

**ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA  
V PRAZE  
PROVOZNĚ EKONOMICKÁ FAKULTA**



**Softwarová implementace matematických modelů  
s využitím PHP a SQL**

**Autor :** Lukáš Pavlík

**Vedoucí bakalářské práce :** Ing. Pelikán Martin Ph.D.

**2007 ©**

## **Prohlášení**

Prohlašuji, že jsem bakalářskou práci na téma: „Softwarová implementace matematických modelů s využitím PHP a SQL“ vypracoval samostatně a použil pouze pramenů, které uvádím v příloze seznamu.

V Praze:

.....

Podpis

## **Poděkování**

Děkuji svému vedoucímu bakalářské práce Ing. Martinu Pelikánovi Ph.D. za poskytnutý čas, který mi věnoval, za cenné odborné rady a metodické vedení při řešení mé bakalářské práce.

## **Souhrn**

Cílem této bakalářské práce bylo přiblížení matematického modelování a jeho realizace prostřednictvím informačních technologií. Zaměřuje se především na vytvoření webové aplikace pro řešení lineárních optimalizačních modelů pomocí simplexové metody. První část práce je orientovaná na teorii použitých postupů a technologií. Je zde popsán operační výzkum, včetně matematického modelování a lineárního programování, dále pak PHP(hypertextový preprocesor), SQL(strukturovaný dotazovací jazyk) a souborový formát CSV(hodnoty oddělené čárkami). Druhá popisuje vytvořenou aplikaci a v poslední je její uživatelský popis.

## **Klíčová slova**

Operační výzkum, matematické modelování, lineární optimalizace, simplexový algoritmus, SQL, PHP, CSV.

## **Summary**

The goal of this bachelor thesis is to describe mathematical modeling and its actual creation using information technologies. The thesis is particularly focused on programming of web-based application, which serve the purpose of solving linear optimization models using simplex method. The first part of the thesis explains the theory on which the implemented procedures and technologies are based on. The thesis describes operational research, including mathematical modeling and linear programming, as well as PHP(hypertext preprocessor), SQL(structured query language) and CSV(Comma-separated values) file format. The second part describes the created application itself and the last part contains the user's manual.

## **Key words**

Operational research, mathematical modeling, linear optimization, simplex algorithm, SQL, PHP, CSV.

## 1. Obsah

1.	<b>OBSAH</b> .....	- 2 -
2.	<b>ÚVOD</b> .....	- 4 -
3.	<b>CÍL PRÁCE A METODIKA</b> .....	- 5 -
4.	<b>LITERÁRNÍ REŠERŠE</b> .....	- 6 -
4.1	OPERAČNÍ VÝZKUM.....	- 6 -
4.1.1	<i>Matematické modelování</i> .....	- 6 -
4.1.2	<i>Lineární programování</i> .....	- 8 -
4.2	PHP .....	- 8 -
4.2.1	<i>Historie</i> .....	- 9 -
4.2.2	<i>Syntax</i> .....	- 9 -
4.3	SQL .....	- 11 -
4.3.1	<i>Historie</i> .....	- 11 -
4.3.2	<i>Příkazy jazyka SQL</i> .....	- 12 -
4.4	FORMÁT CSV .....	- 14 -
5.	<b>SOFTWAREVÁ IMPLEMENTACE MODELU</b> .....	- 16 -
5.1	NÁVRH MODELU.....	- 16 -
5.2	PŘEHLED SKRIPTŮ .....	- 16 -
5.2.1	<i>index.php</i> .....	- 16 -
5.2.2	<i>install.php</i> .....	- 17 -
5.2.3	<i>form.php</i> .....	- 17 -
5.2.4	<i>calculate_equation.php</i> .....	- 18 -
5.2.5	<i>solver.php</i> .....	- 19 -
5.2.6	<i>csv_export.php</i> .....	- 21 -
5.2.7	<i>inc_dbsetting.php</i> .....	- 21 -
5.2.8	<i>inc_header.php, inc_footer.php</i> .....	- 21 -
5.2.9	<i>inc_function.php</i> .....	- 22 -
5.2.10	<i>styles.css</i> .....	- 22 -

<b>6.</b>	<b>VÝSLEDKY .....</b>	<b>- 23 -</b>
6.1	PRVNÍ SPUŠTĚNÍ .....	- 23 -
6.2	ZADÁNÍ DAT .....	- 23 -
6.2.1	<i>Nové zadání</i> .....	- 24 -
6.2.2	<i>Načtení hodnot ze souboru CSV nebo databáze</i> .....	- 24 -
6.3	PŘEVEDENÍ OMEZUJÍCÍCH PODMÍNEK NA ROVNICE .....	- 25 -
6.4	OPTIMALIZACE MODELU .....	- 26 -
6.5	SROVNÁNÍ S PROGRAMEM LINEA .....	- 27 -
6.5.1	<i>Stručná charakteristika programu Linea 3.0</i> .....	- 27 -
6.5.2	<i>Srovnání</i> .....	- 27 -
<b>7.</b>	<b>ZÁVĚR.....</b>	<b>- 29 -</b>
<b>8.</b>	<b>SEZNAM OBRÁZKŮ .....</b>	<b>- 30 -</b>
<b>9.</b>	<b>SEZNAM TABULEK.....</b>	<b>- 31 -</b>
<b>10.</b>	<b>SEZNAM ZDROJŮ .....</b>	<b>- 32 -</b>
10.1	TIŠTĚNÉ ZDROJE .....	- 32 -
10.2	ELEKTRONICKÉ ZDROJE .....	- 32 -
<b>11.</b>	<b>SEZNAM POUŽITÉHO SOFTWARE .....</b>	<b>- 34 -</b>

## 2. Úvod

Během posledních let prudce stoupá obliba webových aplikací, které zastupují standardní programy na pracovních stanicích. Je to způsobeno velkým rozšířením internetu a stále rostoucí rychlostí připojení. Dalším významným faktorem je praktičnost řešení. Uživatel nemusí nic instalovat a navíc může aplikaci(i se stejným nastavením, jak je zvyklý) využívat na kterémkoliv počítači připojeným do světové sítě. Další výhody z toho vyplývající jsou například šetření výkonu a úložného prostoru stolního počítače, přímá aktualizace a oprava chyb v programu vývojáři apod.

Operační výzkum začal vznikat už před začátkem druhé světové války. Během ní a při následném poválečném rozvoji došlo k jeho velkému rozkvětu. Bylo to díky potřebě řešení strategických, taktických a ekonomických problémů. Jeho hlavním cílem je tedy nalezení neoptimálnějšího řešení. V dnešní uspěchané době, kdy se každý snaží co nejúčinněji zužitkovat svůj čas a peníze, jeho využití stále roste.

Cílem této práce je skloubit výše uvedené technologie a vytvořit webovou aplikaci na řešení úloh lineárního programování pomocí simplexové metody.

### 3. Cíl práce a metodika

Cílem této práce je vytvoření uživatelsky přívětivé a intuitivní aplikace pro řešení úloh lineární optimalizace. V průběhu výpočtu bude pro snazší pochopení postupu zobrazovat důležité hodnoty a naznačovat kroky.

Důraz bude dán na zadávání dat, které může uživatel buď zadat ručně, načíst z databáze nebo souboru CSV z programu lineá. Samozřejmostí je jednoduchá instalace, respektive nastavení přístupu k databázi MySQL, která se obejde bez přímého zásahu do vlastního kódu.

V teoretické části bude zmíněn význam operačního výzkumu, skriptovací jazyk PHP, dotazovací jazyk SQL a souborový formát CSV. Následovat bude popis jednotlivých částí programu včetně ukázky výstupů.



## 4. Literární řešerše

### 4.1 Operační výzkum

Operační výzkum je možné charakterizovat jako vědní disciplínu nebo spíše soubor relativně samostatných disciplín, které jsou zaměřeny na analýzu různých typů rozhodovacích problémů.

#### Historie

Není jednoduché a v podstatě ani možné datovat vznik operačního výzkumu jako samostatné disciplíny. Počátky operačního výzkumu spadají už do 30. a 40. let minulého století a jsou spjaty mimo jiné s takovými jmény jako je G B. Dantzig nebo nositel Nobelovy ceny za ekonomii L. Kantorovič. Rozvoj této disciplíny nastává jednak během 2. světové války, kdy byly vytvořeny ve Velké Británii a USA speciální týmy pracovníků pro analýzu složitých strategických a taktických vojenských problémů a operací, ale především potom během 50. let, ve kterých dochází ve světě k bouřlivému poválečnému ekonomickému rozvoji. Rozvoj operačního výzkumu a jeho jednotlivých disciplín vyplýval v tomto období skutečně z praktických potřeb - například některé metody a postupy dále obecně používané, byly vyvinuty v rámci konkrétních praktických studií. Dalším faktorem, ovlivňujícím rozvoj operačního výzkumu, je rozvoj výpočetní techniky. O rozvoji a významu operačního výzkumu svědčí i intenzita, s jakou jsou pořádány odborné konference, i to, jak vycházejí nové publikace a jsou pravidelně vydávány odborné časopisy speciálně s tímto zaměřením.<sup>1</sup>

#### 4.1.1 Matematické modelování

Základním nástrojem operačního výzkumu je matematické modelování. Pokud je tedy analyzován nějaký systém pomocí operačního výzkumu, potom tato analýza využívá model tohoto systému. Při analýze reálného systému prostřednictvím jeho modelu je třeba brát v úvahu, že model je pouze zjednodušeným obrazem tohoto

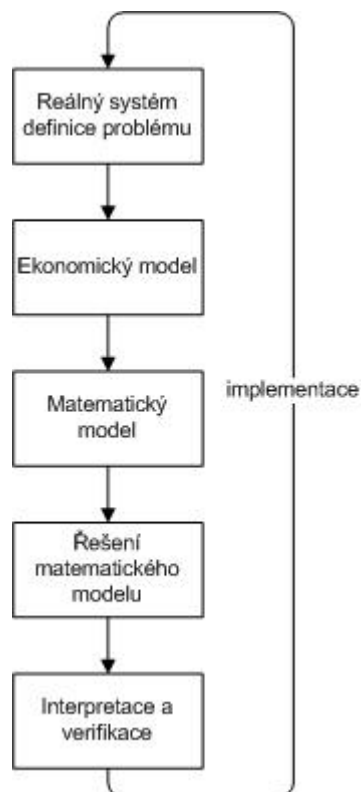
---

<sup>1</sup> Jablonský, J. Operační výzkum. 2002. str 9

systemu. Modelování však má celou řadu výhod, pro které se stává často jediným prostředkem pro studium modelovaného systému.

Jako základní výhody modelového přístupu uvést:

- použití matematických modelů umožňuje strukturalizaci systému a specifikaci všech možných variant stavu systému, kterých může být často neomezené množství
- modely umožňují analýzu chování systému ve zkráceném čase - procesy, které mohou trvat v reálném systému dny, měsíce či roky, mohou být simulovány na počítačích ve zlomcích sekund
- s modely lze snadno manipulovat a provádět četné experimenty pomocí změn jejich parametrů
- náklady na realizaci modelu nejsou sice zanedbatelné, jsou však vždy nižší než při experimentování s reálným systémem<sup>2</sup>



Obrázek 1: fáze aplikace operačního výzkumu

<sup>2</sup> Jablonský, J. Operační výzkum. 2002. str. 10

### 4.1.2 Lineární programování

Lineární programování je disciplína operačního výzkumu, která se zabývá řešením rozhodovacích problémů, ve kterých jde o určení intenzit realizace procesů, které probíhají nebo mohou probíhat v daném systému. Je přitom třeba respektovat všechny podmínky, které realizaci těchto procesů ovlivňují, a najít takové řešení, aby byl cíl rozhodování splněn co nejlépe. Termín lineární programování je složen ze dvou slov, která poměrně přesně vystihují podstatu tohoto odvětví operačního výzkumu. Slovo programování nemá v tomto termínu nic společného s jeho dnes běžným významem, kterým je programování počítačů v nějakém programovacím jazyku. Lineární programování má s počítači společné snad pouze to, že řešení jeho úloh není myslitelné bez použití výpočetní techniky. Programování je zde spíše synonymem pro plánování nebo vytváření programů (scénářů) budoucího vývoje. Slovo lineární vyjadřuje, že jsou všechny vazby v modelech lineárního programování vazbami lineárními, tzn. všechny matematické funkce, použité v těchto modelech, jsou funkce lineární. Lineární programování je tedy prostředkem pro plánování realizace určitých procesů (činností), který zabezpečuje dosažení optimálního výsledku ve vztahu k definovanému cíli.<sup>3</sup>

## 4.2 PHP

PHP je v současnosti velmi rozšířená technologie umožňující snadné programování na straně serveru (server-side programming). Toho lze využít k tvorbě různých interaktivních webových stránek. Stručně lze říci, že skript napsaný v PHP je proveden na serveru podle zadaných kritérií a výsledek je odeslán volajícímu počítači stejným způsobem, jakým se odesílají běžné statické (XHTML) stránky. Jakmile je však stránka načtena klientem, pomocí PHP ji již není možné dále měnit.<sup>4</sup>

---

<sup>3</sup> Jablonský, J. Operační výzkum. 2002. str. 19

<sup>4</sup> Rozsypal, P. Co je to PHP?[online]. 2006,[cit. 2007-05-08]. Dostupné z <<http://php.interval.cz/clanky/co-je-to-php/>>

### 4.2.1 Historie

Jazyk se vyvinul z jednoduchého systému v jazyce Perl pro evidenci přístupů na webové stránky. Kvůli zátěži webového serveru při spouštění interpretu Perlu byl autorem Rasmem Lerdorfem přepsán do jazyka C. Systém původně zamýšlel jen pro svoje osobní použití, ale posléze se zalíbil mnohým uživatelům. Autor tedy na základě požadavků na vylepšení systém rozšířil, doplnil o dokumentaci a následně uvolnil pod názvem Personál Home Page Tools (posléze Personál Home Page Construction Kit). Lendorf při postupném vývoji vytvořil nástroj umožňující přistupovat k databázi SQL, zpracování formulářů a zobrazování výsledků dotazů. Tento program nesl jméno Form Interpreter (FI).

Spojením těchto dvou děl vznikl systém PHP/FI 2.0, jehož kód se dal přímo zapisovat do webových stránek. Tato verze byla vydána jako open-source, aby se nejen opravily chyby, ale i umožnil rychlejší vývoj.

Následující verzi (PHP 3.0) přibyla podpora objektově orientované syntaxe, silnější a konzistentnější syntaxe jazyka a bylo ji již možné spustit v prostředí MS Windows. Postupným přidáváním funkcí a vylepšováním engine bylo dosaženo v současnosti nejnovější uvolněné verze 5.2.3.

### 4.2.2 Syntax

#### Zápis příkazů do HTML kódu

Kód jazyka PHP zapisuje mezi následující tagy:

- a) `<?php [kód jazyka PHP] ?>`
- b) `<? [kód jazyka PHP] ?>`
- c) `<script language="php"> [kód jazyka PHP] </script>`
- d) `<?=$proměnná_k_vypsání ?>`

První tři ukázky jsou ekvivalentní, ale pro přehlednost se doporučuje používat tag `ad a`. Druhý případ funguje pouze tehdy, jestliže je povolena konfigurační direktiva `short_open_tag`, která je v implicitní konfiguraci zapnuta. Poslední ukázka slouží k rychlému vypsání hodnoty proměnné.

## Proměnné

Proměnné v jazyce PHP začínají znakem \$ (dolar). Není nutné je předem deklarovat a přiřadit jim datový typ. Ten se přiřadí automaticky na základě obsahu a může se v průběhu programu měnit. To může být velké zjednodušení, ale zároveň to může způsobit problémy při odladování.

**Tabulka 1: datové typy v PHP**

Datový typ	Popis
boolean	Logické hodnoty(true/false)
integer	Celá čísla
float	Čísla s plovoucí desetinnou čárkou
string	Textový řetězec
array	Pole
object	Objekt
resource	Ukazatel na zdroj externích dat(otevření souboru, připojení k databázi apod)
NULL	Prázdný datový typ

## Komentáře

V prostředí jazyka PHP je možné používat komentáře z jazyka C++ a Unix/Linux shellu.

### *Ukázky komentářů:*

```
// řádkový komentář
```

```
# řádkový komentář
```

```
/* blokový  
komentář */
```

## 4.3 SQL

SQL je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. SQL je zkratka anglických slov Structured Query Language (strukturovaný dotazovací jazyk).<sup>5</sup>

### 4.3.1 Historie

V 70. letech 20. století probíhal ve firmě IBM výzkum relačních databází. Bylo nutné vytvořit sadu příkazů pro ovládání těchto databází. Vznikl tak jazyk SEQUEL (Structured English Query Language). Cílem bylo vytvořit jazyk, ve kterém by se příkazy tvořily syntakticky co nejbliže přirozenému jazyku (angličtině).

K vývoji jazyka se přidaly další firmy. V r. 1979 uvedla na trh firma Relational Software, Inc. (dnešní Oracle Corporation) svůj relační databázový systém Oracle. IBM uvedla v roce 1981 nový systém SQL/DS a v roce 1983 systém DB2. Dalšími systémy byly např. Progress, Informix a SyBase. Ve všech těchto systémech se používala varianta jazyka SEQUEL, který byl přejmenován na SQL.

Relační databáze byly stále významnější, a bylo nutné jejich jazyk standardizovat. Americký institut ANSI původně chtěl vydat jako standard zcela nový jazyk RDL. SQL se však prosadil jako de facto standard a ANSI založil nový standard na tomto jazyku. Tento standard bývá označován jako SQL-86 podle roku, kdy byl přijat.

V dalších letech se ukázalo, že SQL-86 obsahuje některé nedostatky a naopak v něm nejsou obsaženy některé důležité prvky týkající se hlavně integrity databáze. V roce 1992 byl proto přijat nový standard SQL-92 (někdy se uvádí jen SQL2). Zatím nejnovějším standardem je SQL3 (SQL-99), který reaguje na potřeby nejmodernějších databází s objektovými prvky.

Standardy podporuje prakticky každá relační databáze, ale obvykle nejsou implementovány vždy všechny požadavky normy. A naopak, každá z nich obsahuje

---

<sup>5</sup> SQL[online]. 2007,[cit 2007-04-12]. Dostupné z <<http://cs.wikipedia.org/wiki/SQL>>

prvky a konstrukce, které nejsou ve standardech obsaženy. Přenositelnost SQL dotazů mezi jednotlivými databázemi je proto omezená.<sup>6</sup>

### 4.3.2 Příkazy jazyka SQL

Jazyk SQL v sobě zahrnuje jazyk pro definici dat (DDL), jazyk pro manipulaci s daty (DML), jazyk pro řízení dat (DCL) a příkazy umožňující transakce (TCC). SQL není proto jen prostředek pro dotazování, je možné v něm rovněž i definovat data a provádět jejich aktualizaci, tak jak je obvyklé u každého SŘBD.<sup>7</sup>

#### DDL - Data Definition Language

Příkazy pro definici dat představují množinu příkazů pro vytváření struktury databáze, tj. tabulek, indexů a ostatních objektů.<sup>8</sup>

Tabulka 2: příkazy SQL DDL

CREATE DATABASE	CREATE VIEW	DROP SEQUENCE
CREATE TABLE	ALTER VIEW	CREATE PROCEDURE
ALTER TABLE	DROP VIEW	DROP PROCEDURE
DROP TABLE	DROP INDEX	CREATE TRIGGER
CREATE INDEX	CREATE SEQUENCE	DROP TRIGGER
DROP INDEX	ALTER SEQUENCE	

#### *Ukázka kódu: vytvoření databáze*

```
CREATE TABLE `simplex` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `save_date` datetime NOT NULL default '0000-00-00
00:00:00',
  `ip` varchar(15) NOT NULL default '',
```

<sup>6</sup> SQL[online]. 2007,[cit 2007-04-12]. Dostupné z <<http://cs.wikipedia.org/wiki/SQL>>

<sup>7</sup> Vostrovský, V. Vytváření databází Oracle. 2005. str. 58

<sup>8</sup> Vostrovský, V. Vytváření databází Oracle. 2005. str. 61

```

`a` text NOT NULL,
`b` tinytext NOT NULL,
`term` tinytext NOT NULL,
`uf` tinytext NOT NULL,
`uf_type` enum('0','1') NOT NULL default '0',
`description` text,
PRIMARY KEY (`id`)
) TYPE=MyISAM COMMENT='Tabulka ulozenych simplexu'
AUTO_INCREMENT=5 ;

```

## DML - Data Manipulation Language

Příkazy pro manipulaci s daty tvoří jádro jazyka SQL a umožňují vyjádřit v jaké podobě je možno data z databáze získat nebo jakým způsobem existující data upravit.<sup>9</sup>

**Tabulka 3: příkazy SQL DML**

SELECT	INSERT	UPDATE
DELETE		

### *Ukázka kódu: načtení simplexu*

```

SELECT a, b, term, uf, uf_type
FROM simplex
WHERE id=15;

```

## DCL - Data Control Language

Příkazy pro řízení dat umožňují různým způsobem zpřístupňovat data jednotlivým uživatelům a v případě práce s rozsáhlými databázovými systémy členit činnosti do logických celků - transakcí.<sup>10</sup>

<sup>9</sup> Vostrovský, V. Vytváření databází Oracle. 2005. str. 62

<sup>10</sup> Vostrovský, V. Vytváření databází Oracle. 2005. str. 62



Tabulka 4: příkazy SQL DCL

GRANT	ALTER USER	REVOKE GRANT USER
DROP USER	REVOKE	

### TCC - Transaction Control Commands

Transakce je definována jako posloupnost příkazů SQL formující jednotku zotavení z chyb a souběžného zpracování. Každá transakce končí vyvoláním buď příkazu COMMIT WORK (normální ukončení) nebo ROLLBACK WORK (nenormální ukončení). V případě chyby či splnění dané podmínky se zpracování transakce přeruší a databáze se uvede do stavu před transakcí, v opačném případě dojde k provádění příkazů a k dokončení transakce a vytvoření permanentní verze nové databáze.<sup>11</sup>

Tabulka 5: příkazy SQL TCC

SET TRANSACTION	ROLLBACK	COMMIT
SAVEPOINT		

## 4.4 Formát CSV

CSV (Comma-separated values, hodnoty oddělené čárkami) je jednoduchý souborový formát určený pro výměnu tabulkových dat. Soubor ve formátu CSV sestává z řádků, ve kterých jsou jednotlivé položky odděleny znakem čárka (.). Hodnoty položek mohou být uzavřeny do uvozovek ("), což umožňuje, aby text položky obsahoval čárku. Pokud text položky obsahuje uvozovky, jsou tyto zdvojeny.

Jelikož se v některých jazycích včetně češtiny čárka používá v číslech jako oddělovač desetinných míst, existují varianty, které používají jiný znak pro oddělování položek než čárku, nejčastěji středník, případně tabulátor (taková varianta se pak někdy označuje jako TSV, Tab-separated values). Variantu se středníkem (ale stále pod názvem CSV) používá např. česká verze Microsoft Excel.

<sup>11</sup> Vostrovský, V. Vytváření databází Oracle. 2005. str. 62

Díky jednoduchosti, nenáročnosti a čitelnosti i bez specializovaného software se tento formát používá pro výměnu informací mezi různými systémy. Ke stejnému účelu se dnes používá i modernější a univerzálnější (ale složitější) formát XML.

Pro tento formát neexistuje specifikace, popis formátu se však nachází v RFC 4180(<http://www.faqs.org/rfcs/rfc4180.html>).<sup>12</sup>

---

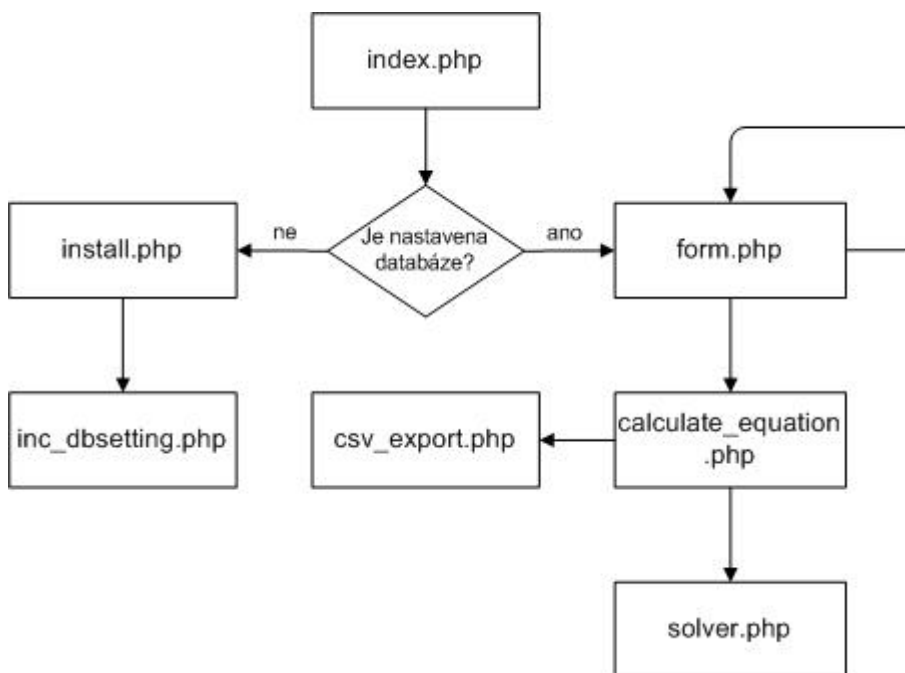
<sup>12</sup> CSV[online]. 2007,[cit 2007-03-8] Dostupné z <<http://cs.wikipedia.org/wiki/CSV>>

## 5. Softwarová implementace modelu

### 5.1 Návrh modelu

Samotná optimalizace úlohy probíhá ve skriptech *form.php*, *calculate\_equation.php* a *solver.php*. Komunikaci s dalšími soubory znázorňuje obrázek 2.

Z důvodu předávání proměnných pomocí serverových proměnných session, je třeba mít na straně klienta zapnuté přijímání cookies.



Obrázek 2: struktura programu

### 5.2 Přehled skriptů

#### 5.2.1 index.php

Výchoze zobrazený soubor v adresáři. Obsahuje stručné informace o projektu a při prvním spuštění odkaz na *install.php*(po nastavení db změna na *form.php*)

### 5.2.2 install.php

Pomocí tohoto skriptu je možné bez úpravy konfiguračních souborů pomocí webového rozhraní jednoduše nastavit parametry MySQL serveru. Je ale třeba, aby bylo možno pomocí PHP funkce zapisovat do souboru *inc\_dbsetting.php*. Skript je spuštěn pouze jednou pomocí odkazu v *index.php*, který se poté změní na pokračování dále na stránky.

Při opětovném spuštění je změna nastavení z bezpečnostních důvodů zakázána. Je tedy pak nutné případnou změnu provést prostřednictvím editace *inc\_dbsetting.php*.

### 5.2.3 form.php

Nejprve vykreslí základní formulář k zadání počtu proměnných a omezujících podmínek, formulář ke specifikaci souboru CSV a seznam uložených simplexů.

Odeslání formuláře nebo klik na uložený simplex způsobí nové spuštění skriptu s tím rozdílem, že se vypíše jiný formulář pro zadání konkrétních hodnot, kam se případně načtou hodnoty z dříve uloženého simplexu. Při zadání nového simplexu je možné navíc uložit popis.

Tabulka 6: form.php - konstanty

Konstanta/proměnná	Index	Výchozí hodnota	Účel
MAX_DIGIT_FIELD	-	5	Velikost a maximální počet znaků v textovém poli
DEF_FORM_VAL	-	0	Výchozí hodnota textových polí
DIVIDE_CHAR	-	&times	Znak pro násobení
\$gtlt	1	&le	Znak $\leq$
	0	=	Znak =
	-1	&ge	Znak $\geq$

Tabulka 7: form.php - propojení se skripty

Směr(metoda)	Skript	Parametry	Účel
IN (POST)	Index.php	-	
IN/OUT (POST)	Form.php	\$variables	Počet proměnných
		\$constrains	Počet omezujících podmínek
		\$action(=load)	Akce při načtení
		\$id	Identifikátor záznamu v databázi
OUT (GET)	calculate_equation.php	\$aa[řádek][sloupec]	Matice A
		\$bb[řádek]	Vektor pravých stran
		\$term[řádek]	Typ nerovnic
		\$uff[sloupec]	Účelová funkce
		\$db	Uložení do db
		\$desc	Popis uloženého simplexu

#### 5.2.4 calculate\_equation.php

V případě požadavku na uložení hodnot do databáze hodnoty uloží. Následně si skript převede omezující podmínky na rovnice, doplní doplňkové a pomocné proměnné a vše uživateli vypíše. Zároveň se započne session(sezení) pomocí kterého se předají vypočtené pole(matice) a proměnné skriptu **solver.php**.

Tabulka 8: calculate\_equation.php - deklarované funkce

Funkce	Návratová hodnota	Parametry	Činnost
prohibitive_rates	integer	\$pole	Vrátí prohibitivní sazbu pro danou matici
print_equations	-	\$a, \$b, \$term, \$vals	Vypíše rovnici o daných parametrech

Tabulka 9: calculate\_equation.php - propojení se skripty

Směr	Skript	Parametry	Datový typ	Metoda	funkce
IN	Form.php	Viz výše	Viz výše	Viz výše	Viz výše
OUT	Solver.php	\$a	Integer (2 rozměrné pole)	SESSION	Matice A
		\$b	Integer(pole)	SESSION	Vektor pravých stran
		\$uf	Integer/string(2 rozměrné pole)	SESSION	Účelová funkce a řetězec s proměnnými
		\$uf_type	boolean	SESSION	Typ účelové fce

### 5.2.5 solver.php

Skript spočítá všechny hodnoty zadaného simplexu a ten následně vypíše. V dalším kroku zkusí vylepšit hodnotu účelové funkce, dopočítá hodnoty a také vypíše. Ke každému kroku navíc zobrazí důležité hodnoty. Tyto operace se provádějí tak dlouho, dokud není řešení optimální nebo nepřípustné.

Tabulka 10: solver.php - konstanty

Konstanta/proměnná	Výchozí hodnota	Účel
ROUND_NUM_SIMPLEX	3	Počet desetinných míst pro zaokrouhlení hodnot simplexové tabulky

OMEGA	&Omega;	Znak omegy
PRAZDNA_MNOZINA	&Oslash	Znak pro prázdnou množinu
MAX_STEP	25	Počet kroků simplexu k zastavení výpočtu

Tabulka 11: solver.php - deklarované funkce

Funkce	Návratová hodnota	Parametry	Činnost
improve_simplex	-	-	Vylepší hodnotu účelové funkce
calculate_simplex	-	-	Spočte hodnoty simplexové tabulky
show_simplex	-	-	Vykreslí simplexovou tabulku
omega_count	float	\$ks_val, \$b_val, \$maxmin	Spočte omegu
find_base	array	\$row	Najde bázi v matici A
abs_sum_col	float	\$col	Spočte sumu absolutních hodnot sloupce
index_extrem_val	integer	\$minmax, \$array	Vrátí index extrémní hodnoty předaného pole
alter_array_items	-	&\$item, \$key, \$operation	Provede operaci nad prvky předaného pole
alter_array_items_ex	-	&\$item, \$key, \$konst	Provede operaci nad prvky předaného pole
show_info	-	\$base_origin, \$poradi=0	Zobrazí důležité hodnoty simplexové tabulky v daném kroku
show_matrix	-	\$matrix, \$sep="; "	Vypíše matici
cnt_matrix_trans	Array	\$pole, \$indexy	Vrací matici transformace

is_var_base	Boolean	\$baze	Zjistí, zda-li je v bázi pomocná proměnná
is_degenerated	Boolean	\$b	Zjistění degenerovaného řešení
kriterium_pripustnosti	Boolean	\$omega	Zjistí splnění kritéria přípustnosti

### 5.2.6 csv\_export.php

Odešle uživateli hodnoty omezujících podmínek ve formátu CSV. Ten je možné otevřít v programu Linea nebo opět načíst pomocí tohoto projektu.

Proměnné jsou předány skriptem *calculate\_equation.php* pomocí session. Po sestavení obsahu souboru je prohlížeči odeslaná HTTP hlavička s MIME typem text/csv a následně i obsah samotného souboru.

Tabulka 12: csv\_export.php - konstanty

Konstanta	Výchozí hodnota	Účel
NR	\r\n	CR
RS	=	znak rovnítka
SEP	;	oddělovač CSV
VARR	X	proměnná

### 5.2.7 inc\_dbsetting.php

Slouží k uchování parametrů připojení k SQL serveru a stavové proměnné \$db['inst'] pro indikaci provedené konfigurace. Před instalací databáze pomocí *install.php* je soubor prázdný.

Jeho obsah se vkládá ve funkci *connect\_dbase()* a v souboru *index.php*.

### 5.2.8 inc\_header.php, inc\_footer.php

Obsahují jednotnou hlavičku a patku stránky vkládanou do všech skriptů.



### 5.2.9 inc\_function.php

Slouží jako knihovna funkcí a konstant.

Tabulka 13: inc\_function.php - konstanty

Konstanta	Hodnota	Účel
EQ	0	rovno
GT	-1	Větší než
LT	1	Menší než
MAX	1	Maximalizace UF
MIN	-1	Minimalizace UF
TXT	0	Textová hodnota UF
VAL	1	Číselná hodnota UF

### 5.2.10 styles.css

Stylovis kaskádových stylů.

## 6. Výsledky

### 6.1 První spuštění

Při prvním vstupu na stránky se zobrazí úvodní strana se základními údaji. Jestliže ještě nebyla provedena instalace databázi, ukáže se odkaz pro pokračování. V opačném případě bude na stejném místě výzva k instalaci, pomocí níž může uživatel snadno aplikaci zprovoznit. Na následující stránce se vypíše formulář pro zadání základních údajů o databázovém serveru, po jehož korektním vyplnění a odeslání je již možno s aplikací pracovat.

#### Nastavení databáze

MySQL server	<input type="text"/>
Login	<input type="text"/>
Heslo	<input type="text"/>
Databáze	<input type="text"/>
	<input type="button" value="Ulož"/>

Obrázek 3: první spuštění

### 6.2 Zadání dat

Stránka je rozdělena na tři sekce podle způsobu načtení dat. V horní části se nachází formulář, kam uživatel zadá počet proměnných a omezujících podmínek, níže je možné načtení hodnot ze souboru CSV ve formátu programu Linea a poslední varianta je nahrání dat z databáze.

**Vložení funkcí**

Počet proměnných

Počet omezujících podmínek

**Soubor z Linei**

Vložte cestu k souboru(\*.csv)

**Uložené simplex**

	Datum	Popis
<input type="button" value="Smaž"/>	<a href="#">2007-03-02 02:29:26</a>	Učebnicový příklad
<input type="button" value="Smaž"/>	<a href="#">2007-06-25 12:43:28</a>	Problém zemědělce(skripta)
<input type="button" value="Smaž"/>	<a href="#">2007-06-25 12:36:55</a>	problém trenára(skripta)

Obrázek 4: způsoby zadání hodnot

### 6.2.1 Nové zadání

Na základě zadání počtu proměnných a omezujících podmínek, skript zobrazí další formulář, kde je možné vložit konkrétní hodnoty optimalizačního modelu. Uživatel zde může simplex uložit do databáze a přidat krátký popis.

### 6.2.2 Načtení hodnot ze souboru CSV nebo databáze

Uložené modely se načítají do stejného formuláře, jako v případě zadání nového. Uživatel si díky tomu může snadno hodnoty předchozích vstupů změnit a sledovat změnu optimalizace.

### Zadejte požadované hodnoty

#### Omezující podmínky:

<input type="text" value="1"/>	$\times x_1 +$	<input type="text" value="1"/>	$\times x_2 +$	<input type="text" value="1"/>	$\times x_3$	<input type="text" value="≤"/>	<input type="text" value="1000"/>
<input type="text" value="0.2"/>	$\times x_1 +$	<input type="text" value="0.3"/>	$\times x_2 +$	<input type="text" value="0.5"/>	$\times x_3$	<input type="text" value="≤"/>	<input type="text" value="240"/>
<input type="text" value="0"/>	$\times x_1 +$	<input type="text" value="0"/>	$\times x_2 +$	<input type="text" value="1"/>	$\times x_3$	<input type="text" value="≥"/>	<input type="text" value="200"/>

#### Účelová funkce

 $Z(x) =$  $\times x_1 +$  $\times x_2 +$  $\times x_3 \rightarrow$ Uložit do databáze 

Obrázek 5: načtení hodnot z databáze nebo CSV

### 6.3 Převedení omezujících podmínek na rovnice

V dalším kroku se zobrazí původní omezení a vyrovnaný model pomocí doplňkových, popř. pomocných proměnných, spolu s účelovou funkcí. Uživatel zde může stáhnout zadání ve formátu CSV nebo přistoupit k optimalizaci.

**Omezující podmínky:**

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 1000 \\ 0.2x_1 + 0.3x_2 + 0.5x_3 &\leq 240 \\ x_3 &\geq 200 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

**účelová funkce:**

$$6x_1 + 7x_2 + 4x_3 \rightarrow \text{MAX}$$

**Rovnicový kanonický tvar:**

$$\begin{aligned} x_1 + x_2 + x_3 + d_1 &= 1000 \\ 0.2x_1 + 0.3x_2 + 0.5x_3 + d_2 &= 240 \\ x_3 - 1d_3 + p_1 &= 200 \\ x_1, x_2, x_3, d_1, d_2, d_3, p_1 &\geq 0 \end{aligned}$$

**účelová funkce:**

$$6x_1 + 7x_2 + 4x_3 - 100p_1 \rightarrow \text{MAX}$$

Obrázek 6: převod omezujících podmínek

## 6.4 Optimalizace modelu

Tuto část vykonává skript *solver.php*, který obdrží matici soustavy, vektor pravých stran a účelovou funkci. Model postupně vylepšuje, dokud není řešení buď optimální nebo do zjištění nepřipustnosti řešení. Uživateli ke každému kroku vypíše simplexovou tabulku a v pravém sloupci její důležité hodnoty. V tabulce zvýrazní klíčový sloupec i řádek.

**Krok 1**

	1	2	3	10	0	0	1000	1000			
$c_b^t$	$x_b$	$x_1$	$x_2$	$x_3$	$x_4$	$d_1$	$d_2$	$p_1$	$p_2$	$b$	$\Omega$
3	$x_3$	0.05	0.063	1	0.125	-0.006	0	0.006	0	5	40
1000	$p_2$	1	1	0	2.5	0	0	0	1	2	0.8
$Z_j - c_j$		999.15	998.188	0	2490.375	-0.019	0	-999.981	0	2015	MIN

**Důležité hodnoty**

Hodnota účelové funkce  $Z^1 = 2015$

Bazické proměnné:  $x_3=5$

$p_2=2$

V bázi je pomocná proměnná

Maticе transformace  $B_1^{-1} = \begin{pmatrix} 0.006; 0 \\ 0; 1 \end{pmatrix}$

Obrázek 7: optimalizace simplexu

## 6.5 Srovnání s programem Linea

### 6.5.1 Stručná charakteristika programu Linea 3.0

Linea je aplikace pro MS Windows vytvořená v programu VisualWorks SmallTalk Ing. Janem Bartoškou. Poslední verze 3.0 z roku 2005 je dostupná na adrese <http://oikt.czu.cz/~bartoska/linea/>

Okno programu je rozděleno na tři části. V levé části se zobrazují předchozí zadané úlohy, v pravé je možné zadat simplex a spodní slouží pro zobrazení výsledků. Aplikace umožňuje export úlohy s postupem do formátu html. Z důvodu absence nápovědy, zde nejsou uvedeny všechny schopnosti aplikace Linea.

The screenshot shows the LINEA 3.0 application window. The top section contains the problem definition:

- Objective function:**  $+6x_1 + 7x_2 + 4x_3$  (set to minimum)
- Restrictive conditions:**
  - $+1x_1 + 1x_2 + 1x_3 < 1000$
  - $+(1/5)x_1 + (3/10)x_2 + (1/2)x_3 < 240$
  - $+1x_3 > 200$

The bottom section displays the Simplex table:

minimization		6	7	4	0	0	0
Base		$x_1$	$x_2$	$x_3$	Slack1	Slack2	Slack3
0	Slack1	1	1	1	1	0	0
0	Slack2	$[1/5]$	$[3/10]$	$[1/2]$	0	1	0
+M	Art3	0	0	1	0	0	-1
Zj - Cj		-6	-7	$-4 + 1^* + M$	0	0	$-1^* + M$

Obrázek 8: ukázka programu Linea 3.0

### 6.5.2 Srovnání

Výhodou programu Simplex je jeho snadné použití na téměř jakékoliv platformě, protože se provádí na webovém serveru a uživateli je zaslána hotová webová

stránka. Uživatel nemusí nic stahovat, stačí zadat adresu <http://bcwork.wz.cz> do prohlížeče. Z výše uvedeného vyplývá, že aplikace bude fungovat i na počítači s nastavenou velmi striktní bezpečnostní politikou. Dále program zobrazuje více informací během prováděné optimalizace a umožňuje uživateli snadnější pochopení algoritmu.

Naopak velkou výhodou programu Linea 3.0 je práce se zlomky. Desetinná čísla zpracovává a ukládá jako zlomky, čímž se redukuje případné nepřesnosti. Obsahuje navíc funkci pro analýzu citlivosti. Drobná nevýhoda je ve zvoleném formátu exportu. V souboru CSV není uložena informace o typu optimalizace(MIN, MAX).

## 7. Závěr

Cílem této práce bylo pomocí skriptovacího jazyku PHP a dotazovacího jazyku SQL vytvořit aplikaci pro optimalizaci úloh lineárního programování. Dále byly použity kaskádové styly z důvodu usnadnění formátování.

V průběhu vývoje aplikace autor zjistil, že by bylo vhodnější programovat objektově orientovaný kód, ale kvůli špatné podpoře v jazyce PHP nebyl tento nápad realizován.

Programový kód je srozumitelně okomentován, takže je možné v budoucnu přidat další funkce, například pro grafické řešení, či analýzu citlivosti apod.

Výsledná aplikace je v době odevzdání umístěna na serveru Webzdarma.cz na adrese <http://bcwork.wz.cz>. Jeho provozovatel si ve všeobecných podmínkách vyhrazuje právo web po měsíční neaktivitě vymazat. Z toho důvodu nemůže autor po uplynutí této doby garantovat přístupnost stránek.



## 8. Seznam obrázků

Obrázek 1: fáze aplikace operačního výzkumu.....	- 7 -
Obrázek 2: struktura programu .....	- 16 -
Obrázek 3: první spuštění.....	- 23 -
Obrázek 4: způsoby zadání hodnot .....	- 24 -
Obrázek 5: načtení hodnot z databáze nebo CSV .....	- 25 -
Obrázek 6: převod omezujících podmínek.....	- 26 -
Obrázek 7: optimalizace simplexu .....	- 26 -
Obrázek 8: ukázka programu Linea 3.0 .....	- 27 -

## 9. Seznam tabulek

Tabulka 1: datové typy v PHP.....	- 10 -
Tabulka 2: příkazy SQL DDL.....	- 12 -
Tabulka 3: příkazy SQL DML.....	- 13 -
Tabulka 4: příkazy SQL DCL.....	- 14 -
Tabulka 5: příkazy SQL TCC.....	- 14 -
Tabulka 6: form.php - konstanty.....	- 17 -
Tabulka 7: form.php - propojení se skripty.....	- 18 -
Tabulka 8: calculate_equation.php - deklarované funkce.....	- 19 -
Tabulka 9: calculate_equation.php - propojení se skripty.....	- 19 -
Tabulka 10: solver.php - konstanty.....	- 19 -
Tabulka 11: solver.php - deklarované funkce.....	- 20 -
Tabulka 12: csv_export.php - konstanty.....	- 21 -
Tabulka 13: inc_function.php - konstanty.....	- 22 -

## 10. Seznam zdrojů

### 10.1 Tištěné zdroje

- 1) BROŽOVÁ, Helena - HOUŠKA, Milan,. Základní metody operační analýzy. 1. vydání. Praha: ČZU. 2003. 244 s. ISBN 8021309512.
- 2) JABLONSKÝ, J. Operační výzkum. Praha:Profesional Publishing, 2002. ISBN 80-86419-42-8
- 3) KOSEK, Jirí. PHP - Tvorba interaktivních internetových aplikací. 1. vydání. Praha: Grada, 1999. 490 s. ISBN 8071693731.
- 4) ŠUBRT, T., BROŽOVÁ, H., DÖMEOVÁ, L., KUČERA, P. Ekonomicko matematické metody II, aplikace a cvičení. Praha:ČZU, 2005. ISBN 80-213-0721-8
- 5) VOSTROVSKÝ, V.: Vytváření databází Oracle. Praha:ČZU, 2005. 134s. ISBN 80-213-1191-6
- 6) WELLING, Luke - THOMSON, Laura. PHP a MySQL - rozvoj webových aplikací, 2. vydání. Praha: SoftPress, 2003. 830s. ISBN 8086497607
- 7) ZÍSKAL, J., HAVLÍČEK, J.: Ekonomicko matematické metodyII, studijní texty pro distanční studium. Praha:ČZU, 2003. ISBN 80-213-0664-5

### 10.2 Elektronické zdroje

- 1) CSV[online]. 2007.[cit 2007-03-08] . Dostupné z <<http://cs.wikipedia.org/wiki/CSV>>
- 2) MIT, INRIA, Keio, HTML 4.01 Specification[online]. 1999. [cit. 2007-02-10]. Dostupné z <<http://www.w3.org/TR/html4/>>
- 3) MOORE, J., FOX, S.,ZMIEVSKY, A. Historie PHP[online]. [cit 2007-03-08]. 2006. Dostupné z <http://gtk.php.net/manual1/cs/html/intro.whatis.php.history.html>>
- 4) MySQL AB:: Developer Zone[online].2007.[cit 2007-03-08].Dostupné z <<http://www.mysql.org/>>
- 5) PHP: Hypertext preprocesor[online].2007. [cit 2007-03-08]. Dostupné z <<http://www.php.net>>
- 6) ROZSYPAL, Petr. Co je to PHP?[online]. 2006. [cit. 2007-05-10]. Dostupné z <<http://php.interval.cz/clanky/co-je-to-php/>>

- 7) SQL[online]. 2007,[cit 2007-04-12]. Dostupné z  
<<http://cs.wikipedia.org/wiki/SQL>>
- 8) ŠEPTÁKOVÁ, E., Transakce v SQL[online]. 2002. [cit 2007-05-10]. Dostupné z  
<<http://www.cs.vsb.cz/amalka/tomica/sql/kap8.html#k3>>
- 9) VRÁNA, J., HOJTSY, G., JELÍNEK, L. Manuál PHP[online]. [cit 2007-03-08].  
Dostupné z <<http://edu.uhk.cz/PHP/index.html>>
- 10) ZAJÍC, P., PHP tutorial[online].2006. [cit 2007-05-10]. Dostupné z  
<<http://www.linuxsoft.cz/php/>>

## **11. Seznam použitého software**

### **Pracovní stanice**

MS Windows XP Professional

Macromedia Dreamweaver 8

ComponentSoftware CS-RCS PRO 5.0

MS Office Word 2003

MS Office Visio 2003

Mozilla Firefox v2.0.0.4

### **Testovací server:**

Linux Slackware 10.1.0

Apache 1.3.33

PHP Version 4.3.11

MySQL 4.0.24

### **Veřejný server(webzdarma.cz):**

Apache 2.0.59

PHP Version 4.3.4

MySQL 4.0.15