

Նշանակումների խնդիրը և դրա լուծման ծրագիրը

Ռուզաննա Մազմանյան

DOI: <https://doi.org/10.58726/27382923-ne2023.1-78>

Հանգուցային բառեր. *գծային ծրագրավորում, ծրագրավորման լեզու, «հունգարական մեթոդ», ալգորիթմ, պարամետր, օպտիմալ լուծում, միջակայք*

Ներածություն

Նշանակումների խնդիրը մաթեմատիկական օպտիմիզացիայի կամ գործողությունների հետազոտման ոլորտի կարևորագույն խնդիրներից է:

Գիտության և արտադրության կառավարման բնագավառում հաճախ անհրաժեշտություն է առաջանում որոշակի քանակի աշխատողների կողմից կատարել նույն քանակի տարբեր տեսակի աշխատանքներ, օրինակ՝ թափուր պաշտոններում հնարավոր թեկնածուների նշանակում, ծրագրավորողների միջև կատարման ենթակա ծրագրային նախագծերի բաշխում և այլն: Նմանատիպ խնդիրները ստացել են նշանակումների խնդիր անվանումը:

Նշանակումների խնդիրը պետք է լուծել այնպես, որ յուրաքանչյուր կատարող (մարդ, մեքենա, գործիք) կատարի միայն մեկ աշխատանք, իսկ յուրաքանչյուր աշխատանք կատարվի միայն մեկ կատարողի կողմից:

Նշանակումների խնդրի լուծման անհրաժեշտ և բավարար պայմանը խնդրի փակ լինելն է, այսինքն, երբ կատարողների թիվը հավասար է աշխատանքների քանակին:

Հոդվածը նվիրված է նշանակումների խնդրի էության և «հունգարական մեթոդ»-ով այդ խնդրի օպտիմալ լուծման ալգորիթմի պարզաբանմանը, ինչպես նաև դրա ծրագրային իրականացմանը արդի C# ծրագրավորման լեզվով:

Ներկայացված աշխատանքի նպատակն է ցույց տալ, թե ինչպես կարելի է բարձր կարգի ժամանակակից ծրագրավորման լեզուների հնարավորությունների օգտագործմամբ ոչ միայն կրճատել խնդրի օպտիմալ լուծման ժամանակը, այլև ծրագրավորման հարուստ գրաֆիկական միջոցների կիրառմամբ խնդրի լուծման քայլերի հաջորդականությունը դարձնել տեսանելի:

Նշանակումների խնդրի դրվածքը

Անհրաժեշտ է կատարել N տարբեր տիպի աշխատանք: Դրանց իրականացման համար պետք է ներգրավել N աշխատող: Յուրաքանչյուր աշխատող պատրաստ է կատարել ցանկացած աշխատանք որոշակի վճարի դիմաց: Ցանկացած աշխատանք պետք է հանձնարարվի մեկ աշ-

խատողի: Պահանջվում է աշխատանքներն աշխատողների միջև բաշխել այնպես, որ բոլոր աշխատանքների կատարման ընդհանուր արժեքը լինի նվազագույն [1, 108-109]:

Նշանակումների խնդրի լուծման համառոտ նկարագրությունը

Գոյություն ունեն գծային ծրագրավորման խնդիրների օպտիմալ լուծման տարբեր մեթոդներ, օրինակ՝ «սիմպլեքս մեթոդ», «չյուղավորումների և սահմանների մեթոդ» և այլն: Մակայն գոյություն ունի նշանակումների խնդրի օպտիմալ լուծման մի մեթոդ, որը հայտնի է «հունգարական մեթոդ» անվամբ:

«Հունգարական մեթոդը» մշակել և հրատարակել է ամերիկացի հայտնի մաթեմատիկոս Հարոլդ Կունը 1955 թվականին: Հեղինակը տվել է «հունգարական մեթոդ» անվանումը՝ պայմանավորված այն հանգամանքով, որ ալգորիթմը մշակելիս նա հիմնվել է երկու հունգարացի մաթեմատիկոսների շատ ավելի վաղ կատարած աշխատանքների վրա:

Նշանակումների խնդիրը «հունգարական մեթոդով» լուծելու համար նախ պետք է կազմել աշխատանքների համար նախատեսված ծախսերի աղյուսակ: Մեթոդի հիմնական էությունը այն է, որ ծախսերի սկզբնական աղյուսակից աստիճանաբար պետք է անցնել դրան համարժեք ոչ բացասական տարրերով և անկախ զրոների համակարգ ունեցող մեկ այլ աղյուսակի այնպես, որ այդ զրոներից ցանկացած երկուսը չպատկանեն նույն տողին կամ նույն սյանը:

«Հունգարական մեթոդով» նշանակումների խնդիրը լուծելու համար նախատեսված ծախսերի աղյուսակը կազմելուց հետո պետք է իրականացնել քայլերի հետևյալ հաջորդականությամբ [1, 110-112; 2, 394-405].

1. Գտնել յուրաքանչյուր տողի փոքրագույն տարրը և տվյալ տողի տարրերից հանել համապատասխան տողի փոքրագույնը:
2. Գտնել յուրաքանչյուր սյան փոքրագույն տարրը և տվյալ սյան տարրերից հանել համապատասխան սյան փոքրագույնը:
3. Աղյուսակում նշել զրոներ պարունակող տողերը և սյուները:
4. Գտնել չնշված տարրերից փոքրագույնը:
5. Չնշված բոլոր տարրերից հանել այդ փոքրագույն տարրը:
6. Նշված բոլոր տողերի և սյուների հատման կետերում գտնվող տարրերին գումարել այդ փոքրագույն տարրը:
7. Ստացված աղյուսակում սկզբում նշել այն զրոները, որոնք տվյալ տողում մեկ հատ են, ընդ որում, եթե տվյալ տողի զրոն պարունակող սյունում կան այլ զրոներ՝ դրանք ջնջել կամ անտեսել (նույն կերպ վարվել նաև սյուների դեպքում):
8. Այս կերպ հեռացնելով կամ անտեսելով ոչ պիտանի զրոները՝

կատացվի վերջնական աղյուսակ, որի յուրաքանչյուր տողում կլինի մեկ չանտեսված գրո, որն էլ ցույց կտա, թե որ աշխատողը որ աշխատանքը պետք է կատարի, ինչը և կլինի խնդրի օպտիմալ լուծումը: Օպտիմալ լուծումը գտնելուց հետո պետք է անցնել կետ 9-ին: Այն դեպքում, երբ վերը նշված ձևափոխությունների արդյունքում աղյուսակում կլինեն տողեր, որոնք չեն պարունակի ճիշտ մեկ հատ չանտեսված գրո, կնշանակի, որ խնդրի օպտիմալ լուծումը գտնված չէ, և, սկսած 1-ին կետից, քայլերի հաջորդակա-նությունը պետք է կրկնել մինչև որ գտնվի օպտիմալ լուծումը:

9. Սկզբնական աղյուսակից վերցնելով և գումարելով վերջնական աղյուսակի գրոներ պարունակող դիրքերին համապատասխան թվերը՝ կատացվի որոնելի նվազագույն գումարը:

Նշանակումների խնդրի լուծման ծրագրի նկարագրությունը

«Հունգարական մեթոդով» նշանակումների խնդրի լուծումը ներկա-յացված է ծրագրավորման միջոցներով: Ծրագրի կոդը գրված է C# օբյեկտ-կողմնորոշված ծրագրավորման լեզվով: Ծրագիրն աշխատում է գրաֆիկական ռեժիմում:

Ծրագիրն աշխատում է երեք ձևի (Form) հետ: Դրանցից յուրաքանչ-յուրն ունի իր որոշակի իմաստը և նշանակությունը: Այդ ձևերը գործում են փոխկապակցված և ապահովում ծրագրի լիարժեք աշխատանքը:

Առաջին ձևը (Form1) տեղեկատվական էջ է այն մասին, թե ինչ խնդիր է լուծում ծրագիրը (նկար 1):

Երկրորդ ձևի (Form2) վրա շարադրված է նշանակումների խնդրի դրվածքը ընդհանուր դեպքում: Այս էջում, խնդիրը «հունգարական մեթո-դով» լուծելու համար, ծրագիրը պահանջում է նաև առաջադրված տարբե-րակներից ընտրել աշխատանքների, կամ որ նույնն է, աշխատողների քանակը (նկար 2): Բերված օրինակում ընտրված է հինգ աշխատանք:

Աշխատանքների քանակն ընտրելուց հետո, մկնիկի միջոցով սեղմելով «ԻՆԻՏԻ ԼՈՒԾՈՒՄԸ ՀՈՒՆԳԱՐԱԿԱՆ ՄԵԹՈԴՈՎ» գրառման վրա, ծրագիրը կանցնի հաջորդ էջ (Form3): Բացված էջում կազմվում է աշխատանքների (աշխատողների) ընտրված քանակին համապատաս-խան դատարկ աղյուսակ, որի տարրերը՝ աշխատաժամանակները, լրաց-նելու համար պետք է սեղմել էջի ներքևում առկա «Ներմուծել աշխա-տանքների կատարման ժամանակները» գրառումը (նկար 3):

Ծրագիրը հնարավորություն է տալիս աշխատաժամանակներն աղյուսակում լրացնել պատահական ընտրությամբ կամ ստեղծաշարից ներմուծման միջոցով: Երկու դեպքում էլ տվյալները պետք է ներմուծել 1-ից 20 միջակայքից: Հարկ է նշել, որ թե՛ աշխատանքների քանակի (4-ից 6) և թե՛ աշխատաժամանակների (1-ից 20) վրա դրված սահմանափակում-

ները խնդրի տվյալները գրաֆիկական ռեժիմում ավելի պատկերավոր ներկայացնելու նպատակ են հետապնդում: Ներմուծման տարբերակը ընտրելու համար ծրագիրը կատարում է հարցում (նկար 4): Հարցմանը “Yes” պատասխանելու դեպքում աղյուսակը լրացվում է թվերի պատահական ընտրությամբ, հակառակ դեպքում՝ ստեղնաշարի միջոցով [3, 95-103, 120-155]:

Բերված օրինակում հարցմանը տրված է “No” պատասխան, և աշխատաժամանակները ներմուծված են ստեղնաշարի միջոցով (նկար 5, նկար 6):

Պետք է նշել, որ տվյալների ներմուծումը ստեղնաշարի միջոցով իրականացված է C# լեզվի գրաֆիկական ռեժիմում աշխատող InputBox պատուհանի միջոցով:

Աշխատաժամանակների ներմուծումից հետո ծրագիրը շարունակում է կատարվել ըստ «հունգարական մեթոդով» նշանակումների խնդրի լուծման՝ վերը շարադրված ալգորիթմի քայլերի հաջորդականության: Բոլոր քայլերի կատարման արդյունքները ցուցադրվում են նույն էջի վրա (Form3): Յուրաքանչյուր անգամ էջի ներքևի մասում նշվում է հաջորդ կատարվելիք քայլը, որը իրականացվում է մկնիկի միջոցով այդ հուշման վրա սեղմելուց հետո միայն: Այսպիսով, «Գտնել յուրաքանչյուր տողի փոքրագույն տարրը» հուշման վրա սեղմելուց հետո ծրագիրը աղյուսակի աջ մասում ավելացնում է լրացուցիչ սյուն և այն լրացնում համապատասխան տողերի փոքրագույն տարրերով (նկար 7), իսկ հաջորդ քայլում յուրաքանչյուր տողի տարրերից հանում է համապատասխան տողի փոքրագույն տարրը (նկար 8) [4, 249-285; 784-825]:

Առաջնորդվելով հուշումներով՝ ծրագիրը աղյուսակի ներքևում ավելացնում է տող, այդտեղ գրանցում յուրաքանչյուր սյան փոքրագույն տարրը (նկար 9), այնուհետև, յուրաքանչյուր սյան տարրերից հանում համապատասխան սյան փոքրագույն տարրը (նկար 10):

Համապատասխան հրահանգից հետո ծրագիրը աղյուսակից հեռացնում է փոքրագույնները որոշող տողը և սյունը (նկար 11):

«Նշել զրոներ պարունակող տողերը և սյուները» հրահանգից հետո աղյուսակում կարմիր գույնով նշվում են զրոներ պարունակող բոլոր տողերը և սյուները (նկար 12):

Ծրագիրը, ըստ ալգորիթմի, գտել և կապույտ գույնով աղյուսակում նշել է կարմիրով չնշված տարրերից փոքրագույնը (նկար 13):

Հաջորդ քայլերում ծրագիրը աղյուսակում կարմիրով չնշված տարրերից հանում է նախորդ քայլում գտած փոքրագույն տարրը (նկար 14), իսկ կարմիրով նշված տողերի և սյուների հատման կետերում գտնվող

տարրերին գումարում այդ փոքրագույնը (նկար 15):

«Աշխատանքների բաշխումն ըստ աշխատողների և ընդհանուր գումարը» հրահանգից հետո աղյուսակում որոշակի գրոներ նշվում են բաց կանաչ գույնով (նկար 16): Դրանցից յուրաքանչյուրը ցույց է տալիս, թե որ աշխատողը որերորդ աշխատանքը պետք է կատարի: Ըստ այդ բաշխումների՝ S փոփոխականի մեջ գումարվում են նախնական աղյուսակի համապատասխան թվերը՝ աշխատաժամանակները, և արտածվում:

Նկար 16-ում ցուցադրված է երրորդ էջի վերջնական տեսքը: Էջի կենտրոնում ձևափոխությունների արդյունքում ստացված վերջնական աղյուսակն է, ձախ կողմում՝ նախնական աղյուսակը, աջում՝ աշխատանքների բաշխումն ըստ աշխատողների և արդյունքում ստացված նվազագույն գումարը:

Ծրագրում օգտագործված են C# ծրագրավորման լեզվի հարուստ գրաֆիկական միջոցները, որի շնորհիվ էլ ծրագիրն ունի պատկերավոր գրաֆիկական ձևավորում:

Ձևերը (Form)՝ էջերը, միմյանց հետ կապված են դրանց ներքնում տեղադրված սլաքների միջոցով, որոնց ուղղությունն էլ ցույց է տալիս նախորդ կամ հաջորդ էջին անցումը:

Եզրակացություն

Հոդվածում ներկայացված է «հունգարական մեթոդով» նշանակումների խնդրի լուծման հնարավորությունները ժամանակակից ծրագրավորման միջոցներով, մասնավորապես C# ծրագրավորման լեզվով:

Մշակված ծրագիրը թույլ է տալիս նախնական տվյալների մուտքագրումն իրականացնել պատահական ընտրությամբ կամ ստեղնաշարի միջոցով: Այն հնարավորություն է տալիս նաև խնդրի լուծման ակտիվիզմով նախատեսված քայլերի կատարմանը հետևել քայլ առ քայլ, քանի որ դրանց արդյունքները ցուցադրվում են էկրանին գրաֆիկական ռեժիմում:

Հոդվածում ներկայացված է նշանակումների խնդրի ծրագրային կատարման օրինակ խնդրի քայլ առ քայլ լուծմամբ և միջանկյալ ու վերջնական արդյունքների ցուցադրմամբ:

Ծրագրավորման ժամանակակից տեխնոլոգիաների օգտագործումը խնդիրներ լուծելիս բարելավում է ուսուցման արդյունավետությունը, որակը և նպաստում սովորողների ինքնուրույն մտածողության զարգացմանը:

Ծրագիրը կարող է օգտագործվել «մաթեմատիկական մոդելավորում», «օպտիմիզացիայի մեթոդներ», «տնտեսամաթեմատիկական մոդելներ» դասընթացներն ուսումնասիրելիս:

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴԻՐ



Նկար 1. Տեղեկատվական էջ



ԽՆԴԻՐԻ ԴՐՎԱԾՔԸ

Անհրաժեշտ է կատարել *N* տարբեր տիպի աշխատանք: Դրա իրականացման համար անհրաժեշտ է ներգրավել *N* աշխատող: Յուրաքանչյուր աշխատող պատրաստ է կատարել ցանկացած աշխատանք որոշակի վճարի դիմաց: Ցանկացած աշխատանք պետք է հանձնարարվի մեկ աշխատողի: Պահանջվում է աշխատանքն աշխատողների միջև բաշխել այնպես, որ ամբողջ աշխատանքի կատարման ընդհանուր արժեքը լինի նվազագույն:

ԽՆԴԻՐԻ ԼՈՒՑՈՒՄԸ ՀՈՒՆԳԱՐԱԿԱՆ ՄԵԹՈԴՈՎ

Աշխատանքների քանակը	
4	
5	
6	

Նկար 2. Աշխատանքների քանակի ընտրության էջ

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5
1					
2					
3					
4					
5					

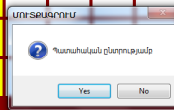


Ներմուծել աշխատանքների կատարման ժամանակները

Նկար 3. Աղյուսակի ձևավորման և աշխատաժամանակների լրացման էջ

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

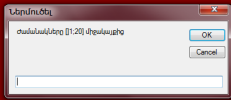
Աշխատող Աշխատանք	1	2	3	4	5
1					
2					
3					
4					
5					



Ներմուծել աշխատանքների կատարման ժամանակները

Նկար 4. Աշխատաժամանակների ներմուծման հարցում

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ



Աշխատող Աշխատանք	1	2	3	4	5
1	7	3	6	9	5
2	7	5	7	5	6
3	7	6	8	8	9
4	3	1	6		
5					



Ներմուծել աշխատանքների կատարման ժամանակները

Նկար 5. Ստեղծարարի միջոցով աշխատաժամանակների ներմուծման ընթացքը

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5
1	7	3	6	9	5
2	7	5	7	5	6
3	7	6	8	8	9
4	3	1	6	5	7
5	2	4	9	9	5



Գտնել յուրաքանչյուր տողի փոքրագույն տարրը

Նկար 6. Աղյուսակի տեսքը աշխատաժամանակների ներմուծումից հետո

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5	Փոքրագ.
1	7	3	6	9	5	3
2	7	5	7	5	6	5
3	7	6	8	8	9	6
4	3	1	6	5	7	1
5	2	4	9	9	5	2



Յուրաքանչյուր տողի տարրերից հանել այդ տողի փոքրագույնը

Նկար 7. Աղյուսակի տեսքը տողերի փոքրագույն տարրերը գտնելուց հետո

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5	Փոքրագ.
1	4	0	3	6	2	3
2	2	0	2	0	1	5
3	1	0	2	2	3	6
4	2	0	5	4	6	1
5	0	2	7	7	3	2



Գտնել յուրաքանչյուր սյան փոքրագույն տարրը

Նկար 8. Աղյուսակի տեսքը տողերի տարրերից փոքրագույնը հանելուց հետո

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5	Փոքրագ.
1	4	0	3	6	2	3
2	2	0	2	0	1	5
3	1	0	2	2	3	6
4	2	0	5	4	6	1
5	0	2	7	7	3	2
Փոքրագ.	0	0	2	0	1	



Ցուրաքանչյուր սյան տարրերից հանել այդ սյան փոքրագույնը

Նկար 9. Աղյուսակի տեսքը սյուների փոքրագույն տարրերը գտնելուց հետո

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5	Փոքրագ.
1	4	0	1	6	1	3
2	2	0	0	0	0	5
3	1	0	0	2	2	6
4	2	0	3	4	5	1
5	0	2	5	7	2	2
Փոքրագ.	0	0	2	0	1	



Հեռացնել փոքրագույնները որոշող տողը և սյունը

Նկար 10. Աղյուսակի տեսքը սյուների տարրերից փոքրագույնը հանելուց հետո

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5
1	4	0	1	6	1
2	2	0	0	0	0
3	1	0	0	2	2
4	2	0	3	4	5
5	0	2	5	7	2



Նշել գրոներ պարունակող տողերը և սյուները

Նկար 11. Աղյուսակի տեսքը փոքրագույնները որոշող տողը և սյունը հեռացնելուց հետո

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5
1	4	0	1	6	1
2	2	0	0	0	0
3	1	0	0	2	2
4	2	0	3	4	5
5	0	2	5	7	2



Գտնել չնշված տարրերից փոքրագույնը

Նկար 12. Աղյուսակի տեսքը գրոներ պարունակող տողերը և սյուները նշելուց հետո

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5
1	4	0	1	6	1
2	2	0	0	0	0
3	1	0	0	2	2
4	2	0	3	4	5
5	0	2	5	7	2



Չնշված տարրերից հանել փոքրագույնը

*Նկար 13. Աղյուսակը փոքրագույն տարրը գտնելուց և
կապույտ գույնով նշելուց հետո
(1-ին տողի և 3-րդ սյան հատման կետում գտնվող տարրը)*

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5
1	4	0	0	5	0
2	2	0	0	0	0
3	1	0	0	2	2
4	2	0	2	3	4
5	0	2	4	6	1



*Նշված տողերի և սյուների հատման կետերում գտնվող
տարրերին գումարել փոքրագույնը*

*Նկար 14. Աղյուսակը կարմիրով չնշված տարրերից
փոքրագույն տարրը հանելուց հետո*

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Աշխատող Աշխատանք	1	2	3	4	5
1	4	0	0	5	0
2	3	1	0	0	0
3	2	1	0	2	2
4	2	0	2	3	4
5	0	2	4	6	1



*Աշխատանքների բաշխումն ըստ աշխատողների
և ընդհանուր գումարը*

*Նկար 15. Աղյուսակը կարմիրով նշված տողերի և սյուների հատման կետերում
գտնվող տարրերին փոքրագույն տարրը գումարելուց հետո*

ՆՇԱՆԱԿՈՒՄՆԵՐԻ ԽՆԴՐԻ ԼՈՒԾՈՒՄԸ

Նախնական աղյուսակ

```

1 2 3 4 5
1 7 3 6 9 5
2 7 5 7 5 6
3 7 6 8 8 9
4 3 1 6 5 7
5 2 4 9 9 5
    
```

Աշխատող Աշխատանք	1	2	3	4	5
1	4	0	0	5	0
2	3	1	0	0	0
3	2	1	0	2	2
4	2	0	2	3	4
5	0	2	4	6	1

Աշխատանքի բաշխում

```

1 -> 5
2 -> 4
3 -> 3
4 -> 2
5 -> 1
    
```

S = 21



*Նկար 16. Աշխատանքների բաշխումն ըստ աշխատողների և
ընդհանուր նվազագույն գումարը*

Գրականություն

1. Бродецкий Г.Л., Гусев Д.А. Экономико-математические методы и модели в логистике, процедуры оптимизации, учебное пособие. Москва, Издательский центр “Академия”, 2012, 190 с.
2. Волков И.К., Загоруйко Е.А. Исследование операций, учебное пособие. Издательство МГТУ имени Н.Э. Баумана, 2002, 436 с.
3. Культин Н.Б. Misrosoft Visual C# в задачах и примерах, БХВ Петербург, 2009, 320 с.
4. Шилдт Г. C# 4.0 полное руководство. Москва, Вильямс, 2011, 1056 с.

Задача о назначениях и программа ее решения

Рузанна Мазмания

Резюме

Ключевые слова: линейное программирование, язык программирования, венгерский метод, алгоритм, параметр, оптимальное решение, интервал

В статье разобрана одна из фундаментальных задач линейного программирования – задача о назначениях. В ней рассматривается программная реализация решения задачи о назначениях.

Специфические особенности задач о назначениях послужили поводом для появления эффективного венгерского метода их решения. Основная идея венгерского метода заключается в переходе от исходной квадратной матрицы стоимости к эквивалентной ей матрице с неотрицательными элементами и системой независимых нулей, из которых никакие два не принадлежат одной и той же строке или одному и тому же столбцу.

Алгоритм основан на двух идеях:

- если из всех элементов некой строки или столбца вычесть одно и то же число, такие, что все элементы матрицы останутся неотрицательными, то оптимальное решение не изменится,
- если есть решение нулевой стоимости, то оно оптимально.

Программа составлена на объектно-ориентированном языке программирования C# и работает в графическом режиме C#.

Программа, реализующая решение задачи, в начале требует ввод исходных данных, в графическом режиме составляет квадратную матрицу цен, а после, путем выполнения последовательности шагов алгоритма венгерского метода, достигается окончательное решение задачи.

По составленной квадратной матрице цен алгоритм находит оптимальное распределение работ по исполнителям, так чтобы все работы были распределены и каждому исполнителю досталась ровно одна работа. Программа считает также общие затраты на выполнение всех работ.

Пошаговая реализация решения задачи о назначениях венгерским методом демонстрируется в графическом режиме языка программирования C#.

The Assignment Problem and the Program for its Solution

Ruzanna Mazmanyan

Summary

Key words: *linear programming, programming language, Hungarian method, algorithm, parameter, optimal solution, interval*

The article deals with one of the fundamental problems of linear programming – the assignment problem. It considers the software implementation of solving the assignment problem.

The specific features of assignment problems gave rise to an effective Hungarian method for their solution. The main idea of the Hungarian method is to move from the original square cost matrix to an equivalent matrix with non-negative entries and a system of independent zeros, no two of which belong to the same row or the same column.

The algorithm is based on two ideas:

- if one and the same number is subtracted from all elements of a certain row or column, such that all elements of the matrix remain non-negative, then the optimal solution will not change,
- if there is a zero cost solution, then it is optimal.

The program is written with the object-oriented programming language C# and runs in C# graphics mode.

The program that implements the solution of the problem at the beginning requires the input of initial data, forms a square matrix of prices in the graphical mode, and then, by performing a sequence of steps of the Hungarian method algorithm, the final solution of the problem is achieved.

According to the compiled square price matrix, the algorithm finds the optimal distribution of work among performers, so that all tasks are distributed and each performer gets exactly one task. The program also calculates the total cost necessary to fulfill the entire work.

In the graphical mode of the C# programming language, a step-by-step implementation of solving the assignment problem using the Hungarian method is demonstrated.

Ներկայացվել է 11.03.2023 թ.

Գրախոսվել է 28.04.2023 թ.

Ընդունվել է տպագրության 25.05.2023 թ.