

Alkatrészek automatikus elhelyezése nyomtatott áramköri lapokon

ETO 621.3.049.76.001.2:681.3.06 ESZTER

A nyomtatott áramköri kártyák kézi tervezése iterációs lépések sorozatából áll. A tervező valamilyen kezdeti alkatrész-elrendezésből kiindulva megkísérli megrajzolni a fóliautak nyomvonalát. Ha ez nem sikerül, változtat az elrendezésen, és folytatja a nyomvonalak kijelölését. Gyakorlottabb tervező — bár az elhelyezés és fóliatervezés ciklikus ismétlése itt is megtörténik — több iterációs lépést fejből végez el, s ezzel gyorsítja az eljárást.

A kézi tervezés jól bevált módszerét a számítógépekkel nem sikerült megvalósítani, mivel a megoldás éveken tartó számolást eredményezne. A tervezési idő csökkentése érdekében a két lépést; az elhelyezést és fóliatervezést külön kellett választani, ami természetesen számos problémát vetett fel. Az alkatrészek elrendezését a számítógépnek úgy kell elvégeznie, hogy közben nem ellenőrizheti megoldása helyességét, nem „tudhatja”, hogy az adott elrendezésben a fóliautak megtervezhetők-e. Ez a probléma célfüggvények segítségével oldható meg. Az elhelyező programok algoritmusai az illető célfüggvény optimális vagy ahhoz közelálló értékének meghatározására törekednek. A „jó” elrendezés — amely biztosítja a fóliautak sikeres megtervezhetőségét — igen sokban függ a célfüggvény helyes megválasztásától, ami ugyancsak nem egyszerű feladat.

1. A feladat megfogalmazása

Az előző részben csak általánosságban beszéltünk alkatrészek elhelyezéséről. Ezek akár integrált áramköri IC tokok vagy alapelemek, akár diszkrét elemek lehetnek. Az IC tokok, illetve a diszkrét elemek elhelyezése azonban két különböző feladatot jelent. Az előbbiek csak a kártyán előre meghatározott tokhelyekre kerülhetnek, míg az utóbbiak helyzetére általában nincsenek korlátok. Az alábbiakban elsősorban az IC tokok elhelyezési problémáival fogunk foglalkozni.

A feladat matematikailag következőképpen fogalmazható meg. Adott n számú elem és n számú elemhely, valamint két $n \times n$ méretű mátrix: C és D . $c(i, j)$ jelentse az i -edik és j -edik elem közötti összekötések számát, $d(i, j)$ pedig az i -edik és j -edik elemhely távolságát. Ha p az első n egész szám valamely permutációja, akkor elhelyezési célfüggvényünk a következő

$$F(p) = \sum_{i,j=1}^n c_{ij} \cdot d_{ipj}, \quad c_{ii} = 0.$$

A távolságot derékszögű koordinátakülönbségekkel fejezzük ki, vagyis

$$d_{ij} = |x_i - x_j| + |y_i - y_j|,$$

ahol (x_k, y_k) a k -adik elemhely referenciapontjának

koordinátái a kártyához rögzített koordináta-rendszerben, meghatározott mértékegységben. A feladat tehát olyan permutáció megkeresése, amelyre az $F(p)$ célfüggvény értéke minimális lesz. Ez egzakt megoldás esetén $n!$ nagyságrendű permutáció megvizsgálását jelenti, ami gyakorlatilag még számítógéppel sem valósítható meg.

Ezen ún. másodrendű hozzárendelési feladat közelítő megoldására számos eljárás található az irodalomban. Steinberg [1] pl. a problémát az összekötetlen halmazok megkeresésével elsőrendű hozzárendelési feladatra vezette vissza, ami többféle módon (pl. a magyar módszerrel) viszonylag könnyen megoldható. Az eredmény jósága nagymértékben függ az összekötetlen halmazok megválasztásától.

Az eljárást Rutman [2] az elemek cserélgetésével ugyan megjavította, de az algoritmus időigénye túl nagy.

Egyes eljárások [3, 4] az elhelyezési probléma megoldását más oldalról közelítik meg: pl. a kereszteződéseknek vagy a vezetékek töréspontjainak számát minimalizálják. Kísérletek történtek az erő-nyomaték egyensúlyán alapuló módszerek (ACCEL) felhasználására is [5, 6] amelyet elsősorban a diszkrét elemek elhelyezésére dolgoztak ki.

2. Az elhelyezési probléma megoldása

A diszkrét optimálási eljárások közös jellemzője, hogy egy — általában többváltozós — függvény minimumát vagy maximumát keresik megadott tartományon belül. Megoldásukra több numerikus módszer ismeretes: pl. a korlátozás és szétválasztás módszere (*branch and bound*), teljes leszámolás stb. E megoldások azonban igen lassúak és időigényük a változók számának növekedésével ugrásszerűen növekszik.

Emiatt számos heurisztikus módszert is kidolgoztak, amelyek adott ún. kezdeti elhelyezésből indulnak ki. Ezek segítségével gyors eredményt érhetünk el ugyan, de a siker nagymértékben függ a kezdeti elhelyezéstől. A legjobb megoldást a szekvenciális módszer adja, amelynek időigénye is viszonylag kicsi, és az eredménynek az optimumtól való eltérése pontosan meghatározható.

A szekvenciális módszerek [7, 8] a diszkrét optimálás jellegzetes példái, a keresett elhelyezést két lépésben alakítják ki. Az első lépésben egy gyors kezdeti elrendezést hoznak létre, amely az optimálistól általában még igen erősen eltér. A második lépésben a kezdeti elrendezést az elemek áthelyezésével (cserélgetésével) finomíthatják tovább.

A cserélgető eljárások közös jellemzője, hogy csak a teljes permutáció egy meghatározott részhalmazával számolnak. Ilyen megoldás pl. a Monte-Carlo mód-

szer, amely a teljes permutációból véletlenszerűen választ ki értékeket.

Az ESZTER 30 programcsomagban az elhelyezési probléma megoldását az ún. páros, ill. hármas cserékkel végezzük.

A páros csere a permutálandó elemekből alkotható összes lehetséges elempár szisztematikus felcserélését jelenti, míg a hármas cserék esetén az összes lehetséges elemhármas elemeinek ciklikus fekcserélését végezzük el.

A cserélgetési eljárások természetesen nem adhatnak minden esetben optimális megoldást, hiszen a feladat egzakt elvégzéséhez az elemek összes lehetséges permutációjára szükség volna. A kártyatervezés fázisainak emiatt szétválasztása miatt azonban a gyakorlatban nem is törekszünk optimális megoldásra. Az elhelyező program által szolgáltatott eredmény ugyanis ideális összekötéseket tételez fel, amelyeket az utána következő föltervező-program általában nem tud megvalósítani.

Matematikailag megfogalmazva a szekvenciális módszer feladata a következő.

Helyezzünk el n számú elemet m számú helyre ($n \leq m$), úgy, hogy az elhelyezési célfüggvény minimális legyen. (Elemen itt a legkisebb permutálható egységet értjük.) Az IC alapelemek és tokok permutálhatósági eltéréseivel a későbbiekben foglalkozunk.

Ha $J(i)$ jelenti az i -edik elem lehetséges helyeinek halmazát, $P(k)$ a k -adik helyre kerülő elemet, és F az elhelyezési célfüggvényt, akkor keresendő

$$\min F[P(1), P(2), \dots, P(n)],$$

és $1 \leq j \leq n$, ahol

$$j \in J[P(j)] \text{ és } j=k.$$

3. Az elhelyező program felépítése és működése

Az elhelyező programcsomag 51 rutinból áll, ezek alkotják az 1. ábrán látható főbb programrészeket. Az algoritmusok jellege szerint a program két nagyobb egységre bontható.

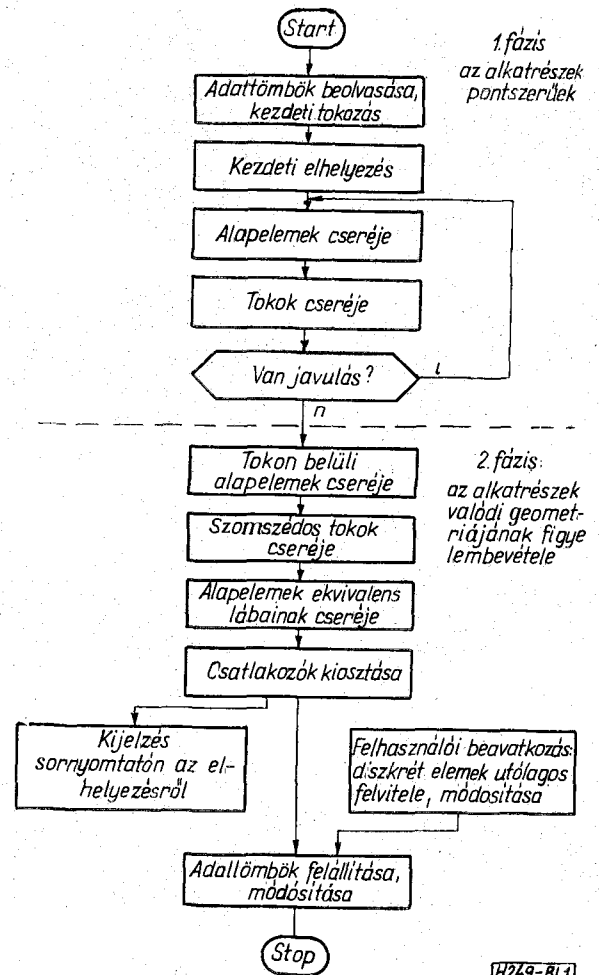
Az első fázisban a tokokat a beválogatott alapelemekkel együtt pontszerűnek tételezzük fel. A tokok — ezen belül az alapelemek — lábairól induló vezeték tehát mind egy pontból — a tokok képzeletbeli geometriai középpontjából — indulnak ki. Ebből következik, hogy az első fázisban a beválogatott alapelemek tokon belüli elhelyezkedése tetszőleges lehet, ezáltal viszonylag gyors elhelyezést kapunk.

A második fázisban a tokok, ill. alapelemek tényleges méreteikkel szerepelnek. A fázis egyes lépései során a program működése fokozatosan lassul, ugyanakkor azonban mind közelebb jutnak az optimális megoldáshoz.

Az alábbiakban az elhelyező program főbb programrészeinek működését ismertetjük.

3.1 Adattömbök beolvasása, kezdeti tokozás

Az ESZTER programrendszer az egyes programok közötti adatcserét egy közös alapadattömb (AAT) segítségével bonyolítja le. Kiinduláskor ebből kell



1. ábra. Az elhelyező program tömbvázlata

kiválasztani és megfelelően átalakítani az elhelyező programok számára szükséges adatokat. Ezután az alapelemeket bizonyos összefüggések és megfontolások alapján tokokba válogatjuk.

Szempontok:

- egy tokba csak azonos típusú és meghatározott számú alapelem kerülhet;
- az azonos típusú alapelemek közül lehetőség szerint azokat tesszük egy tokba, amelyeknek egymás között a legtöbb összekötésük van;
- ha az egymás közti összekötések száma nulla, azokat választjuk egy tokba, amelyek más típusú elemekhez minimálisan kötődnek;
- a felhasználó által előre megadott tokok változatlanok maradnak.

Az így kialakított tokok természetesen nem tekinthetők véglegesnek, hiszen a program következő lépéseiben többször módosulhatnak.

3.2 Kezdeti elhelyezés

Az elhelyezést az előbb kialakított tokok egyenkénti felvitelével végezzük.

Első lépésként kiválasztjuk azt a tokot, amelyik a csatlakozóhoz maximális számú, vezetékkel kapcsolódik, és úgy próbáljuk elhelyezni a lehetséges tok-

helyeken, hogy az összekötő vezetékek összes hossza minimális legyen. Következő elemként rendre azokat a tokokat választjuk ki, amelyek a csatlakozóhoz és a már elhelyezett tokokhoz maximálisan csatlakoznak, és olyan helyeket keresünk számukra, hogy az össz-huzalhossz a lehető legkisebb mértékben növekedjék.

Ha a kiszemelt tokhely már foglalt, akkor egy transzformációs lépés segítségével a korábban idehelyezett tokot megpróbáljuk más, üres tokhelyre áttenni, természetesen a célfüggvény minimumon tartásával.

Megemlítendő, hogy

- a felhasználó által előre megadott tokokat a program csatlakozóként kezeli;
- a koordinátákkal megadott diszkrét elemeket a program ugyancsak csatlakozóknak tekinti;
- az üres koordinátákkal megadott diszkrét elemeket a program figyelmen kívül hagyja;
- a letiltott tokhelyek nem vesznek részt a kezdeti elhelyezésben.

A kezdeti elhelyezés $\frac{m(m+1)(m+2)}{6}$ lépés után fejeződik be.

3.3 A cserélgetést eljárások

3.3.1 Az alapelemek cserélgetése

Elsőként a legkedvezőtlenebb elhelyezkedésű alapelemet választjuk ki, és páros cserékkel a megengedett alapelemhelyek közül abba a pozícióba helyezzük, ahol a célfüggvény értékében a legnagyobb javulás várható. Ezt a műveletet az összes többi, kedvezőtlenebb elhelyezkedésű alapelemre is elvégezzük.

Szemponatok:

- alapelemek csak azonos típusú tokok között cserélhetők;
- a cserében a tokok kihasználatlan alapelemei is résztvesznek;
- miután a tokok és alapelemeik pontszerűek, a tokon belüli helyzet figyelmen kívül hagyható.

Az alapelemek cserélgetése akkor fejeződik be, ha már egyetlen páros csere sem javítja a célfüggvény értékét, vagyis az összhuzalhossz nem csökken.

3.3.2 A tokok cserélgetése

Az eljárás az előző lépésben kialakított tokokkal végzi a páros cseréket, lényegében az előző pontban mondottak szerint. Ha a tokok cserélgetése során a célfüggvény értékében javulás történt, visszatérünk az alapelemek cserélgetéséhez, ellenkező esetben a 3.3.3 lépés következik.

3.3.3 Az alapelemek áthelyezése a tokon belül

Az előző lépések eredményeként kialakítottuk a tokok páros cserékre nézve optimális elhelyezését a kártyán, valamint az alapelemek optimális hozzárendelését a tokokhoz.

Miután eddig a tokokat, ill. a beválogatott alapelemeket pontszerűnek tekintettük, ahol a virtuális huzalok a tokok geometriai középpontjából indulnak ki, szükséges, hogy — a lábak tényleges elhelyezke-

dését figyelembe véve — megtervezzük az alapelemek tokon belüli elhelyezkedését. Ennek megfelelően — a tokokat rögzítettnek tekintve — tokon belüli cserékkel próbáljuk az optimális megoldást megtalálni.

Szemponatok:

- a tokok valódi geometriájának figyelembevétele meghatározott sorrendben történik;
- a felcserélendő alapelemek kiválasztása, ill. elhelyezése csak a megfelelő tok alapelemeinek, ill. elemhelyeinek halmazából lehetséges;
- a rögzített tokok változatlanok maradnak.

Ha az összes tokon végighaladva a célfüggvény értékében javulást értünk el, az utóbbi eljárást megismételjük. Ellenkező esetben áttérünk a következő lépésre.

3.3.4 Az optimum további javítása

Egyes mintapéldák futtatása során kiderült, hogy az egy alapelemet tartalmazó, valamint az aszimmetrikus felépítésű, több alapelemes tokok egymás melletti felhelyezése az eddigi lépésekkel nem vagy csak alig befolyásolható.

Ennek kompenzálására a már belőtt elhelyezett programcsomagba újabb programot iktattunk be, amely — a rögzített tokok kivételével — minden tok felcserélését megkísérli a toklábak sorával párhuzamosan elhelyezkedő két szomszédjával. A valóságban azonban csak azon cseréket végzi el, amelyek a célfüggvény értékét javítják.

Hasonló elv szerint történik az alapelemek ekvivalens (logikailag egyenértékű) lábainak felcserélése is, ami a kereszteződések számának csökkenéséhez vezet.

3.4 Csatlakozók kiosztása

A csatlakozók kiosztását a program az elhelyezés elkészülte után végzi el. Evégett először megkeresi a csatlakozókra kivezető vonalak azon pontját, amely a csatlakozóhoz legközelebb esik, majd ezekhez a pontokhoz hozzárendeli a legközelebbi csatlakozó érintkezőt. A feladat lényegében elsőrendű hozzárendelési probléma, amelynek megoldása a kezdeti elhelyezésnél is használt algoritmussal történik.

3.5 Eredményközlés, javítás

A felhasználót egy grafikus kijelző program tájékoztatja az elhelyezés eredményéről. A kijelzés áttekinthető képet nyújt a kártyán levő tokok helyzetéről, valamint a diszkrét elemek helyéről.

A felhasználó utólagosan megadhatja vagy módosíthatja a diszkrét elemek koordinátáit. Ezeket a program regisztrálja.

3.6 Adattömbök felállítása, módosítása

Az elhelyező program összeállítja a vonalas összekötési listát a fóliatervező program számára. A lista tartalmazza az összekötendő pontok X, Y koordinátáit, valamint a huzalozás szempontjából foglaltak tekintett pontok pontos jegyzékét. Az elhelyező program gondoskodik a kihasználatlan kapuk bemeneteinek, illetve a kapuk kihasználatlan

lábainak jellegük szerinti bekötéséről a föld-, ill. tápfeszültségre.

Ez a program végzi a vonalista szükségessé váló módosítását is az ekvivalens lábak cserélgetése, valamint a felhasználói beavatkozások után.

4. Fejlesztési lehetőségek

Az elhelyező program jelenlegi állapotában a kártyára kerülő diszkrét elemek koordinátáit a felhasználónak kell megadnia. Ez több szempontból nem kedvező (pl. az előre lerögzített diszkrét elemek az elhelyező algoritmusokat gátolhatják az optimális megoldások megtalálásában), ezért célszerűnek látszik a diszkrét elemek elhelyezésének automatizálása. A program így optimális alkatrész-elrendezést adhat vegyes kártyák esetén is. Könnyen belátható, hogy egy 14 vagy 16 lábú tok és egy hozzá kapcsolódó 2–3 lábú diszkrét elem közül a diszkrét elem helyzetének változtatása adhatja a kedvezőbb megoldást.

A vegyes kártyák automatikus megtervezését végző programjaink előreláthatólag 1974 második felétől kezdve lesznek felhasználhatók.

I R O D A L O M

[1] Steinberg, L.: The backboard wiring problem: a placement algorithm. *SIAM Review*, Vol. 3, No. 1. January, 1961. pp. 37—50

- [2] Rutmon, R. A.: An algorithm for placement of interconnected elements based on minimum wire length. *Proc. SJCC, AFIPS*, Vol. 25. 1964. pp. 477—491
- [3] Mamelak, J. S.: The placement of computer logic modules. *Journal of the ACM*, Vol. 13, No. 4. Oct. 1966. pp. 615—629
- [4] Pomentale, T.: An algorithm for minimizing backboard wiring functions. *Comm. ACM*, 8, 1965. pp. 699—703
- [5] Fisk, C. J.—Caskey, D. L.—West, L. E.: ACCEL: Automated circuit card etching layout. *Proc. IEEE* Vol. 55. No. 11. Nov. 1967. pp. 1971—1982
- [6] Leevers, D. F. A.: The use of a graphical display in the automatic design of printed circuit boards. *IEEE Conference publication*, No. 51. April 1969. pp. 11—20
- [7] Nicholson, T. A. J.: A sequential method for discrete optimization problems and its application to the assignment, travelling salesman, and three machine scheduling problems. *Journal of Inst. Maths. Applies* 3, 1967, pp. 362—375
- [8] Garside, R. G.—Nicholson, T. A. J.: Permutation procedure for the backboard-wiring problem. *Proc. IEEE* Vol. 115, No. 1. January 1968. pp. 27—30
- [9] Barthó L.: Nyomatott áramköri modulok automatikus elhelyezése és összekötése. *Számítógéptechnika '71 Konferencia kiadvány*, 1971, Esztergom
- [10] Munkres, J.: Algorithms for the assignment and transportation problems. *SIAM Journal*, Vol. 5. No. 1. March 1957, pp. 32—38
- [11] Gilmore, P. C.: Optimal and suboptimal algorithms for the quadratic assignment problem. *SIAM Journal*, Vol. 10, No. 2. June 1962. pp. 305—313