

Ինֆորմատիկա

УДК 517.19

Գ. Ա. ՄԿՐՏՅԱՆ

**ՄԱՔՍԻՄԱԼ ԵՐԿԱՐՈՒԹՅԱՆ ՍԱՀՄԱՆԱՓԱԿՈՒՄԸ ՀԱՇՎԻ ԱՌՆՈՂ
ԳԼՈՐԱԼ ՈՒՂԵԳԾՄԱՆ ԱԼԳՈՐԻԹՄ**

Ներածություն: Գերանձ ինտեգրալ սխեմաների նախագծման տեխնոլոգիայի զարգացմանը զուգընթաց միացումներում առաջացող հապաղման էֆեկտը կարևորագույններից մեկն է: Հարկ է լինում գրեթե բոլոր կրիտիկական շղթաներին ուժեղացուցիչներ (բուֆեր) ավելացնել՝ հապաղման դեպքում ազդանշանը ընդունելի դարձնելու նպատակով: Իր հերթին՝ յուրաքանչյուր այդպիսի ուժեղացուցիչ ավելացնելը մեծացնում է սխեմայի չափերը կամ՝ դարձնում այն ավելի խիտ, ինչը և բացասաբար է ազդում սխեմայի արդյունավետության վրա: Այդ իսկ պատճառով բուֆերների տեղադրման համար փնտրվում է ոչ խիտ տարածք: Սակայն հաճախ նույնիսկ այդպիսի տարածքներ գտնելն անհնար է դառնում խիտ սխեմաներում, եթե գլոբալ ուղեգծման ալգորիթմը հաշվի չի առնում նման հնարավորությունը:

Ահա թե ինչու նվազագույն քանակով բուֆեր պահանջող գլոբալ ուղեգծումը մեծ հետաքրքրություն է առաջացնում [1-3]: Գլոբալ ուղեգծումը կառուցվում է այնպես, որ հետագայում հնարավոր լինի բուֆերներ ավելացնել այդ շղթաներին: Ներկա հոդվածում դիտարկվում է այդպիսի ուղեգծման կառուցման ալգորիթմ բուֆերների տեղադրման համար հնարավոր ազատ տարածքների նախատեսմամբ: Այն հիմնված է կարճագույն ճանապարհ գտնելու ալգորիթմի վրա, որով փորձ է արվում մինիմալացնել կառուցվող ուղեգծման ծառի գումարային կշիռը: Դա կախված է այն տարածքների խտություններից, որտեղով անցնում է ուղեգծումը, ինչպես նաև բուֆերների զբաղեցրած տիրույթների խտություններից: Վերջում ներկայացվում են փորձարարական արդյունքները և դրանց համեմատությունը այլ ալգորիթմների արդյունքների հետ:

Աշխատանքում օգտագործվել են Շիրվանիի [4] և Կանգի [5] գրքերում սահմանված տերմինները:

Լրացուցիչ սահմանումներ և խնդրի դրվածքը: Խնդիրը տրված ուղեգծման համար լավագույն ձևով բուֆերներ տեղադրելը չէ միայն, այլ նաև այնպիսի գլոբալ ուղեգծման կառուցումը, որի համար գոյություն ունենա հապաղման ընդունելի արժեք ապահովող բուֆերների գոնե մեկ տեղադրում: Աշխատանքում օգտագործվել է հապաղումը գնահատելու արագ և հարմար գնահատական: Ալգորիթմը հնարավոր է հեշտությամբ ձևափոխել ավելի

ճշգրիտ գնահատականներ (օրինակ՝ Էլմորի գնահատականը [6]) ստանալու նպատակով:

Կառուցենք գլոբալ ուղեգծման $G(V, E)$ գրաֆը հետևյալ կերպ:

• Ամբողջ սխեմայի եռաչափ տիրույթը տրոհենք գլոբալ ուղեգծման վանդակների՝

$$CMap = \{c_{ijk} / 0 \leq i < C_{num}, 0 \leq j < R_{num}, 0 \leq k < L_{num}\},$$

որտեղ C_{num} -ը և R_{num} -ը համապատասխանաբար սյունների և տողերի նախօրոք տրված քանակներն են, իսկ L_{num} -ը ուղեգծման շերտերի քանակն է:

• Յուրաքանչյուր վանդակին համապատասխանեցնենք մեկական գազաթ G գրաֆում՝

$$V = \{v_{ijk} / 0 \leq i < C_{num}, 0 \leq j < R_{num}, 0 \leq k < L_{num}\}:$$

• Բոլոր հարևան վանդակների համար G գրաֆում ավելացնենք կողայն վանդակներին համապատասխան գազաթների միջև՝

$$E = \left\{ (v_{i,j,k}, v_{i+1,j,k}) / 0 \leq i < C_{num} - 1, 0 \leq j < R_{num}, 0 \leq k < L_{num} \right\} \cup$$

$$\cup \left\{ (v_{i,j,k}, v_{i,j+1,k}) / 0 \leq i < C_{num}, 0 \leq j < R_{num} - 1, 0 \leq k < L_{num} \right\} \cup$$

$$\cup \left\{ (v_{i,j,k}, v_{i,j,k+1}) / 0 \leq i < C_{num}, 0 \leq j < R_{num}, 0 \leq k < L_{num} - 1 \right\}:$$

$G(V, E)$ գրաֆի ամեն մի $e = (v_{i_1 j_1 k_1}, v_{i_2 j_2 k_2})$ կողին համապատասխանեցնենք արժեքների $\{length_e, C_e, L_e\}$ եռյակ, որտեղ՝

• $length_e$ -ն կողի երկարությունն է, որը հավասար է համապատասխան $c_{i_1 j_1 k_1}$ և $c_{i_2 j_2 k_2}$ վանդակների կենտրոնների միջև եղած մանհետենյան հեռավորությանը (մենք կոդիտարկենք այն դեպքը, երբ կողերի երկարությունները հավասար են միևնույն L_e նախօրոք տրված թվին ($length(e) = L_e, e \in E$));

• C_e -ն կողի թողունակությունն է, որը ցույց է տալիս մաքսիմալ շղթաների քանակը, որոնք կարող են անցնել $c_{i_1 j_1 k_1}$ վանդակից $c_{i_2 j_2 k_2}$ վանդակ;

• L_e -ն կողի զբաղվածությունն է, որը ցույց է տալիս այն շղթաների քանակը, որոնք արդեն անցնում են $c_{i_1 j_1 k_1}$ վանդակից $c_{i_2 j_2 k_2}$ վանդակ:

Սահմանում: e կողի խտություն ասելով կհասկանանք L_e / C_e հարաբերությունը:

Ամեն մի c_{ij0} վանդակում կա որոշակի ազատ տարածք, որը կարող է օգտագործվել բուֆերներ տեղադրելու համար: Այստեղ դիտարկվում է միայն զրոյական շերտը ժամանակակից տեխնիկական հնարավորությունների սահմանափակության պատճառով:

Սահմանում: Տարածքային Խտության Քարտեզը (Area Congestion Map) ճիշտ $C_{num} \times R_{num}$ էլեմենտ պարունակող $ACM = \{ACM_{ij} / 0 \leq i < C_{num}, 0 \leq j < R_{num}\}$ բազմություն է, որի յուրաքանչյուր ACM_{ij} էլեմենտն իրենից

ներկայացնում է արժեքների $ACM_{ij} = \{C_{ij}^{ACM}, L_{ij}^{ACM}\}$ զույգ, որտեղ C_{ij}^{ACM} -ը ցույց է տալիս բուֆերների առավելագույն քանակը, որ հնարավոր է տեղադրել c_{ij0} տիրույթում, իսկ L_{ij}^{ACM} -ը՝ նույն տիրույթում արդեն տեղադրված բուֆերների քանակը:

Մահնանում: ACM_{ij} էլեմենտի խտություն ասելով՝ կհասկանանք $L_{ij}^{ACM} / C_{ij}^{ACM}$ հարաբերությունը:

$G(V, E)$ գրաֆում շղթան ներկայացվում է որպես գրաֆի գագաթների բազմություն՝ $N = \{t_0, t_1, \dots, t_n\} \subset V$, որտեղ առաջին t_0 գագաթը համապատասխանում է սնուցող կոնտակտին (աղբյուրին): t_0, t_1, \dots, t_n գագաթները կանվանենք շղթայի տերմինալներ:

Մահնանում: $T \subset G$ ուղեգծման ծառի [5] շառավիղն է, որը աղբյուրից դեպի ամենահեռու տերմինալը ընկած ճանապարհի երկարությունն է:

Դիցուք $T(V_T, E_T)$ -ն ուղեգծման ծառ է $G(V, E)$ գրաֆում և $B \subseteq V_T$ այն գագաթների բազմությունն է (հետագայում կանվանենք բուֆերների կետեր), որտեղ անհրաժեշտ է տեղադրել բուֆերներ T ծառի համար, որպեսզի այն բավարարի հապաղման վրա դրված սահմանափակումներին:

Մահնանում: $T(V_T, E_T)$ ծառի կշիռ ասելով՝ կհասկանանք հետևյալ գումարը.

$$cost(T(V_T, E_T)) = \sum_{e \in E_T} cost_e(e) + \sum_{v_j \in B} cost_v(v_j),$$

որտեղ $cost_e(e)$ -ն e կողի կշիռն է, որը կախված է այդ կողի խտությունից, իսկ $cost_v(v_j)$ -ն տեղադրված բուֆերի կշիռն է, որը կախված է ACM_{ij} էլեմենտի խտությունից:

Դիցուք $T' \subseteq T$ -ն T ուղեգծման ծառի որևէ ենթածառ է: T' ենթածառի $t_0^{T'}$ արմատը V_T բազմությունից t_0 -ին ամենամոտ ընկած գագաթն է: T' ենթածառի շառավիղը հավասար է $\max_{v \in V_T} dist(v, t_0^{T'})$, որտեղ $dist(v, t_0^{T'})$ -ը T ծառում v և $t_0^{T'}$ գագաթների միջև ընկած ճանապարհի երկարությունն է:

Այսպիսով, խնդիրը հետևյալն է. տրված N շղթայի, G գրաֆի և ACM -ի համար կառուցել նվազագույն կշիռ ունեցող T ուղեգծման ծառ համապատասխան բուֆերների կետերով այնպես, որ այդպիսիք չպարունակող բոլոր առավելագույն $T' \subseteq T$ ենթածառերի շառավիղները լինեն նախօրոք տրված R_{max} արժեքից փոքր:

Առաջարկված ալգորիթմը: Այն հիմնված է գրաֆում երկու գագաթների միջև կարճագույն ճանապարհի գտնելու ալգորիթմի վրա: Մշակվել է գրաֆի գագաթներին միշտ համապատասխանեցնելու հատուկ եղանակ:

Կասենք, որ v գագաթը նշված է, եթե նրան համապատասխանեցված է արժեքների $w_v^i = \{len_v^i, cost_v^i, prev_v^i\}$ եռյակների $W_v \neq \emptyset$ բազմություն: Յուրաքանչյուր եռյակի վերջին $prev_v^i$ արժեքը կամ մատնանշում է մեկ այլ v գագաթ և այդ v գագաթի միշտ պատկանող եռյակ, կամ ունի գրոյական արժեք:

Յուրաքանչյուր W_v նիշ ունի կշիռ, որը հավասար է այդ նիշի էլեմենտներից փոքրագույն կշիռ ունեցող էլեմենտի կշռին:

$w_v^i \in W_v$ եռյակը կանվանենք ավելցուկային, եթե $w_v^i \in W_v^{red} \subseteq W_v$, որտեղ $W_v^{red} = \{w_v^i / w_v^i \in W_v \& ((\exists w_v^j \in W_v, cost_v^j \leq cost_v^i, len_v^j \leq len_v^i) \vee (len_v^i \geq R_{max}))\}$:

Ալգորիթմի աշխատանքի ընթացքում յուրաքանչյուր v գագաթ կարող է լինել հետևյալ վիճակներից մեկում.

- նշված և ղիտարկված,
- նշված և չղիտարկված,
- չնշված և չղիտարկված:

Ալգորիթմի սկզբում փնտրվող T ծառը բաղկացած է միայն մեկ t_0 գագաթից: Այնուհետև, կատարելով հետևյալ քայլերը, T -ին հերթով միացվում են մնացած $\{t_1, \dots, t_n\} \subseteq N$ գագաթները, որի արդյունքում շղթայի համար կառուցվում է վերջնական ուղեգծումը:

Ալգորիթմ:

1. G գրաֆի բոլոր գագաթները չնշված և չղիտարկված են:
2. T ենթածառի բոլոր v գագաթները ստանում են $W_v = \{\{0,0, \{0,0\}\}$

նիշ:

3. Բոլոր նշված և չղիտարկված գագաթներից ընտրվում է փոքրագույն արժեքով նիշ ունեցող որևէ v_{min} գագաթ:

4. v_{min} գագաթի հարևան յուրաքանչյուր v' չղիտարկված գագաթ ստանում է նոր $W_{v'} = W_{v'} \cup W_{v'}^{v_{min}}$ նիշ, որտեղ՝

• $W_{v'}$ -ը v' գագաթի նիշի նախորդ արժեքն է, եթե v' -ը արդեն նշված էր, և դատարկ է՝ հակառակ դեպքում;

• $W_{v'}^{v_{min}}$ -ը նշող էլեմենտների բազմություն է, որն ստացվում է $W_{v_{min}}$ նիշից և $e_{v_{min}v'} = (v_{min}, v')$ կողից հետևյալ կերպ՝

$$W_{v'}^{v_{min}} = \left\{ \left\{ len_{v_{min}}^i + length_{e_{v_{min}v'}}, cost_{v_{min}}^i + cost_e(e_{v_{min}v'}), \{w_{v_{min}}^i, v_{min}\} \right\} / w_{v_{min}}^i \in W_{v_{min}} \right\} \cup \left\{ \left\{ length_{e_{v_{min}v'}}, cost_{v_{min}}^i + cost_e(e_{v_{min}v'}) + cost_v(v_{min}), \{w_{v_{min}}^i, v_{min}\} \right\} / w_{v_{min}}^i \in W_v \right\}:$$

5. $W_{v'}$ -ից հեռացվում են մինիմալ քանակով ավելցուկային էլեմենտներ այնպես, որ $W_{v'}^{red}$ -ը դառնա դատարկ:

6. v_{min} գագաթը ստանում է նշված և ղիտարկված վիճակ:

7. Եթե $v_{min} \notin N$, ապա վերադառնում ենք 3-ին: Հակառակ դեպքում՝ $W_{v_{min}}$ նիշից ընտրվում է փոքրագույն կշիռ պարունակող որևէ $w_{v_{min}}^i$ եռյակ և կառուցվում է v_{min} գագաթը T ծառին միացնող $P_{v_{min}}^i$ ճանապարհ եռյակների նախորդ արժեքի հիման վրա: Կառուցված $P_{v_{min}}^i$ ճանապարհը ավելացվում է T ենթածառին և միացվում է նրան v_{min} տերմինալը:

• Նիշի յուրաքանչյուր էլեմենտի $prev'_v$ կոմպոնենտը կամ ինֆորմացիա է պահում մեկ այլ նիշի որևէ էլեմենտի մասին, կամ ունի գրոյական արժեք: Օգտվելով դրանից՝ յուրաքանչյուր w'_v նիշի համար հնարավոր է կառուցել P'_v ճանապարհ, որը v գագաթը միացնում է T ծառին: Ընդ որում այդ էլեմենտի $cost'_v$ կոմպոնենտը ցույց է տալիս P'_v ճանապարհի արժեքը, իսկ len'_v կոմպոնենտը՝ $T \cup P'_v$ ծառում v գագաթին ամենամոտ բուֆերի կամ սնուցող տերմինալի հեռավորությունը v գագաթից: 4-րդ քայլում v_{min} գագաթը դիտարկելիս նրան կից յուրաքանչյուր v' չդիտարկված գագաթի նիշին նոր էլեմենտներ են ավելացվում, որոնք կառուցվում են v_{min} -ի նիշի էլեմենտներից՝ յուրաքանչյուրից երկուական, հետևյալ կերպ:

1. Ենթադրվում է, որ v_{min} գագաթում բուֆեր չի տեղադրվում (նոր նիշի len'_v -ը հավասար է $len'_{v_{min}}$ գումարած կողի երկարությունը, իսկ $cost'_v$ -ը՝ նախորդի $cost'_{v_{min}}$ գումարած կողի կշիռը):

2. Ենթադրվում է, որ v_{min} գագաթում մեկ բուֆեր է տեղադրվում (նոր նիշի len'_v -ը հավասար է կողի երկարությանը, $cost'_v$ -ը՝ նախորդի $cost'_{v_{min}}$ գումարած v_{min} գագաթում բուֆեր տեղադրելու արժեքը և կողի կշիռը):

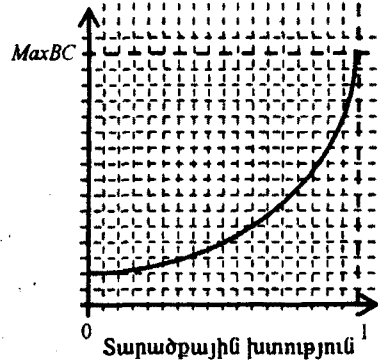
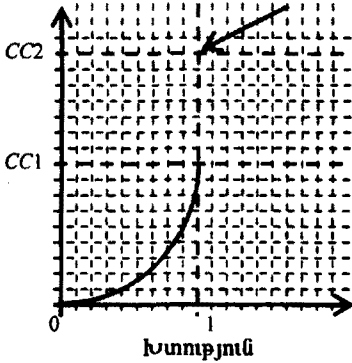
Պնդում 1: $|W_v \setminus W_v^{red}| \leq \lceil R_{max} / L_e \rceil$ բոլոր $v \in V$ -ի համար:

Հիմնավորում: Քանի որ $W_v \setminus W_v^{red}$ -ից կամայական երկու էլեմենտ անհամեմատելի են (հակառակ դեպքում նրանցից մեկը կդառնար ավելցուկային), ապա կարող ենք պնդել, որ եթե $w \in W_v$, $w' \in W_v$ և $w_{len} = w'_{len}$, ապա կամ $w \in W_v^{red}$, կամ $w' \in W_v^{red}$: Այստեղից՝ $w, w' \in W_v \setminus W_v^{red} \Rightarrow w_{len} \neq w'_{len}$: Ուստի $|W_v \setminus W_v^{red}|$ -ը մեծ չէ տարբեր len արժեք ունեցող ճակատային գագաթների քանակից: Եվ քանի որ գրաֆի էջերի երկարությունները հավասար են L_e -ի, ապա $|W_v \setminus W_v^{red}|$ -ը փոքր կամ հավասար է $[0, R_{max}]$ միջակայքում իրարից առնվազն L_e հեռավորության վրա գտնվող կետերի քանակին: Ակնհայտ է, որ այդպիսի կետերի քանակը ոչ ավելին է, քան $\{0, L_e, 2 \times L_e, 3 \times L_e, \dots, K \times L_e\}$, $K \times L_e < R_{max}$ հաջորդական էլեմենտների քանակը: Այսպիսով, ստացանք $|W_v \setminus W_v^{red}| \leq \lceil R_{max} / L_e \rceil$:

Ազդրիքնում օգտագործվել են $cost_e(e)$ և $cost_v(v)$ էլիստիկ ֆունկցիաները: Առաջինը օգտագործվել է շղթայի գրաֆի համապատասխան կողով անցնելու արժեքը գնահատելու համար: Այն կախված է համապատասխան ռեզիոնի խտությունից (նկարի ձախ պատկերը): Այդ կախվածությունը ցույց տվող ֆունկցիան $[0, 1]$ հատվածում գծային չէ (մենք չենք ուզում խոչընդոտել շղթայի ուղեգծումը ոչ խիտ ռեզիոններում), և խտությունը 1-ին ձգտելիս այն ձգտում է նախօրոք տրված CCI -ին: $[1, \infty]$ միջակայքում այն դառնում է գծային և այս դեպքում հնարավոր է խուսափել խիտ և գերխիտ ռեզիոններից:

Երկրորդը վերաբերում է ռեգիոնում բուֆեր տեղադրելու արժեքին: Այն կախված է այդ ռեգիոնի տարածքային խտությունից (նկարի աջ պատկերը): Այդ կախվածությունը ևս գծային չէ, և խտությունը 1-ին ձգտելիս այն ձգտում է նախօրոք տրված $MaxBC$ արժեքին:

Ուղեգծման մեկ խտրացիայի ընթացքում $CC1$, $CC2$ և $MaxBC$ արժեքները չեն փոխվում: Սակայն ուղեգծման ամբողջ պրոցեսում դրանք կարող են վերահաշվարկվել յուրաքանչյուր խտրացիայի համար:



Ալգորիթմում օգտագործվել են հետևյալ ֆունկցիաները.

$$cost_e(v_i, v_j) = ManhattanDist(v_i, v_j) \times (\{L_{(v_i, v_j)} / C_{(v_i, v_j)}\})^2 \times CC1 + \\ + ManhattanDist(v_i, v_j) \times (\{L_{(v_i, v_j)} / C_{(v_i, v_j)}\}) \times CC2,$$

$$cost_s(v_{ij}) = MaxBC \times \max(0.1, (\{L_{ij}^{ACM} / C_{ij}^{ACM}\})^2):$$

Մասնավոր դեպքում, երբ շրթան բաղկացած է երկու կոնտակտից, ալգորիթմն ավարտվում է այն պահին, երբ տերմինալին համապատասխան գազաթը դառնում է նշված և դիտարկված: Քանի որ ալգորիթմը միշտ ընտրում է փոքրագույն կշռով նիշ ունեցող գազաթը և գազաթին նոր նիշ վերագրելիս այդ գազաթի նիշի կշիռը չի փոքրացնում, ապա ակնհայտ է, որ այդպես կառուցված ճանապարհի արժեքը կլինի փոքրագույնը:

Պնդում 2. Ներկայացված ալգորիթմը տալիս է օպտիմալ լուծում երկու կոնտակտներից բաղկացած շրթանների համար:

Այժմ գնահատենք ալգորիթմի բարդությունը: Ալգորիթմը կոնտակտները միացնում է ծառին հերթով՝ ամեն միացման համար կիրառելով նույն բարդության ալգորիթմ: Ուստի նրա բարդությունը $(N_{pin} - 1) \times O_{path}$ է, որտեղ

N_{pin} -ը կոնտակտների քանակն է, իսկ O_{path} -ը՝ մեկ միացում գտնելու բարդությունը: Գործնականում մեկ միացում կատարելու համար ալգորիթմը դիտարկում է գրաֆի գազաթների որոշակի սահմանափակ ենթաբազմությունը: Սակայն վատագույն դեպքում այն կոնտակտների ամբողջ գրաֆը՝ $|V|$ գազաթ: Մեկ գազաթ դիտարկելիս այն կառուցացնի ոչ ավելի քան $12 \times R_{max} / L_e$ նոր էլեմենտ նիշերին ավելացնելու համար: Ամեն մի նորաստեղծ էլեմենտի համար այն պետք է վերահաշվարկի համապատասխան գազաթի տեղը հերթում: Այս քայլի բարդությունը $\log(Q_{num})$ է, որտեղ Q_{num} -ը հերթում էլեմենտների քանակն է, որը չի գերազանցում $|V|$ -ն: Այսպիսով, այդ քայլի բարդությունը

$\log(|V|)$ է, իսկ ամբողջ ալգորիթմինը՝ $|V| \times (N_{pin} - 1) \times (12 \times (R_{max}/L_e) \times \log(|V|))$:

Պնդում 3: N_{pin} կոնտակտ պարունակող շղթայի համար ներկայացված ալգորիթմի բարդությունը $O(|V| \times N_{pin} \times (R_{max}/L_e) \times \log(|V|))$ է:

Գործնականում R_{max} -ը կարող է բավականաչափ մեծ լինել և, հետևաբար, R_{max}/L_e -ն ևս մեծ կլինի: Այդ դեպքում ալգորիթմի արագագործությունը մեծացնելու նպատակով մեկ այլ՝ $W_{R_{max}}$, պարամետր է օգտագործվում, որը սահմանափակում է մեկ զագաթի միջի էլեմենտների քանակը: Երբ այն դառնում է ավելի մեծ, որոշակի էվրիստիկայի հիման վրա ավելորդ էլեմենտները հեռացվում են: Օրինակ, երբ ունենք երկու այդպիսի անհամեմատելի էլեմենտ, որոնց *len* արժեքների տարբերությունը փոքր է, իսկ *cost*-երի տարբերությունը մեծ է, ապա մեծ *cost*-ովը կարող է հեռացվել: Նման ձևով կարելի է նաև հեռացնել համեմատաբար մեծ *len* արժեք ունեցող էլեմենտները:

Գործնականորեն, կիրառելով $W_{R_{max}}$ պարամետրն ու օգտագործելով Look Ahead Cost տեխնիկան [7], ալգորիթմի բարդությունը կարելի է փոքրացնել մինչև $O(|V| \times W_{R_{max}} \times \log(|V| \times W_{R_{max}}))$:

Փորձարարական արդյունքներ: Ներկայացված ալգորիթմը կարող է օգտագործվել սխեմայի բոլոր շղթաները այնպես ուղեգծելու համար, որ հետագայում բոլոր ահնրաժեշտ շղթաների համար հնարավոր լինի բուֆերներ տեղադրել: Մեր դեպքում օգտագործվել է հետևյալ հոսքը:

1. Կառուցել $G(V, E)$ գլոբալ ուղեգծման գրաֆը և *ACM* տարածքային խտության քարտեզը:

2. Կարգավորել շղթաները՝ հաշվի առնելով նրանց կրիտիկականությունը, տերմինալների քանակը և շղթայի տերմինալներն ընդգրկող փոքրագույն ուղղանկյան մակերեսը:

3. Հերթով ուղեգծել շղթաները նկարագրված ալգորիթմով՝ ընդհուպ մինչև գերհագեցում խտության և տարածքային խտության քարտեզներում:

4. Կրկին ուղեգծել բոլոր գերհագեցած ռեգիոններով անցնող շղթաները՝ արգելելով գերհագեցումը այդ քարտեզներում:

Մեզ հաջողվեց գտնել փորձնական սխեմաներ և ուղեգծման գործող ծրագիր միայն *Labyrinth* [8] ուղեգծման ալգորիթմի համար: *Labyrinth* համակարգը հիմնված է կարճագույն ճանապարհի գտնելու ալգորիթմի վրա: Նրա սկզբնական փուլում շղթան տրոհվում է երկու կոնտակտ պարունակող ենթաշղթաների, որոնք հետագայում հերթով ուղեգծվում են:

Ստացված ալգորիթմի արդյունքները հնարավոր չի եղել համեմատել մաքսիմալ երկարության սահմանափակումը բավարարող ուղեգծման այլ ալգորիթմների հետ դրանց բացակայության պատճառով: Քանի որ *Labyrinth* համակարգը բուֆերների հնարավոր դիրքերը հաշվի չի առնում, նրա արդյունքների զգալի մասը պիտանի չէ բուֆերների միջոցով հապաղման խնդիրը լուծելու համար: Համեմայն դեպս, մենք կարողացանք համեմատել այլ էական պարամետրեր, ինչպիսիք են գլոբալ ուղեգծման գումարային

երկարությունը (wirelength), ամենամեծ խտություն ունեցող ռեգիոնի խտությունը (max congestion) և գերհագեցած ռեգիոնների գերհագեցվածությունների գումարը (overflow) (աղյ. 1): Աղյուսակների առաջին սյունը ցույց է տալիս սխեմայի անվանումը, զլորալ ուղեգծման ռեգիոնների քանակը և շղթաների քանակը:

Աղյուսակ 1

Benchmark	Labyrinth				Max Length GR			
	wirelength	max. congestion	overflow	time (sec)	wirelength	max. congestion	overflow	time (sec)
ibm01, 64x64, 11507 nets	73253	1,33333	474	65	68754	1,28571	252	130
ibm02, 80x64, 19291 nets	183196	1,36364	370	133	178964	1,05882	21	380
ibm03, 80x64, 21621 nets	163009	1,2	231	134	161051	1,1	6	301
ibm04, 96x64, 26163 nets	185752	1,45	1147	240	185560	1,3913	1364	526
ibm05, 128x64, 27777 nets	420583	1	0	313	438294	0,928571	0	657
ibm06, 128,64, 33354 nets	317820	1,1	148	379	301386	1,1	30	965
ibm07, 192x64, 44394 nets	227551	1	0	414	245299	0,904732	0	1077
ibm08, 192x64, 47944 nets	91430	0,666667	0	150	104478	0,428571	0	428
ibm09, 256x64, 50393 nets	45950	0.5	0	109	60109	0,357143	0	304
ibm10, 256x64, 72759 nets	651557	1,18519	408	1510	638687	1,18519	198	3432

Աղյուսակ 2-ը համեմատում է խտության քարտեզների որոշ պարամետրեր՝ գերհագեցած վանդակների տոկոսը, 20%-ից ավելի գերհագեցում ունեցող վանդակների տոկոսը և 50%-ից ավելի գերհագեցում ունեցող վանդակների տոկոսը:

Աղյուսակ 2

Benchmark	Labyrinth			Max Length GR		
	>0%	>20%	>50%	>0%	>20%	>50%
ibm01, 64x64, 11507 nets	4,4891%	8,94E-34%	8,92E-34%	2,9638%	6,73E-34%	0,00E+00%
ibm02, 80x64, 19291 nets	2,6149%	6,32E-34%	1,23E-33%	0,2575%	2,80E-45%	5,21E-33%
ibm03, 80x64, 21621 nets	1,8324%	7,04E-34%	1,78E-33%	0,1189%	3,92E-44%	1,29E-33%
ibm04, 96x64, 26163 nets	4,8813%	6,23E-34%	2,40E-33%	5,5491%	8,72E-33%	0,00E+00%
ibm05, 128x64, 27777 nets	0,0000%	4,20E-44%	6,72E-33%	0,0000%	1,11E-43%	1,81E-33%
ibm06, 128,64, 33354 nets	0,8090%	2,54E-33%	4,54E-34%	0,2100%	7,63E-33%	3,14E-33%
ibm07, 192x64, 44394 nets	0,0000%	5,91E-33%	5,91E-33%	0,0000%	3,16E-33%	3,60E-33%
ibm08, 192x64, 47944 nets	0,0000%	3,07E-33%	3,09E-33%	0,0000%	4,69E-32%	2,23E-33%
ibm09, 256x64, 50393 nets	0,0000%	2,34E-33%	1,89E-43%	0,0000%	2,55E-33%	4,54E-34%
ibm10, 256x64, 72759 nets	1,0047%	1,92E-32%	6,48E-32%	0,2096%	1,19E-31%	2,28E-43%

Երկու ծրագրային փաթեթները համեմատվել են Intel® Xeon™ 2,4GHz մեքենաների օգնությամբ Linux համակարգում:

Դիսկրետ մաթեմատիկայի ամբիոն

Ստացվել է 31.07.2005

ԳՐԱԿԱՆՈՒԹՅԱՆ

1. **Minghorng L., Wong D.F.** Maze Routing with Buffer Insertion and Wiresizing, DAC, Los Angeles, CA, 2000, p. 374–378.
2. **Alpert Ch., Kahng A.B., Bao Liu, Ion Mandoulis, and Zellkovsky A.** Minimum-Buffered Routing of Non-Critical Nets for Slew Rate and Reliability Control. Proc. IEEE-ACM Intl. Conf. on Computer-Aided Design, San-Jose, CA, 2001, p. 408–415.
3. **Cong J. and Xin Yuan** Routing Tree Construction Under Fixed Buffer Locations. Proc. IEEE-ACM Intl. Conf. on Computer-Aided Design, Los Angeles, CA, 2000, p. 379–384.
4. **Shervani N.A.** Algorithms for VLSI Physical Design Automation. 3rd edition, Kluwer Academic Publishers, 1999.
5. **Kahng A.B., Robins G.** On Optimal Interconnections for VLSI. Kluwer Academic Publishers, 2001.
6. **Elmore W.C.** – J. Appl. Phys., 1948, v. 19, p. 55–63.
7. **Mkrtchyan G.A.** – Information Technologies and Management, 2005, № 2.
8. **Kastner R., Bozozgadeh E. and Sarrafzadeh M.** An exact algorithm for coupling-free routing. Proc. Int. Symp. on Physical Design, Sonoma, CA, 2001, p. 10–15.

Г. А. МКРТЧЯН

ГЛОБАЛЬНАЯ ТРАССИРОВКА, УЧИТЫВАЮЩАЯ ОГРАНИЧЕНИЯ МАКСИМАЛЬНОЙ ДЛИНЫ

Резюме

Для достижения приемлемой задержки во время проектирования быстродействующих схем в каждую критическую цепь вставляются усилители, которые влияют на плотность размещения глобальной трассировки, порой делая их непригодными. Данная статья рассматривает проблему построения глобальной трассировки, которая учитывает возможные места для усилителей в схемах с повышенной плотностью.

G. A. MKRTCHYAN

MAX LENGTH DRIVEN GLOBAL ROUTING

Summary

In order to maintain signal integrity and reasonable delay in high frequent designs, each critical net must undergo buffer insertion. At the same time a big number of nets in such designs create restrictions in terms of algorithms' run time. This paper studies the problem of constructing minimum cost buffered routing trees in high utilized and congested designs with possible buffer locations.