ЦЕЛОЧИСЛЕНО ЛИНЕЙНО ПРОГРАМИРАНЕ

Целочисленото програмиране (ЦП) разработва методи за решаване задачи, в които някои или зсички променливи приемат *целочислени* стойности. Задачата може да бъде *напьлно целочислена* или *частично целочислена* според това дали *ветки* или само *някои* променливи са це-лочислени, а моделите на тези два тип задачи се отнасят съответно към чист*ото* ЦП или *смесеното* ЦП (СЦП). Ако целевата функция и огра-ниченията са линейни, формулирана е задача на *целочисленото линейно програмиране* (ЦЛП). Най-често решавани са именно целочислените линейни задачи (ЦЛЗ).

Основните трудности при използването на алгоритмите на ЦЛП са свързани със значителния разход на машинно време. Затруднения възникват и поради грешки от закръглянето, което се извършва при машинното изпълнение на алгоритмите.

Един от най-простите начини за избягване на тези затруднения е задачата да се реши като задача на непрекъснатото (обикновеното) ЛП и получените оптимални решения да се *закрьглят* до най-близките цели стойности. Не съществува обаче гаранция, че такова приблизител-но решение ще удовлетворява ограниченията във вид на неравенства, а ограниченията във вид на равенства винаги ще бъдат нарушени.

Целочислените променливи често означават брой обекти (маши-ни, изделия, хора) и ако закръглените им стойности удовлетворяват ограниченията, те са допустими. Ако променливите обаче имат логически характер *(x* = ! или 0, дадена работа се включва или не се включва в плана), дробните стойности нямат смисъл в схемите на фор-малната логика, а типичните ограничения във вид на равенства биха били нарушени от закръглените стойности.

Представа за някои области на приложение и за важността на ме-тодите на ЦП дават разгледаните примери по-нататък.

7.1. НЯКОИ ЗАДАЧИ И ПРИЛОЖЕНИЯ НА **ЦЕЛОЧИСЛЕНОТО ПРОГРАМИРАНЕ**

Често целочислените задачи възникват по естествен път, напри-мер, когато определяните решения са от типа "да-не". Съществен ин-терес обаче представляват и "изкуствените" задачи, в които изкуствено се въвеждат двоични (булеви) променливи, за да се преодолеят някои аналитични трудности или некоректности на началния модел.

7.1.1. ЗАДАЧА ЗА ИЗБОР НА ПРОЕКТИ

Обсъждат се *n* проекта за изпълнение през следващите *m* годи-ни. В табл.7.1 са представени годишните разходи и очакваният доход за всеки проект, както и достъпните средства през годините на изпъл-нение. Необходимо е да се определи кои от n-те проекта би трябвало да се изпълняват през m-годишния период, ако критерий за избор е сумарният доход.

Разходи за *Таблица 7.1*

Проект	Година 1	Година 2		Година m	Доходи
1	c_{11}	c_{12}		c_{1m}	R_1
2	c_{21}	c_{22}		c_{2m}	R_2
.		.		.	.
n	c_{n1}	c_{n2}	...	c_{nm}	Rn
Достъпни средства	B_1	B_2	...	B_m	

Вижда се, че за всеки проект се взема решение от типа "да-не". Такова решение се представя чрез двоичната променлива x_j, която се поставя в съответствие на j-тия проект, $j=\overline{1,n}$ и чиито стойности 1 или 0 кодират решенията "да" или "не".

Тогава моделът на задачата е от вида

Да се максими $J=\sum_{i=1}^{n} R_i x_i,$

ограниченията

$$\sum_{i=1}^{n} c_{ij} x_i \le B_j, \quad j=\overline{1,m},$$

$$x_i = 0,1, \quad i=\overline{1,n}.$$

Дробни стойности на x_i нямат смисъл, защото стойностите 1,0 кодират решенията от тип "да-не".

7.1.2. ЗАДАЧА ЗА ПЛАНИРАНЕ ПРИ ПОСТОЯННИ ЕЛЕМЕНТИ НА РАЗХОДИТЕ

Често разходите за производство на j-тия вид продукция в обем x_i могат да се представят като сума от постоянна и променлива част

$$C_j(x_j)=\begin{cases} k_j + c_j x_j, & x_j > 0, \\ 0 \quad , & x_j = 0, \end{cases}$$

където k_i са постоянни разходи, конто не зависят от количеството x_i на продукцията, а c_i са разходите за производство на единица продукция. Целта би могла да се запише във вида:

$$J=\sum_{j=1}^{N} c_j(x_j),$$

Да се минимизира

където *N* е броят на видоветепродукция. Целевата функция *J* не е линейна по x_i поради прекъснатостта и в началото на координатна-та система и използването на ЛП за определяне на стойностите x_i е невъзможно.

Преодоляването на това аналитично затруднение е възможно по следния начин. Нека y_j са допълнителни двоични променливи

Тези условия могат да се запі ѝно ограничение

$$y_j=\begin{cases} 0, & x_j = 0, \\ 1, & x_j > 0. \end{cases}$$

където $M > 0$ е достатъчно голямо, така че условието $x_j \le M$ винаги да се изпълнява. Тогава началната задача се преобразува в задача на СЦП (или в нова напълно целочислена задача, ако всички x_i са целочислени) от следния вид:

Да се минимизира $J=\sum_{j=1}^{N}(c_j x_j + k_j y_j)$

при ограниченията а) ограниченията на началната задача; б) новите ограничения

$$\left.\begin{aligned} x_j - M y_j &\le 0 \\ y_j &= 0,1 \end{aligned}\right\} \quad \text{за всички } j, \text{ за които } k_j > 0.$$

Преобразуването е коректно, защото новите ограничения осигуря-ват това, че $y_j = 1$, когато $x_j > 0$ и тогава k_j се включва в целевата Функция. А при $x_j = 0$, когато ограниченията позволяват да се избира между $y_i = 0$ и $y_i = 1$, изборът $y_i = 0$ осигурява по-малка стойно J от $y_i = 1$ (тъй като $\kappa_i > 0$) и оптимизиращият алгоритъм ще избере стойността $y_i = 0$.

Примерът илюстрира построяването на изкуствена задача на СЦП чрез използването на "излишни" променливи y_i, конто не носят нова информация за оптималното решение на началната задача, но правят възможно получаването му.

7.1.3. ЗАДАЧА ЗА СЪСТАВЯНЕ НА ПРОИЗВОДСТВЕНО РАЗПИСАНИЕ

Върху една машина се изработват няколко крайни продукта. За всеки продукт са известии редът на изпълнение на различните техно-логични операции и срокът на доставка. Необходимо е да се състави разписание за изпълнение общо на *n* различии операции върху маши-ната за минимално възможно време.

При съставянето на модела на задачата се отчитат три типа ограничения. Първият тип е свързан с реда на изпълнение на операциите, който се определя от технологичните изисквания. Нека x_j е времето на началото на j-та операция, $j = \overline{1,n}$, а a_j - продължителността на ней-ното изпълнение. Ако операцията i трябва да предшествува операцията j, съответното ограничение е

$$x_i + a_i \le x_j.$$

Вторият тип ограничения отразява факта, че две операции не мо-гат да се изпълняват едновременно на една машина. За операциите i и j

$$\text{или } x_i \ge x_j + a_j, \text{ т.е. } x_i - x_j \ge a_j$$
$$\text{или } x_j \ge x_i + a_i, \text{ т.е. } x_j - x_i \ge a_i$$

в зависимост съответно от това дали *j* е преди г, или г е преди *j* в оптималното решение.

Такива логически ограничения от вида "или-или" пораждат *не-изпъкнали* пространства на допустимите решения, при конто ЛП е не-приложимо. За преодоляването на тази трудност се въвежда двоичната променлива y_j,

$$y_{ij} = \begin{cases} 0, & \text{ако операцията } j \text{ е преди операцията } i, \\ 1, & \text{ако операцията } i \text{ е преди операцията } j. \end{cases}$$

Нека $M > 0$ е достатъчно голямо. Тогава ограничението "или-или" е еквивалентно на следните две *едновременни* ограничения

$$My_{ij} + x_i - x_j \ge a_j \;\; и \;\; M(1 - y_{ij}) + x_j - x_i \ge a_i.$$

Нека оптималното решение е $y^*_{ij} = 0$. Тогава второто ограничение е излишно (в смисъл, че винаги се изпълнява), а първото ограничение е активно. Ако $y^*_{ij} = 1$, първото ограничение е излишно, а второто е активно. Вижда се, че въвеждането на двоичната промеклива y_{ij} логическото ограничение се преобразува в обикновени ограничения от модел на СЦП.

Третият тип ограничения отчита зададените срокове за изпълне-ние на операциите, конто произтичат от срока на доставка. Ограниченията са от вида

$$x_i + a_i \le d_i,$$

където d_j моментът от време, когато трябва да завърши изпълнението на **j-та** операция.

Променливите, свързани с избора на решение, са $x_i, \; i = \overline{1,n}$. При-ема се, че първият начален момент от последователността е равен на нула, $x_1 = 0$.

Нека *t* е сумарното време за изпълнение на всичките п операции. Гогава моделът на задачата е от вида

Да се минимизира $J = t$ при ограниченията

$$x_j + a_j \le t, \quad j = \overline{1,n}$$

плюс ограниченията от трите типа, разгледани по-горе.

Въвеждането на спомагателните двоични променливи позволява изразяването на комбинационни отношения. В разгледания пример се оценяват две комбинации - на всички останали ограничения най-на-пред с първото, а след това с второто от двете алтернативни ограничения ("или - или"). Решението от тип "да - не" коя от тези две комбинации е по-добра от гледна точка на стойността на *J* се представя от променливата y_{ij}.

7.1.4. ЗАДАЧА ЗА **ИЗБОР НА *К* ОТ *N* ОГРАНИЧЕНИЯ**

Разглежда се случай, когато е необходимо да се изпълняват К от *N* ограничения, $K < N$ от вида

$$f_i(x_1, x_2, \ldots, x_n) \le d_i, \quad i = \overline{1,N}.$$

Предварително не е известно кои от ограниченията трябва да се изпъл-няват и една част от оптимизационния процес се състои в определянето На такава *комбинация от К ограничения,* при която целевата функция има най-добра възможна стойност. Въвежда се двоичната променлива

$$y_i = \begin{cases} 0, & \text{ако } i\text{-то ограничение се изпълнява} \\ 1, & \text{ако } i\text{-то ограничение не се изпълнява} \end{cases}$$

Ако $M > 0$ е достатъчно голямо, изпълнението на K от N-те ограничения се осигурява от условията

$$f_i(x_1, x_2, \ldots, x_n) \le d_i + My_i, \quad i = \overline{1,N}$$
$$y_1 + y_2 + \ldots + y_N = N - K,$$
$$y_i = 0, 1, \quad i = \overline{1,N}$$

Вижда се, че в десните части на $N - K$ ограничения ще се появяват стойностите $d_i + M$ и тези ограничения ще станат излишни.

7.1.5. ФУНКЦИИ С N ВЪЗМОЖНИ СТОЙНОСТИ

Нека е необходимо функцията f да има една от N зададени въз-можни стойности $d_i,\ i = \overline{1,N}$, т.е.

$f(x_1, x_2, ..., x_n) = d_1$ или d_2, или ..., d_N. Това може да бъде

постигнато чрез формулиране на задача на ЦП

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{N} d_i y_i$$
$$\sum_{i=1}^{N} y_i = 1$$
$$y_i = 0, 1, \quad i = \overline{1,N},$$

където

$$y_i = \begin{cases} 1, & \text{ако } d_i \text{ е стойността на } f, \\ 0, & \text{ако } d_i \text{ не е стойността на } f. \end{cases}$$

Тези нови условия се включват към ограниченията на решавана-та задача. Подобно на предишния пример, предварително не е известно коя е стойността на функцията и една част от оптимизационния про-цес се състои в избирането на такава стойност сред *N-те* зададени, при която целевата функция на решаваната задача има най-добра стойност.

По същия начин се решава и задачата за избиране на една от ня-колко възможни стойности на дясната част на дадено ограничение.

7.2. МЕТОДИ ЗА РЕШАВАНЕ НА ЗАДАЧИ НА ЦП

Целочислената природа на променливите в задачите на ЦП затруднява разработването на ефективни алгоритми, конто търсят оптималното решение директно сред допустимите целочислени точки на пространството на решенията. Затрудненията се дължат на много голелия брой комбинации от стойности на променливите, конто трябва да се оценят. Поради това, подходът за решаване на целочислени задачи се основава на мощния резултат от непрекъснатото ЛП, че оптималното решение се намира в ъглова точка на пространството на решенията. Този подход може да бъде систематизиран по следния начин.

1. "Отслабват" се ограниченията на началната целочислена задача, като се отстраняват изискванията за целочисленост на променливите. В резултат на това възниква задача на непрекъснатото ЛП.
2. Решава се "отслабената" задача и се определя непрекъснатото оптимално решение.
3. Като се започва от непрекъснатото оптимално решение, последователно се въвеждат специални допълнителни ограничения, които отчитат изискванията за целочисленост и деформират допустимата област на отслабената задача дотогава, докато оптималното решение стане целочислено.

Търсенето на целочисленото оптимално решение започва от непрекъснатото поради това, че твърде често двете оптимални решения са разположени близо едно до друго, което пък позволява целочисленото решение да бъде определено по-бързо.

Същността на подхода е в решаването на последователност от задачи на непрекъснатото ЛП, което от изчислителна гледна точка е по-ефективно от директното търсене на целочислени решения.

Специалните ограничения се построяват, като се използуват два метода: *метода на клоните и границите* и *метода на отсичащите равнини.* И при двата метода добавяните специални ограничения отстраняват части от допустимата облает на отслабената задача, конто не съдържат точки с целочислени координати.

От гледна точка на изчислителни загуби и точност методът на клоните и границите е значително по-ефективен от метода на отсича-щите равнини и е в основата на повечето програмни продукта за решаване на задачи на ЦП. Като цяло обаче (с изключение на някои специални задачи) ефективността на съществуващите методи за решаване на задачи на ЦП е недостатъчна. Ето защо създаването на по-ефективни Методи е област на значителен изследователски интерес.

7.3. МЕТОД НА КЛОНИТЕ И ГРАНИЦИТЕ

Методът е приложим за решаване на двата вида задачи - напълно целочислени и частично целочислени. Същността му ще бъде обяснена с помощта на пример.

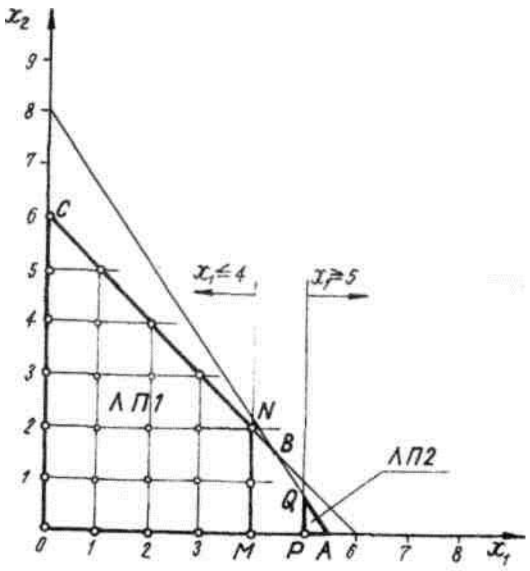
Пример 7.1. Разглежда се следната задача на ЦЛП

Да се максимизира J = $6x_1 + 5x_2$ при ограничения

$$x_1 + x_2 \leq 6,$$

$$16x_1 + 11x_2 \leq 88,$$

$$x_1, x_2 \geq 0, \ \textbf{цели}$$



Фиг. 7.1

На фиг 7.1. е показано с точки пространството на решенията на задачата на ЦЛП. Пространството на решенията на отслабената задача ЛПО на непрекъснатото ЛП, която възниква след отстраняване на изис-кванията за целочисленост, е четириъгълникът *О АВС*. Оптималното решение на задачата ЛПО, получено по симплекс-метода, е x^*_1= 4,40, x^*_2 — 1, 60, J^* = 34,40.

Тъй като решението на ЛПО не е целочислено, пространството на решенията на ЛПО се изменя по такъв начин, че да се подобрят възмо-жностите за определяне на целочислено решение. За целта произволно се избира една от променливите, която в точката на оптимума на ЛПО не е цело числена, да генерира необходимите изменения. Ако например се избере $x_1 (x^*_1$ = 4,40), вижда се, че областта 4 < x_1 < 5 от простран-ството на решенията на ЛПО не съдържа целочислени решения и може да бъде отстранена. Очевидно, целочислените решения трябва да удо-влетворяват едно от следните две условия $x_1 \leq 4$ **или** $x_1 \geq 5$,което позволява началното пространство ПрЛПО на задачата ЛПО да се замени с две пространства ПрЛП1 и ПрЛП2

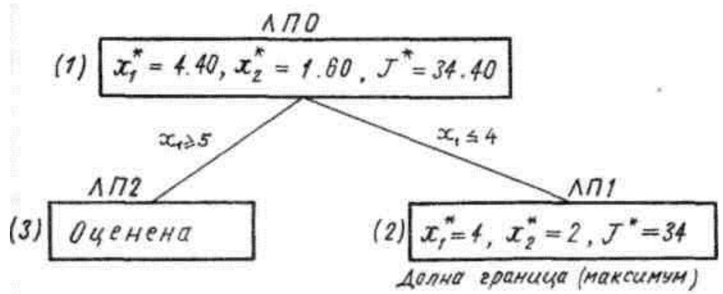
ПрЛП1=ПрЛП0 и ограничението ($x_1 \leq 4$),

ПрЛП2=ПрЛП0 и ограничението ($x_1 \geq 5$).

По този начин отслабената непрекъсната задача ЛПО се заменя с две нови задачи на непрекъснатото ЛП - задачите ЛП1 и ЛП2. На фиг.7.1 пространствата ПрЛП1 и ПрЛП2 са представени съответно от четириъгълника *OMNC* и триъгълника *PAQ,* означени са и задачите ЛП1, ЛП2, породени от новите пространства. Вижда се, че двете про-ранства съдържат всички допустими целочислени решения на начал-ната ЦЛЗ и в този смисъл са еквивалентни на ПрЛП0.

Тъй като двете нови ограничения $x_1 \leq 4$ и x1 ≥ 5 се изключват взаимно, задачите ЛП1 и ЛП2, които възникват след поотделното въве-ждане на тези ограничения в задачата ЛПО, са различии и се решават поотделно.

Описаното разделяне на текущото пространство на решенията на две взаимно изключващи се подпространства и свързаното с това по-строяване на две нови задачи (подзадачи) на ЛП от началната (текуща-та) задача се означават с понятието *разклоняване.* На фиг.7.2 е показано разклоняването на ЛПО на двете подзадачи ЛП1 и ЛП2, като клоните се определят от ограниченията $x_1 \leq 4$ и $x_1 \geq 5$. Процесът на разклоняване се представя чрез *дърво,* всеки възел на което съответствува на подзадача. Самото дърво се нарича *дърво на решението* (или дърво на изброяването). Променливата x_1, използвана за да се извърши разклоняването, се нарича *променлива на разклоняването.*



Фиг. 7.2

Да предположим, че избираме произволно най-напред да бъде решена подзадачата ЛП1

Да се максимизира $J = 6x_1 + 5x_2$ при ограничения

$$x_1 + x_2 \le 6$$

$$16x_1 + 11x_2 \le 88$$

$$x_1 \le 4$$

$$x_1, x_2 \ge 0.$$

Разликата от ЛПО е тази, че тук е включено ново ограничение което не се удовлетворява от оптималното решение на ЛПО $x^*_1 = 4{,}40$. В т.4.4.2 беше представено използването на дуалния симплекс-метод когато в модела на задачата е необходимо да се добави ново ограничение. Като се приложи описаната процедура, се определя оптималното решение на ЛП1

$$x^*_1 = 4, \quad x^*_1 = 2, \quad J^* = 34.$$

Полученото решение е целочислено и подзадачата ЛП1 се означава като *оценена (измерена),* в смисъл, че тя няма да се разглежда по-ната-тък, тъй като не може да даде *по- добро* решение.

Целочисленото решение на ЛП1 определя *долна граница* за оптималната стойност на целевата функция на задачата на ЦП. Понятието "граница" е следващото важно понятие за разглеждания метод. Гра-ницата позволява да се отстраняват както неперспективни задачи, конто не биха могли да дадат *по- добро* решение и няма смисъл да бъдат решавани, така и получаваните нови целочислени решения, ако те не са по-добри от съществуващото, което определя границата. В случая долната граница (задачата е за максимизация) е J^* — 34. Тъй като оптималното решение на ЛПО $J^* = 34{,}40$ и коефициентите в целевата функция са цели числа, от ЛПО не може да произлезе задача, която да има по-добро решение от $J^* = 34$. В този смисъл задачата ЛП2 може да бъде отстранена без решаване, като неперспективна, тъй като нейното решение не би могло да бъде по-добро от това на задачата ЛП1 $J^* = 34$. Задачата ЛП2 също е *оценена,* т.е. няма да се разглежда по-нататък.

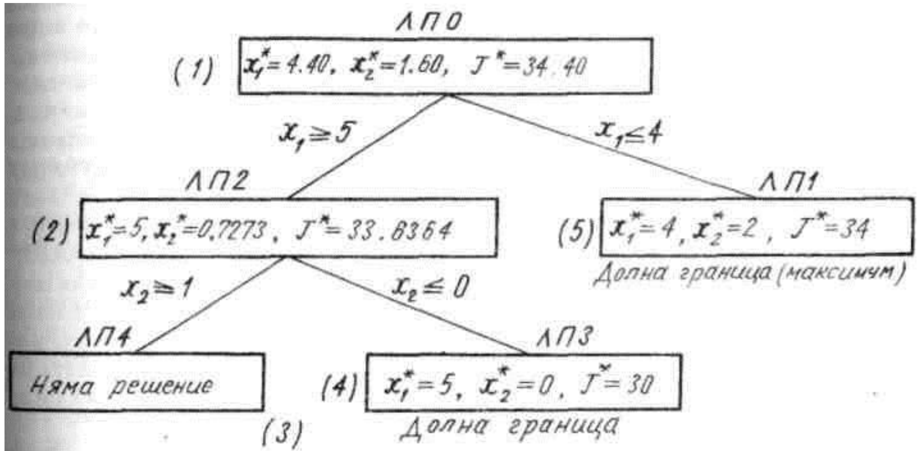
От изложеното се вижда, че една подзадача е *оценена,* ако е получено нейното допустимо целочислено решение или ако е показано, че тя не може да даде по-добро решение от това, което е определило текущата граница. Тя е оценена също, ако въобще няма допустимо решение.

В разглеждания пример всички подзадачи са оценени и оптималното решение на началната ЦЛЗ е това, което съответствува на текущата долна граница - решението на ЛП1.

В случая случайният избор на *X* като променлива на разклоня-ването и на задачата ЛП1 като първа за решаване от двете нови задачи беше твърде сполучлив. Той позволи бързо да се получи допустимо целочислено решение и да се оцени и втората подзадача ЛП2. Като променлива на разклоняването обаче би могла да бъде избрана промен-лиивата x_2, а като първа за решаване - подзадачата ЛП2, което би довело до значително по-различен изчислителен процес от разглеждания. Не-ка например най-напред да се решава задачата ЛП2 (фиг.7.3). Нейното оптимално решение е $x^*_1 = 5$, $x^*_2 = 0{,}7273$ и $J^* = 33{,}6364$. Тъй като стойността x^*_2 не е целочислена, x_2 се избира като променлива на разклоняване и от подзадачата ЛП2 възникват двете нови подзадачи ЛПЗ и ЛП4, съответно с клони $x_2 \le 0$ и $x_2 \ge 1$. Техните пространства на решение са ПрЛПЗ = ПрЛП2 и ограничението
($x_2 \le 0$) =

= ПрЛП0 и ограничението ($x_1 \ge 5$) и ограничението ($x_2 \le 0$); ПрЛП4 = ПрЛП2 и ограничението ($x_2 \ge 1$) =

= ПрЛП0 и ограничението ($x_1 \ge 5$) и ограничението ($x_2 \ge 1$).



Фиг. 7.3

Нека като първа се решава подзадачата ЛП4. Прилагането на дуалния симплекс-метод показва, че тази подзадача няма допустимо целочислено решение, т.е. тя е оценена. Да предположим, че като следваща за решаване произволно се избира подзадачата ЛПЗ. Нейното решение $x^*_1 = 5$, $x^*_2 = 0$ и $J^* = 30$ удовлетворява изискванията за целочисленост. Това решение определя текуща долна граница $J^* = 30$ за оптималната стойност на целевата функция на ЦЛЗ. Последна се решава подзадачата ЛП1. Както беше показано по-горе, нейното оптимално решение е $x^*_1 = 4$, $x^*_2 = 2$, $J^* = 34$ и то определя нова, по-добра долна граница. Тъй като всички задачи са оценени, решението на началната ЦЛЗ се определя от оптималното решение, което съответствува на текущата Долна граница - решението на ЛП1.

Разгледаният изчислителен процес премина през решаването на задачите ЛПО->ЛП2->ЛП4->ЛПЗ->ЛП1 и е значително по-неефективен от първия процес ЛП0->ЛШ-»ЛП2. Той илюстрира един основен не-достатък на метода на клоните и границите - липсата на обосновани правила за избиране на променливата на разклоняване при дадена под-задача и на следващата за решаване подзадача сред останалите нереше-ни подзадачи. Според едно най-просто правило целочислените промен-ливи се номерират и подреждат в естествен ред *x1,x2, ... ,*x_n и сред тези от тях, конто в оптималното решение на съответната подзадача имат нецелочислени стойности, се избира първата в естествения ред (тази с *най- мальк номер)* като променлива на разклоняването.

Съществуват евристични правила за избор на перспективни клони, конто невинаги са резултатни при решаването на общата задача на ЦЛП.

Алгоритъмът на клоните и границите се използува и за решаване на задачи на СЦП, като непрекъснатите променливи (тези, конто могат да имат дробни стойности) не се избират като променливи на разклоняване. Нека да изменим разглеждания пример така, че променливата x_1 да е целочислена, x_2- непрекъсната. Подзадачата ЛП1 дава допустимо решение $x_1 = 4$ и долна граница $J^* = 34$. В случая обаче подзадачата ЛП2 не може да бъде оценена без да се реши, тъй като целевата функция може да има нецелочислени стойности. Решението на тази подзадача е $x^*_1= 5$, $x^*_2 = 0{,}7273$ и $J^* = 33{,}6364$. Вижда се, че решението на ЛП1 е по-добро и то е оптималното решение на разгледаната задача на СЦП.

Често оптималната стойност *J**, подучена при решаването на под-задача, която е възникнала от начална задача на СЦП, се нарича *граница на подзадачата.* Ако началната задача е напълно целочислена, грани-цата на възникнала от нея подзадача се определя, като подучената за подзадачата стойност J* се закръгли до най-близкото по-малко цяло число.

Методът на клоните и границите е предложен от Ланд и Доиг (1960). Съвременните реализации се основават на алгоритъма на Да-кин (1965). В случая на задача за максимизация този алгоритъм може да бъде представен в следния обобщен вид:

Начална стъпка. Полага се долната граница $J^* = -\infty$. Решава се отслабената задача ЛПО. Ако ЛПО е оценена, изчисленията се прекра-тяват, в обратния случай се преминава към стъпка 1.

Стъпка 1. Разклоняване. Избира се подзадача сред оставащите не-оценени подзадачи (например последната създадена). Избира се една от целочислените променливи x_i, чиято стойност x^*_j в оптималното решение на подзадачата не е целочислена (например първата такава в естествения ред на целочислените променливи). Построяват се две нови подзадачи чрез включване към избраната подзадача на едно от ограни-ченията $x_j \le [x^*_i]$ и $x_j \ge [x^*_i]+1$, където *[*x^*_i*]* означава най-голямото цяло число $\le x^*_i$.

Стъпка 2. Ограничаване. Определи се границата на всяка от двете нови подзадачи чрез решаването и с използуване на дуалния симплекс-метод. В случая на задача на чистото ЦП стойността на J се закръгля до най-близкото по-малко цяло число, а при задача на СЦП закръгление не се извършва.

Стъпка 3. Оценяване. За всяка от новите подзадачи се проверява изпълнението на следните условия:

а) границата и е $\le$ J*, където *J** е текущата долна граница на началната задача;

б) подзадачата няма допустимо решение;

в) оптималното и решение отговаря на изискванията за целочи-сленост.

Ако се изпълнява кое и да е от тези условия, подзадачата е оценена. Оценените подзадачи се отстраняват от разглеждане. Ако подзадачата е оценена вследствие изпълнение на условие в) и оптималното и решение е по-добро от досегашното, то го заменя, а съответствуващата му стойност на целевата функция става нова текуща граница *J**. В този случай условие а) се проверява отново за всички неоценени подзадачи с новата, по-добра стойност на *J**.

Стъпка 4. Проверка на оптималността. Ако няма повече неоценени подзадачи, изчисленията се прекратяват, като решението, което съответствува на текущата долна граница, е оптималното решение на началната напълно или частично целочислена задача. В обратния случай се преминава към стъпка 1.

Алгоритъмът лесно се модифицира за решаване на задачи за ми-нимизация. Въвежда се понятието *горна граница J** за оптималната стойност на целевата функция на началната задача и условие а) от стъпка 3 на алгоритъма се изменя във вида

границата на подзадачата $\ge J^*$.

Алгоритъмът може да се използува и за определяне на *квазиопти-мално* решение, основното предимство, при което са намалените изчи-слителни загуби. Едно решение е "достатъчно добро", ако съответната стойност J е "достатъчно близко" до оптималната (или някаква жела-на) стойност *J***, т.е. ако се изпълнява

$$J \ge J^{**} - M \text{ или } J \ge (1-\alpha)J^{**}$$

при зададени положителни константи М и α.За определяне на ква-зиоптимално решение условие а) в стъпка 3 на алгоритъма се заменя с едно от двете условия

границата на подзадачата $\le J^* + M$

или $(1 - \alpha)$ границата на подзадачата $\le$ J*, като условието се проверява след проверката на условие в) и установя-ването евентуално на нова долна граница. Ускорението се дължи на това, че алгоритъмът оценява (отстранява от разглеждане) дадена подзадача, ако нейната граница не е "достатъчно по-добра" от текущата долна граница.

7.4. АЛГОРИТМИ, КОИТО РЕАЛИЗИРАТ МЕТОДА НА ОТСИЧАЩИТЕ РАВНИНИ

Методът на отсичащите равнини е предложен от Гомори. Основната идея ще бъде разкрита с пример.

Необходимо е да се максимизира

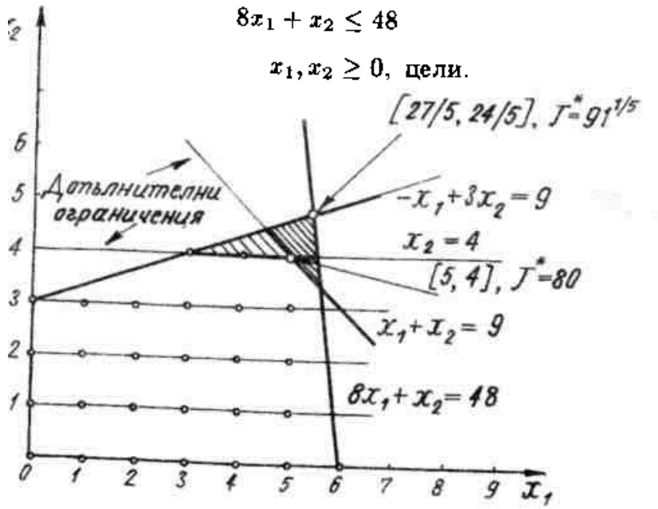
$$J = 8x_1 + 10x_2$$

при ограничения

$$-x_1 + 3x_2 \le 9$$

$$8x_1 + x_2 \le 48$$

$$x_1, x_2 \ge 0, \text{ цели}.$$



Фиг. 7.4

Пространството на допустимите решения на задачата с отслабени ограничения е показано на фиг.7.4. Нейното оптимално решение е $x^*_1 = 5^{2/5}$, $x^*_2 = 4^{4/5}$ и $J^* = 91^{1/5}$. Началното пространство се преобразува в нов изпъкнал многостен (в случая - многоъгълник), чиято екстремна точка е оптималното решение на началната целочислена за-дача. При това всички нейни допустими целочислени решения лежат вътре във или на границата на многостена. На фиг.7.4 е показано как въвеждането на две нови подходящи ограничения позволява да се получи нова екстремна точка (5,4), която е оптималното решение на началната задача. Защрихованата "отсечена" област не съдържа точки с целочислени координати.

7.4.1. ПЪРВИ (ДРОБЕН) АЛГОРИТЪМ НА ГОМОРИ

Алгоритъмът е предназначен за решаване на напълно целочисле-ни задачи. Необходимо условие за използването му е *коефициентите и десните части на ограниченията да са целочислени.* Например ограничението

1 13

се записва във вида

$$6x_1 + 2x_2 \le 39.$$

Наличието на дробни коефициенти би довело до нарушаване на целочислеността на допълнителните променливи, които се въвеждат за решаване на задачата и е недопустимо.

На първата стъпка на алгоритъма се решава задачата с отслабени ограничения. Ако полученото решение е целочислено, то е и решението на началната задача. Иначе се въвеждат последователно допълнителни ограничения, при което възниква нова задача на ЛП, чието решение е целочислено и съвпада с оптималното решение на началната задача на чистото ЦП.

Нека симплекс-таблицата, която съдържа оптималното решение на задачата с отслабени ограничения, е от вида

Таблица 7.2.

БПР	x_1	... x_i ...	x_m	z_1	... z_j ...	z_n	Решение
J	0	... 0 ...	0	c_1	... c_j ...	c_n	b_0
x_1	1	... 0 ...	0	a^1_1	.... a^j_1...	a^n_1	b_1
.	.						
x_i	0	.. 1 ...	0	a^1_i	... a^j_i ...	a^n_i	b_i
.	.						
x_m	0	... 0 ...	1	a^1_m	a^j_m	a^n_m	b_m

Където x_i, $i = 1,m$ са *базисните,* а $z_j, j = 1,n$ - *небазисните* променливи. Да предположим, че на i-тия ред е получена нецелочислена стойност b_i На базисната променлива x_i. Тази променлива може да бъде изразена чрез небазисните променливи

$$x_i = b_i - \sum_{j=1}^{n} a_i^j z_j, \quad b_i \text{ – не цяло}$$

Всеки ред на симплекс-таблицата, който поражда такова равенство, се нарича *ред-източник (произвеждащ ред).* Тъй като коефициентите на J са цели, J трябва също да е целочислена и първият ред на таблицата може също да се избира като произвеждащ ред. Стойностите b_i и a^j_i се представят във вида

$$b_i = [b_i] + f_i, \quad a^j_i = [a^j_i] + f_{ij}$$

където с $N = [c]$ е означено най-голямото цяло число, което удовлетво-рява условието $N \leq c$. Следователно f_i, и f_{ij} удовлетворяват условията $0 < f_i < 1$ и $0 \leq f_{ij} < 1$, т.е. f_i е положителна дроб, а f_{ij} - неотрицател-на дроб. Например, ако $c = 1^1/_3$, $[c] = 1$ и $f = c - [c] = \frac{1}{3}$, ако $c = -2^2/_3\ [c] = -3$ и $f = 1/3$.

След заместване с новите означения в уравнението за x_i, се полу-чава

$$x_i = [b_i] + f_i - \sum_{j=1}^{n} \left[\left[a_i^j \right] + f_{ij} \right] z_j$$

или

$$f_i - \sum_{j=1}^{n} f_{ij} z_j = x_i - [b_i] + \sum_{j=1}^{n} [a_i^j] z_j.$$

Тъй като *всички* x_i и z_i трябва да имат цели стойкости, дясната част трябва да е целочислена. Следователно и лявата част трябва да приема цели стойности. Но $f_{ij} > 0$ и $z_i > 0$ за всички *i,j,* следователно $\sum_{j=1}^{n} f_{ij} z_j \geq 0$ и се изпълнява неравенството $f_i - \sum_{j=1}^{n} f_{ij} z_j \leq f_i$. Тъй като $f_i < 1$, то и $f_i - \sum_j f_{ij} z_j < 1$. Както беше показано, лявата част на това неравенство трябва да има цели стойности. От последното неравенство следва, че *необходимото условие* за нейната целочисленост е

$$f_i - \sum_j f_{ij} z_j \leq 0.$$

Това ограничение се записва във вида

$$S_i = \sum_j f_{ij} z_j - f_i,$$

където $S_i >= 0$ е *допълнителна променлива,* конто по определение трябва да има целочислени стойности. Последното *ограничение във вид на равенство* определя *сечението на Гомори за напълно целочислената задача.*

Тъй като небазисните променливи $z_3 = 0$, то очевидно $S_i = f_i$, т.е. тази компонента на решението не е допустима. Това означава, че полученото по-рано решение на отслабената задача на ЛП не удовлет-ворява новото ограничение. В този случай бихме могли да използува-ме дуалния симплекс-метод, с помощта на който да се отсече част от областта, която не съдържа точки с целочислени координати.

Симплекс-таблицата на отслабената задача се преобразува чрез добавяне на нов ред и нов стълб, които съответстват на полученото уравнение на сечението на Гомори и се получава решението в табл.7.3.

Таблица 7.3.

БПР	x_1	$\cdots$	x_i	$\cdots$	x_m	z_1	$\cdots$	z_j	$\cdots$	z_n	S_i	Решение
J	0	$\cdots$	0	$\cdots$	0	c_1	$\cdots$	c_j	$\cdots$	c_n	0	b_0
x_1	1	$\cdots$	0	$\cdots$	0	a_1^1	$\cdots$	a_1^j	$\cdots$	a_1^n	0	b_1
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
x_i	0	$\cdots$	1	$\cdots$	0	a_i^1	$\cdots$	a_i^j	$\cdots$	a_i^n	0	b_i
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
x_m	0	$\cdots$	0	$\cdots$	1	a_m^1	$\cdots$	a_m^j	$\cdots$	a_m^n	0	b_m
S_i	0	$\cdots$	0	$\cdots$	0	$-f_{i1}$	$\cdots$	$-f_{ij}$	$\cdots$	$-f_{in}$	1	$-f_i$

Ако решението, получено по дуалния симплекс-метод, е целочи-слено, процесът на решаване приключва. В обратния случай се въве-жда ново сечение на Гомори въз основата на симплекс-таблицата на оптималното решение, определено по дуалния симплекс-метод и отно-во се прилага този метод. Тази процедура се повтаря до получаването на целочислено решение. Ако на някоя итерация се разкрие отсъствие на допустимо решение (при прилагането на дуалния симплекс-метод), Началната задача на ЦП няма допустимо целочислено решение.

Алгоритъмът се нарича *дробен,* защото всички ненулеви коефици-енти на сечението на Гомори са по-малки от единица.

Общият брой на ограниченията на породената задача не е по-го-лям от *(rn + n).* Ако породената задача има повече от m+n ограничения, една или няколко от допълнителните променливи *Si* стават базисни, съответните уравнения стават излишни и се изключват от таблицата.

Основните недостатъци на този алгоритъм са възможността за определяне на неоптимално целочислено решение вследствие на грешки при закръгленията и отсъствието на допустимо решение до последна-та итерация на процеса на решение. Има алгоритми, конто избягват тези недостатъци, но в изчислително отношение те не са достатъчно ефективни.

Пример 7.2. За разглежданата по-горе задача на ЦП, чието реше-ние е показано на фиг.7.4., симплекс-таблицата на оптималното решение на отслабената задача е

Таблица 7 4

БПР	x_1	x_2	x_3	x_4	Решение
J	0	0	72/25	34/25	$91^{1/5}$
x_2	0	1	8/25 -	1/25	$4^{4/5}$
x_1	1	0	1/25	3/25	$5^{2/5}$

Като произвеждащ може да се избере кой да е от редовете. Оби-кновено се избира редът, на който съответствува $\max\{f_i\}$. В случая това е редът на x_2 ($4/5 > 2/5$). За този ред може да се запише

$$x_2 + \frac{8}{25}x_3 + \frac{1}{25}x_4 = 4^{4/5}$$

или

$$x_2 + \left(0 + \frac{8}{25}\right)x_3 + \left(0 + \frac{1}{25}\right)x_4 = 4 + 4/5,$$

откъдето уравнението на сечението на Гомори е

$$S_1 - \frac{8}{25}x_3 - \frac{1}{25}x_4 = -\frac{4}{5}.$$

Новата симлекс-таблица е

Таблица 7.5.

БПР	|			x_4	S_1	Решение
J	0	0	72/25	34/25	0	$91^{1/5}$
x_2	0	1	8/25	1/25	0	$4^{4/5}$
x_1	1	0	- 1/25	3/25	0	$5^{2/5}$
<-S_1	0	0	- 8/25	- 1/25	1	-4/5

Тъй като S_1 има недопустима стойност (< 0), прилага се дуzл-ният симплекс-метод. В съответствие с правилата за определяне на изключваната и включваната променлива, изключва се променлива S_1 и се включва променливата x_3. Преобразуваната таблица, чиито еле-менти са получени по съответните правила за дуалния симплекс-метод. е

Таблица 7.6.

БПР	X_i	X_2	x_3	x_4	S_1	Решение
J	0	0	0	1	9	84
x_2	0	1	0	0	1	4
x_1	1	0	0	1/8	- 1/8	$5^{1/2}$
x_3	0	0	1	1/8	-25/8	$2^{1/2}$

Вижда се, че новата стойност на x_2 е целочислена, но решение-то за x_1 е нецелочислено, затова въвеждането на сечения продължава. Уравнението на x_1 се представя във вида

$$x_1 + \left(0 + \frac{1}{8}\right)x_4 + \left(-1 + \frac{7}{8}\right)S_1 = 5 + \frac{1}{2}$$

и новото пораждащо сечение се описва с уравнението

$$S_2 - \frac{1}{8}x_4 - \frac{7}{8}S_1 = -\frac{1}{2}.$$

Това сечение се добавя към последната таблица и се получава табл.7.7.

Таблица 7.7

БПР	x_1	x_2	x_3	|	S_1	S_2	Решение
J	0	0	0	1	9	0	84
x_2	0	1	0	0	1	0	4
x_1	1	0	0	1/8	-1/8 -	0	$5^{1/2}$
x_3	0	0	1	1/8	25/8	0	$2^{1/2}$
<-S_2	0	0	0	-1/8	7 /8	1	- 1/2

След прилагането на дуалния симплекс-метод се получава таблица 7.8.

Оптималното решение на началната задача е $x^*_1 = 5$, $x^*_2 = 4$ и $J^* = 80$. Вижда се, че стойността на J намалява с последователното въвеждане на нови ограничения.

Да проверим как сеченията отсичат части от допустимото про-странство на отслабената задача. От последната таблица се вижда, че $x_2 + S_1 = 4$, откъдето $x_2 <= 4$ е едното ново допълнително ограничение. От същата таблица може да се запише

$$x_1 - S_1 + S_2 = 5.$$

Таблица 7.8

БПР	x_1	x_2	x_3 0	x_4	S_1	S_2	Решение
J	0	0		0	2	8	**80**
x_2	0	1	0	0	1	0	4
x_1	1	0	0	0	- 1	1	5
x_3	0	0	1	0	-4	1	2
x_4	0	0	0	1	7	-8	4

След заместване на S_i от уравнението $x_2 + S_1 = 4$ се получава

$$x_1 + x_2 + S_2 = 9, \text{ т.е. } x_1 + x_2 \leq 9.$$

Правите, конто съответствуват на двете нови ограничения, са показани на фиг.7.4.

В примера като произвеждащ се избираше редът, на който съответствува $\max_i\{f_i\}$. Често за такъв се избира редът, за който се изпълнява $\max_i \left\{ f_i / \sum_j f_{ij} \right\}$. Второто правило е по-ефективно, тъй като в повечето случаи по-добре представя "силата" на дробното сечение, измервана с големината на отрязваната област от пространството на решенията.

7.4.2. АЛГОРИТЪМ ЗА РЕШАВАНЕ НА ЧАСТИЧНО ЦЕЛОЧИСЛЕНИ ЗАДАЧИ

Нека $x_к$ е целочислена променлива в задача на СЦП. Съответни-ят й ред в симплекс-таблицата на оптималното решение на отслабената задача поражда съотношението

$$x_k = b_k - \sum_{j=1}^{n} a_k^j z_j = [b_k] + f_k - \sum_{j=1}^{n} a_k^j z_j$$

или

$$x_k - [b_k] = f_k - \sum_{j=1}^{n} a_k^j z_j.$$

Тъй като някои от променливите z_i не са целочислени, трябва да се построи процедура на отсичане по-различна от използваната при напъл-но целочислени задачи. Това може да бъде направено като се използва фактът, че за целочислената променлива $x_к$ трябва да се изпълнява едно от двете условия

$$x_k \leq [b_k] \quad \text{или} \quad x_k \geq [b_k] + 1$$

След заместване за $x_к$ от съотношението по-горе тези условия приемат вида

Ако с N^+ и N^- се означат множествата на индексите j, за конто съот-ветно $a^j_к \geq 0$ и $a^j_к < 0$, от последните две неравенства се получава

$$f_k \leq \sum_{j \in N^+} a_k^j z_j \quad \text{и} \quad f_k \leq \frac{f_k}{f_k - 1} \sum_{j \in N^-} a_k^j z_j.$$

Тъй като тези две условия не могат да се изпълняват едновременно, по-долу се показва, че те може да се обединят в едно ограничение от вида

където $S_k \geq 0$ е допълнителна променлива. Това уравнение определи търсеното *сечение на Гомори за частично целочислената задача* и е *необходимото условие за целочисленост на $x_к$.*

В оптималната симплекс-таблица всички небазисни променливи $z_i = 0$, следователно стойността на S_k не е допустима. По-нататък изчисленията трябва да продължат с използването на дуалния симплекс-метод.

Възможността за обединяване на двете условия по-горе в едно ограничение се показва по следния начин. Условията се записват в опростен ...то $a = \sum_{j \in N^+} a_k^j z_j$, $b = f_k \left(\sum_{j \in N^-} a_k^j z_j \right) / (f_k - 1)$. Очевидно $a \geq 0$ и $6 \geq 0$. Тъй като двете условия не се изпълняват едновременно, ако се изпълнява първото от тях, може да се запише $f_k \leq a$ и $f_k \geq b$. Оттук $a + b \geq f_k$ и след умножаване На двете страни по —1 и въвеждане на допълнителната променлива S_k, Може да се запише

$$S_k - \{a + b) = -f_k.$$

Пример 7.3. Разглежда се примерът 7.2, като се изисква само про-Менливата x_1 да е целочислена. От таблицата с оптималното решение На отслабената задача

$$x_1 - \frac{1}{25} x_3 + \frac{3}{25} x_4 = 5 + \frac{2}{5},$$

откъдето

$$N^- = \{3\}, \quad J^+ = \{4\}, \quad f_1 = \frac{2}{5}.$$

Таблица 2.9

Ы1Р	x_1	x_2	x_3	x_4	S_1	Решение
J	0	0 1	72/25	34/25	0	$91^{1/5}$
x_2	0	0	8/25	1/25	0	$4^{4/5}$
x_1	1	0	-1/25	3/25	0	$5^{2/5}$
$\leftarrow S_2$	0		-2/75	-3/25	1	-2/5

Уравнението на сечението на Гомори е

$$S_1 - \left\{ S_1 - \frac{2}{75}x_3 - \frac{3}{25}x_4 = -\frac{2}{5} \right\} = -\frac{2}{5}$$

или

Това ограничение се включва в същата таблица и се получава табл.7.9. След прилагането на дуалния симплекс-метод се определя реше-нието в табл.7.10.

Вижда се, че оптималната стойност $J^* = 86^2/^3$ се получава при $x^*_1 = 5$ и $x^*_2 = 4^2/^3$.

БПР	x_1	x_2	x_3	x_4	S_1	I аолица /.10.
J	0	0	116/45	0	34/3	$86^2/^3$
x_2	0	1	602/1875	0	1/3	$4^2/^3$
x_1 x_4	1 0	0	-1/15 2/9	0 1	1	5 $3^{1/3}$
		0			-25/3	

Счита се,че методите на отсичането не са подходящи за решаване-то на задачи с по-голяма размерност. Те са ефективни при решаването на някои задачи със специална структура. Напоследък, в съчетание с елементи на разклоняване, идеите за отсичащи равнини се използуват при разработването на нови алгоритми на ЦП.

7.5. ДВОИЧНИ ЦЕЛОЧИСЛЕНИ ЗАДАЧИ

Целочислената променлива x с горна граница n, $0 \leq x \leq n$ може да бъде изразена чрез n двоични променливи $y_1, y_2, \ldots, y_n$ по следния начин

$$x = y_1 + y_2 + \ldots + y_n$$

Друго по-икономично представяне, при което броят на двоичните про-менливи обикновено е по-малък от n, е следното

$$x = y_0 + 2y_1 + 2^2 y_2 + \ldots + 2^r y_r,$$

където r е най-малкото цяло число, което удовлетворява $2^{r+1} - 1 \geq n$. По такъв начин всяка целочислена задача може да бъде представена като задача на двоичното ЦП. За решаването на такива задачи може да се използува разгледаният метод на клоните и границите, като се направят следните изменения:

— в отслабените задачи на ЛП за двоичните променливи X_i се въвеждат ограниченията $0 \leq x_i \leq 1$;

— условията на разклоняване $x_i \leq [x^*_i]$ и $x_i \geq [x^*_i] + 1$ отпадат и се заменят с $x_i = 1$ и $x_i = 0$. С други думи, двете нови задачи при разклоняването възникват като в целевата функция и ограниченията на съответната отслабена задача x_i приема фиксирани стойности, съ-ответно 1 или 0.

Основен недостатък на този алгоритъм е решаването във всеки възел на задача на ЛП. Това е избегнато в метода, предложен от Е. Балаш за решаване на двоични целочислени задачи, известен като *ади-тивен алгоритъм.* Този метод може да се разглежда като модификация на по-общия метод на клоните и границите. Простотата на пресмята-нията с двоични променливи е използувана при него по такъв начин, че единствените необходими операции са събиране и изваждане. Това е определило и наименованието - адитивен алгоритъм.

7.5.1. АДИТИВЕН АЛГОРИТЪМ

Разглежда се следната задача

Да се минимизира

$$J = \sum_{j=1}^{n} c_j x_j$$

при ограниченията

$$\sum_{j=1}^{n} a_{ij} x_j + S_i = b_i, \quad i = \overline{1,m}$$

$$x_j = 0 \text{ или } 1, \quad j = \overline{1,n}$$

$$S_i \geq 0, \quad i = \overline{1,m},$$

където $S_i \geq 0$ са допълнителни променливи. За използуването на алго-ритъма е необходимо началната симплекс-таблица на отслабената за-дача да определи т.нар. *дуално допустимо* решение, т.е. оптимално, но недопустимо решение. Необходимо е с-ъщо ограниченията да са от типа $(\leq)$.

Получаването на начално дуално допустимо решение на отслабената задача е възможно, ако $c_j \geq 0,\ j = \overline{1,n}$. Ако има някое $c_j < 0$, то се преобразува чрез допълването на x_i, т.е. чрез заместването жу = 1 - x'_i в целевата функция и ограниченията, където x'_i е двоична променлива Задачата, в която всички $c_j \geq 0$, се нарича *дуално реигима.*

Понякога, след преобразуването на отслабената задача в дуално решима, началната симплекс-таблица определя допустимо решение. В такива случаи това решение е и оптимално, тъй като оптимумът на J се постига, като всички променливи x_i се положат равни на нула. Адити-вният алгоритъм се използва за решаване на задачи, когато началното решение е дуално-допустимо (оптимално и недопустимо).

Основната идея се състои в изброяването (преглеждането, проби-рането) на всички 2^n възможни решения на задачата, при което само в част от тях се изследват явно, а останалите се отстраняват автоматично. В началото всички променливи x_i, имат нулева стойност. Тъй като задачата е за минимизация, решението е оптимално, но недопустимо -някои променливи S_i са отрицателни. На всяка стъпка от алгоритъма една променлива x_i се приравнява на единица, така че да се постигне допустимост на решението $S_i \geq 0$. Изборът на такива променливи се извършва в съответствие с тестове.

Елементите на алгоритъма се разглеждат най-напред с пример.

Пример 7.4. Да се максимизира

$$W = 4x'_1 + 3x'_2 - 6x'_3 - 3x'_4 + 4x'_5 \text{ при}$$

ограниченията

$$x'_1 + x'_2 + x'_3 + 3x'_4 + x'_5 \leq 5$$

$$8x'_1 + 4x'_3 - 5x'_4 + 4x'_5 \leq 9$$

$$12x'_1 - 7x'_2 + 4x'_4 - 4x'_5 \geq 3$$

$$x'_j = 0 \text{ или } 1,\ j = \overline{1,5}.$$

Най-напред задачата се формулира като задача за минимизация на $\tilde{J} = -W$

$$\min \tilde{J} = -4x'_1 + 3x'_2 - 6x'_3 - 3x'_4 + 4x'_5$$

и всички коефициенти се преобразуват в неотрицателни стойности, като се използува субституцията

$$x'_j = 1 - x_j,\ j = 1,2,5 \text{ и } x'_j = x_j,\ j = 3,4.$$

Целевата функция на преобразуваната задача е

$$J = 4x_1 + 3x_2 + 6x_3 + 3x_4 + 4x_5$$

Тя е получена след субституцията и пренебрегване на константата в дя-сната част на $\tilde{J}$, тъй като тя не влияе на избора на x_i. След записване на третото ограничение във вида $\leq$ и въвеждане на допълнителните про-менливи S_1, S_2 и S_3, преобразуваната задача се представя във формата, представена в таблица 7.11.

Таблица 7.11

x_1	x_2	x_3	x_4	x_5	S_1	S_2	S_3	Дясна част
4 -	3	6	3	4	0	0	0	J
1	- 1	1	3	- 1	1	0	0	2
-8	0	4	-5	- 4	0	1	0	-3
12	-7	0	-4	-4	0	0	1	-2

В началото всички $x_j = 0$ и стойностите на допълнителните променливи S_i^0 и на целевата функция J^0 са

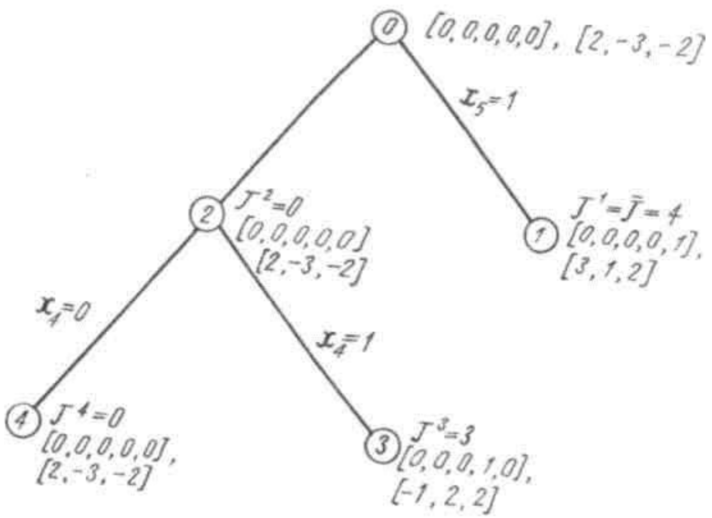
$$[S_1^0, S_2^0, S_3^0] = [2,-3,-2] \quad \text{и} \quad J^0 = 0.$$

Тъй като S_2^0 и S_3^0 са отрицателни, началното решение [0,0,0,0,0] е недо-пустимо и поне една променлива x_i трябва да приеме стойност 1, така, че решението да се измести по-близо до допустимата област. От табли-цата се вижда, че всички коефициенти при променливата x_3 в ограниченията с отрицателни стойности на S_i са неотрицателни. Полагането $x_3 = 1$ само би увеличило степента на недопустимост (например S_2 би получила стойността $S'_2 = -7$), поради което x_3 трябва да запази началната си нулева стойност. Вижда се също, че всяка от променливите x_1, x_2 и x_4 по отделно не може да осигури допустимост, но комбинация от тях при стойности 1 може да доведе до допустими S_i. Затова тези променливи не се изключват от по-нататъшно разглеждане. Ако пък се положи $x_5 = 1$, може да се постигне допустимо решение. Поради това процедурата ще осигури полагането на $x_5 = 1$, което ще доведе до решението

$$[x_1, x_2, x_3, x_4, x_5] - [0,0,0,0,1].$$

$$[S_1^1, S_2^1, S_3^1] = [S_1^0 - (-1), S_2^0 - (-4), S_3^0 - (-4)] = [3,1,2]$$

И J^1 — 4. Тъй като товае първото допустимо решение, то се запазва и определи *горна граница* J за всяко бъдещо допустимо решение, $J — J^1 = 4$. Както при метода на клоните и границите, от бъдещите допустими



Фиг. 7.5

решения интерес ще представляват само тези, на конто съответствуват по-добри (по-малки) стойности на целевата функция от *J*.

Дървото на решенията, които се получават по процедурата, разглеждана като метод на клоните и границите, е показано на фиг.7.5. До всеки възел са показани стойностите на променливите x_i и S_i, j = 1,5, i = 1,3 и на целевата функция. Началният възел (0) представя задача-та, за която всички $x_i = 0$. От този възел излизат два клона, породени от условията $x_5 = 0$ и $x_5 = 1$. Като се избере клонът $x_5 = 1$, във възелз (1) се определя допустимото решение, с $J^1 = 4$.

Тъй като всички коефициенти $c_i \geq 0$ и се решава задача за минимизация, нито един клон с $x_i = 1$, j = 1,4 не може да подобри стойност-та на целевата функция. Следователно възелът (1) е *оценен.* Остава да се анализира решението във възела (2), който съответствува на един-ствения останал клон $x_5 = 0$. Решението в този възел е

$$[x_1,x_2,x_3,x_4,x_5\} = [0,0,0,0,0], \quad [S_1^2,S_2^2,S_3^2] = [2,-3,-2], \quad J^2 = 0.$$

Разликата между решенията във възлите (0) и (2) не е в *стойностите* на променливите и на функцията *J,* а в това, че във възела (0) всяка от променливите x_i, j = 1,5 е *свободна* променлива, т.е. може да приема стоиности 0 или 1 и да генерира разклоняване, докато във възела (2) свободни са само променливите x_i, j = 1,4, а x_5 е фиксирана на нулева стойност.

Изборът на променлива на разклоняването във възела (2) се из-вършва по същия начин както във възела (0), като се използва и информацията за горната граница $\bar{J}$. Променливата x_3 се изключва от разглеждане, тъй като условието $x_3 = 1$ влошава както оптималността ($J = 6 > \bar{J}$), така и допустимостта на решението (отрицателната про-менлива S_2 намалява още повече). Променливата x_1 също може да бъде устранена, тъй като $c_1 = 4$ не води до по-добра стойност на J от $\bar{J}$. Следователно, изборът трябва да бъде направен между x_2 и x_4. Тъй като нито една от тези променливи не отстранява недопустимостта (ко-гато стойността ѝ нарастне на 1), като променлива на разклоняване ще се избере тази от тях, на която съответствува най-голямото намаление на недопустимостта. За целта за всяка *свободна* променлива x_i се дефи-нира "мяра на недопустимостта на решението", което се получава след полагането x_3 = 1. Тази мяра е емпирична и се въвежда по следния начин: $v_j = \sum \min\{0, S_i - a_{ij}\}$

Като променлива на разклоняване се избира променливата, за която абсолютната стойност на v_j е най-малка. В случая

$v_2 = \min\{0,2 - (-1)\} + \min\{0, -3 - 0\} + \min\{0, -2 - (-7)\} =$

$= 0+(-3) + 0 = -3$ $v_4 = \min\{0,2-3\} + \min\{0,-3- (-5)\} + \min\{0, -2 - (-$

$4)\} =$

$= -1+0 + 0= -1$

и като променлива на разклоняване се избира x_4. На фиг.7.5 $x_4 = 1$ е клонът, който води до възела (3), дефиниран от $x_5 = 0$ и $x_4 = 1$, където

$$[x_1,x_2,x_3,x_4,x_5] = [0.0,0,1,0], \ [S_1^3,S_2^3,S_3^3] = [-1,2,2], \quad J^3=3.$$

Свободни променливи в този възел са x_1,x_2 и x_3. Тъй като $c_1 = 4$, $c_2 = 3$ и $c_3 = 6$, полагането на x_1,x_2 или x_3 равно на единица би довело съответно до стойностите J = 3 + 4, 3 + 3и3 + 6, които са по-лоши от J. Следователно x_1,x_2 и x_3 се изключват от разглеждане и тъй като това *са всички* свободни променливи във възела (3), по-нататъшното разклоняване от този възел не е възможно и той е *оценен.*

Остава единствено възелът (4), дефиниран от $x_5 = 0$ и $x_4 = 0$. Тук

$$[S_1^4,S_2^4,S_3^4] = [2,-3,-2], \qquad J^4 = 0.$$

Свободни променливи са x_1,x_2 и x_3. Неперспективността на x_3 вече бе-Ше показана, а x_1 и x_2 не могат да направят решението в (4) допустимо. Следователно не съществуват променливи на разклоняване и възелът (4) е оценен. Всички възли са оценени и решението се определя от възела (1)

$$[x_1,x_2,x_3,x_4,x_5] = [0,0,0,0,1] \quad \text{и} \quad J = 4.$$

Решението на началната задача се получава след преминаване към началните променливи

$$x'_1 = x'_2 = 1, \quad x'_3 = x'_4 = x'_5 = 0, W = 7.$$

Лесно се проверява, че алгоритъмът преглежда всичките 2^5 решения. При оценяването на възела (1) са взети предвид всички решения, в конто $x_5 = 1$, конто са $2^{5-1} = 16$ на брой. При оценяването на възела (3), определен от $x_5 = 0$ и $x_4 = 1$, са взети предвид $2^{5-2} = 8$ решения Същият е и броят на решенията, взети предвид при оценяването на възела (4). Общо са изброени $16 + 8 + 8 = 2^5$ решения, като само 5 от тях са анализирани експлицитно. Поради това разгледаната процедура често се нарича *имплицитно изброяване* (implicit enumeration).

При формализирането на процедурата се използват понятията *свободна променлива* и *частично решение.* Една двоична променлива е *свободна* в даден възел, ако не е фиксирана чрез никои от клоните, кои-то свързват този възел с възела (0). Нейната начална нулева стойност може да бъде изменена на 1, ако това приближава решението до допу-стимата област.

Частичното решение е набор от една или повече променливи с фиксирани стойности - нула или единица. На всяка операция (или възел) *S* то се представя удобно чрез *подредено множество Rs-* Елементи на множеството *Rs* са индексите на фиксираните променливи със знак плюс или минус, който означава фиксирана стойност единица или нула на съответната променлива. Множеството *Rs* е подредено, в смисъл, че всеки нов елемент се записва *вдясно* на наличното частично решение.

Дървото на решенията може да бъде описано чрез частични решения, тъй като всеки възел се определя от фиксирани стойности на някои променливи, т.е. от съответно частично решение. Ако с празното множество $Ro = \theta$ се означи частичното решение във възела (0) (всички променливи са свободни), възлите от фиг.7.5 могат да се представят по следния начин:

Възел (0): $R_0 = \theta$
Възел (1): $R_1 = \{5\}$
Възел (2): $R_2 = \{-5\}$
Възел (3): $R_3 = \{-5,4\}$
Възел (4): $R_4 = \{-5,-4\}$

Частичните решения могат да бъдат генерирани последователно едно от друго. Общото правило за получаване на следващото частично решение от едно *оценено* решение (възел) е следното. Ако *всички* елементи на *оцененото* частично решение R_s са отрицателни, изброяването е напълно завършено. В обратния случай се избира *най-десният положителен елемент,* изменя се знакът му и се отстраняват отрицателните елементи вдясно от него. Например, ако решението $R_s = \{2,6,5,-4,3,7,-8,-9\}$ е оценено, $R_{s+1} = \{2,6,5,-4,3,-7\}$. Смисълът на правилото се изясня-ва, като се вземе предвид, че адитивният алгоритъм добавя новите променливи със стойност единица. Отрицателният елемент в такъв случай означава, че *предшествуващото* частично решение, в което този елемент е бил положителен, е било оценено. А когато всички елементи на оце-неното частично решение *Rs* са отрицателни, съответните променливи са били разгледани със стойкости 0 и 1 и не съществуват повече клони за разглеждане (т.е. изброяването е завършено).

Да предположим, че в общата двоична задача за минимизация, разгледана в началото на раздела, всички коефициенти $c_i \geq 0$ (след евентуална субституция на някои променливи). Разглежданият пример позволява да се представят в обобщен вид тестовете, използувани за оценяване на частичните решения и за избор на нови променливи, чиито стойности ще се фиксират равни на единица.

Тест 1, Нека x_S е свободна променлива. Ако $a_{iS} \geq 0$ за *всички г,* съответствуващи на $S_i^k < 0$, x_S се отстранява като неперспективна, тъй като не може да приближи решението до допустимата област.

Тест 2. Ако x_S е свободна променлива и $c_S + J^k \geq \bar{J}$, x_S се отстранява като неперспективна$_x$ тъй като не може да подобри съществуващото решение, определило *J.*

Тест 3. Нека на ограничението

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n + S_i = b_i$$

съответствува $S_i^k < 0$. Ако с M_k се означи множеството от *свободни* променливи, конто не са отстранени чрез тестовете 1 и 2, нито една от тези променливи не е перспективна, ако поне за една $S_i^k < 0$ се изпълнява условието

Това означава, че променливите от M_k съвместно, когато стойностите им станат равни на 1, не могат да направят решението допустимо и трябва да бъдат отстранени. В такъв случай частичното решение R_k е оценено.

Тест 4. Ако множеството M_k след прилагането на теста 3 не е празно, като променлива на разклоняването x_r се избира тази, на която съответствува

където

$$v_j^k = \sum_{i=1}^{m} \min\{0, S_i^k - a_{ij}\}.$$

Ако $v_r^k = 0$, стойноетта $x_r = 1$ заедно с частичното решение R_k води до *подобрено* допустимо решение. В този случай $R_{k+1} = \{R_k, r\}$ ^{e} оценено. Ако обаче $v_r^k < 0$, описаните тестове се прилагат към R_{k+1} дотогава, Докато изброяването се извърши напълно, т.е. докато *всички* елементи на оцененото частично решение станат отрицателни.

Пример 7.5. Елементите на адитивния алгоритъм ще бъдат представени с използуване на предишния пример

Възел (итерация) 0 Полагат се

$R_0 = \emptyset,\ \bar{J} = \infty$. Определят се

$$[S_1^0, S_2^0, S_3^0] = [2,-3,-2], \quad J^0 = 0.$$

Чрез теста 1 се изключва x_3. Променливите от множеството M_0 {1,2,4,5} не се отстраняват чрез теста 3, защото

i = 2: - 8 - 5 - 4 = -17 <-3 i = 3: - 7 - 4 - 4 = -15 < -2. За

прилагането на теста 4 се определят:

$v_1^0 = 0 + 0 + (-2- 12) = -14$ $v_2^0 =$

$0 + (-3 - 0) + 0 = -3$

$v_4^0 = (2-3) + 0 + 0 = -1$ v_5^0

$= 0 + 0 + 0 = 0,$

следователно r = 5.

Възел (итерация) 1

Полага се $R_1 = \{5\}$. Определят се

$$[S_1^1, S_2^1, S_3^1] = \{2-(-1), -3-(-4), -2-(-4)\} = [3,l,2], \quad J^1 = 4.$$

Решението е допустимо ($v_5^0 = 0$), следователно $\bar{J} = J^1 = 4$ и R_1 е оцене-но.

Възел (итерация) 2

Полагат се $R_2 = \{-5\}, \bar{J}. = 4$ и се определя

$$[S_1^2, S_2^2, S_3^2] = [2,-3,-2], \quad J^2 = 0.$$

Чрез теста 1 се изключва x_3. Чрез теста 2 се отстраняват x_1 и x_3. Променливите от множеството $M_2 = \{2,4\}$ не се отстраняват чрез теста 3. Чрез теста 4 се определят $v_2^2 = -3$, $v_4^2 = -1$, следователно r = 4.

Възел (итерация) 3

Полага се $R_3 = \{-5,4\}$ и се определя

$$[S_1^3, S_2^3, S_3^3] = [-1,2,2], \quad J^3 = 3.$$

Чрез теста 1 се изключва x_3. Чрез теста 2 се отстраняват x_1, x_2 и x_3, *Т*ъй като $M_3 = \theta$, R_3 е оценено.

Възел (итерация) 4

Полага се $R_4 = \{-5,—4\}$. Определя се

$$[S_1^2, S_2^4, S_3^4] = [2,-3,-2], \quad J^4 = 0.$$

Чрез теста 1 се изключва x_3. Чрез теста 2 се отстраняват x_1 и x_3. Променливата от множеството $M_4 = \{2\}$ се отстранява чрез теста 3. Следователно R_4 е оценено. Тъй като всички елементи на R_4 са отрицателни, изброяването е пълно. Оптималното решение е R_i, при което $J^1 = 4$.

7.6. ВЪЗМОЖНОСТИ ЗА ПОВИШАВАНЕ НА ЕФЕКТИВНОСТТА НА АЛГОРИТМИТЕ

Най-съществени резултати са постигнати при разработването на алгоритми за решаване на двоични задачи на ЦП. Предложен е алгоритмичен подход, чрез който са решени двоични задачи на ЦП - чисто и смесено, в които броят на променливите е няколко хиляди. Решените задачи имат обаче *разредени* матрици *А,* т.е. броят на ненулевите кое-фициенти във функционалните ограничения е твърде малък - около 5 %. В общия случай се счита, че все още само задачи с няколко десетки двоични променливи (до 100) могат да бъдат решавани за приемливо време.

В съвременните алгоритмични реализации се използува комбинация от три вида методи: 1*) усъвършенствувани алгоритми на клоните и границите; 2) генериране на отсичащи равнини* и *3) автоматична предварителна обработка на задачата.* Последната се основава на компютъ-рен анализ на началната формулировка, в резултат на който задачата се преформулира във вид, който се решава по-бързо, без да се отстраняват допустими решения. Някои от използуваните идеи са твърде прости. Например някои променливи се отстраняват от модела, като се фик-сират на 0 или 1, което пък е възможно поради наличието на някои специфични ограничения, например

$$5x_1 + 3x_2 < 3, \implies x_1 = 0$$

$$5x_3 - 3x_4 \leq 2 \implies x_3 = 0 \text{ и } x_4 = 1.$$

Някои ограничения се отстраняват от модела, тъй като са излишни при двоични променливи, например

$$5x_1 + 3x_2 \leq 9.$$

Двоичните променливи позволяват понякога *стесняването* на функционално ограничение чрез намаляване на някои от коефициентите му, например

$$6x_1 + 8x_2 + x_3 \geq 3 \implies 3x_1 + 3x_2 + x_3 \geq 3.$$

По този начин се стеснява отслабената задача на ЛП, т.е. отстраняват се някои нейни допустими решения, без да се елиминират допустими целочислени решения.

Когато има група от променливи, които представят взаимно из-ключващи се алтернативи, често се използува "тригер", който задава стойност на една или повече променливи, след като на друга промени ли ва от групата е присвоена определена стойност от алгоритъма.

Изложените идеи се използуват при създаването на *евристични алгоритмы* за решаване на задачи с голяма размерност. Тези алгори-тми определят бързо добри допустими решения, които не винаги са оптимални, но обикновено са по-добри от тези, които биха се получили чрез закръгляне на непрекъснатото решение на отслабената задача до допустимо целочислено решение.

След бурно развитие през последните три десетилетия, създаването на алгоритми за решаване на целочислени задачи продължава да бъде област на активни изследвания.

7.7. ЗАКЛЮЧЕНИЕ

Много задачи за управление, проектиране и избор на решение мо-гат да бъдат формулирани като задачи на ЦП. Те са по-трудни от зада-чите, в които отсъствуват изисквания за целочисленост на промени-вите, вследствие на което достъпните алгоритми за решаването им са значително по-малко ефективни от симплекс-метода. Най-важен фактор за времето на изчисление при ЦП е броят на променливите и струк-турата на задачата. Двоичните ЦЛЗ се решават по-лесно от задачите с общи целочислени променливи. Добавянето на нови непрекъснати променливи (СЦП) не влияе съществено върху времето на решение. За някои задачи със специална структура са разработени специализирани алгоритми, които решават големи задачи, с няколко хиляди двоични променливи. В общия случай обаче решението може да бъде определено ако броят на променливите е много по-малък.

Съществуващите програми за задачи на ЦП се основават на метода на клоните и границите. Предполага се участие на човека в процеса на решение, при което последователно се уточнява начинът на решение на задачата чрез избиране между няколко възможни начина на решение. В новия алгоритмичен подход се комбинира използуването на метода на клоните и границите, генерирането на отсичащи равнини и автоматичната предварителна обработка на задачата. Разработват се евристични алгоритми за задачи с голяма размерност.

Създадените алгоритми и програмни продукти все още не съот-ветствуват на значимостта на задачите на ЦП, които възникват в упра-влението и проектирането. Може да се очаква, че изследванията в тази облает и особено тези, свързани с приложенията на паралелни компю-търни системи, ще доведат до нови мощни алгоритми, с които решава-нето на задачи с голям брой променливи ще стане възможно.