

Практикалық жұмыс 8

Ең ара қашықтық есебін параллель есептеу

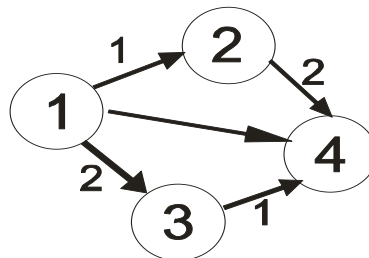
Мақсаты: Студенттерге ең кіші ара қашықтық туралы есепті параллель есептеп үйрету.

Кілттік сөздер: параллель есептеу, граф, ең кіші ара қашықтық

Оқытудың әдістемесі мен формасы: практикалық жұмыс

Бүтін санды программалау есебінің арасында көңіл бөлетін бір топ есептерді тандап алуға болады. Бұл есептердің экстремумдық шешімін табу үшін берілген сандар жиынтығын алмастыру немесе комбинациялау арқылы түрлендіреміз. Осындай есептердің бірі ең кіші ара қашықтық туралы есебі. Осы мысалда ең кіші ара қашықтық есебін параллель есептеуге болатындығын көрсетеміз. Ол үшін алдымен ең кіші ара қашықтық туралы есебін түсіндіріп кетсек:

1. Граф дегеніміз төбелерді қосатын доғалар жиыны.
2. Бағытталған қырлары болатын графты тор деп атайды. Сонымен ең кіші ара қашықтық есебінде бір шығу нүктесі және бір жинау нүктесі болатын торды қарастырады.



Сурет 66–Граф

Егер осы тордың доғаларын ара қашықтық ретінде қарастырсақ, онда ең кіші ара қашықтық туралы есеп пайда болады.

Есептің шарты:

1. Тор түрінде беріледі.
2. $\left(i \begin{array}{c} C_{i,j} \\ j \end{array} \right)$ доғалар жиынтығымен беруге болады.

$$S : \{ 1 \overset{1}{\rightarrow} 2, 1 \overset{2}{\rightarrow} 3, 2 \overset{2}{\rightarrow} 4, 3 \overset{1}{\rightarrow} 4, 3 \overset{1}{\rightarrow} 4, 1 \overset{2}{\rightarrow} 4 \}$$

Ең кіші ара қашықтық есептің математикалық моделі:

$$F = \sum_{(i,j)} \sum_{\in S} C_{i,j} \cdot X_j \rightarrow \min \quad (1)$$

$$X_{i,j} = 1, (i, j) \text{ тиімді матрица, } 0 - \text{ кері жағдай} \quad (2)$$

Сонымен (1) - (2) есебін шығару үшін келесі алгоритмді қолданамыз:

1. $y_1=0$ бір шығу нүктесі.

2. Кез-келген $y_1=\min (y_i+C_{i,j})$ - ,барлық қалған $y_1 \ j = \overline{2,n}$ - табамыз.

3. Табылған y_1 – мәндерін тордың сәйкес төбесіне қоямыз.

4. Егер $y_i - y_i = C_{i,j}$ болса, $\Rightarrow (i, j)$ доғаларды тиімді маршрутқа кіргіземіз.

Мысалы: 7 төбесі бар тор берілсін.

$(1-2), (2-7), (1-4), (3-4), (3-5), (4-5), (4-6), (5-7), (6-7), (1-3), (3-2), (2-5), (5-6)$

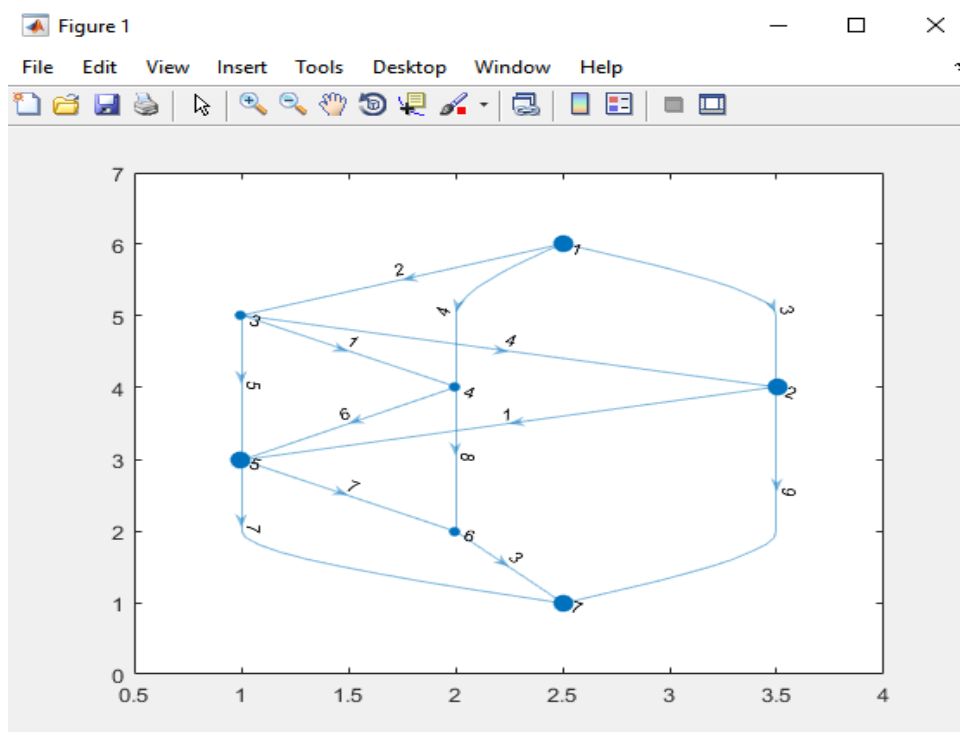
Бұл есепті параллель есептеп шығаруды көрсетіп тиімділігін анықтау үшін Matlab 2015 b жүйесінде алдымен тізбектей кейін параллель есептеу арқылы U шығу нүктесінен V нүктесіне дейінгі ең қысқа жолды табуға және нәтижесін графиктік түрғыда көруге болады. Есептеу Intel Core i5, 2.5GHz 4 ядролы процессорында тізбектей, кейін параллель есептелді.

Тізбектей есептеу	Параллель есептеу программасы
<pre>%from s = 1 2 1 3 3 4 4 5 6 1 3 2 5]; %to t = [2 7 4 4 5 5 6 7 7 3 2 5 6]; % distance weights = [3 9 4 1 5 6 8 7 3 2 4 1 7]; D = digraph(s, t, weights) tic shortPath = shortestpath(D, 1, 7) p = plot(D,'EdgeLabel',D.Edges.Weight) highlight(p, shortPath)</pre>	<pre>%from s =[1 2 1 3 3 4 4 5 6 1 3 2 5]; %to t = [2 7 4 4 5 5 6 7 7 3 2 5 6]; % distance weights = [3 9 4 1 5 6 8 7 3 2 4 1 7]; D = digraph(s, t, weights) D = parallel.pool.Constant(D); tic F = digraph(s, t, weights) p= plot(F,'EdgeLabel',F.Edges.Weight) highlight(p, shortPath)</pre>

<pre> clc shortPath = shortestpath(D, 1, 7) toc </pre>	<pre> clc [shortPath,d] = shortestpath(F, 1, 7) toc </pre>
<pre> p = GraphPlot with properties: NodeColor: [0 0.4470 0.7410] MarkerSize: 4 Marker: 'o' EdgeColor: [0 0.4470 0.7410] LineWidth: 0.5000 LineStyle: '-' NodeLabel: {'1' '2' '3' '4' '5' '6' '7'} EdgeLabel: {'3' '2' '4' '1' '9' '4' '1' '5' '6' '8' '7' '7' '3'} XData: [2.5000 3.5000 1 2 1 2 2.5000] YData: [6 4 5 4 3 2 1] Show all properties </pre>	<pre> D = digraph with properties: Edges: [13x2 table] Nodes: [7x0 table] F = digraph with properties: Edges: [13x2 table] Nodes: [7x0 table] p = GraphPlot with properties: NodeColor: [0 0.4470 0.7410] MarkerSize: 4 Marker: 'o' EdgeColor: [0 0.4470 0.7410] LineWidth: 0.5000 LineStyle: '-' NodeLabel: {'1' '2' '3' '4' '5' '6' '7'} EdgeLabel: {1x13 cell} XData: [2.5000 3.5000 1 2 1 2 2.5000] YData: [6 4 5 4 3 2 1] </pre>

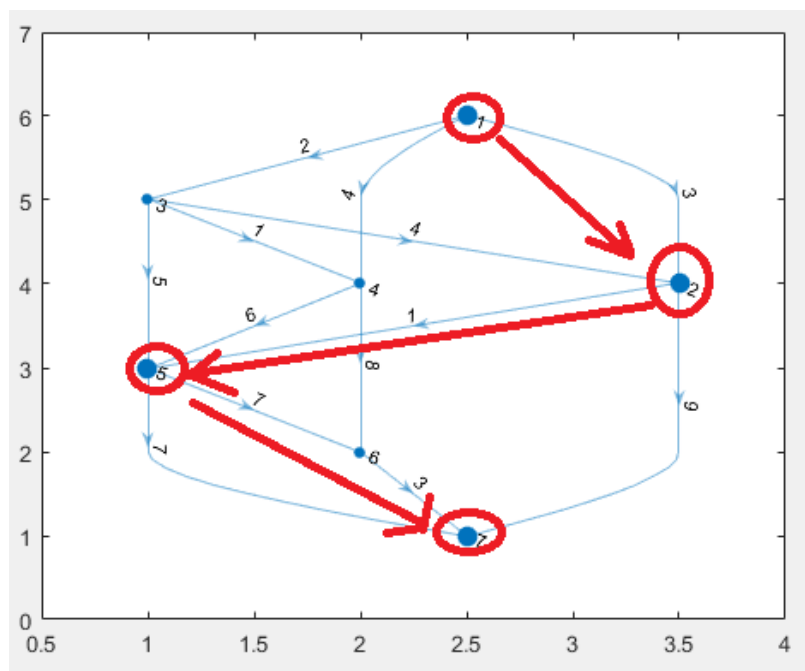
	Show all properties
shortPath = 1 2 5 7 d = 11 Есептеу уақыты 0.072378 секунд.	shortPath = 1 2 5 7 d = 11 Есептеу уақыты 0.043587 секунд.

Келесі суреттегі графика (67-сурет). Ең қысқа жолдары көрсетіледі.



Сурет 67- Ең қысқа ара қашықтық

Есептеу уақыттары әр түрлі, бірдей нәтижелер мен графиктертер алдық. Яғни, ең қысқа жол = 1 2 5 7 болса , жол ара қашықтығы = 11 болды.



Сурет 68- Ең қысқа ара қашықтық тиімді жолдары

Есептеу уақыттарын келесі кестеден көре аласыздар.

Кесте 7- Есептеу уақыттары

Тақырыбы	Есептеу уақыттары	Жылдамдату коэффициенті
Тізбектей есептеу	0,0724	
Параллель есептеу	0,0436	1,66

Осыдан ең кіші ара қашықтық есебін де параллель есептеуге болатындығын көре алдық. Түйіндері одан да көп есептер болса тиімділік одан да жақсы артатыны белгілі.

Тапсырма

Төмендегі интегралды өз варианттарыңыз бойынша таңдап алдымен local, кейін баптаған кластерлеріңізде есептеп нәтижесін салыстырыңыз.

- 1) $(1-2), (1-4), (1-3), (2-4), (2-5), (4-3), (4-5), (3-5).$
- 2) $(5-2), (5-4), (5-6), (6-4), (6-3), (4-1), (3-4), (3-1), (2-4), (2-1).$
- 3) $(1-2), (1-4), (1-5), (1-3), (2-4), (3-5), (3-6), (5-4), (6-5), (6-4).$
- 4) $(2-1), (2-5), (2-6), (3-4), (3-6), (1-6), (1-4), (5-3), (5-6), (6-4).$
- 5) $(3-1), (3-2), (3-4), (3-6), (1-6), (2-4), (2-5), (4-6), (5-4), (5-6).$

Бақылау сұрақтары

1. Ең қысқа ара қашықтық есебі параллель есептеу әр кезде тиімді бола ма?
2. Берілген мысалда есептеудің жедел орындалуы қандай шамамен есептеледі?