ԵՐԵՎԱՆԻ ՊԵՏԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

Մինասյան Վահագն Գագիկի

ԱՌՑԱՆՑ ԱԼԳՈՐԻԹՄԱԿԱՆ ՄՈԴԵԼՆԵՐ ՏՎՅԱԼՆԵՐԻ ՄԱՔՍԻՄԱԼ ԿԱՆՈՆԱՎՈՐՎԱԾ ԵՆԹԱԿԱՌՈՒՑՎԱԾՔԻ ԽՆԴԻՐՆԵՐՈՒՄ

Ա.01.09 «Մաթեմատիկական կիբեռնետիկա և մաթեմատիկական տրամաբանություն»
մասնագիտությամբ
ֆիզիկամաթեմատիկական գիտությունների թեկնածուի
գիտական աստիճանի հայցման ատենախոսության

Ս Ե Ղ Մ Ա Գ Ի Ր

Երևան-2011

ЕРЕВАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Минасяан Ваагн Гагикович

МОДЕЛИ ОНЛАЙН АЛГОРИТМОВ В ЗАДАЧАХ МАКСИМАЛЬНОЙ КАНОНИЗИРОВАННОЙ ПОДСТРУКТУРЫ ДАННЫХ

А В Т О Р Е Ф Е Р А Т

диссертации на соискание ученой степени кандидата
физико-математических наук по специальности
01.01.09 – «Математическая кибернетика и математическая логика»

Ереван-2011

Ատենախոսության թեման հաստատվել է Երևանի պետական համալսարանում

Գիտական ղեկավար՝	ֆիզ. մաթ. գիտ. դոկտոր	Լ. Հ. Ասլանյան
Պաշտոնական ընդդիմախոսներ՝	տեխ. գիտ. դոկտոր	Գ. Հ. Խաչատրյան
	ֆիզ. մաթ. գիտ. թեկնածու	Հ. Ց. Հակոբյան
Առաջատար կազմակերպություն՝	ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ	

Պաշտպանությունը կայանալու է 2011 թ. Հունիսի 14-ին ժամը $14^{\underline{00}}$-ին ԵՊՀ-ում գործող ԲՈՀ-ի 044 «Մաթեմատիկական կիբեռնետիկա և մաթեմատիկական տրամաբանություն» մասնագիտական խորհրդի նիստում, հետևյալ հասցեով՝ 0025, Ալեք Մանուկյան 1:

Ատենախոսությանը կարելի է ծանոթանալ ԵՊՀ-ի գրադարանում:

Սեղմագիրն առաքված է 2011 թ. մայիսի 14-ին:

Մասնագիտական խորհրդի գիտական քարտուղար՝
ֆիզ. մաթ. գիտ. թեկնածու Վ. Ժ. Դումանյան

Тема диссертации утверждена в Ереванском государственном университете

Научный руководитель:	доктор физ.-мат. наук	Л. А. Асланян
Официальные оппоненты:	доктор тех. наук	Г. Г. Хачатрян
	кандидат физ.-мат. наук	Г. Ц. Акопян
Ведущая организация:	Институт проблем информатики и автоматизации НАН РА	

Защита состоится 14-го июня 2011 г. в $14^{\underline{00}}$ на заседании действующего в ЕГУ специализированного совета ВАК 044 «Математическая кибернетика и математическая логика» по адресу: 0025, г Ереван, ул. Алека Манукяна, 1.

С диссертацией можно ознакомиться в библиотеке ЕГУ.

Автореферат разослан 14-го мая 2011 г.

Ученый секретарь специализированного совета,
кандидат физ.-мат. наук В. Ж. Думанян

ԱՇԽԱՏԱՆՔԻ ԸՆԴՀԱՆՈՒՐ ԲՆՈՒԹԱԳԻՐԸ

Թեմայի արդիականությունը: Ատենախոսությունը նվիրված է երկու տարբեր կոմբինատոր խնդիրների համար առցանց ալգորիթմներ կառուցելուն: Դրանք են երկկողմանի գրաֆի մաքսիմալ անկախ բազմություն գտնելու, և երկու սիմվոլային հաջորդականությունների ամենաերկար ընդհանուր ենթահաջորդականություն գտնելու խնդիրները: Այս երկու խնդիրները հայտնի կոմբինատոր խնդիրներ են, և դրանցից յուրաքանչյուրը համարվում է ոլորտի դասական խնդիրներից մեկը: Գրականության մեջ երկու խնդիրներն էլ հանգամանալիորեն ուսումնասիրված են, սակայն մուտքային տվյալների առցանց ներկայացման պարագայում այդ խնդիրների համար արդյունավետ ալգորիթմներ կառուցելու հետ կապված հարցերը դեռ բավարար լուծում չունեն: Ատենախոսության մեջ նշված խնդիրները դիտարկվում են հենց այս համատեքստում, և ատենախոսության արդիականությունը պայմանավորված է առցանց ալգորիթմների նկատմամբ հետաքրքրությամբ, որն իր հերթին, բացի տեսական նկատառումներից, պայմանավորված է նաև նշված երկու խնդիրներին վերաբերվող բազմաթիվ կրառություններով՝ կապված դրանցում տվյալների հոսքերի ի հայտ գալու հետ:

Առցանց (online) են համարվում այն ալգորիթմները, որոնք կարող են մշակել մուտքային տվյալները մաս առ մաս՝ հաջորդականորեն ստանալով այդ մասերը և դրան զուգընթաց մուտքային տվյալների արդեն մշակված մասին համապատասխան ելք տրամադրելով: Ի տարբերություն դրա՝ սովորական (կամ offline) են համարվում այն ալգորիթմները, որոնք ենթադրում են, որ մուտքային տվյալները տրված են ամբողջությաբ՝ ալգորիթմի աշխատանքի հենց սկզբից: Այսպիսով՝ առցանց ալգորիթմները բաղկացած են հաջորդական իտերացիաներից, որոնցից հերթականը մշակում է մուտքային տվյալների հերթական մասը: Առցանց ալգորիթմների հետազոտման անհրաժեշտությունը պատմականորեն կապված է տարբեր կիրառություններում տվյալների հոսքերի ի հայտ գալու հետ: Տվյալների հոսքեր մշակող ալգորիթմների ուսումնասիրումը որպես ինտենսիվ հետազոտական տիրույթ ձևավորվել է բոլորովին վերջերս: Ոլորտի նախատիպը կարող է համարվել այն, որ դեռ վաղուց, մասնավորապես ավտոմատների տեսության տիրույթի շրջանակներում, իրականացվել են տվյալների մեկանգամյա կամ բազմակի ընթերցումով հաշվարկներ: Հայտնի է նաև Յու. Ժուռավլյովի դասական լոկալ ալգորիթմների մոդելը, որը լոկալ դիտարկումների միջոցով նպատակ ունի իրականացնել տվյալների գլոբալ հատկությունների արդյունավետ վերլուծում[1]: Ընդհանուր ալգորիթմական ոլորտը դանդաղ է ծավալվել, և սկզբում տվյալների մշակումը եղել է պարզունակ և սահմանափակ:

Այսօր նշվում է օրինակ, որ պատմականորեն, մինչև 2003 թվականը մարդկությունը կուտակել է շուրջ $5 \cdot 10^{18}$ (five exabyte) ինֆորմացիա, այն պարագայում, երբ ներկայումս այդ նույն ծավալի ինֆորմացիա արտադրվում է յուրաքանչյուր երկու օրվա ընթացքում: Պարզ է,

[1] Ю. И. Журавлев. Избранные научные труды. М., 1998.

որ սովորական պայմանների համար մշակված ալգորթմները կարող են բավարար արդյունավետ չլինել տվյալների նման ծավալների դեպքում:

Դասական տվյալների հոսքը գործ ունի մեծածավալ տվյալների հաջորդականությունների հետ, որոնք մշակվում են տվյալների վրայով p անցումների (passes) միջոցով և սահմանափակ s ծավալի օպերատիվ հիշողության օգտագործմամբ: Պարզ է, որ ցանկալի է, որ իրականացվի միայն մեկ անցում, և որ հիշողության ծավալը լինի բազմանդամային լոգարիթմիկ (polylogarithmic), սակայն սա ալգորիթմի նկատմամբ խիստ պահանջ չէ և սոսկ ընդհանուր ցանկություն է: Ուսումնասիրությունների սկզբնական փուլում որոշ խնդիրների համար պարզվել է, թե p և s պարամետրերի որ արժեքների դեպքում են այդ խնդիրները լուծելի (կամ մոտարկելի): Այդ սկզբնական աշխատանքներում դիտարկվել են մասնավորապես համակարգչային ժապավենի վրա գրված տվյալների ընտրման և դրանց կարգավորման խնդիրները, և օրինակ՝ ցույց է տրվել, որ n երկարության հաջորդականության մեդիանը հաշվելիս, երբ օգտագործվում են երկու անցումներ, անհրաժեշտ է առնվազն $\Omega(\sqrt{n})$ հիշողություն, և որ $O(\sqrt{n} \cdot \log n)$ հիշողությունը բավարար է: Դասական տվյալների հոսքերից սովորաբար առանձնացնում են երեք մոդելներ. ժամանակային շարքեր (time series), որտեղ հոսքը կազմված է ինչ որ հաջորդական չափումներով տրվող արժեքներից, դրամային ռեգիստորներ (cash register), որտեղ հոսքի հերթական տարրը դրական թիվ է և բնութագրում է համապատասխան դրամային ռեգիստորի աճը, և տուրնիկետներ (turnsite), որոնք դրամային ռեգիստորներից տարբերվում են նրանով, որ հոսքի տարրերը կարող են լինել նաև բացասական թվեր:

Իննսունականների ընթացքում ձևավորվեցին արդյունավետ լուծումներ՝ կապված ժամանակյին շարքերի մոդելում տվյալների վիճակագրական վերլուծությունների հետ: Օգտագործվող հիշողության ծավալի բազմանդամային լոգարիթմիկ սահմանափակման պարագայում այս տիպի շատ խնդիրների համար հնարավոր չէ առաջադրել պահանջվող վիճակագրական բնութագրի ճշգրիտ արժեքը տրամադրող առցանց ալգորիթմ: Փոխարենը շատ վիճակագրական բնութագրերի համար առաջադրվել են նշված սահմանափակման մեջ տեղավորվող և պահանջվող վիճակագրական բնութագիրը տրված ճշտությամբ մոտարկող դետերմինացված, ինչպես նաև հավանականային ալգորիթմներ: Այդպիսի վիճակագրական բնութագրերից են հաճախականային մոմենտները, էնտրոպիան, քվանտիլները (quantiles), ինվերսիաների քանակը, և այլն: Որոշ կառուցվածքային բնութագրերի մոտարկման համար նույնպես առաջադրվել են արդյունավետ ալգորիթմներ, օրինակ՝ հայտնի է թվային հաջորդականության ամենաերկար աճող ենթահաջորդականության երկարությունը տրված $\epsilon > 0$ ճշտությամբ մոտարկող առցանց ալգորիթմ, որն օգտագործում է $O\left(\frac{1}{\epsilon} \cdot \sqrt{n}\right)$ հիշողություն[2], սակայն ավելի բարդ կառուցվածքային բնութագրերի համար, ինչպիսին օրինակ երկու սիմվոլային հաջորդականությունների ամենաերկար ընդհանուր ենթահաջորդականության երկարությունն է, նմանատիպ ալգորիթմներ հայտնի չեն:

[2] P. Gopalan, T. S. Jayram, R. Krauthgamer, R. Kumar. Estimating the sortedness of adata stream. In Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 318–327, 2007.

Ընդհանուր առմամբ՝ մուտքային տվյալների առցանց ներկայացման մոդելում կառուցվածքային բնութագրեր որոշելն էապես ավելի բարդ է վիճակագրական բնութագրեր որոշելուց:

Ի տարբերություն վիճակագրական բնութագրերի հետ կապված խնդիրների, որոնց համար հայտնի են թվով $O(1)$ անցումներ կատարող և բազմանդամային լոգարիթմիկ հիշողություն օգտագործող մոտավոր (կամ հավանականային մոտավոր) ալգորիթմներ, մուտքային տվյալների կառուցվածքային բնութագրերի հետ կապված շատ խնդիրնեի համար, ինչպես օրինակ գրաֆների հետ կապված խնդիրներն են, այդպիսի ալգորիթմներ հայտնի չեն: Այս կապակցությամբ ներմուծվել են համարյա հոսքային (semi-streaming) մոդելներ, որտեղ ավելի լայն դասի խնդիրներ (spanners, matching, diameter estimation) կարող են լուծվել մեկ կամ մի քանի անցումներով, սակայն այս մոդելներում աշխատանքային հիշողության սահմանափակումը $O(n \cdot \text{polylog}(n))$ կարգի է, ինչը գրաֆների հետ կապված խնդիրների պարագայում թույլ է տալիս պահել գագաթները և նրանց հետ կապված որոշ ինֆորմացիա, բայց ոչ բոլոր կողերը:

Աշխատանքի նպատակն ու խնդիրները: Ատենախոսության հիմնական նպատակը արդյունավետ առցանց ալգորիթմներ կառուցելն է: Ատենախոսության մեջ ուսումնասիրման են ենթարկվել երկկողմանի գրաֆների մաքսիմալ անկախ բազմության, թվային հաջորդականության ամենաերկար աճող ենթահաջորդականության և երկու սիմվոլային հաջորդականությունների ամենաերկար ընդհանուր ենթահաջորդականության կառուցման խնդիրները: Նպատակն է մուտքային տվյալների առցանց ներկայացման պարագայում լուծել այդ խնդիրները և ուսումնասիրել հարակից կոմբինատոր կառուցվածքները:

Հետազոտության մեթոդները: Ատենախոսության մեջ կիրառվել են կոմբինատոր անալիզի, գրաֆների տեսության դասական մոտեցումները և հանրահաշվական մեթոդներ որոնք առնչվում են մասնակի կարգերի և կավարների տեսության հետ:

Արդյունքների նորությունը: Ատենախոսության մեջ ստացված բոլոր արդյունքները նոր են: Այդ թվում՝ մուտքային տվյալների հաջորդական մշակմամբ ալգորիթմներ՝ երկկողմանի գրաֆի գագաթների մաքսիմալ անկախ բազմության կառուցման, երկու սիմվոլային հաջորդականությունների ամենաերկար ընդհանուր ենթահաջորդականության կառուցման, և այդ ալգորիթմի զուգահեռ իրականացման եղանակներ:

Տեսական և կիրառական նշանակությունը: Ատենախոսության տեսական նշանակությունը կայանում է դիտարկվող ալգորիթմական խնդիրների լուծման մոդելների կազմման, առաջացող կառուցվածքների ուսումնասիրման և դրանց՝ վերջնական ալգորիթների տեսքով կապակցման մեջ: Կիրառական նշանակությունը կապված է գլոբալ ցանցերի, դիտարկման սենսորային համակարգերի, բանկային ֆինանսական հոսքերի կառավարման, ինչպես նաև հաշվողական գենոմիկայի խնդիրներում կիրառությունների հետ:

Ստացված արդյունքների ապրոբացիան: Ատենախոսության հիմնական արդյունքները զեկուցվել են ԵՊՀ Դիսկրետ մաթեմատիկայի և տեսական ինֆորմատիկայի ամբիոնի սեմինարում: Զեկուցվել են նաև Ինֆորմատիկայի և կիրառական մաթեմատիկայի ֆակուլտետի ընդհանուր սեմինարում, ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի (ԻԱՊԻ) գիտական սեմինարում, ՀՀ ԳԱԱ ԻԱՊԻ և Մոլեկուլյար կենսաբանության ինստիտուտների համատեղ՝ Ինֆորմացիոն կենսաբանության լաբորատորիայի սեմինարում: Ատենախոսության որոշ մասը ներկայացվել է «Կոմպյուտերային գիտություններ և ինֆորմացիոն տեխնոլոգիաներ» CSIT-09 միջազգային գիտաժողովում: Ատենախոսության հիմնական արդյունքները ընդգրկված են հրատարակված երեք աշխատություններում:

Աշխատանքի կառուցվածքն ու ծավալը: Ատենախոսությունը բաղկացած է առաջաբանից երեք գլուխներից, օգտագործված գրականության ցանկից (36 անուն) և եզրակացությունից: Ատենախոսության ծավալը 93 էջ է:

ԱՇԽԱՏԱՆՔԻ ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆԸ

Ատենախոսությունը բաղկացած է առաջաբանից, երեք գլուխներից, գրականության ցանկից և եզրակացությունից: Առաջին գլխում սահմանվում են հիմնական հասկացությունները, ներմուծվում է մաքսիմալ կանոնավորված (բնութագրական) ենթակառուցվածք գտնելու խնդրի ընդհանուր մոդելը, և բերվում են դիտարկվող երկու խնդիրները քննարկելու համար անհրաժեշտ տեղեկություններ: Երկրորդ գլուխը նվիրված է այդ խնդիրներից առաջինին՝ երկկողմանի գրաֆի մաքսիմալ անկախ բազմություն գտնող առցանց ալգորիթմ կառուցելուն: Երրորդ գլուխը նվիրված է քննարկվող երկրորդ խնդրին՝ երկու հաջորդականությունների ամենաերկար ընդհանուր ենթահաջորդականություն գտնող առցանց ալգորիթմ կառուցելուն: Երկրորդ և երրորդ գլուխների վերջին պարագրաֆները նվիրված են համապատասխան խնդիրների վերաբերյալ ստացված արդյունքների ամփոփմանն ու համառոտ վերլուծությանը:

ԳԼՈՒԽ 1-ում ներմուծվում են սույն ատենախոսությանը և տարվող հետազոտություններին վերաբերվող հիմնական հասկացությունները, և բերվում են անհրաժեշտ տեղեկություններ աշխատանքի հիմնական հետազոտությունների իրականացման համար:

Պարագրաֆ 1.1-ում նկարագրվում է առցանց ալգորիթմի հասկացությունը, և դրա համատեքստում առաջադրվում է մաքսիմալ կանոնավորված ենթակառուցվածք գտնելու խնդրի ընդհանուր մոդելը:

Պարագրաֆ 1.2-ում բերվում են երկկողմանի գրաֆի մաքսիմալ անկախ բազմություն գտնելու խնդիրը երկրորդ գլխում քննարկելու համար անհրաժեշտ տեղեկություններ:

Պարագրաֆ 1.3-ում բերվում են երկու հաջորդականությունների ամենաերկար ընդհանուր ենթահաջորդականություն գտնելու խնդիրը երրորդ գլխում քննարկելու համար անհրաժեշտ տեղեկություններ:

ԳԼՈՒԽ 2-ում ուսումնասիրվում է երկկողմանի գրաֆի մաքսիմալ անկախ բազմությունների բազմության կառուցվածքը, ինչը բացահայտվում է մաքսիմալ անկախ բազմություն կառուցող առցանց ալգորիթմի մշակման ընթացքում: Այստեղ առաջադրվում են այդ կառուցվածքի կոնկրետ տարրերի կառուցման, ինչպես նաև ամողջ կառուցվածքի նկարագրման արդյունավետ ալգորիթմներ:

Պարագրաֆ 2.1-ը մաքսիմալ կանոնավորված ենթակառուցվածքի (Պարագրաֆ 1.1-ում առաջադրված) մոդելի սահմաններում դիտարկում է երկկողմանի գրաֆի մաքսիմալ անկախ բազմություն գտնող առցանց ալգորիթմի կառուցման խնդիրը: Այս համատեքստում առաջ է գալիս ավելի լայն, մաքսիմալ անկախ բազմությունների բազմության կառուցվածքի նկարագրման խնդիրը, որը և ուսումնասիրվում է: Դիցուք $G = (U, V, E)$ երկկողմանի գրաֆ է, և $\mathcal{M}$-ը նրա մաքսիմալ անկախ բազմությունների բազմությունն է: Նշանակենք $W = U \cup V$, և

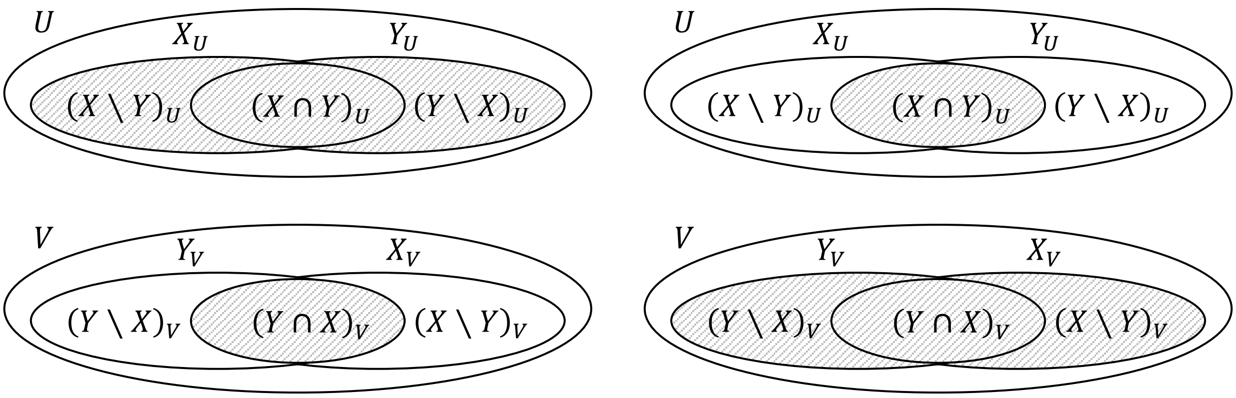
$$\text{կամայական } R \subseteq W \text{ համար՝ } R_U = R \cap U \text{ և } R_V = R \cap V: \qquad (2.1.1)$$

Նաև կատարենք հետևյալ նշանակումները՝

$$X, Y \in \mathcal{M} \text{ համար՝ } X \vee Y = (X_U \cup Y_U) \cup (X_V \cap Y_V), \qquad (2.1.4)$$

$$X, Y \in \mathcal{M} \text{ համար՝ } X \wedge Y = (X_U \cap Y_U) \cup (X_V \cup Y_V): \qquad (2.1.5)$$

Նկար 2.1.1-ում պատկերված են G գրաֆը, և նրա ինչ որ X և Y մաքսիմալ անկախ բազմություններ: Նկարի ձախ կողմում նշագծված մասը համապատասխանում է $X \vee Y$



Նկար 2.1.1

բազմությանը, իսկ աջ կողմում՝ $X \wedge Y$ բազմությանը: Այս պարագրաֆում ցույց է տրվում, որ (2.1.4)-ով և (2.1.5)-ով սահմանվող գործողությունները փակ են $\mathcal{M}$-ի նկատմամբ, ինչպես նաև՝ ձևակերպվում և ապացուցվում է հետևյալ թեորեմը:

Թեորեմ 2.1.5

$(\mathcal{M}, \vee, \wedge)$ հանրահաշվական համակարգը բաշխական կավար է:

$(\mathcal{M}, \vee, \wedge)$ կավարում մաքսիմալ անկախ բազմությունների կազը կնշենք «բարձր» և «ցածր» բառերով:

Պարագրաֆ 2.2-ում հետազոտվում է $(\mathcal{M}, \vee, \wedge)$ կավարի կառուցվածքը: Հայտնի է, որ վերջավոր բաշխական կավարներում կամայական տարր փոխմիարժեքորեն ներկայացվում է ինչ որ միավորմամբ անբաղադրելի տարրերի անկրճատելի միավորմամբ[3]. այս իմաստով՝ կամայական վերջավոր բաշխական կավար փոխմիարժեքորեն տրվում է իր միավորմամբ անբաղադրելի տարրերի բազմությամբ: Կամայական $S \subseteq U$ գագաթների բազմության համար, եթե գոյություն ունի S-ը պարունակող մաքսիմալ անկախ բազմություն, ապա $\lceil S \rceil$-ով կնշանակենք դրանցից (կավարի իմաստով) ամենացածրը. եթե գոյություն չունի, ապա պայմանականորեն կհամարենք, որ $\lceil S \rceil = \emptyset$: Կատարենք հետևյալ նշանակումը.

$$\mathcal{U} = \{\lceil u \rceil \mid u \in U \text{ և } \lceil u \rceil \neq \emptyset\}: \qquad (2.2.3)$$

Այս պարագրաֆում ձևակերպվում և ապացույցվում է հետևյալ թեորեմը:

Թեորեմ 2.2.2

$\mathcal{U}$-ն $(\mathcal{M}, \vee, \wedge)$-ի միավորմամբ անբաղադրելի տարրերի բազմությունն է:

Այնուհետև ձևակերպվում և ապացուցվում են հետևյալ թեորեմները, որոնց ապացույցներով տրվում են $(\mathcal{M}, \vee, \wedge)$ կավարի կոնկրետ տարրերը կառուցող, ինչպես նաև ամբողջ կավարը նկարագրող ալգոիթմներ:

Թեորեմ 2.2.10

G երկկողմանի գրաֆի համար գոյություն ունի $O(|E|)$ բարդության ալգորիթմ, որը տրված $R \subseteq W$ բազմության և Z մաքսիմալ զուգակցման հիման վրա կառուցում է R-ը պարունակող մաքսիմալ անկախ բազմություն, եթե այդպիսին կա, և եթե չկա, ապա հայտնում է այդ մասին:

Այս թեորեմի ապացույցով տրվող ալգորիթմի բարդությունն ավելի մեծ չէ քան մաքսիմալ զուգակցման հիման վրա որևէ անկախ բազմության կառուցման հայտնի ալգորիթմի բարդությունը:

Թեորեմ 2.2.12

G երկկողմանի գրաֆի համար գոյություն ունի $O(|W| \cdot |E|)$ բարդության ալգորիթմ, որը կառուցում է G-ի բոլոր միավորմամբ անբաղադրելի մաքսիմալ անկախ բազմությունները:

Նկատենք, որ որևէ մքսիմալ անկախ բազմություն գտնող հայտնի լավագույն ալգորիթմի բարդությունը $O\left(\sqrt{|W|} \cdot |E|\right)$ է:

Պարագրաֆ 2.3-ում քննարկվում է երկկողմանի գրաֆի մաքսիմալ անկախ բազմություն գտնող առցանց ալգորիթմի կառուցման խնդիրը: Այդտեղ ձևակերպվում և ապացուցվում է հետևյալ թեորեմը:

---

[3] Г. Биркгоф, Теория структур, М., 1952, 407 стр.

Թեորեմ 2.3.4

G երկկողմանի գրաֆի համար գոյություն ունի ամենաբարձր մաքսիմալ անկախ բազմությունը կառուցող $O(|W|\cdot|E|)$ բարդության առցանց ալգորիթմ:

Այս թեորեմի ապացույցով տրվող ալգորիթմի բարդությունն ավելի փոքր չէ քան երկկողմանի գրաֆի որևէ մաքսիմալ անկախ բազմություն գտնող հայտնի առցանց ալգորիթմի բարդությունը, սակայն ի տարբերություն հայտնի ալգորիթմների, առաջադրվող ալգորիթմի աշխատանքի համար անհրաժեշտ չէ հիշողության մեջ պահել մուտքային երկկողմանի գրաֆն ամբողջությամբ: Բավարար է պահել միայն գրաֆի այն հատվածը, որը ամենաբարձր մաքսիմալ անկախ բազմությունից ավելի բարձր է: Վատագույն դեպքում այդ հատվածը պահելու համար անհրաժեշտ է նույն կարգի հիշողություն, որքան ամբողջ գրաֆը պահելու համար:

Պարագրաֆ 2.4-ում վերլուծվում և ամփոփվում են երկրորդ գլխում ստացված արդյունքները:

ԳԼՈՒԽ 3-ում մաքսիմալ կանոնավորված ենթակառուցվածքի (Պարագրաֆ 1.1-ում առաջադրված) մոդելի սահմաններում առցանց ալգորիթմ առաջադրելու տեսանկյունից դիտարկվում է երկու սիմվոլային հաջորդականությունների ամենաերկար ընդհանուր ենթահաջորդականություն (դրա երկարությունը) գտնելու խնդիրը. այս խնդիրը կանվանենք LCS խնդիր (LCS-length խնդիր):

Պարագրաֆ 3.1-ում բերվում են LCS խնդիրը լուծող ալգորիթմի բարդության համար հայտնի ստորին գնահատականները և LCS խնդիրը լուծող հայտնի ալգորիթմների համառոտ նկարագրությունները: LCS խնդրի համար հայտնի է $\Omega(ns)$ ստորին գնահատականը, որը վերաբերվում է լուծող ծառերի որոշակի մոդելի մեջ տեղավորվող ալգորիթմներին, և որտեղ n-ը մուտքային հաջորդականությունների երկարությունն է, իսկ s-ը՝ դրանցում հանդիպող իրարից տարբեր սիմվոլների քանակը[4]: Եթե հանդես եկող իրարից տարբեր սիմվոլների քանակի վրա սահմանափակում չկա, ապա ստորին գնահատականը $\Omega(n^2)$ է: Հայտնի են LCS խնդիրը լուծող ալգորիթմներ, որոնք չեն տեղավորվում լուծող ծառերի նշված մոդելի մեջ: Դրանցից է «չորս ռուսների» մեթոդի վրա հիմնված մի ալգորիթմ, որի բարդության գնահատականը $O(n^2/\log n)$ է: Այս գնահատականը ասիմպտոտիկորեն լավագույնն է LCS խնդիրը լուծող բոլոր հայտնի ալգորիթմների՝ միայն մուտքային հաջորդականությունների երկարությունից կախված գնահատականների մեջ: Հայտնի են LCS խնդիրը լուծող շատ ալգորիթմներ, որոնց բարդության գնահատականները կախված են մուտքային հաջորդականությունները բնութագրող այլ պարամետրերից, սակայն այդ բոլոր գնահատականները վատագույն դեպքում, ինչպես նաև միջինում, $O(n^2)$-ուց ավելի փոքր չեն: Չնայած այս հանգամանքին՝ կիրառական նշանակություն ունեն հենց այդ ալգորիթմները,

[4] A. Aho, C. Hirschberg, J. Ullman. Bounds on the Complexity of the Longest Common Subsequence Problem. Journal of the Association for Computing Machinery, Vol. 23, No. 1, 1976, pp. 1-12.

մասնավորապես այն պատճառով, որ դրանց գնահատականներում հանդես եկող հաստատունները գործնականում ավելի փոքր են լինում ($1/\log n$)-ից:

Պարագրաֆ 3.2-ում նկարագրվում է առաջադրվող առցանց ալգորիթմը: Դիցուք Σ-ն վերջավոր այբուբեն է, $|\Sigma| \geq 1$, և A-ն ու B-ն Σ-ի վրա որոշված վերջավոր հաջորդականություններ են.

$$A = a_1 \cdots a_i \cdots a_m \in \Sigma^m, m \geq 0, \text{ և } B = b_1 \cdots b_j \cdots b_n \in \Sigma^n, n \geq 0: \tag{3.2.1}$$

l-ով կնշանակենք A-ի և B-ի ամենաերկար ընդհանուր ենթահաջորդականության երկարությունը: Կատարենք նաև հետևյալ նշանակումները.

$$i_k = \min\{p_k \mid \exists (p_r, q_r)_{r=1}^{k} : a_{p_r} = b_{q_r}, 1 \leq r \leq k, \text{ և } 1 \leq p_1 < \cdots < p_k \leq m \text{ և } 1 \leq q1 < \ldots < qk \leq n, 1 \leq k \leq l \text{ համար}, \tag{3.2.2}$$

$$j_k = \min\{q_k \mid \exists (p_r, q_r)_{r=1}^{k} : a_{p_r} = b_{q_r}, 1 \leq r \leq k, \text{ և } 1 \leq p_1 < \cdots < p_k \leq m \text{ և } 1 \leq q1 < \ldots < qk \leq n, 1 \leq k \leq l \text{ համար}: \tag{3.2.3}$$

Սահմանում 3.2.1

Կամայական k-ի համար, $1 \leq k \leq l$, (3.1.2) բանաձևով որոշվող i_k ինդեքսը կանվանենք *B-ի նկատմամբ A-ի k-րդ նշիչ*, և (3.1.3) բանաձևով որոշվող j_k ինդեքսը կանվանենք *A-ի նկատմամբ B-ի k-րդ նշիչ*:

Առաջադրվող առցանց ալգորիթմը հիմնված է հերթական իտերացիայի ժամանակ մուտքային հաջորդականությունների մշակված մասերին համապատասխանող նշիչների նորացման վրա: Դիտարկվում է տվյալների առցանց տրման հետևյալ մոդելը: Դիցուք A-ն և B-ն համապատասխանաբար առաջին և երկրորդ մուտքային հաջորդականությունների՝ ալգորիթմի կողմից մշակված մասերն են, և ալգորիթմը հերթական իտերացիայի ժամանակ ստանում է ինչ որ x սիմվոլ, $x \in \Sigma$, որպես առաջին մուտքային հաջորդականության հաջորդ տարր: Նշանակենք $A' = Ax$, և l'-ով նշանակենք A' և B հաջորդականությունների ամեաերկար ընդհանուր ենթահաջորդականույան երկարությունը: Կամայական k-ի համար, $1 \leq k \leq l'$, i'_k-ով և j'_k-ով նշանակենք համապատասխանաբար B-ի նկատմամբ A'-ի, և A'-ի նկատմամբ B-ի k-րդ նշիչը: Հերթական իտերացիայի ժամանակ ստանալով x տարրը՝ ալգորիթմը $(i_k)_{k=1}^{l}$ և $(j_k)_{k=1}^{l}$ նշիչները նորացնում է համապատասխանաբար $(i'_k)_{k=1}^{l'}$ և $(j'_k)_{k=1}^{l'}$ նշիչների: Ալգորիթմի առանձնահատկությունը կայանում է նրանում, որ այն դիտարկում է միայն նորացման ենթակա նշիչները: Յուրաքանչյուր նորացման փոխմիարժեքորեն համապատասխանում է ինչ որ դոմինանտ համընկնում, ուստի ալգորիթմը կատարում է այնքան այդպիսի դիտարկումներ, որքան դոմինանտ համընկնումներ կան A և B հաջորդականությունների միջև. այդ քանակությունը կնշանակենք d-ով: Այս պարագրաֆի շարադրանքի արդյունքում ձևակերպվում և ապացուցվում է հետևյալ թեորեմը:

Թեորեմ 3.2.20

Առաջադրվող առցանց ալգորիթմը A և B հաջորդականությունների համար LCS խնդիրը լուծում է կատարելով d նորացում:

Պարագրաֆ 3.3-ում նկարագրվում է առաջադրված ալգորիթմի մի բարելավում, և այդ բարելավված ալգորիթմի բնութագրերը համեմատվում են հայտնի առցանց ալգորիթմների համապատասխան բնութագրերի հետ: Առաջադրվող բարելավումը նախնական ալգորիթմից առանձնանում է ավելի արագագործ տվյալների կառուցվածքներ օգտագործելով: Արդյունքում ստացված ալգորիթմի առանձնահատկությունն այն է, որ յուրաքանչյուր նորացում կատարվում է $O(\log w)$ ժամանակում, որտեղ w-ն համակարգչի այն բառերի երկարությունն է, որում պահվում են A և B հաջորդականությունների ինդեքսները: Այս պարագրաֆում ձևակերպվում և ապացուցվում է հետևյալ թեորեմը:

Թեորեմ 3.2.3
Առաջադրված ալգորիթմի բարելավված տարբերակը A և B հաջորդականությունների համար LCS խնդիրը լուծում է կատարելով $O(n + d \log w)$ գործողություն և զբաղեցնելով $\Theta(n + d)$ հիշողություն, իսկ LCS-length խնդիրը լուծելու համար՝ $\Theta(n)$ հիշողություն:

Գործնականում $w \leq 32$, և համարյա բոլոր հաջորդականությունների զույգերի համար $n \leq d$: Սա նշանակում է, որ առաջադրվող բարելավված ալգորիթմի բարդությունը գործնականում $O(d)$ է: Աղյուսակ 3.2.1-ում բերված են հայտնի առցանց ալգորիթմների և առաջադրվող ալգորիթմի բնութագրերի համառոտ նկարագրերը (առաջադրվող ալգորիթմին համապատասխանում է վերջին տողը): Աղյուսակում նաև նշված են ալգորիթմների բարդությունների գնահատականների մեջ հանդես եկող գործակիցները: Դրանք չեն արտահայտում ալգորիթմի հերթական քայլի ժամանակ կատարվող տարրական գործողությունների քանակը, այլ արտահայտում են նշված ալգորիթմների հերթական իտերացիաների բարդությունների մոտավոր հարաբերությունը:

Անվանում	Ժամանակ	Գործակից	Հիշ. LCS	Հիշ. LCS-lenght
Ալգ. 1	$O(m + r\log l)$	1	$\Theta(n + r)$	$\Theta(n)$
Ալգ. 2	$O(lm \log(n/l))$	1	$\Theta(ln)$	$\Theta(ln)$
Ալգ. 3	$O\big(ns + l(n - l)\big)$	2	$\Theta(ns + d)$	$\Theta(ns)$
Ալգ. 4	$O(ns + ds)$	1	$\Theta(ns + d)$	$\Theta(ns)$
Ալգ. 5	$O(n \log s + l(n - l) \log s)$	2	$\Theta(n + d)$	$\Theta(n)$
Ալգ. 6	$O(n \log s + ds \log s)$	1	$\Theta(n + d)$	$\Theta(n)$
Առաջադրվող	$O(n + d)$	20	$\Theta(n + d)$	$\Theta(n)$

Աղյուսակ 3.2.1

Աղյուսակի առաջին տողում r-ով նշանակված է A և B հաջորդականությունների միջև բոլոր համընկնումների քանակը: Պարզ է, որ աղյուսակի առաջին երկու տողերում նշված ալգորիթմների բարդություններն ավելի մեծ են մյուսների բարդություններից, ուստի առաջադրվող ալգորիթմն իմաստ ունի համեմատել միայն 3-ից 6-րդ ալգորիթմների հետ: Նկատենք, որ առաջադրվող ալգորիթմն ավելի նախընտրելի է s-ի մեծ արժեքների դեպքում:

Աղյուսակ 3.2.2-ում ներկայացված է համապատասխան վերլուծությունը: Երրորդ սյունյակում նշված է այն պայմանը, որի բավարարման դեպքում առաջադրվող ալգորիթմն ավելի արագագործ է, չորրորդում՝ հիշողության այն քանակը, որքանով, համեմատած առաջադրվող ալգորիթմի հետ, ավել հիշողություն է զբաղեցնում համապատասխան հայտնի ալգորիթմը:

Անվանում	Ժամանակ	Պայման	Ավել հիշ.	Նշում
Ալգ. 3	$O\big(ns + l(n-l)\big)$	$s > 10 + \frac{1}{n}\big(10d - l(n-l)\big)$	$\Theta(ns)$	$d \le l(n-l)$
Ալգ. 4	$O(ns + ds)$	$s > 20$	$\Theta(ns)$	չկա
Ալգ. 5	$O(n\log s + l(n-l)\log s)$	$s > 2^{10\frac{n+d}{n+l(n-l)}}$	0	$d \le l(n-l)$
Ալգ. 6	$O(n\log s + ds\log s)$	$s > 7$	0	չկա

Աղյուսակ 3.2.2

Աղյուսակի երկրորդ և երրորդ տողերում նշված ալգորիթմները առցանց են միայն այն վերապահումով, որ դրանք մուտքային հաջորդականությունների տարրերը ստանում են զույգերով՝ մեկի հերթական տարրի հետ նաև մյուսի հերթական տարրը: Այդ ալգորիթմների բարդությունները կախված չեն d-ից, սակայն հայտնի է, որ $d \le l(n-l)$:

Պարագրաֆ 3.4-ում քննարկվում են առաջադրված ալգորիթմի զուգահեռ իրականացման հետ կապված հարցերը, և առաջադրվում է LCS խնդիրը լուծող առցանց զուգահեռ ալգորիթմ: Այս պարագրաֆում ձևակերպվում և ապացուցվում է հետևյալ թեորեմը:

Թեորեմ 3.4.6
Առաջադրված ալգորիթմի զուգահեռ իրականացումը A և B հաջորդականությունների համար LCS խնդիրը լուծում է կատարելով $O\left(\frac{\log t}{t}(n + d\log w)\right)$ զուգահեռ գործողություն և զբաղեցնելով $\Theta(n+d)$ հիշողություն, իսկ LCS-length խնդիրը լուծելու համար՝ $\Theta(n)$ հիշողություն, որտեղ t-ն հասանելի հաշվողական պրոցեսորների քանակն է:

LCS խնդիրը լուծող հայտնի շատ ալգորիթմների նկատմամբ առաջադրվող զուգահեռ ալգորիթմն ունի երկու կարևոր առավելություն. առաջինը՝ այն առցանց ալգորիթմ է, այսինքն զուգահեռ են իրականացվում հերթական տարրի մշակման համար կատարվող գործողությունները, և երկրորդ՝ այն աշխատունակ է տրամադրության տակ գտնվող կամայական t քանակի պրոցեսորների պարագայում:

Պարագրաֆ 3.5-ում վերլուծվում և ամփոփվում են երրորդ գլխում ստացված արդյունքները:

ԱՇԽԱՏԱՆՔԻ ՀԻՄՆԱԿԱՆ ԱՐԴՅՈՒՆՔՆԵՐԸ

Ատենախոսության մեջ ստացվել են հետևյալ հիմնական արդյունքները:

1. Հետազոտվել է երկկողմանի գրաֆի մաքսիմալ անկախ բազմությունների բազմության կառուցվածքը, որը բացահայտվում է մաքսիմալ անկախ բազմություն կառուցող առցանց ալգորիթմի մշակման ընթացքում: Ցույց է տրվել, որ այդ կառուցվածքը բաշխական կավար է՝ մաքսիմալ անկախ բազմությունների միավորմամբ և հատմամբ նկարագրվող պարզ գործողությունների նկատմամբ: Առաջադրվել է ալգորիթմ, որը մքսիմալ զուգակցման հիման վրա կառուցում է տրված գագաթները պարունակող կավարի իմաստով ամենացածր մաքսիմալ անկախ բազմությունը (եթե այդպիինը կա). այս ալգորիթմի բարդությունը $O(|E|)$ է, ինչը նույնն է, ինչ որևէ մաքսիմալ անկախ բազմություն կառուցող հայտնի ալգորիթմի բարդությունը: Առաջադրվել է նաև ալգորիթմ, որը կառուցում է բոլոր (կավարի իմաստով) միավորմամբ անբաղադրելի մաքսիմալ անկախ բազմությունները, և ըստ այդմ՝ նկարագրում է բոլոր մաքսիմալ անկախ բազմությունների բազմությունը. այս ալգորիթմի բարդությունը $O(|W| \cdot |E|)$ է, այն պարագայում, երբ որևէ մաքսիմալ անկախ բազմություն կառուցող հայտնի լավագույն ալգորիթմի բարդությունը $O\left(\sqrt{|W|} \cdot |E|\right)$ է: Առաջադրվել է նաև (կավարի իմաստով) ամենաբաձր մաքսիմալ անկախ բազմությունը կառուցող առցանց ալգորիթմ, որի բարդությունը $O(|W| \cdot |E|)$ է, ինչը ավելի մեծ չէ որևէ մաքսիմալ անկախ բազմություն կառուցող հայտնի լավագույն առցանց ալգորիթմի բարդությունից, սակայն առաջադրվող ալգորիթմը, կախված խնդրից, օգտագործում է ավելի քիչ հիշողություն:

2. Հետազոտվել է երկու հաջորդականությունների ամենաերկար ընդհանուր ենթահաջորդականություն գտնող առցանց ալգորիթմ կառուցելու խնդիրը: Առաջադրվել է այդ խնդիրը լուծող $O(n + d \log w)$ բարդության առցանց ալգորիթմ, որտեղ n-ը մուտքային հաջորդականություններից երկարի երկարությունն է, d-ն՝ դրանց միջև դոմինանտ համընկնումների քանակը, w-ն՝ համակարգչի այն բառերի երկարությունը, որում պահվում են մուտքային հաջորդականությունների ինդեքսները: Այս գնահատականը գործնականում $O(d)$ է: Առաջադրվող ալգորիթմի առանձնահատկություններից է այն, որ նրա բարդությունը կախված չէ s-ից՝ մուտքային հաջորդականություններում հանդես եկող իրարից տարբեր սիմվոլների քանակից: Ըստ այդմ՝ առաջադրվող ալգորիթմը նույնանպատակ հայտնի ալգորիթմներից նախընտրելի s-ի համեմատաբար մեծ արժեքների դեպքում: Առաջադրվող ալգորիթմը նմանատիպ հայտնի որոշ ալգորիթմներից ավելի արդյունավետ է արդեն այն դեպքում, երբ $s > 7$:

3. Հետազոտվել են վերը նշված առցանց ալգորիթմի զուգահեռ աշխատանքի հետ կապված խնդիրները, և առաջադրվել է վատագույն դեպքում $O\left(\frac{\log t}{t}(n + d \log w)\right)$ բարդություն ունեցող առցանց զուգահեռ ալգորիթմ, որտեղ t-ն հասանելի հաշվողական պրոցեսորների քանակն է: Առաջադրվող զուգահեռ ալգորիթմի առանձնահատկություններից է այն, որ ի տարբերություն նույն խնդիրը լուծող հայտնի զուգահեռ ալգորիթմների, առաջադրվող զուգահեռ ալգորիթմն առցանց ալգորիթմ է, այսինքն մուտքային

հաջորդականությունների տարրերը դիտարկում է հաջրդաբար, և հերթական տարրի մշակման համար անհրաժեշտ գործողությունները կատարվում են զուգահեռ: Մեկ այլ առանձնահատկություն կայանում է նրանում, որ ի տարբերություն նույն խնդիրը լուծող հայտնի զուգահեռ ալգորիթմների, առաջադրվող ալգորիթմն աշխատունակ է տրամադրության տակ գտնվող կամայական քանակի պրոցեսորների պարագայում, ընդ որում այդ քանակությունը կարող է փոփոխվել ալգորիթմի աշխատանքի ընթացքում:

ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ԹԵՄԱՅՈՎ ՀՐԱՏԱՐԱԿԱԾ ՀԻՄՆԱԿԱՆ ԱՇԽԱՏԱՆՔՆԵՐԻ ՑԱՆԿԸ

1. V. Minasyan, On the length of the longest increasing subsequence of sequence of elements drown from an arbitrary partially ordered domain. Computer Science and Information Technologies (CSIT) conference, 28.09-02.10.2009, Yerevan, pp. 92-94,.
2. V. Minasyan, On the Structure of Maximum Independent Sets in Bipartite Graphs, International Journal “Information Theories and Applications”, Vol. 17, Number 2, 2010, pp. 177-188.
3. V. Minasyan, A new algorithm for the longest common subsequence problem, International Journal “Information Theories and Applications”, Vol. 17, Number 3, 2010, pp. 213-221.

ABSTRACT

Vahagn G. Minasyan

"Models of online algorithms in problems of maximum canonized substructure of data"

This thesis is dedicated to designing online algorithms for two different combinatorial problems. The problems are finding a maximal independent set of a bipartite graph, and finding a longest common subsequence of two strings. Both of these are well known combinatorial problems, and each of them is known through the set of important applications. Both problems are thoroughly discussed in publications, but in the case of online representation of input data, the issues related to designing efficient algorithms for these problems yet do not have a sufficient solution. The thesis addresses problems in this context, and in relevance to the interest in online algorithms. The last, in its turn, is due to not only for theoretical reasons, but also due to appearances of data streams in many applications related to the mentioned two problems.

The main objective of the thesis is to develop efficient online algorithms. The following problems are addressed in the context of designing efficient online algorithms: the problem of finding a maximum independent set of a bipartite graph, the problem of finding a longest increasing subsequence of a sequence of numbers and finding a longest common subsequences of two strings. The purpose is to solve these problems and study related constructions in the case of the online input data discipline.

In theoretical level the thesis is investigating models and data structures of solve these problems, exploring emerging combinatorial relations and integrating them into the final algorithms for mentioned optimization problems. The practical value of dissertation is related to the objectives of global networks, sensor monitoring systems, financial transaction flow analysis, computational genomics, and others.

The thesis contains the following main results.

1. The structure of the set of all maximum independent sets of bipartite graph is studied, which occurs while developing an online algorithm for finding a maximum independent set. It is shown that this structure is a distributive lattice with respect to some simple operations described by union and intersection over maximum independent sets. An algorithm is presented which for a given maximum matching finds the lowest (in terms of the lattice) maximum independent set containing the given vertices (if such exists). The complexity of this algorithm is $O(|E|)$, where E is the set of edges, which is the same as the complexity of the best known algorithm which finds some maximum independent set. Also an algorithm is presented which finds all join-irreducible (in terms of the lattice) maximum independent sets, and by so it describes the set of all maximum independent sets. The complexity of this algorithm is $O(|W| \cdot |E|)$, where W is the set of vertices, whereas the complexity of the best known algorithm which finds some maximum independent set is $O\left(\sqrt{|W|} \cdot |E|\right)$. It is also presented another online algorithm which finds the highest (in terms of the lattice) maximum independent set

in $O(|W| \cdot |E|)$ operations. This bound is not greater then the bound of the best known online algorithm which finds some maximum independent set, however in some cases (depending on the input bipartite graph) the presented algorithm uses less memory than the known one.

2. The problem of constructing an online algorithm for finding a longest common subsequence of two strings is studied. An online algorithm is presented which solves the problem in $O(n + d\log w$ time, where n is the length of the longer input sequence, d is the number of dominant matches between the input sequences and w is the size of computer word holding the indices of input sequences. In practical level the mentioned bound can be treated as $O(d)$. One of the properties of the presented algorithm is that its time bound does not depend on s, which is the count of different symbols occurring in the input sequences. Thus the presented algorithm is more preferable over the known algorithms in the case of relatively large values of s. Particularly the presented one is more efficient then some known algorithms already for $s > 7$:

3. The problems related to the parallel implementation of the above mentioned online algorithm are discussed, and an online parallel algorithm is presented with complexity of $O\left(\frac{\log t}{t}(n + d\log w)\right)$ in the worst case, where t is the number of available processors. One property of the presented parallel algorithm is that unlike the known parallel algorithms for the LCS problem, the presented one is an online algorithm, meaning that it handles sequentially the elements of input sequences and performs parallel computations during each iteration. Another property is that unlike many known parallel algorithms the presented one is able to work with arbitrary and varying number of available processors.

РЕЗЮМЕ

Минасян Ваагн Гагикович

«Модели онлайн алгоритмов в задачах
максимальной канонизированной подструктуры данных»

Диссертация посвящена разработке онлайн алгоритмов для двух различных комбинаторных задач. Это задача нахождения максимального независимого множества двудольного графа и задача нахождения наибольшей общей подпоследовательности для двух заданных последовательностей символов. Обе задачи являются известными комбинаторными задачами, и каждый из них является классической задач в своей области. В литературе обе задачи тщательно обсуждены, но в случае онлайн представления входных данных, вопросы, связанные с проектированием эффективных алгоритмов для этих задач все еще не имеют исчерпывающего решения. Диссертация рассматривает эти задачи именно в этом контексте, и его актуальность связна с интересом к онлайн алгоритмам, что в свою очередь обусловено не только теоретическим интересом, но и тем, что во многих приложениях связанных с упомянутыми задачами появляются потоки данных.

Разработка эффективных онлайн алгоритмов является основной целью исследования диссертации. В частности, мы обсуждаем задачу нахождения максимального независимого множества двудольного графа, задачу нахождения наибольшей возрастающей подпоследовательности числовой последовательности, и задачу нахождения наибольшей общей подпоследовательности двух последовательностей символов. Работа нацелена на решение этих задач и исследование характерных комбинаторных структур в случае онлайн представления входных данных.

Теоретическое значение диссертации состоит в разработке моделей для решениярассматриваемых задач, в изучении возникающих здесь комбинаторных структур и в их объединении в окончательные алгоритмы решения этих задач. Практическое значение диссертации связано с задачами глобальных сетей, систем сенсорного мониторинга, анализа финансовых потоков, вычислительной геномикой и других.

В диссертации получены следующие основные результаты.

1. Изучена структура множества всех максимально независимых множеств вершин двудольного графа, которая возникает при разработке онлайн алгоритмов для нахождения максимального независимого множества графа. Показано, что эта структура является дистрибутивной решеткой относительно простых операций, описываемых объединением и пересечением максимально независимых множеств. Приведен алгоритм который для данного максимального паросочетания и данных вершин находит самое низкое (в терминах решетки) максимальное независимое множество, содержащее эти вершины (если такое существует). Сложность этого алгоритма $O(|E|)$, где E это множество ребер графа. Эта оценка совпадает с оценкой сложности наилучшего известного алгоритма нахождения некоторого максимального

независимого множество в графе. Приведен также алгоритм нахождения всех неразложимых (в терминах решетки) максимально независимых множеств, и таким образом описывающих все множество максимально независимых множеств. Сложность этого алгоритма $O(|W| \cdot |E|)$, где W это множество вершин. Наилучший известный алгоритм, который находит некоторое максимальное независимое множество требует время $O\left(\sqrt{|W|} \cdot |E|\right)$. Также представлен онлайн алгоритм, который находит наивысшее (в терминах решетки) максимальное независимое множество за время $O(|W| \cdot |E|)$. Эта оценка не превосходит сложность лучшего известного онлайн алгоритма, который находит некоторое максимальное независимое множество, однако в некоторых случаях (в зависимости от входного двудолноги графа) представленный алгоритм использует меньшую память.

2. Изучена проблема проектирования онлайн алгоритма для нахождения наибольшей общей подпоследовательности двух символовый последовательностей. Приведен онлайн алгоритм, который решает проблему за $O(n + d \log w)$ время, где n длина наиболее длинной входной последовательности, d число доминирующих совпадений между входными последовательностями и w размер компьютерного слова где хронятся индексы входных последовательностей. На практике упомянутая оценка может быть интерпретирована как $O(d)$. Одним из свойств представленного алгоритма, является то что оценка его времени работы не зависит от s, что есть количество различных символов, входящих во входные последовательности. Таким образом, предложенный алгоритм является предпочтительным по сравнению с известными алгоритмами в случае относительно больших значениях s.

3. Изучена проблема, связанная с параллельной реализации вышеупомянутого онлайн алгоритма. Предложен онлайн параллельный алгоритм со сложностью $O\left(\frac{\log t}{t}(n + d \log w)\right)$ в худшем случае, где t число доступных процессоров. Одним из свойств представленного параллельного алгоритма является то, что в отличие от известных параллельных алгоритмов, представленный паралельный алгоритм работает онлайн, что означает, что он последовательно обрабатывает элементы входных последовательностей и выполняет параллельные вычисления во время каждой итерации. Другим свойством алгоритма является то, что в отличие от многих известных алгоритмов он способен работать с любым переменным количеством доступных процессоров.