LENGYEL IMRE—KUBA ATTILA

Síküveg szabásának optimalizálása (I.)

1. Bevezetés

Tanulmányunkban húzott síküveg szabásának optimalizálásával, az Üvegipari Művek Orosházi Üveggyárában felmerült probléma matematikai modellezésével és számítógépes megoldásával foglalkozunk.

Hasonló probléma több vállalatnál (pl. fémipari, textilipari, bőripari, stb. vállalatnál) előfordul, mivel sok helyen van szükség eltérő méretű idomoknak valamilyen alapanyagból való kiszabására és a darabolási műveletek elvégzése közben jelentős mennyiségű hulladék keletkezik. Bizonyos termékek gyártásánál az önköltség csökkentése mindig is kiemelt feladat volt, az utóbbi években pedig egyre sürgetőbbé vált, ennek egyik legjelentősebb módja lehet pl. az anyagköltségek csökkentése.

Ebben a tanulmányban olyan számítógépes eljárást ismertetünk, amellyel a síküveg-szabásnál keletkező felületveszteséget minimalizálni lehet. Természetesen a darabolási folyamat optimalizálásának ez csak egy bizonyos szempontból való megközelítése, más szempontok (szabáshoz szükséges idő, raktározás, anyagmozgatás, stb.) esetleg merőben más jellegű feladatok megoldását igénylik. A modellezés során a gyakorlati szempontokat részesítettük előnyben, egyszerű matematikai eszközöket felhasználó eljárás kidolgozására törekedtünk.

A síküveg hagyományosnak tekinthető, eddig alkalmazott szabásánál gyári szakemberek becslése szerint kb. 33—35% hulladék keletkezik, ezzel szemben a számítógép által adott szabástervek vesztesége 20—22%. Vagyis a számítógépes eljárás a hagyományos szabási módszerhez képest kb. 13%-kal csökkentheti a hulladékot, vagy másképp megfogalmazva, ugyanannyi alapanyag feldarabolása esetén kb. 20%-kal nő a hasznos, eladható üvegfelület. A módszer szorosan illeszkedik a jelenlegi technológiai folyamathoz, műszaki változtatások, beruházások nélkül alkalmazható bizonyos átszervezések végrehajtása után. Tanulmányunkban a feladat modellezésének folyamatát mutatjuk be, az optimalizálást végző végleges programrendszert és alkalmazásának tapasztalatait külön cikk [14] ismerteti.

1.1. *A szabási probléma leírása*

A gyárban *alaptáblának* nevezett téglalap alakú síküveget, félkészterméket gyártanak és raktároznak. A megrendelők (házgyárak, építőipari szövetkeztek, stb.) meghatározott méretű, minőségű és tételszámú téglalap alakú síküveget kérnek, ezek a *rendelés*ek, amit a szabász-üzem a raktáron levő félkésztermékekből szab ki. A szabásnál fellépő felületveszteség minimalizá-

lására kellett megoldást keresni, ahol a felületveszteség = (összes felhasznált alaptábla felület) – (leszabott rendelések felülete).

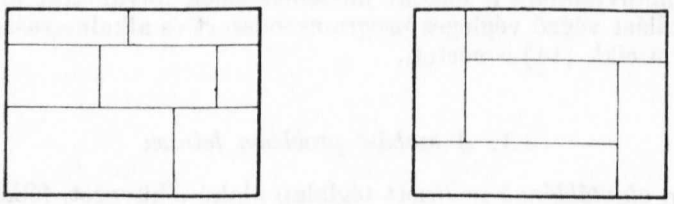
A rendelések egy adott időszakra vonatkozó (pl. hónap, dekád, stb.) összességét *rendelésállománynak* hívjuk. *Szabásképnek* (szabásrajznak, leszabási változatnak, stb.) nevezzük azt az elrendezést, ami szerint a rendeléseket az alaptáblákból leszabják. A szabásképek egy olyan sorozata, amelyben a rendelésállomány minden rendelése legalább az előírt tételszámban kiszabható: a rendelésállomány *szabásterve*. Egy rendelésállományhoz több szabásterv tartozik, ezek közül kell az optimálisat kiválasztani.

A rendelések sürgősségi jellegük folytán két csoportra oszthatók: a *napi sürgős és az előre ismert, hosszabb határidejű* rendelésekre. A napi sürgős rendelések kis tételszámúak és szinte azonnal hozzá kell kezdeni a leszabáshoz, itt az optimalizálás nagy nehézségek leküzdése árán is csak viszonylag gyengébb eredményekkel kecsegtet. Az üveggyár jelenleg nem rendelkezik számítógéppel, így az optimális szabásterv elkészítése és a tényleges lszabás csak különböző helyeken (pl. számítóközpontban és a szabász-üzemben) és időben egymáshoz képest esetleg több napos eltéréssel végezhető el, amiből következik, hogy csak a hosszabb határidejű rendelések darabolásának optimalizálása jöhet szóba. Ezek a rendelések az összes mennyiségnek a nagyobb hányadát alkotják és általában elég nagy a tételszámuk (legalább 50).

A gyárban jelenleg kétféle hagyományos módon darabolják fel az alaptáblákat. *Kézi szabásnál* a vágófejeket és a téglalapok elhelyezkedését a szabással foglalkozó munkás kézzel állítja be, így a felületveszteség nagyon függ a szakértelmétől és a tapasztalatától. A másik hagyományos szabás a *sorozatvágás*, ahol egy automata vágógép gyorsan és pontosan darabolja, pneumatikusan mozgatja az üvegtáblát, de itt nincs lehetőség a hulladék lényeges csökkentésére, mivel *egy alaptáblából csak egyféle téglalapokat* tud kivágni. A szabás megkezdése előtt kiszámítják az adott rendeléshez legmegfelelőbb alaptáblát és a raktárból bekészítik. Ezeknél a hagyományos szabási módszereknél a felületveszteség kb. 33–35%. A sorozatvágás nagy vesztesége elsősorban abból származik, hogy nem teszi lehetővé különböző méretű téglalapok egyidejű leszabását.

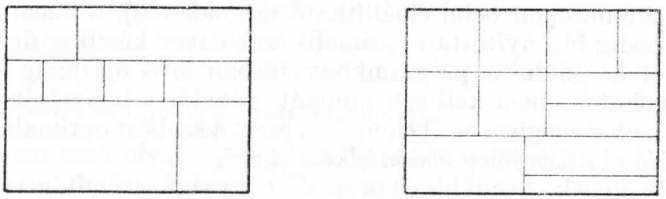
A síküveg szabásánál alkalmazott szabászgépek döntő többsége széltől-szélig karcolja az üvegtáblát (ezt *guillotine-vágásnak* is nevezik), azaz nincs lehetőség a vágófejek felemelésére (1. ábra).

A téglalapok illesztésére, ütemezésére, a felületveszteség minimalizálására a *két-asztalos*, nemrégiben elkészített szabászgépen van lehetőség. Itt először



1. ábra

Példa guillotine- (a) és nem-guillotine vágásra (b)



2. ábra

Két-ütemű (a) és általános két-ütemű (b) szabásképek

az alaptáblákból csíkokat vágnak, majd a csíkokat egyenként darabolják fel a kívánt méretre — ezt *két-ütemű guillotine vágásnak* is nevezik (lásd a 2a ábrát). Mi erre a szabási módszerre készítettük el az optimalizációs eljárást.

Ez a szabási probléma a két-dimenziós szabási, lefedési feladatok egyik speciális esete, amikor téglalapból kell téglalapokat kiszabni. Az ilyen jellegű szabási feladatokkal foglalkozott P. C. Gilmore és R. E. Gomory alapvető cikksorozatában [1—4]. A szabási problémát lineáris programozási feladatként fogalmazták meg, amelyet módosított szimplex-algoritmussal oldottak meg úgy, hogy a bázisba bevonandó vektorokat kisegítő feladatok (hátizsák-problémák) segítségével határozták meg. Ezáltal a nagyméretű lineáris programozási feladat kisebb méretű, könnyebben kezelhető feladatokra vezették vissza. Az általuk kifejlesztett eljárást alkalmazta az üvegiparban R. G. Dyson és A. S. Gregory [5], valamint Oli B. G. Madsen [6—7] általános két-ütemű guillotine-vágásra (lásd a 2b ábrát). A két-dimenziós feladat általánosabb (több-ütemű guillotine-vágás) megoldására adott algoritmust J. C. Herz [8], N. Christofides és C. Whitlock [9], valamint A. Adamowicz és A. Albano [10]. Az általunk leírtakhoz elvében hasonlít az a módszer, amely Lampl Tamás [11] dolgozatában szerepel. Az Orosházi Üveggyár által kitűzött feladat részletesebb leírása pedig [12]-ben található meg.

Megoldásként a legegyszerűbbnek tűnő eljárást választottuk: az összes szóba jöhető szabásképet elkészítjük és egészértékű lineáris programozással (ILP) választjuk ki közülük az optimális szabástervet alkotó szabásképeket. Az összes szabáskép elkészítése és a nagyméretű ILP feladat kezelése okozza az igazi nehézségeket. Az összes szabásképet mindig a konkrét rendelésállomány ismeretében egy gyors kombinatorikus algoritmussal készítjük el, az ILP feladatot pedig programkönyvtárban levő programcsomag segítségével oldjuk meg.

Miért döntöttünk úgy, hogy ezzel a módszerrel keressük meg az optimális szabástervet? Az alábbi szempontokat vettük figyelembe:

— az üvegszabás feltételei annyira speciálisak (a két-ütemű guillotine-vágás) — a két-dimenziós feladat lényegében két egy-dimenziós feladatra redukálódott —, hogy a feltételeket jól kihasználó, egyszerű kombinatorikus eljárás készíthető a szabásképek előállítására,
— a szabásképek száma bizonyos redukciók és a félkésztermékek méreteinek kicsiny száma miatt nem nagy (1000—1500-nál nem több),
— a gyártól a számítógép térben és időben messze van, ezért van idő az optimális szabástervek elkészítésére (nem kell megelégedni egy közelítő,

ámde pillanatokon belül előállítható megoldással), a lineáris programozással pedig bizonyítottan optimális szabásterv készíthető.
— fel lehet használni a programkönyvtárban levő optimalizáló programcsomagokat — nem kell sok munkát igénylő és bonyolult optimalizáló programokat megírni és „belőni" —, ezért a szabást optimalizáló rendszer *rövid idő* alatt és főleg *olcsón* elkészíthető,
— bármely másik, legalább „közepes" nagyságú számítógépre az eljárás könnyedén „áttelepíthető", mivel LP feladatra általában minden programkönyvtárban van megoldás.

A hosszabb határidejű rendelések optimalizálásánál lehetőség van az optimális szabástervhez szükséges félkésztermékek gyártására, azaz a termelésirányításra is (tetszőlegesen nagy fiktív raktárkészletet feltételezve). Ugyanis előírhatjuk azt, hogy olyan méretű, mennyiségű és minőségű alaptáblát gyártsanak, amelyre majd szükség lesz. Azonban a termelés-visszacsatolásnak ezt a direkt lehetőségét (és vele együtt a félkésztermék-raktár nélküli szabást) megakadályozza az a gyártási sajátosság, hogy az alaptáblák minősége csak a húzás után dől el. Más szóval az optimális tervben előírt minőségű alaptáblák helyett eltérő minőségű alaptáblákat is kapunk. Amiből mostmár az következik, hogy *szükség van az elkészült félkésztermékek raktározására és a síküveg-szabás optimalizálásánál ennek a raktárkészletnek a figyelembevételére is.* Továbbá a kézi, ill. a sorozatvágáshoz is ebből a raktárból veszik ki az alaptáblákat.

2. Üzemi, műszaki feltételek

A következőkben a két-ütemű üvegszabás speciális gyári feltételeit ismertetjük.

A rendelések paraméterei a következők: vastagság, minőség, szélesség, hosszúság és tételszám. A félkésztermékek paraméterei ugyanezek. Természetesen egy rendelést csak az ugyanolyan vastagságú és minőségű alaptáblából lehet kiszabni. Azaz egy alaptáblából csak azok a rendelések elégíthetők ki, amelyeknek a vastagsága és a minősége az alaptábláéval megegyező. Tehát a minőség és a vastagság a rendelések halmazát olyan diszjunkt részhalmazokra bontja, amelyeket egymástól függetlenül kell szabni. Következésképpen ezekre a részhalmazokra külön-külön elkészített optimális szabástervek együtt alkotják az egész rendelésállomány optimális szabástervét.

A továbbiakban egy csoporton belül vizsgáljuk a problémát. Legyen a csoport rendeléseinek halmaza: R,

$$R = \{r_i \mid r_i(x_i, y_i, t_i),\ i = 1, 2, \ldots, n\},$$

ahol: n a csoport rendeléseinek a száma,
x a szélessége,
y a hosszúsága,
t a tételszáma a rendelésnek.

A megfelelő félkésztermékek halmaza: F,

$$F = \{F_j \mid F_j(A_j, H_j, D_j),\ j = 1, 2, \ldots, p\}$$

ahol: p az alaptábla-féleségek száma,

A a szélességét,
H a hosszúságát,
D a darabszámát jelöli a félkésztermékeknek.

Természetesen csak olyan x_i és y_i rendelések fogadhatók el, amelyekre van olyan j, hogy

$$\max \{x_i, y_i\} \leq \max \{A_j, H_j\}$$

és

$$\min \{x_i, y_i\} \leq \min \{A_j, H_j\},$$

azaz a rendelés „ráfér" valamelyik alaptáblára.

Mivel adott vastagságnál az alaptáblák szélessége állandó, ezért

$$A = A_j \ (j = 1, 2, \ldots, p).$$

Az optimalizálást a már említett két-ütemű vágásra készítettük el, amelynek a modellezés szempontjából fontos jellemzői az alábbiak:

(i) az alaptábla egyik oldalával párhuzamosan csíkokat vágunk (első-ütem), majd a csíkokat egyenként daraboljuk fel a kívánt méretre (második-ütem) (lásd a 3. ábrát);

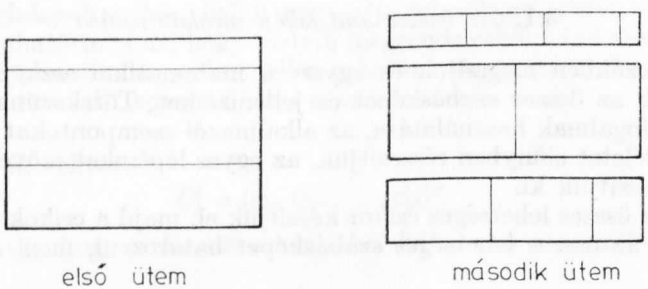
(ii) a csíkok száma legfeljebb S_1, egy csíkon belül a téglalapok száma pedig legfeljebb S_2 lehet (erre azért van szükség, mivel a vágófejek száma miatt egy asztalon legfeljebb ennyi vágást lehet egyszerre elvégezni);

(iii) az eltérő méretű téglalapok száma egy szabásképben legfeljebb T (erre a feltételre azért volt szükség, mert az üzemben a vágóasztalok mellett csak meghatározott számú megrendelés tárolható és csomagolható egyszerre).

A szükséges fogalmak és vágási feltételek ismeretében megfogalmazhatjuk a síküveg-szabás kapcsán felmerülő optimalizálási probléma megoldásra váró feladatát:

Keresendő olyan eljárás, amely a konkrét üzemi feltételeket figyelembe véve, az előre megadott rendelésekre — lehetőleg a félkésztermékek raktárkészletét is felhasználva — olyan szabástervet szolgáltat, amelynek maradéktalan teljesítése esetén a felületveszteség minimális lesz.

A Gilmore—Gomory-eljárás a (ii) feltételt, a vágófejek számának korlátozását figyelembe tudja venni, amint az [2]-ben már szerepel. A (iii) feltételt



3. ábra

Az alaptábla két-ütemű feldarabolása

is be lehet építeni az eljárásba — az általunk ismert publikációkban ilyen feltétellel nem találkoztunk —, természetesen ez a két feltétel bonyolultabbá, nehezebben kezelhetővé teszi az algoritmust, mint az általános esetben.

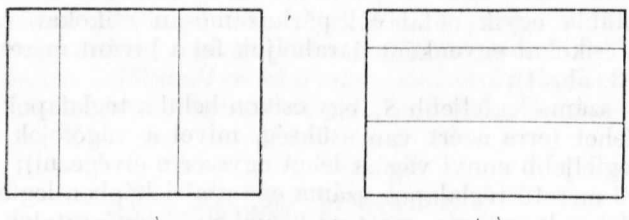
3. A leszabás matematikai modellezése

A feladatot — mint azt már említettük — két lépésben oldjuk meg:

1. fázis: az összes szóba jöhető, a feltételeknek eleget tevő szabásképek elkészítése,
2. fázis: a szabásképekből az optimális szabásterv meghatározása.

A szabás első lépésében a csíkok levágása az alaptábla helyzetétől függően kétféle módon történhet:

(i) az alaptábla szélességével párhuzamosan vágunk (lásd a 4a ábrát),
(ii) az alaptábla hosszával párhuzamosan vágunk (4b ábra).



4. ábra

A két lehetséges csík-vágás

A továbbiakban az 1. fázisban mindkét vágási módot vizsgáljuk. Természetesen egy szabástervben csak az egyiket alkalmazhatjuk, attól függően, hogy a leszabás műszaki körülményei melyiket teszik lehetővé. Jelenleg az (i) módon lehet szabni, azonban a műszaki feltételek változtatására lehetőség van. Az összes szabásképből az optimális szabástervhez tartozókat a 2. fázisban egészértékű lineáris programozással határozzuk meg.

3.1. *Az összes szabáskép meghatározása*

A következőkben megadjuk és egyszerű matematikai eszközöket felhasználva leírjuk az összes szabásképet és jellemzőiket. Törekszünk az egyszerű jelölések és fogalmak használatára, az alkalmazói szempontokat és az algoritmikus szemléletet előnyben részesítjük, az egyes lépéseket szöveges magyarázatokkal egészítjük ki.

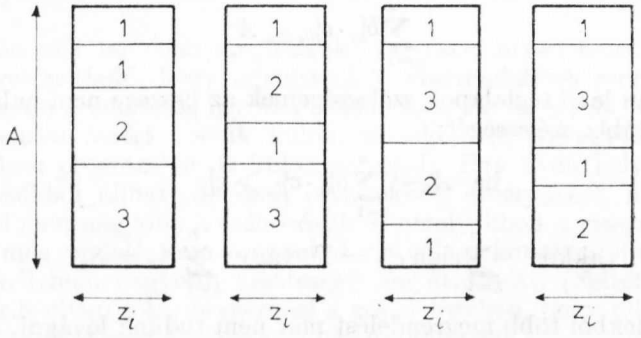
Először az összes lehetséges csíkot készítjük el, majd a csíkok egymás mellé illesztésével az összes lehetséges szabásképet határozzuk meg.

3.1.1. *Az alaptábla szélességével párhuzamos csík-vágás*

A csíkok szélessége az $x_1, y_1, x_2, y_2, \ldots, x_n, y_n$ méretek közül kerül ki, hiszen egy téglalap legfeljebb kétféle helyzetben, az x és az y méretének megfelelő szélességű csíkban szabható le (a 4a ábrán látható a csík-vágás). Az előbb felsorolt méretek között lehet megegyező is, ezért $z_1, z_2, \ldots, z_m$ ($m \leq 2n$, $z_i < z_j$, ha $i < j$) jelölje a csík-szélességként szóba jöhető szélesség- és hosszméretek olyan együttes felsorolását, ahol már minden szám különböző.

A z_i-hez ($i = 1, 2, \ldots, m$) hozzárendelünk több méretet, mégpedig azoknak a téglalapoknak a másik méretét, amelyekben a z_i szélességként vagy hosszúságként szerepel. Azaz z_i-hez hozzárendeljük Z^i-t:

$$Z^i = \{z_i^1, z_i^2, \ldots, z_i^N\} \quad (1 \leq N \leq n,\ z_i^k < z_i^j, \text{ ha } k < j).$$



5. ábra

A csíkok azonosak a leszabás optimalizálása szempontjából

Erre azért volt szükség, mivel egy csíkban csak olyan rendelések szerepel hetnek, amelyeknek egyik mérete megegyezik a csík szélességével, ezért a z_i szélességű csíkban levő téglalapok egyik mérete z_i, a másik mérete pedig a Z^i elemei közül kerül ki. A csíkon belül a téglalapok sorrendje tetszőleges ezért pl. az 5. ábra csíkjai a leszabás optimalizálása szempontjából azonosnak tekinthetők.

Mivel a téglalapok csíkon belüli sorrendje tetszőleges, ezért a csíkot egyértelműen meghatározza az, hogy melyik megrendelésből hány szerepel benne. Következésképpen minden csíkot jellemezhetünk egy vektorral, az ún. „csík-összeállítási-vektorral", egyszerűbben a *csík-vektorral.*

Legyen a csík-vektor:

$$C_j^i = (c_{j1}^i, c_{j2}^i, \ldots, c_{jn}^i)^*$$

ahol c_{jk}^i mutatja, hogy a k-adik rendelésből hány van a z_i szélességű j-edik csíkban.

A következő 6 feltételnek kell eleget tenni:

(1) $$c^i_{jk} \geq 0 \qquad (k = 1, 2, \ldots, n);$$

(2) $$\text{ha } c^i_{jk} > 0, \text{ akkor } z_i = x_k \text{ vagy } z_i = y_k;$$

(3) $$\sum_{k=1}^{n} c^i_{jk} \leq S_2$$

— legfeljebb ennyi téglalap lehet egy csíkban;

(4) $$\sum_{k=1}^{n} \operatorname{sign} c^i_{jk} \leq T$$

— legfeljebb ennyiféle rendelés lehet egy csíkban.

Legyenek $c^i_{jk_1}, c^i_{jk_2}, \ldots, c^i_{jk_\alpha}$ a C^i_j nemzéró komponensei és $b^i_{k_s}$ az r_{k_s} rendelés z_i-től eltérő mérete (négyzet alakú téglalapnál z_i-vel egyenlő) $(s = 1, 2, \ldots, \alpha)$, akkor

(5) $$\sum_{s=1}^{\alpha} b^i_{k_s} \cdot c^i_{jk_s} \leq A$$

— a csíkban levő téglalapok szélességeinek az összege nem haladhatja meg az alaptábla szélességét;

(6) $$\text{ha } A - \sum_{s=1}^{\alpha} b^i_{k_s} \cdot c^i_{jk_s} \geq z^1_i$$

$$\text{akkor } \sum_{k=1}^{n} \operatorname{sign} c^i_{jk} = T \text{ vagy } \sum_{k=1}^{n} c^i_{jk} = S_2$$

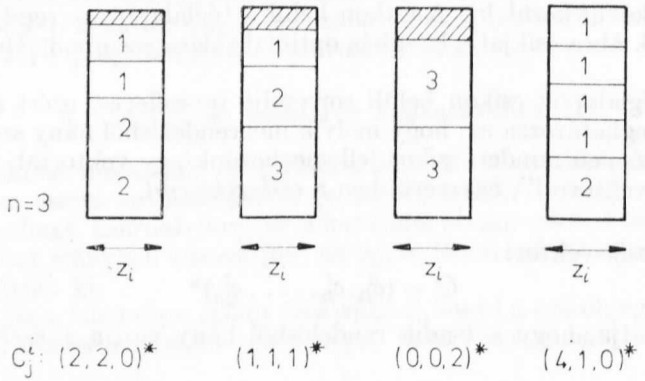
— a hulladékból több megrendelést már nem tudunk levágni.

Képezzük z_i-hez az összes lehetséges csík-vektort, ezáltal meghatározzuk az összes z_i szélességű csíkot.

A továbbiakban a C^i_j csík *veszteségén* a

$$w^i_j = z_i A - \sum_{k=1}^{n} c^i_{jk} \cdot x_k \cdot y_k$$

mennyiséget értjük.



6. ábra

Példák csíkok összeállítására 3 rendelés esetén

Tegyük fel, hogy a $z_1, z_2, \ldots, z_m$ szélességekhez az összes csík-vektort már meghatároztuk. Ki kell szűrni közülük azokat, amelyek nem szerepelhetnek az optimális szabástervben.

Egymás után alkalmazva a következő redukciós szempontokat vettük figyelembe:

(i) Ha egy rendeléshez két homogén csík tartozik (homogén egy csík, amikor csak egyféle téglalap van a csíkban, azaz $\Sigma_k \operatorname{sign} c_{jk}^i = 1$), pl. r_k-hoz az x_k és az y_k szélességű, és

$$\left[\frac{A}{x_k}\right] \leq \left[\frac{y_k}{x_k}\right] \cdot \left[\frac{A}{y_k}\right],$$

akkor az y_k szélességű csíkot elhagyjuk ([ ] az egész rész jele). Ugyanis, ha az optimális szabásterv valamelyik szabásképében szerepelne ez az y_k szélességű csík, akkor az x_k szélességűt a helyére rakva nem növelnénk a veszteséget.

(ii) Minden csík esetében megnézzük, vannak-e olyan csíkok, amelyekkel az úgy helyettesíthető, hogy ugyanazok a megrendelések szerepelnek az új csíkokban is, de a veszteség így már kisebb lesz. Ez az előbbi (i) redukciónak az általánosítása (azért vettük külön, mivel az (i) az egyszerűsége miatt a számítógépes programban is külön szerepel). Egy ilyen helyettesítés több különböző csíkból állhat, de csak olyanokból, amelyeknek a vizsgált csík szélességénél nem nagyobb a szélességük és amelyekben a vizsgált csík rendelésein kívül más rendelés nem szerepel. A vizsgált csíkot elhagyjuk, ha találunk a veszteségnél nem nagyobb veszteségű helyettesítést. (Feltételezzük, hogy a vizsgált csík olyan sokszor szerepel a szabástervben, hogy mindegyik behelyettesítendő csíkot legalább egyszer lehet alkalmazni.)

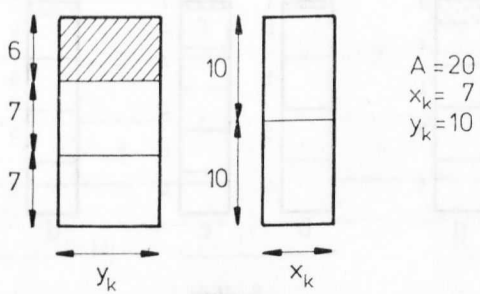
Pl.: legyen a helyettesítendő csík: C_1^i ($\Sigma_k \operatorname{sign} c_{1k}^i = 2$ és $c_{11}^i > 0$, $c_{12}^i > 0$), tegyük fel, hogy az optimális szabástervben B_1-szer szerepel és a vesztesége: w_1^i. Ekkor a téglalapokból $B_1 c_{11}^i$ és $B_1 c_{12}^i$ darabot kell levágni.

Legyen C_2^i egy behelyettesítő csík ($\Sigma_k \operatorname{sign} c_{2k}^i = 2$ és $c_{21}^i > 0$, $c_{22}^i > 0$), amelynek a vesztesége: w_2^i.

Ekkor

$$B_2 = \min_{k=1,2} \left\{ \left[\frac{B_1 c_{1k}^i}{c_{2k}^i} \right] \right\}$$

-szor lehet ezzel a csíkkal helyettesíteni az eredetit.



7. ábra

Numerikus példa az y_k szélességű homogén csík elhagyására

(Bevezettük a [ ] jelölést, amely bármely E valós számra

$$[E] = \begin{cases} [E] & \text{ha } E - [E] = 0, \\ [E] + 1, & \text{egyébként.} \end{cases}$$

Tegyük fel, hogy

$$B_2 \approx \frac{B_1 \cdot c_{11}^i}{c_{21}^i}$$

ekkor r_2-ből maradt

$$B_1 \cdot c_{12}^i - \frac{B_1 c_{11}^i}{c_{21}^i} \cdot c_{22}^i$$

darab. Ezt a mennyiséget homogén csíkból vágjuk ki, amely $\Sigma_k \operatorname{sign} c_{3k}^i = 1$ és $c_{32}^i > 0$, a vesztesége: w_3^i. A homogén csíkot B_3-szor kell venni,

$$B_3 \approx \frac{1}{c_{32}^i} \cdot \left(B_1 \cdot c_{12}^i - \frac{B_1 \cdot c_{11}^i}{c_{21}^i} \cdot c_{22}^i \right).$$

Az eredeti csíkot elhagyjuk, ha

$$B_1 \cdot w_1^i \geq B_2 \cdot w_2^i + B_3 \cdot w_3^i,$$

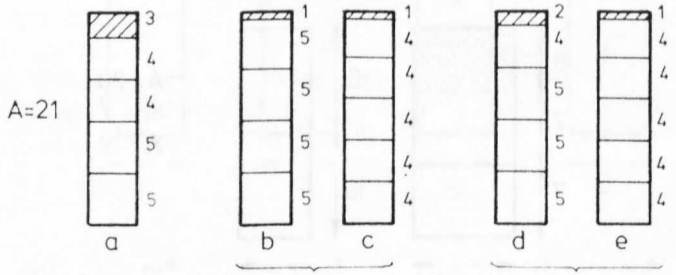
azaz behelyettesítve és B_1-el egyszerűsítve:

$$w_1^i \geq \frac{c_{11}^i}{c_{21}^i} \cdot w_2^i + \left(\frac{c_{12}^i}{c_{32}^i} - \frac{c_{11}^i \cdot c_{22}^i}{c_{21}^i \cdot c_{32}^i} \right) \cdot w_3^i. \tag{7}$$

Ebben a képletben már minden mennyiség ismert. Természetesen c_2^i homogén csík is lehet. A $T \geq \Sigma_k \operatorname{sign} c_{jk}^i > 2$ esetek is hasonlóan vizsgálhatók.

Megjegyzés: Az egészértékűséget a képletek megadásánál figyelmen kívül hagyjuk.

Példa: Ha a 8. ábrán levő a) csík 10-szer szerepel a szabástervben, akkor a b) csíkot 5-ször, a c) csíkot pedig 4-szer kell alkalmaznunk ahhoz, hogy ugyanazokat a megrendeléseket kisebb veszteséggel állítsuk elő. Hasonlóképpen, ha az a) csík 15-ször szerepel a szabástervben, akkor 10 db d) csík és 4 db c) csík kisebb veszteséggel pótolja.



8. ábra

Numerikus példa a (ii) csík-helyettesítésre

Az összes csík-vektor elkészítése és az előbb leírt redukciós lépések végrehajtása után megkaptuk a szabásrajzok képzéséhez szükséges összes csíkot és jellemzőiket. A következőkben a szabásképek előállítását ismertetjük.

Állítsuk sorba a csík-vektorokat:

$$(8) \qquad C_1^1, C_2^1, \ldots, C_{d_1}^1, C_1^2, C_2^2, \ldots, C_{d_2}^2, \ldots, C_1^m, C_2^m, \ldots, C_{d_m}^m,$$

ahol d_i a redukció után megmaradt z_i szélességű csíkok száma.

Az összes szabásképet a csíkok egymás mellé illesztésével készítjük el. Egy szabásképen belül a téglalapok sorrendje, elhelyezkedése az optimalizálás szempontjából tetszőleges (9. ábra), ezért a szabásképet egyértelműen meghatározza, hogy melyik téglalapból hány szerepel benne. Ebből következik, hogy az ún. „szabáskép-összeállítási-vektorral", röviden *szabás-vektorral* a szabásrajzot egyértelműen jellemezhetjük.

A csík-vektorok (8) felsorolásából kell kiválasztani azokat, amelyek egymás mellé illesztésével szabásképet akarunk meghatározni. Nézzünk egy kiválasztást, amelyben a $C_{j_1}^{m_1}$ csík-vektor Q_{j_1}-szer, ..., a $C_{j_\alpha}^{m_\alpha}$ csík-vektor Q_{j_α}-szor szerepel.

A kiválasztás által meghatározott szabás-vektor (legyen ez a j-edik) a

$$K_j = (K_{j1}, K_{j2}, \ldots, K_{jn})^* = \sum_{k=1}^{\alpha} Q_{j_k} \cdot C_{j_k}^{m_k}$$

összegvektor, ahol K_{ji} azt mutatja, hogy r_i-t hányszor kell levágni ebben a szabásképben.

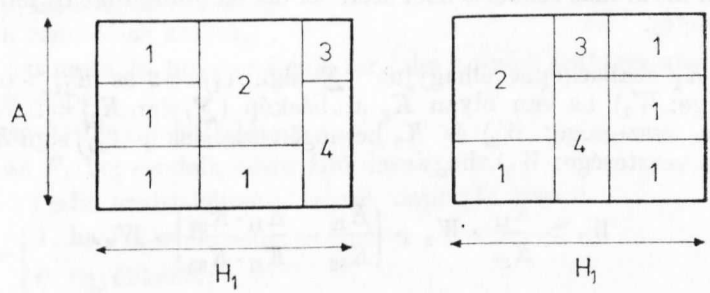
A szabás-vektoroknak a következő 4 feltételt kell kielégíteniök:

$$(9) \qquad \sum_{k=1}^{\alpha} Q_{j_k} \leq S_1$$

— legfeljebb S_1 csíkból állhat;

$$(10) \qquad \sum_{k=1}^{n} \operatorname{sign} K_{jk} \leq T$$

— legfeljebb ennyiféle rendelés lehet egy szabásképben;



9. ábra

Az optimalizálás szempontjából azonosnak tekintett szabásrajzok

(11) $$\sum_{k=1}^{\alpha} Q_{jk} \cdot z_{m_k} \leq H_p$$

— a csík-szélességek összege a legnagyobb alaptábla-hossznál kisebb.

A K_j-hez hozzárendelünk egy alaptábla-méretet, H_h^j-t:

$$H_h^j = \min \left\{ H_h \Big| \sum_{k=1}^{\alpha} Q_{jk} \cdot z_{m_k} \leq H_h, h = 1, 2, \ldots, p \right\}$$

(12) $$\text{Ha } H_h^j - \sum_{k=1}^{\alpha} Q_{jk} \cdot z_{m_k} \geq z_1,$$

$$\text{akkor } \sum_{k=1}^{\alpha} Q_{jk} = S_1 \text{ vagy } \sum_{k=1}^{n} \operatorname{sign} K_{jk} = T$$

— vagyis ha újabb csíkot lehetne illeszteni a meglevők mellé, akkor ezt már csak a (9) vagy (10) feltételek megsértésével lehetne.

A K_j-hez hozzárendelünk egy *veszteséget*, amely

$$W_j = AH_h^j - \sum_{k=1}^{n} K_{jk} \cdot x_k \cdot y_k.$$

A lineáris programozási feladat felírásához a szabásképek előbb meghatározott paramétereire van szükség. A szabásképek várhatóan nagy száma miatt itt is egyszerű redukciókat hajtunk végre (a csík-redukcióhoz hasonlóan) elhagyva azokat a szabásképeket, amelyek nem szerepelhetnek az optimális szabástervben.

(i) A j-edik homogén szabásképet elhagyjuk, akkor ($\sum_s \operatorname{sign} K_{js} = 1$ és $K_{jk} > 0$), ha van olyan K_i homogén szabásképp ($\sum_s \operatorname{sign} K_{is} = 1$ és $K_{ik} > 0$), hogy $H_h^j \cdot K_{ik} > H_h^i \cdot K_{jk}$.

Egyenlőség esetén csak az egyiket tartjuk meg.

(ii) A megmaradó szabásképeknél megnézzük, van-e olyan helyettesítés, amelyet a szabásképp helyére téve a veszteség nem nő, ha van ilyen, akkor ezt a vizsgált szabásképet elhagyjuk. A helyettesítés több különböző szabásképből állhat, de csak olyanokból, amelyekben a helyettesítendő szabásrajz rendelésein kívül más rendelés nem szerepel (az előbbi redukciós feltétel ennek speciális esete).

Pl.: A K_1 szabásképet elhagyjuk ($\sum_j \operatorname{sign} K_{1j} = 2$ és $K_{11} > 0$, $K_{12} > 0$, a vesztesége: W_1), ha van olyan K_2 szabásképp ($\sum_j \operatorname{sign} K_{2j} = 2$ és $K_{21} > 0$, $K_{22} > 0$, a vesztesége: W_2) és K_3 homogén szabásképp ($\sum_j \operatorname{sign} K_{3j} = 1$ és $K_{32} > 0$, a vesztesége: W_3), hogy

$$W_1 \geq \frac{K_{11}}{K_{21}} \cdot W_2 + \left(\frac{K_{12}}{K_{32}} - \frac{K_{11} \cdot K_{22}}{K_{21} \cdot K_{32}} \right) \cdot W_3.$$

A $T \geq \sum_j \operatorname{sign} K_{ij} > 2$ eset behelyettesítésének képlete is hasonlóan adódik. Az egészértékűségtől itt is eltekintettünk.

A redukciók után megmaradt szabás-vektorok száma legyen: M, a sorrendjük pedig rögzített:
$K_1, K_2, \ldots, K_M$, ahol W_j és H_h^j a megfelelő veszteség, ill. alaptáblahossz ($j = 1, 2, \ldots, M$).

3.1.2. *Az alaptábla hosszával párhuzamos csík-vágás*

Jelenleg az alaptábla szélességével párhuzamosan vágnak, mivel azonban a vágási irány megváltoztatására lehetőség van, ezért röviden kitérünk a hosszal párhuzamos vágásra is (4b ábra).

A gondolatmenet hasonló az előbb leírtakhoz, először az összes csíkot határozzuk meg, majd a csíkok egymás mellé illesztésével a szabásképeket. Itt a csíkokhoz kell egy alaptábla-hosszt rendelni és a szabásképeket alkotó csíkok hosszának maximuma lesz a szabáskép hossza. Mivel a csík a veszteségét „viszi magával" és az optimális szabásterv szabásképeinél az egyéb veszteség elenyésző a sok lehetséges összeállítás miatt, ezért a hosszal párhuzamos csíkvágás előnyösebbnek tűnik. Másrészt a csíkok száma ebben az esetben az alaptábla változó hossza miatt lényegesen több és ezáltal a szabásképek száma is ugrásszerűen megnő. Sajnos a két vágás összevetéséről nem állnak rendelkezésünkre adatok.

3.2. *Az egészértékű lineáris programozási feladat*

A fent leírt módon megkaptuk az összes szóba jöhető szabásképet. Ezután egészértékű lineáris programozási feladat (ILP) megoldásával megkapjuk az optimális szabástervet. Az ILP a következőképpen írható fel:

$$
\begin{aligned}
& u_j \in \{0, 1, 2, \ldots\} && (j = 1, 2, \ldots, M) \\
& \sum_{j=1}^{M} K_{ji} \cdot u_j \geq t_i && (i = 1, 2, \ldots, n) \\
& \frac{\sum_{j=1}^{M} q_{jk} \cdot u_j \leq D_k}{\sum_{j=1}^{M} H_h^j \cdot u_j \to \min}, && (k = 1, 2, \ldots, p)
\end{aligned}
\tag{13}
$$

ahol M a szabásképek száma,
n a rendelések száma,
K_{ji} azt mutatja, hogy az i-edik rendelés a j-edik szabásrajzban hányszor szerepel,
t_i az i-edik rendelés tételszáma,
D_k az F_k-ból rendelkezésre álló darabszám,
H_h^j a j-edik szabásképhez tartozó alaptábla hossza,

$$
q_{jk} = \begin{cases} 1, \text{ ha } H_h^j = H_k \\ 0 \text{ egyébként.;} \end{cases}
$$

A célfüggvény a felhasználandó félkésztermékek összfelületére utal; a cél ennek a minimalizálása, mivel felületveszteségen az összes felszabandó alap-

tábla felületének és a leszabott rendelések felületének különbségét értjük. A rendelések összfelülete konstans, az alaptáblák szélessége is állandó, ezért vehettük fel ilyen alakban a célfüggvényt. Gyakorlati oldalról ez azt jelenti, hogy a többlettermelést hulladéknak vesszük, azaz nem tároljuk a feleslegesen levágott darabokat.

A lehetséges megoldások halmaza üres is lehet, amennyiben a raktárban levő félkésztermékek összfelülete nem elég nagy. Bizonyos biztonsági tartalék feltételezésével ezt előre meg lehet becsülni. Természetesen akkor a legkisebb a felületveszteség, ha a D_k-kat tartalmazó feltételeket megszüntetjük, azaz tetszőlegesen nagy raktárkészletet feltételezünk. Ekkor a termelésirányítás is megvalósítható. A feladat együtthatómátrixa „ritka", egy oszlopban legfeljebb T elem pozitív, ezt előnyösen kihasználó eljárásokat érdemes konstruálni, míg viszont hátrányosnak tűnik az, hogy a modellben kevés a feltétel és sok a változó ($n \ll M$).

4. A modell számítógépes megvalósítása

A matematikai alapok leírása után az alaptáblák, a rendelések legfontosabb jellemzőit és a leszabás konkrét üzemi feltételeit adjuk meg, majd a próbafeldolgozás tapasztalatait ismertetjük.

Az üveggyárban a két-asztalos szabászgéppel természetesen az alaptáblának csak egy bizonyos osztályát szeretnék feldarabolni, amelyeknek fontos paraméterei összefoglalva az alábbiak:

(i) Az alaptáblák a minőségük szerint 4 osztályba sorolhatók: A, B, C és D minőség (természetesen ez változhat).

(ii) Az alaptáblák méretei jelenleg a következő értékek lehetnek:

Vastagság (mm)	Szélesség (mm)	Hosszúság (mm)
3	2000	2000, 2200, 2400
4	2000	2000, 2400, 2600
5–6	3000	2000, 2400, 2600

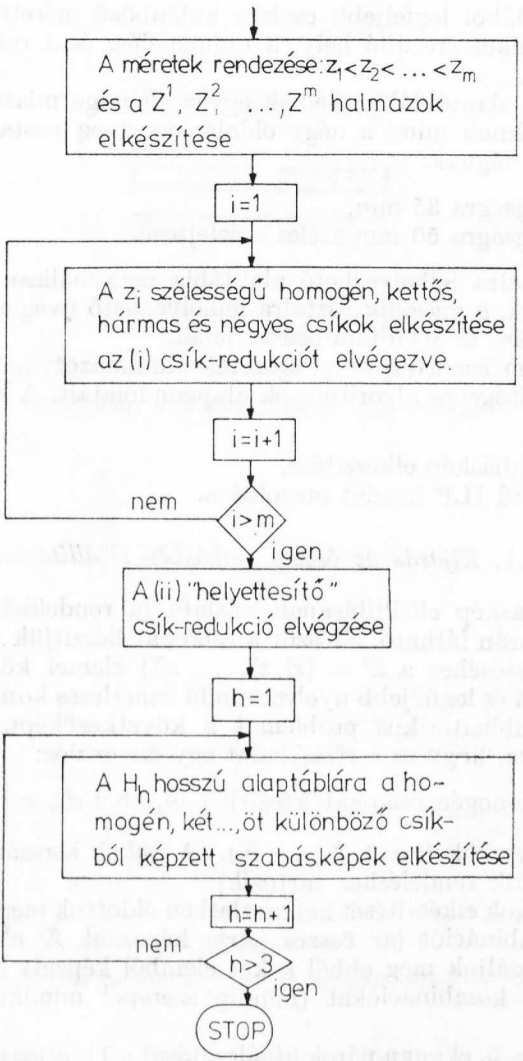
A rendelések jellemzői közül a minőség és a vastagság nyilvánvalóan megegyezik az alaptáblákéval. A szélesség és a hosszúság 256–2000 mm között milliméterenként változhat és gyakorlati megfontolások miatt a tételszám legalább 50 ($t_i \geq 50$).

A konkrét műszaki feltételek az alábbiak (a teljesség kedvéért a már említett feltételeket is megismételjük):

a) A szabás két vágóasztalon történik, az elsőn az alaptáblát csíkokra vágják fel, a másodikon a csíkokat egyenként darabolják fel a kívánt méretre.

b) Az első asztalon a gép 9 vágófejjel rendelkezik, a második asztalon 12 vágófej van, de itt is csak legfeljebb 9 működhet egyszerre. Mindkét asztalon az aktív 9 vágófej közül az első rögzített, a többi állítható, ezért $S_1 = S_2 = 8$ (azaz legfeljebb 8 csík szabható egy alaptáblából és egy csíkból legfeljebb 8 téglalap).

c) A vágófejek egymástól legalább 256 mm-re vannak (ez a minimálisan levágható téglalap-méret).



Bemenet: F és R elemei.
Kimenet: szabás-vektorok, veszteségek és alaptábla-hosszak.

Megjegyzés: kettős csíkon a $\sum_k \text{sign } c_{jk}^i = 2$ csíkot értjük, a hármas és négyes csík értelmezése hasonló.

10. ábra

A szabásképek előállításának vázlata

d) Egy alaptáblából legfeljebb csak 4 különböző méretű rendelést szabhatunk ki a rendelkezésre álló hely, a csomagolási és a raktározási gondok miatt, azaz $T = 4$.

e) A legyártott alaptáblák széleinek egyenetlensége miatt az a gyakorlat, hogy az alaptábláknak mind a négy oldalán az üveg vastagságától függően ún. „selejtcsíkot" vágnak:

3—4 mm vastagságra 35 mm,
5—6 mm vastagságra 50 mm széles a selejtcsík.

f) Az első asztalra felhelyezhető alaptábla maximálisan 3000 mm széles és 3500 mm hosszú, a második asztalra felhelyezhető üveglap pedig maximálisan 2000 mm széles és 3000 mm hosszú lehet.

A következőkben ismertetjük az általunk alkalmazott módszer két lépését megvalósító számítógépes algoritmusok alapgondolatait. A két lépés a következő:

(i) az összes szabáskép elkészítése,
(ii) a nagyméretű ILP feladat megoldása.

4.1. *Eljárás az összes szabáskép előállítására*

Az összes szabáskép előállításának vázlata (a rendelésállomány egy csoportjára) a 10. ábrán látható. Először a csíkokat készítjük el. A z_i szélességű csík-vektorok képzéséhez a $Z^i = \{z_i^1, z_i^2, \ldots, z_i^N\}$ elemei közül kell az összes legalább elsőrendű és legfeljebb nyolcadrendű ismétléses kombinációt kiválasztani. Ezt a kombinatorikus problémát a következőképpen oldottuk meg, erősen kihasználva, hogy ez a részfeladat egy-dimenziós:

1) Először a homogén csíkokat készítjük el, ahol $c_{kj}^i = \left[\frac{A}{z_i^j}\right]$ és az (i) csík-redukciót is elvégezzük ($j = 1, 2, \ldots, N$). (A csíkok sorszámozásától eltekintünk z_i^j a k_j = edik rendeléshez tartozik).

2) A kettős csíkok elkészítését két részletben oldottuk meg. Először az összes másodrendű kombinációt (az összes párt) képezzük Z^i elemeiből, majd ezt felhasználva vizsgáljuk meg ebből a két elemből képzett legfeljebb nyolcadrendű ismétléses kombinációkat (mindig szerepel mindkét elem a kombinációban).

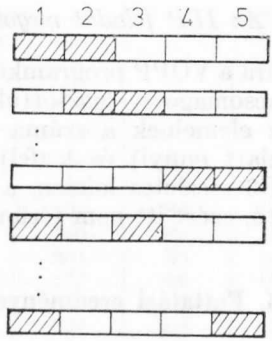
Pl. Legyen $N = 5$, ekkor a párok kiválasztását a 11. ábra szerint végezzük el, ahol a vonalkázott rész a pár tagjait jelöli. Legyen egy ilyen pár z_i^q és $z_i^s (z_i^q < z_i^s)$ és tegyük fel, hogy z_i^q a j., z_i^s a k. rendeléshez tartozik, a belőlük előállítható csík-vektorok képzését a 12. ábra mutatja.

3) A hármas és négyes csíkok elkészítése hasonló módon történik, mint a kettős csíkoké.

Az algoritmus alapján világos, hogy az összes lehetséges csík-vektort előállítjuk az előbb vázolt módon.

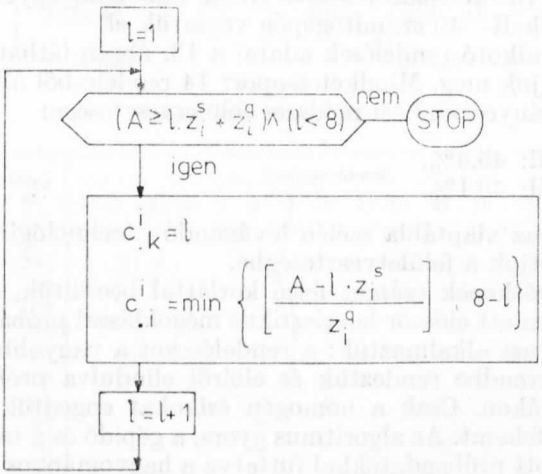
4) Az összes csík-vektor elkészítése után a (ii) „helyettesítő" csík-redukciót vizsgáljuk meg. A végrehajtása időigényes, ezért a képletet egyszerűsített formában vesszük figyelembe. A kettős csíkoknál (7) helyett csak a

$$w_1^i \geq \frac{c_{11}^i}{c_{21}^j} \cdot w_2^j + \frac{c_{12}^i}{c_{32}^i} \cdot w_3^i$$



11. ábra

A másodrendű kombinációk kiválasztása



12. ábra

Kettős csíkok képzésének blokkdiagramja

képletet nézzük, a hármas és a négyes csíkoknál is hasonlóan járunk el. Mivel a csíkok a szélességük szerint monoton nem-csökkenő sorrendben vannak, ezért hátulról visszafelé haladunk. A helyettesítést kereső eljárás az egy fa „endorder" bejárásához hasonló, ahol a fa gyökere a helyettesítendő csík. Egy út az egy lehetséges helyettesítés. A balrészfán lemegyünk a levélig és onnan megyünk balról-jobbra végig a leveleken (itt csak a leveleket keressük meg).

5) A szabásképeket a csíkok elkészítéséhez hasonlóan állítjuk elő. Először az egyféle, majd a két-, három-, négy- és ötféle csíkokat tartalmazó szabásrajzokat keressük meg a H_1 hosszú alaptáblára, azután a H_2 hosszú alaptáblára, stb. A legalább hatféle csíkot tartalmazó szabásképeket nem készítjük el, mivel kis valószínűséggel fordulnak elő. A szabásképek előállításánál a (9—12) feltételeket kellett figyelni, és azt, hogy kisebb alaptáblából ne legyen leszabható, azaz kétszer ne állítsuk elő.

4.2. *Az ILP feladat megoldása*

Az ILP feladat megoldására a VOPP programkönyvtárban levő VDG3 nevű vegyes-egészértékű programcsomagot választottuk ki, amely DOS-ban futtatható. Az együtthatómátrix elemeinek a száma legfeljebb 8192 lehet (ez a FORTRAN tömb-méret miatt ennyi) és legfeljebb 400 lehet a sorok, ill. az oszlopok száma. A VDG3 részletes leírása, a szükséges környezet ismertetése [13]-ban megtalálható, ezért itt nem térünk ki erre.

5. Futtatási eredmények

A következőkben a gyártól kapott rendelésállomány két csoportjának próbafeldolgozását ismertetjük, összevetve a számítógépes eljárás optimális szabástervét a hagyományos „sorozatvágás" és egy heurisztikus „mohó" algoritmus eredményeivel. A futtatásokat a József Attila Tudományegyetem Kibernetika Laboratóriumának R–40 számítógépén végeztük el.

A csoportokat alkotó rendelések adatai a 13. ábrán láthatók. A méreteket centiméterben adjuk meg. Mindkét csoport 14 rendelésből áll. Erre a két csoportra a hagyományos szabási módszer felületvesztesége:

az 1. csoportnál: 45,6%,
a 2. csoportnál: 40,1%.

Természetesen az alaptábla szélén levágandó „technológiai" selejtcsíkot is mindig beleszámítjuk a felületveszteségbe.

Amikor a szabásképek számát felső korláttal becsültük, ez a szám olyan nagy lett, hogy emiatt először heurisztikus megoldással próbálkoztunk. Az ún. „mohó" algoritmust alkalmaztuk: a rendeléseket a nagyobb méretük szerint nem-növekvő sorrendbe rendeztük és elölről elindulva próbáltuk elhelyezni őket az alaptáblákon. Csak a homogén csíkokat engedtük meg, azaz egydimenziós volt a feladat. Az algoritmus gyors, a gépidő és a tárigény elenyésző, de a gyártól kapott próbaadatokkal futtatva a hagyományos szabáshoz képest átlagosan 6–9%-kal csökkent csak a felületveszteség. Pl. az 1. csoportnál 37% lett, azaz a sorozatvágáshoz képest 8,6%-kal csökkent, míg a 2. csoportnál 6,9% a megtakarítás.

A VDG3 program-adatok méretei miatt egy csoporton belül legfeljebb 20 rendelés optimalizálását láttuk célszerűnek — amit a gyártól kapott adatok is igazoltak, mivel egy csoportban húsznál több rendelés elvétve fordult csak elő —, így 400 szabásképet tudtunk figyelembe venni. Az 1. csoportnál a csíkok száma 26 lett, ezekből a csíkokból 862 szabásképet kaptunk (az (i) és (ii) szabáskép-redukciók végrehajtásával ez 650 alá csökkenthető). Felmerült a probléma, hogyan válasszuk ki a legfeljebb 400 szabásképet? A próbafeldolgozásnál ezt heurisztikusan végeztük el azon elv alapján, hogy a kiválasztott szabásképek a legkisebb veszteségűek legyenek és a közel 400 szabásképben a rendelések gyakorisága közel egyenlő legyen. Így 395 szabásképet kaptunk (a raktárt figyelmen kívül hagytuk), a VDG3 ezekből állította össze az optimális szabástervet, amely a 14. ábrán látható a 2. csoport optimális szabástervével együtt. A szabás-vektor i-edik komponense a beolvasás sorrendjében i-edik rendeléshez tartozik, a nullákat nem tüntettük fel (a szabás-vektor itt sorvektor). Pl. az S1 szabás-vektort 200×220 cm-es alaptáblára kell 204-szer

Vastagság: 3 mm, minőség: C.			Vastagság: 3 mm, minőség: B.		
Azonosító	Méretek	Tételszám	Azonosító	Méretek	Tételszám
A1	134×74	420	B1	152×80	220
A2	50×74	207	B2	184×84	700
A3	134×132	300	B3	100×100	200
A4	50×132	150	B4	162×78	300
A5	132×102	204	B5	130×82	150
A6	50×102	170	B6	134×74	280
A7	132×132	204	B7	50×74	138
A8	86×132	204	B8	134×132	200
A9	84×84	204	B9	50×132	100
A10	134×44	204	B10	132×102	136
A11	150×134	80	B11	132×132	136
A12	96×85	80	B12	86×132	136
A13	82×82	80	B13	84×84	136
A14	150×44	80	B14	134×44	136
1. csoport			2. csoport		

13. ábra

A rendelésállomány két csoportja

Azonosító	Hány db	Szabás-vektorok 1	2	3	4	5	6	7	8	9	10	11	12	13	14	Alaptábla hossz
S1	204	1	1	1							1					220
S2	96	1	1	1	1											220
S3	122	1	1		1			1								220
S4	32				1			1	1							240
S5	50				1			1		2						240
S6	52								2	2						240
S7	34					1			2							240
S8	85					2	2									220
S9	40											1	2		1	240
S10	40											1		2	1	240

1. *csoport.* Az optimális szabásterv felületvesztesége: 21,7%

Azonosító	Hány db	Szabás-vektorok 1	2	3	4	5	6	7	8	9	10	11	12	13	14	Alaptábla hossz
T1	1													4		200
T2	75				1	2										220
T3	51				1					1		1				220
T4	85	1								1		1				220
T5	135	1							1	1						220
T6	66								1					2	1	240
T7	71				1		2								1	220
T8	69		1				2	2								240
T9	232		2													200
T10	167		1	1												200
T11	33			1							1		1			240
T12	104				1						1		1			220

2. *csoport.* Az optimális szabásterv felületvesztesége: 23,4%

14. ábra

A két csoport optimális szabásterve

alkalmazni és a szabásmintában az A1, A2, A3, A10 jelű rendelések mindegyike egyszer szerepel. Másrészt az A1 rendelés az S1, S2 és S3 szabásképekben kerül leszabásra.

A szabásképek előállításának gépideje 2—3 perc csoportonként az R—40 számítógépen, a VDG3 futási idejére nehéz becslést adni, mivel a folytonos optimum megtalálása után változó számú transzformációt hajt végre, amíg a megadott határon belül levő egészértékű megoldást eléri. Pl. az 1. csoportnál 21 transzformációt végzett el. Az egészértékű megoldás a folytonos optimumtól 0,7%-kal tért el, ez az eltérés az alkalmazás szempontjából lényegtelen. A szabás-vektorokból a szabásképek elkészítése egyszerűen elvégezhető.

	Sorozatvágás	„Mohó" alg.	Optimális
1. csoport	45,6%	37,0%	21,7%
2. csoport	40,1%	33,2%	23,4%

15. ábra
A felületveszteségek összevetése

A gyakorlatban jogos igény a gyár szakemberei részéről a rendelések kielégítésének és az alaptáblák bekészítésének „folyamatossága". Azaz, ha egy rendelés szerepel egy szabásképben, akkor addig szerepeljen a soron következőkben, amíg ki nincs elégítve. A lineáris programozással kapott optimális szabásterv nem tesz eleget a folyamatosságnak, a szabásképek csak „majdnem folyamatos" sorrendbe állíthatók. A 14. ábrán már ilyen sorrendben vannak, az 1. csoportnál csak egy rendelés leszabása szakítódik meg, és az is csak egyszer. A 2. csoportnál, ha a T1 szabáskép helyett a T6-ot nem 66-szor, hanem 68-szor alkalmazzuk, akkor csak egy rendelés leszabása szakítódik meg kétszer. Mindkét sorrend a gyakorlatban megfelelő, mivel egy rendelés tárolására van hely. Ezt a két sorrendet próbálgatással kaptuk. A legelőnyösebb sorrend kialakítására adott algoritmust Dyson és Gregory [5], valamint Madsen [6—7], ez utóbbi szerző [7]-ben részletes futtatási adatokat is közöl. A folyamatosság problémájára egy elméleti jellegű megközelítés található [12]-ben.

Összefoglalás

Tanulmányunkban az Orosházi Üveggyár síküveg-szabás problémájának egy lehetséges megoldását mutattuk be. Az üzemi műszaki feltételeket nagymértékben kihasználó, gyorsan és könnyen elkészíthető modellt adtunk meg a gyakorlati alkalmazhatóságot tartva mindig szem előtt. A dolgozatban leírt eljárás „életképességének" az igazolására próbafeldolgozásokat végeztünk, amely igazolta elképzeléseinket, habár „csak" szuboptimális szabásterveket kaptunk a nagy veszteségű szabásképek kihagyása miatt. A modell felállításával kapcsolatban és az elkészült programok nyomán számos olyan kérdés merült fel, amely nem csupán az optimum jobb megközelítését célozza, hanem a hatékonyabb felhasználást is lehetővé teszi. Ilyen pl.: a korlátozott számú szabáskép legmegfelelőbb kiválasztása, a speciális ILP feladatot előnyösen megoldó

módszer alkalmazása. Továbbá olyan sorrendben megadni az optimálisnak kapott szabásképeket, hogy a vágási folyamatban a lehető legkényelmesebben legyen végrehajtható (vagyis a darabolás folyamán lehetőleg kevés olyan rendelésnek kelljen a szabászasztal mellett helyet biztosítani, amelynek a leszabását fel kellett függeszteni más megrendelések miatt) stb. Hasonló problémákról és az optimalizálásra elkészült programrendszerről külön cikk készül [14].

Köszönetnyilvánítás: Ezúton köszönjük meg Moldoványi Ferenc és Tóth Pál (Orosházi Üveggyár Szervezési Osztály) értékes segítségét, amellyel az üvegszabás műszaki, üzemi körülményeinek tisztázását nagymértékben elősegítették.

(Beérkezett: 1981. március 12-én.)

IRODALOM

[1] Gilmore, P. C.—Gomory, R. E.: A linear programming approach to the cutting stock problem. Operations Research 9, 1961 (849—859).
[2] Op. cit. Part II. Operations Research 11, 1963 (863—888).
[3] Gilmore, P. C.—Gomory, R. E.: Multistage cutting stock problems of two and more dimensions. Operations Research 13, 1965 (94—120).
[4] Gilmore, P. C.—Gomory, R. E.: The theory and computation of knapsack functions. Operations Research 14, 1966 (1045—1074).
[5] Dyson, R. G.—Gregory, A. S.: The cutting stock problem in the flat glass industry. Operational Research Quarterly, 1974 (41—54).
[6] Madsen, Oli B.: Glass cutting in a small firm. Mathematical Programming 17, 1979 (85—90).
[7] Madsen, Oli B. G.: A cutting sequencing algorithm. Research report from IMSOR, No 12/1979 (Lyngby).
[8] Herz, J. C.: Recursive computational procedure for two-dimensional stock cutting. IBM Journal Research and Dev. 1972 (462—469).
[9] Christofides, N.—Whitlock, C.: An algorithm for two-dimensional cutting problems. Operations Research 25, 1977 (30—44).
[10] Adamowicz, A.—Albano, A.: A solution of the rectangular cutting-stock problem. IEEE Trans. on Systems, Man and Cyber. 6, 1976 (302—310).
[11] Lampl, T.: Anyagleszabási tervek készítése és programozása matematikai eszközökkel. Operációkutatási Esettanulmányok (SZÁMOK), 1971.
[12] Lengyel, I.: Síküveg szabásának optimalizálása. Diplomamunka JATE, 1979.
[13] R-40 POS VOPP DOS/ES, Diszkrét optimalizálás, Robotron.
[14] Galambos, G.—Csirik J.: Síküveg szabásának optimalizálása II., publikáció alatt.

OPTIMIZATION OF THE CUTTING OF FLATGLASS

A possible solution to the problem of cutting of flatglass in the Glass Work of Orosháza is presented. We propose a model that can be implemented rapidly and easily and makes use of the technological conditions of the factory keeping practical applicability always in view. In order to prove the "viability" of the procedure trial calculation was made that verified our expectations, though "only" sub-optimal cutting patterns were obtained because of the pre-selection of cutting patterns with great losses. In the course of setting up the model and calculating the programmes several problems have arisen that are aimed not only at a better approximation to the optimum, but also enable a more efficient utilization. Such are for example: the best preselection of the cutting patterns in a limited number, the adaptation of an integer LP method which utilizes the special structure of the problem. Furthermore, producing the cutting patterns in such a sequence that in the cutting process they may be most comfortably carried out (i.e. possibly few such orders should be stored beside the cutting table that have to be put aside later on because of other orders), etc. A separate article [14] is under preparation on these problems and on the computer programme for optimization.

ОПТИМИЗАЦИЯ РАСКРОЯ ЛИСТОВОГО СТЕКЛА

В данной работе приводится одно из возможных решений проблемы раскроя листового стекла на Стекольном заводе в Орошхаза. Излагается модель, в значительной мере обеспечивающая использование технических условий завода и которая быстро и легко может составляется, никогда не упуская из виду практическую используемость. В порядке доказательства «жизнеспособности» метода, приводимого в работе проводилась опытная обработка, которая подтвердила имеющиеся предположения, хотя были получены «лишь» субоптимальные проекты раскроя из–за упускания раскроечных схем, которые приводят к большим потерям. В связи с развертыванием модели и на основании разрабатываемых на ее базе программ возникает такой вопрос, который направлен не только на достижение оптимума, а также позволят добиваться более эффективного использования. Такими являются, например, наилучший выбор раскроечной схемы, применение метода наиболее выгодно решающий специфическое задание целочисленного линейного программирования. Далее, оптимально получаемые раскроечные схемы даются в такой очередности, чтобы в процессе резки они выполнялись с наименьшими трудностями (т. е. в ходе раскроя на нарезной стол следует подавать такие заказы, из-за которых, по возможности, как можно меньше нужно было бы приостанавливать работу в связи с выполнением других заказов) и т. д. Относительно аналогичных проблем и програмной системы оптимизации будет подготовлена специальная статья.