ÚVAHY O MOŽNOSTIACH POUŽITIA MODULU OPTIMIZER NA RIEŠENIE ÚLOH LINEÁRNEHO PROGRAMOVANIA

Ing. Vladimír Jerz, PhD.
Katedra výrobných systémov
Strojnícka fakulta STU
Bratislava

1. *Východiská*

Optimalizačný modul Optimizer je nástroj, ktorý uľahčuje hľadanie optimálnych, resp. suboptimálnych riešení v prostredí simulačného systému Witness. Jeho použiteľnosť je podmienená v prvom rade možnosťou formulovať účelovú funkciu, ktorej maximalizácia alebo minimalizácia je cieľom riešenia. Nespornou výhodou tohto nástroja je to, že parametrami účelovej funkcie (v terminológii lineárneho programovania „cenami procesov" [2]) môžu byť okrem konštánt aj veličiny, ktoré sa v priebehu simulácie získavajú ako výstupné veličiny simulačného experimentu so zložitými modelmi, napríklad aj stochastickými. Premennými sú konštanty – vstupné veličiny modelu. Modul umožňuje doplniť model aj o vlastné obmedzenia, formulované podobne ako vlastné obmedzenia lineárneho optimalizačného modelu.

Podstatou postupu hľadania čo najlepšieho riešenia je vykonávanie postupnosti simulačných behov s rôznymi kombináciami vstupných premenných veličín modelu. Modul dáva k dispozícii niekoľko voliteľných metód na hľadanie takej postupnosti týchto kombinácií, ktorá by za daných podmienok čo možno najrýchlejšie a najspoľahlivejšie viedla k optimu.

Ponúka sa otázka, či možno využiť modul Optimizer na riešenie klasických úloh lineárneho programovania. Je zrejmé, že teoreticky áno, nájdené riešenie však nemusí byť optimálne, ale len blízke optimálnemu a zväčša nemožno ani dokázať, či je optimálne, alebo ako „ďaleko" je od optimálneho riešenia. Premenné modelu možno modelovať ako prvky *Variables*. Treba zabezpečiť aj minimálne predpoklady na to, aby sa mohol rozbehnúť simulačný beh, t. j. vytvoriť aspoň jednu súčiastku (prvok *Part*), ktorá prinajmenšom vstúpi do modelu a potom z neho odíde.

Predmetom nasledujúcich úvah je hľadanie odpovede na otázku, či možno s určitosťou dosiahnuť, aby postupnosť riešení konvergovala k optimálnemu riešeniu, ktoré poznáme, pretože ho súbežne vypočítame aj postupmi operačnej analýzy. Zároveň to umožní formulovať závery o konvergencii k optimálnemu riešeniu v prípadoch, keď z dôvodu zložitosti modelu nie je možné jednoznačné optimum vypočítať a riešenie získané pomocou modulu Optimizer s ním porovnať.

2. *Experimentovanie s modelom*

Na experimentovanie bol použitý model úlohy lineárneho programovania so štyrmi premennými:

Účelová funkcia:

$$Z = 800x_1 + 1500x_2 + 500x_3 + 1000x_4 \overset{!}{=} \max$$

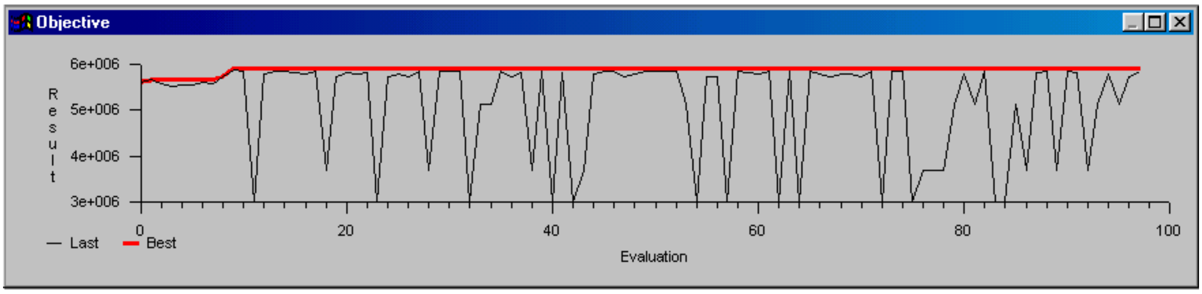
Vlastné obmedzenia:

$$\begin{aligned} 10x_1 + 15x_2 + 8x_3 + 5x_4 &\leq 60000 \\ 4x_1 + 2x_2 + 5x_3 + 10x_4 &\leq 40000 \\ 20x_1 + 25x_2 + 18x_3 + 15x_4 &\leq 120000 \end{aligned}$$

Podmienky nezápornosti:

$$x_{1,\,2,\,3,\,4} \geq 0$$

Najprv bola zvolená metóda Hill Climb. Experimentovanie bolo nastavené tak, aby bolo ukončené po vykonaní 1000 simulačných behov, alebo 200 behov bez zvýšenia hodnoty účelovej funkcie. V rámci experimentu bolo vykonaných 210 simulačných behov, pričom už pri deviatom behu (obr. 1) bola dosiahnutá maximálna hodnota účelovej funkcie 5880000 (maximálna z uvedených 210). S ohľadom na nastavené parametre bol experiment ukončený po vykonaní ďalších 200 behov, pri ktorých nebolo zaznamenané zvýšenie hodnoty účelovej funkcie. Vektor riešenia mal hodnoty (3600, 500, 100, 2200).

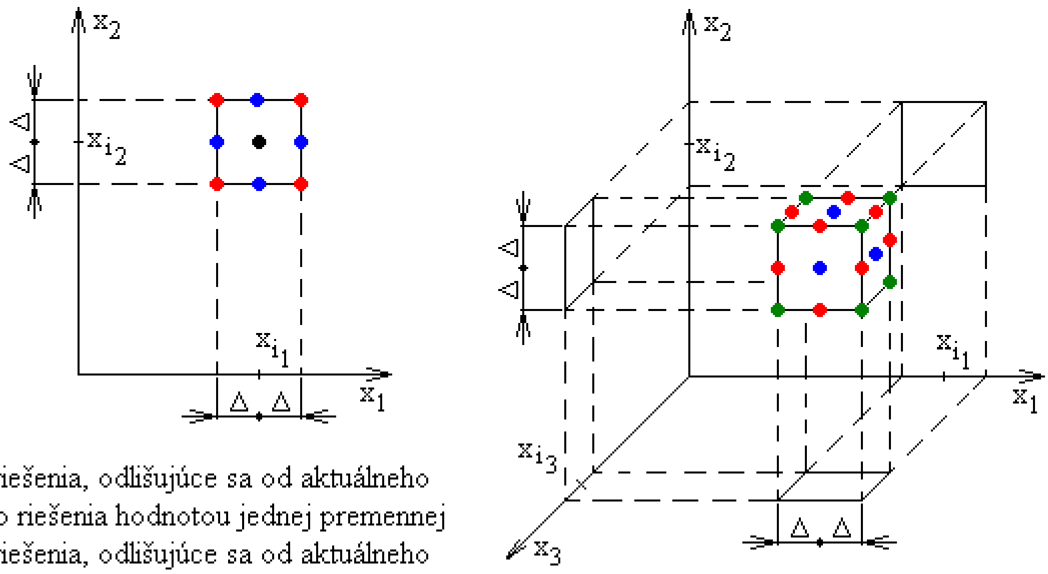


Obr. 1

Experiment bol zopakovaný s nastaveniami: ukončenie po 10000 behoch alebo 5000 behoch bez zvýšenia hodnoty účelovej funkcie. Ani v tomto prípade nebolo nájdené riešenie s vyššou hodnotou účelovej funkcie (po vykonaní 5010 behov, t. j. 5000 behov bez zvýšenia hodnoty účelovej funkcie), čo by mohlo viesť k domnienke, že nájdené riešenie je optimálne.

S ohľadom na charakter úlohy a voľbu metódy by sa dalo predpokladať, že v blízkom okolí dosiahnutého riešenia neexistuje lepšie riešenie.

Za riešenie z blízkeho okolie nájdeného riešenia budeme považovať všetky riešenia, ktoré sa od nájdeného odlišujú aspoň v jednej hodnote vektora riešení o najmenší krok (ktorý bol nastavený pri formulácii podmienok experimentovania na hodnotu 100). Takých riešení z blízkeho okolia je v našom prípade 80 (vo všeobecnosti je počet riešení z blízkeho okolia 3^n-1, kde n je počet premenných modelu; pre $n = 2$ a $n = 3$ možno tieto riešenia graficky zobraziť – *obr.*1). Výpočtom pomocou tabuľkového editora bolo zistené, že z uvedených 80 riešení z blízkeho okolia 46 riešení vyhovuje vlastným obmedzeniam aj podmienkam nezápornosti a v 7 z nich aj účelová funkcia dosahuje vyššiu hodnotu (*tab*. 1). Tieto riešenia a ani lepšie riešenia zo širšieho okolia však algoritmom Hill Climb neboli nájdené.



● - riešenia, odlišujúce sa od aktuálneho i-teho riešenia hodnotou jednej premennej
● - riešenia, odlišujúce sa od aktuálneho riešenia hodnotou dvoch premenných
● - riešenia, odlišujúce sa od aktuálneho riešenia hodnotou troch premenných

Obr. 2

Tab. 1

x1	x2	x3	x4	Z
3600	**500**	**100**	**2200**	**5880000**
3500	500	100	2300	5900000
3500	600	0	2200	5900000
3500	600	0	2300	6000000
3500	600	100	2200	5950000
3600	500	0	2300	5930000
3600	600	0	2200	5980000
3700	500	0	2200	5910000

Najlepšie z riešení z blízkeho okolia aktuálneho riešenia uvedených v tabuľke možno nájsť aj pomocou modulu Optimizer tak, že sa zadá oblasť prípustných hodnôt premenných len z tohto blízkeho okolia, krok 100 a použije sa metóda All Combinations. Rovnaký postup, ale s použitím metódy Hill Climb neviedol k nájdeniu lepšieho riešenia (ani vykonaním 1000 behov, pri ktorých sa prehľadávalo len 80 susediacich riešení, sa nenašlo ani jedno lepšie riešenie). Metódou SA sa našlo najlepšie riešenie z blízkeho okolia v 23. behu.

Ďalší experiment bol vykonaný s použitím metódy SA. Boli nastavené podobné parametre ako v predchádzajúcom prípade: ukončenie po 10000 behoch alebo 5000 behoch bez zvýšenia hodnoty účelovej funkcie a ostatné parametre boli nastavené na odporúčané (default) hodnoty. Trvanie experimentovania sa podstatne predĺžilo, bolo však nájdené riešenie (3100, 900, 100, 2200) s lepšou hodnotou účelovej funkcie (6080000). Toto riešenie bolo nájdené ako riešenie 290. behu z celkového počtu 5291 behov. Preskúmaním blízkeho okolia pomocou tabuľkového editora bolo zistené, že z uvedených 80 riešení z blízkeho okolia opäť 46 riešení vyhovuje vlastným obmedzeniam aj podmienkam nezápornosti a opäť v 7 z nich aj účelová funkcia dosahuje vyššiu hodnotu. Tieto riešenia a ani lepšie riešenia zo širšieho okolia však ani algoritmom SA neboli nájdené.

(Z podstaty modelu vyplýva, že ak neuvažujeme vlastné obmedzenia a podmienky nezápornosti, vždy bude hodnota účelovej funkcie, zodpovedajúca bodom v blízkom okolí vyššia alebo rovnaká v polovici z uvedených bodov a nižšia alebo rovnaká v druhej polovici týchto bodov.)

Na nájdenie skutočného optimálneho riešenia by bolo vhodné použiť metódu, ktorá overí všetky riešenia z množiny prípustných riešení (All Combinations). Všetkých riešení je však v tomto prípade 2825761 a prípustných riešení 542196 (a aj to za predpokladu, že hodnoty jednotlivých premenných sú z intervalu <0; 4000> a menia sa s krokom 100). Čas riešenia by sa v prípade overovania všetkých riešení neúmerne predĺžil.

Ponúka sa však možnosť urobiť v prvej etape experiment s veľkým krokom (napr. 1000) a nájdené riešenie považovať za východiskové pre ďalšiu etapu, v ktorej by sa krok zmenšil (napr. na 400) atď. Rozpätie hodnôt jednotlivých premenných v rámci jednotlivých etáp možno upravovať s ohľadom na podmienky nezápornosti. Ak v optimálnom riešení niektorého kroku dosiahne niektorá premenná hornú odhadnutú hranicu, treba túto hranicu posunúť. V tabuľke 2 sú znázornené jednotlivé etapy:

Tab. 2

	počet komb.		rozpätie x1		opt. x1	rozpätie x2		opt. x2	rozpätie x3		opt. x3	rozpätie x4		opt. x4	Z
krok	spolu	príp.	od	do		od	do		od	do		od	do		
1000	652	144	0	4000	0	0	4000	3000	0	4000	0	0	4000	3000	7500000
400	576	98	0	1000	0	2000	4000	2800	0	1000	0	2000	4000	3200	7400000
200	225	59	0	400	0	2400	3200	3000	0	400	0	2800	3600	3000	7500000
100	225	50	0	200	0	2800	3200	3000	0	200	0	2800	3200	3000	7500000
50	225	50	0	100	0	2900	3100	3000	0	100	0	2900	3100	3000	7500000

Týmto postupom bolo nájdené riešenie (0, 3000, 0, 3000) s hodnotou účelovej funkcie 7500000 už v prvom kroku a v nasledujúcich krokoch nebolo nájdené lepšie riešenie.

Na overenie „kvality" riešení, dosiahnutých pomocou modulu Optimizer bola uvedená úloha lineárneho programovania vyriešená aj simplexovou metódou – výpočtom bolo nájdené jediné optimálne riešenie (0; 2727; 0; 3455) s hodnotou účelovej funkcie 7545455.

Podobnému experimentovaniu bolo podrobených niekoľko rôznych modelov s troma aj dvoma premennými. Priebeh približovania sa k optimálnemu riešeniu bol podobný, i keď s ohľadom na podstatne menší priestor prípustných riešení bolo priblíženie sa k optimu oveľa výraznejšie. Problémy s hľadaním „lepších" riešení sa objavovali predovšetkým v tesnej blízkosti hraníc oblastí prípustných riešení napriek tomu, že v prípade lineárnych optimalizačných úloh ide o konvexné oblasti.

3. Záver

Vykonané experimenty ukázali, že použitie metód Hill Climb ani SA neumožnilo priblížiť sa spoľahlivo a dostatočne k optimálnemu riešeniu. Hlavnou príčinou bolo, že algoritmus hľadania lepšieho riešenia v okolí aktuálneho riešenia neumožnil nájsť žiadne takéto riešenie, napriek tomu, že ich existovalo viac v najbližšom aj vzdialenejšom okolí aktuálneho riešenia.

Použitie modulu Optimizer na riešenie lineárnych optimalizačných úloh prináša uspokojivé výsledky len v prípade menšieho počtu premenných, malého počtu prípustných hodnôt jednotlivých (diskrétnych) premenných, najmä ak možno naplánovať overenie všetkých prípustných kombinácií týchto hodnôt.

Dá sa predpokladať, že tieto závery budú platiť aj v prípade optimalizácie bežných zložitých simulačných modelov, pre ktoré je typická veľká miera stochastickosti a pri ktorých optimálnosť riešenia nemožno overiť analytickými metódami. Napriek tomu použitie algoritmov Hill Climb a SA môže umožniť rýchle nájdenie výrazne lepších riešení, ako by boli nájdené odhadmi, nemožno sa však spoliehať na to, že sú optimálne.

Literatúra:

[1] Jerz, V., Tolnay, M.: *Simulácia diskrétnych systémov*. Bratislava: Vydavateľstvo STU, 2006. 162 s., ISBN 80-227-2384-3

[2] Sakál, P., Jerz, V.: *Operačná analýza v praxi manažéra*. Trnava: SP SYNERGIA – TRIPSOFT, 2003, 335 s. ISBN 80-968734-3-1

[3] Sakál, P., Jerz, V.: *Operačná analýza v praxi manažéra II. (Systémová a operačná analýza)*. Trnava: SP SYNERGIA – TRIPSOFT, 2006, 343 s. ISBN 80-969390-5-X

[4] Sakál, P., Štrpka, A.: *Operačná a systémová analýza. Zbierka príkladov I.* Bratislava: ES SVŠT, 1985, 253 s.

[5] Witopt.hlp. Witness 2006. Lanner Group 2006

Ing. Vladimír Jerz, PhD.
Katedra výrobných systémov
Strojnícka fakulta STU Bratislava
Námestie slobody 17
812 31 Bratislava
tel.: 57296554
e-mail: vladimir.jerz@stuba.sk