## 54.

Minimum:

% Coefficients of the objective function for minimization

f = [-3; 1; -1];

% Coefficients of the equality constraints

Aeq = [1, 1, -1;

2, 5, 4;

-1, -4, -5];

beq = [1; 3; -2];

% Lower bounds for the variables

lb = [0; 0; 0];

% Solve the linear programming problem

[x, fval] = linprog(f, [], [], Aeq, beq, lb, []);

% Display the results

disp('Optimal values of the variables (for minimization):')

disp(x)

disp('Optimal value of the objective function (minimized):')

disp(fval)

Maksimum:

% Coefficients of the objective function for maximization (converted to minimization)

f = [3; -1; 1];

% Coefficients of the equality constraints

Aeq = [1, 1, -1;

2, 5, 4;

-1, -4, -5];

beq = [1; 3; -2];

% Lower bounds for the variables

lb = [0; 0; 0];

% Solve the linear programming problem

[x, fval] = linprog(f, [], [], Aeq, beq, lb, []);

% Adjust the objective function value for maximization

fval = -fval;

% Display the results

disp('Optimal values of the variables (for maximization):')

disp(x)

disp('Optimal value of the objective function (maximized):')

disp(fval)

## 57.

% Define the range for x1 and x2

x1 = linspace(0, 4, 400);

x2 = linspace(0, 8, 400);

% Define the constraints

x2\_1 = 1 - x1;

x2\_2 = 2 + 2 \* x1;

x2\_3 = 4 - x1;

% Plot the constraints

figure;

hold on;

plot(x1, x2\_1, 'r', 'LineWidth', 2, 'DisplayName', 'x1 + x2 = 1');

plot(x1, x2\_2, 'b', 'LineWidth', 2, 'DisplayName', '-2x1 + x2 = 2');

plot(x1, x2\_3, 'g', 'LineWidth', 2, 'DisplayName', 'x1 + x2 = 4');

xline(3, 'k--', 'LineWidth', 2, 'DisplayName', 'x1 = 3');

% Shade the feasible region

x\_fill = [0, 0, 3, 3, 0];

y\_fill = [1, 4, 1, 0, 1];

fill(x\_fill, y\_fill, 'k', 'FaceAlpha', 0.1, 'EdgeAlpha', 0);

% Set labels and title

xlabel('x1');

ylabel('x2');

title('Feasible Region for the Linear Programming Problem');

legend;

grid on;

axis([0 4 0 8]);

hold off;

## 63. Cüt-cüt müqayisə metodu

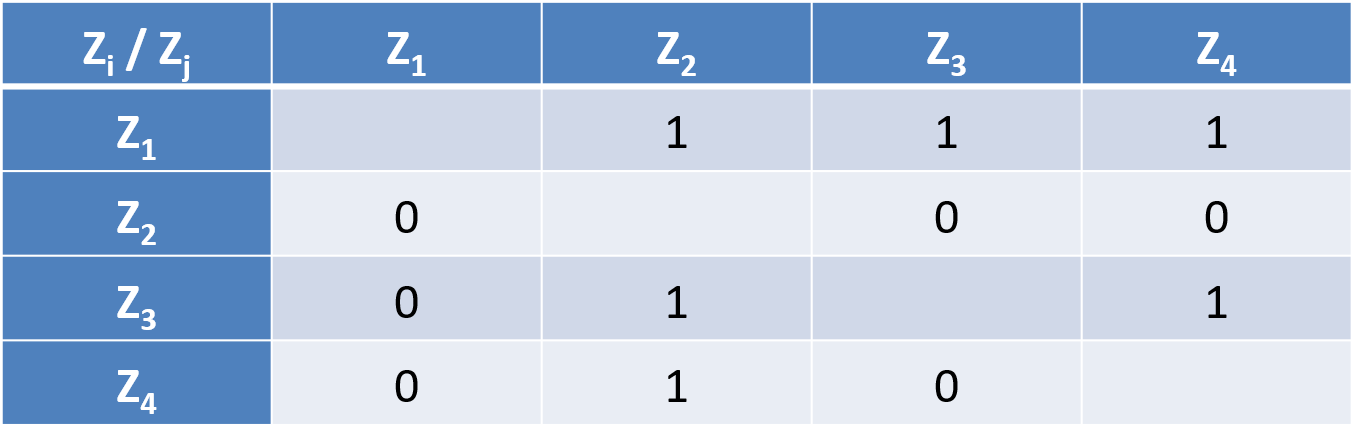
Cüt-cüt müqayisə metodu (Pairwise Comparison Method), qərar vermə proseslərində alternativləri bir-biri ilə müqayisə etmək üçün istifadə olunan bir metodologiyadır. Bu metod, alternativlərin bir-birinə nisbətən üstünlük və ya qabiliyyətlərini qiymətləndirməyə kömək edir və daha sonra ən yüksək prioritetli alternativi təyin etmək üçün nəticələri birləşdirir. Metodun Əsas Addımları:

1. Alternativlərin Seçilməsi: Qiymətləndiriləcək olan alternativlər seçilir. Bu alternativlər müxtəlif məhsul və ya xidmətlər ola bilər, hər biri müxtəlif xüsusiyyətlərə malik olur.

2. Qarşılaşdırma Matrixinin Hazırlanması: Hər bir alternativin digər alternativlərə nisbətən qiymətləri təyin olunur. Bu, bir qarşılaşdırma matrisində göstərilir, hər bir element dəyəri müqayisə edilən iki alternativin nisbətini ifadə edir.

3. Qiymətləndirmə Prosesi: Hər bir iştirakçı, hər iki alternativi qiymətləndirir və onların arasında müqayisə edir. Dəyərləndirmə əsasən, hər alternativ üzrə təşkil olunan dəyər sıraları tərtib edilir.

4. Nəticələrin Birləşdirilməsi: İştirakçıların qiymətləndirmələri birləşdirilir və hər alternativ üzrə ümumi nəticələr hesablanır.

5. Ən Yüksək Prioritetli Alternativin Seçilməsi: Hesablanan nəticələr əsasında, ən yüksək prioritetli və ya ən üstün alternativ təyin edilir və qiymətləndirilmə prosesi başa çatır.

Burda hər biri üçün uyğun C1=3, C2=0, C3=2, C4=1. Daha sonra bu qiymətlər toplanır. Uyğun olara Ci-lər bu cəmə bölünür. Nəticə uyğun şərtin çəkisidir

Misal: Bir şirkət yeni bir layihəni başlatmaq istəyir və əlavə edilmiş bir iş növünü seçmək üçün dördən çox variant arasında seçim etməli olduğunu düşünür. İştirakçılar tərəfindən alternativlərin bir-biri ilə qiymətləndirilməsi aşağıdakı kimi ola bilər:

- Variant A: 8 bal

- Variant B: 6 bal

- Variant C: 7 bal

- Variant D: 5 bal

Bu nümunədə, hər iştirakçı hər iki variantı qiymətləndirir və verilən nəticələr birləşdirilir. Əsasən, Variant A, ən yüksək nəticə ilə qiymətləndirildiyi üçün ən yüksək prioritetli variant kimi seçilə bilər.

## 64. Cüt-cüt müqayisə metodu ilə nəqliyyat problemi

Cüt-cüt müqayisə metodu (Pairwise Comparison Method), nəqliyyat problemləri kimi alternativlərin qiymətləndirilməsində də tətbiq edilə bilən bir metodologiyadır. Bu metodu nəqliyyat probleminə uyğun olaraq təsvir etmək üçün aşağıdakı nümunəni düşünə bilərsiniz:

Nümunə: Şəhər içi Nəqliyyat Sistemi Seçimi

Bir şəhərdə yeni bir nəqliyyat sistemi tətbiq etmək istəyirsiniz və bu üç variant arasında seçim etməli olduğunuzu düşünürsünüz:

1. Metro: Şəhərin istənilən yerinə sürətli çatdırılmanı təmin edən nəqliyyat vasitəsidir.

2. Tramvay: Şəhər daxilində hərəkət edən, ümumiyyətlə təhlükəsiz və aşağı hərəkətli sürətli nəqliyyat vasitəsi.

3. Avtobus: Mühərriklər işləyən, müəyyən bir marştrut üzrə hərəkət edən nəqliyyat vasitəsi.

Bu üç variant arasında seçim etməyə qərar verən bir qrup iştirakçı bu metodla hər iki variantı qarşılaşdırmağa cəhd edə bilər. Hər bir iştirakçı, hər bir variantın müxtəlif xüsusiyyətlərini qiymətləndirir və buna uyğun olaraq dəyərləndirilə bilər:

İştirakçı 1:

- Metro vs. Tramvay: Metro daha sürətlidir.

- Metro vs. Avtobus: Metro daha təhlükəsiz və rahatdır.

- Tramvay vs. Autobus: Tramvay daha rahat və daha asandır.

İştirakçı 2:

- Metro vs. Tramvay: Metro daha sürətlidir lakin, Tramvay şəhər daxilində daxili işləyir.

- Metro vs. Avtobus: Avtobus mühərriklə ilə işlənir.

- Tramvay vs. Avtobus: Avtobus sürətli marşrutu rahatdır.

## 65. Cüt-cüt müqayisə metodu ilə logistika məsələsinin həlli

Cüt-cüt müqayisə metodu (Pairwise Comparison Method), logistika problemlərinin həllində də effektiv bir metod olaraq tətbiq edilə bilər. Logistika, malların, xidmətlərin və informasiyanın optimal şəkildə idarə edilməsini və təhlükəsiz tədarükünü təmin edən kompleks proseslərin bütünüdür. Bu məsələlər həll edilərkən fərqli alternativləri qiymətləndirmək üçün bu metod istifadə edilə bilər.

Nümunə: Lojistik Şirkətinin Daşınma Optimalizasiyası

Bir logistika şirkəti, müştərilərinə optimal və effektiv daşınma xidməti təqdim etmək istəyir. Bu məqsədlə, hər bir nəqliyyat metodunu (yol və dəniz yolu daşınmaları, hava nəqliyyatı və s.) qiymətləndirmək üçün cüt-cüt müqayisə metodu tətbiq edə bilər:

1. Yol Nəqliyyatı: Yerli və beynəlxalq daşınma üçün ən təsirlisi, lakin daha uzun zaman tələb edir və yol şəraitindən asılı olaraq müəyyən risklərə malikdir.

2. Dəniz Nəqliyyatı: Çox güvənilir, ancaq daha uzun yol alır və logistika planlaması daha uzun müddət tələb edir.

3. Hava Nəqliyyatı: Sürətli və qısa vaxtda mal tədarükü təmin edir, lakin daha bahalı ola bilər.

Bu nümunədə, logistika şirkəti üçün hər bir nəqliyyat alternativini məsuliyyətlə dəyərləndirən iştirakçılar seçilir. Hər iştirakçı, hər iki alternativi qiymətləndirir və onların arasında müqayisə edir. Hər bir nəqliyyat metodunun avantajları və məhdudiyyətləri hesablanır və nəticədə ən yüksək prioritetli nəqliyyat metodları seçilə bilər.

## 66. Ardıcıl müqayisə üsulunun tətbiqinə aid məsələ: ən yaxşı alternativin seçilməsi

Ardicıl müqayisə (Sequential Comparison Method), alternativləri ardıcıl şəkildə qiymətləndirib, hər birini digəri ilə müqayisə edərək ən yaxşı alternativi təyin etmək üçün istifadə olunan bir metodologiyadır. Bu metod, hər bir alternativin müqayisədən keçərək, hansının digər alternativlərdən daha üstün olduğunu təyin etmək üçün nəticələri birləşdirir.

Məsələ: İşə Götürmə Prosesi Üçün Ardıcıl Müqayisə

Bir şirkət işə götürmə prosesində ardıcıl müqayisə üsulundan istifadə edərək ən yaxşı namizədi seçmək istəyir. Şirkət müvafiq kriterlər əsasında hər bir namizədi qiymətləndirir və ardınca bu qiymətləndirmələri müqayisə edir.

1. Namizəd A: Ən yaxşı təcrübə və uyğun bacarıqlar.
2. Namizəd B: Güclü kommunikasiya və liderlik bacarığı, lakin təcrübə azdır.
3. Namizəd C: Çox yaxşı akademik və təcrübi fəaliyyət, lakin kommunikasiya bacarığı zəifdir.

Hər bir namizəd, müvafiq mərhələlərdən keçirilir və hər bir mərhələdə şirkətin tələblərinə görə qiymətləndirilir. Ardıcıl müqayisə zamanı, hər bir namizədin əsas mərhələlərdən keçirilərək hansının daha çox potensiala malik olduğunu müəyyənləşdirirlər.

Qiymətləndirmə Mərhələləri:

- Namizəd A: İlk mərhələsində yüksək performans göstərir, lakin ikinci mərhələsində kommunikasiya bacarığı daha zəifdir.

- Namizəd B: İlk mərhələsində performans aşağıdır, lakin kommunikasiya bacarığı yüksəkdir.

- Namizəd C: Akademik bacarıqlar və təcrübi fəaliyyətlərdə ən yüksək nəticələri əldə edir, lakin liderlik və kommunikasiya mərhələlərində zəifdirlər.

## 67. Ardıcıl müqayisə üsulunun tətbiqinə aid məsələ: firmanın əlavə gəlir əldə etməsi üçün ekspert təklifində ən əlverişli variantın seçilməsi

Ardıcıl müqayisə üsulu, firmalar və ya digər təşkilatlar tərəfindən alternativ variantları qiymətləndirmək və əlavə gəlir əldə etmək üçün ən optimal seçim etmək üçün istifadə olunan üsuldur. Bu üsul hər bir variantın nisbi üstünlüklərini qiymətləndirərək təşkilat üçün ən yaxşı variantı müəyyən etməyə kömək edir.

Nümunə: Firmanın Əlavə Gəlirləri üzrə Ekspert Təklifləri

Bir firmanın əlavə gəlir əldə etmək üçün əldə etdiyi dörd fərqli ekspert təklifi var:

1. Ekspert A: Yeni marketinq kampaniyasına başlamaq üçün xərcləri artırın.

2. Ekspert B: Mövcud məhsulların yeni seqmenti üçün təşviq və reklam kampaniyası.

3. Ekspert C: Yeni məhsul və ya xidmətin hazırlanması və işə salınması.

4. Ekspert D: Yeni bazar araşdırması və müştəri tələblərinin tədqiqi.

Bu təkliflər arasında müqayisəli müqayisə metodunu tətbiq etməklə şirkətin ən sərfəli variantını seçmək mümkündür. Bir iş meneceri və ya bir qrup iştirakçı bu metodu izləyə və hər bir təklifi aşağıdakı meyarlara əsasən qiymətləndirə bilər:

Meyarlar:

- Qiymət: Hər bir təklifin maliyyə tələbləri və potensial gəlirləri.

- İdarəetmə zərurəti: Təklifi həyata keçirmək üçün rəhbərliyin və üzvlərin cəlb edilməsi zərurəti.

- Risklər: Təklifin yaradılması nəticəsində yarana biləcək risklər və ya son tarixlər.

Tədricən Müqayisə Prosesi:

- Təklif A və Təklif B: Xərclərin, idarəetmə ehtiyaclarının və risklərin müqayisəsi.

- Təklif B və Təklif C: Yeni məhsulun təqdimatı və yeni seqmentə təqdimat və reklam kampaniyası.

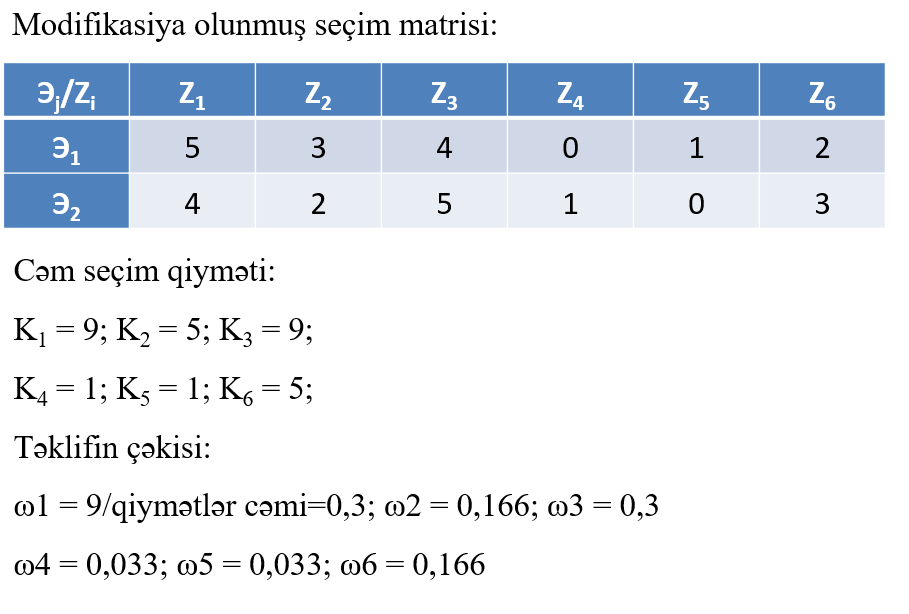
- Təklif C və Təklif D: Yeni məhsulun təqdimatı və yeni bazar araşdırması və müştəri tələblərinin araşdırılması.

Bu misalda hər bir təklifi firmanın məqsədlərinə daha uyğun olaraq qiymətləndirən ardıcıl müqayisə üsulundan istifadə edilir. Ən yaxşı variantı seçmək üçün nəticələr birləşdirilir və ən yüksək prioritet təklif təyin edilir.

## 68. Strukturlaşdırılmamış problemlərin həll prinsipləri

Strukturlaşdırılmamış problemlər, adından da aydın olacağı kimi, məhsulun və ya nəticənin bəlli qaydalar və həll yolları ilə dəyərləndirilməsində çətinlik yaradan məsələlərdir. Bu cür problemləri həll etmək üçün bir neçə əsas prinsip və yanaşma mövcuddur. Nümunəvi strukturlaşdırılmamış bir problemi aşağıda nəzərdən keçirək:

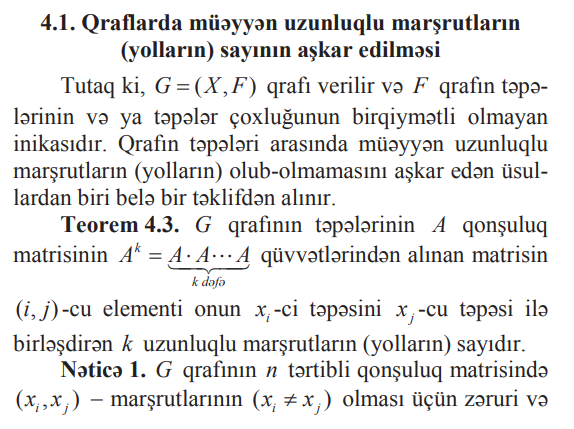
Məsələ: Yeni bir marketinq strategiyasının tətbiqi

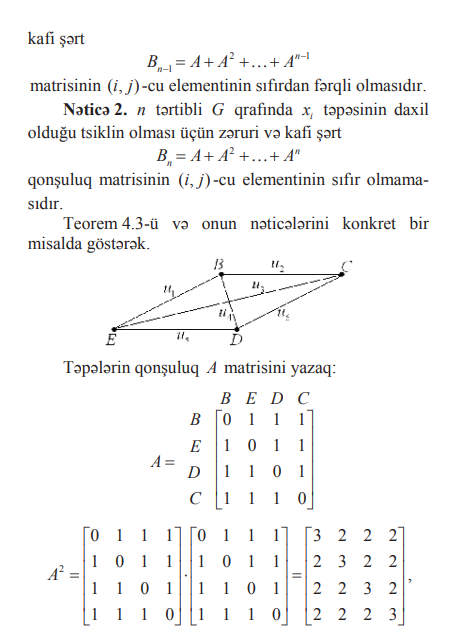
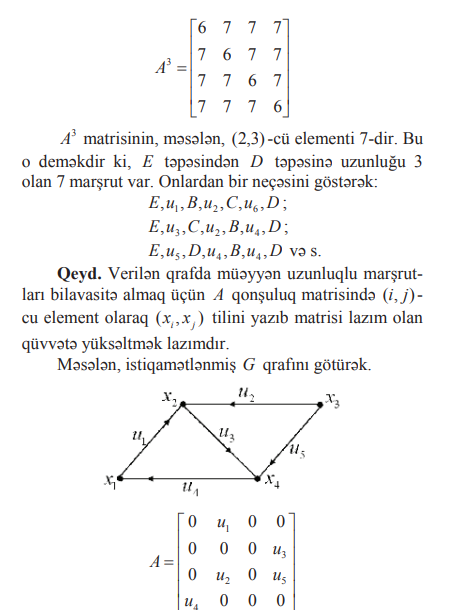
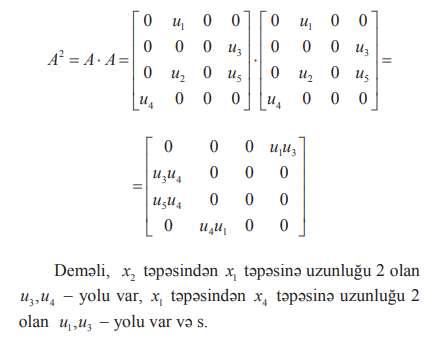
Bir şirkət yeni bir marketinq strategiyasını tətbiq etməyə qərar verib. Bu zaman 2 ekspertdən qarşılarına qoyduqları 6 məqsəd üzrə dəyərləndirmə almaq istəyirlər. Ekspertlərin 6 məqsədlə bağlı dəyərləndirmələri:

Qiymətlərin cəmi 30-dur

69=68 | 70-73- 68-lə oxşardır mövzu və ekspert sayı dəyişir. | 74=63-64.

## 75. Qraflarda müəyyən uzunluqlu marşrutların sayının aşkar edilməsi





## 76. Monte-Karlo üsulu ilə pi ədədinin hesablanması

Monte-Karlo üsulu statistik metodlarla təqribi həllərin tapılmasında istifadə edilən bir yanaşmadır. Bu üsulla pi (π) ədədini təqribi olaraq hesablamaq üçün aşağıdakı addımları izləmək mümkündür:

Monte-Karlo üsulu ilə pi ədədinin hesablanması

1. Nöqtələrin İstiqamətləndirilməsi - Bir vahid radiuslu dairənin daxilində təsadüfi nöqtələr yaratmaq.

2. Nöqtələrin Sayılması - Dairənin daxilində və kvadratın daxilində neçə nöqtə olduğunu saymaq.

3. Pi Ədədinin Təxmini - Pi ədədini, dairənin sahəsi ilə kvadratın sahəsi arasındakı nisbət əsasında hesablamaq.

Addımlar

1. Kvadratın sahəsi: 1\*1=1 (Çünki kvadratın hər tərəfi 1-dir).

2. Dairənin sahəsi: (Çünki dairənin radiusu 1-dir).

Dairə kvadratın daxilində olduğuna görə, təsadüfi nöqtələrdən dairənin daxilinə düşən nöqtələrin sayı ilə ümumi nöqtələrin sayı arasındakı nisbət, dairənin sahəsinin kvadratın sahəsinə nisbəti olacaq.

Beləliklə:

Buradan π-nı hesablamaq üçün:

Bu üsulla əldə edilən pi ədədinin dəqiqliyi nümunə sayının (neçə nöqtənin istifadə olunması) artması ilə artır. Bu sadə metod Monte-Karlo üsulunun əsas prinsiplərini göstərir və statistik metodlarla təqribi həllərin necə tapıldığını izah edir.

## 77. Monte-Karlo üsulu ilə inteqralın hesablanması

## 78. Qrafın maksimal möhkəm əlaqəli altqraflara ayrılışı

Qrafların maksimal möhkəm əlaqəli altqraflara ayrılması qraf teoriyasında önəmli bir konudur. Bu proses qrafın strukturunu daha yaxşı anlamağa və analiz etməyə kömək edir. Maksimal möhkəm əlaqəli altqraf, qrafın hamısını özündə əhatə edən və təkrarlanan qövsləri olmayan altqraf deməkdir.

Maksimal Möhkəm Əlaqəli Altqrafların Tapılması

Qrafın maksimal möhkəm əlaqəli altqraflarını tapmaq üçün iki yanaşma mövcuddur: DFS (Dərinlikə gedən axtarış) və BFS (Genişliyə gedən axtarış).

DFS (Dərinlikə Gedən Axtarış) ilə Tapılması

DFS alqoritmi ilə maksimal möhkəm əlaqəli altqrafların tapılması aşağıdakı adımları izləyir:

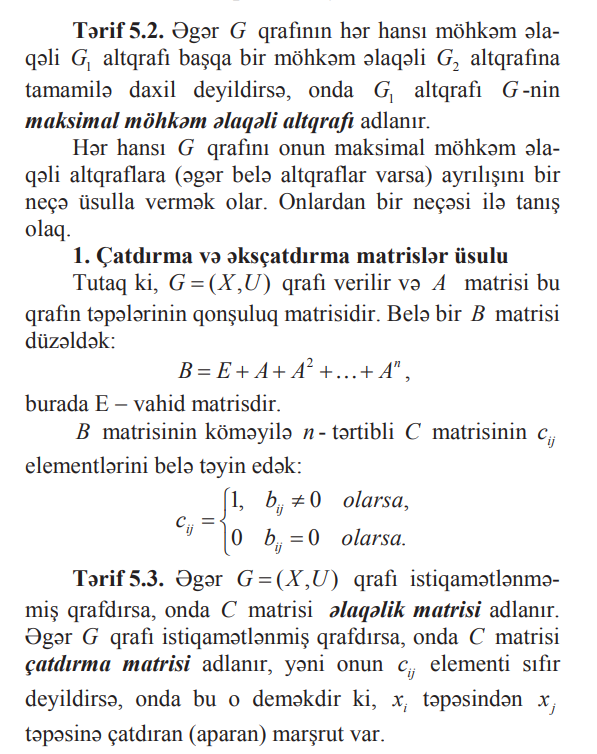
* **Başlanğıc Düyü Seçin:** DFS-ni başlamaq üçün bir başlanğıc düyün seçin.
* **Dərəcələr Üzərində Hərəkət Edin:** Başlanğıc düyündən başlayaraq, qrafdakı digər düyünlərə dərəcələr üzərində hərəkət edin.
* **Təmizləmə:** DFS əməliyyatını yerinə yetirdikdən sonra, digər başlanğıclarla davam edin və hər bir altqrafı yadda saxlayın.
* **Yenidən Başlayın:** DFS-ni hamı üçün başlanğıc düyün kimi qəbul edərək yenidən başlayın.

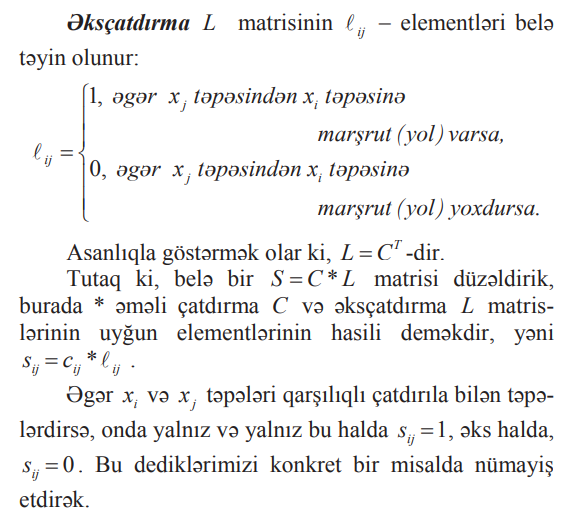
BFS (Genişliyə Gedən Axtarış) ilə Tapılması

BFS alqoritmi ilə maksimal möhkəm əlaqəli altqrafların tapılması aşağıdakı adımları izləyir:

* **Başlanğıc Düyü Seçin:** BFS-ni başlamaq üçün bir başlanğıc düyün seçin.
* **Yaxınlıq Tərəfə Hərəkət Edin:** Başlanğıc düyündən başlayaraq, yaxınlıq tərəfə hərəkət edin və bütün yaxınlıq düyünlərini tərəfə hərəkət edin.
* **Düyü Ziyarət:** Hər bir yaxınlıq düyününü ziyarət edin və yadda saxlayın.
* **Təmizləmə:** BFS əməliyyatını yerinə yetirdikdən sonra, digər başlanğıclarla davam edin və hər bir altqrafı yadda saxlayın.

## 79. Qraflarda ekstremal məsələlər və onların həll üsulları



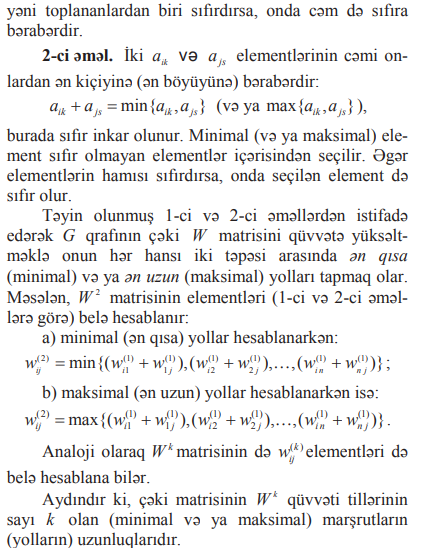
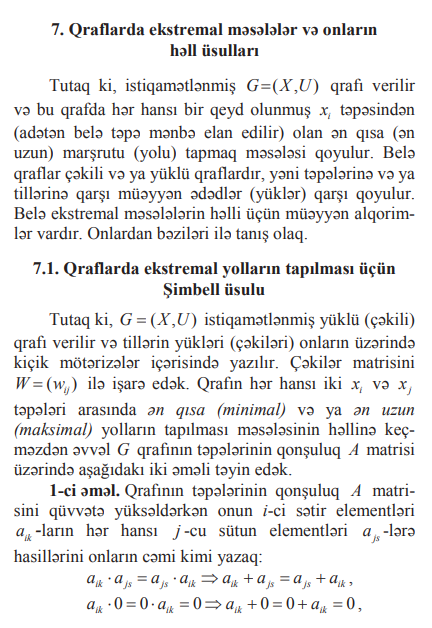


## 79. Qraflarda ekstremal məsələlər və onların həll üsulları

## 

Davamı:

## 80. Qraflarda ekstremal yolların tapılması



## 81. Qraflarda ən qısa yolun tapılması

Qraflarda ən qısa yolun tapılması problemi çox yayılmış bir problemdir və müxtəlif alqoritmlərlə həll olunur. Bu alqoritmlərin hər birinin fərqli üstünlükləri və çatışmazlıqları vardır. Aşağıda ən geniş istifadə olunan alqoritmlər və onların tətbiq sahələri verilmişdir:

1. Dijkstra Alqoritmi – müsbət çəkili qraflarda bir başlanğıc nöqtədən digər bütün nöqtələrə ən qısa yolu tapmaq üçün istifadə olunur.

İş prinsipi:

* Başlanğıc nöqtədən başlayaraq digər nöqtələrə olan məsafələri təxminən hesablayır və hər dəfə ən kiçik məsafəni seçir.
* Ağırlıqların mənfi olmaması halında çox effektiv işləyir.

2. Bellman-Ford Alqoritmi - mənfi çəkili qraflarda ən qısa yolu tapmaq üçün istifadə olunur.

İş prinsipi:

* Hər bir düyün üçün digər bütün düyünlərdən keçən potensial yolları qiymətləndirir.
* Mənfi ağırlıqlarla işləyə bilər və mənfi dövrləri aşkar edə bilir.

3. Floyd-Warshall Alqoritmi - qrafın hər bir cüt nöqtəsi arasında ən qısa yolu tapmaq üçün istifadə olunur.

İş prinsipi:

* Hər bir cüt nöqtə arasında ən qısa yolları təyin etmək üçün dinamik proqramlaşdırma üsulunu istifadə edir.
* Bütün nöqtələr arasındakı yolların uzunluğunu hesablamaq üçün uyğundur.

## 83. Maksimal yolun tapılması alqoritmi

Qraflarda maksimal yolun tapılması problemi, müəyyən bir qrafın ən uzun yolunu tapmaqdır. Bu, NP-tam məsələdir, yəni həlli üçün məlum bir polynomial zamanlı alqoritm yoxdur və problemi həll etmək çox vaxt tələb edir. Lakin, bəzi hallarda bu məsələni müəyyən məhdudiyyətlər daxilində həll edə biləcək yanaşmalar və alqoritmlər mövcuddur.

1. Dinamiki proqramlaşdırma (DAG üçün): Əgər qraf döngəsiz bir yönlü qrafdırsa (Directed Acyclic Graph, DAG), dinamik proqramlaşdırma yanaşmasından istifadə edə bilərsiniz.

2. DFS (Depth-First Search) istifadə etməklə sadə yanaşma: Bu metod, hər bir düyün üçün DFS tətbiq edərək bütün mümkün yolları yoxlayır və ən uzun yolu tapır. Lakin, bu yanaşma eksponensial vaxt tələb edir və böyük qraflar üçün səmərəli deyil.

3. TSP (Traveling Salesman Problem) təyin olunmuş məsələ kimi: Bu metod, qrafda hər bir düyünü ziyarət etməklə ən uzun yolu tapmağa çalışır. Bu metod, ümumi vəziyyətlər üçün çox vaxt tələb edir və səmərəli deyil.

## 84. Şəbəkədə ekstremal axın məsələləri

Şəbəkədə ekstremal axın məsələləri, qraf nəzəriyyəsi və s. sahələrdə əhəmiyyətli problemlərdən biridir. Bu məsələlər şəbəkədə mümkün olan maksimum axının tapılmasını, minimum kəsimi müəyyən etməyi və digər optimallaşdırma problemlərini əhatə edir. Bu problemlər geniş miqyasda tətbiq olunur, məsələn, kommunikasiya şəbəkələri, nəqliyyat sistemləri və enerji şəbəkələri kimi sahələrdə.

Ekstremal axın məsələlərinin əsas növləri bunlardır:

1. Maksimum Axın Problemi (Maximum Flow Problem)

2. Minimum Kəsim Problemi (Minimum Cut Problem)

3. Maksimum Axın - Minimum Kəsim Teoremi

Maksimum Axın Problemi - şəbəkədə mənbədən (source) məqsədə (sink) mümkün olan maksimum axını tapmaqdır.

Şəbəkə Tərifi:

- Qraf (G): Şəbəkə qrafı G = (V, E) şəklində təyin olunur, burada (V) qovşaqların (nodes) və (E) isə qövslərin (edges) çoxluğudur.

- Tutum (c): Hər bir qövsün maksimum keçirmə qabiliyyəti c(u, v) ilə ifadə olunur.

- Axın (f): Hər bir qövs üzrə axın f(u, v) ilə təyin olunur.

Şərtlər:

1. Axının Mühafizəsi Şərti: Hər bir qovşaq üçün daxil olan axın çıxan axına bərabər olmalıdır (mənbə və məqsəd qovşaqları istisna olmaqla).

2. Tutum Məhdudiyyəti: Hər bir qövs üzrə axın onun tutumundan artıq ola bilməz, yəni 0 ≤ f(u, v) ≤ c(u, v).

Məqsəd: Mənbədən məqsədə olan ümumi axını maksimumlaşdırmaq.

Ford-Fulkerson Alqoritmi: Ford-Fulkerson metodu maksimum axın problemini həll etmək üçün istifadə olunan məşhur bir alqoritmdir. Bu alqoritm iterativ olaraq artırıcı yollar (augmenting paths) taparaq axını artırır.

Minimum Kəsim Problemi - şəbəkəni mənbədən məqsədə doğru ayıran kəsimdəki qövs tutumlarının cəminin minimumunu tapmaqdır.

Teoremi: Maksimum axın - Minimum kəsim teoremi göstərir ki, şəbəkədə maksimum axın dəyəri minimum kəsim dəyərinə bərabərdir. Bu teorem axın və kəsim problemlərinin əlaqəsini ifadə edir.

## 86. Şəbəkədə minimal axının axtarılması

Şəbəkədə minimal axın problemi, maksimal axın problemindən fərqli olaraq, müəyyən bir şəbəkədə minimal axın miqdarını tapmağa yönəldilir.

1. Minimal Kəsik Problemi: Şəbəkədə mənbədən quyuya qədər olan kəsiklər arasında minimal kəsik tapılır. Maksimal axın-minimal kəsik teoreminə görə, minimal kəsik maksimal axının qiymətinə bərabərdir. Beləliklə, minimal kəsik problemi, maksimal axın alqoritmləri ilə həll edilə bilər.

2. Minimal Axın Problemi: Bu problem minimal axının həlli məqsədi ilə müəyyən edilir, lakin nəzərə alınmalıdır ki, axın problemi kontekstində minimal axın problemi əksər hallarda maksimal axın problemi kimi təsvir edilir. Belə ki, müəyyən bir şəbəkədə minimal axın adətən 0 olur (çünki axın olmaya bilər).

Əgər həqiqi bir minimal axın problemi ilə qarşılaşmaq istəyiriksə, bu, daha çox xüsusi bir tətbiq və şəbəkə kontekstində olmalıdır. Aşağıda minimal axın problemi üçün bəzi yanaşmalar verilmişdir.

1. Minimal Kəsik Problemi - Minimal kəsik problemi, şəbəkədə mənbədən quyuyə qədər olan kəsiklər arasında minimal kəsik tapmağı məqsəd qoyur. Maksimal axın-minimal kəsik teoreminə görə, minimal kəsik maksimal axının qiymətinə bərabərdir. Maksimal axın problemini həll etmək üçün istifadə olunan Ford-Fulkerson və ya Edmonds-Karp alqoritmləri minimal kəsik problemini də həll edir.

2. Minimal Axın Problemi (Minimal Xərcli Axın Problemi kimi) - Minimal axın problemini həqiqi bir kontekstdə həll etmək istəyirsinizsə, minimal xərcli maksimal axın problemi kimi də yanaşa bilərsiniz. Bu halda, şəbəkədə minimal xərcli axın tapılır.

Minimal Xərcli Maksimal Axın Problemi

Minimal xərcli maksimal axın problemini həll etmək üçün Potensial Metodu və ya Çevik Qiymətləndirmə Alqoritmi istifadə edilə bilər.

Minimal xərcli maksimal axın probleminin həllini MATLAB-da yazmaq üçün, “intlinprog” funksiyasından istifadə edə bilərsiniz. `intlinprog` funksiyası xətti proqramlaşdırma problemlərini həll etmək üçün istifadə olunur.

function [maxFlow, minCost] = minCostMaxFlow(capacity, cost, source, sink)

% Şəbəkənin düyünlərinin sayı

n = size(capacity, 1);

% Giriş dəyişənlərini təyin edin

f = cost(:); % xərc funksiyası

% Məhdudiyyətlər üçün bərabərlik matrisini qurun

Aeq = zeros(n, numel(cost));

beq = zeros(n, 1);

for i = 1:n

for j = 1:n

if i ~= j

Aeq(i, i + (j - 1) \* n) = 1;

Aeq(i, j + (i - 1) \* n) = -1;

end

end

end

% Mənbə və quyu üçün tələblər

beq(source) = sum(capacity(source, :));

beq(sink) = -sum(capacity(:, sink));

% Axın məhdudiyyətləri

lb = zeros(numel(cost), 1); % alt sərhəd 0-dır

ub = capacity(:); % üst sərhəd qrafdakı kapasitələrdir

% Tam ədədi proqramlaşdırma ilə minimal xərcli maksimal axını həll edin

options = optimoptions('intlinprog', 'Display', 'off');

[x, minCost] = intlinprog(f, 1:numel(cost), [], [], Aeq, beq, lb, ub, options);

% Maksimal axını hesablayın

maxFlow = sum(x(Aeq(source, :) == 1));

% Axın matrisi

flow = reshape(x, n, n);

disp('Axın matrisi:');

disp(flow);

end

% Qrafın təyin edilməsi

n = 6; % Düyünlərin sayı

capacity = [

0 10 5 0 0 0;

0 0 15 10 0 0;

0 0 0 0 10 0;

0 0 0 0 10 10;

0 0 0 0 0 10;

0 0 0 0 0 0

];

cost = [

0 2 6 0 0 0;

0 0 1 4 0 0;

0 0 0 0 2 0;

0 0 0 0 2 3;

0 0 0 0 0 1;

0 0 0 0 0 0

];

source = 1; % Mənbə düyün (1 indekslə)

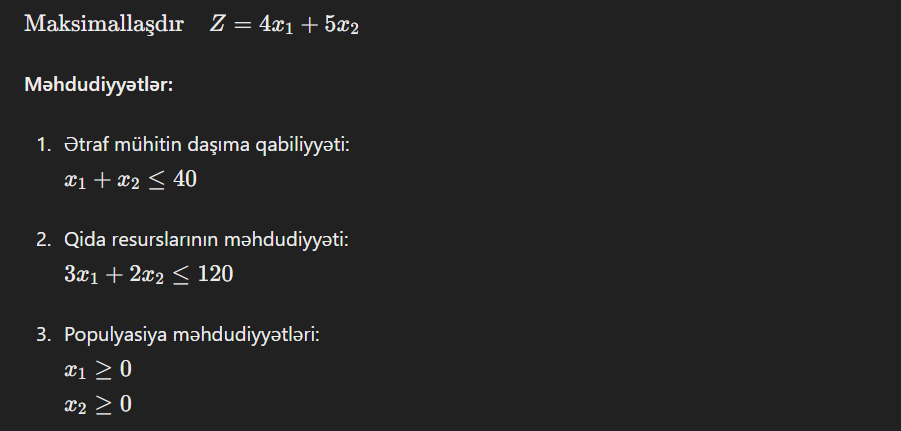
sink = 6; % Quyu düyün (6 indekslə)

[maxFlow, minCost] = minCostMaxFlow(capacity, cost, source, sink);

fprintf('Maksimal axın miqdarı: %d\n', maxFlow);

fprintf('Minimal xərcli axın: %d\n', minCost);

## 87. Bioloji populyasiya məsələsinin kompüterdə modelləşdirilməsi

**Məsələ:** Bir ərazidə iki fərqli növün (x1 və x2) populyasiyasını idarə etmək istəyirik. Məqsəd, hər iki növün populyasiyasının məhdudiyyətləri nəzərə alaraq maksimum fayda əldə etməkdir. Fayda funksiyası xətti olaraq verilir:

Copy code

% Məqsəd funksiyası əmsalları

f = [-4, -5];

% Məhdudiyyətlər matrisləri

A = [1, 1; 3, 2];

b = [40; 120];

% Aşağı məhdudiyyətlər (x1 və x2 üçün)

lb = [0, 0];

% Xətti proqramlaşdırma probleminin həlli

[x, fval] = linprog(f, A, b, [], [], lb, []);

% Nəticələrin göstərilməsi

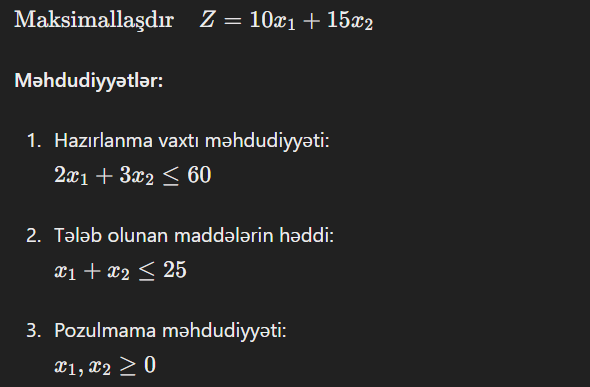
disp('Optimal dəyərlər:');

disp(['x1 = ', num2str(x(1))]);

disp(['x2 = ', num2str(x(2))]);

disp(['Maksimum fayda: ', num2str(-fval)]);

## 89. Rasion haqda məsələnin kompüterdə modelləşdirilməsi

**Məsələ:** Bir fermada iki fərqli qida maddəsinin (q1 və q2) hazırlanma prosesi müəyyənləşdirilmiş şərtlərə əsasən təyin edilməlidir. Məqsəd, istənilən qiymətə maksimum rasion təyin etməkdir. Rasion funksiyası aşağıdakı kimi verilir:

% Məqsəd funksiyası əmsalları

f = [-10, -15];

% Məhdudiyyətlər matrisləri

A = [2, 3; 1, 1];

b = [60; 25];

% Aşağı məhdudiyyətlər (x1 və x2 üçün)

lb = [0, 0];

% Xətti proqramlaşdırma probleminin həlli

[x, fval] = linprog(f, A, b, [], [], lb, []);

% Nəticələrin göstərilməsi

disp('Optimal dəyərlər:');

disp(['q1 = ', num2str(x(1))]);

disp(['q2 = ', num2str(x(2))]);

disp(['Maksimum rasion: ', num2str(-fval)]);

## 90. Eksperimental verilənlərin interpolyasıya üsulu ilə kompüterdə modelləşdirilməsi

**Məsələ:** Laboratuvar testlerindən elde edilmiş eksperimental verilənlər, bir polinom interpolasiya metodunu istifadə edərək modelin bir təqribi (approximation) ilə modelləşdirilməlidir. Verilənlər aşağıdakı cədvəldə göstərilir:

% Verilənlər

x = [0, 1, 2, 3, 4];

y = [1, 4, 11, 22, 37];

% Interpolasiya üsulu ilə polinomun təyini

p = polyfit(x, y, 4); % 4-ci dərəcəli polinom

% Polinomun kəsişmə nöqtələri

x\_interp = linspace(0, 4, 100); % 0-dan 4-ə qədər aralıqlarla interpolasiya et

y\_interp = polyval(p, x\_interp);

% Nəticələrin qrafik təsviri

figure;

plot(x, y, 'o', 'MarkerSize', 10, 'LineWidth', 2); % Eksperimental verilənlər

hold on;

plot(x\_interp, y\_interp, 'LineWidth', 2); % Interpolasiya ilə təyin edilmiş polinom

xlabel('x');

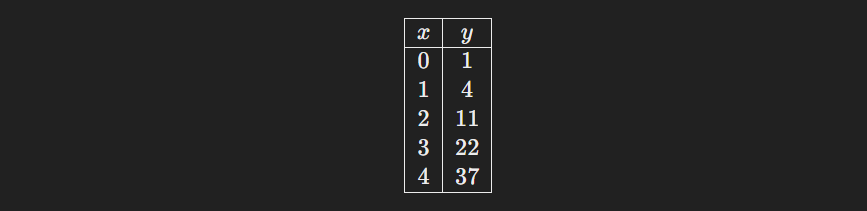
ylabel('y');

title('Eksperimental Verilənlərin Interpolyasiya ilə Modelləşdirilməsi');

legend('Eksperimental Verilənlər', 'Interpolasiya Polinomu', 'Location', 'northwest');

grid on;

## 91. Eksperimental verilənlərin ƏKKÜ ilə kompüterdə modelləşdirilməsi

**Məsələ:** Laboratuvar testlərindən əldə edilmiş eksperimental verilənlər, ƏKKÜ metodunu istifadə edərək modelin bir təqribi ilə modelləşdirilməlidir. Verilənlər aşağıdakı cədvəldə göstərilir:

% Verilənlər

x = [0, 1, 2, 3, 4];

y = [1, 4, 11, 22, 37];

% Lineer proqramlaşdırma üçün matrikslər

A = [x', ones(length(x), 1)]; % x və 1-lərdən ibarət matriks

b = y'; % y verilənləri

% Fayda funksiyası

f = [0, 0, -1]; % [x0, x1, -1]

% Aşağı məhdudiyyətlər (x0 və x1 üçün)

lb = [-Inf, -Inf, 0];

% Lineer proqramlaşdırma probleminin həlli

[x, fval] = linprog(f, [], [], A, b, lb, []);

% Polinomun kəsişmə nöqtələri

x\_interp = linspace(0, 4, 100); % 0-dan 4-ə qədər aralıqlarla interpolasiya et

y\_interp = x(1) \* x\_interp + x(2);

% Nəticələrin qrafik təsviri

figure;

plot(x, y, 'o', 'MarkerSize', 10, 'LineWidth', 2); % Verilənlər

hold on;

plot(x\_interp, y\_interp, 'LineWidth', 2); % ƏKKÜ ilə təyin edilmiş polinom

xlabel('x');

ylabel('y');

title('Eksperimental Verilənlərin ƏKKÜ ilə Modelləşdirilməsi');

legend('Eksperimental Verilənlər', 'ƏKKÜ Polinomu', 'Location', 'northwest');

grid on;

## 92. Membranın vibrasiyasının kompüterdə modelləşdirilməsi

**Məsələ:** Bir şirkət, fərqli materiallardan ibarət olan bir membranın optimal olaraq istifadə edilməsini planlaşdırır. Məqsəd, material mənbələrinin məhdudiyətləri və maliyyə limitləri altında membranın material istifadəsini maksimize etməkdir.

**Model:** Bu məsələdə, fərqli materiallardan ibarət bir membranın optimal material istifadəsini təmin etmək üçün bir lineer programlama modeli təqdim edəcəyik:

% Başlanğıc parametrlər

n = 3; % Material növlərinin sayı

m = 4; % İstifadə növlərinin sayı

% Material istifadə matrix

A = [

1, 0, 1; % Material 1 və 3 istifadə edilir

0, 2, 1; % Material 2 və 3 istifadə edilir

3, 2, 0; % Material 1, 2, 3 istifadə edilir

1, 1, 1 % Material 1, 2, 3 istifadə edilir ];

% Material limitləri

lb = zeros(n, 1); % Materialların minimum sayıları

ub = [Inf; 30; Inf]; % Material 2 limiti

% Maliyyə limiti

budget = 10000; % Maximal maliyyə limiti (örnəyin 10000 dollar)

% Hədəf funksiyası (material istifadəsinin maksimumu)

f = -ones(n, 1); % Materialların maksimal istifadəsi

% Lineer programlama həlli

[x, fval] = linprog(f, [], [], A', b, lb, ub);

% Optimal həllin göstərilməsi

disp('Optimal həll:');

disp(['Material istifadələri:']);

disp(x');

disp(['Optimal material istifadəsi: ', num2str(-fval)]);

## 93. Rəqsi hərəkətdə olan sistemin kompüterdə modelləşdirilməsi

Matlab-da rəqsi hərəkətdə olan bir sistemin modelləşdirilməsi üçün fərqli üsullardan istifadə edə bilərsiniz. Bu, mexaniki, elektrik, hidrolik və ya digər növ sistemləri əhatə edə bilər. Əsas məqsəd sistemin tənliklərini qurmaq və Matlab ilə bu tənlikləri həll etməkdir.

Rəqsi hərəkətdə olan mexaniki sistemlər adətən ikinci dərəcəli diferensial tənliklər ilə təsvir olunur. Məsələn, sadə harmonik osilatorun diferensial tənliyi aşağıdakı kimidir:

Burada:

- m- kütlə,

- c - amortizasiya əmsalı,

- k - bərkidici qüvvə əmsalı,

- x - yerdəyişmə,

- F(t) - xarici qüvvədir.

Matlab-da bu tənlikləri həll etmək üçün `ode45` kimi həll alqoritmlərindən istifadə olunur.

Aşağıda sadə harmonik osilatorun Matlab-da modelləşdirilməsi üçün nümunə göstərilmişdir.

1. Sistemin Parametrlərinin Təyin Edilməsi

% Parametrlər

m = 1; % kütlə (kg)

c = 0.5; % amortizasiya əmsalı (Ns/m)

k = 10; % bərkidici qüvvə əmsalı (N/m)

F0 = 1; % xarici qüvvənin amplitudası (N)

omega = 2; % xarici qüvvənin tezliyi (rad/s)

% Başlanğıc şərtlər

x0 = [0; 0]; % yerdəyişmə və sürət [x(0), x'(0)]

2. Sistem Tənliklərinin Təyin Edilməsi

function dxdt = harmonic\_oscillator(t, x, m, c, k, F0, omega)

dxdt = zeros(2,1);

dxdt(1) = x(2);

dxdt(2) = (F0 \* cos(omega \* t) - c \* x(2) - k \* x(1)) / m;

end

3. Sistemin Həll Edilməsi və Qrafik Təsviri

% Zaman intervalları

tspan = [0 10];

% Tənliklərin həlli

[t, x] = ode45(@(t,x) harmonic\_oscillator(t, x, m, c, k, F0, omega), tspan, x0);

% Nəticələrin qrafik təsviri

figure;

plot(t, x(:,1));

title('Yerdəyişmə vaxtdan asılı olaraq');

xlabel('Zaman (s)');

ylabel('Yerdəyişmə (m)');

grid on;

figure;

plot(t, x(:,2));

title('Sürət vaxtdan asılı olaraq');

xlabel('Zaman (s)');

ylabel('Sürət (m/s)');

grid on;

Açıqlama

1. Parametrlər: Bu hissədə sistemin parametrləri müəyyən edilir. Kütlə, amortizasiya əmsalı, bərkidici qüvvə əmsalı və xarici qüvvənin xüsusiyyətləri təyin olunur.

2. Sistem Tənlikləri: `harmonic\_oscillator` funksiyası sistemin diferensial tənliklərini müəyyən edir. Bu funksiya `ode45` alqoritmi tərəfindən istifadə olunur.

3. Tənliklərin Həlli və Qrafik Təsviri: `ode45` funksiyası sistem tənliklərini həll edir və nəticələri `t` (zaman) və `x` (həll vektorları) matrisləri kimi qaytarır. Daha sonra nəticələr qrafiklə təsvir olunur.

## 94. Mexaniki rəqsi hərəkətin Simulink-də modellədirilməsi

Simulink, Matlab tərəfindən təqdim edilən bir qrafik proqramlaşdırma mühiti olduğundan, mexaniki sistemlərin və rəqsi hərəkətlərinin dəqiq şəkildə modelləşdirilməsi üçün mükəmməl bir aləmdir. Bu mühitdə, diferensial tənlikləri təsvir edən model bloklarını və simulasiya bloklarını istifadə edərək sistemin davranışını vəziyyət simulyasiyası ilə izləmək olar. Bir rəqsi hərəkətli sistem Simulink-də modelləşdirmək üçün aşağıdakı addımları izləyə bilərsiniz:

Simulink-də Rəqsi Hərəkətli Sistem Modelləşdirilməsi

1. Simulink Modelinin Yaradılması

1. Matlab açın.

2. Komanda sətrində `simulink` yazın və Enter düyməsinə basın.

3. Açılan pəncərədə "File" -> "New" -> "Model" seçin.

2. Sistem Tənliklərinin Modelləşdirilməsi

1. Mekaniki Sistemlərin Təsviri: Mekaniki sistemləri təsvir etmək üçün Simulink-dəki "Mechanical System" və ya "Physical System" bloklarını istifadə edə bilərsiniz. Bu bloklar, kütlələri, amortizatorları, yayları, dəyərlərini, tərkibini və fiziki xüsusiyyətlərini quraşdırmaq üçün olanak verir.

2. Giriş və Çıxış Blokları: Sistemə giriş və çıxış etmək üçün "Input" və "Output" bloklarını əlavə edin. Məsələn, rəqsi hərəkətdə olan bir sistəm üçün, giriş qüvvələrini və çıxış pozisiyasını təyin etmək lazımdır.

3. Diferensial Tənliklərin Modelləşdirilməsi: Rəqsi hərəkətdə olan sistemin diferensial tənliklərini modelləşdirmək üçün "Integrator" bloklarını və ya təsvir edilmiş tənlikləri əlavə edin. Bu bloklar, sistemin hərəkətini (yer və sürət) təsvir edir.

3. Simulink Simulyasiyası

1. Simulasiya Parametrləri: Simulasiya vaxtı, integrator ayarları və digər parametrləri təyin edin.

2. Simulasiya Başlatmaq: "Run" düyməsinə basın və simulasiyanı başlatın.

3. Nəticələrin Təhlili: Simulink, simulyasiya zamanı sistemin davranışını göstərəcək. Nəticələri və sistem parametrlərini təhlil edin.

Nümunə

1. Harmonik Osilator Bloku Əlavəsi

Simulink-də harmonik osilator üçün integrator bloku əlavə edilir:

* "Library Browser" açılır.
* "Continuous" bölməsindən "Integrator" blokunu modelə sürükləyin.

2. Model Qurulması

* İntegrator bloku üçün parametrləri təyin edin

3. Əlavə Bloklar daxil edə bilərsiniz

## 95. Dinamiki proseslərin Simulink-də modelləşdirilməsi

Simulink, dinamik proseslərin və sistemlərin modelləşdirilməsi və simulasiyası üçün güclü bir alətdir. Bu platformada, elektromekanik sistemlərdən, otomatik nəzarət sistemlərinə, fizika sistemlərindən, sibernetikaya qədər bir çox sahədə dinamik prosesləri təsvir etmək mümkündür. Burada geniş bir yelpazədə proseslərin modelləşdirilməsi üçün əsas addımları təqdim edirəm.

Simulink-də dinamik prosesləri modelləşdirmək üçün əsas addımlar aşağıdakılardır:

1. Simulink Modelinin Yaradılması

1. Matlab açın.

2. Komanda sətrində `simulink` yazın və Enter düyməsinə basın.

3. Açılan pəncərədə "File" -> "New" -> "Model" seçin.

Bu addımdan sonra yeni bir Simulink modeli açılacaq.

2. Dinamik Proseslərin Təsviri

- İntegrator Blokları: Fiziki vəziyyətləri təsvir etmək üçün, məsələn, mekaniki sistemlərdə yer və ya sürət integratorları.

- Transfer Funksiyası Blokları: Elektromekanik sistemlər və otomatik nəzarət sistemlərində istifadə olunur.

- Stateflow Chart: Statik və dinamik davranışları təsvir etmək üçün.

- Logic və Math Operation Blokları: Sinyalların proseslənərək hesablamaq üçün.

- Matlab Function Blokları: Özəl funksiyalar yazmaq və blokda tətbiq etmək üçün.

3. Modeli Qurun

- Blokları sürükləyin və quraşdırın: Modelinizdə müxtəlif blokları modelinizə sürükləyin və lazımı parametrləri quraşdırın.

- Parametrləri təyin edin: Hər blokun parametrlərini modelinizə uyğun olaraq təyin edin. Məsələn, kütlə, amortizasiya, bərk, qüvvələr və digərləri kimi.

4. Simulasiya Parametrlərinin Təyin Edilməsi

- Simulasiya zamanı: Zaman aralığını, integrator və simulyasiya tipini təyin edin.

- Giriş və çıxışları təyin edin: Sistemə giriş və çıxış təsirləri daxil edin.

5. Simulasiya Başlatmaq

- Simulasiya düyməsinə vurun: Modelinizin simulyasiyasını başlatmaq üçün "Run" düyməsinə vurun.

6. Nəticələrin Təhlili və Optimizasiya

- Nəticələri analiz edin: Simulink nəticələri əldə edəcək və onları qrafiklə, 3D animasiyalarla və ya digər fəaliyyət izləmə vasitələri ilə təsvir edə bilərsiniz.

## 96. Qeyri-xətti məsələlərin kompüterdə modelləşdirilməsi

Qeyri-xətti məsələlərin kompüterdə modelləşdirilməsi, hər hansı bir məsələnin matəmatik təsvirini alaraq, onun alqoritmlər və proqramlaşdırma dilləri vasitəsilə həllini ifadə edir. Bu proses, həll üsuluna və problemin özünün kompleksliyinə görə fərqli təcrübələr və alqoritmlər istifadə edə bilər.

Qeyri-xətti məsələlərin kompüterdə modelləşdirilməsi üçün əsas addımlar aşağıdakılardır:

1. Məsələnin Formal Təsviri: Problem, matəmatik formulalar və şərtlərlə ifadə olunur. Bu, həll üsullarının və alqoritmlərin inkişafına zəmin yaradır.

2. Alqoritmlərin Hazırlanması: Məsələnin təsviri əsasında, həll üsulları və alqoritmlər qurulur. Bu addımda, məsələni daha asandırmaq, numerik metotlardan istifadə etmək və ya qraf teoriyası tətbiq etmək kimi müxtəlif yollar mövcuddur.

3. Proqramlaşdırma Dillərinin İstifadəsi: Alqoritmlər, proqramlaşdırma dilləri (məsələn, Python, MATLAB, C++, Java kimi) ilə kodlanır. Proqramlaşdırma dili, məsələnin tələblərinə uyğun olaraq seçilməlidir.

4. Simulyasiya və Analiz: Kodlanmış proqram, məsələnin simulyasiyasını aparır və nəticələri təhlil edir. Bu addımda, məsələnin həllinin effektivliyi, doğruluğu və praktik tətbiqi qiymətləndirilir.

Nümunə

Qeyri-xətti məsələ: Bir logistik şirkəti, ən optimal yolu tapmaq üçün şəhərlər arasında bir qrafda ən qısa yol problemi həll etməliyik.

1. Formal Təsvir: Şəhərlər arasındakı yol məsafələri və xüsusiyyətləri təyin edirik. Bu, bir qraf teorisinin məsələsidir.

2. Alqoritmlərin Hazırlanması: Dijkstra alqoritmi ilə ən qısa yol tapmaq üçün yalnız tək bir təyinat yolları kimi bir istinad deyil.

## 97. Dinamik prosesləri ifadə edən məsələlərin qoyuluşu və kompüterdə modelləşdirilməsi

**Məsələ:** Bir şirkət, dinamik bir prosesin idarəsini optimizasiya etmək istəyir. Məqsəd, prosesin tələblərinə əsasən optimal reaksiya həllini tapmaqdır.

**Model:** Bu məsələdə, bir kimya prosesində optimal material istifadəsini təmin etmək üçün bir lineer programlama modeli təqdim edərək prosesin dinamik məhsuldarlığını artırmağa çalışacağıq.

% Başlanğıc parametrlər

n = 3; % Kimya prosesində istifadə olunan material növlərinin sayı

m = 4; % İstehsal növlərinin sayı

% Material istifadə matrix

A = [

1, 0, 1; % Material 1 və 3 istifadə edilir

0, 2, 1; % Material 2 və 3 istifadə edilir

3, 2, 0; % Material 1, 2, 3 istifadə edilir

1, 1, 1 % Material 1, 2, 3 istifadə edilir

];

% Tələblər vektoru

b = [10; 20; 15; 30]; % İstehsal tələbləri

% Material limitləri

lb = zeros(n, 1); % Materialların minimum sayıları

ub = [Inf; 30; Inf]; % Material 2 limiti

% Lineer programlama həlli

[x, fval] = linprog(-b, A', lb, [], [], zeros(n, 1), []);

% Optimal həllin göstərilməsi

disp('Optimal həll:');

disp(['Material istifadələri:']);

disp(x');

disp(['Optimal istehsal: ', num2str(-fval)]);

## 98. Təsadüfi proseslərin kompüterdə modelləşdirilməsi

**Məsələ:** Bir hissənin qalıq ömrünü təqvimi, təsadüfi proseslərə əsaslanan bir model istifadə edərək təhlil etmək istəyirik. Məqsəd, hissənin əmək vaxtını və səhv nisbətini aktiv olaraq nəzərdə tutmaqdır.

**Model:** Aşağıdakı MATLAB kodu, təsadüfi bir hissənin qalıq ömrünü modelləmək üçün Monte-Karlo simulasiya metodunu təsvir edir:

% Başlanğıc parametrlər

mu = 1000; % Ortalama ömür (saat)

sigma = 50; % Standart sapma

% Simulasiya parametrləri

n\_sim = 1000; % Simulasiya sayı

% Təsadüfi hissə ömrü üçün Monte Carlo simulasiyası

lifetime = normrnd(mu, sigma, n\_sim, 1);

% Simulasiya nəticələrinin göstərilməsi

disp('Simulasiya nəticələri:');

disp(['Ortalama ömür: ', num2str(mean(lifetime)), ' saat']);

disp(['Standart sapma: ', num2str(std(lifetime)), ' saat']);

% Histogram göstərilməsi

figure;

histogram(lifetime, 30, 'Normalization', 'probability');

xlabel('Hissə ömrü (saat)');

ylabel('Göstərilən tezlik');

title('Təsadüfi Hissə Ömrü Histogramı');

## 99. Optimal planlaşdırma məsələlərinin modelləşdirilməsi

**Məsələ:** Bir depo yönetim şirkəti, depolardakı malları mümkün olan en optimal şekildə yönetmek istəyir. Məqsəd, malların depolar arasında transferini minimum məsafə və zamanla həyata keçirməkdir.

**Optimizasiya Modeli:** Aşağıdakı MATLAB kodu, depo yönetim şirkətinin malların optimal transferini təmin etmək üçün lineer programlama ilə təsvir edilmişdir:

% Malların sayı və depo sayıları

n = 4; % Malların sayı

m = 3; % Depo sayıları

% Mallar və depo məlumatları

mallar = [20, 15, 25, 30]; % Malların həcmi (m³)

depo\_mesafeleri = [

10 15 20;

12 14 18;

8 10 12;

16 18 20

]; % Depo aralığı (km)

% Lineer programlama probleminin təyin edilməsi

f = reshape(depo\_mesafeleri, [], 1); % Amaç funksiyası: Depo məsafələrini tek sütuna sıralamaq

A = kron(eye(n), ones(1, m)); % Kısıt matriksi: Hər bir malların depo sayı qədər hərəkət edə biləcəyi məsafələr

b = mallar'; % Kısıt vektoru: Hər bir malın həcmi

% Lineer programlama həlli

lb = zeros(n\*m, 1); % Değişkənlərin minimum sərhədləri

ub = ones(n\*m, 1); % Değişkənlərin maksimum sərhədləri

[x, fval] = linprog(f, [], [], A, b, lb, ub);

% Optimal həllin göstərilməsi

disp('Optimal həll:');

disp(['Depo məsafələri:']);

disp(reshape(x, m, n)');

disp(['Optimal məsafə: ', num2str(fval)]);

## 100. Nəqliyyat məsələsinin kompüterdə modelləşdirilməsi

**Məsələ:** Bir nakliyyat şirkəti, bir dizi yükü fərqli mənzillərə daşımaq üçün optimal rota planlaması etmək istəyir. Məqsəd, nakliyyatın ümumi məsafəsini və dolanma vaxtını minimuma endirməkdir.

% Yük və mənzil məlumatları

n = 5; % Yük sayı

m = 3; % Mənzil sayı

% Yük və mənzil parametrləri

yukler = [10, 20, 15, 25, 30]; % Yük həcmi (ton)

menzillər = [50, 30, 40]; % Mənzil məsafələri (km)

% Kəşf matriksi (km)

kesf = [

10 15 20 25 30;

12 14 18 21 24;

8 10 12 16 20 ];

% Lineer programlama probleminin təyin edilməsi

f = reshape(kesf, [], 1); % Amaç funksiyası: Kesf matriksini tek sütuna sıralamaq

A = kron(eye(n), ones(1, m)); % Kısıt matriksi: Hər bir yükün mənzil sayı qədər hərəkət edə biləcəyi kəşf sayı

b = menziller'; % Kısıt vektoru: Hər bir mənzilin məsafəsinə bərabər və ya kiçik olmalıdır

% Lineer programlama həlli

lb = zeros(n\*m, 1); % Değişkənlərin minimum sərhədləri

ub = ones(n\*m, 1); % Değişkənlərin maksimum sərhədləri

[x, fval] = linprog(f, [], [], A, b, lb, ub);

% Optimal həllin göstərilməsi

disp('Optimal həll:');

disp(['Kesf matriksi:']);

disp(reshape(x, m, n)');

disp(['Optimal məsafə: ', num2str(fval)]);