JEGYZŐKÖNYV

Mesterséges Intelligencia

Extra féléves feladat

Flowshop 2 gépre, Tabu keresés, Johnson Algoritmus

Készítette: **Bartók-Balog Péter**
Neptunkód: **ASNQPH**

Tartalom

A feladat leírása:

Két gép ütemezése Flow-Shop-al, valamint a feladat megoldása tabukereséssel, illetve Johnson algoritmussal. A flow-shop feladat egy ún. egyutas, többoperációs gyártásütemezési feladat. A gyártásütemezési feladatok formális leírására az alfa, béta, gamma jelölést alkalmazzuk (α|β|γ). Az erőforrások az ütemezési időszakban folyamatosan rendelkezésre állnak. Az erőforrások egyszerre csak egy munkán dolgoznak. A munkák legkorábbi indítási időpontja nulla (bármikor indíthatóak). Minden egyes munkához adott m számú operáció tartozik, melyeknek pontosan ismert a végrehajtási ideje: $p_{i,j,i} \in 1, 2, \ldots, n, j \in 1, 2, \ldots, m$. Az operációk végrehajtási sorrendje kötött és minden munka esetében azonos. Az operációk végrehajtása nem szakítható meg. A gépek között a munkák várakozhatnak, a műveletközi tárolók mérete nem korlátos. Az ütemezés célja az utolsóként elkészülő munka befejezési időpontjának minimalizálása.

Ezt a feladatot szükséges véletlenszámmal legenerálni a következő szempontok szerint:

- a munkák száma legyen 20,50,100
- a gépek száma legyen 2
- A legenerált feladatok legyenek perzisztensen tárolva (vagy seedhez kötéssel procedurálisan generálva).

A feladatot a Tabu kereséssel kell megoldani, és Johnson algoritmussal kell megoldani. A tabu keresés az egyszerű szomszédsági keresésre épül, azzal a különbséggel, hogy a keresés során bevezetünk egy tabu listát, melyben tároljuk, hogy mik azok a bázisok, ahol már jártunk. Ezt determinisztikus elfogadási kritériumnak nevezzük. Ez a lista változásokat vagy eredményeket is tarthat nyilván. Ehhez a tabulistához általában rendelünk egy maximális elemszámot. Ha ez az elemszám nagyon nagy, akkor a keresés túlságosan korlátozottá válik, ha túl kicsi, akkor ismétlődő keresési útvonal alakulhat ki. A Johnson algoritmus a két gépes flow-shop feladatokra optimálist ad. Ez a flow-shop feladat legtágabb olyan részfeladata, amire van egzakt megoldó algoritmus.

A megoldás lépései:

Elsősorban egy kezdeti tömb generálása, a munkák és gépek különböző darabszámára, amikre majd lefut a program. Bemeneti kezdő sor létrehozása, majd annak randomizálása, és eltárolása, majd egy iteráció létrehozása, ami vezérli majd a tabu keresés hosszát. Ezt követi a kezdeti sorhoz tartozó szomszédok keresése, és azok tárolása egy listában. Majd a különböző job-ok megfelelő átszervezése, hogy azok rendesen index szerint legyenek tárolva a táblában. Ezután történik Ti meghatározása, amihez létrehozunk két mátrixot, az egyik a befejezési, a másik pedig a várakozási idő fogja tárolni, és végül majd a kettőt összeegyeztetve jön ki a végeredmény. A befejezési időt mindig az előző feladat befejezési idejétől kezdjük, és így haladunk végig az egyes állomásokon, majd leellenőrizzük, hogy a befejezés táblába ténylegesen a jó értékek kerültek-e, és eltároljuk azokat egy megoldás táblába, majd visszaterünk a kiszámolt Ti értékkel. A végére marad a tabu keresés megírása, ami egy kezdeti iteráció megadásával történik, majd megadunk két listát, amiben a szomszédok, illetve az elhagyott állomásokat fognak kerülni, valamint megadjuk a tabu lista maximális méretét. Meghatározzuk az egyes szomszédokhoz rendelt eredményt, majd oda lépünk tovább, ahol az eredmény a legoptimálisabb, az elhagyott szomszédot pedig felvesszük a tabu listába. Ha a tabu lista eléri a maximum értéket, kitöröljük annak legelső, 0.-ik indexű elemét, és haladunk tovább. A johnson algoritmus létrehozza a kezdeti mátrixot 2 gépre nézve, majd kiválasztja belőle a

minimumot, majd ha 1-es gépen fut valami, akkor a sor végére teszi, ha 2-es gépen fut valami, akkor a sor elejére, így halad tovább amíg el nem fogynak a munkák.

Választott nyelv, keretrendszer, könyvtárak:

A program C sharp-ban írodott, az alapértelmezett .net keretrendszerrel, és az alapértelmezett könyvtárakkal, technológiával.

Feladathoz tartozó kérdések:

Adott munkaszámnál (20,50,100) vizsgálja meg a kereső algoritmus paramétereinek változtatásával elért hatást:

20 gépnél tabumax csökkentése esetén nőtt az iterációk száma, növelés esetén ugyanannyi volt. 50 gépnél tabumax csökkentése esetén csökkent, növelésnél ugyanannyi volt. 100 gépnél tabumax csökkentéssel ugyanannyi volt, növelésnél ugyanannyi maradt.

Hány iteráció kell az optimum megtalálásához?

- 20 gépnél, ha a tabumax = 35, akkor 1 iteráció kellett
- 20 gépnél, ha a tabumax= 20 akkor 1 iteráció kellett
- 50 gépnél, ha a tabumax = 35 akkor 2 iteráció kellett
- 50 gépnél, ha a tabumax = 20 akkor 2 iteráció kellett
- 100 gépnél, ha a tabumax = 35 akkor 2 iteráció kellett
- 100 gépnél, ha a tabumax = 20 akkor 2 iteráció kellett

Hány iterációszám kell ahhoz, hogy a relatív eltérés ugyanannyi legyen 20 munkáról 50 munkára ugrás esetén?

A relatív iteráció 1-el nőtt.

Futási eredmények:

Munkák száma = 20

Gépek száma = 2

9 7

5 5

9 9

9 6

8 8

6 7

6 8

9 5

6 8

8 5

9 6

5 6

8 5

5 9

7 5

8 5

9 7

8 6

9 6

5 6

2, 12, 14, 20, 6, 7, 9, 5, 3, 17, 1, 19, 18, 11, 4, 16, 15, 13, 10, 8,

Az iterációk amibe került: 0 --Tabumax = 35

Az iterációk amibe került: 0 --Tabumax = 20

Munkák száma = 50

Gépek száma = 2

6 9

9 8

9 9

5 6

8 5

8 7

9 7

6 7

7 5

8 6

6 8

7 5

6 5

5 5

9 8

5 9

7 9

7 6

5 9

5 8

5 5

8 7

7 7

5 7

5 7

5 9

5 6

6 8

9 5

5 6

7 7

9 7

5 7

8 6

6 7

5 8

8 6

8 7

9 8

5 5

8 5

7 5

8 6

6 6

5 8

7 7

5 9

5 7

7 6

5 9

4, 14, 16, 19, 20, 21, 24, 25, 26, 27, 30, 33, 36, 40, 45, 47, 48, 50, 1, 8, 11, 28, 35, 44, 17, 23, 31, 46, 3, 39, 15, 2, 38, 32, 22, 7, 6, 49, 43, 37, 34, 18, 10, 42, 41, 29, 13, 12, 9, 5,

Az iterációk amibe került: 1 --Tabumax = 35

Az iterációk amibe került: 1 --Tabumax = 20

Munkák száma = 100

Gépek száma = 2

8 9

9 5

9 9

6 5

5 7

9 7

7 8

6 9

7 5

6 7

8 6

6 6

5 8

8 5

8 6

6 8

7 8

8 8

7 5

9 7

8 6

8 7

5 8

5 7

7 9

7 5

7 6

6 9

7 9

8 9

6 9

7 5

6 9

8 6

7 9

6 8

7 8

5 6

6 8

6 7

9 9

9 6

6 6

8 5

6 7

8 5

8 7

5 6

7 6

5 9

5 7

7 8

6 9

8 8

9 7

6 6

7 9

7 9

7 6

6 9

5 6

6 8

9 8

9 7

9 5

5 6

8 5

7 7

8 5

7 9

8 6

7 5

8 8

8 6

7 6

8 8

9 7

8 9

7 6

9 5

8 8

8 6

7 5

5 7

7 7

5 5

8 6

7 7

7 7

6 9

6 8

8 5

7 5

8 6

5 8

5 5

7 5

8 7

6 9

6 6

5, 13, 23, 24, 38, 48, 50, 51, 61, 66, 84, 86, 95, 96, 8, 10, 12, 16, 28, 31, 33, 36, 39, 40, 43, 45, 53, 56, 60, 62, 90, 91, 99, 100, 7, 17, 25, 29, 35, 37, 52, 57, 58, 68, 70, 85, 88, 89, 1, 18, 30, 54, 73, 76, 78, 81, 3, 41, 63, 98, 77, 64, 55, 47, 22, 20, 6, 94, 87, 82, 79, 75, 74, 71, 59, 49, 42, 34, 27, 21, 15, 11, 97, 93, 92, 83, 80, 72, 69, 67, 65, 46, 44, 32, 26, 19, 14, 9, 4, 2,

Az iterációk amibe került: 1 --Tabumax = 35

Az iterációk amibe került: 1 --Tabumax = 20