místa konce jízdy. Obdobné, jako předcházející veličina. Tuto hodnotu můžeme analogicky použít i v případě, že

[6] Hodnota „čas začátku jízdy" i další obdobné hodnoty budou v tomto modelu reprezentovány kladnými reálnými čísly, u nichž celá část odpovídá dnům (počtu dnů od nějakého pevně stanoveného dne, např. počátek prvního plánovacího období) a desetinná část čísla odpovídá určitému dennímu času (např. 0,25 odpovídá času 6:00).

požadujeme, aby po skončení jízdy následoval řidičův odpočinek a vozidlo bylo k dispozici k dalšímu použití.

- $S_j$... Odjetá vzdálenost, kladné reálné číslo. Vzdálenost v kilometrech, kterou je potřeba v rámci jízdy j absolvovat.
- $H_j$... Předpokládaná[7] doba řízení jednoho řidiče v rámci jízdy j, kladné reálné číslo. Do hodnoty $H_j$ bude započítána pouze doba, po kterou se řidič věnuje řízení vozidla. Pokud se jedná o jízdu se dvěma řidiči, vyplní se polovina celkové předpokládané doby řízení. Hodnota je udávaná v hodinách.
- $H_j^{ost}$... Předpokládaná doba ostatních prací jednoho řidiče v rámci jízdy j, kladné reálné číslo. Hodnota $H_j^{ost}$ představuje dobu, po kterou se řidič nevěnuje přímo řízení vozidla, ale jiným činnostem souvisejícím s řízením vozidla (např. údržba vozidla, nakládka vozidla nebo dohled nad vozidlem). Za tyto činnosti náleží řidiči mzda, ale nepočítají se do denních dob řízení. I v tomto případě platí pravidlo, že se u jízdy se dvěma řidiči vyplní polovina celkové předpokládané doby ostatních prací. Hodnota je udávaná v hodinách.
- $\sigma_j$... Podmínka zachování řidiče, binární číslo. Nabývá hodnoty 1 u jízdy j pokud je nutné, aby byl na jízdu, která po jízdě j těsně následuje v rámci jedné alternativy shodného úkolu, nasazen stejný řidič a hodnotu 0 v opačném případě. Tento parametr se zavádí zejména kvůli nutnosti rozdělit každou jízdu, která přesahuje konec týdne. Zavedením tohoto parametru zajistíme obsazení jízd, vzniklých rozdělením původní jízdy, stejným řidičem.
- $\chi_j^d$... Jízda kontrolovaná vůči dennímu odpočinku, binární číslo. Určuje, zda je u jízdy j nutné kontrolovat dodržování denních dob odpočinku. Nabývá hodnotu 1, pokud je nutné dodržování odpočinku kontrolovat a 0 v opačném případě. Dodržování dob odpočinku není nutné například u „jízd", při kterých nedojde k vlastnímu řízení vozidla, takovéto „jízdy" mohou odpovídat například údržbě vozidla, účasti na školení nebo plánované dovolené. Dodržování denních dob odpočinku se nekontroluje ani u vícedenních jízd. Vyvážený poměr mezi dobou

---

[7] Při vytváření plánu nelze přesně určit čas, který jízda zabere. Aby bylo možné plánování provést je nutné tento čas předem odhadnout, proto časy označujeme jako „předpokládané".

řízení a dobou odpočinku v rámci takovýchto jízd musí být již při zadávání jízdy určen tak, aby mohl řidič denní odpočinek čerpat v rámci průběhu času určeného na absolvování jízdy.

- $\chi_j^t$... Jízda kontrolovaná vůči týdennímu odpočinku, binární číslo. Obdobné, jako předchozí veličina.
- $L_j^{rid}$ ... Potřebný počet řidičů. Ve většině případů bude hodnota tohoto atributu rovna 1. Mohou se ovšem vyskytovat i jiné hodnoty - 0 (např. pro speciální „jízdu“, která odpovídá plánované údržbě vozidla), 2 (např. pro dálkovou jízdu, při které jedou dva řidiči a v řízení se střídají) i více (např. školení, na které musí jít právě pět řidičů)
- $L_j^{voz}$ ... Potřebný počet vozidel - může nabývat pouze hodnot 0 (např. pro speciální „jízdu“, která odpovídá plánovanému čerpání řidičovi dovolené) nebo 1 (standardní jízda). V případě, že bude v nějaké zakázce požadována současná jízda většího počtu vozidel, musí být naplánováno více úkolů (toto omezení je nutné kvůli přesně určenému přiřazení řidičů k jednotlivým vozidlům).

### 2.2.4 Řidič x Vozidlo

Ke každé kombinaci řidiče a vozidla musí být určeny následují charakteristiky:

- $C_{br}$ ... Přípustnost kombinace řidiče a vozidla, binární číslo. Nabývá hodnotu 1 v případě, že je řidič r oprávněn řídit vozidlo b a 0 v opačném případě.
- $c_{br}$ ... Koeficient preference kombinace řidiče a vozidla. Reálné číslo nabývající hodnotu z intervalu $\langle 0,1 \rangle$. Pokud je například řidič zvyklý řídit jedno konkrétní vozidlo, bude kombinaci tohoto řidiče a vozidla přiřazen koeficient preference 1. Kombinacím tohoto řidiče s ostatními vozidly shodného typu přiřadíme preference například v hodnotě *0,2* a kombinaci řidiče s ostatními vozidly preferenci v hodnotě *0*.
- $n_{br}$ ... Očekávaná nadspotřeba na ujetý kilometr, způsobená kombinací řidiče a vozidla. Reálné číslo nabývající kladné i záporné hodnoty, kladné bude v případě, že se podaří statisticky vysledovat, že má některý řidič r na vozidle b

průměrně větší spotřebu než ostatní řidiči. Pokud má spotřebu nižší než ostatní řidiči hodnota $n_{br}$ bude záporná.

### 2.2.5 Řidič x Jízda

Ke každé kombinaci řidiče a jízdy musí být určeny následují charakteristiky:

- $D_{rj}$ ... Přípustnost kombinace řidiče a jízdy, binární číslo. Nabývá hodnotu *1* v případě, kdy může být řidič *r* přiřazen na jízdu *j* a *0* v opačném případě.
- $d_{rj}$ ... Koeficient preference kombinace řidiče a jízdy. Reálné číslo nabývající hodnoty z intervalu $\langle 0,1 \rangle$. Nenulová hodnota může být použita například u řidiče, který zná trasu jízdy a ostatní řidiči tuto trasu neznají.

### 2.2.6 Jízda x Vozidlo

Každá kombinace jízdy a vozidla je popsána následujícími charakteristikami:

- $E_{bj}$ ... Přípustnost kombinace jízdy a vozidla, binární číslo. Nabývá hodnotu *1* v případě, kdy může být vozidlo *b* použito na jízdu *j* a *0* v opačném případě. Protože v rámci jedné alternativy úkolu nemůže docházet ke změně vozidla, je věcně správné již ve vstupních datech zajistit, aby hodnoty $E_{bj}$ byly pro všechny jízdy v jedné alternativě totožné.
- $e_{bj}$ ... Koeficient preference kombinace vozidla a jízdy. Reálné číslo nabývající hodnotu z intervalu $\langle 0,1 \rangle$.
- $N_{bj}$ ... Spotřeba vozidla na jízdě, kladné reálné číslo. Předpokládaný náklad za spotřebované palivo, které na jeden kilometr spotřebuje vozidlo *b* při jízdě *j*.

### 2.2.7 Přeplánování stávajícího plánu

Stávající plán je při přeplánování reprezentován dvěma maticemi.

- $X'_{rj}$ ... Původní obsazení jízd řidiči, binární číslo. Nabývá hodnotu *1* v případě, kdy je dle původního plánu určen řidič *r* pro jízdu *j* a hodnotu *0* v opačném případě. Nulová hodnota je použita i v případě, kdy jízda $j$ v původním plánu nefiguruje.

- $Y'_{bj}$... Původní obsazení jízd vozidly, binární číslo. Nabývá hodnotu *1* v případě, kdy je dle původního plánu určeno vozidlo $b$ pro jízdu $j$ a hodnotu *0* v opačném případě.

### 2.2.8 Mezní hodnoty zákonných omezení

Zákonná omezení jsou reprezentována těmito konstantami:

- $m^t$... Minimální týdenní odpočinek
- $m^d$... Minimální denní odpočinek
- $M^{tt}$... Maximální dvoutýdenní doba řízení
- $M^t$... Maximální týdenní doba řízení
- $M^d$... Maximální denní doba řízení

### 2.2.9 Plánovací období

Plánovací období je určeno dvěma hodnotami:

- $t^0$... Počátek plánovacího období, celé číslo představující datum[8]. Kvůli sledování týdenních dob řízení a dob odpočinků, které vychází ze zákonů a závazných předpisů platných v dopravě, zavádíme technické omezení: Je nutné aby plánovací období vždy začínalo na začátku kalendářního týdne (v pondělí v 0:00 hodin). V praxi nás tato podmínka příliš neomezí, pokud budeme potřebovat začít plánovat jízdy například od středy, přidáme do plánovacího období pondělí a úterý. Rozvrh na tyto dva dny byl patrně určen již při předchozím plánování, stačí tedy v aktuálním plánovacím období přiřadit pondělním a úterním jízdám fixní obsazení odpovídající plánu za minulé období.
- $W$... Celkový počet týdnů v plánovacím období, přirozené číslo.

---

[8] Jelikož plánovací období vždy začíná v čase 0:00, je desetinná složka čísla představující čas nulová.

## 2.3 Přípravné výpočty před vlastní optimalizací

Pro zjednodušení zápisů podmínek v optimalizační úloze si před začátkem vlastní optimalizace vypočítáme některé hodnoty. Takto vypočítané hodnoty budou poté figurovat v optimalizační úloze. Základem pro výpočet budou některá z výše popsaných vstupních dat.

- $Q$ ... Očekávaná průměrná odpracovaná doba připadající na jednoho řidiče za aktuální plánovací období.

  Hodnotu určíme pomocí následujícího vzorce:

$$Q = \frac{\sum_{r=1}^{R} \varphi_r + \sum_{j=1}^{J} \left(H_j + H_j^{ost}\right) \cdot L_j^{\check{r}id} \cdot v_j}{R} \tag{2.1}$$

  kde $v_j = q_j \Big/ \sum_{k=1}^{J} q_k \cdot I(u_j = u_k) \cdot I(p_k = 1)$ jsou váhy jednotlivých alternativ úkolů.

  Váhy jednotlivých alternativ jsou určeny pomocí preferencí daných alternativ úkolů.

  Hodnota preference alternativy $q_j$ by měla být shodná u všech jízd v jedné alternativě, proto je možné vzájemně porovnávat pouze hodnotu $q_j$ u prvních jízd z každé alternativy. Tyto váhy by pro výpočet neměly mít příliš velký význam, protože se dá předpokládat, že realizace každé z alternativ bude trvat přibližně stejnou dobu jako realizace ostatních alternativ stejného úkolu.

  Hodnota očekávané průměrné odpracované doby připadající na jednoho řidiče se spočítá jako součet předem fixovaných odpracovaných hodin všech řidičů a odhadu počtu hodin, které bude nutné k vykonání všech úkolů, dělený celkovým počtem řidičů.

- $G$ ... Očekávaná průměrná odjetá vzdálenost připadající na jedno vozidlo za aktuální plánovací období.

  Hodnotu určíme pomocí následujícího vzorce:

$$G = \frac{\sum_{b=1}^{B} \Omega_b + \sum_{j=1}^{J} S_j \cdot v_j}{B} \tag{2.2}$$

kde $v_j = q_j \Big/ \sum_{k=1}^{J} q_k \cdot I(u_j = u_k) \cdot I(p_k = 1)$ jsou váhy jednotlivých alternativ úkolů.

Hodnota očekávané průměrné vzdálenosti se, obdobně jako průměrná očekávaná odpracovaná doba, spočítá jako součet předem fixované vzdálenosti odjeté jednotlivými vozidly a odhadu celkové vzdálenosti, kterou bude nutné ujet pro vykonání všech úkolů, dělený celkovým počtem vozidel.

- $\Gamma_{ji}, j,i \in \{1..J\}$... Matice následných jízd v rámci téže alternativy jednoho úkolu, čtvercová matice s binárními prvky. $\Gamma_{ji}$ nabývá hodnotu 1 v případě, kdy jsou jízdy $j$ a $i$ po sobě jdoucí součásti shodné alternativy jednoho úkolu, zároveň s tím musí platit, že jízda $j$ bezprostředně předchází jízdě $i$. V opačném případě nabývá $\Gamma_{ji}$ hodnotu 0.

Hodnotu matice vypočteme pomocí následujícího vzorce:

$$\Gamma_{ji} = I(u_j = u_i) \cdot I(a_j = a_i) \cdot I(p_j + 1 = p_i) \qquad (2.3)$$

Hodnota 1 se vloží ke kombinacím jízd, které mají shodný úkol, shodnou alternativu a pořadí má jízda s indexem $j$ o jedna nižší než jízda s indexem $i$, k ostatním kombinacím jízd se vloží 0.

Hodnoty této matice budou použity v několika podmínkách i ve složkách účelové funkce. Tím, že vypočítáme hodnoty této matice před vlastní optimalizací, se zjednoduší zápis optimalizační úlohy i vlastní ověřování těchto podmínek v průběhu optimalizačního algoritmu.

## 2.4 Proměnné

Nyní uvedeme označení a popis významu všech proměnných, které figurují v matematickém modelu popisujícím naši optimalizační úlohu:

- $X_{rj}$ ... Obsazení jízdy řidičem, binární proměnná. Nabývá hodnoty 1 v případě, že je řidič r určen k absolvování jízdy j a hodnoty 0 v opačném případě.

- $Y_{bj}$ ... Obsazení jízdy vozidlem, binární proměnná. Nabývá hodnoty 1 v případě, že je vozidlo b určeno pro jízdu j a hodnoty 0 v opačném případě.

- $Z_j$... Realizace jízdy, binární proměnná. Nabývá hodnoty 1 v případě, že bude realizována alternativa úkolu, jejíž součástí je jízda j a hodnoty 0 v opačném případě.

- $o_{rw}^{w}$... Čas začátku týdenního odpočinku, reálná proměnná představující datum a čas. Hodnota proměnné odpovídá začátku naplánované doby týdenního odpočinku r-tého řidiče ve w-tém týdnu od počátku plánovacího období. w nabývá hodnot 1..W (počet týdnů v plánovacím období).

- $o_{rg}^{d}$... Čas začátku denního odpočinku, reálná proměnná představující datum a čas. Hodnota proměnné odpovídá začátku naplánované doby denního odpočinku r-tého řidiče v g-tém dnu od počátku plánovacího období. g nabývá hodnot 1..7W (počet dnů v plánovacím období).

- $X_{rj}^{d}$... Jízda před denním odpočinkem, binární proměnná. Tato proměnná nabývá hodnotu *1*, pokud jsou zároveň splněny následující podmínky: 1) řidič *r* je určen k absolvování jízdy *j*, 2) denní odpočinek řidiče *r* (za den shodný s dnem začátku jízdy j) začne až po ukončení jízdy j, 3) jízda j je „kontrolovaná" vůči dennímu odpočinku $\left(\chi_j^d = 1\right)$ a hodnoty *0* v opačném případě (jízda začíná až po denním odpočinku nebo není obsazena daným řidičem nebo se jedná o jízdu, u které není nutné kontrolovat přesah s denními odpočinky). Jedná se o pomocnou proměnnou, kterou zavádíme kvůli linearizaci podmínek kontrolujících, zda nedochází k časovému přesahu jízd a dob denních odpočinků. V případě nepoužití této proměnné by bylo nutné zavést kvadratické omezení. Přesný způsob použití této proměnné bude patrný z nadefinovaných podmínek 2.26 a 2.27 (viz níže).

- $X_{rj}^{t}$... Jízda před týdenním odpočinkem, binární proměnná. Tato proměnná nabývá hodnotu *1*, pokud jsou zároveň splněny následující podmínky: 1) řidič *r* je určen k absolvování jízdy j, 2) týdenní odpočinek řidiče *r* (za týden shodný s týdnem začátku jízdy $j$) začne až po ukončení jízdy $j$, 3) jízda $j$ je „kontrolovaná“ vůči týdennímu odpočinku $\left(\chi_j^t = 1\right)$ a hodnoty $0$ v opačném případě. Proměnná $X_{rj}^t$ se zavádí z obdobného důvodu jako proměnná $X_{rj}^d$.

## 2.5 Omezující podmínky v modelu

Již bylo uvedeno, že výsledný plán přidělení jednotlivých řidičů a vozidel jízdám musí splňovat větší množství podmínek. Následuje kompletní výčet těchto podmínek i s uvedením jejich matematického zápisu, v tomto zápisu budou použity výše definované proměnné i veličiny odpovídající vstupním datům.

- **Z každé alternativy úkolu se musí uskutečnit buď všechny jízdy nebo žádná.** Nebo-li hodnota proměnné $Z$ se pro všechny jízdy z téže alternativy téhož úkolu musí shodovat.

  *Matematický zápis podmínky:*

  $$\underset{j=1}{\overset{J-1}{\forall}}\ \underset{i=j+1}{\overset{J}{\forall}} : \Gamma_{ji} \cdot \left(Z_j - Z_i\right) = 0 \tag{2.4}$$

  Pro ověření této podmínky postačí zkontrolovat všechny po sobě jdoucí jízdy ve všech alternativách, k tomu nám poslouží předem určená matice následných jízd.

- **Pro každý úkol musí být uskutečněna právě jedna alternativa.**

  *Matematický zápis podmínky:*

  $$\forall j \in \left\{ j,\ a_j = 1 \wedge p_j = 1 \right\} : \sum_{i=1}^{J} Z_i \cdot I\left(p_i = 1\right) \cdot I\left(u_i = u_j\right) = 1 \tag{2.5}$$

  Tuto podmínku není nutné prověřovat pro všechny jízdy, postačí ověření pro všechny úkoly. Každý úkol je ve výběru reprezentován první jízdou své první alternativy. Pro každý úkol se ověří podmínka, že se z prvních jízd všech alternativ tohoto úkolu uskuteční právě jedna jízda.

- **Pro všechny jízdy, z kterých se skládá jedna alternativa konkrétního úkolu, musí být použito stejné vozidlo.**

*Matematický zápis podmínky:*

$$\underset{b=1}{\overset{B}{\forall}}\underset{j=1}{\overset{J-1}{\forall}}\underset{i=j+1}{\overset{J}{\forall}} : \Gamma_{ji} \cdot \left(Y_{bj} - Y_{bi}\right) = 0 \tag{2.6}$$

I v tomto případě postačí ověření pro jízdy, které po sobě v rámci stejné alternativy jednoho úkolu těsně následují, k tomu využijeme matici následných jízd.

- **Ověření splnění „podmínky zachování řidiče".**

  *Matematický zápis podmínky:*

$$\underset{r=1}{\overset{R}{\forall}}\underset{j=1}{\overset{J-1}{\forall}}\underset{i=j+1}{\overset{J}{\forall}} : \Gamma_{ji} \cdot \sigma_j \cdot \left(X_{rj} - X_{ri}\right) = 0 \tag{2.7}$$

  Na rozdíl od předcházející podmínky, kde musela být vozidla zachována u všech jízd v jedné alternativě, v tomto případě postačí zachování řidičů pouze u jízd, pro které platí $\sigma_j = 1$.

- **Pro každou jízdu musí být použit správný počet vozidel.**

  *Matematický zápis podmínky:*

$$\underset{j=1}{\overset{J}{\forall}} : \sum_{b=1}^{B} Y_{bj} = L_j^{voz} \cdot Z_j \tag{2.8}$$

  V případě, kdy se jízda $j$ uskuteční, musí platit, že počet vozidel použitých na jízdu $j$ odpovídá počtu potřebných vozidel $\left(L_j^{voz}\right)$. Pro neuskutečněnou jízdu musí být počet použitých vozidel nulový.

- **Pro každou jízdu musí být použit správný počet řidičů.**

  *Matematický zápis podmínky:*

$$\underset{j=1}{\overset{J}{\forall}} : \sum_{r=1}^{R} X_{rj} = L_j^{rid} \cdot Z_j \tag{2.9}$$

  V případě, kdy se jízda $j$ uskuteční, musí platit, že počet řidičů přiřazených na jízdu $j$ odpovídá počtu potřebných řidičů $\left(L_j^{rid}\right)$. Pro neuskutečněnou jízdu musí být počet přidělených řidičů nulový.

- **Každá naplánovaná kombinace řidiče a jízdy musí být přípustná.**

*Matematický zápis podmínky:*

$$\underset{j=1}{\overset{J}{\forall}}\underset{r=1}{\overset{R}{\forall}} : X_{rj} \leq D_{rj} \qquad (2.10)$$

- **Každá naplánovaná kombinace vozidla a jízdy musí být přípustná.**

  *Matematický zápis podmínky:*

$$\underset{j=1}{\overset{J}{\forall}}\underset{b=1}{\overset{B}{\forall}} : Y_{bj} \leq E_{bj} \qquad (2.11)$$

- **Každá naplánovaná kombinace řidiče a vozidla musí být přípustná.** Tato podmínka musí být splněna u všech jízd.

  *Matematický zápis podmínky:*

$$\underset{j=1}{\overset{J}{\forall}}\underset{r=1}{\overset{R}{\forall}}\underset{b=1}{\overset{B}{\forall}} : X_{rj}.Y_{bj} \leq C_{rb} \qquad (2.12)$$

Tato podmínka je kvadratická. Dá se ovšem využít skutečnosti, že obě proměnné, zahrnuté v podmínce, jsou binární. Díky tomu můžeme kvadratickou podmínku nahradit lineární podmínkou ve tvaru:

$$\underset{j=1}{\overset{J}{\forall}}\underset{r=1}{\overset{R}{\forall}}\underset{b=1}{\overset{B}{\forall}} : X_{rj} \leq (C_{rb} - 1) \cdot Y_{bj} + 1 \qquad (2.13)$$

která je s podmínkou 2.12 ekvivalentní. Ekvivalence obou podmínek je doložena tabulkou všech možných hodnot.

| X | Y | C | podmínka 2.12 | podmínka 2.13 |
|---|---|---|---|---|
| 0 | 0 | 0 | platí | platí |
| 1 | 0 | 0 | platí | platí |
| 0 | 1 | 0 | platí | platí |
| 1 | 1 | 0 | neplatí | neplatí |
| 0 | 0 | 1 | platí | platí |
| 1 | 0 | 1 | platí | platí |
| 0 | 1 | 1 | platí | platí |
| 1 | 1 | 1 | platí | platí |

Tabulka č. 1: Porovnání logických výrazů.

- **Žádné vozidlo nemůže být v jednom momentu použito na více než jednu jízdu.**

  *Matematický zápis podmínky:*

$$\underset{b=1}{\overset{B}{\forall}} \forall (j,i) \in JJ^{voz} : Y_{bj} + Y_{bi} \leq 1 \tag{2.14}$$

kde $JJ^{voz} = \{(j,i) \in JxJ, j < J \wedge j < i \wedge u_j \neq u_i \wedge t_i^z < t_j^k\}$ je množina uspořádaných dvojic jízd, na kterých je nutné provést kontrolu pro vozidla. V této množině jsou zahrnuty pouze jízdy z různých úkolů, které se alespoň částečně časově překrývají. Dvě jízdy ze stejného úkolu není nutné porovnávat. Pokud jsou dvě jízdy součástí jednoho úkolu mohou nastat dvě možnosti. Buď jsou jízdy součástí různých alternativ, potom je zřejmé, že se může uskutečnit maximálně jedna z těchto dvou jízd. Nebo jsou jízdy součástí jedné alternativy, jízdy v jedné alternativě na sobě časově navazují, proto se také nemohou překrývat.

- **Žádný řidič nemůže být v jednom momentu přiřazen na více než jednu jízdu.**

  *Matematický zápis podmínky:*

$$\underset{r=1}{\overset{R}{\forall}} \forall (j,i) \in JJ^{rid} : X_{rj} + X_{ri} \leq 1 \tag{2.15}$$

kde $JJ^{rid} = \{(j,i) \in JxJ, j < J \wedge j < i \wedge u_j \neq u_i \wedge t_i^z - \tau_i^p < t_j^k + \tau_j^o\}$ je množina uspořádaných dvojic jízd, na kterých je nutné provést kontrolu pro řidiče.

Pro řidiče platí obdobná podmínka jako pro vozidla, pouze s tím rozdílem, že u řidičů je nutné k času začátku (resp. konce) jízdy upravit o dobu potřebnou k dopravení řidiče na místo začátku jízdy (resp. odjezdu z místa konce jízdy).

- **Nepřekročení maximální dvoutýdenní doby řízení.** V průběhu každých dvou po sobě jdoucích týdnů nesmí řidič překročit maximální povolenou dobu řízení.

  *Matematický zápis podmínky:*

$$\underset{r=1}{\overset{R}{\forall}} \underset{w=1}{\overset{W-1}{\forall}} : \sum_{j \in \Pi_w^{tt}} X_{rj} \cdot H_j \leq M^{tt} \tag{2.16}$$

kde $\Pi_w^{tt} = \{j,\ 7w - 7 \leq t_j^z - t^0 < 7w + 7\}$ je množina všech jízd, které začínají (i končí) ve w-tém nebo w+1. týdnu plánovacího období.

- **Nepřekročení maximální dvoutýdenní doby řízení v prvním týdnu.** Součet doby řízení v posledním týdnu předcházejícím před začátkem aktuálního plánovacího období a v prvním týdnu aktuálního plánovacího období nesmí překročit povolenou hodnotu.

*Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} : \sum_{j \in \Pi_1^t} X_{rj} \cdot H_j + \lambda_r^{tt} \leq M^{tt} \tag{2.17}$$

kde $\Pi_1^t = \left\{ j, \left\lceil \left( t_j^z - t^o \right) / 7 \right\rceil = 1 \right\}$ je množina všech jízd, které začínají (i končí) v prvním týdnu plánovacího období.

- **Nepřekročení maximální dvoutýdenní doby řízení v posledním týdnu.** V průběhu posledního týdne plánovacího období nesmí řidič překročit 60% maximální dvoutýdenní povolené doby řízení.

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} : \sum_{j \in \Pi_W^t} X_{rj} \cdot H_j \leq 0{,}6 \cdot M^{tt} \tag{2.18}$$

kde $\Pi_W^t = \left\{ j, \left\lceil \left( t_j^z - t^o \right) / 7 \right\rceil = W \right\}$ je množina všech jízd, které začínají (i končí) v posledním týdnu plánovacího období.

- **Hodnota proměnné začátek týdenního odpočinku $\left( o_{rw}^w \right)$ musí připadnout vždy do odpovídajícího týdne.**

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \overset{W}{\underset{w=1}{\forall}} : 7w - 7 \leq o_{rw}^w - t^0 < 7w \tag{2.19}$$

- **Hodnota proměnné začátek denního odpočinku $\left( o_{rg}^d \right)$ musí připadnout vždy do odpovídajícího dne.**

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \overset{7W}{\underset{g=1}{\forall}} : g - 1 \leq o_{rg}^d - t^0 < g \tag{2.20}$$

- **Jednotlivé týdenní doby odpočinku se nesmí překrývat.**

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \overset{W-1}{\underset{w=1}{\forall}} : o_{rw}^w + m^t \leq o_{rw+1}^w \tag{2.21}$$

Podmínku stačí ověřit pouze pro dvojice po sobě jdoucích týdnů.

- **Jednotlivé denní doby odpočinku se nesmí překrývat.** [9]

  *Matematický zápis podmínky:*

$$\underset{r=1}{\overset{R}{\forall}}\ \underset{g=1}{\overset{7W-1}{\forall}} : o_{rg}^{d} + m^{d} \leq o_{rg+1}^{d} \tag{2.22}$$

  Podmínku stačí ověřit pouze pro dvojice po sobě jdoucích dnů.

Jednotlivé denní doby odpočinku se nesmí překrývat s jízdami „kontrolovanými vůči dennímu odpočinku". Tento požadavek se dá splnit dvěma, vzájemně se vylučujícími, způsoby. Buď musí platit, že konec jízdy předchází začátku denního odpočinku nebo musí odpočinek skončit již před začátkem jízdy. Požadovanou podmínku tedy získáme složením dvou jednodušších podmínek, mezi které vložíme logickou spojku nebo, což v přímém přepisu odpovídá kvadratické podmínce. Naším cílem ovšem je vyhnout se této kvadratické podmínce a nahradit ji několika lineárními podmínkami, k tomu nám dopomůže pomocná proměnná "Jízda před denním odpočinkem" $\left(X_{rj}^{d}\right)$.

Kontrolu dodržení této podmínky je nutné provést pro všechny jízdy „kontrolované vůči dennímu odpočinku" $\left(\chi_{j}^{d} = 1\right)$. U každé takovéto jízdy se musíme přesvědčit, zda časově nekoliduje s denním odpočinkem za den těsně předcházející dnu začátku jízdy, s denním odpočinkem za den začátku jízdy a v případě, že jízda zasahuje do dvou dnů musíme zkontrolovat i přesah s denním odpočinkem za den konce jízdy. (Jízdy zasahující do více než dvou dnů nemohou patřit mezi jízdy „kontrolované vůči dennímu odpočinku".) Kvůli dodržení požadavku je tedy nutné ověřit následující čtyři podmínky:

- **Denní odpočinek, za den předcházející dni začátku jízdy, musí skončit před začátkem jízdy.**

  *Matematický zápis podmínky:*

$$\underset{r=1}{\overset{R}{\forall}} \forall j \in \left\{j, \chi_{j}^{d} = 1 \wedge t_{j}^{z} - t^{0} \geq 1\right\} : t_{j}^{z} + \left(1 - X_{rj}\right) \cdot 10W \geq o_{rz_{j}-1}^{d} + m^{d} \tag{2.23}$$

---

[9] Denní a týdenní doby odpočinku se vzájemně překrývat mohou.

kde $z_j = \lceil t_j^z \rceil - t^0$ odpovídá pořadí dne, ve kterém jízda začíná.

Jízdy začínající první den do této podmínky nevstupují. Člen $(1 - X_{rj}) \cdot 10W$ zajistí splnění podmínky v případě, kdy řidič není pro jízdu vybrán.

- **Denní odpočinek, za den následující po dni začátku jízdy, musí začínat po skončení jízdy.**

*Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \forall j \in \left\{ j, \chi_j^d = 1, \lceil t_j^z \rceil \neq \lceil t_j^k \rceil \right\} : t_j^k + (X_{rj} - 1) \cdot 10W \leq o_{rk_j}^d \quad (2.24)$$

kde $k_j = \lceil t_j^k \rceil - t^0$ odpovídá pořadí dne, ve kterém jízda končí.

Do této podmínky vstupují pouze „kontrolované“ jízdy, které zasahují do dvou dnů. Člen $(X_{rj} - 1) \cdot 10W$ zajistí splnění podmínky v případě, kdy řidič není pro jízdu vybrán.

- **Jízda, ze shodného dne, začínající před denním odpočinkem, musí před denním odpočinkem i skončit.**

*Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \forall j \in \left\{ j, \chi_j^d = 1 \right\} : t_j^k + (X_{rj}^d - 1) \cdot 10W \leq o_{rz_j}^d \quad (2.25)$$

kde $z_j = \lceil t_j^z \rceil - t^0$ odpovídá pořadí dne, ve kterém jízda začíná.

Člen $(X_{rj}^d - 1) \cdot 10W$ zajistí splnění podmínky v případě, kdy řidič *r* není pro jízdu vybrán nebo jízda nezačíná před denním odpočinkem řidiče *r*.

- **Jízda, ze shodného dne, začínající po denním odpočinku.**

*Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \forall j \in \left\{ j, \chi_j^d = 1 \right\} : t_j^z + \left( X_{rj}^d - X_{rj} + 1 \right) \cdot 10W \geq o_{rz_j}^d + m^d \quad (2.26)$$

kde $z_j = \lceil t_j^z \rceil - t^0$ odpovídá pořadí dne, ve kterém jízda začíná.

Člen $X_{rj}^d \cdot 10W$ zajistí splnění podmínky v případě, kdy jízda proběhla před začátkem denního odpočinku a člen $(1 - X_{rj}) \cdot 10W$ zajistí splnění podmínky v případě, kdy řidič není pro jízdu vybrán. Ze zápisu této a předchozí podmínky je vidět, že pomocná proměnná "Jízda před denním odpočinkem" $\left(X_{rj}^{d}\right)$ plní roli jakéhosi přepínače, který rozhodne, která z těchto dvou podmínek se má vyhodnocovat a zároveň zajistí splnění druhé podmínky. Proměnná $X_{rj}^{d}$ nefiguruje v žádné další podmínce, ke správnému určení její hodnoty dojde v průběhu optimalizačního výpočtu. V případě, že by hodnota proměnné nebyla správně vyplněna, nebude splněna alespoň jedna z těchto dvou podmínek.

Obdobný požadavek, jako u denních odpočinků, musí platit u týdenních odpočinků. Ani týdenní doby odpočinku se nesmí překrývat s jízdami „kontrolovanými vůči týdennímu odpočinku" $\left(\chi_j^t = 1\right)$. Postup vyhodnocení bude obdobný jako u denních odpočinků. Jediný rozdíl spočívá ve vynechání podmínky kontrolují přesah jízdy s týdenním odpočinkem za týden následující po týdnu začátku jízdy. Vynechání je možné díky zavedenému omezení, které nedovolí, aby jízda zasahovala do více týdnů.

- **Týdenní odpočinek, za týden předcházející týdnu, ve kterém jízda proběhne, musí skončit před začátkem jízdy.**

  *Matematický zápis podmínky:*

  $$\underset{r=1}{\overset{R}{\forall}} \forall j \in \left\{j, \chi_j^t = 1 \wedge t_j^z \geq 7\right\}: t_j^z + \left(1 - X_{rj}\right) \cdot 10W \geq o_{rw_j - 1}^t + m^t \tag{2.27}$$

  kde $w_j = \left\lceil \left(t_j^z - t^o\right)/7 \right\rceil$ odpovídá pořadí týdne, ve kterém jízda začíná.

  Jízdy začínající v prvním týdnu do této podmínky nevstupují. Člen $\left(1 - X_{rj}\right) \cdot 10W$ zajistí splnění podmínky v případě, kdy řidič není pro jízdu vybrán.

- **Jízda, ze shodného týdne, začínající před týdenním odpočinkem, musí před týdenním odpočinkem i skončit.**

  *Matematický zápis podmínky:*

  $$\underset{r=1}{\overset{R}{\forall}} \forall j \in \left\{j, \chi_j^t = 1\right\}: t_j^k + (X_{rj}^t - 1) \cdot 10W \leq o_{rw_j}^t \tag{2.28}$$

kde $w_j = \lceil (t_j^z - t^0)/7 \rceil$ odpovídá pořadí týdne, ve kterém jízda proběhne.

Člen $(X_{rj}^t - 1) \cdot 10W$ zajistí splnění podmínky v případě, kdy řidič není pro jízdu vybrán nebo jízda nezačíná před týdenním odpočinkem.

- **Jízda, ze shodného týdne, začínající po týdenním odpočinku.**

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \forall j \in \{j, \chi_j^t = 1\} : t_j^z + (X_{rj}^t - X_{rj} + 1) \cdot 10W \geq o_{rw_j}^t + m^t \qquad (2.29)$$

kde $w_j = \lceil (t_j^z - t^0)/7 \rceil$ odpovídá pořadí týdne, ve kterém jízda začíná.

Člen $X_{rj}^t \cdot 10W$ zajistí splnění podmínky v případě, kdy jízda proběhne před začátkem týdenního odpočinku a člen $(1 - X_{rj}) \cdot 10W$ zajistí splnění podmínky v případě, kdy řidič není pro jízdu vybrán.

- **Denní doba řízení nesmí překročit dovolenou hodnotu.** Denní doba řízení se určí jako součet dob řízení mezi jednotlivými denními odpočinky.

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \overset{7W-1}{\underset{g=1}{\forall}} : \left\{ \sum_{j \in \Pi_g^d} H_j \cdot \chi_j^d \cdot (X_{rj} - X_{rj}^d) + \sum_{j \in \Pi_{g+1}^d} H_j \cdot \chi_j^d \cdot X_{rj}^d \right\} \leq M^d \qquad (2.30)$$

kde $\Pi_g^d = \{j, \lceil t_j^z \rceil - t^0 = g\}$ je množina jízd začínajících v den $g$.

První člen odpovídá součtu dob řízení z „kontrolovaných“ jízd, které byly řidiči přiděleny a začínají v den *g* po denním odpočinku za den *g*, do druhého členu jsou zahrnuty doby řízení z jízd, které začínají v den *g+1* a končí před denním odpočinkem dne *g+1*.

- **Doba řízení mezi posledním denním odpočinkem před plánovacím obdobím a prvním denním odpočinkem v plánovacím období nesmí překročit povolenou denní dobu řízení.**

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} : \lambda_r^d + \sum_{j \in \Pi_1^d} H_j \cdot \chi_j^d \cdot X_{rj}^d \leq M^d \qquad (2.31)$$

kde $\Pi_1^d = \{j, \lceil t_j^z \rceil - t^0 = 1\}$ je množina jízd začínajících během prvního dne plánovacího období.

- **Doba řízení mezi koncem denního odpočinku za poslední den plánovacího období a koncem posledního dne nesmí překročit povolenou denní dobu řízení.**

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} : \sum_{j \in \Pi_{7W}^d} H_j \cdot \chi_j^d \cdot \left(X_{rj} - X_{rj}^d\right) \leq M^d \tag{2.32}$$

kde $\Pi_{7W}^d = \{j, \lceil t_j^z \rceil - t^0 = 7W\}$ je množina jízd začínajících v posledním dnu plánovacího období.

- **Týdenní doba řízení nesmí překročit dovolenou hodnotu.** Obdobně jako je tomu u denní doby řízení, bude i týdenní doba řízení určena jako součet dob řízení mezi jednotlivými týdenními odpočinky.

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} \overset{W-1}{\underset{w=1}{\forall}} : \left\{ \sum_{j \in \Pi_w^t} H_j \cdot \chi_j^t \cdot \left(X_{rj} - X_{rj}^t\right) + \sum_{j \in \Pi_{w+1}^t} H_j \cdot \chi_j^t \cdot X_{rj}^t \right\} \leq M^t \tag{2.33}$$

kde $\Pi_w^t = \{j, \lceil (t_j^z - t^0)/7 \rceil = w\}$ je množina jízd začínajících ve w-tém týdnu.

Podmínka je analogická jako u kontroly délky denních dob řízení.

- **Doba řízení mezi posledním týdenním odpočinkem před plánovacím obdobím a prvním týdenním odpočinkem v plánovacím období nesmí překročit povolenou týdenní dobu řízení.**[10]

  *Matematický zápis podmínky:*

$$\overset{R}{\underset{r=1}{\forall}} : \lambda_r^t + \sum_{j \in \Pi_1^t} H_j \cdot \chi_j^t \cdot X_{rj}^t \leq M^t \tag{2.34}$$

---

[10] Kontrola nepřekročení maximální týdenní doby řízení v posledním týdnu není nutná, protože týdenní odpočinek trvá minimálně dva dny a maximální týdenní doba řízení odpovídá šesti denním dobám řízení.

kde $\Pi_1^t = \{j, \lceil (t_j^z - t^o)/7 \rceil = 1\}$ je množina jízd začínajících během prvního týdne plánovacího období.

## 2.6 Účelová funkce

Cílem této úlohy bude, za dodržení výše uvedených podmínek, minimalizovat váženou sumu **dílčích účelových funkcí**. Dílčí účelové funkce budeme pro přehlednost značit $\Phi_1$ až $\Phi_{12}$. Ne všechny z dvanácti dílčích optimalizačních parametrů je nutné minimalizovat, některé parametry je naopak nutné maximalizovat (jedná se například o součet preferencí vybraných kombinací řidičů a vozidel). Maximalizaci takovýchto parametrů zajistíme tak, že do odpovídajících dílčích účelových funkcí vložíme opačné hodnoty těchto parametrů. Díky tomu bude možné všechny dílčí účelové funkce minimalizovat. Nyní uvedeme výčet všech dílčích účelových funkcí:

- **Maximalizace preferencí výběru řidiče vzhledem k vozidlu.**

$$\Phi_1 = (-1) \cdot \sum_{b=1}^{B} \sum_{j=1}^{J} \sum_{r=1}^{R} \left( X_{jr} \cdot Y_{bj} \cdot c_{br} \cdot S_j \right) \tag{2.35}$$

Jedná se o součet preferencí naplánovaných kombinací vozidel a řidičů přes všechny jízdy, vážený celkovou délkou jízdy. Celá suma je vynásobena *-1*, protože se jedná o optimalizační parametr, který se snažíme maximalizovat.

- **Minimalizace očekávané nadspotřeby paliva způsobené kombinací řidičů a vozidel.**

$$\Phi_2 = \sum_{b=1}^{B} \sum_{j=1}^{J} \sum_{r=1}^{R} \left( X_{jr} \cdot Y_{bj} \cdot n_{br} \cdot S_j \right) \tag{2.36}$$

- **Maximalizace preference výběru vozidla vzhledem k jízdě.**

$$\Phi_3 = (-1) \cdot \sum_{b=1}^{B} \sum_{j=1}^{J} \left( Y_{bj} \cdot e_{bj} \cdot S_j \right) \tag{2.37}$$

Jedná se o součet preferencí naplánovaných kombinací vozidel a jízd, vážený celkovou délkou jízdy. Celá suma je vynásobena *-1*, protože se jedná o optimalizační parametr, který se snažíme maximalizovat.

- **Minimalizace očekávané spotřeby paliva**, připadající na kombinaci jízdy a vozidla.

$$\Phi_4 = \sum_{b=1}^{B} \sum_{j=1}^{J} \left( Y_{bj} \cdot N_{bj} \cdot S_j \right) \tag{2.38}$$

- **Minimalizace počtu střídání řidičů v rámci téže alternativy jednoho úkolu.**

$$\Phi_5 = 0,5. \sum_{r=1}^{R} \sum_{j=1}^{J-1} \sum_{i=j+1}^{J} \left( \Gamma_{ji} \cdot \left| X_{rj} - X_{ri} \right| \right) \tag{2.39}$$

V této funkci opět používáme pomocnou matici následných jízd. U každého řidiče procházíme všechny jízdy, pro které existuje následná jízda ze stejné alternativy téhož úkolu, pokud je řidič na jednu z těchto dvou jízd nasazen a na druhou ne, zvyšujeme součet o *1*. Každá změna řidičů se v celkovém součtu projeví dvakrát, proto je součet vynásoben konstantou *0,5*.

- **Minimalizace překročení průměrné odpracované doby u řidičů.**

$$\Phi_6 = \sum_{r=1}^{R} \rho_r^{+} \cdot \left[ \sum_{j=1}^{J} X_{rj} \cdot \left( H_j + H_j^{ost} \right) + \varphi_r - Q \right]^{+} \tag{2.40}$$

Přes všechny řidiče sčítáme kladné části rozdílu mezi dobou odpracovanou řidičem a průměrnou odpracovanou dobou vynásobené koeficientem penalizace za překročení průměrné odpracované doby.

- **Minimalizace nedosažení průměrné odpracované doby u řidičů.**

$$\Phi_7 = \sum_{r=1}^{R} \rho_r^{-} \cdot \left[ \sum_{j=1}^{J} X_{rj} \cdot \left( H_j + H_j^{ost} \right) + \varphi_r - Q \right]^{-} \tag{2.41}$$

Přes všechny řidiče sčítáme záporné části rozdílu mezi dobou odpracovanou řidičem a průměrnou odpracovanou dobou vynásobené koeficientem penalizace za nedosažení průměrné odpracované doby.

- **Minimalizace překročení průměrné ujeté vzdálenosti u jednotlivých**

**vozidel.**

$$\Phi_8 = \sum_{b=1}^{B} \beta_b^+ \cdot \left[ \sum_{j=1}^{J} Y_{bj} \cdot S_j + \Omega_b - G \right]^+ \quad (2.42)$$

Přes všechny vozidla sčítáme kladné části rozdílu mezi vzdáleností naplánovanou pro vozidlo a průměrnou vzdáleností vynásobené koeficientem penalizace za překročení průměrné vzdálenosti.

- **Minimalizace nedosažení průměrné ujeté vzdálenosti u jednotlivých vozidel.**

$$\Phi_9 = \sum_{b=1}^{B} \beta_b^- \cdot \left[ \sum_{j=1}^{J} Y_{bj} \cdot S_j + \Omega_b - G \right]^- \quad (2.43)$$

Přes všechny vozidla sčítáme záporné části rozdílu mezi vzdáleností naplánovanou pro vozidlo a průměrnou vzdáleností vynásobené koeficientem penalizace za nedosažení průměrné vzdálenosti.

- **Minimalizace počtu změn obsazení jízd řidiči oproti původnímu plánu**, při přeplánování rozvrhu.

$$\Phi_{10} = \sum_{(j,r) \in \Lambda^r} \left(1 - X_{jr}\right) \quad (2.44)$$

kde $\Lambda^r = \left\{(j,r), X'_{jr} = 1\right\}$ je množina všech přiřazení řidičů na jízdy dle původního plánu.

- **Minimalizace počtu změn přiřazení vozidel na jízdy oproti původnímu plánu**, při přeplánování rozvrhu.

$$\Phi_{11} = \sum_{(j,b) \in \Lambda^b} \left(1 - Y_{jb}\right) \quad (2.45)$$

kde $\Lambda^b = \left\{(j,b), Y'_{jb} = 1\right\}$ je množina všech přiřazení vozidel na jízdy dle původního plánu.

- **Maximalizace součtu koeficientů preference uskutečněných alternativ.**

$$\Phi_{12} = (-1) \cdot \sum_{j \in S} \left(Z_j . q_j\right) \quad (2.46)$$

kde $S = \{j, p_j = 1\}$ je množina všech jízd, které jsou první ve své alternativě.

Do součtu je zahrnut koeficient preference každé uskutečněné alternativy právě jednou, proto sčítáme koeficienty pouze přes uskutečněné první jízdy z každé alternativy. Celá suma je vynásobena *-1*, protože se jedná o optimalizační parametr, který se snažíme maximalizovat.

Jednotlivé váhy u složek účelové funkce budou posledním vstupním souborem, který bude vstupovat do optimalizační úlohy, tyto váhy budeme značit $\Theta_1$ až $\Theta_{12}$. Celková účelová funkce optimalizační úlohy bude vypadat následovně:

$$min \sum_{k=1}^{12} \Theta_k \cdot \Phi_k \tag{2.47}$$

# 3. Možností řešení celočíselných optimalizačních úloh

Tímto jsme definovali celý optimalizační model. Je vidět, že se jedná o poměrně složitou optimalizační úlohu, ve které figurují diskrétní (konkrétněji binární) i spojité proměnné. Jedná se tedy úlohu smíšeného celočíselného programování. Řešení celočíselných úloh, někdy nazývané diskrétní programování, je obecně komplikovanější než řešení úloh, ve kterých figurují jen spojité proměnné. Zatímco při řešení běžných úloh se využívá konvexnosti a souvislosti množiny přípustných řešení, např. pohyb od jednoho krajního bodu polyedrické množiny ke druhému nebo posun ve směru gradientu v okolí daného bodu. U úloh diskrétního programování bývá oblast přípustných řešení nekonvexní a nesouvislá, proto standardní metody selhávají a je nutné použít speciální postupy.

Jednou z možností jak řešit úlohy s konečnou množinou přípustných řešení, což je i náš případ, je prostý propočet hodnoty účelové funkce u všech přípustných řešení. Tento postup je ovšem pro větší úlohy příliš časově náročný.

Mezi často používané metody řešení celočíselných úloh patří postup, kdy se nejprve řeší úloha bez omezení na celočíselnost proměnných. Po vyřešení takovéto úlohy se vypočítané hodnoty proměnných, které mají být celočíselné, zaokrouhlí. Tato metoda se používá například při určování akciových portfolií, protože i počet akcií je celočíselná hodnota. S touto metodou je ovšem nutné zacházet velice opatrně, protože ne vždy vede zaokrouhlení k optimálnímu výsledku a obecně se dá říci, že pro úlohy s binomickými proměnnými, což je náš případ, tato metoda není vhodná.

## 3.1 Metoda sečných nadrovin

Další skupina metod řešení celočíselných úloh vychází z metody popsané v předchozím odstavci, také se nejdříve vyřeší úloha bez celočíselných omezení. Pokud je získané řešení takto regularizované úlohy celočíselné, je vyřešena i původní úloha. Pokud řešení celočíselné není, přidáme k původní úloze dodatečné lineární omezení, které splňuje dvě základní vlastnosti:

1. Získané neceločíselné řešení nesplňuje dodatečné lineární omezení.

2. Libovolné celočíselné přípustné řešení vyhovuje dodatečnému lineárnímu omezení.

V dalším kroku řešíme úlohu rozšířenou o dodatečné lineární omezení a celý proces opakujeme dokud nezískáme celočíselné řešení, takovéto řešení je optimální. Přidání každého dodatečného lineárního omezení odpovídá zavedení nadroviny, která oddělí od polyedru přípustných řešení regularizované úlohy starý neceločíselný optimální bod, přičemž se všechny celočíselné body zachovají. Proto se takovéto metody nazývají metody sečné nadroviny nebo metody řezů. Jednotlivé metody spadající do této skupiny se od sebe odlišují způsobem konstrukce dodatečných omezení. Takovéto konstrukce musí splňovat dvě základní vlastnosti:

1. Je požadován důkaz konečnosti procesu řezů. Při neuváženém zavádění sečných nadrovin nemusí vést proces k úspěšnému konci - nalezení optimálního řešení.
2. Je důležité zamezit příliš velkému růstu rozsahu úlohy přidáváním dodatečných omezení.

Konkrétní příklady algoritmů založených na metodě sečných nadrovin je možné nalézt například ve [4,6].

## 3.2 Metoda větvení a mezí

Jiná skupina metod řešení celočíselných úloh je založena na skutečnosti, že množina všech přípustných řešení celočíselných úloh bývá konečná. V algoritmu se poté procházejí jednotlivá přípustná řešení, která se postupně vyhodnocují. Cílem algoritmu ovšem není projít všechna přípustná řešení, ale naopak vyčlenit množiny přípustných řešení, ve kterých nemůže být obsaženo optimální řešení. Takovéto vyčleňování dokáže podstatně zkrátit dobu výpočtu. Konečnost takovýchto algoritmů je dána konečností množiny všech přípustných řešení a skutečností, že každé z přípustných řešení se vyhodnocuje maximálně jednou.

Jedním z těchto algoritmů je metoda větvení a mezí, známá pod zkratkou B&B (z anglického Branch and Bound). Přiblížíme zde podobu algoritmu pro úlohu s binomickými proměnnými, která odpovídá našim potřebám. (Omezené celočíselné proměnné se vždy dají převést na příslušný počet binomických proměnných.)

Metodu bychom mohli popsat následujícím způsobem. Předpokládejme, že *INCUMBENT* je hodnota účelové funkce nejlepšího doposud nalezeného celočíselného přípustného řešení úlohy. Vytvoří se rozpadový strom, který na začátku tvoří jeden uzel představující původní úlohu. Tento uzel zůstane kořenem stromu. Každému uzlu se přiřadí optimální hodnota *LP* úlohy s uvolněním celočíselných omezení. Tato hodnota je zároveň horní mezí pro řešení celočíselné úlohy. Jednotlivé uzly se budou dále dělit pomocí následujícího algoritmu:

1. Vybere se nějaký list stromu (tj. doposud nerozdělený uzel). Tento list představuje problém, který označíme jako *CP*.
2. Vybere se některá z binomických proměnných, označme ji *x*. Následně se *CP* se rozdělí na dvě nové úlohy $NP_1$ a $NP_2$ doplněním vzájemně se vylučujících podmínek: $x=1$ a $x=0$. Tyto nové úlohy přidávají nové listy do stromu a uzel CP přestává být listem.
3. Pro list $NP_1$ se vyřeší uvolněná $LP_1$ úloha. Pokud je úloha $LP_1$ neřešitelná nebo je hodnota jejího optimálního řešení horší než *INCUMBENT*, list $NP_1$ se odstraní. V případě, že má úloha $LP_1$ celočíselné řešení, lepší než dosavadní *INCUMBENT*, aktualizuje se hodnota *INCUMBENT* i v tomto případě se list $NP_1$ odstraní. Stejná procedura se vykoná pro uzel $NP_2$. Jestliže se odstraní oba uzly $NP_1$ a $NP_2$, pak se může vyloučit i uzel *CP*.

Tento cyklus se opakuje tak dlouho, dokud strom není prázdný nebo není splněna dodatečná podmínka:

$$(\nu_{LP} - INCUMBENT)/(1 + |\nu_{LP}|) \leq T \qquad (3.1)$$

kde $\nu_{LP}$ je hodnota optimálního řešení uvolněné úlohy a *T* je předem stanovená toleranční mez.

Můžeme si všimnout, že naše optimalizační úloha není ryze lineární – všechny omezující podmínky sice lineární jsou, ale v některých dílčích účelových funkcích se vyskytují nelineární prvky (absolutní hodnota, kladná a záporná část a kvadratický člen). Tomuto problému se budeme podrobněji věnovat při vlastní realizaci programu.

# 4. Realizace v systému GAMS

K praktické realizaci optimalizačního výpočtu nám poslouží modelovací systém GAMS (General Algebraic Modeling Systém). V němž budeme používat programový balík (neboli řešič) s názvem CPLEX. Systém GAMS je speciální programovací jazyk vyšší úrovně, kterým lze formulovat matematické modely pomocí názorných algebraických výrazů. Díky tomu může GAMS využívat široké spektrum uživatelů. Systém GAMS byl navržen tak, aby umožňoval zápis matematického modelu nezávisle na řešícím algoritmu. GAMS dále umožňuje jednoduše modifikovat dříve vytvořené modely. Podrobnější popis systému GAMS je možné nalézt v [1-3].

CPLEX používá pro řešení celočíselných úloh výše popsaný algoritmus větvení a mezí. Oproti popsanému algoritmu má metoda, kterou požívá GAMS několik vylepšení, které slouží především ke zrychlení výpočtu. V obecném algoritmu se při větvení uzlů používá libovolná proměnná, zatímco v postupu, který využívá CPLEX je uzel volen s ohledem na výsledek uvolněné úlohy.

## 4.1 Popis zdrojového kódu v jazyku GAMS

Program v jazyku GAMS, který byl sestaven pro řešení našeho modelu, se skládá z několika částí. Na začátku programového kódu jsou deklarovány **indexové množiny**, například množina všech řidičů nebo dnů v plánovacím období. Tyto množiny se v programu používají k indexování výčtu jednotlivých parametrů, proměnných a rovnic. K deklarování těchto indexových množin se používá příkaz *„SET“*.

Poté jsou deklarovány jednotlivé parametry odpovídající výše popsaným vstupním souborům. Vesměs se jedná o jedno a dvojrozměrné vektory, z nichž každý popisuje určitou vstupní oblast, například vlastnosti řidičů, vozidel nebo kombinací řidičů a jízd. V tomto případě se používá příkaz *„PARAMETER“*.

Do takto deklarovaných parametrů jsou zároveň nahrávány hodnoty ze vstupních souborů. Aby nemusely být vstupní hodnoty zadávané přímo v GAMSovském programovém kódu, což by bylo v praxi nepoužitelné, je program sestaven tak, aby použil externí vstupní data. V našem případě jsme vstupní data uspořádali do několika tabulek, které společně tvoří jednoduchou databázi. Pro naši úlohu postačí tabulky vytvořené v aplikaci MS Excel, které jsou zároveň výhodné pro svoji přehlednost. Každá z tabulek je vždy na jednom listu excelovského sešitu, který je pojmenován *„VSTUPNI_SOUBORY.xls“*. Data z tohoto excelovského sešitu jsou pomocí speciální procedury jazyka GAMS nahrána do odpovídajících parametrů.

Při nahrávání dat je nutné vyřešit ještě jeden problém. Tento problém spočívá v předem nejasné velikosti vstupních souborů – před nahráním konkrétní tabulky totiž není zřejmé, kolik bude mít tato tabulka řádků, ovšem při další práci s daty už je nutné tento počet znát. Řešení tohoto problému je následující: indexové množiny, které odpovídají počtu řádků jednotlivých tabulek jsou na počátku deklarovány v maximální možné velikosti, odpovídající maximálnímu možnému počtu řádků jednotlivých vstupních tabulek. Po nahrání vstupních excelovských tabulek, je vždy definována podmnožina původní indexové množiny, která svojí velikostí odpovídá počtu neprázdných řádků vstupní excelovské tabulky. V GAMSovském programu se poté pro indexování používají takto získané podmnožiny.

V další části programu dochází k několika pomocným výpočtům, například určení hodnot pomocné „matice následných jízd“, výpočtu průměrné pracovní doby, připadající na jednoho řidiče nebo průměrné ujeté vzdálenosti přepadající na jedno vozidlo.

Poté následuje deklarování proměnných, které figurují v modelu. Stejně, jako výše popsané parametry jsou i proměnné indexovány pomocí indexových množin. GAMS umožňuje použití několika datových typů proměnných, v našem případě používáme pouze proměnné dvou typů a to kladné reálné proměnné a binární proměnné. V systému GAMS je nutné vždy zadat i jednu specifickou proměnnou, která odpovídá hodnotě účelové funkce. K deklarování proměnných se používá příkaz *„VARIABLE“*.

V systému GAMS jsou proměnné reprezentovány několika specifickými charakteristikami. Tyto charakteristiky používá systém GAMS při vlastním optimalizačním výpočtu. Charakteristiky se zároveň dají použít při zadávání modelu, což nám zjednoduší práci a vlastní model se stává přehlednější.

Z charakteristik uvádíme čtyři základní:

- Dolní mez (v GAMSovském kódu značená pomocí přípony „*.LO*“ za jménem příslušné proměnné, dolní mez proměnné *X* označíme tedy *X.LO*) udává dolní mez, kterou proměnná během výpočtu nepřesáhne.
- Horní mez (značená příponou „*.UP*“), obdobné, jako dolní mez.
- Fixní hodnota (značená příponou „*.FX*“), je vyplněna v případě, že proměnná může nabývat pouze jedinou hodnotu neboli horní a dolní mez se navzájem rovnají.
- Hodnota proměnné (značená příponou „*.L*“), určuje vlastní hodnotu proměnné během výpočtu a při úspěšném ukončení výpočtu je tato hodnota optimální pro danou proměnnou.

Této struktury proměnných je v našem programu využito k zadání některých omezení, které jsou v popisu modelu zařazeny mezi ostatní podmínky. Jedná se například o podmínku kontrolující, zda začátek konkrétního denního odpočinku spadá do odpovídajícího dne, toto je zajištěno nastavením odpovídajících horních a dolních mezí u proměnných. K dalším omezením, která jsou nahrazena takovouto inicializací proměnných patří všechny omezení na nepřípustnost některých kombinací (řidič x jízda, vozidlo x jízda a řidič x vozidlo). U všech nepřípustných kombinací je fixní hodnota odpovídajících proměnných nastavena na *0*. Tímto se z příslušných proměnných stávají konstanty a dochází tedy ke zmenšení rozsahu úlohy, což může být důležité zvláště při velkém počtu takovýchto omezení.

Poté v programovém kódu následuje sekce, ve které jsou definovány jednotlivé omezující podmínky, odpovídající popisu optimalizačního modelu. V jazyku GAMS je nutné zadávat podmínky ve dvou krocích. V prvním kroku je vždy uveden výčet podmínek, včetně indexových množin, přes které je nutné podmínky ověřovat. Součástí výčtu může být i slovní popis jednotlivých podmínek. V druhém kroku je poté zapsán vlastní algebraický zápis podmínek. I mezi podmínky je nutné přidat jednu speciální podmínku a tou je algebraický zápis účelové funkce. Tento zápis se musí rovnat proměnné, odpovídající hodnotě účelové funkce (viz výše). K zadání podmínek se používá příkaz „*EQUATIONS*“.

V dalším kroku jsou nastaveny některé parametry ovlivňující vlastní optimalizační výpočet, mezi tyto parametry patří například limit určující maximální počet iterací nebo limit na maximální možnou dobu optimalizačního výpočtu. Pokud by v průběhu výpočtu došlo k překročení některé z těchto hodnot, byl by výpočet ukončen bez nalezení optimální hodnoty.

Pomocí klíčového slova *„MODEL“* je určeno, které rovnice a nerovnice jsou do modelu zahrnuty, v našem případě používáme volbu *„ALL“*, což značí, že jsou do modelu zahrnuty všechny výše definované rovnice a nerovnice. Vlastní spuštění výpočtu se provádí příkazem *„SOLVE“*, součástí tohoto příkazu je i určení, jakým způsobem má být model řešen, v našem případě je použita volba *„MIP“* – smíšené celočíselné programování.

Po výpočtové části programu následuje poslední sekce a tou je export zjištěného „optimálního“ plánu a hodnoty účelové funkce do připraveného excelovského sešitu.

Tištěná verze, právě popsaného, programového kódu je k této práci, elektronickou verzi je možné nalézt na přiloženém médiu, pod názvem „optimální_prideleni.gms“. Na přiloženém médiu je dále možné nalézt několik vzorových vstupních excelovských souborů, ke každému z nich je přiřazeno i optimální rozvrh, opět ve formě excelovského sešitu.

## 4.2 Výstupní soubor systému GAMS

Systém GAMS generuje při každém spuštění výpočtu výstupní textový soubor. Tento soubor informuje o průběhu a výsledcích výpočtu. Rozsah tohoto souboru je ovšem příliš velký, proto je celá verze tohoto souboru k práci přiložena pouze v elektronické formě. Podstatné části výstupního souboru uvedeme a popíšeme v této kapitole. Tyto části výstupního souboru budou vytištěny odlišným typem písma a od zbývajícího textu odděleny ohraničením.

V první část výstupního souboru je obsažena kopie spuštěného GAMSovského programového kódu s očíslovanými řádky. Na tato čísla řádků jsou v dalším textu výstupního souboru uváděny odkazy usnadňující orientaci.

V další části výstupního souboru nalezneme výpis omezení modelu. Omezení jsou zapsána ve formě, kterou GAMS používá pro vlastní výpočet. Neznámé na levé straně, konstanta na pravé. Pokud by výstupní soubor obsahoval všechna omezení z modelu, mohl by být příliš dlouhý. Proto se dá již v programovém kódu nastavit, kolik rovnic se má z každé skupiny zobrazení objevit ve výstupním souboru.

```
---- JEDNA_VARIANTA =E= pro kazdy ukol bude realizovana prave jedna varianta
JEDNA_VARIANTA(J1).. Z(J1) + Z(J2) =E= 1 ; (LHS = 0, INFES = 1 ***)
REMAINING 8 ENTRIES SKIPPED
```

Uvedený příklad je přepisem podmínky 2.5 pro první a druhou jízdu. Řetězcem „=*E*=" je v GAMSu značena rovnost. Informace v závorce slouží k představě o hodnotě rovnice před spuštěním výpočtu. Hodnota *LHS* udává hodnotu levé strany a tři hvězdičky značí, že počáteční nastavení není přípustné, přičemž hodnota *INFES* udává, o kolik je omezení překročeno.

Poté následuje výpis proměnných obsažených v modelu. Stejně, jako omezení se nevypisují všechny proměnné, ale jejich počet se dá omezit.

```
---- Z realizace jizdy
Z(J1)
        (.LO, .L, .UP = 0, 0, 1)
     1    STEJNA_VARIANTA(J1,J3)
     1    JEDNA_VARIANTA(J1)
    -1    POC_BUSY(J1)
    -1    POC_RIDICU(J1)
   0.2    UCEL_FCE
REMAINING 27 ENTRIES SKIPPED
```

Uvedený příklad popisuje proměnnou $Z_1$ z výše popsaného matematického modelu. Text informuje o počátečním nastavení (horní, dolní mez a inicializační hodnota). Dále je vypsán seznam rovnic, ve kterých se proměnná vyskytuje. Čísla před rovnicemi udávají koeficienty, se kterými se proměnná v příslušných rovnicích vyskytuje.

Dále nalezneme informaci o velikosti modelu – počtu rovnic a proměnných, o době překladu programového kódu o době běhu programu, před spuštěním vlastní optimalizace.

```
MODEL STATISTICS

BLOCKS OF EQUATIONS        29     SINGLE EQUATIONS      4,572
BLOCKS OF VARIABLES        15     SINGLE VARIABLES      2,689
NON ZERO ELEMENTS      15,633     DISCRETE VARIABLES    1,527

GENERATION TIME    =    0.547 SECONDS     61 Mb  WIN220-143 Jul 27, 2005
EXECUTION TIME     =   12.032 SECONDS     61 Mb  WIN220-143 Jul 27, 2005
```

Poté následuje blok informací o vlastním optimalizačním výpočtu a výsledné optimální hodnotě.

```
              S O L V E      S U M M A R Y

     MODEL   PRIDELENI              OBJECTIVE  HODNOTA_UC_FCE
     TYPE    MIP                    DIRECTION  MINIMIZE
     SOLVER  CPLEX                  FROM LINE  539

**** SOLVER STATUS                  1 NORMAL COMPLETION
**** MODEL STATUS                   8 INTEGER SOLUTION
**** OBJECTIVE VALUE                -10456.7081

 RESOURCE USAGE, LIMIT                234.140      100000.000
 ITERATION COUNT, LIMIT            221588          1000000

GAMS/Cplex    Aug  1, 2005 WIN.CP.CP 22.0 029.032.041.VIS For Cplex 9.1
Cplex 9.1.2, GAMS Link 29
Cplex licensed for 1 use of lp, qp, mip and barrier, with 2 parallel threads.

Solution satisfies tolerances.

MIP Solution:        -10456.708112         (221588 iterations, 2202 nodes)
Final Solve:         -10456.708112         (0 iterations)

Best possible:       -13586.581415
Absolute gap:          3129.873303
Relative gap:             0.299317
```

Z uvedených informací se dá vyčíst, že výpočet proběhl úspěšně. Vlastní běh algoritmu trval *234* sekund a bylo nutné provést *221 588* iterací. Hodnota účelové funkce výsledného řešení je *-10 456,7081.* Poslední část této sekce výstupního souboru informuje o hodnotě účelové funkce v regularizované úloze (totožná úloha, bez omezení na celočíselnost).

Poslední část výstupního souboru nám ukazuje konečný stav rovnic (nerovnic). Konečný stav je vypsán pro každou rovnici obsaženou v modelu. Pro názornost uvádíme i v tomto případě rovnici odpovídající podmínce 2.5.

```
---- EQU JEDNA_VARIANTA  pro kazdy ukol bude realizovana prave jedna varianta
       LOWER        LEVEL          UPPER          MARGINAL
J1     1.000        1.000          1.000             .
```

Hodnoty *LOWER* a *UPPER* určují povolené rozmezí levých stran rovnice, hodnota *LEVEL* určuje výslednou hodnotu. Hodnota MARGINAL obsahuje hodnotu duální proměnné k této rovnici. (Symbol „." v GAMSu odpovídá nulové hodnotě.) Význam duální proměnné není v celočíselném programování zdaleka tak důležitý, jako v lineárním programování.

## 4.3 Linearizace účelové funkce

Jak již bylo uvedeno výše, účelová funkce v základním optimalizačním modelu není lineární, což nám doposud nečinilo žádné obtíže. Řešič CPLEX je uzpůsoben i na řešení nelineárních úloh. Ovšem praktickým otestováním bylo zjištěno, že zavedení nelineárních prvků podstatně prodlužuje dobu výpočtu. Proto byly nelineární prvky v základní účelové funkci nahrazeny lineárními. Toto nahrazení bylo možné uskutečnit pouze za cenu podstatného zvýšení počtu proměnných, které v optimalizační úloze figurují. Přidání nových proměnných s sebou nese i přidání dalších podmínek, které nově přidané proměnné určují. Ale i takto rozsáhlou úlohu vyřeší CPLEX podstatně rychleji než původní úlohu s nelineární účelovou funkcí. Od této chvíle budeme tedy v systému GAMS řešit úlohu lineárního programování.

Nyní uvedeme přehled nelineárních dílčích účelových funkcí, u každé z nich uvedeme způsob nahrazení původní funkce novou lineární funkcí:

- **Funkce $\Phi_1$ a $\Phi_2$**

  *Původní tvar funkcí:*

$$\Phi_1 = (-1) \cdot \sum_{b=1}^{B} \sum_{j=1}^{J} \sum_{r=1}^{R} \left( X_{jr} \cdot Y_{bj} \cdot c_{br} \cdot S_j \right) \tag{2.35}$$

$$\Phi_2 = \sum_{b=1}^{B} \sum_{j=1}^{J} \sum_{r=1}^{R} \left( X_{jr} \cdot Y_{bj} \cdot n_{br} \cdot S_j \right) \tag{2.36}$$

Vidíme, že se v zápisu obou funkcí vyskytují kvadratické členy $X_{jr} \cdot Y_{bj}$. Tento násobek nahradíme nově zavedenou proměnnou, která určí, zda je na jízdu současně přidělen konkrétní řidič i vozidlo. Zároveň bude nutné do modelu přidat další podmínky, které zajistí doplnění odpovídajících hodnot do „nové“ proměnné.

*Nově zaváděná proměnná:*

$Z_{rbj}^{XY}$ ... **Současné obsazení jízdy řidičem i vozidlem,** binární proměnná.

Nabývá hodnoty *1* v případě, že je pro jízdu *j* současně přidělen řidič *r* a vozidlo *b* a hodnoty *0* v opačném případě.

*Upravený tvar funkcí:*

$$\Phi_1 = (-1) \cdot \sum_{b=1}^{B} \sum_{j=1}^{J} \sum_{r=1}^{R} \left( Z_{rbj}^{XY} \cdot c_{br} \cdot S_j \right) \tag{4.1}$$

$$\Phi_2 = \sum_{b=1}^{B} \sum_{j=1}^{J} \sum_{r=1}^{R} \left( Z_{rbj}^{XY} \cdot n_{br} \cdot S_j \right) \tag{4.2}$$

*Nově zaváděné podmínky:*

$$\underset{j=1}{\overset{J}{\forall}} \underset{r=1}{\overset{R}{\forall}} \underset{b=1}{\overset{B}{\forall}} : Z_{rbj}^{XY} \leq 0{,}5 \cdot \left( X_{rj} + Y_{bj} \right) \tag{4.3}$$

$$\underset{j=1}{\overset{J}{\forall}} \underset{r=1}{\overset{R}{\forall}} \underset{b=1}{\overset{B}{\forall}} : Z_{rbj}^{XY} \geq X_{rj} + Y_{bj} - 1 \tag{4.4}$$

Nově zaváděné podmínky zajistí, aby hodnota proměnné $Z_{rbj}^{XY}$ odpovídala hodnotám, kterých nabývají proměnné $X_{jr}$ a $Y_{bj}$ ($Z_{rbj}^{XY}$ se musí rovnat součinu $X_{jr} \cdot Y_{bj}$). Podmínka 4.3 zajistí nulovou hodnotu v proměnné $Z_{rbj}^{XY}$, pokud je hodnota alespoň jedné z proměnných $X_{jr}$, $Y_{bj}$ nulová. Podmínka 4.4 zajistí hodnotu *1* v proměnné $Z_{rbj}^{XY}$, v momentě, kdy jsou proměnné $X_{jr}$ a $Y_{bj}$ nenulové.

Na první pohled by se mohlo zdát, že je možné nově zavedenou proměnnou $Z_{rbj}^{XY}$ v celém modelu nahradit proměnné $X_{jr}$ a $Y_{bj}$. S tím, že by se každý výskyt těchto proměnných nahradil odpovídající sumou přes volný index proměnné $Z_{rbj}^{XY}$, čímž by došlo ke snížení počtu proměnných i podmínek. Tento postup by ovšem výrazně zkomplikován existencí speciálních jízd, u kterých není použit řidič (případně vozidlo). U těchto jízd jsou tudíž všechny hodnoty $Z_{rbj}^{XY}$ nulové. Proto raději ponecháme zbývající část modelu v původním tvaru.

- **Funkce $\Phi_5$**

*Původní tvar funkce:*

$$\Phi_5 = 0{,}5.\sum_{r=1}^{R}\sum_{j=1}^{J-1}\sum_{i=j+1}^{J}\left(\Gamma_{ji}\cdot\left|X_{rj} - X_{ri}\right|\right) \tag{2.39}$$

Nelinearitu v této funkci způsobuje absolutní hodnota. Tuto absolutní hodnotu nahradíme zavedením dalších dvou nových proměnných, z nichž jedna bude odpovídat kladné části výrazu $X_{rj} - X_{ri}$ a druhá záporné části výrazu $X_{rj} - X_{ri}$.

*Nově zaváděné proměnné:*

$X_{rj}^{odch}$... **Odchod původního řidiče po jízdě *j*,** binární proměnná. Nabývá hodnoty *1* v případě, že je na jízdu *j* přidělen řidič *r* a zároveň existuje jízda *i*, která těsně následuje po jízdě *j* v téže alternativě shodného úkolu a na jízdu *i* řidič přidělen není a hodnoty *0* v opačném případě.

$X_{rj}^{nast}$... **Nastoupeni nového řidiče po jízdě *j*,** binární proměnná. Nabývá hodnoty *1* v případě, že není na jízdu *j* přidělen řidič *r* a zároveň existuje jízda *i*, která těsně následuje po jízdě *j* v téže alternativě shodného úkolu a na jízdu *i* řidič přidělen je a hodnoty *0* v opačném případě.

*Upravený tvar funkce:*

$$\Phi_5 = 0{,}5\cdot\sum_{r=1}^{R}\sum_{j=1}^{J-1}\left(X_{rj}^{odch} + X_{rj}^{nast}\right) \tag{4.5}$$

*Nově zaváděné podmínky:*

$$\forall_{r=1}^{R}\forall_{j=1}^{J-1}: X_{rj}^{odch} - X_{rj}^{nast} = \sum_{i=j+1}^{J}\Gamma_{ji}\cdot\left(X_{rj} - X_{ri}\right) \tag{4.6}$$

Další omezení na nově přidané proměnné již nejsou nutná. Tím, že jsou kladné násobky „nových" proměnných zahrnuty v minimalizační účelové funkci, bude vždy alespoň jedna z odpovídajícího si páru proměnných nulová.

- **Funkce $\Phi_6$ a $\Phi_7$**

*Původní tvar funkcí:*

$$\Phi_6 = \sum_{r=1}^{R} \rho_r^+ \cdot \left[ \sum_{j=1}^{J} X_{rj} \cdot \left( H_j + H_j^{ost} \right) + \varphi_r - Q \right]^+ \tag{2.40}$$

$$\Phi_7 = \sum_{r=1}^{R} \rho_r^- \cdot \left[ \sum_{j=1}^{J} X_{rj} \cdot \left( H_j + H_j^{ost} \right) + \varphi_r - Q \right]^- \tag{2.41}$$

V těchto funkcích jsou nelineárními členy kladná a záporná část, což je obdobný problém jako v předchozím případě. Obdobné bude tedy i jeho řešení – i v tomto případě zavedeme nové proměnné odpovídající kladné a záporné části vyhodnocovaného výrazu.

*Nově zaváděné proměnné:*

$X_r^{prekr}$... **Překročení průměrné odpracované doby u řidiče *r*,** nezáporná reálná proměnná.

$X_r^{nedos}$... **Nedosažení průměrné odpracované doby u řidiče *r*,** nezáporná reálná proměnná.

*Upravený tvar funkcí:*

$$\Phi_6 = \sum_{r=1}^{R} X_r^{prekr} \cdot \rho_r^+ \tag{4.7}$$

$$\Phi_7 = \sum_{r=1}^{R} X_r^{nedos} \cdot \rho_r^- \tag{4.8}$$

*Nově zaváděné podmínky:*

$$\underset{r=1}{\overset{R}{\forall}} : X_r^{prekr} - X_r^{nedos} = \sum_{j=1}^{J} X_{rj} \cdot \left( H_j + H_j^{ost} \right) + \varphi_r - Q \tag{4.9}$$

- **Funkce $\Phi_8$ a $\Phi_9$**

*Původní tvar funkcí:*

$$\Phi_8 = \sum_{b=1}^{B} \beta_b^+ \cdot \left[ \sum_{j=1}^{J} Y_{bj} \cdot S_j + \Omega_b - G \right]^+ \tag{2.42}$$

$$\Phi_9 = \sum_{b=1}^{B} \beta_b^- \cdot \left[ \sum_{j=1}^{J} Y_{bj} \cdot S_j + \Omega_b - G \right]^- \tag{2.43}$$

Tento případ je analogický s předchozím, proto bude analogický i postup. Jediný rozdíl spočívá v tom, že v předchozím případě se vyhodnocovalo správné vyvážení pracovní doby řidičů, v tomto případě se jedná o ujetou vzdálenost připadající na vozidlo.

*Nově zaváděné proměnné:*

$Y_b^{prekr}$ ... **Překročení průměrné ujeté vzdálenosti u vozidla *b*,** nezáporná reálná proměnná.

$Y_b^{nedos}$ ... **Nedosažení průměrné ujeté vzdálenosti u vozidla *b*,** nezáporná reálná proměnná.

*Upravený tvar funkcí:*

$$\Phi_8 = \sum_{b=1}^{B} Y_b^{prekr} \cdot \beta_b^+ \tag{4.10}$$

$$\Phi_9 = \sum_{b=1}^{B} Y_b^{nedos} \cdot \beta_b^- \tag{4.11}$$

*Nově zaváděné podmínky:*

$$\forall_{b=1}^{B} : Y_b^{prekr} - Y_b^{nedos} = \sum_{j=1}^{J} Y_{bj} \cdot S_j + \Omega_b - G \tag{4.12}$$

## 4.4 Ovlivňování průběhu výpočtu

Systém GAMS umožňuje uživateli v některých ohledech ovlivňovat průběh výpočtu. Takovéto ovlivňování se provádí pomocí nastavování hodnot několika řídících parametrů, nebo-li voleb.[11] Mezi takovéto parametry patří např. limit na maximální počet iterací („IterLim") nebo limit na dobu výpočtu („ResLim"). Pokud je v průběhu výpočtu překročen některý z těchto limitů, je výpočet zastaven. Dále je možné například nastavit maximální možnou hodnotu účelové funkce („Cutoff"), při takovémto nastavení jsou všechna řešení s vyšší (tzn. horší) hodnotou účelové funkce odstraněna.

---

[11] Kompletní výčet parametrů, ovlivňujících průběh výpočtu, je možné nalézt v manuálu [1].

Hlavní parametr, který se používá k ovlivňování výpočtu celočíselných úloh je v GAMSu označen názvem *OptCR* (Relative optimality criterion). Tento parametr určuje s jakou relativní přesností hledá GAMS řešení. Hodnota *OptCR* odpovídá hodnotě *T* z nerovnosti 3.1 GAMS v průběhu výpočtu kontroluje, zda hodnota *INCUMBENT* (dosud nejlepší nalezené celočíselné řešení) nedosáhlo minimální tolerované hodnoty, dané volbou *OptCR*. Pokud *INCUMBENT* tuto toleranci splňuje je výpočet ukončen a tento *INCUMBENT* je považován za výsledné řešení. Pokud bychom chtěli získat skutečně optimální řešení, museli bychom nastavit parametr *OptCR* na *0*. Vystavujeme se ovšem nebezpečí, že v tomto případě bude výpočet příliš dlouhý. Což je doloženo i v tabulce č. 2. Tato tabulka ukazuje výsledky optimalizačního výpočtu jedné optimalizační úlohy (rozsah úlohy byl zhruba 100 jízd) v závislosti na různých hodnotách parametru *OptCR*. Z tabulky je jasně patrné, že relativně „dobré“ řešení je nalezeno poměrně rychle, ale určení skutečného optima je časově náročné. Takovéto ovlivňování průběhu výpočtu je využitelné i v praxi – v momentě, kdy dopravce vytváří plán s dostatečným předstihem, může nastavit nízkou hodnotu parametru *OptCR a* tím hledat skutečně optimální řešení. Zatímco v situaci, kdy je nutné plán určit rychle, nastaví dopravce vyšší hodnotu *OptCR*.

| hodnota OptCR | hodnota účelové funkce výsledného řešení | doba výpočtu (počet sekund) | počet iterací |
|---|---|---|---|
| 0,9 | -148 533 | 46 | 27 937 |
| 0,8 | -152 817 | 74 | 35 145 |
| 0,7 | -176 050 | 78 | 35 351 |
| 0,6 | -176 050 | 80 | 35 351 |
| 0,5 | -209 361 | 91 | 35 351 |
| 0,4 | -209 361 | 91 | 35 351 |
| 0,3 | -209 361 | 94 | 35 351 |
| 0,2 | -229 573 | 269 | 43 143 |
| 0,1 | -240 235 | 2 542 | 777 016 |
| 0,01 | -243 602 | 40 179 | 13 413 405 |

Tabulka č. 2: Závislost hodnoty účelové funkce a doby výpočtu na parametru OtCR.

# Závěr

V této práci jsme sestavili lineární model, který dostatečným způsobem postihuje problém optimálního určení rozvrhu přiřazení řidičů a vozidel na jízdy. Na základě tohoto modelu byl sestaven algoritmus v optimalizačním programu GAMS. Tento algoritmus je schopný pro zadanou množinu vstupních dat určit vhodný rozvrh obsazení jednotlivých jízd. Dále jsme zjistili, že nalezení optimálního řešení trvá i na výkonném osobním počítači poměrně dlouhý časový úsek. Naproti tomu přípustné řešení, které se svojí „kvalitou" optimálnímu řešení velice blíží, jsme schopni nalézt v akceptovatelném čase.

Součástí práce je i pět vzorových úloh. U každé z úloh uvádíme zadání (ve formě excelovského sešitu se vstupními daty), určený optimální rozvrh (taktéž ve formě excelovského sešitu s výstupními daty) a výstupní textový soubor z programu GAMS, informující o průběhu výpočtového algoritmu. Všechny tyto soubory jsou umístěny na přiloženém médiu.

# Použitá literatura


[1] Brooke A., Kendrick D., Meeraus A., Raman R. (1998): GAMS A USER'S GUIDE.
GAMS Development Corporation, Washington, DC.

[2] Fryšová D., Hantych P., Charazma P., Kadláček R., Klincar M., Mydlo M., Pavlicová M., Popela P., Tlustý P., Večeř J. (1993): Modelovací systém GAMS.
MFF UK, Praha.

[3] Fryšová D., Pavlicová M. (1994): Celočíselné úlohy řešené systémem GAMS.
MFF UK, Praha.

[4] Korbut A. A., Finkeľštejn J. J. (1972): Diskrétne programovanie.
Alfa, Bratislava.

[5] Machačka I. (2004): Práce osádek a tachografy v Evropské unii.
SYSTEMCONSULT, Pardubice.

[6] Nemhauser G. L., Rinnooy Kan A.H.G., Todd M. J (1989): Optimization.
Elsevier Science Publisher B. V., North-Holland.

# Přílohy:

## A. *Tištěné přílohy*

- Programový kód v jazyku GAMS, řešící plánování obsazení jízd řidiči a vozidly.

## B. *Elektronické přílohy*

- Soubor: „*optimální_prideleni.gms*“, programový kód v jazyku GAMS.
- Soubor: „*vstupni_soubory_pr1.xls*“. Příklad 1 – exelovský vstupní soubor.
- Soubor: „vystupni_soubory_pr1.xls“. Příklad 1 – exelovský výstupní soubor.
- Soubor: „optimální_prideleni_pr1.lst“. Příklad 1 – textový výstupní soubor systému GAMS.
- Soubor: „vstupni_soubory_pr2.xls“. Příklad 2 – exelovský vstupní soubor.
- Soubor: „vystupni_soubory_pr2.xls“. Příklad 2 – exelovský výstupní soubor.
- Soubor: „optimální_prideleni_pr2.lst“. Příklad 2 – textový výstupní soubor systému GAMS.
- Soubor: „vstupni_soubory_pr3.xls“. Příklad 3 – exelovský vstupní soubor.
- Soubor: „vystupni_soubory_pr3.xls“. Příklad 3 – exelovský výstupní soubor.
- Soubor: „optimální_prideleni_pr3.lst“. Příklad 3 – textový výstupní soubor systému GAMS.
- Soubor: „vstupni_soubory_pr4.xls“. Příklad 4 – exelovský vstupní soubor.
- Soubor: „vystupni_soubory_pr4.xls“. Příklad 4 – exelovský výstupní soubor.
- Soubor: „optimální_prideleni_pr4.lst“. Příklad 4 – textový výstupní soubor systému GAMS.
- Soubor: „vstupni_soubory_pr5.xls“. Příklad 5 – exelovský vstupní soubor.
- Soubor: „vystupni_soubory_pr5.xls“. Příklad 5 – exelovský výstupní soubor.
- Soubor: „optimální_prideleni_pr5.lst“. Příklad 5 – textový výstupní soubor systému GAMS.

```
$TITLE MODEL OPTIMALNIHO OBSAZENI RIDICU A AUTOBUSU

* Cilem tohoto modelu je urcit optimalni obsazeni autobusovych jizd jednotlivymi ridici a autobusy.


*     DEFINOVANI MNOZIN A NACTENI HODNOT VSTUPUJICICH DO OPTIMALIZACE


* Ve vlastnim GAMSovskem kodu budou nadefinovany pouze mnoziny a parametry, se kterymi se bude
* pracovat. Zatimco vsechny (menicise hodnoty) budou nacteny z pomocneho souboru, ktery je ve formatu
* sesitu v MS-Excelu. Jmeno tohoto souboru je "VSTUPNI_SOUBORY.xls"

* Jelikoz není v GAMSU mozne definovat indexove mnoziny s variabilni velikosti, budou indexove mnoziny
* (pro ricice, autobusy, jizdy, dny a tydny v planovacim obdobi) budou definovany ve dvou krocich. Nejdříve
* budou nadefinovany indexove mnoziny, pro "maximalni" pocet clenu a po importovani vsech vstupnich dat
* budou nadefinovany podmnoziny s poctem prvku odpovidajicim importovanym datum.

SET     BUSY_MAX     neomezena indexova mnozina autobusu     /B1*B100/
        RIDICI_MAX   neomezena indexova mnozina autobusu     /R1*R100/
        JIZDY_MAX    neomezena indexova mnozina jizd         /J1*J5000/
        DNY_MAX      neomezena indexova mnozina dnu          /1*364/
        TYDNY_MAX    neomezena indexova mnozina tydnu        /1*52/;


* Omezujici hodnoty dane zakonem:

SET    TIPY_OMEZ    vycet zakonnych omezeni
    / MIN_w          minimalni tydenni odpocinek (h)
      MIN_d          minimalni denni odpocinek  (h)
      MAX_ww         maximalni dvoutydenni doba rizeni (h)
      MAX_w          maximalni tydenni doba rizeni (h)
      MAX_d          maximalni denni doba rizeni  (h)  /;

PARAMETER      OMEZENI[TIPY_OMEZ]     konkretni hodnoty jednotlivych omezeni;
execute "gdxxrw  D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls  PAR=OMEZENI
        RNG=OMEZENI!B2:C6;  CDIM=0  RDIM=1"
execute_load 'VSTUPNI_SOUBORY.gdx'   OMEZENI;


* Vycet slozek ucelove funkce:

SET    SLOZKY_UCEL_FCE jednitlive slozky ucelove funkce
    / PREF_RxB         reference vyberu ridice vzhledem k autobusu,
      SPOTR_RxB        nadspotreba paliva zpusobena kombinaci ridice a autobusu,
      PREF_BxJ         priorita vyberu autobusu vzhledem k jizde,
      SPOTR_BxJ        spotreba paliva autobusu vzhledem k jizde,
      STRIDANI_R       pocet stridani ridicu v ramci jedne varianty,
      PREKR_PRUM_R     prekroceni prumerne doby rizeni u ridicu,
      NEDOS_PRUM_R     nedosazeni prumerne doby rizeni u ridicu,
      PREKR_PRUM_B     prekroceni prumerne ujete vzdalenosti u autobusu,
      NEDOS_PRUM_B     nedosazeni prumerne ujete vzdalenosti u autobusu,
      PREPLAN_R        pocet zmen ridicu pri pleplanovani,
      PREPLAN_B        pocet zmen autobusu pri pleplanovani,
      PREF_ALT         preference uskutecnenych alternativ /;


* Prirazeni vah k jednotlivym slozkam ucelove funkce - odpovida ruznym optimalizacnim strategiim

PARAMETER      STRATEGIE[SLOZKY_UCEL_FCE]      vahy u slozek ucelove funkce;
```

```
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=STRATEGIE
    RNG=VAHY!B2:C13; CDIM=0 RDIM=1"
execute_load 'VSTUPNI_SOUBORY.gdx' STRATEGIE;

* Vymezeni planovaciho obdobi

SET   ZACATEK_KONEC   pomocna mnozina pro nacteni z excelu
    / 0   pocatek planovaciho obdobi
      T   konec planovaciho obdobi /;

PARAMETER    t[ZACATEK_KONEC]      zacatek a konec planovaciho obdobi;
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=t
    RNG=PLAN_OBDOBI!B2:C3; CDIM=0 RDIM=1"
execute_load 'VSTUPNI_SOUBORY.gdx' t;

* Na zaklade importovanych hodnot muze dojit k urceni poctu tydnu a dnu v planovacim obdobi

SCALAR        D_POC     pocet dnu v planovacim obdobi
              W_POC     pocet tydnu v planovacim obdobi;

W_POC = CEIL{(t["T"]-t["0"])/7};
D_POC = W_POC * 7;

SET   DNY[DNY_MAX]              indexova mnozina dnu planovaciho obdobi
      TYDNY[TYDNY_MAX]          indexova mnozina tydnu planovaciho obdobi;

* urceni podmnozin:
DNY[DNY_MAX]$(ORD[DNY_MAX] LE D_POC) = yes;
TYDNY[TYDNY_MAX]$(ORD[TYDNY_MAX] LE W_POC) = yes;


* Import vyctu autobusu a jejich charakteristik.
SET   BUS_CHARAKT            vycet charakteristik autobusu
    / PREKROC_KM             koef. penalizace za prekroceni prum. poctu Km
      NEDOSAZENI_KM          koef. penalizace za nedosazeni prum. poctu Km
      FIX_KM                 pocet predem fixovanych kilometru
      POTVRZENI              pomocna velicina - u pouzitelnych autobusu bude vyplneno 1 /;

PARAMETER    BUS[BUSY_MAX,BUS_CHARAKT]     charakteristiky jednotlivych autobusu;
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=BUS
    RNG=AUTOBUSY!A2:E102; CDIM=1 RDIM=1"
execute_load 'VSTUPNI_SOUBORY.gdx' BUS;

SET   BUSY[BUSY_MAX]          indexova mnozina autobusu;

* urceni podmnoziny:
BUSY[BUSY_MAX]$(BUS[BUSY_MAX,"POTVRZENI"] Gt 0) = yes;


* Import vyctu ridicu a jejich charakteristik.
SET   RIDIC_CHARAKT          vycet charakteristik ridicu
    / PREKROC_HOD            koef. penalizace za prekroceni prum. odprac. doby
      NEDOSAZENI_HOD         koef. penalizace za nedosazeni prum. odprac. doby
      FIX_HOD                pocet predem fixovanych odprac.hodin
      TYDEN_HOD              pocet odrizenych hodin v posl. tydnu pred zacatkem plan. obdobi
      DEN_HOD                pocet odrizenych hodin po posl. den. odp. pred zacatkem plan. obdobi
      TYD_ODP_HOD            pocet odrizenych hodin po posl. tyden. odp. pred zacatkem plan. obdobi
      POTVRZENI              pomocna velicina - u pouzitelnych ridicu bude vyplneno 1/;
PARAMETER    RID[RIDICI_MAX,RIDIC_CHARAKT]  charakteristiky jednotlivych ridicu;
```

```
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=RID
    RNG=RIDICI!A2:H102; CDIM=1 RDIM=1"
execute_load 'VSTUPNI_SOUBORY.gdx' RID;


SET   RIDICI[RIDICI_MAX]          indexova mnozina ridicu;
* urceni podmnoziny:
RIDICI[RIDICI_MAX]$(RID[RIDICI_MAX,"POTVRZENI"] Gt 0) = yes;


* Import vyctu jizd a jejich charakteristik.
SET   JIZDA_CHARAKT          vycet charakteristik jizd
      / UKOL                poradi ukolu
        ALTERNATIVA   poradi alternativy v ramci ukolu
        PORADI           poradi jizdy v alternative
        PREF_ALTER     koef. preference alternativy
        ZACATEK         cas zacatku jizdy
        KONEC            cas konce jizdy
        PRIJEZD          cas prijezdu ridice na misto zacatku jizdy
        ODJEZD           cas odjezdu ridice z mista konce jizdy
        POC_KM          delka trasy jizdy
        DOBA_RIZENI   predpokladana doba rizeni
        DOBA_OST       predpokladana doba ostatnich praci
        ZACH_RID        podminka zachovani ridice
        KONTR_DEN     identifikaror jizdy kontrolovane vuci dennimu odpocinku
        KONTR_TYDEN  identifikaror jizdy kontrolovane vuci tydennimu odpocinku
        POC_RID          potrebny pocet ridicu
        POC_BUS         potrebny pocet autobusu
        POTVRZENI      pomocna velicina - u jizd vstupujicich do planovani bude vyplneno 1 /;
PARAMETER     JIZD[JIZDY_MAX,JIZDA_CHARAKT]  charakteristiky jednotlivych jizd;
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=JIZD
RNG=JIZDY!A2:R5002; CDIM=1 RDIM=1"
execute_load 'VSTUPNI_SOUBORY.gdx' JIZD;

SET   JIZDY[JIZDY_MAX]          indexova mnozina jizd;

* urceni velikosti podmnoziny:
JIZDY[JIZDY_MAX]$(JIZD[JIZDY_MAX,"POTVRZENI"] Gt 0) = yes;


* Import charakteristik vsech kombinaci ridicu a autobusu.
SET   R_B_CHARAKT          vycet charakteristik kombinace ridic x autobus
    / PRIPUSTNOST          pripustnost kombinace
      PREFERENCE           koef. preference kombinace
      NADSPOTREBA          ocekavana nadspotreba /;

PARAMETER     RIDIC_BUS[RIDICI_MAX,BUSY_MAX,R_B_CHARAKT]       charakteristiky vsech
    kombinaci ridicu a autobusu;
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=RIDIC_BUS
    RNG=RIDIC_BUS!A2:E10002 CDIM=1 RDIM=2"
execute_load 'VSTUPNI_SOUBORY.gdx' RIDIC_BUS


* Import charakteristik vsech kombinaci ridicu a jizd.
SET   R_J_CHARAKT          vycet charakteristik kombinace ridic x jizda
    / PRIPUSTNOST        pripustnost kombinace
      PREFERENCE         koef. preference kombinace /;

PARAMETER     RIDIC_JIZDA[RIDICI_MAX,JIZDY_MAX,R_J_CHARAKT]       charakteristiky vsech
    kombinaci ridicu a jizd;
```

```
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=RIDIC_JIZDA
    RNG=RIDIC_JIZDA!A2:D65536 CDIM=1 RDIM=2"
execute_load 'VSTUPNI_SOUBORY.gdx' RIDIC_JIZDA


* Import charakteristik vsech kombinaci autobusu A jizd.
SET   B_J_CHARAKT           vycet charakteristik kombinace autobus x jizda
     / PRIPUSTNOST             pripustnost kombinace
       PREFERENCE              koef. preference kombinace
       NADSPOTREBA             ocekavana nadspotreba /;

PARAMETER    BUS_JIZDA[BUSY_MAX,JIZDY_MAX,B_J_CHARAKT]     charakteristiky vsech
    kombinaci autobusu a jizd;
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=BUS_JIZDA
    RNG=BUS_JIZDA!A2:E65536 CDIM=1 RDIM=2"
execute_load 'VSTUPNI_SOUBORY.gdx' BUS_JIZDA


* Import puvodniho planu obsazeni jizd ridici.
PARAMETER    PLAN_RIDIC[RIDICI_MAX,JIZDY_MAX]     puvodni naplanovani obsazeni ridicu;
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=PLAN_RIDIC
    RNG=PLAN_RIDICI!A2:C5001 CDIM=0 RDIM=2"
execute_load 'VSTUPNI_SOUBORY.gdx' PLAN_RIDIC


* Import puvodniho planu obsazeni jizd autobusy.
PARAMETER    PLAN_BUS[BUSY_MAX,JIZDY_MAX]     puvodni naplanovani obsazeni autobusu;
execute "gdxxrw D:\SOKOL\VSTUPNI_SOUBORY\VSTUPNI_SOUBORY.xls PAR=PLAN_BUS
    RNG=PLAN_AUTOBUSY!A2:C5001; CDIM=0 RDIM=2"
execute_load 'VSTUPNI_SOUBORY.gdx' PLAN_BUS


* zavedeni dalsich pomocnych indexovych mnozin
ALIAS (JIZDY,J,I);
ALIAS (RIDICI,R);
ALIAS (BUSY,B);
ALIAS (DNY,D);
ALIAS (TYDNY,W);
ALIAS (JIZDY_MAX,J_M,I_M);
ALIAS (RIDICI_MAX,R_M);
ALIAS (BUSY_MAX,B_M);
ALIAS (DNY_MAX,D_M);
ALIAS (TYDNY_MAX,W_M);

* urceni poctu prvku v jednotlivych nnozinach
SCALAR        B_POC     celkovy pocet autobusu
              R_POC     celkovy pocet ridicu
              J_POC     celkovy pocet jizd;

B_POC = CARD[BUSY];
R_POC = CARD[RIDICI];
J_POC = CARD[JIZDY];


*      PRIPRAVNE VYPOCTY PRED VLASTNI OPTIMALIZACI

* Kvuli zjednoduseni zapisu i vypoctu vlastni optimalizacni ulohy se ze vstupnich dat urci nektere agregace,
* ktere pote budou vystupovat ve vlastni optimalizacni uloze.
```

```
*   Urceni prumerne doby rizeni a prumerne ujete vzdalenosti:

* Nejdrive se urci vahy jednotlivych alternativ.
PARAMETER VAHY[J_M]      vahy jednotlivych variant;

VAHY[JIZDY]=JIZD[JIZDY,"PREF_ALTER"]/SUM{J,JIZD[J,"PREF_ALTER"]*1$(JIZD[JIZDY,"UKOL"]
    Eq JIZD[J,"UKOL"])*1$(JIZD[J,"PORADI"] Eq 1)};

* Nasleduje vypocet prumerne odpracovane doby jednoho ridice:

SCALAR Q      prumerna doba;
Q=(SUM{R,RID[R,"FIX_HOD"]} + SUM{J,(JIZD[J,"DOBA_RIZENI"] +
     JIZD[J,"DOBA_OST"]) * JIZD[J,"POC_RID"] * VAHY[J]})/R_POC

* Nasleduje vypocet prumerne ujete vzdalenosti pripadajici na jeden autobus:

SCALAR G      prumerna vzdalenost;
G=(SUM{B,BUS[B,"FIX_KM"]}+ SUM{J,JIZD[J,"POC_KM"]*VAHY[J]})/B_POC;


* Nasleduje vypocet  tri pomocnych ctvercovych matic J x J. Jejich nadefinovani jednak zjednodusi zapis
* nekolika podminek ve vlastnim modelu a navic vypocet hodnot usnadni vlastni beh optimalizacniho algoritmu
* - vyrazne se snizi pocet testovanych podminek. Konkretne se jedna o matice:
*    1) matice JJ_BUS - Pomoci teto matice se oznaci dvojice jizd, ktere se casove prekryvaji - v rozmezi
*       zacatku a konce jizdy.  Matice bude pouzita pro zjisteni, zda autobus neni zaroven pridelen na vice jizd.
*    2) matice JJ_RID - pomoci teto matice se oznaci dvojice jizd, ktere se casove prekryvaji _ v rozmezi zacatku
*       a konce jizdy, vcetne dob prijezdu a odjezdu. Tato matice bude pouzita pro zjisteni, zda ridic neni zaroven
*       pridelen na vice jizd.
*    3) matice GAMA - Pomoci teto matice se oznaci dvojice jizd, ktere po sobe bezprostredne nasleduji v ramci
*       jedne varinty.

PARAMETER   JJ_RID[J_M,I_M]        matice soucasnych jizd pro ridice
            JJ_BUS[J_M,I_M]        matice soucasnych jizd pro autobusy
            GAMA[J_M,I_M]          matice naslednych jizd;

* V definici parametru, stejne, jako v definici promennych a omezujicich rovnic, nelze diky omezeni jazyka
* GAMS pouzit mnoziny, ktere byly nadefinovany jako podmnoziny nejakych mnozin. Proto musi byt pouzito
* indexovani pomoci povodnich (velkych mnozin). Veliciny odpovidajici neuskutecnenym jizdam (ridicum,
* autobusum...) ovsem zustanou po celou nulove.

* V nasledujicich prirazenich je na rozdil od ostatnich prirazeni pouzito indexovani pomoci puvodni (velke)
*  mnoziny jizd - tato konstrukce je pouzita kvuli funkci "ORD" (poradi prvku v mnozine), tato funkce totiz na
*  podmnozinach "nefunguje".

JJ_BUS[J_M,I_M]$(J[J_M] and I[I_M]) = 1$ {ORD(J_M) Lt J_POC AND ORD(I_M) Gt ORD(J_M) AND
    JIZD[J_M,"UKOL"] NE JIZD[I_M,"UKOL"] AND JIZD[I_M,"ZACATEK"] Lt JIZD[J_M,"KONEC"]};


JJ_RID[J_M,I_M]$(J[J_M] and I[I_M]) = 1$ {ORD(J_M) Lt J_POC AND ORD(I_M) Gt ORD(J_M) AND
    JIZD[J_M,"UKOL"] NE JIZD[I_M,"UKOL"] AND JIZD[I_M,"ZACATEK"] - JIZD[I_M,"PRIJEZD"] Lt
    JIZD[J_M,"KONEC"] + JIZD[J_M,"ODJEZD"]};


GAMA[J_M,I_M]$(J[J_M] and I[I_M]) = 1$(JIZD[J_M,"UKOL"] Eq JIZD[I_M,"UKOL"] AND
    JIZD[J_M,"ALTERNATIVA"] Eq JIZD[I_M,"ALTERNATIVA"] AND  JIZD[J_M,"PORADI"] +1 Eq
    JIZD[I_M,"PORADI"]);

* S pomoci matice GAMA bude definovana podmnozina JJ mnoziny vsech jizd (J), do mnoziny JJ budou
* zahrnuty pouze jizdy, pro ktere existuje bezprostredne nasledujici jizda v teze alternative.
```

```
SET    JJ[J_M]   podmnozina jizd majicich "nasledovnika";
JJ[J]${SUM(I,GAMA[J,I]) Eq 1} = yes;


*         DEFINOVANI VSECH PROMENNYCH, KTERE BUDOU POUZITY V MODELU

VARIABLES     X[R_M,J_M]               prirazeni ridicu jizdam
              Y[B_M,J_M]               prirazeni autobusu jizdam
              Z[J_M]                   realizace jizdy
              Od[R_M,D_M]              cas zacatku denniho odpocinku
              Ow[R_M,T_M]              cas zacatku tydenniho odpocinku
              Xw[R_M,J_M]              jizda pred tydennim odpocinkem
              Xd[R_M,J_M]              jizda pred dennim odpocinkem
              XY[R_M,B_M,J_M]          pomocna promenna s kombinaci ridice a autobusu
              HODNOTA_UC_FCE           hodnota ucelove funkce;

*nasleduji pomocne promenne, ktere v ucelove funkci nahradi kladnou a zapornou cast
VARIABLES     PREKR_PRM_RID[R_M]             prekroceni prumerne odpracovane doby u ridicu
              NEDOS_PRM_RID[R_M]             nedosazeni prumerne odpracovane doby u ridicu
              PREKR_PRM_BUS[B_M]             prekroceni prumerne odjete vzdalenosti u autobusu
              NEDOS_PRM_BUS[B_M]             nedosazeni prumerne odjete vzdalenosti u autobusu
              STRIDANI_POS[R_M,J_M]          kladna cast promenne stridani ridicu
              STRIDANI_NEG[R_M,J_M]          zaporna cast promenne stridani ridicu;

* omezeni nekterych promennych na binarni nebo nezaporne:
BINARY VARIABLES        X, Y, Z, Xw, Xd, XY, STRIDANI_POS, STRIDANI_NEG;
POSITIVE VARIABLES      PREKR_PRM_RID, NEDOS_PRM_RID, PREKR_PRM_BUS,
                        NEDOS_PRM_BUS;


*       VLATNI MODEL - OMEZUJICI PODMINKY

* Omezeni zacatku odpocinku do spravnych dnu a tydnu. (Opet pouzita indexace prez neomezenou mnozinu
* kvuli funkci "ORD".)

Od.LO[R,D_M]$D[D_M]  =  t["0"] + ORD(D_M)-1;
Od.UP[R,D_M]$D[D_M]  =  t["0"] + ORD(D_M);
Ow.LO[R,W_M]$W[W_M]  =  t["0"] + ORD(W_M)*7 - 7;
Ow.UP[R,W_M]$W[W_M]  =  t["0"] + ORD(W_M)*7;


* V pripade, ze existuje nejaky puvodni plan (a dochazi tedy k preplanovani) vlozi se hodnoty puvodniho
* naplanovani do inicializacnich hodnot promennych X a Y.

X.l[R,J] $ (PLAN_RIDIC[R,J]) = 1;
Y.l[B,J] $ (PLAN_BUS[B,J]) = 1;


* Pro nepripustne kombinace jizd s autobusy a s ridici se jeste pred vlastni optimalizaci nastavi fixni hodnoty
* odpovidajicich promennych Y, X a XY na nulu.

* U vsech jizd musi byt u nepripustne kombinace ridic x jizda promenna X i XY vyplnena hodnotou 0.
X.fx[R,J] $ (RIDIC_JIZDA [R,J,"PRIPUSTNOST"] NE 1) = 0;
XY.fx[R,B,J] $ (RIDIC_JIZDA [R,J,"PRIPUSTNOST"] NE 1) = 0;

* U vsech jizd musi byt u nepripustne kombinace autobus x jizda promenna Y i XY vyplnena hodnotou 0.
Y.fx[B,J] $ (BUS_JIZDA [B,J,"PRIPUSTNOST"] NE 1) = 0;
XY.fx[R,B,J] $ (BUS_JIZDA [B,J,"PRIPUSTNOST"] NE 1) = 0;
```

```
* U vsech jizd musi byt u nepripustne kombinace ridic x autobus promenna XY vyplnena hodnotou 0.
XY.fx[R,B,J] $ (RIDIC_BUS[R,B,"PRIPUSTNOST"] NE 1) = 0;

* U vsech jizd s nulovou potrebou ridicu budou promenne X a XY vyplneny hodnotou 0.
X.fx[R,J] $ (JIZD[J,"POC_RID"] Eq 0) = 0;
XY.fx[R,B,J] $ (JIZD[J,"POC_RID"] Eq 0) = 0;

* U vsech jizd s nulovou potrebou autobusu budou promenne Y a XY vyplneny hodnotou 0.
Y.fx[B,J] $ (JIZD[J,"POC_BUS"] Eq 0) = 0;
XY.fx[R,B,J] $ (JIZD[J,"POC_BUS"] Eq 0) = 0;

* Vycet podminek:

EQUATIONS
* jednotliva omezeni v modelu
STEJNA_VARIANTA [J_M,I_M]           z jadne varianty se bud neuskutecni zadna jizda nebo vsechny jizdy
JEDNA_VARIANTA [J_M]                pro kazdy ukol bude realizovana prave jedna varianta
STEJNY_BUS [B_M,J_M,I_M]            pro vsechny jizdy z jedne alternativy musi byt pouzit stejny autobus
STEJNY_RIDIC [R_M,J_M,I_M]          u vybranych jizd je nutne zkontrolovat zachovani ridice pri prechodu
                                    na dalsi jizdu
POC_BUSY [J_M]                      pro kazdou uskutecnenou jizdu musi byt pouzit odp. pocet autobusu
POC_RIDICU [J_M]                    pro kazdou uskutecnenou jizdu musi byt pouzit odp. pocet ridicu
NO_CURR_BUS [B_M,J_M,I_M]           autobus nesmi byt pouzit v jeden okamzik na vice jizd
NO_CURR_RIDIC [R_M,J_M,I_M]         ridic nesmi byt pouzit v jeden okamzik na vice jizd
OMEZ_2W [R_M,W_M]                   dvoutydenni omezeni poctu ridicem odrizenych hodin
OMEZ_1ST_W [R_M]                    omezeni na soucet odrizenych hodin v posl. tydnu pred plan. obd. a
                                    v prvnim tydnu
OMEZ_LAST_W [R_M]                   omezeni na soucet odrizenych hodin v posl. tydnu plan. obdobi
NO_CURR_Ow [R_M,W_M]                tydenni odpocinky jenoho ridice se nesmi prekryvat
NO_CURR_Od [R_M,D_M]                denni odpocinky jenoho ridice se nesmi prekryvat

* kontrola presahu dennich odpocinku a 'kontrolovanych' jizd
ODP_NASL_D [R_M,J_M,D_M]            denni odpocinek za nasledujici den musi zacit po jizde
ODP_TYZ_D1 [R_M,J_M,D_M]            jizda ze shodneho dne zacinajici pred odpocinkem
ODP_TYZ_D2 [R_M,J_M,D_M]            jizda ze shodneho dne zacinajici po odpocinku
ODP_PREDCH_D [R_M,J_M,D_M]          denni odpocinek za predch. den musi skoncit pred zacatkem jizdy

* kontrola presahu tydennich odpocinku a 'kontrolovanych' jizd
ODP_TYZ_W1 [R_M,J_M,W_M]            jizda ze shodneho tydne zacinajici pred odpocinkem
ODP_TYZ_W2 [R_M,J_M,W_M]            jizda ze shodneho tydne zacinajici po odpocinku
ODP_PREDCH_W [R_M,J_M,W_M]          tydenni odpocinek za predch. tyden skonci pred zacatkem jizdy

* kontrola denni doby rizeni
DENNI_DOBA_RIZ [R_M,D_M]            denni doba rizeni nesmi prekrocit povolenou hodnotu
DENNI_DOBA_RIZ_1st [R_M]            prvni denni doba rizeni  nesmi prekrocit povolenou hodnotu
DENNI_DOBA_RIZ_last [R_M]           posledni denni doba rizeni nesmi prekrocit povolenou hodnotu

* kontrola tydenni doby rizeni
TYDENNI_DOBA_RIZ [R_M,w_M]          tydenni doba rizeni nesmi prekrocit povolenou hodnotu
TYDENNI_DOBA_RIZ_1st [R_M]          prvni tydenni doba rizeni  nesmi prekrocit povolenou hodnotu

* ucelova funkce vcetne pomocnych rovnic
ROZD_PRUM_DOBA[R_M]                 rozdil od prumerne odpracovane doby u ridicu
ROZD_PRUM_VZDALENOST[B_M]           rozdil od prumerne odjete vzdalenosti u autobusu
STRIDANI_RIDICU[R_M,J_M]            spravne dosazeni do pomocne promenne stridani ridicu
RIDIC_BUS_up [R_M,B_M,J_M]          zjisteni kombinace ridic-autobus na jizde - horni mez
RIDIC_BUS_lo [R_M,B_M,J_M]          zjisteni kombinace ridic-autobus na jizde - dolni mez
UCEL_FCE                            prirazeni hodnoty ucelove funkce;
```

```
* Vlastni zapis podminek:

STEJNA_VARIANTA [J,I]$(GAMA[J,I]) .. Z[J] - Z[I] =E= 0;

JEDNA_VARIANTA [J] $ (JIZD[J,"ALTERNATIVA"] Eq 1 AND JIZD[J,"PORADI"] Eq 1) ..
    SUM{I$(JIZD[J,"UKOL"] Eq JIZD[I,"UKOL"] AND JIZD[I,"PORADI"] Eq 1),Z[I]} =E= 1;

STEJNY_BUS [B,J,I] $ (GAMA[J,I]) .. Y[B,J] - Y[B,I] =E= 0;

STEJNY_RIDIC [R,J,I] $ (GAMA[J,I]) .. (X[R,J] - X[R,I])*JIZD[J,"ZACH_RID"] =E= 0;

POC_BUSY [J] .. SUM(B,Y[B,J]) =E= Z[J] * JIZD[J,"POC_BUS"];

POC_RIDICU [J] .. SUM(R,X[R,J]) =E= Z[J] * JIZD[J,"POC_RID"];

NO_CURR_BUS [B,J,I] $ (JJ_BUS[J,I]) .. Y[B,J] + Y[B,I] =L= 1;

NO_CURR_RIDIC [R,J,I] $ (JJ_RID[J,I]) .. X[R,J] + X[R,I] =L= 1;

OMEZ_2W [R,W_M] $ (ORD(W_M) Lt W_POC) .. SUM(J${7*ORD(W_M)-7 LE JIZD[J,"ZACATEK"]-
    t["0"] AND JIZD[J,"ZACATEK"]-t["0"] Lt 7*ORD(W_M)+7}, X[R,J]*JIZD[J,"DOBA_RIZENI"]) =L=
    OMEZENI["MAX_ww"];

OMEZ_1ST_W [R] .. SUM{J$(JIZD[J,"ZACATEK"]-t["0"] Lt 7),X[R,J]*JIZD[J,"DOBA_RIZENI"]} +
    RID[R,"TYDEN_HOD"] =L= OMEZENI["MAX_ww"];

OMEZ_LAST_W [R] .. SUM{J$(7*W_POC -7 LE JIZD[J,"ZACATEK"]-t["0"]), X[R,J]*
    JIZD[J,"DOBA_RIZENI"]} =L= 0.6*OMEZENI["MAX_ww"];

NO_CURR_Ow [R,W_M] $ {ORD(W_M) Lt W_POC} .. Ow[R,W_M] + OMEZENI["MIN_w"]/24 =L=
    Ow[R,W_M+1];

NO_CURR_Od [R,D_M] $ {ORD(D_M) Lt W_POC*7} .. Od[R,D_M] + OMEZENI["MIN_d"]/24 =L=
    Od[R,D_M+1];


*   denni odpocinky – kontrola:
ODP_NASL_D [R,J,D_M]$(JIZD[J,"KONTR_DEN"] Eq 1 AND TRUNC{JIZD[J,"ZACATEK"]} Ne
    TRUNC{JIZD[J,"KONEC"]} AND ORD{D_M}-1 Eq TRUNC{JIZD[J,"KONEC"]-t["0"]}) ..
    JIZD[J,"KONEC"] + (X[R,J]-1)*10*W_POC =L= Od[R,D_M];

ODP_TYZ_D1 [R,J,D_M]$(JIZD[J,"KONTR_DEN"] Eq 1 AND TRUNC{JIZD[J,"ZACATEK"] -t["0"]} Eq
    ORD{D_M}-1) .. JIZD[J,"KONEC"] + (Xd[R,J]-1)*10*W_POC =L= Od[R,D_M];

ODP_TYZ_D2 [R,J,D_M]$(JIZD[J,"KONTR_DEN"] Eq 1 AND TRUNC{JIZD[J,"ZACATEK"]-t["0"]} Eq
    ORD{D_M}-1) .. JIZD[J,"ZACATEK"] + (Xd[R,J]-X[R,J]+1)*10*W_POC =G= Od[R,D_M] +
    OMEZENI["MIN_d"]/24;

ODP_PREDCH_D [R,J,D_M]$(JIZD[J,"KONTR_DEN"] Eq 1 AND JIZD[J,"ZACATEK"]-t["0"] GE 1 AND
    ORD{D_M} Eq TRUNC{JIZD[J,"ZACATEK"]-t["0"]}) .. JIZD[J,"ZACATEK"] + (1-X[R,J])*10*
    W_POC =G= Od[R,D_M] + OMEZENI["MIN_d"]/24;

*   tydenni odpocinky – kontrola:
ODP_TYZ_W1 [R,J,W_M]$(JIZD[J,"KONTR_TYDEN"] Eq 1 AND TRUNC{(JIZD[J,"ZACATEK"] -
    t["0"])/7} Eq ORD{W_M}-1) .. JIZD[J,"KONEC"] + (Xw[R,J]-1)*10*W_POC =L= Ow[R,W_M];

ODP_TYZ_W2 [R,J,W_M]$(JIZD[J,"KONTR_TYDEN"] Eq 1 AND TRUNC{(JIZD[J,"ZACATEK"] -
    t["0"])/7} Eq ORD{W_M}-1) .. JIZD[J,"ZACATEK"] + (Xw[R,J]-X[R,J]+1)*10*W_POC =G=
    Ow[R,W_M] + OMEZENI["MIN_w"]/24;
```

```
ODP_PREDCH_W [R,J,W_M]$(JIZD[J,"KONTR_TYDEN"] Eq 1 AND JIZD[J,"ZACATEK"] GE 7 AND
      ORD{W_M} Eq TRUNC{(JIZD[J,"ZACATEK"]-t["0"])/7}) .. JIZD[J,"ZACATEK"] + (1-
      X[R,J])*10*W_POC =G= Ow[R,W_M] + OMEZENI["MIN_w"]/24;


*    denni doba rizeni – kontrola:
DENNI_DOBA_RIZ [R,D_M]$(ORD(D_M) Lt W_POC*7) .. SUM{J$(TRUNC{JIZD[J,"ZACATEK"]-t["0"]}
      Eq ORD{D_M}-1),JIZD[J,"DOBA_RIZENI"] * JIZD[J,"KONTR_DEN"] * (X[R,J]-Xd[R,J])} +
      SUM(J$(TRUNC{JIZD[J,"ZACATEK"]-t["0"]} Eq ORD{D_M}),JIZD[J,"DOBA_RIZENI"] *
      JIZD[J,"KONTR_DEN"] * Xd[R,J]) =L= OMEZENI["MAX_d"];

DENNI_DOBA_RIZ_1st [R] .. SUM{J$(JIZD[J,"ZACATEK"]-t["0"] Lt 1),JIZD[J,"DOBA_RIZENI"] *
      JIZD[J,"KONTR_DEN"] * Xd[R,J]} + RID[R,"DEN_HOD"] =L= OMEZENI["MAX_d"];

DENNI_DOBA_RIZ_last [R] .. SUM{J$(JIZD[J,"ZACATEK"]-t["0"] GE W_POC*7-1), (X[R,J]-Xd[R,J]) *
      JIZD[J,"DOBA_RIZENI"] * JIZD[J,"KONTR_DEN"] } =L= OMEZENI["MAX_d"];


*    tydenni doba rizeni - kontrola
TYDENNI_DOBA_RIZ [R,w_M]$(ORD(w_M) Lt W_POC) .. SUM{J$(TRUNC{(JIZD[J,"ZACATEK"]-
      t["0"])/7} Eq ORD{w_M}-1),JIZD[J,"DOBA_RIZENI"] * JIZD[J,"KONTR_TYDEN"] * (X[R,J]-
      Xt[R,J])} + SUM(J$(TRUNC{(JIZD[J,"ZACATEK"]-t["0"])/7} Eq ORD{D_M}),
      JIZD[J,"DOBA_RIZENI"] * JIZD[J,"KONTR_TYDEN"] * Xt[R,J]) =L= OMEZENI["MAX_w"];

TYDENNI_DOBA_RIZ_1st [R] .. SUM{J$(JIZD[J,"ZACATEK"]-t["0"] Lt 7),JIZD[J,"DOBA_RIZENI"] *
      JIZD[J,"KONTR_TYDEN"] * Xt[R,J]} + RID[R,"TYD_ODP_HOD"] =L= OMEZENI["MAX_w"];


*    ucelova funkce a pomocne rovnice
ROZD_PRUM_DOBA[R].. PREKR_PRM_RID[R] - NEDOS_PRM_RID[R] =E= SUM{J,X[R,J] *
      (JIZD[J,"DOBA_RIZENI"]+JIZD[J,"DOBA_OST"])} + RID[R,"FIX_HOD"] - Q;

ROZD_PRUM_VZDALENOST[B].. PREKR_PRM_BUS[B] - NEDOS_PRM_BUS[B] =E= SUM(J,Y[B,J] *
      JIZD[J,"POC_KM"]) +BUS[B,"FIX_KM"] - G;

STRIDANI_RIDICU[R,JJ] .. STRIDANI_POS[R,JJ] - STRIDANI_NEG[R,JJ] =E= X[R,JJ] -
      SUM(I$GAMA[JJ,I],X[R,I]);

RIDIC_BUS_up [R,B,J] .. XY[R,B,J] =L= (X[R,J] + Y[B,J]) / 2;

RIDIC_BUS_lo [R,B,J] .. XY[R,B,J] =G= X[R,J] + Y[B,J] - 1;


UCEL_FCE ..     HODNOTA_UC_FCE =E=
-     STRATEGIE["PREF_RxB"] * SUM{(B,J,R), XY[R,B,J]* RIDIC_BUS[R,B,"PREFERENCE"]*
        JIZD[J,"POC_KM"]}
+     STRATEGIE["SPOTR_RxB"] * SUM{(B,J,R), XY[R,B,J]* RIDIC_BUS[R,B,"NADSPOTREBA"]*
        JIZD[J,"POC_KM"]}
-     STRATEGIE["PREF_BxJ"] * SUM{(B,J),Y[B,J]* BUS_JIZDA[B,J,"PREFERENCE"]*
        JIZD[J,"POC_KM"]}
+     STRATEGIE["SPOTR_BxJ"] * SUM{(B,J),Y[B,J]* BUS_JIZDA[B,J,"NADSPOTREBA"]*
        JIZD[J,"POC_KM"]}
+     STRATEGIE["STRIDANI_R"] * SUM{(R,JJ),(STRIDANI_POS[R,JJ]+STRIDANI_NEG[R,JJ])}*0.5
+     STRATEGIE["PREKR_PRUM_R"] * SUM{R,PREKR_PRM_RID[R]*RID[R,"PREKROC_HOD"]}
+     STRATEGIE["NEDOS_PRUM_R"] * SUM{R,NEDOS_PRM_RID[R]*RID[R,"NEDOSAZENI_HOD"]}
+     STRATEGIE["PREKR_PRUM_B"] * SUM{B,PREKR_PRM_BUS[B]*BUS[B,"PREKROC_KM"]}
+     STRATEGIE["NEDOS_PRUM_B"] * SUM{B,NEDOS_PRM_BUS[B]*BUS[B,"NEDOSAZENI_KM"]}
+     STRATEGIE["PREPLAN_R"] * SUM{(J,R)$PLAN_RIDIC[R,J],1-X[R,J]}
+     STRATEGIE["PREPLAN_B"] * SUM{(J,B)$PLAN_BUS[B,J],1-Y[B,J]}
-     STRATEGIE["PREF_ALT"] * SUM{J$(JIZD[J,"PORADI"] Eq 1),Z[J]*JIZD[J,"PREF_ALTER"]};
```

```
* Spusteni vlastniho optimalizacniho programu - minimalizuje se promenna HODNOTA_UC_FCE, za dodrzeni
* vsech vyse uvedenych podminek. Pro minimalizaci je pouzito celociselne linearni programovani s nastevenym
* resicem CPLEX. Pred spustenim dochazi k nastaveni nekolika parametru, ktere ovlivnuji vypocet.

* pro úlohy tipu MIP (smisene celociselne linearni programovani) urcujeme resic CPLEX
OPTION MIP = CPLEX;

* vypnuti krizovych referenci ve vystupnim textovam souboru
$OFFSYMXREF

* pocet vypsanych rovnic ve vystupnim textovam souboru pouzivame pouze 1, kvuli zkraceni výstupního
* souboru
OPTION LIMROW = 1;

* pocet zobrazovanych hodnot ve vystupnim textovam souboru
OPTION LIMCOL = 1;

* maximalni pocet iteraci
OPTION ITERLIM = 1000000;

*omezeni na pocet casovych jednotek (sekund), ktere muze trvat vypocet
OPTION RESLIM = 100000;


* vlastni deficice modelu - parametr ALL znaci,ze do modelu jsou zahrnuty vsechny vyse zapsane rovnice
MODEL   PRIDELENI      urceni optimalniho planu        /ALL/;

* parametr urcujici s jakou presnosti se hleda optimalni reseni (pri nastaveni parametru na 0 se hleda skutecne
* optimum, defaultni hodnota je 0.1
PRIDELENI.OPTCR = 0.3;

* spusteni vypoctu, urcujeme, ze se bude minimalizovat hodnota promenne "HODNOTA_UC_FCE" a ze se
* jedna o ulohu tipu MIP
SOLVE   PRIDELENI     MINIMIZING     HODNOTA_UC_FCE      USING    MIP;


* Vypocet konkretnich hodnot jednotlivych slozek ucelove funkce. Do vystupniho souboru jsou pro prehlednost
* pridany i jednotlive vahy ze vstupu a vypocteny soucin.
SET     UCEL_FCE_VYSTUP          slozky vystupniho souboru s hodnotou ucelove funkce
     /VAHY                       vahy ze vstupniho souboru
     HODN_DILCI_FCE              nevazena hodnota dilci ucelove funkce
     SOUCIN                      hodnota dilci fce vynasobena prislusnou vahou/;

PARAMETER     OPTIMAL_UCEL_FCE[SLOZKY_UCEL_FCE,UCEL_FCE_VYSTUP]       vysledne
        hodnoty ucelove funkce;

OPTIMAL_UCEL_FCE["PREF_RxB","HODN_DILCI_FCE"] =
-   SUM{(B,J,R),XY.l[R,B,J]*RIDIC_BUS[R,B,"PREFERENCE"]*JIZD[J,"POC_KM"]};

OPTIMAL_UCEL_FCE["SPOTR_RxB","HODN_DILCI_FCE"] =
    SUM{(B,J,R),XY.l[R,B,J]*RIDIC_BUS[R,B,"NADSPOTREBA"]*JIZD[J,"POC_KM"]};

OPTIMAL_UCEL_FCE["PREF_BxJ","HODN_DILCI_FCE"] =
-   SUM{(B,J),Y.l[B,J]*BUS_JIZDA[B,J,"PREFERENCE"]*JIZD[J,"POC_KM"]};

OPTIMAL_UCEL_FCE["SPOTR_BxJ","HODN_DILCI_FCE"] =
    SUM{(B,J),Y.l[B,J]*BUS_JIZDA[B,J,"NADSPOTREBA"]*JIZD[J,"POC_KM"]};
```

```
OPTIMAL_UCEL_FCE["STRIDANI_R","HODN_DILCI_FCE"] =
    SUM{(R,JJ),(STRIDANI_POS.l[R,JJ]+STRIDANI_NEG.l[R,JJ])}*0.5;

OPTIMAL_UCEL_FCE["PREKR_PRUM_R","HODN_DILCI_FCE"] =
    SUM{R,PREKR_PRM_RID.l[R]*RID[R,"PREKROC_HOD"]};

OPTIMAL_UCEL_FCE["NEDOS_PRUM_R","HODN_DILCI_FCE"] =
    SUM{R,NEDOS_PRM_RID.l[R]*RID[R,"NEDOSAZENI_HOD"]};

OPTIMAL_UCEL_FCE["PREKR_PRUM_B","HODN_DILCI_FCE"] =
    SUM{B,PREKR_PRM_BUS.l[B]*BUS[B,"PREKROC_KM"]};

OPTIMAL_UCEL_FCE["NEDOS_PRUM_B","HODN_DILCI_FCE"] =
    SUM{B,NEDOS_PRM_BUS.l[B]*BUS[B,"NEDOSAZENI_KM"]};

OPTIMAL_UCEL_FCE["PREPLAN_R","HODN_DILCI_FCE"] =
    SUM{(J,R)$PLAN_RIDIC[R,J],1-X.l[R,J]};

OPTIMAL_UCEL_FCE["PREPLAN_B","HODN_DILCI_FCE"] =
    SUM{(J,B)$PLAN_BUS[B,J],1-Y.l[B,J]};

OPTIMAL_UCEL_FCE["PREF_ALT","HODN_DILCI_FCE"] =
-   SUM{J$(JIZD[J,"PORADI"] Eq 1),Z.l[J]*JIZD[J,"PREF_ALTER"]};

OPTIMAL_UCEL_FCE[SLOZKY_UCEL_FCE,"VAHY"] =  STRATEGIE[SLOZKY_UCEL_FCE];

OPTIMAL_UCEL_FCE[SLOZKY_UCEL_FCE,"SOUCIN"] =
     OPTIMAL_UCEL_FCE[SLOZKY_UCEL_FCE,"VAHY"]*OPTIMAL_UCEL_FCE[SLOZKY_UCEL_FC
     E,"HODN_DILCI_FCE"];


* Zjistene hodnoty se exportuji do excelovskaho sesitu umisteneho ve stejnem adresari, jako vstupni soubor.
* Jelikoz je problematicke exportovat z GAMSu hodnoty promennych, budou zavedeny nove pomocne
* parametry, do nich budou prirazeny optimalizaci urcene hodnoty promennych a nasledne budou exportovany
* hodnoty techto parametru.

PARAMETERS          Z_EXP[J_M]            realizace jizdy - pro export
                    X_EXP[R_M,J_M]        prirazeni ridicu jizdam - pro export
                    Y_EXP[B_M,J_M]        prirazeni autobusu jizdam - pro export
                    Od_EXP[R_M,D_M]       cas zacatku denniho odpocinku - pro export
                    Ow_EXP[R_M,W_M]       cas zacatku tydenniho odpocinku - pro export;

Z_EXP[J]  = Z.l[J];
X_EXP[R,J] = X.l[R,J];
Y_EXP[B,J] = Y.l[B,J];
Od_EXP[R,D] = Od.l[R,D];
Ow_EXP[R,W] = Ow.l[R,W];

execute_unload 'gdxxrwwrite.gdx' OPTIMAL_UCEL_FCE,Z_EXP,X_EXP,Y_EXP,Od_EXP,Ow_EXP;

execute  "gdxxrw gdxxrwwrite.gdx  o=D:\SOKOL\VSTUPNI_SOUBORY\VYSTUPNI_SOUBORY.xls
    par=Z_EXP rng=REALIZACE_JIZD!A2:B3000;  CDIM=0  RDIM=1"

execute  "gdxxrw gdxxrwwrite.gdx  o=D:\SOKOL\VSTUPNI_SOUBORY\VYSTUPNI_SOUBORY.xls
par=X_EXP rng=OBSAZENI_RIDICI!A2:C50000;  CDIM=0  RDIM=2"

execute  "gdxxrw gdxxrwwrite.gdx  o=D:\SOKOL\VSTUPNI_SOUBORY\VYSTUPNI_SOUBORY.xls
    par=Y_EXP rng=OBSAZENI_AUTOBUSY!A2:C50000;  CDIM=0  RDIM=2"
```

```
execute "gdxxrw gdxxrwwrite.gdx  o=D:\SOKOL\VSTUPNI_SOUBORY\VYSTUPNI_SOUBORY.xls
     par=Od_EXP rng=DENNI_ODPOCINKY!A2:C50000;  CDIM=0  RDIM=2"

execute "gdxxrw gdxxrwwrite.gdx  o=D:\SOKOL\VSTUPNI_SOUBORY\VYSTUPNI_SOUBORY.xls
     par=Ow_EXP rng=TYDENNI_ODPOCINKY!A2:C50000;  CDIM=0  RDIM=2"

execute "gdxxrw gdxxrwwrite.gdx  o=D:\SOKOL\VSTUPNI_SOUBORY\VYSTUPNI_SOUBORY.xls
     par=OPTIMAL_UCEL_FCE rng=UCEL_FCE!A2:D14;  CDIM=1  RDIM=1"
```