

Absztrakt

A dolgozat célja egy olyan modern szemléletű webalkalmazás készítése, mely effektíven támogatja egy asztalosipari cég munkáját. Az alapvető ügyintézési folyamatok mellett (ügyfelek, rendelések, raktárkészlet kezelés) a korábban felvezetett termékleírások alapján szabásminta generálást valósít meg konfigurálható módon.

A szabásminta generálás folyamata során két-dimenziós optimalizálást elvégezve, olyan optimális szabásminta létrehozása a cél, amely tartalmaz a célgép függvényében vízszintesen, vagy függőlegesen átmenő vágásokat. A szabásminta ilyen jellegű előállításakor az alkatrészek pozíciójának meghatározásán túl, a legkisebb anyagvesztés elérése mellett, figyelembe kell venni a korábbról megmaradt alapanyagokat is.

Kulcsszavak

2D optimalizáció, Guillotine vágás, ügyviteli rendszer

Motiváció

Családi vonatkozásokra visszavezethető ennek a szoftvernek az igénye. Szükség volt egy olyan ügyviteli rendszerre, melyben a napi ügyintézés mellett a termékek alkatrészeinek tekintetében a táblakiosztásokat is képes elvégezni, számlafigyelés, átmenő vágás, és maradékfelhasználással együtt. A dolgozat elkészülésének idejében a nyilvánosan elérhető információk alapján nem volt olyan költséghatékony rendszer a piacon, ami a megjelent igényeket lefedte volna.

Hasonló fejlesztések

- ▶ A piacon található többféle szabásminta előállító szoftver, azonban ezeknek a kialakítása, mind funkcionalitásai nem csak a szabásminta optimalizálásban, de az egyéb ügyviteli lehetőségeikben is eltérnek.
- ▶ Például a **BIPPO** és **PaneCutter** rendszerek, csak szabásminta optimalizálásra alkalmasak, melynek eredményét a képernyőn jelenítik meg nyomtatható formában. A szabásmintán az alkatrészek irányát is megtartják, de a **PaneCutter**-nél a vágási paraméterek is definiálhatók.
- ▶ **AMORF** és **Optimik4** alkalmazásoknál már megjelennek a raktárkezelési funkciók is. Az **AMORF** már kereszt és hosszirányú vágásokkal is képes számolni az optimalizálás során, viszont az **Optimik4** a maradék táblák felhasználásával is számol. Mindezek mellett az **Optimik4**-ban elérhető a pénzügyi adatok és a dokumentumok kezelése is, mellyel támogatható a napi ügyvitel egy része.

Sikerek

- ▶ Kari TDK részvétel
- ▶ OTDK részvétel
- ▶ 1 konferenciacikk

Saját módszer bemutatása

A dolgozatban a hasonló rendszerek feltárása mellett a szakirodalomban megjelenő optimalizáló algoritmusok kerülnek bemutatásra azok főbb jellemzőik ismertetésével együtt. A tárgyalt optimalizáló algoritmusok az alábbiak:

- ▶ 2- és 3-fázisú Guillotine vágások,
- ▶ Sarokba helyezés módszer,
- ▶ Heurisztikus elhelyezési módszerek (Bottom-Left, Bottom-Left-Fill).

Az egyes algoritmusok működése részletesen ismertetve lett saját példákon ábrákkal szemléltetve a műben és egy olyan megoldás készült, mely saját ötleten alapulva Sík felosztása módszernek lett elnevezve. A rekurzív eljárás az alábbi lépésekből tevődik össze:

1. Szélességük alapján csökkenő sorrendbe helyezzük az alkatrészeket.
2. A legszélesebb alkatrész a bal felső sarokba, helyezzük, majd lefelé meghosszabbítva jobb oldalát, két szabad területet kapunk, egyet a lehelyezett alkatrész alatt, a másikat mellette.
3. Ezután kezdődik el az első fázis: elkezdjük a legelső elem alatti területre elhelyezni az alkatrészeket, mindig úgy, hogy a jobb oldali oldalukat meghosszabbítva kapott alattuk lévő területet használjuk, egészen addig, amíg el nem fogy a hely. Amennyiben elfogyott, akkor az utoljára lehelyezett elem melletti síkon lerakunk egy következő elemet.
4. Abban az esetben, hogyha alá elfér egy következő alkatrész akkor, azzal épp úgy, mint az első fázisnál, elkezdjük a következő fázist, és rakjuk egymás alá az alkatrészeket, amíg van hely. Amikor elfogyott megnézzük az utoljára lehelyezett elem melletti részt, és indítjuk a következő fázist.
5. Amennyiben nem tudunk oda elemet elhelyezni, akkor megnézzük az utolsó előtti lehelyezett elemet, és azzal indítjuk a következő ciklust.
6. Addig lépünk vissza és indítjuk a következő rekurziót backtrack jellegűen, amíg el nem fogynak a lehelyezendő elemeink.

Az optimalizáló motor egy ASP.NET alkalmazáson belül lett elhelyezve, mely rétegelt felépítést követ. Az adatokat MS SQL adatbázisban tárolja. A szabásmintákat a képernyőn való megjelenítés mellett PDF-ként is legenerálja a rendszer.

Eredmények értékelése

Az elkészült Sík felosztása optimalizáló módszer össze lett hasonlítva a 2-fázisú Guillotine módszerrel. A kvantitatív kiértékelés érdekében egy saját formula lett bevezetve, amely a maradék területek minimalizálását részesíti előnyben az (1) egyenlet szerint.

$$\text{eredmeny} = \frac{T_{\text{maradek}}}{T_{\text{egesz}}} \cdot \frac{\sum_{i=0}^n \frac{T_{\text{maradek}_i}}{T_{\text{alkatresz}_{\text{max}}}}}{n} \quad (1)$$

A két módszer implementálva lett a szoftverben és különböző teszteseteken keresztül történt a kiértékelés.

Összességében elmondható, hogy maradék felhasználás nélkül eredményesebb a Sík felosztása algoritmus, viszont a maradék felhasználásnál ilyen következtetés nem vonható le. Mivel az előző optimalizálás maradékait használjuk fel, a Sík felosztása módszer néhány esetben kevesebb alapanyagot használ fel, kevesebb lesz a maradék, így ez azt eredményezheti, hogy a maradék felhasználásával, több új alapanyagra van szükség, amiből több maradék keletkezik. Mindazonáltal figyelembe kell venni, hogy a Sík felosztása jobbnak tekinthető, elrendezése a maradék alapanyag táblákat nagyobb hatásokkal tudja feldolgozni.

A hasonló rendszerekkel foglalkozó fejezetben megvizsgált szoftverekkel összevetve, elmondható, hogy több funkcióval rendelkezik, mint a megtekintett szoftverek nagy része, azonban ezeknek egyike sem annyira mély, és kidolgozott. A szabásminta generáló komponens tekintve egy szinten van a többi a szoftverrel. A dolgozat részeként elkészült alkalmazás egy átmenetet képez tudásban és lehetőségekben a **PaneCutter** és az **Optimik4** alkalmazások között, mivel az előbbi szabásminta generálása volt a leginkább kezdetleges, az utóbbi pedig rendelkezik az újonnan elkészült alkalmazás funkcióival, azonban jobban kiforrott módon szolgáltatja összevetve a saját szoftverrel. A **PaneCutter**-rel szemben az elkészült alkalmazás szabásminta generálása jobbnak tekinthető, illetve több algoritmust tud használni.

