Метода гранања и ограничавања

(Branch & Bound)

Милан Станојевић

Садржај

1 Основни појмови

Метода гранања и ограничавања (МГО) је метода која се користи за **егзактно** решавање оптимизационих проблема за које нису познати полиномијални алгоритми који у општем случају у „разумном времену” могу решити инстанце проблема „већих” димензија. Појмови под наводницима су непрецизни и разликују се од типа проблема и околности под којим се решавају.[1] Проблеми из ове класе се зову НП-тешки (НПТ). За разлику од њих, оптимизационе проблеме за које су познати полиномијални алгоритми зовемо полиномијално решиви (ПР). МГО гарантује да је добијено решење заиста оптимално (за разлику од хеуристичких метода за које таква гаранција не постоји).

МГО спада у методе **посредног претраживања** које претражују само део допустивог скупа[2] и посредно доказују да се оптимално решење не налази у делу који није претражен.

Да би могао да се идентификује део допустивог скупа који сигурно не садржи оптимално решење, МГО дели допустиви скуп на мање делове и тако га уситњава све док се не испуни услов да неки од тих делова буде искључен из даљег претраживања. Због оваквог поступка се за методу гранања и ограничавања каже да користи стратегију „подели па владај” (лат. divide et impera).

У методи МГО се користе следећи појмови:

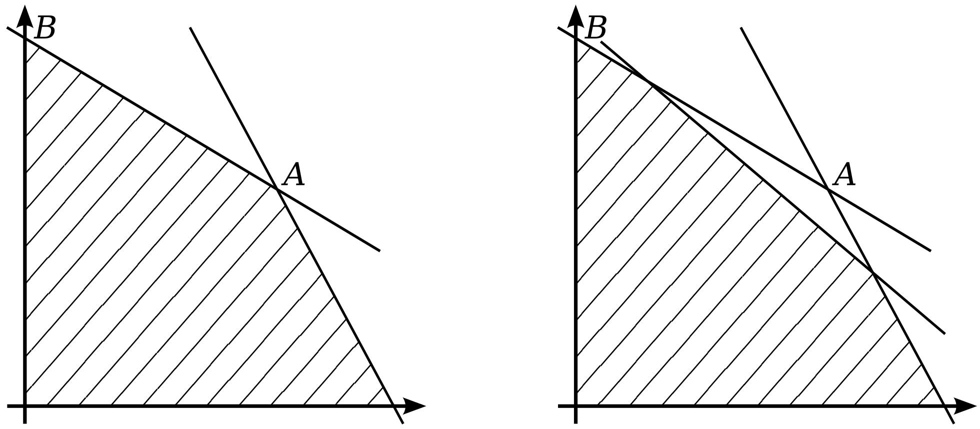
- релаксација и рестрикција проблема;
- горња и доња граница проблема;
- гранање и одсецање стабла претраживања.

Релаксација и рестрикција

Посматрајмо два скупа X_1 и X_2 из $\mathbb{R}^2$ приказана на слици 1.1, дефинисана линеарним и природним ограничењима и одговарајуће проблеме оптимизације реалне функције циља $f(x)$ над тим скуповима:

[1] Грубо речено, под проблемима „већих” димензијама се обично подразумевају реални проблеми, а под „разумним” време од неколико секунди до неколико сати.

[2] За разлику од метода непосредног претраживања, у којима се претражује целокупан скуп допустивих решења.



а) Допустиви скуп X_1 б) Допустиви скуп X_2

Слика 1.1: Рестрикција и релаксација линеарним ограничењем

$$\begin{array}{ll} (\max) & f(x) \\ \text{п.о.} & \\ & x \in X_1 \end{array} \tag{1.1}$$

и

$$\begin{array}{ll} (\max) & f(x) \\ \text{п.о.} & \\ & x \in X_2 \end{array} \tag{1.2}$$

На слици 1.1 се види да је скуп X_2 добијен тако што је скупу X_1 додато једно линеарно ограничење.

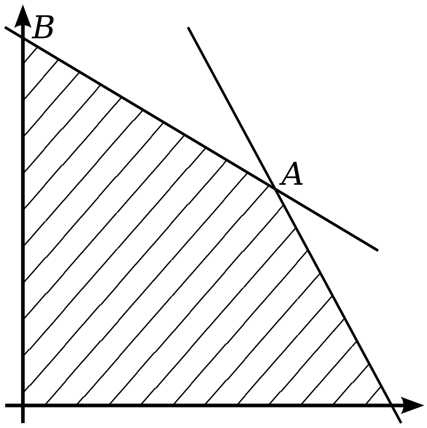
Означимо са x_1^* и x_2^* оптимална решења ова два проблема, респективно. За проблеме (1.1) и (1.2) важи $f(x_1^*) \geq f(x_2^*)$. Наиме, додавање ограничења може да доведе само до тога да оптимална вредност функције циља буде лошија или једнака оптималној вредности функције циља полазног проблема. Важи и обрнуто, ако се неко ограничење проблема занемари, то може да доведе само до тога да оптимална вредност функције циља буде боља или једнака оптималној вредности функције циља полазног проблема.

Занемаривање (уклањање) ограничења назива се **релаксација**, а додавање ограничења **рестрикција**.

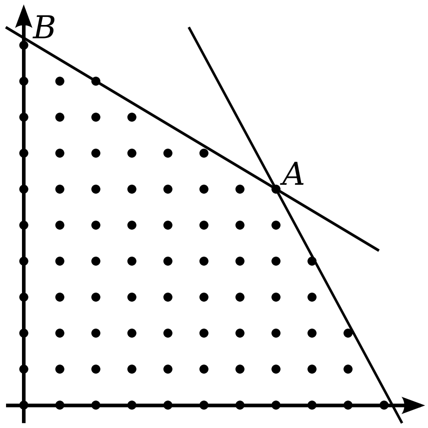
Пример 1.1 *У примеру приказаном сликом 1.1, у случају да је оптимално решење проблема (1.1) у тачки B, важиће $f(x_1^*) = f(x_2^*)$. Ако би јединствено оптимално решење проблема (1.1) било у тачки A, то решење је недопустиво за проблем (1.2), тако да би оптимална вредност функције циља тог проблема била лошија, односно, важило би $f(x_1^*) > f(x_2^*)$.* □

На примеру 1.1 може се видети да важи и следећа тврдња. Претпоставимо да је проблем (1.2) полазни, а да је проблем (1.1) добијен његовом релаксацијом. **Ако је оптимално решење релаксираног проблема допустиво и за полазни проблем, онда је оно и оптимално за полазни проблем.**

Рестрикција и релаксација се могу извести и додавањем, односно занемаривањем услова целобројности. На слици 1.2 је приказан полазни скуп X_1 и нови скуп X_3 који је добијен тако што је скупу ограничења проблема (1.1) додат услов $x \in \mathbb{Z}^2$.



а) Допустиви скуп X_1



б) Допустиви скуп X_3

Слика 1.2: Рестрикција и релаксација условом целобројности

Посматрајмо проблем:

$$\begin{array}{ll} (\max) & f(x) \\ \text{п.о.} & \\ & x \in X_3 \end{array} \tag{1.3}$$

Означимо његово оптимално решење са x_3^*.

Пример 1.2 *У примеру приказаном сликом 1.2 важи општи услов*

$$f(x_1^*) \geq f(x_3^*), \tag{1.4}$$

само што ће у овом случају, ако је оптимално решење проблема (1.1) у тачки B, важити $f(x_1^*) > f(x_3^*)$, пошто према слици та тачка нема све координате целобројне. Ако би оптимално решење проблема (1.1) било у тачки A која припада и скупу X_3, тада би то решење било оптимално и за проблем (1.3) и важило би $f(x_1^*) = f(x_3^*)$. □

Иако се занемаривањем ограничења (релаксацијом) допустиви скуп „повећава", решавање реласираних проблема може да има знатно мању рачунску сложеност. За многе познате проблеме из класе НПТ постоје одговарајући релаксирани проблеми који су полиномијално решиви. Погледајмо пример проблема трговачког путника.

Пример 1.3 *Математички модел проблема трговачког путника на задатом графу $G = (V, E)$ је:*

$$(\min) f(x) = \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{1.5}$$

п.о.

$$\sum_{i\in V} x_{ij} = 1, \quad \forall j \in V \tag{1.6}$$

$$\sum_{j\in V} x_{ij} = 1, \quad \forall i \in V \tag{1.7}$$

$$u_i - u_j + n x_{ij} \leq n - 1, \quad \forall (i,j) \in E : i,j \neq 1 \tag{1.8}$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in E \tag{1.9}$$

где су: c_{ij} дужина гране $(i,j) \in E$, x_{ij} управљачка променљива која има вредност 1 када грана (i,j) припада Хамилтоновој контури и 0 у супротном, u_i континуалне ненегативне помоћне променљиве које означавају редослед чворова у контури и n број чворова графа G. Функција циља (1.5) представља дужину Хамилтонове контуре, ограничења (1.6) и (1.7) обезбеђују да Хамилтонова контура прође кроз сваки чвор тачно једном, ограничење (1.8) спречава формирање тзв. подконтура и (1.9) представља услов целобројности.

Овај модел се може релаксирати на следећа два начина, тако да се добије полиномијална релаксација[3]:

1. *Ако се укину ограничења целобројности (1.9) (и уместо њих уведу линеарна ограничења $0 \leq x_{ij} \leq 1$), модел се своди на проблем линеарног програмирања, који спада у ПР класу проблема, и*

2. *Елиминисањем ограничења (1.8) модел се своди на проблем асигнације (распоређивања), који, иако је комбинаторан, такође спада у класу ПР.*

Ако би оптимално решење било које од ове две релаксације полазног проблема (1.5)-(1.9) било допустиво и за полазни проблем, добило би се оптимално решење полазног НПТ проблема у полиномијалном времену. □

На жалост, решавањем релаксације у највећем броју случајева се не добија допустиво решење полазног проблема.

Горња и доња граница

Претпоставимо да треба решити проблем целобројног програмирања (1.3) и да смо га занемаривањем услова целобројности релаксирали и свели на проблем (1.1). За свако допустиво решење проблема (1.3) $x_3 \in X_3$ важе неједначине:

$$f(x_1^*) \geq f(x_3^*) \geq f(x_3). \tag{1.10}$$

[3] *Поред ових, постоје још неке полиномијалне релаксације наведеног проблема.*

Другим речима, оптимално решење проблема (1.3) не може бити боље од оптималног решења његове релаксације и не може бити горе од било ког допустивог решења тог проблема.

Прва неједначина из (1.10) произилази из закључка да се занимаривањем ограничења, оптимална вредност функције циља не може погоршати, тј. произилази из неједначине (1.4). Друга неједначина из (1.10) произилази из дефиниције оптималности решења x_3^*.

За задати проблем оптимизације (1.3) вредност $\overline{f} = f\left(x_1^*\right)$ представља **горњу границу**, а вредност $\underline{f} = f\left(x_3\right)$ **доњу границу** оптималне вредности функције циља. У случају да се решава проблем минимизације, ове две границе мењају места, све приказане неједначине мењају смер, а све остале тврдње остају исте.

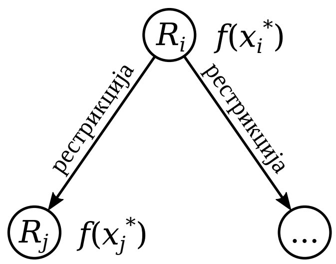
Гранање и одсецање

Основна идеја за коришћење појмова релаксације, рестрикције и горње и доње границе у процесу решавања тешких проблема из класе НПТ је следећа. Уместо да претражујемо експоненцијално велики допустиви скуп полазног проблема P, налазимо оптимално решење његове полиномијалне релаксације R. Ако је то решење допустиво за P, полазни проблем је решен. Ако није, онда је могуће проблему R додати ограничења (дакле рестрикција) која ће добијено решење учинити недопустивим за новодобијени релаксирани проблем, али тако да тај нови проблем остане полиномијалан. Додавањем ограничења, део допустивог скупа проблема R који није допустив за P, а у коме се налази претходно добијено решење, се искључује из даљег претраживања, а остатак дели на више делова чија унија садржи сва допустива решења P.[4] Због тога се оптимално решење проблема P налази бар у једном од тих делова. Оптимизацијом функције циља над различитим деловима допустивог скупа креира се више нових (пот)проблема, који су рестрикције релаксације R.

У методици операционих истраживања, да би се објаснио процес дељења допустивог скупа и генерисања нових потпроблема над тим скуповима, најчешће се користи графички приказ који процес претраживања приказује као коренско стабло. У сваком чвору стабла решава се по један потпроблем. У таквом начину приказивања, дељење допустивог скупа увођењем ограничења и креирање нових потпроблема назива се **гранање**.

За сваки потпроблем R_j, $j > i$, који је настао тако што је проблему R_i додато ограничење (пошто представља рестрикцију проблема R_i), важи да не може имати решење боље од оптималног решења проблема R_i (слика 1.3). Другим речима, вредност решења проблема R_i представља горњу границу за све његове потпроблеме R_j, тј. $f\left(x_i^*\right) \geq f\left(x_j^*\right)$. Ако је у процесу оптимизације, у неком ранијем кораку (или хеуристичким решавањем полазног проблема) добијено неко допустиво решење x_d полазног проблема P за које важи $f\left(x_d\right) \geq f\left(x_i^*\right)$, тада проблеме R_j нема смисла ни решавати, зато што њихови допустиви скупови не садрже оптимално решење про-

[4] Ово се јасно види у примеру 2.2, стр. 9.



Слика 1.3: Рестрикција

блема P. У овом случају се може рећи да **доња граница** $f(x_d)$ полазног проблема није већа или једнака од **горње границе** потпроблема R_j.

На описани начин се део допустивог скупа полазног проблема који је садржан у допустивом скупу проблема R_i може искључити из даљег претраживања, а да се при томе може гарантовати да се оптимално решење не налази у њему. Одређивање горњих и доњих граница и њихово међусобно приближавање током извршавања алгоритма МГО зове се **ограничавање**, а елиминација делова допустивог скупа се назива **одсецање**.

У реалним применама МГО, одсецањем се могу елиминисати значајне области допустивог скупа. На тај начин и инстанце релативно великих димензија могу бити (понекад) решене у разумном времену.

2 Примери

Пример 2.1 *Решити проблем ранца разматрајући 4 предмета чије су вредности 15, 12, 4 и 2 новчане јединице, респективно, а тежине 8, 5, 3 и 2 kg, респективно. Изабрати које од наведена 4 предмета понети у ранцу носивости 10 kg, тако да њихова укупна вредност буде максимална, а да при томе њихова укупна тежина не превазиђе носивост ранца.*

Математички модел проблема се може записати на следећи начин:

$$
\begin{array}{lrl}
P: & (\max) & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\
 & \text{п.о.} & \\
 & & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\
 & & x_k \in \{0,1\} \quad \forall k = 1,2,3,4
\end{array}
$$

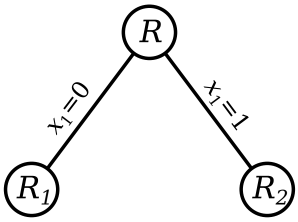
Овај проблем спада у класу НПТ и биће решен методом гранања и ограничавања.

Први корак је формирање полиномијалне релаксације. У овом случају је природно занемарити услов целобројности и тако добити проблем линеарног програмирања који спада у класу ПР.

$$
\begin{array}{lrl}
R: & (\max) & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\
 & \text{п.о.} & \\
 & & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\
 & & 0 \leq x_k \leq 1 \quad \forall k = 1,2,3,4
\end{array}
$$

Слика 2.1: Почетна релаксација

Решавањем проблема R *добија се решење* $x^R = (0,625; 1; 0; 0)$, $f\left(x^R\right) = 21,375$, *које није целобројно. Променљива* x_1 *има вредност која није допустива за полазни проблем. Она може бити или 0 или 1. Стога вршимо гранање, тј. генеришемо два проблема* R_1 *и* R_2 *у којима је променљива* x_1 *фиксирана (слика 2.2). На овај начин смо из даљег разматрања избацили део допустивог скупа проблема* R *у коме је* $0 < x_1 < 1$, *(што је недопустиво за полазни проблем), а оптимално решење проблема* P *се и даље налази у бар једном од допустивих скупова проблема* R_1 *и* R_2.



$$
\begin{array}{lll}
R_1: & (\max) & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\
 & \text{п.о.} & \\
 & & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\
 & & x_1 = 0 \\
 & & 0 \leq x_k \leq 1 \quad \forall k = 2,3,4 \\
R_2: & (\max) & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\
 & \text{п.о.} & \\
 & & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\
 & & x_1 = 1 \\
 & & 0 \leq x_k \leq 1 \quad \forall k = 2,3,4
\end{array}
$$

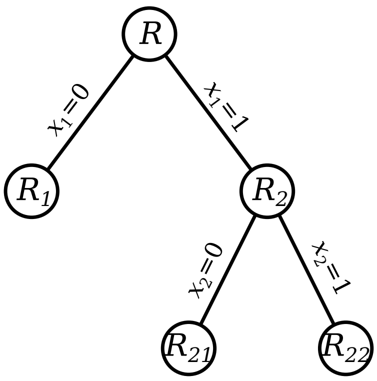
Слика 2.2: Гранање проблема R

Оптимално решење проблема R_1 је $x^{R_1} = (0;1;1;1)$, $f(x^{R_1}) = 18$. Ово решење је целобројно и допустиво за полазни проблем P. Могу се донети два закључка:

1. *Не постоји ни једно решење проблема P са вредношћу $x_1 = 0$ које је боље од x^{R_1}. То значи да нема потребе даље претраживати тај део допустиве области.*

2. *Пошто је x^{R_1} допустиво решење проблема P, његова вредност представља доњу границу оптималног решења овог проблема, тј. ако оптимално решење постоји, његова вредност не може бити мања од 18.*

Оптимизацијом проблема R_2 добијамо решење $x^{R_2} = (1;0,9;0;0)$, $f(x^{R_2}) = 19,8$, које је опет разломљено. Вредност 19,8 представља горњу границу вредности оптималног решења проблема P у делу допустивог скупа у коме је $x_1 = 1$, а пошто је она већа од до сада познате доње границе 18, још увек постоји шанса да се пронађе решење боље од x^{R_1}. Зато вршимо гранање проблема R_2 и формирамо потпроблеме R_{21} и R_{22}.



$$
\begin{array}{lll}
R_{21}: & (\max) & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\
 & \text{п.о.} & \\
 & & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\
 & & x_1 = 1 \\
 & & x_2 = 0 \\
 & & 0 \leq x_k \leq 1 \quad \forall k = 3,4 \\
R_{22}: & (\max) & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\
 & \text{п.о.} & \\
 & & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\
 & & x_1 = 1 \\
 & & x_2 = 1 \\
 & & 0 \leq x_k \leq 1 \quad \forall k = 3,4
\end{array}
$$

Слика 2.3: Гранање проблема R_2

Решавањем проблема R_{21} добија се решење $x^{R_{21}} = (1;0;0,667;0)$, $f(x^{R_{21}}) = 17,667$. Оно је и разломљено (недопустиво за P), а његова вредност је мања од доње границе

18. Даљом рестрикцијом овог проблема сигурно не можемо добити оптимално решење проблема P, тако да овај део области можемо избацити из даље анализе.

Решавањем проблема R_{22} утврђује се да је допустива област овог проблема празна.

Овим смо разрешили све делове допустиве области проблема P и можемо гарантовати да је до сада најбоље нађено решење x^{R_1} и оптимално за P.

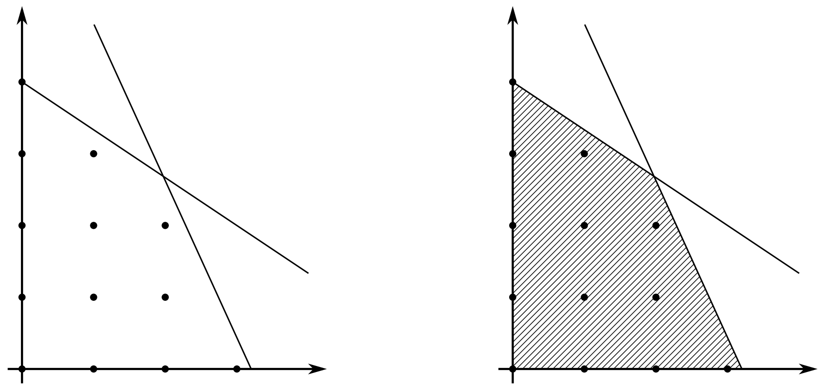
***Закључак:** Да би укупна вредност понетих предмета била максимална, потребно је спаковати предмете 2, 3, и 4. Њихова вредност износи 18 новчаних јединица, а укупна тежина 10 kg.* □

***Пример 2.2** Решити проблем целобројног линеарног програмирања:*

$$
\begin{array}{lrl}
P: & (\max) & f(x) = 4x_1 + 5x_2 \\
 & \textit{п.о.} & \\
 & & 2x_1 + 3x_2 \le 12 \\
 & & 11x_1 + 5x_2 \le 35 \\
 & & x_1, x_2 \in \mathbb{N}_0
\end{array}
$$

Задати проблем спада у класу НПТ. Први корак је формирање полиномијалне релаксације. Занемаривањем услова целобројности добија се проблем линеарног програмирања:

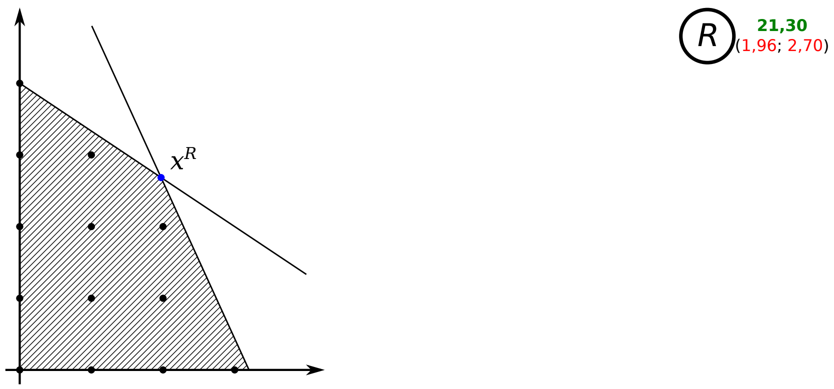
$$
\begin{array}{lrl}
R: & (\max) & f(x) = 4x_1 + 5x_2 \\
 & \textit{п.о.} & \\
 & & 2x_1 + 3x_2 \le 12 \\
 & & 11x_1 + 5x_2 \le 35 \\
 & & x_1, x_2 \ge 0
\end{array}
$$



а) Целобројни допустиви скуп   б) Допустиви скуп X релаксације R

Слика 2.4: Релаксација проблема P занемаривањем услова целобројности

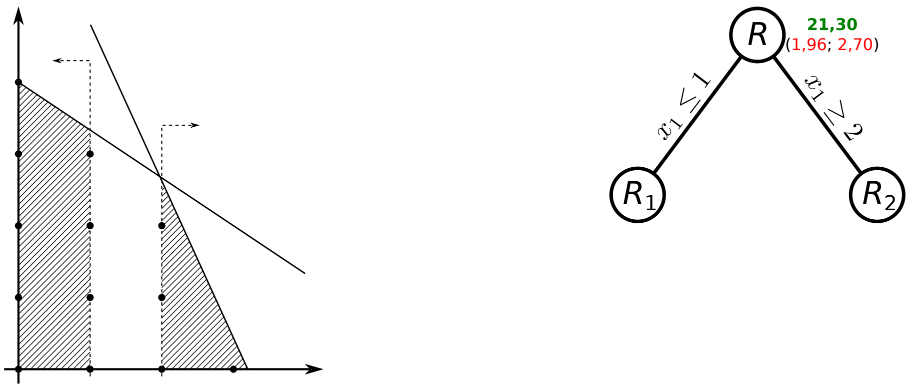
Решавањем проблема R добија се решење $x^R = (1,96; 2,70)$, $f\left(x^R\right) = 21,30$ (слика 2.5).



а) Оптимално решење проблема R *б) Прва горња граница проблема*

Слика 2.5: Разломљено оптимално решење релаксације R

Пошто решење није целобројно, уводимо ограничења која ће то да спрече. Почињемо са првом променљивом.[1] *Њена вредност је $1 < x_1^R < 2$, а мора да буде или $x_1 \leq 1$ или $x_1 \geq 2$. Додавањем ових ограничења допустивом скупу X релаксације R, ми га делимо на подскупове X_1 и X_2. На слици 2.6 се види и да је део скупа X за који је $1 < x_1 < 2$, који није допустив за проблем P, искључен из даље анализе, а да при томе није искључено ниједно допустиво решење проблема P.*



а) Дељење допустивог скупа X на X_1 и X_2 *б) Гранање стабла претраживања*

Слика 2.6: Дељење допустивог скупа и гранање

На овај начин формирамо два потпроблема:[2]

[1]Избор променљиве која има недопустиву вредност и за коју ће се уводити ограничења веома утиче на брзину извршавања алгоритма. Због тога су развијене различите стратегије које у зависности од врсте проблема дају различите резултате. Овде то неће бити разматрано пошто то не утиче на коначно решење и бирамо променљиве редом.

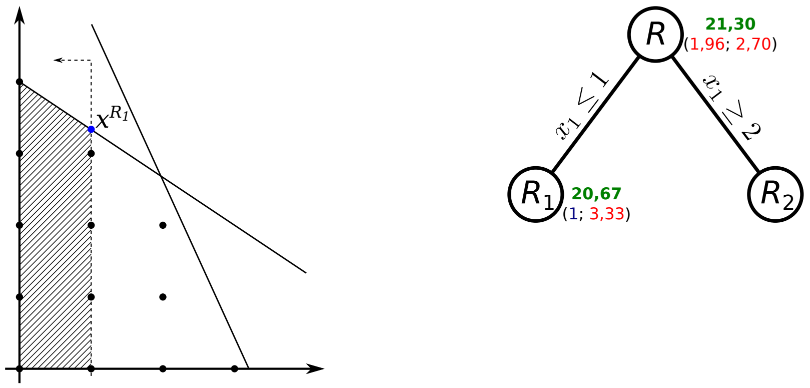
[2]Даље у овом примеру су у моделима потпроблема изостављена природна ограничења $x_j \geq 0$, која се подразумевају.

$$
\begin{aligned}
R_1: \quad & (\max) \quad f(x) = 4x_1 + 5x_2 \\
& \text{п.о.} \\
& 2x_1 + 3x_2 \leq 12 \\
& 11x_1 + 5x_2 \leq 35 \\
& x_1 \leq 1
\end{aligned}
$$

и

$$
\begin{aligned}
R_2: \quad & (\max) \quad f(x) = 4x_1 + 5x_2 \\
& \text{п.о.} \\
& 2x_1 + 3x_2 \leq 12 \\
& 11x_1 + 5x_2 \leq 35 \\
& x_1 \geq 2
\end{aligned}
$$

Решавамо прво проблем R_1.[3]

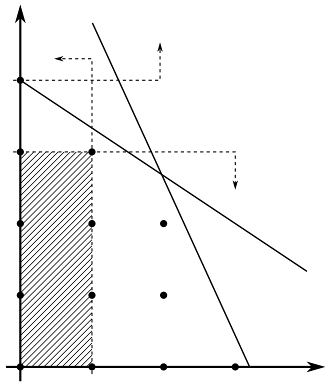


а) Решење потпроблема R_1 *б) Горња граница за потпроблеме R_1...*

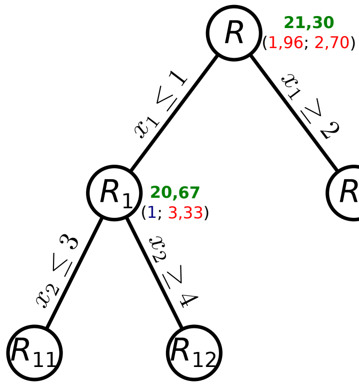
Слика 2.7: Решавање проблема R_1 и постављање горње границе за све његове потпроблеме

Добија се решење $x^{R_1} = (1; 3,33)$, $f\left(x^{R_1}\right) = 20,67$, коме је прва променљива целобројна, али друга променљива је разломљена (слика 2.7). Стога сада гранамо по променљивој x_2. Формирамо два нова потпроблема који деле скуп X_1 на подскупове X_{11} и X_{12}. При томе се један део допустивог скупа X_1 који није допустив за P, за који је $3 < x_2 < 4$, искључује из даље анализе (слика 2.8).

[3] Слично као и код избора променљивих по којима ће се гранати, и избор редоследа решавања потпроблема значајно утиче на ефикасност алгоритма. И за овај избор постоје различите методе. Без улажења у детаље, изабраћемо први од нерешених проблема.



а) Дељење допустивог скупа X_1 на X_{11} и X_{12}



б) Гранање проблема R_1

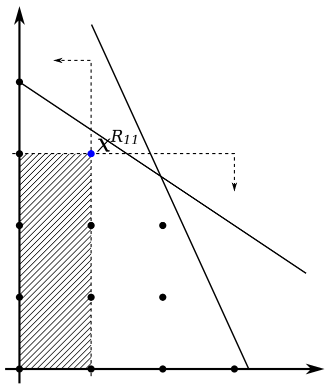
Слика 2.8: Дељење допустивог скупа X_1 и гранање стабла претраживања

Тиме формирамо потпроблеме:

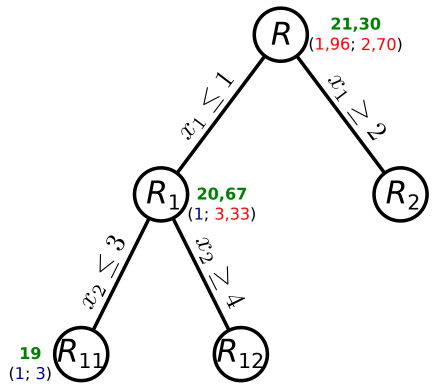
$$
\begin{aligned}
R_{11}: \quad & (\max) \quad f(x) = 4x_1 + 5x_2 \\
& \text{п.о.} \\
& \qquad 2x_1 + 3x_2 \leq 12 \\
& \qquad 11x_1 + 5x_2 \leq 35 \\
& \qquad x_1 \leq 1 \\
& \qquad x_2 \leq 3
\end{aligned}
$$

и

$$
\begin{aligned}
R_{12}: \quad & (\max) \quad f(x) = 4x_1 + 5x_2 \\
& \text{п.о.} \\
& \qquad 2x_1 + 3x_2 \leq 12 \\
& \qquad 11x_1 + 5x_2 \leq 35 \\
& \qquad x_1 \leq 1 \\
& \qquad x_2 \geq 4
\end{aligned}
$$



а) Решење потпроблема R_{11}

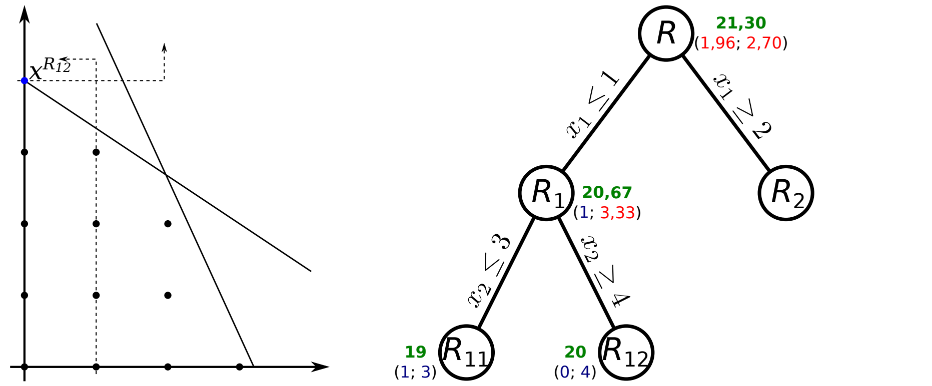


б) Доња граница за цео проблем

Слика 2.9: Решавање потпроблема R_{11} и постављање доње границе полазног проблема

Решавањем подпроблема R_{11} *први пут добијамо решење које је допустиво за* P, $x^{R_{11}} = (1;3)$. *Вредност функције циља тог решења* $f\left(x^{R_{11}}\right) = 19$ *постаје доња граница вредности оптималног решења проблема, тј. већ сада знамо да ће ова вредност бити у границама* $[19;21,30]$. *Подскуп* X_{11} *више нема потребе да делимо, пошто знамо да је решење* $x^{R_{11}}$ *најбоље целобројно решење у том подскупу.*

Решавамо проблем R_{12}.

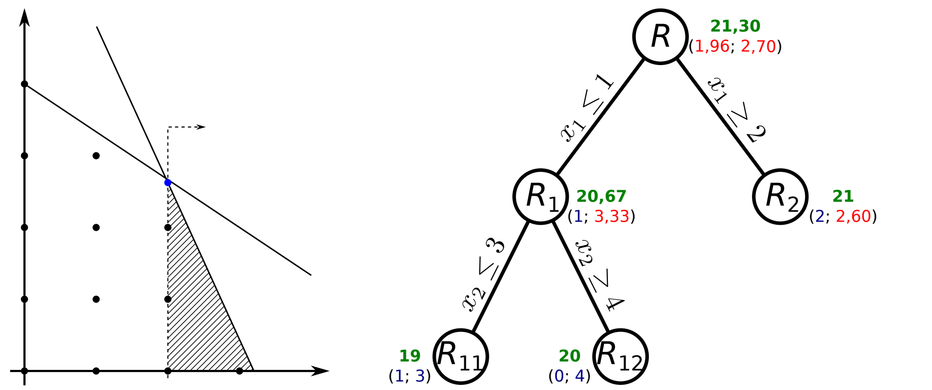


а) Решење подпроблема R_{11} б) Нова доња граница за цео проблем

Слика 2.10: Побољшавање доње границе проблема

Допустиви скуп X_{12} *садржи само једну тачку и она је (наравно) оптимална за проблем* R_{12}. *Решење* $x^{R_{12}} = (0;4)$ *је целобројно, а његова вредност* $f\left(x^{R_{12}}\right) = 20$ *је боља од претходне доње границе проблема 19. Овим поправљамо доњу границу, тј. до сада најбоље нађено допустиво решење полазног проблема.*

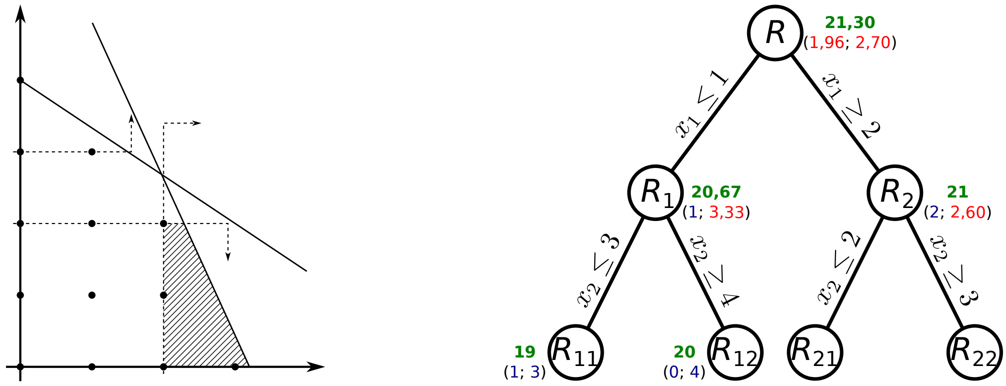
Остао је нерешен још проблем R_2.



а) Решење подпроблема R_2 б) Горња граница за R_2...

Слика 2.11: Решавање проблема R_2 и постављање горње границе за све његове подпроблеме

Његовим решавањем добија се решење $x^{R_2} = (2;2,60)$, $f\left(x^{R_2}\right) = 21$, *које је по првој променљивој допустиво, али друга променљива је разломљена (слика 2.11). Пошто* *је горња граница за све потпроблеме* R_2... *21, а доња граница целог проблема 20, још постоји шанса да се у том интервалу нађе боље решење. Гранамо по променљивој* x_2 *и формирамо два нова потпроблема који деле скуп* X_2 *на подскупове* X_{21} *и* X_{22} *(слика 2.12).*



а) Дељење допустивог скупа X_2 *на* X_{21} *и* X_{22} *б) Гранање проблема* R_2

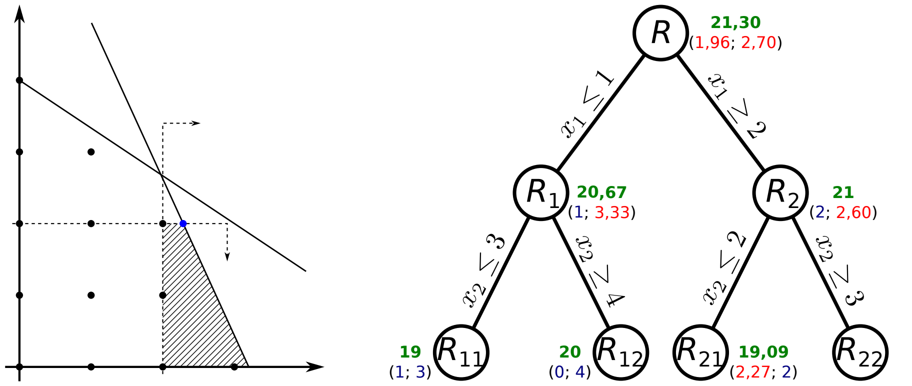
Слика 2.12: Дељење допустивог скупа X_2 *и гранање стабла претраживања*

Формирамо потпроблеме:

$$
\begin{array}{lrl}
R_{21}: & (\max) & f(x) = 4x_1 + 5x_2 \\
 & \textit{п.о.} & \\
 & & 2x_1 + 3x_2 \leq 12 \\
 & & 11x_1 + 5x_2 \leq 35 \\
 & & x_1 \geq 2 \\
 & & x_2 \leq 2
\end{array}
$$

и

$$
\begin{array}{lrl}
R_{22}: & (\max) & f(x) = 4x_1 + 5x_2 \\
 & \textit{п.о.} & \\
 & & 2x_1 + 3x_2 \leq 12 \\
 & & 11x_1 + 5x_2 \leq 35 \\
 & & x_1 \geq 2 \\
 & & x_2 \geq 3
\end{array}
$$



а) Решење потпроблема R_{21}

б) Горња граница потпроблема R_{21} је мања од доње границе P

Слика 2.13: Решавање потпроблема R_{21} и разрешавање подскупа X_{21}

Решавањем потпроблема R_{21} добијамо решење $x^{R_{21}} = (2,27; 2)$, $f\left(x^{R_{21}}\right) = 19,06$. Оно, не само да је разломљено, него је и лошије од доње границе проблема P. Ако је и најбоље решење над скупом X_{21} лошије од до сада најбољег нађеног решења, то значи да се у њему сигурно не налази оптимално решење полазног проблема.

Допустиви скуп X_{22} је празан (може се видети на слици 2.12), тако да проблем R_{22} нема оптимално решење.

Овим смо разрешили све потенцијалне делове допустиве области проблема P, односно R, и можемо закључити да је до сада најбоље нађено решење и оптимално за полазни проблем.

Закључак: $x^* = (0; 4)$, $f\left(x^*\right) = 20$. □

Пример 2.3 Задат је потпун граф са шест чворова и табела са дужинама грана:

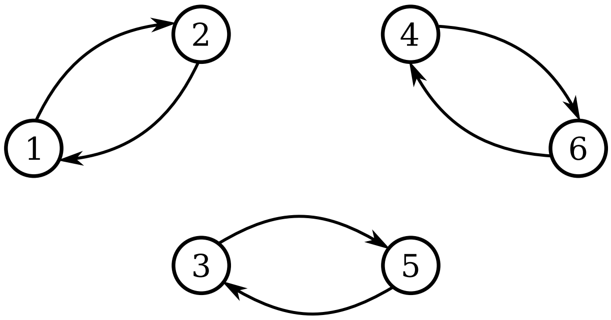
од\до	1	2	3	4	5	6
1	–	5	15	20	17	15
2	7	–	6	8	14	16
3	13	16	–	19	11	20
4	15	10	18	–	19	4
5	14	17	9	18	–	15
6	16	19	17	8	12	–

Одредити најкраћу Хамилтонову контуру у задатом графу.

Ово је проблем трговачког путника (Traveling Salesman Problem – TSP), а пошто је задата матрица дужина грана асиметрична, онда се ради о тзв. асиметричном проблему трговачког путника (ATSP). Овај проблем се може представити математичким моделом (1.5)-(1.9) (стр. 4). Занемаривањем ограничења (1.8) полазни проблем се своди на проблем асигнације. Овако добијена релаксација полазног проблема се може решити тзв. Кун-Мункрешовом (мађарском) методом која има сложеност $O\left(n^3\right)$, тј. која је полиномијална. Пошто гране (i, i), $i \in V$ не постоје, а за проблем асигнације је потребна матрица растојања са свим вредностима, може се увести да је $c_{ii} = M$, где *је M неки велики позитиван број који је значајно већи од осталих вредности матрице. Овим се постиже да се додељивање (i, i) никада не нађе у оптималном решењу, пошто оно у овом случају нема интерпретацију у полазном проблему. Тако формирамо полазну релаксацију R.*

Овај пример се разликује од претходних по томе што је у питању минимизација функције циља, па доња и горња граница имају обрнути смисао, и по томе што релаксирани проблем не решавамо алгебарски, него користимо специјализовани алгоритам.

Решавањем проблема асигнације се добија решење које се може приказати следећим графом, а вредност функције циља је 44.

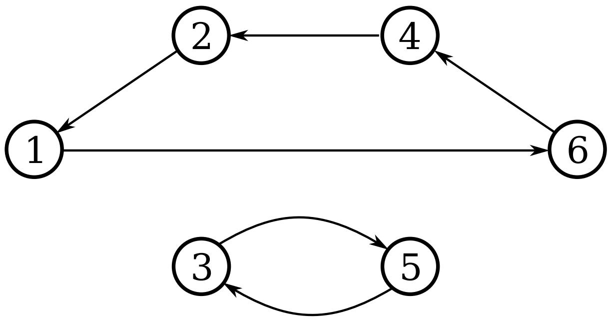


Слика 2.14: Графовски приказ решења прве релаксације R

На слици 2.14 се јасно види да добијено решење не представља Хамилтонову контуру. Решење је недопустиво зато што се састоји од неколико тзв. потконтура – контура које не садрже све чворове графа. Поступак ограничавања (рестрикције) се у оваквом случају састоји у избору једне потконтуре (обично оне са најмањим бројем чворова) и „забрањивању" појединих грана из те контуре. Полазни релаксирани проблем гранамо на онолико потпроблема колико изабрана контура има грана, „забрањујући" по једну грану у сваком од њих. У овом примеру ћемо изабрати контуру 1 – 2 – 1 и у првом потпроблему R_1 ће бити уведено ограничење $x_{12} = 0$, а у другом R_2, $x_{21} = 0$[4].

Применом Кун-Мункрешове методе на проблем R_1 добија се следеће решење.

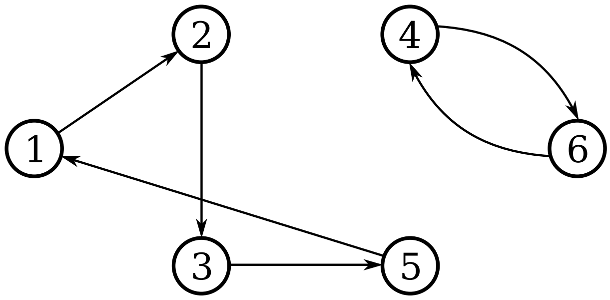
[4] *Имплементација ових ограничења се може састојати у додељивању веома великих вредности дужинама грана које се „забрањују" и решавањем проблема истим полиномијалним алгоритмом.*



Слика 2.15: Графовски приказ решења првог потпроблема R_1 *са додатим ограничењем* $x_{12} = 0$

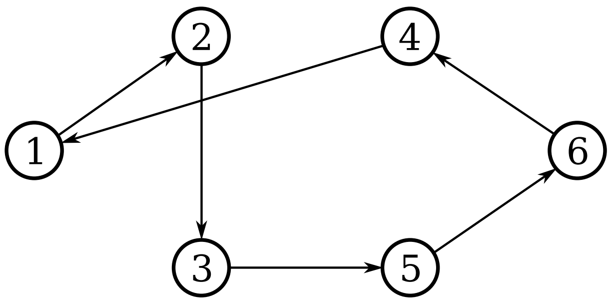
На слици 2.15 види се решење потпроблема R_1 *у коме је недопустива грана* $(1, 2)$. *Вредност решења је 60.*

На слици 2.16 је решење проблема R_2 *у коме је недопустива грана* $(2, 1)$. *Вредност решења је 48.*



Слика 2.16: Графовски приказ решења проблема R_2 *са ограничењем* $x_{21} = 0$

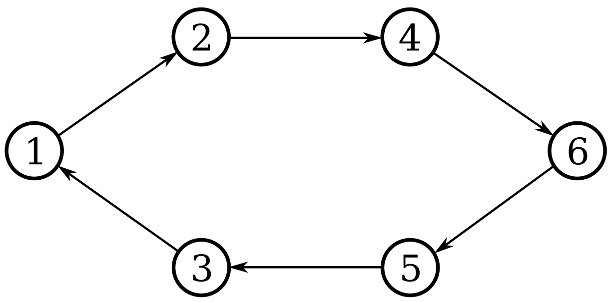
Ни једно од претходна два решења није допустиво, тако да настављамо са додавањем ограничења у потпроблему који је до сада имао најбољу вредност функције циља. У следећем кораку у решењу проблема R_2 *бирамо најкраћу потконтуру (4 − 6 − 4) и „забрањујемо" поједине гране из ње. Формираћемо проблем* R_{21} *у коме ће поред гране* $(2, 1)$ *бити недопустива и грана* $(4, 6)$. *Решење овог проблема приказано је на слици 2.17, а вреност функције циља је 60.*



Слика 2.17: Графовски приказ решења потпроблема R_{21}

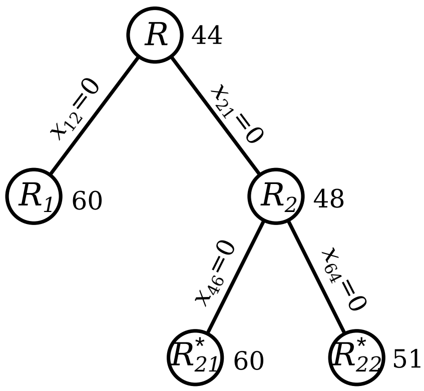
Са слике се види да решење представља Хамилтонову контуру, тј. оно је допустиво решење полазног проблема. Вредност решења постаје горња граница вредности оптималног решења. Даље, можемо закључити да гранање проблема R_1 не може довести до бољег решења (пошто су за то подстабло горња и доња граница једнаке).

Решње проблема R_{22} у коме су „забрањене” гране $(2, 1)$ и $(6, 4)$ је приказано на слици 2.18, а вредност решења је 51.



Слика 2.18: Графовски приказ решења потпроблема R_{22}

Пошто је ово решење и допустиво и боље од до сада најбољег добијеног допустивог решења, оно сада одређује горњу границу проблема (51), а пошто је она боља од свих доњих граница потпроблема које до сада нисмо гранали, констатујемо да је то оптимално решење полазног проблема.



Слика 2.19: Стабло претраживања и вредности функција циља потпроблема

На крају, на слици 2.19 је приказано стабло претраживања које је формирано решавањем овог проблема. Потпроблеми R_{21} и R_{22} имају оптимална решења која су допустива за полазни проблем, док остали потпроблеми имају решења која не формирају Хамилтонову контуру.

Закључак: *Најкраћа Хамилтонова контура у задатом графу је 1 – 2 – 4 – 6 – 5 – 3 – 1, а њена дужина је 51.* □

3 Алгоритам

Овде је приказан општи алгоритам решавања проблема из класе НПТ облика

$$P: \qquad \begin{array}{cc} (\max) & f(x) \\ \text{п.о.} & \\ & x \in X \end{array}, \tag{3.1}$$

дакле проблема максимизације функције циља над задатим допустивим скупом. Исти алгоритам се може применити и на проблеме минимизације, с том изменом што горња и доња граница мењају места и свим приказаним неједначинама се мења смер.

Алгоритам је приказан у облику псеудокода у коме се користе следеће ознаке:

P – полазни задати проблем,

X – допустиви скуп проблема P,

R – релаксација полазног проблема,

T – текући потпроблем који се решава,

$x,\ f$ – оптимално решење текућег проблема и вредност тог решења,

$\underline{x},\ \underline{f}$ – тренутно најбоље нађено допустиво решење проблема P и његова вредност,

$\overline{f}_R$ – горња граница потпроблема R,

$L = \left(\left(R_1, \overline{f}_{R_1}\right), \ldots, \left(R_l, \overline{f}_{R_l}\right)\right)$ – листа парова нерешених потпроблема и њихових горњих граница,

$|L|$ – број елемената листе L,

L_1 – први елемент листе L,

$\left(\left(R, \overline{f}_R\right)\right)$ – листа у којој се налази само проблем R са својом горњом границом,

$\leftarrow$ – додељивање вредности,

$\rightarrowtail$ – резултат дефинисане операције,

$\cup$ – унија, додавање елемента у листу,

$\setminus$ – скуповно одузимање, брисање елемента из листе.

Алгоритам 3.1 Метода гранања и ограничавања

Input: Полазни проблем P.
  Релаксирање: $P \rightarrowtail R$.
  $L \leftarrow ((R, \infty))$, $\underline{f} \leftarrow -\infty$, $\underline{x} \leftarrow \mathrm{N/A}$.
  while $|\underline{L}| > 0$ **do** ▷ Док има проблема у листи
    $\left(T, \overline{f}_T\right) \leftarrow L_1$, $L \leftarrow L \backslash \left(T, \overline{f}_T\right)$.
    Решавање: $T \rightarrowtail x, f$.
    if $f > \underline{f}$ **then**
      if $x \in X$ **then** ▷ Поправљање доње границе
        $\underline{x} \leftarrow x$, $\underline{f} \leftarrow f$.
        Из листе L избацити све елементе ▷ Одсецање
        чија је горња граница мања или једнака $\underline{f}$.
      else ▷ Недопустиво решење ⇒ гранање
        Изабрати део решења x који нарушава допустивост и
        формулисати ограничења $\{1, \ldots, p\}$ која то спречавају.
        За свако ограничење $k \in \{1, \ldots, p\}$ формулисати
        нови проблем T_k тако што се текућем
        проблему T дода k-то ограничење.
        $L \leftarrow L \cup ((T_k, f))$.
      end if
    end if
  end while

Output: $\underline{x}$ је оптимално решење проблема P, а $\underline{f}$ је вредност тог решења.

Оно што је упадљиво код овако записаног алгоритма је одсуство било какаве алгоритамске структуре типа стабла, која је коришћена у ранијем тексту, приликом објашњавања принципа методе и решавања примера. Наиме, гранање алгоритма (по коме је метода и добила име) је логички конструкт који је погодан за објашњавање основне идеје методе и њено повезивање са концептом стабла претраживања. Уместо хијерархијске структуре, алгоритам МГО користи концепт листе проблема, која је лакша за имплементацију и бржа за манипулацију каква је овде потребна.

Корак гранања је специфичан и његова имплементација се прилагођава типу проблема који се решава и релаксацији која је изабрана. Додатна ограничења се морају формулисати тако да се допустиви скуп надређеног проблема T подели на p дисјунктних делова[1], при чему део допустивог скупа проблема T који није допустив за проблем P, а у коме се налази решење проблема T, мора бити недопустив за добијене нове потпроблеме (видети нпр. пример 2.2, слика 2.6, стр. 10).

Редослед проблема у листи L је веома важан за ефикасност алгоритма. Време извршавања алгоритма може бити значајно скраћено ако се за критеријум сортирања листе нерешених проблема изабере одговарајуће правило или хеуристички алгоритам, што зависи од природе проблема или неких његових особина. На пример, са

[1] У поглављу 2 у примерима 2.1 и 2.2 увек је $p = 2$. У примеру 2.3 је такође $p = 2$, али само зато што су потконтуре по којима се гранало садржавале по две гране.

одговарајућим избором проблема се може раније добити боља горња граница и тиме у кораку одсецања смањити број проблема које треба решити. Иако то у алгоритму није експлицитно наведено, одсецање се врши и у случају када текући проблем T нема допустиво решење или када је вредност добијеног решења f мања од доње границе $\underline{f}$, тиме што се не додају нови потпроблеми у листу, а текући проблем се скида са листе.

МГО спада у експоненцијалне алгоритме, али се добрим стратегијама избора потпроблема и њеном хибридизацијом ефикасним хеуристикама, време њеног извршавања може свести на прихватљиву меру чак и за проблеме реалних димензија. Због ове особине, ова метода има широку примену у пракси.

И поред бројних унапређења које су уграђене у имплементације ове методе, перформансе МГО се још увек могу побољшати, тако да она и дан данас представља изазов за нова научна истраживања.