

## ZADÁNÍ SEMESTRÁLNÍ PRÁCE

### Nástroj pro řešení úloh lineárního programování

Na moment si představte, že jste vedoucí v podniku, který generuje dva produkty:  $A$  a  $B$ . Ty je třeba vyrobit a zabalit. Výroba produktu  $A$  trvá 1 hodinu a balení další 2 hodiny, zatímco výroba produktu  $B$  trvá 2 hodiny a balení 1 hodinu. Pracovní doba výrobního oddělení je 8 hodin. Pracovní balírny si ovšem se svým vedoucím kvůli náročnému úklidu domluvili 6hodinovou pracovní dobu. Zisk za jeden kus produktu  $A$  jsou 3 jednotky, za produkt  $B$  inkasuje podnik 2 jednotky. Vaším úkolem je zjistit, kolik kusů produktu  $A$  a  $B$  by podnik měl vyrobit a zabalit, aby maximalizoval zisk. Grafické znázornění úlohy je ukázáno na obrázku 1.

V této úloze lineárního programování tedy maximalizujeme účelovou funkci

$$z = 3x_1 + 2x_2,$$

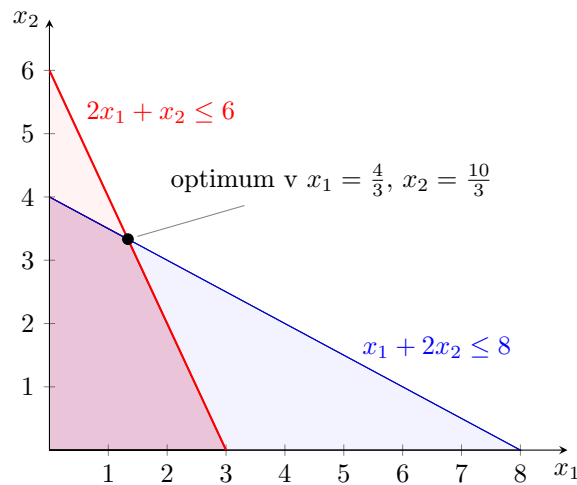
kde  $x_1$ , resp.  $x_2$  jsou počty kusů vytvořených produktů  $A$ , resp.  $B$ . Funkci optimalizujeme s ohledem na omezující podmínky:

$$x_1 + 2x_2 \leq 8 \text{ (výrobní čas),}$$

$$2x_1 + x_2 \leq 6 \text{ (čas balení),}$$

$$x_1 \geq 0,$$

$$\text{a } x_2 \geq 0.$$



Obrázek 1: Grafické znázornění úlohy lineárního programování.

Pokud bychom v této úloze nepřipouštěli rozdělané výrobky na konci směny, tj. hledané řešení musí být celočíselné, jednalo by se o úlohu celočíselného lineárního programování, která je ovšem NP-těžká. V této semestrální práci se budeme držet při zemi: budeme řešit úlohy lineární optimalizace s reálným řešením a pouze jednou účelovou funkcí.

Pomocí lineárního programování je možné řešit takřka nekonečné množství reálných problémů. Od optimalizace procesů, dopravních sítí, přidělování vysílacích frekvencí, logistiky, plánování zdrojů, genomiky a mnoho dalšího.

## Zadání

Naprogramujte v jazyce ANSI C přenositelnou<sup>1</sup> **konzolovou aplikaci**, která bude řešit úlohy lineárního programování zadané ve zjednodušeném formátu LP.

Program bude spouštěn příkazem `lp.exe`<sup>2</sup> s kombinací následujících argumentů – výrazy v lomných závorkách (`<>`), resp. hranatých závorkách (`[]`) označují povinné, resp. nepovinné argumenty (příklad spuštění programu je uveden v ukázce konzolového výstupu 1):

- |                                    |   |
|------------------------------------|---|
| <code>&lt;input-file&gt;</code>    | Soubor s popisem úlohy ve formátu LP. V případě, že uživatel zadá neexistující soubor, program vypíše chybové hlášení <code>"Input file not found!\n"</code> a vrátí hodnotu 1.   |
| <code>-o &lt;path&gt;</code>       | Výstupní soubor s řešením úlohy. Pokud umístění neexistuje, bude vypsáno hlášení <code>"Invalid output destination!\n"</code> a program skončí s návratovou hodnotou 2. V případě, že uživatel tento přepínač nezadá, bude výsledek optimalizace vypsán na obrazovku. Do tohoto souboru neuvádějte chybová hlášení. |
| <code>--output &lt;path&gt;</code> | Stejně jako v případě přepínače <code>-o</code> . Použití obou přepínačů <code>-o</code> a <code>--output</code> není chybou, program pak bude akceptovat poslední zadanou hodnotu.   |

V případě nalezení konečného optimálního řešení úlohy program vrátí hodnotu `EXIT_SUCCESS`. Chybové stavy týkající se zpracování vstupních souborů nebo samotného algoritmu optimalizace jsou popsány v dalších sekcích.

Hotovou práci odevzdejte v jediném archivu typu ZIP prostřednictvím automatického odevzdávacího a validačního systému. Postupujte podle instrukcí uvedených na webu předmětu. Archiv nechť obsahuje všechny zdrojové soubory potřebné k přeložení programu, **Makefile** pro Windows i Linux (pro překlad v Linuxu připravte soubor pojmenovaný **Makefile** a pro Windows **Makefile.win**) a dokumentaci ve formátu PDF vytvořenou v typografickém systému  $\text{\TeX}$  ( $\text{\LaTeX}$ ). Bude-li některá z částí chybět, kontrolní skript Vaši práci odmítne.

## Specifikace vstupních souborů (formát LP)

Pro zachycení optimalizačního modelu bude program používat redukovanou a zobecněnou verzi formátu LP, který je popsán v [1]. Vstupní soubory mohou obsahovat následující sekce:

- |                          |   |
|--------------------------|---|
| <b>Maximize/Minimize</b> | Výraz uvozující řádek se zápisem optimalizované účelové funkce. Program musí být schopen zpracovat standardní operátory <code>+</code> , <code>-</code> , <code>*</code> , <code>=</code> nebo závorky <code>()</code> , <code>[]</code> a <code>{}</code> . Oproti originální verzi ovšem nevyžadujeme, aby jednotlivé operandy a operátory byly v matematických výrazech striktně odděleny mezerou (to platí i v ostatních sekcích souboru). Názvy proměnných tedy dříve uvedené operátory a závorky obsahovat nesmí. Při násobení není nutné použít operátor <code>*</code> , například <code>"2.5z"</code> značí <i>2,5 krát z</i> , zatímco <code>"z2"</code> je pouze název proměnné. |
| <b>Subject To</b>        | Sekce obsahující seznam podmínek ve formátu <code>"&lt;název&gt;: &lt;výraz&gt;".</code> Navíc oproti účelové funkci mohou podmínky obsahovat porovnávací operátory <code>&lt;</code> , <code>&gt;</code> , <code>&lt;=</code> a <code>&gt;=</code> .   |

<sup>1</sup>Je třeba, aby bylo možné váš program přeložit a spustit na PC s operačním prostředím Win32/64 (tj. operační systémy Microsoft Windows NT/2000/XP/Vista/7/8/10/11) a s běžnými distribucemi Linuxu (např. Ubuntu, Debian, Red Hat, atp.). Server, na který budete vaši práci odevzdávat a který ji otestuje, má nainstalovaný operační systém Debian GNU/Linux 11 (bullseye) s jádrem verze 5.10.0-28-amd64 a s překladačem gcc 10.2.1.

<sup>2</sup>Přípona `.exe` je povinná i při sestavení pro Linux, zejména při automatické kontrole validačním systémem.

<b>Bounds</b>	Omezení hodnot rozhodovacích proměnných. V této sekci jsou povoleny pouze porovnávací operátory uvedené výše.
<b>Generals</b>	Obsahuje seznam použitých rozhodovacích proměnných oddělených znakem mezery. Pokud je v souboru nalezena proměnná, která v této sekci není uvedena, program skončí s chybovou hláškou <code>"Unknown variable '&lt;j&gt;'!\n"</code> , kde <code>&lt;j&gt;</code> je neznámá proměnná, a návratovou hodnotou 10. Pokud sekce obsahuje nepoužitou rozhodovací proměnnou <code>&lt;n&gt;</code> , program vypíše pouze varování <code>"Warning: unused variable '&lt;n&gt;'!\n"</code> .
<b>End</b>	Uvozuje konec souboru, tzn. že se vyskytuje vždy jako poslední a sekce uvedené za ním jsou syntaktickou chybou.

Až na návěští **End** není pořadí jednotlivých sekcí fixní. Na výskyt neplatných operátorů, neznámých sekcí a jiných problémů program reaguje vypsáním chybového hlášení `"Syntax error!\n"` a skončí s návratovou hodnotou 11. Komentáře v souboru jsou uvozeny znakem `"\"`. Ukázkou vstupního souboru si můžete prohlédnout v konzolovém rozhraní 1 na straně 4.

## Optimalizační algoritmus

Při analýze úlohy jistě narazíte na problémy degenerovaných úloh a jiné, které budeme pro jednoduchost ignorovat. Algoritmus hledání optimálního řešení úlohy lineárního programování může tedy teoreticky skončit následovně.

### 1. Nalezení konečného optimálního řešení

V takovém případě program vypíše optimální hodnoty rozhodovacích proměnných a skončí s návratovou hodnotou `EXIT.SUCCESS` (viz ukázka konzolového rozhraní 1). Optimálních řešení může být více, s čímž validační skript počítá.

### 2. Úloha je neomezená

Účelová funkce může nabývat libovolně velkých hodnot, aniž by porušila některou z omezujících podmínek, tj. optimum je v nekonečnu. Program na tuto skutečnost upozorní chybovým hlášením `"Objective function is unbounded.\n"` a skončí s návratovou hodnotou 20.

### 3. Neexistence přípustného řešení

Soustava omezení nemá žádnou společnou přípustnou oblast, tj. neexistuje žádný bod, který by vyhovoval všem omezením současně – úloha je nesplnitelná. V takovém případě program vypíše chybovou hláškou `"No feasible solution exists.\n"` a vrátí hodnotu 21.

## Užitečné techniky a odkazy

- [1] **Specifikace formátu LP**  
[https://www.gurobi.com/documentation/current/refman/lp\\_format.html](https://www.gurobi.com/documentation/current/refman/lp_format.html)
- [2] **Algoritmus Shunting yard**  
[https://en.wikipedia.org/wiki/Shunting-yard\\_algorithm](https://en.wikipedia.org/wiki/Shunting-yard_algorithm)
- [3] **Lineární programování** (2. kapitola, str. 18)  
<http://najada.fav.zcu.cz/~ryjacek/students/ps/TGD2.pdf>
- [4] **Simplexový algoritmus**  
[https://en.wikipedia.org/wiki/Simplex\\_algorithm](https://en.wikipedia.org/wiki/Simplex_algorithm)

**Řešení úlohy je zcela ve vaší kompetenci** – uvedené dokumenty je možné využít při řešení úlohy, ale můžete zvolit libovolné algoritmy a techniky, které podle vás nejlépe povedou k cíli.

## Přílohy

Konzolové rozhraní 1: Ukázka sestavení a činnosti programu lp.exe.

```
1 user@machine:~$ cd pc && ls
2 doc src CMakeLists.txt dokumentace.pdf Makefile Makefile.win
3 user@machine:~/pc$ make &>/dev/null && ls
4 build doc src CMakeLists.txt dokumentace.pdf Makefile Makefile.win lp.exe
5 user@machine:~/pc$ ./lp.exe
6 Input file not found!
7 user@machine:~/pc$ echo $?
8 1
9 user@machine:~/pc$ cat ../vyroba.lp
10 \ uloha z uvodu zadani
11 Subject To \ poradi neni fixni
12     vyroba:  $x_1 + 2x_2 \leq 8$ 
13     baleni:  $2 * x_1 + 1 * x_2 \leq 6$ 
14 Maximize
15      $3x_1 + 2 * x_2$  \ ucelova funkce
16 Generals
17      $x_2$   $x_1$   $x_3$ 
18 Bounds
19      $0 \leq x_1$ 
20      $0 \leq x_2$ 
21 End
22 user@machine:~/pc$ ./lp.exe ../vyroba.lp
23 Warning: unused variable 'x_3'!
24  $x_1 = 1.3333$ 
25  $x_2 = 3.3333$ 
26 user@machine:~/pc$ echo $?
27 0
28 user@machine:~/pc$ ./lp.exe --output vystup_1.txt ../vyroba.lp -o vystup_2.txt
29 Warning: unused variable 'x_3'!
30 user@machine:~/pc$ echo $?
31 0
32 user@machine:~/pc$ ls
33 build doc src CMakeLists.txt dokumentace.pdf Makefile Makefile.win lp.exe
   vystup_2.txt
34 user@machine:~/pc$ cat vystup_2.txt
35  $x_1 = 1.3333$ 
36  $x_2 = 3.3333$ 
37 user@machine:~/pc$ ./lp.exe -o ./out_dir/vystup_3.txt ../vyroba.lp
38 Invalid output destination!
39 user@machine:~/pc$ echo $?
40 2
41 user@machine:~/pc$ sed -i '1,5d' ../vyroba.lp # odstraneni prvnich 5 radek
42 user@machine:~/pc$ ./lp.exe --output vystup_1.txt ../vyroba.lp
43 Syntax error!
44 user@machine:~/pc$ echo $?
45 11
46 user@machine:~/pc$ ls # soubor "vystup_1.txt" nevznikl
47 build doc src CMakeLists.txt dokumentace.pdf Makefile Makefile.win lp.exe
   vystup_2.txt
48 user@machine:~/pc$
```