

Univerzita Tomáše Bati
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Studentská soutěžní práce

STOČ 2011

**Mobilní aplikace celočíselného programování metodou
větví a mezí**

2011

Adam Krhovják

Obsah

1. Úvod	3
3. Metoda větví a mezí	4
4. Algoritmus metody větví a mezí	7
5. Aplikace BranchAndBound.apk	8
6. Závěr	10
7. Literatura.....	10

1. Úvod

Zvažme například výrobu televizních přijímačů. Řekněme, že model lineárního programování by nám mohl poskytnout výrobní plán 205,7 sad za týden. Za této situace by většina lidí neměla problém uvést, že produkce by měla být 205 sad za týden nebo dokonce říct: „Zhruba 200 sad za týden.“ Na druhou stranu předpokládejme, že jsme si koupili skladiště, abychom takto dokončené zboží mohli uskladnit a dále, že velikost skladiště přímo odpovídá velikosti vyrobené sady. Pak na základě modelu, který jsme navrhli, bychom koupili 0,7 skladiště v jedné lokaci, 0,6 v další a tak dále. Pochopitelně, že množství zakoupených skladišť koresponduje s celočíselnými hodnotami a rádi bychom tedy, aby náš model odrážel tuto skutečnost.

Takovéto omezení úplnosti se může zdát poměrně neškodné, ale ve skutečnosti má dalekosáhlé důsledky. Význam modelování s celočíselnými proměnnými se ukázal být přínosný nejen na oblast celočíselné produkce, ale spolu s nimi můžeme správně modelovat fixní náklady, nejrůznější logické požadavky a mnohé další aspekty. Účelem těchto statí je popsat techniku řešící výše zmíněný problém, stejně tak i její možná úskalí a ukázat ji na zajímavé celočíselné aplikaci. Nejdříve se však odrazme od základní terminologie.

3. Metoda větví a mezí

Mluvíme-li o metodě větví a mezí, mluvíme o kombinatorické metodě, která hledá celočíselné řešení pomocí postupné optimalizace úloh lineárního programování, do kterých se navazují celočíselné podmínky na neceločíselné proměnné. Uvedený proces se znázorňuje pomocí stromového grafu skládajícího se z uzlů a větví.

Při výkladu uvedené metody vyjdeme z úlohy

$$\max \{z = \mathbf{c}^T \mathbf{x}, \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in \mathbf{Z}\} \quad (1)$$

Mějme množinu Q , která obsahuje celočíselné řešení:

$$Q = \{\mathbf{x} | \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in \mathbf{Z}\} \quad (2)$$

Symbolem $C(Q)$ vytyčme konvexní obal množiny.

Množinu spojitým řešením označme \tilde{Q}

$$\tilde{Q} = \{\mathbf{x} | \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \quad (3)$$

Množiny $C(Q)$ a \tilde{Q} vytvářejí neohraničené konvexní polyedrické množiny

Protože metodu větví a mezí chápeme jako kombinatorickou metodu, uplatněme kritérium optimálnosti, $\mathbf{c}^T \mathbf{x}^* \geq \mathbf{c}^T \mathbf{x}$ kde \mathbf{x}^* je řešením optimálním a \mathbf{x} je libovolné přípustné řešení. Dále pak uvažujme horní, resp. dolní hranici \bar{z}_i resp. \check{z}_i , kde i je krok vzniklou uspořádáním hodnot cílové funkce $z = \mathbf{c}^T \mathbf{x}$.

Horní hranice, která je stanovena vzhledem na množinu řešení \tilde{Q} , se vytváří z hodnot cílové funkce, dané řešením neceločíselné úlohy.

$$\begin{aligned} \bar{z}_i &= z_i^* \text{ jestliže } \mathbf{x}^* \text{ je optimálním řešením úlohy} \\ \check{z}_i &= \infty \text{ jestliže } \tilde{Q} \text{ je prázdná množina } \vee z \text{ je neohraničená na } \tilde{Q} \end{aligned} \quad (4)$$

Dolní hranici pak rozumíme hodnoty funkce dané existujícím maximálním celočíselným řešením. Tedy jinými slovy se jedná o nejlepší celočíselné řešení, které jsme doposud výpočtem dosáhli. Umožňuje nám posoudit další celočíselná řešení v procesu vyhledávání optima funkce, což můžeme zapsat takto:

$$\begin{aligned} \check{z}_i &= z_i^* \text{ pro } z_i^* > z_{i-1} \\ \check{z}_i &= z_{i-1} \text{ pro } z_i^* \leq z_{i-1} \end{aligned} \tag{5}$$

Předpokládejme řešení úlohy lineárního programování

$$\mathbf{x}^T = \{x_1, x_2, \dots, x_n\}$$

Mějme množinu

$$G = \{g_1, g_2, \dots, g_n\}$$

přičemž platí

$$\forall n \in G: 0 < n < 1$$

Nechť g_k je prvek množiny G , tak, že pro všechny $g \in G$ platí $g_k \geq g$

pak k -tou složku vektoru \mathbf{x} uvažme

$$x_k = [x_k] + g_k, [x_k] \in \mathbf{Z}, g_k \in G$$

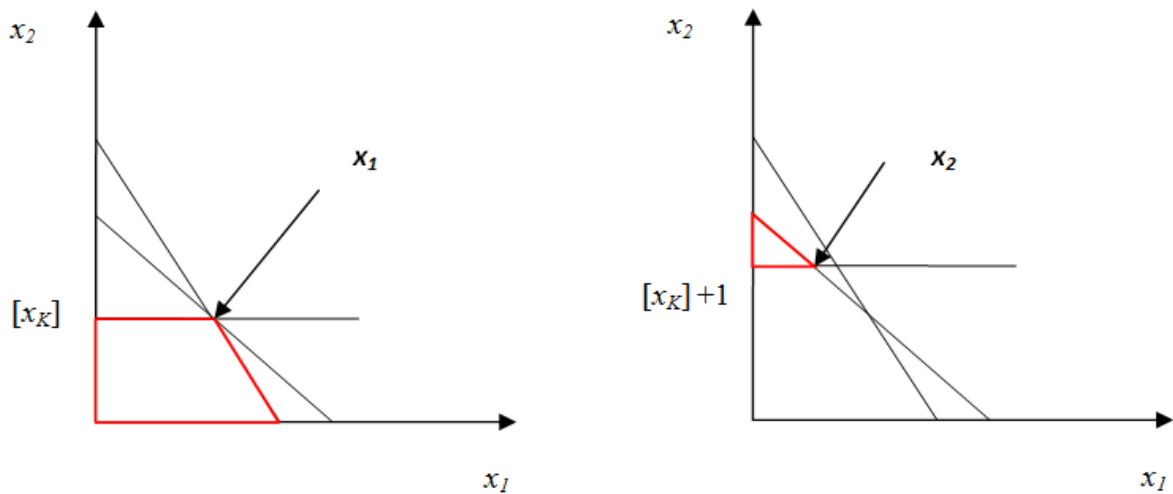
kde hranatá závorka reprezentuje celočíselnou část. Pro každé neceločíselné řešení postačuje zavést dvě omezující podmínky a to

$$\begin{aligned} x_k &\leq [x_k] \\ x_k &\geq [x_k] + 1 \end{aligned}$$

Z čehož vyplývá, že prostor

$$[x_k] < x_k < [x_k] + 1$$

můžeme zavrhnout, protože žádné z případných celočíselných řešení neleží v této oblasti, což lze vyjádřit následujícím způsobem.



Obrázek 1: Tvorba omezení v metodě větví a mezí

Grafické znázornění na obrázku ukazuje tvorbu příslušných omezení v závislosti na úloze dvou proměnných, kde červená hranice vymezuje množinu přípustných řešení.

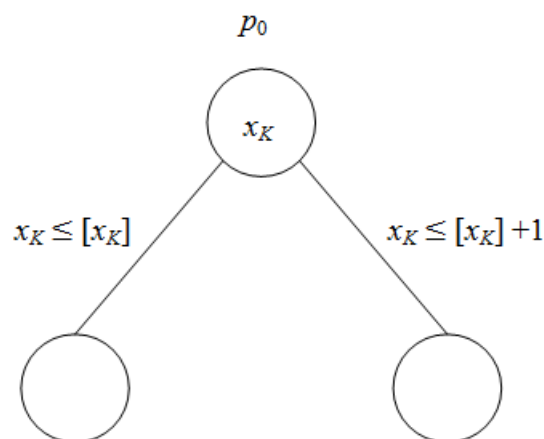
Přidáním těchto omezení k řešením na množině získáme \tilde{Q} obdržíme dvě nové úlohy, které můžeme definovat následujícím způsobem:

$$1. \max \left\{ z = \mathbf{c}^T \mathbf{x}, \tilde{Q} \cap \{x | x_K \leq [x_K]\} \right\} \quad (6)$$

$$2. \max \left\{ z = \mathbf{c}^T \mathbf{x}, \tilde{Q} \cap \{x | x_K \geq [x_K] + 1\} \right\} \quad (7)$$

Jednotlivé kroky výpočtového mechanismu pak znázorňuje stromový diagram $\Lambda = (p, b)$, kde p představuje vrcholy diagramu a b pak jednotlivé větve, spojující jednotlivé vrcholy. Strom vizualizuje dělení množin na podmnožiny. Vrchol znázorňuje přípustné řešení, ze kterého se uskutečňuje větvení, kde příslušnou větví rozumíme omezení celočíselnosti.

Uvažme následující diagram



Obrázek 2: Větvení řešení

Z obrázku je patrné, že ve vrcholu p_0 leží řešení spojité úlohy, což znamená, že minimálně jedna ze složek vektoru není \mathbf{x} celočíselná. Do vrcholu obvykle zapisujeme horní hranici hodnoty cílové funkce z . A k jednotlivým větvím pak připisujeme příslušná omezení. Vrcholy, které jsou následovníky, nesou nové řešení úlohy při daném omezení.

Rozšířením o omezení na celočíselnost je tedy nutné řešit úlohu znovu. Aplikováním algoritmu simplexové metody na úlohu s nově vzniklými omezeními získáme řešení, které bude v celočíselných proměnných. Pokud nejsou všechny proměnné celočíselné, je nutné

znovu zvolit neceločíselnou proměnnou, tu omezit na celočíselnou a přistoupit k dalšímu větvení.

Na základě předchozího uveďme postup při řešení úlohy metodou větví a mezí.

4. Algoritmus metody větví a mezí

Krok 1. Nalezneme řešení spojité úlohy lineárního programování, které bude odpovídat vrcholu $i = 1$ stromového diagramu, jehož horní mezí bude řešení spojité úlohy a dolní mezí jeho zaokrouhlení na nejbližší nižší celá čísla. Jsou-li všechny proměnné celočíselné, výpočet končí a obdrželi jsme optimální řešení. V případě, že ne, vrchol 1 se stává větvícím.

Krok 2. Pro větvení vybereme proměnnou s největší desetinnou částí a podle ní vytvoříme dva nové podproblémy, které představují možné následovníky.

Krok 3. Řešíme 1. podproblém

- a) Pokud jsme získali řešení, které není přípustné, větev zavrhneme a přejdeme k 4. kroku
- b) Jestliže je řešení přípustné, můžou nastat dvě situace:
 1. Všechny proměnné jsou v celočíselných hodnotách. Pokračujeme 4. krokem.
 2. Alespoň jedna proměnná je neceločíselná.

Tento vrchol pak přiřadíme do množiny adeptů

Krok 4. Řešíme 2. Podproblém se stejnými možnými případy jako ve předcházejícím podproblému.

Krok 5.

- a) Vrchol i má dva možné potomky v množině adeptů. Jednoho z nich zvolíme jako větvící vrchol. Přejdeme k 6. kroku.
- b) Vrchol i má jediného možného potomka v množině adeptů.
- c) Vrchol i nemá žádného potomka v množině adeptů
 1. Jestliže je množina adeptů prázdná. Algoritmus končí.
 2. Pokud ne, jeden z vrcholů adeptů určíme za nejbližší větvící vrchol.

Krok 6. Nejbližší větvící vrchol, pak vyloučíme z množiny adeptů a označíme ho jako vrchol $i+1$. Pokračujeme 2. Krokem. Nejlepší celočíselné řešení je pak optimální.

5. Aplikace BranchAndBound.apk

Účelem této kapitoly není podrobný popis jednotlivých částí zdrojového kódu, nýbrž ukázat praktické využití aplikace BranchAndBound.apk, která byla speciálně navržena pro maximalizační úlohy celočíselného programování.

Mějme následující problém:

Majitel obchodu se rozhodl rozšířit svůj sortiment nákupem nových lisů a soustruhů. Odhadl, že každý nový lis zvýší jeho zisk o 100\$ denně a každý nový soustruh zvýší zisk o 100\$ denně. Přičemž cena jednoho lisu činí 8000\$ a cena jednoho soustruhu 4000\$. Každý lis zabírá v obchodě plochu 15ft² a soustruh 30ft². Vlastník má k dispozici 40000\$ a 200ft² místa v obchodě. Otázkou tedy je, kolik kusů každého stroje má vlastník koupit, aby maximalizoval svůj denní nárůst zisku.

Matematická formulace

Cílová funkce:

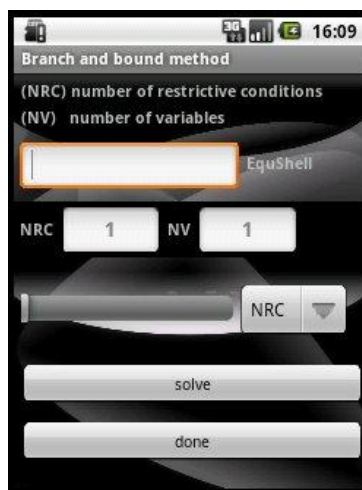
$$z = 100x_1 + 150x_2 \rightarrow \max$$

Omezující podmínky:

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

Ten nám poskytl chybějící ingredienci pro uvedení zmíněné mobilní aplikace, ve které si v několika krocích daný problém vyřešíme. Po spuštění aplikace se objeví standardní, grafické uživatelské rozhraní, jež v sobě nese základní prvky nutné pro komunikaci s problémem.



Obrázek 3: Uživatelské rozhraní aplikace

Užitím posuvníku si nejdříve nastavíme hodnotu NRC, představující počet omezení dané úlohy. Podobně budeme postupovat i při nastavování hodnoty NV, jež nese počet proměnných. Chceme-li přenastavovat hodnoty NRC a NV pomocí posuvníku, je nejprve nutné zvolit v menu napravo, kterou z nich chceme ovlivňovat, což je patrné z Obrázku 4



Obrázek 4: Vkládání informací o počtu omezujících podmínek a proměnných

V této chvíli můžeme začít zadávat koeficienty jednotlivých omezení včetně hodnot koeficientů účelové funkce. Nutno dodat, že po přidání každé podmínky je nutné koeficienty uložit prostřednictvím tlačítka done. A konečně, když jsme všechny vstupní informace o úloze předali aplikaci, stačí kliknout na tlačítko solve, čímž vyvoláme dialog, ve kterém vidíme optimální celočíselné řešení. Uvedený výklad pak znazorňuje Obrázek 5.



Obrázek 5: Vkládání koeficientů s následným řešením problému

6. Závěr

Cílem této práce je hlouběji popsat techniku, jež má rozsáhlé důsledky pro úlohy z ekonomické praxe a poukázat na problém, že řešení úlohy celočíselného programování jako úlohy neceločíselného lineárního programování obecně není možné. Dále pak, že řešení nelze vidět ani v pouhém zaokrouhlování, jelikož bychom nemuseli získat přípustné nebo optimální řešení a proto je nutné takovýto problém řešit některou z metod celočíselného programování. A právě o jedné z metod tato práce podrobně pojednává. Vědomí, že i malý problém se může užitím zmíněné metody stát velmi obtížným a těžko schůdným mě vedl k vytvoření aplikace, prostřednictvím které můžete problém vyřešit z “kapsy“. ☺

7. Literatura

- [1] LAŠCIAK, Adam, et al. Optimálne Programovanie. Bratislava : SNTL, 1983. 565 s.
- [2] ŠŮCHA, Přemysl. Celočíselné lineární programování. 11.3.2004 [cit. 2011-01-24]. Dostupné z WWW:[<http://dce.felk.cvut.cz/sucha/articles/ilp.pdf>].