Miskolci Egyetem
Gépészmérnöki és Informatikai Kar
Informatikai Intézet
Alkalmazott Informatikai Intézeti Tanszék

# ET

# Erőforrás tervezés
# Resource Planning

2017/18 2. félév

4. rész

Dr. Kulcsár Gyula
egyetemi docens

# Tartalom

1. Párhuzamosan működő teljesen egyenértékű erőforrások jellemzői (P modell).
2. A $P \,||\, \sum_i C_i$ feladat egzakt megoldása.
3. A $P \,||\, C_{max}$ feladat heurisztikus megoldása.
4. A $P \,|\, d_i \,|\, L_{max}$ feladat heurisztikus megoldása.
5. A $P \,|\, p_i = 1;\ r_i = \text{integer};\ d_i = \text{integer} \,|\, L_{max}$ feladat egzakt megoldása.

# Párhuzamosan működő teljesen egyenértékű erőforrások ütemezése

# A P modell jellemzői

- az erőforrások teljesen egyenértékűek,
- az erőforrások az ütemezési időszakban folyamatosan rendelkezésre állnak,
- egy erőforrás egyszerre csak egy munkán dolgozhat,
- egy munkán egyszerre csak egy erőforrás dolgozhat,
- a munkák legkorábbi indítási időpontja nulla:
  $r_i = 0$ $(i=1, 2, \ldots, n)$,
- minden egyes munkához egyetlen operáció tartozik, melyeknek pontosan ismert a végrehajtási ideje:
  $p_i$ $(i=1, 2, \ldots, n)$,
- az operációk végrehajtása nem szakítható meg.

# A P modell jellemzői

A megoldás (ütemterv) reprezentálása:

- Minden erőforrásnak saját ütemezési vektora van.
- A vektorok hossza megegyezik az erőforrásokhoz rendelt munkák (operációk) számával.
- Minden egyes vektor tartalmazza az adott erőforráson végrehajtandó munkák indítási sorrendjét.

# Párhuzamosan működő erőforrások ütemezése MSPT-szabály alkalmazásával

# Az ütemezés célja

- A munkák befejezési időpontjainak összege legyen minimális.

Az ütemezési modell formális leírása:

$$P \,||\, \sum_i C_i$$

# Az MSPT-algoritmus

# Az MSPT algoritmus

Modified Shortest Processing Time, MSPT
"módosított legrövidebb műveleti idejű munka előre„:

- Rendezzük a munkákat az SPT szabály alkalmazásával a műveleti idő szerint nemcsökkenő sorrendbe.
- Ezután képezzünk az erőforrások (gépek) számának megfelelő hosszú csoportokat.
- Az első csoport munkái az elsők rendre az egyes gépeken, a második csoport munkái a másodikok rendre az egyes gépeken és így tovább.

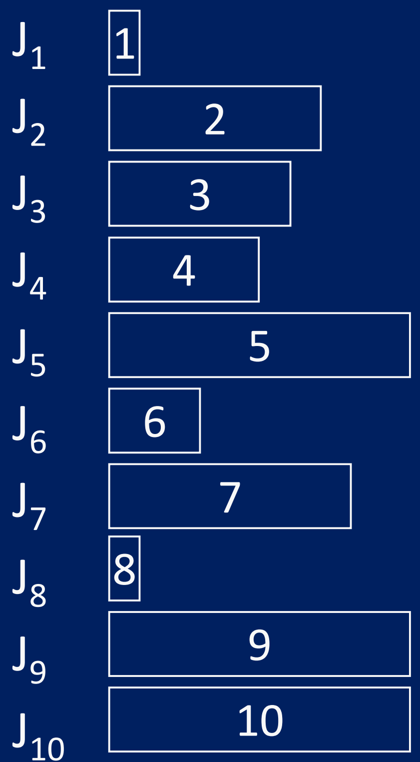
# Optimális megoldás

- Az MSPT szabály a $P||\sum C_i$ feladat optimális megoldását adja.
- Az állítás bizonyítása megtalálható az alábbi könyvben:
  Peter Brucker, Scheduling Algorithms, Springer, 2007.

Megjegyzés:

- Vegyük észre, hogy az adott feltételek mellett a $\sum C_i$ célfüggvény minimalizálása egyben az átlagos átfutási időt is minimalizálja.

# Illusztratív példa az MSPT-algoritmus működésére

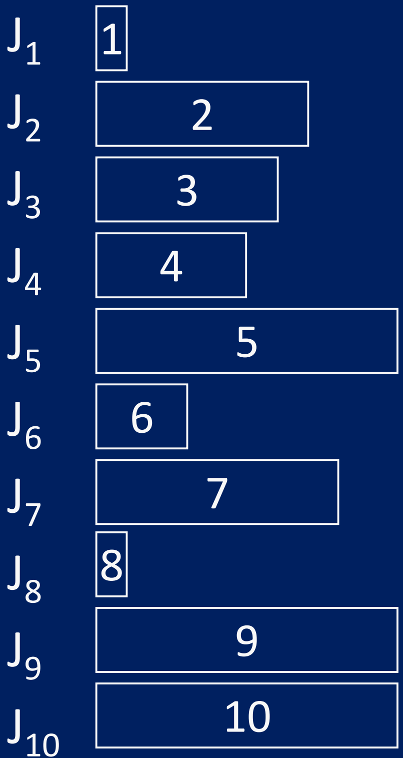



| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 1 | 7 | 6 | 5 | 10 | 3 | 8 | 1 | 10 | 10 |

$S_{M1}=\{\quad\}$

$S_{M2}=\{\quad\}$

$S_{M3}=\{\quad\}$

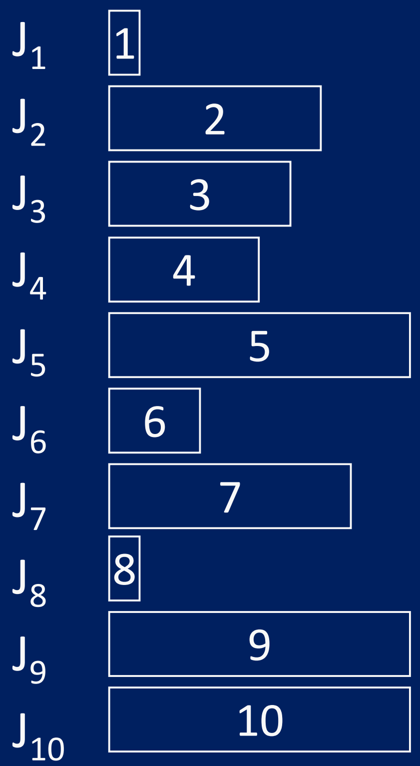



SPT: { 1, 8, 6, 4, 3, 2, 7, 5, 9, 10 }

$S_{M1}$={ }

$S_{M2}$={ }

$S_{M3}$={ }

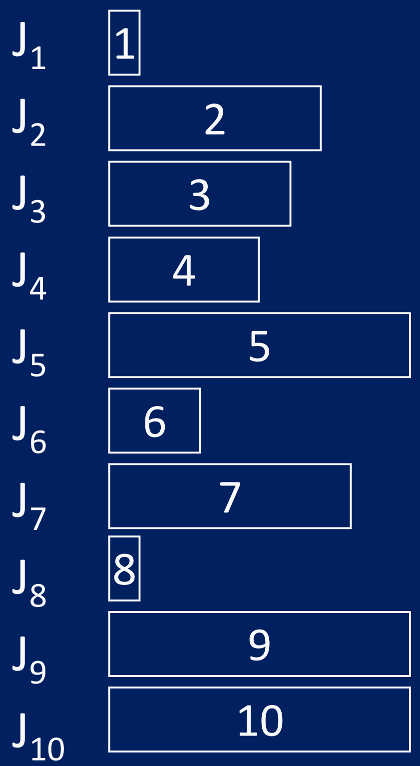



SPT: { $\underbrace{1, 8, 6}$, 4, 3, 2, 7, 5, 9, 10 }

$S_{M1}$={ 1 }

$S_{M2}$={ 8 }

$S_{M3}$={ 6 }

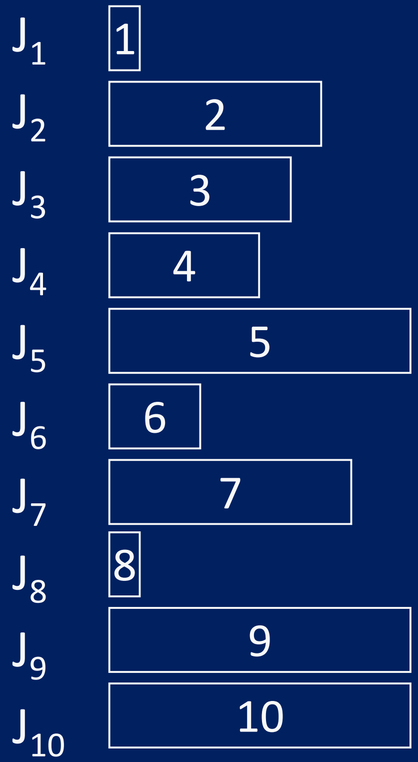



SPT: { $\underbrace{1, 8, 6}, \underbrace{4, 3, 2}, \underbrace{7, 5, 9}, 10$ }

$S_{M1}$={ 1, 4, 7, 10 }

$S_{M2}$={ 8, 3, 5 }

$S_{M3}$={ 6, 2, 9 }

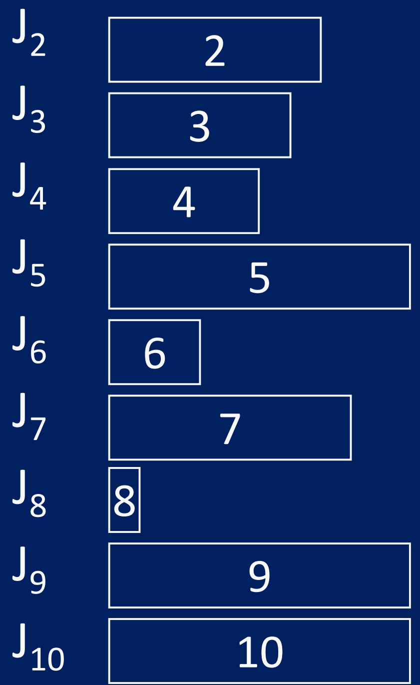



SPT: { 1, 8, 6, 4, 3, 2, 7, 5, 9, 10 }

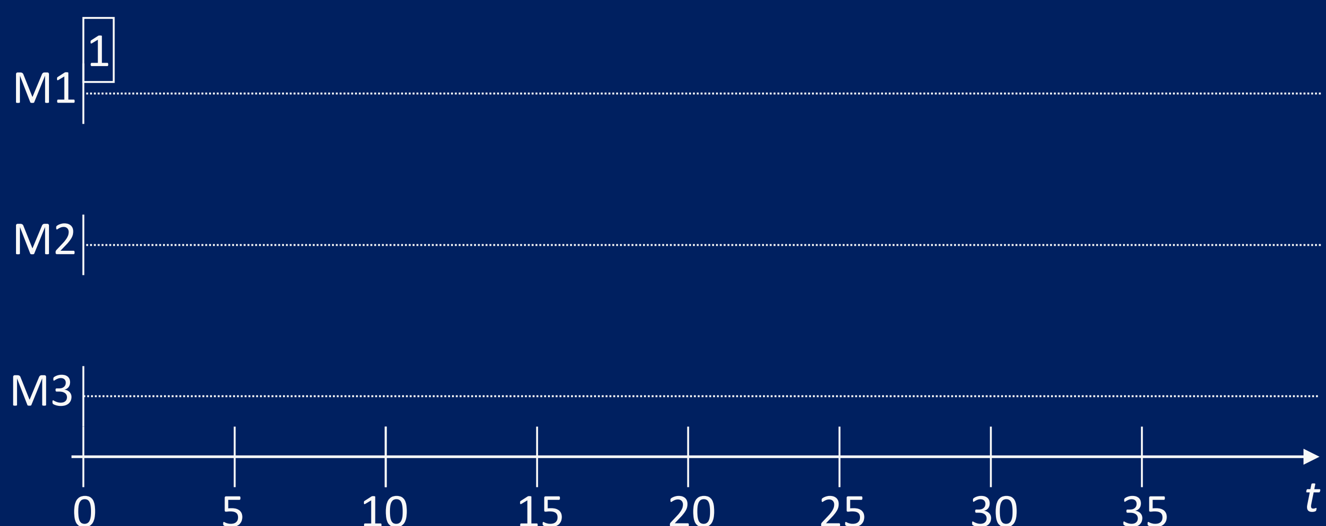
$S_{M1}$={ 1, 4, 7, 10 }

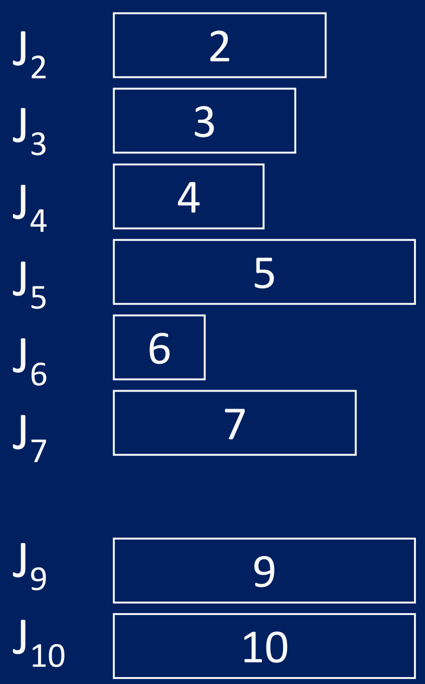
$S_{M2}$={ 8, 3, 5 }

$S_{M3}$={ 6, 2, 9 }

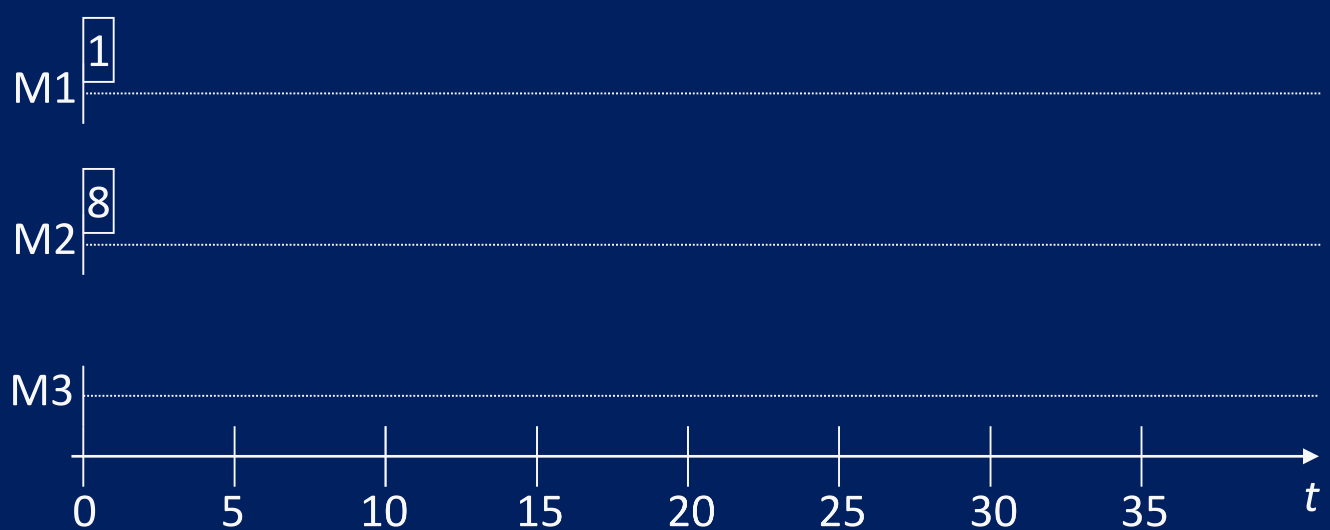


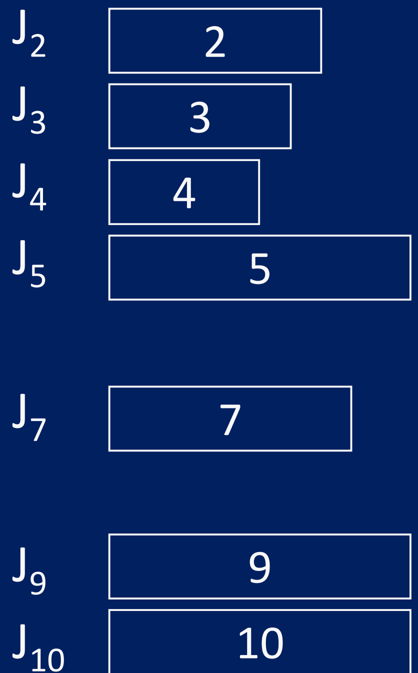

$$S_{M1}=\{\ 1,\ 4,\ 7,\ 10\ \}$$
$$S_{M2}=\{\ 8,\ 3,\ 5\ \}$$
$$S_{M3}=\{\ 6,\ 2,\ 9\ \}$$



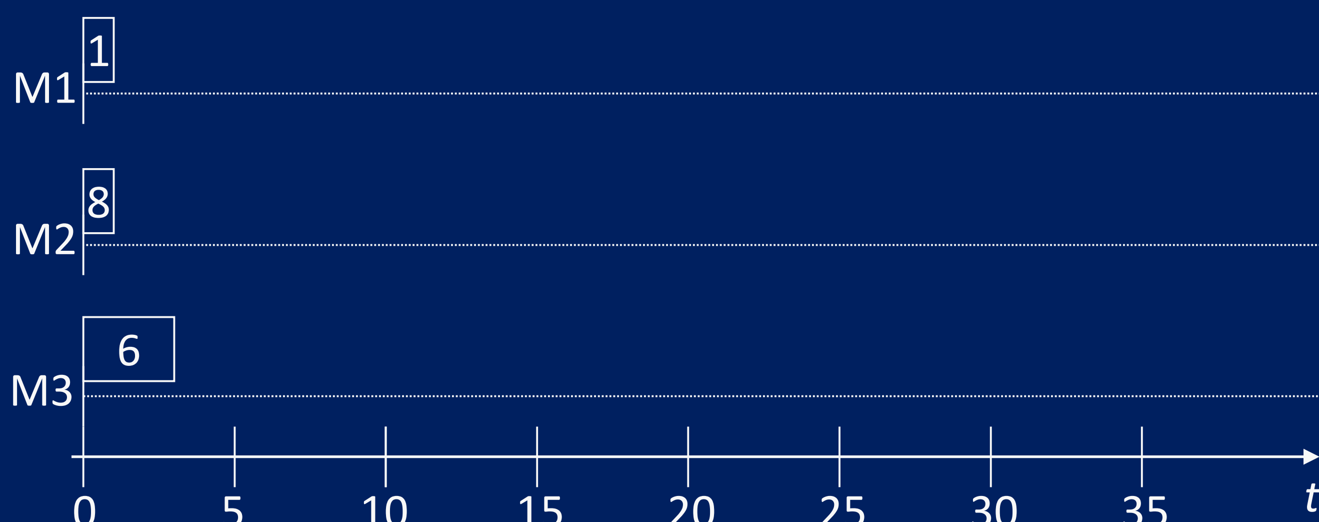


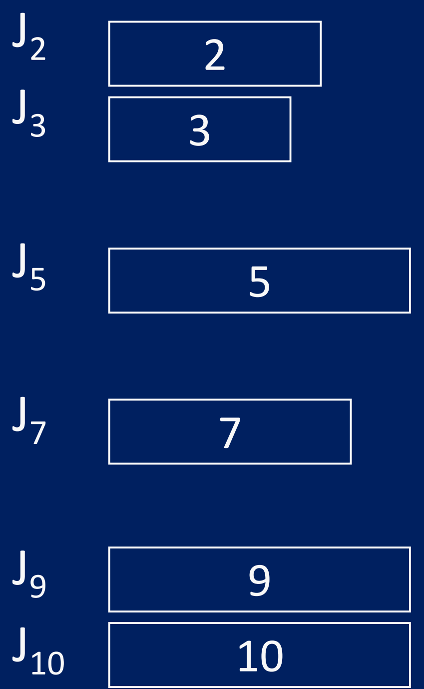

$$S_{M1}=\{\ 1,\ 4,\ 7,\ 10\ \}$$
$$S_{M2}=\{\ 8,\ 3,\ 5\ \}$$
$$S_{M3}=\{\ 6,\ 2,\ 9\ \}$$



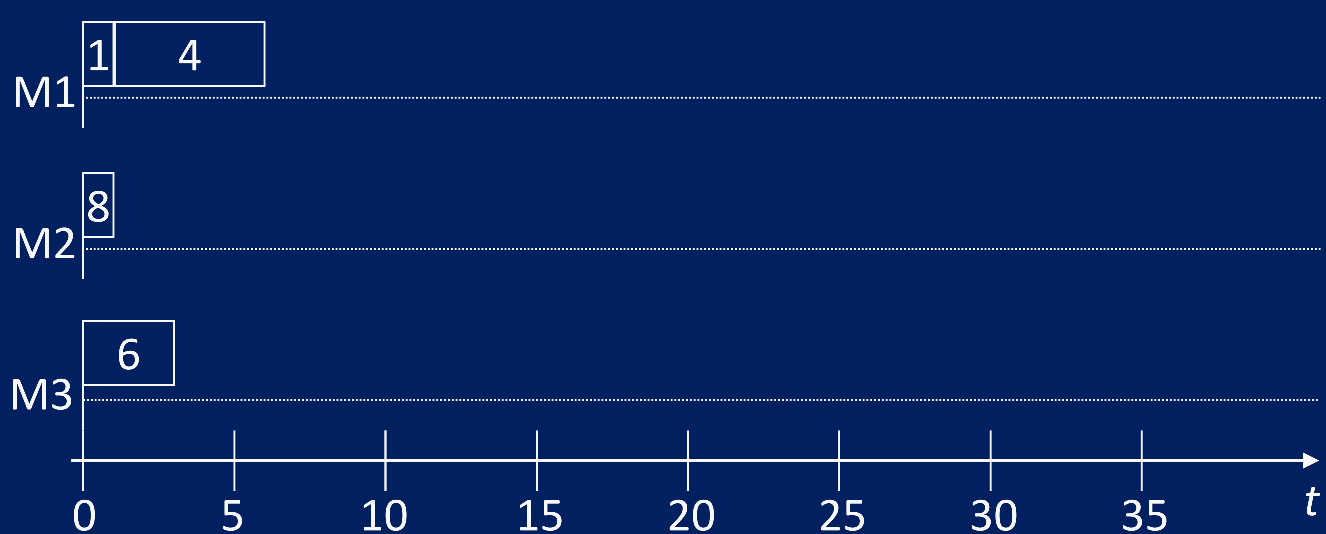


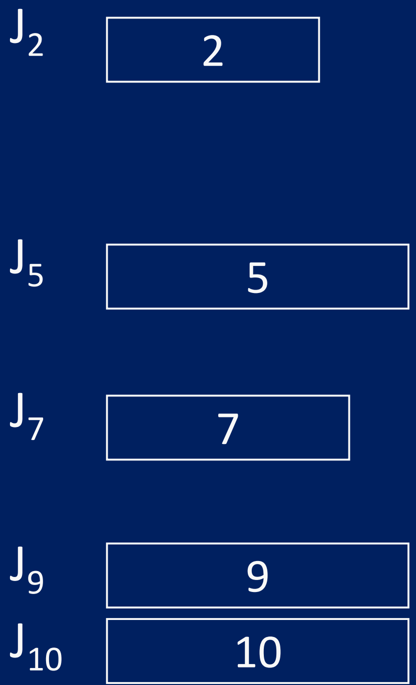

$$S_{M1}=\{ 1, 4, 7, 10 \}$$
$$S_{M2}=\{ 8, 3, 5 \}$$
$$S_{M3}=\{ 6, 2, 9 \}$$



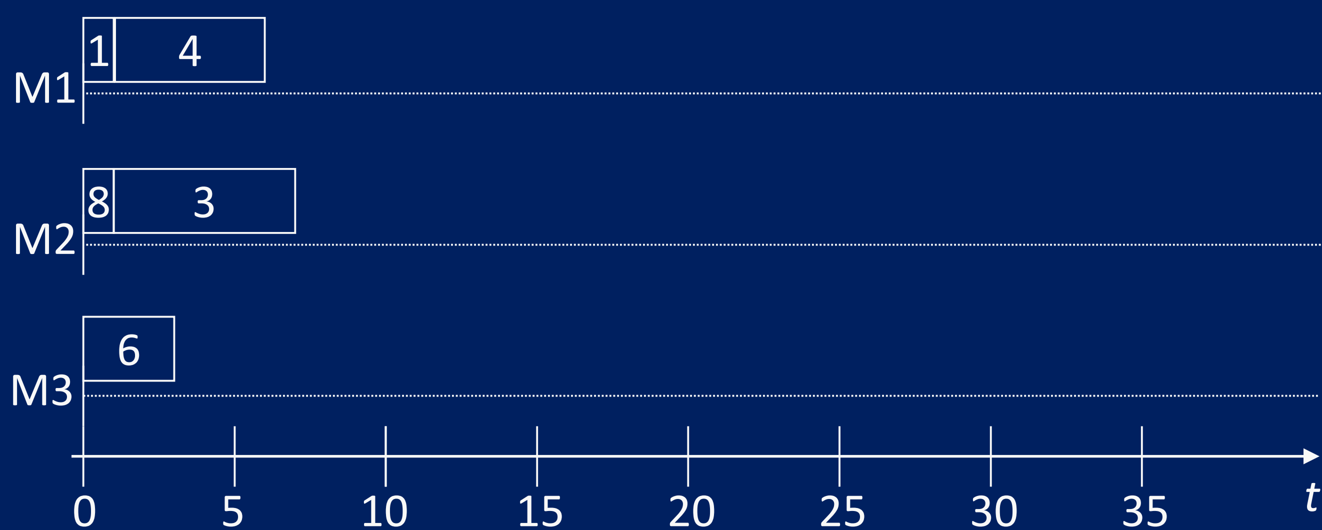


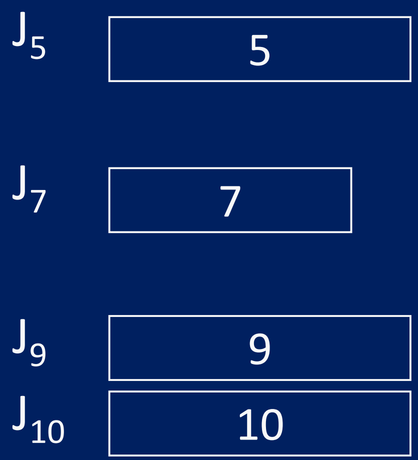

$$S_{M1}=\{\ 1,\ 4,\ 7,\ 10\ \}$$
$$S_{M2}=\{\ 8,\ 3,\ 5\ \}$$
$$S_{M3}=\{\ 6,\ 2,\ 9\ \}$$



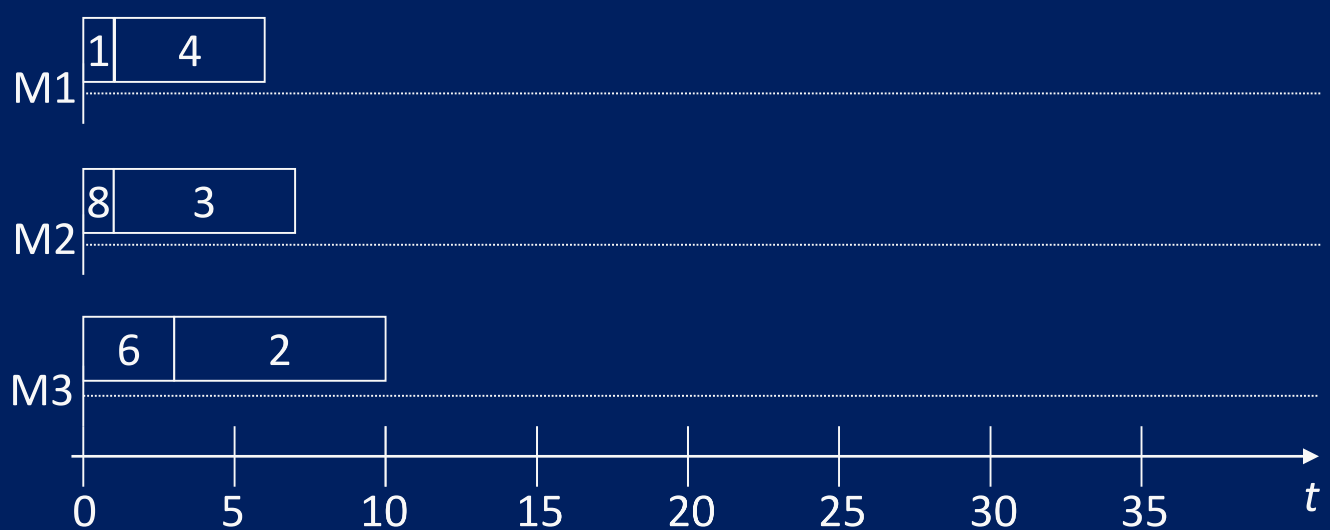


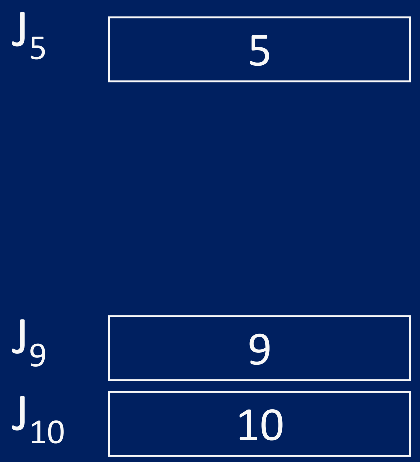

$$S_{M1}=\{\ 1,\ 4,\ 7,\ 10\ \}$$
$$S_{M2}=\{\ 8,\ 3,\ 5\ \}$$
$$S_{M3}=\{\ 6,\ 2,\ 9\ \}$$



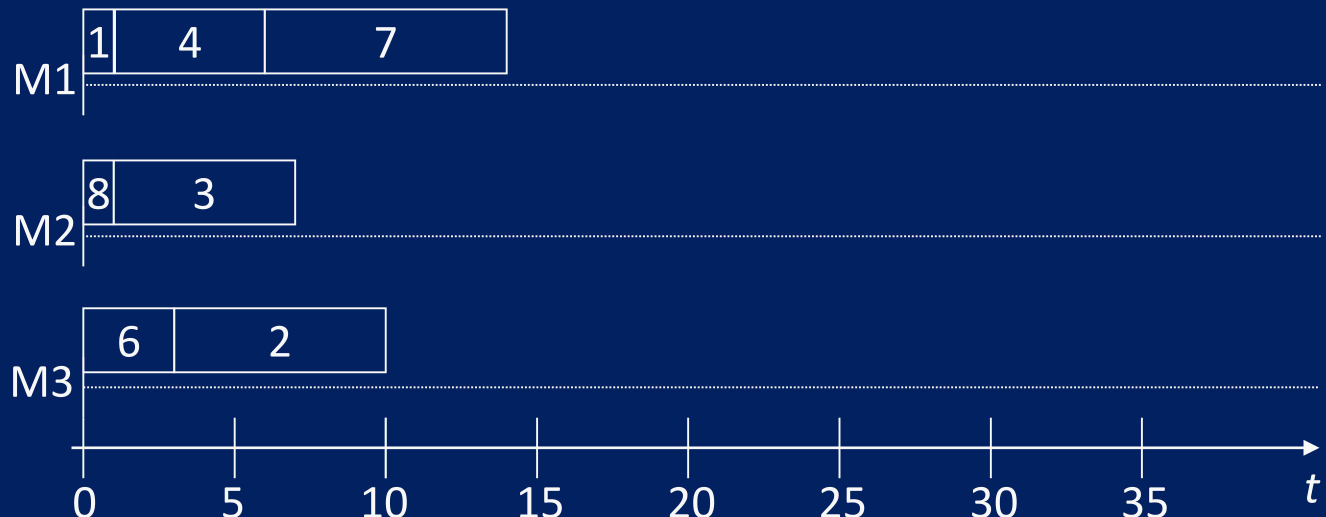



$$S_{M1}=\{\ 1,\ 4,\ 7,\ 10\ \}$$
$$S_{M2}=\{\ 8,\ 3,\ 5\ \}$$
$$S_{M3}=\{\ 6,\ 2,\ 9\ \}$$





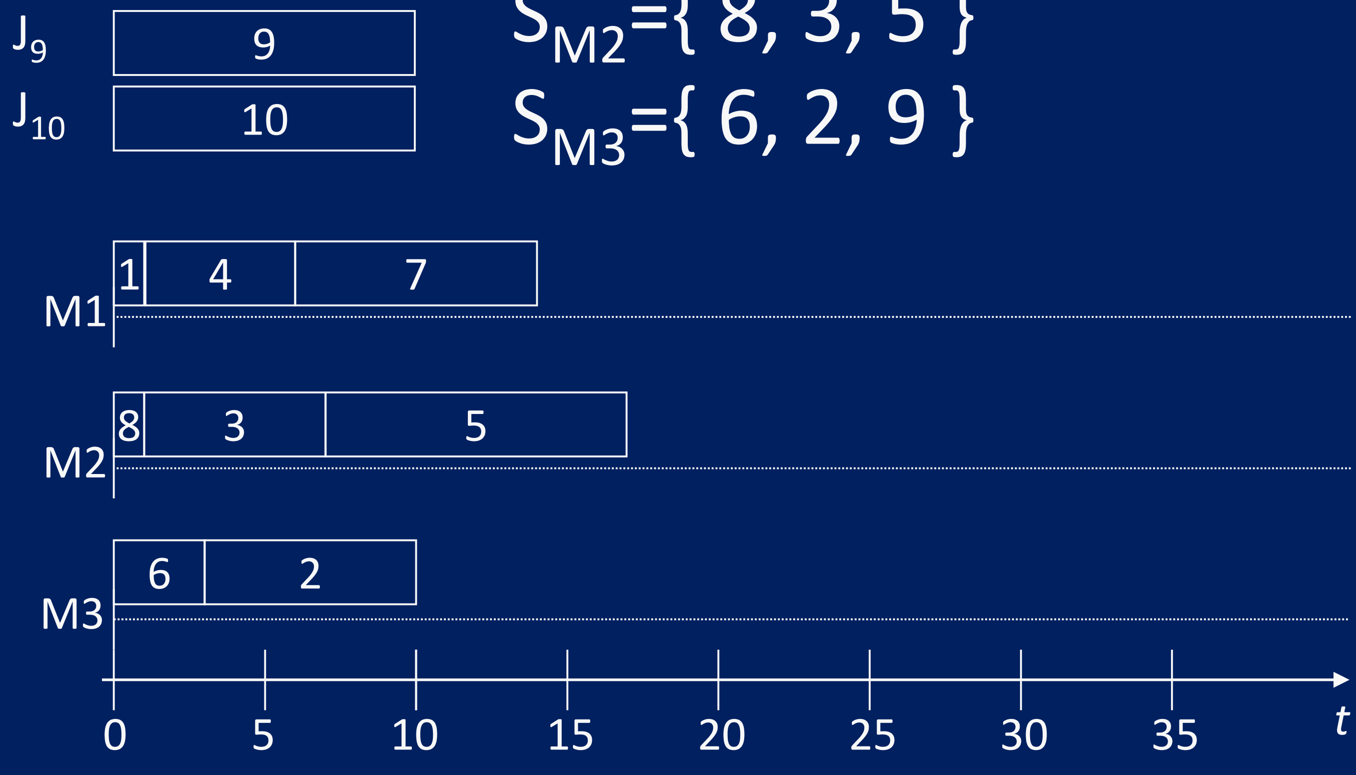

$$S_{M1}=\{\ 1,\ 4,\ 7,\ 10\ \}$$
$$S_{M2}=\{\ 8,\ 3,\ 5\ \}$$
$$S_{M3}=\{\ 6,\ 2,\ 9\ \}$$



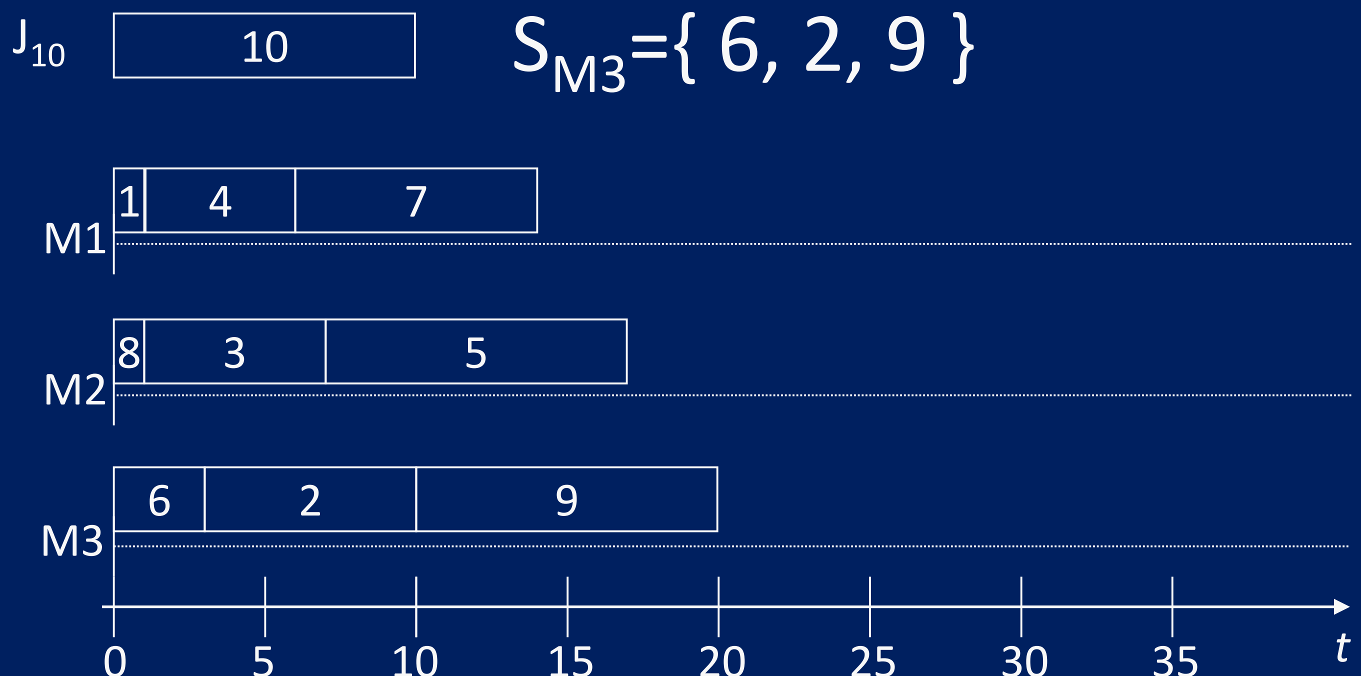
$S_{M1}=\{\ 1, 4, 7, 10\ \}$

$S_{M2}=\{\ 8, 3, 5\ \}$

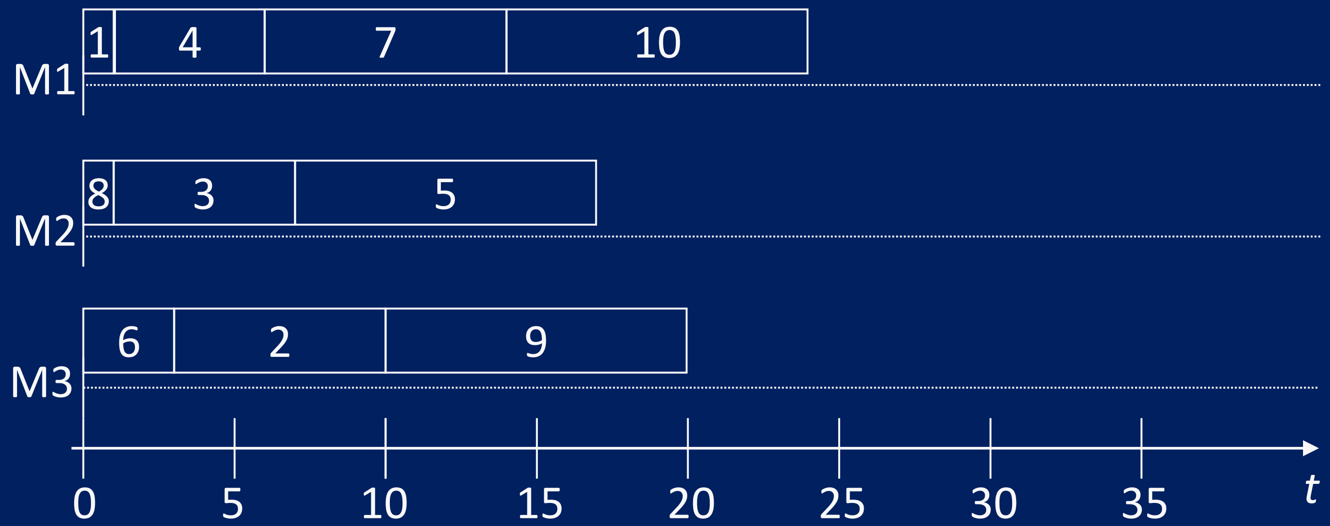
$S_{M3}=\{\ 6, 2, 9\ \}$



$S_{M1}=\{\ 1,\ 4,\ 7,\ 10\ \}$
$S_{M2}=\{\ 8,\ 3,\ 5\ \}$
$S_{M3}=\{\ 6,\ 2,\ 9\ \}$



$$S_{M1}=\{\ 1,\ 4,\ 7,\ 10\ \}$$
$$S_{M2}=\{\ 8,\ 3,\ 5\ \}$$
$$S_{M3}=\{\ 6,\ 2,\ 9\ \}$$



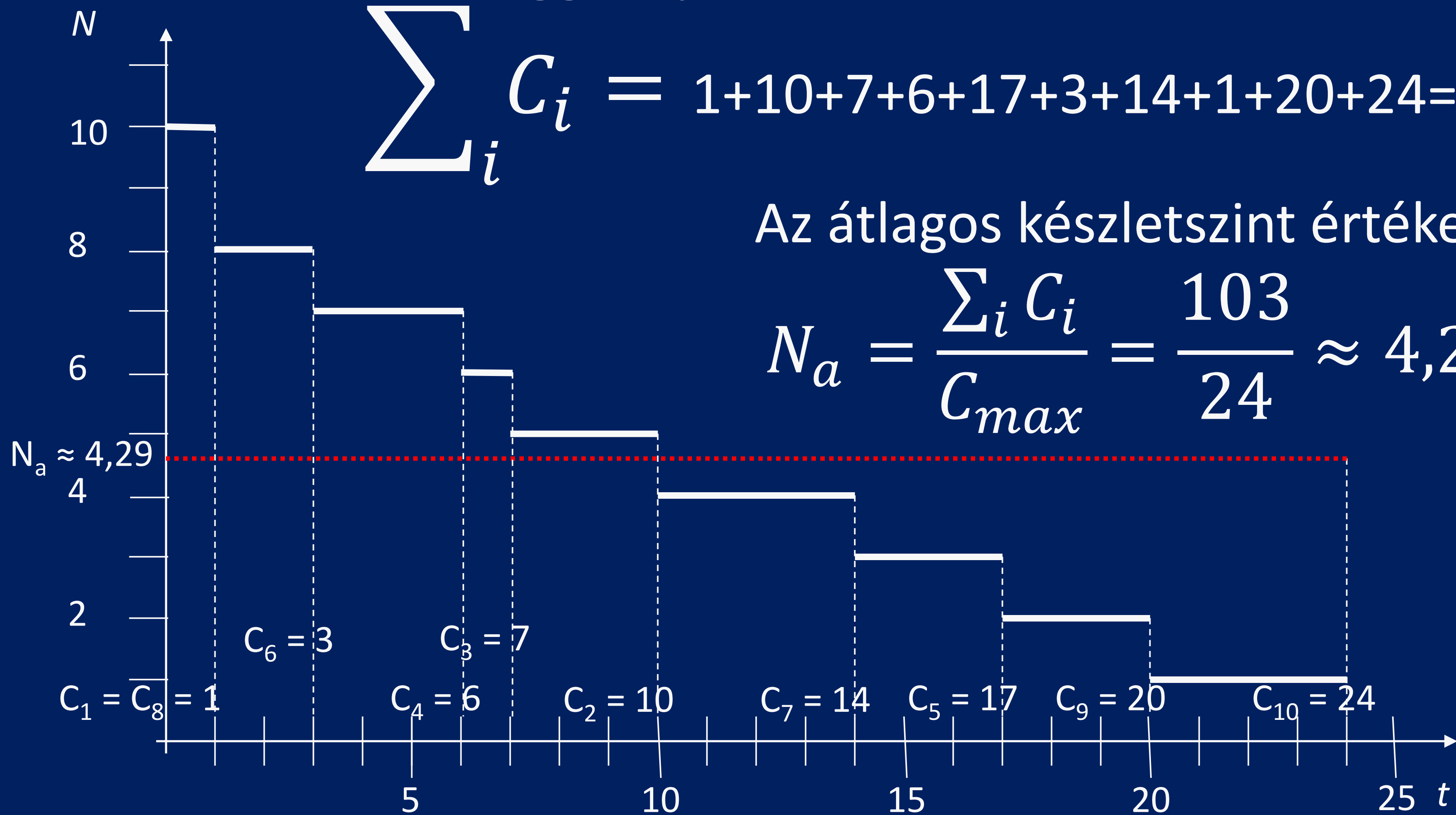
# Készlet-diagram

A célfüggvény értéke:

$$\sum_i C_i = 1+10+7+6+17+3+14+1+20+24=103$$

Az átlagos készletszint értéke:

$$N_a = \frac{\sum_i C_i}{C_{max}} = \frac{103}{24} \approx 4{,}29$$



# Párhuzamosan működő erőforrások ütemezése LPT+LIST algoritmus alkalmazásával

# Az ütemezés célja

- A összes munka (halmaz) teljesítésének időpontja legyen minimális
(az utolsóként elkészülő munka befejezési időpontja a lehető legkorábbi legyen).

$$C_{max} = \max\{C_1, \dots , C_N\}$$

$$C_{max} \rightarrow \min$$

Az ütemezési modell formális leírása:

$$P \mid\mid C_{max}$$

# Az LPT+LIST algoritmus

# Az LPT+LIST algoritmus

LPT+LIST algoritmus:

- 1. fázis: Előkészítés LPT algoritmussal.
  Rendezzük a munkákat a műveleti idő szerint nemnövekvő sorrendbe (Longest Processing Time, LPT).
  Ennek az a célja, hogy először a nagyobb terhelést jelentő munkákat helyezzük el az ütemtervben, majd aztán a kisebbeket. Ennek az előnye az, hogy a menet közben viszonylag jól kiegyenlített terhelések végére ne kellejen egy nagy "kilógó" munkát beilleszteni.
- 2. fázis: Ütemezés LIST algoritmussal.
  Az LPT szerint rendezett munkákat egyesével tesszük be az ütemtervbe úgy, hogy mindig a lista elejéről a még be nem ütemezettek közül a legnagyobb műveleti idejű munkáról hozunk döntést. A soron következő munkát ahhoz a géphez rendeljük hozzá, amelyik a legkorábban felszabadul a terhelései alól. A beütemezendő munkát mindig a kiválasztott gép ütemezési vektorának a végére helyezzük el.

# Heurisztikus megoldás

- A $P||C_{max}$ ütemezési feladat általános esetben nem oldható meg egzakt módon polinomiális futási idejű algoritmussal. Ezért heurisztikus megoldási módszert használunk, amely gyorsan előállít egy jó közelítő megoldást, de az optimális megoldást nem tudja garantálni.

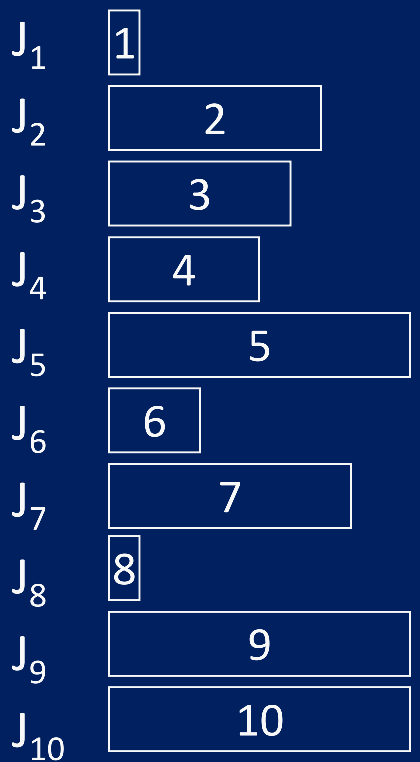
Az LPT+LIST heurisztikus módszer alapja:

- A munkák minél egyenletesebb szétosztása.

# Megjegyzések

1. Az LPT+LIST algoritmus ***felépítő jellegű***:
   az ütemterv kidolgozása közben hozott döntést a későbbi döntések nem változtatják meg.
   – Ha egy munkát hozzárendeltünk egy géphez, és beállítottuk a végrehajtási sorban a pozícióját, akkor a későbbi dontések ezt már nem változtatják meg, mert mindig a lista végére illesztjük a soron következő munkát.
2. A feladat megoldására ***kereső algoritmusok***at is alkalmazhatunk. (Lásd a később!)

# Illusztratív példa az LPT+LIST algoritmus működésére

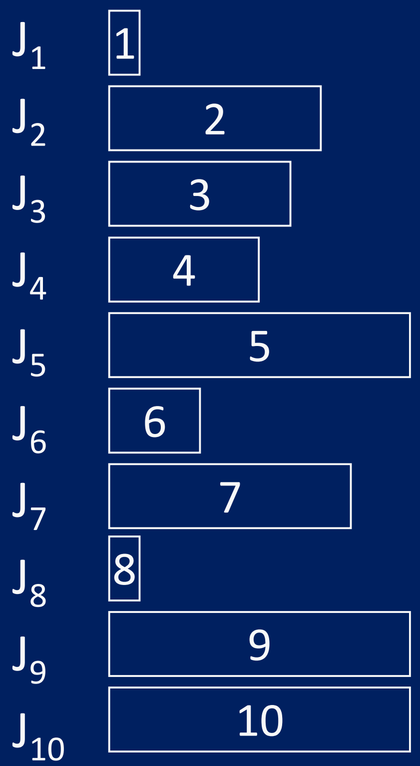



| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 1 | 7 | 6 | 5 | 10 | 3 | 8 | 1 | 10 | 10 |

$S_{M1}=\{\ \ \}$

$S_{M2}=\{\ \ \}$

$S_{M3}=\{\ \ \}$



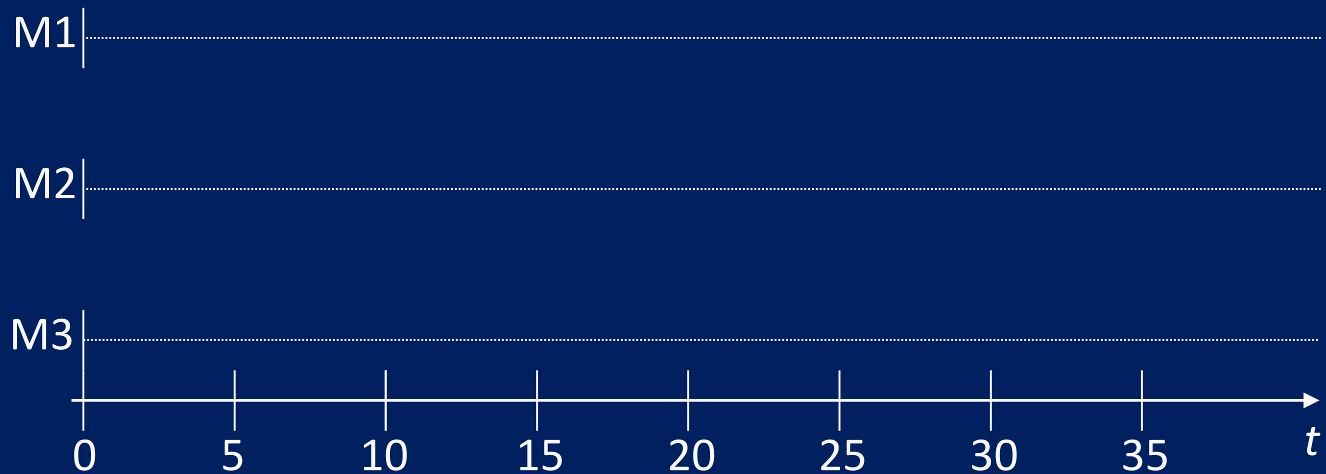

$LPT\{\,5, 9, 10, 7, 2, 3, 4, 6, 1, 8\,\}$

$\min\{\,C_{M1}, C_{M2}, C_{M3}\,\} \Rightarrow M^*$

$S_{M1}=\{\ \}$

$S_{M2}=\{\ \}$

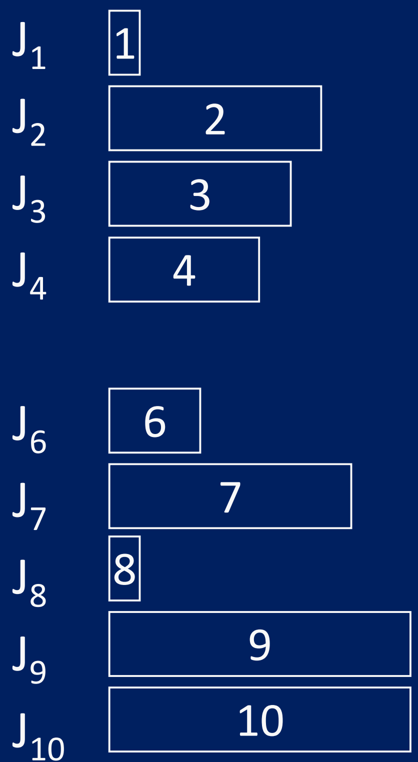
$S_{M3}=\{\ \}$




$C_{M1}= 0$

$C_{M2}= 0$

$C_{M3}= 0$



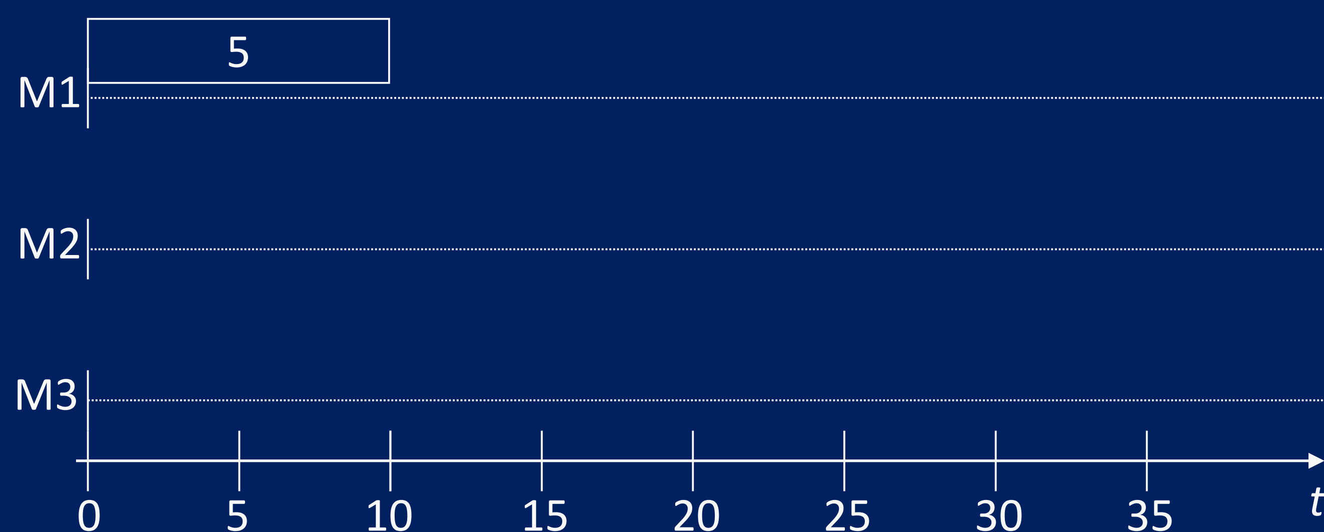

LPT{ 9, 10, 7, 2, 3, 4, 6, 1, 8 }

$\min\{ C_{M1}, C_{M2}, C_{M3} \}$ → M*

$S_{M1}$={ 5 }

$S_{M2}$={ }

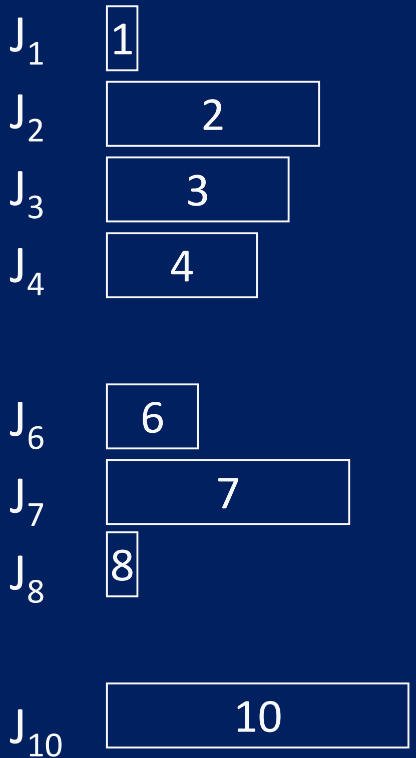
$S_{M3}$={ }




$C_{M1}$= 10

$C_{M2}$= 0

$C_{M3}$= 0



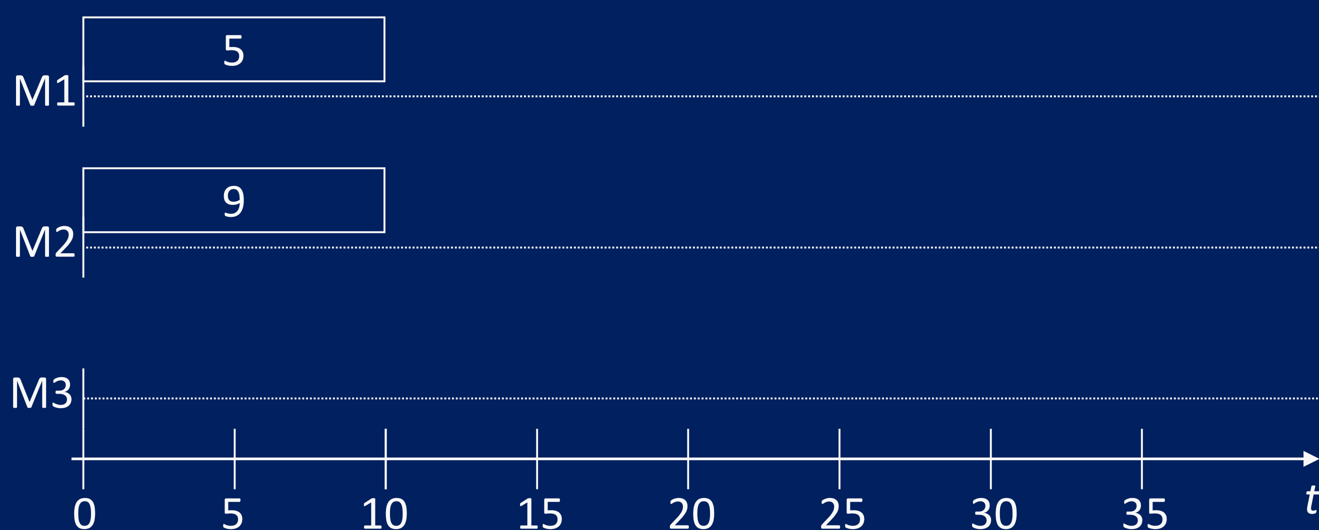

$$\mathrm{LPT}\{\ 10, 7, 2, 3, 4, 6, 1, 8\ \}$$

$$\min\{\ C_{M1}, C_{M2}, C_{M3}\ \} \Rightarrow M^*$$

$$S_{M1} = \{\ 5\ \}$$

$$S_{M2} = \{\ 9\ \}$$

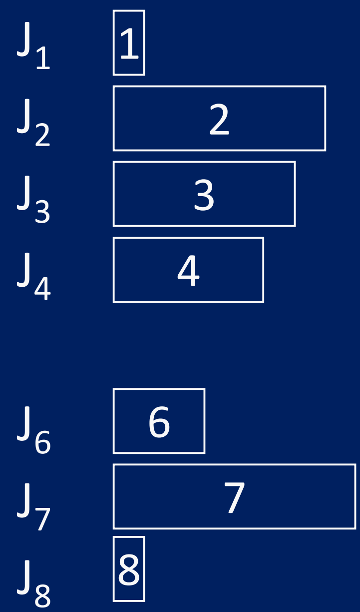
$$S_{M3} = \{\ \ \}$$




$$C_{M1} = 10$$

$$C_{M2} = 10$$

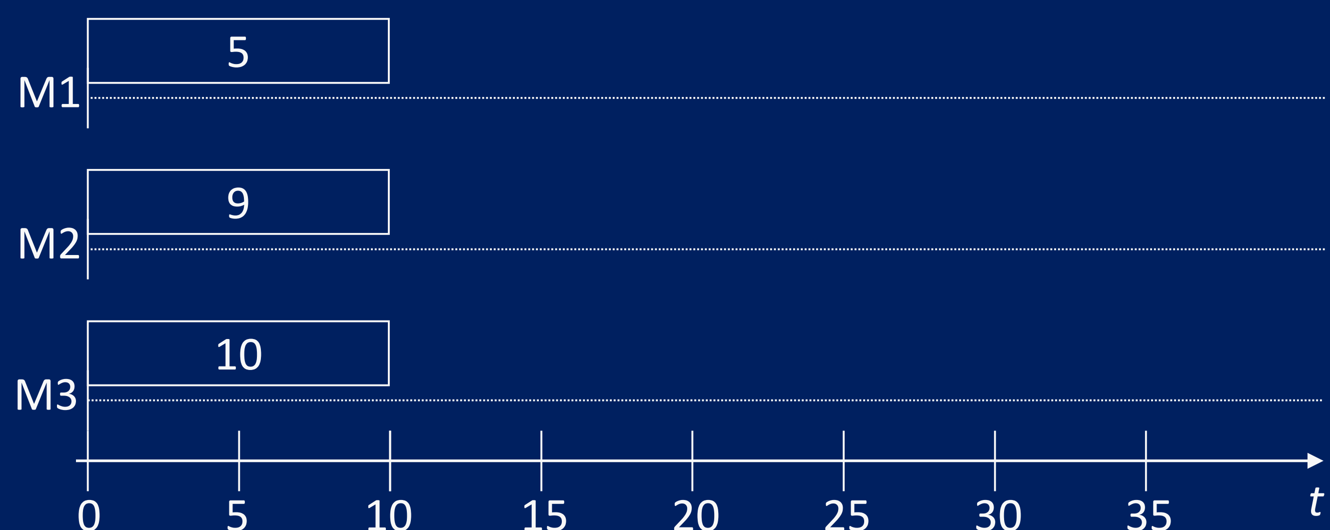
$$C_{M3} = 0$$



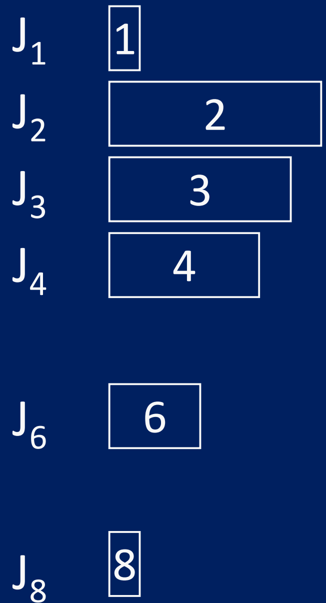

$$\mathrm{LPT}\{\ 7, 2, 3, 4, 6, 1, 8\ \}$$

$$\min\{\ C_{M1}, C_{M2}, C_{M3}\ \} \Rightarrow M^*$$

$$S_{M1}=\{\ 5\ \}$$
$$S_{M2}=\{\ 9\ \}$$
$$S_{M3}=\{\ 10\ \}$$



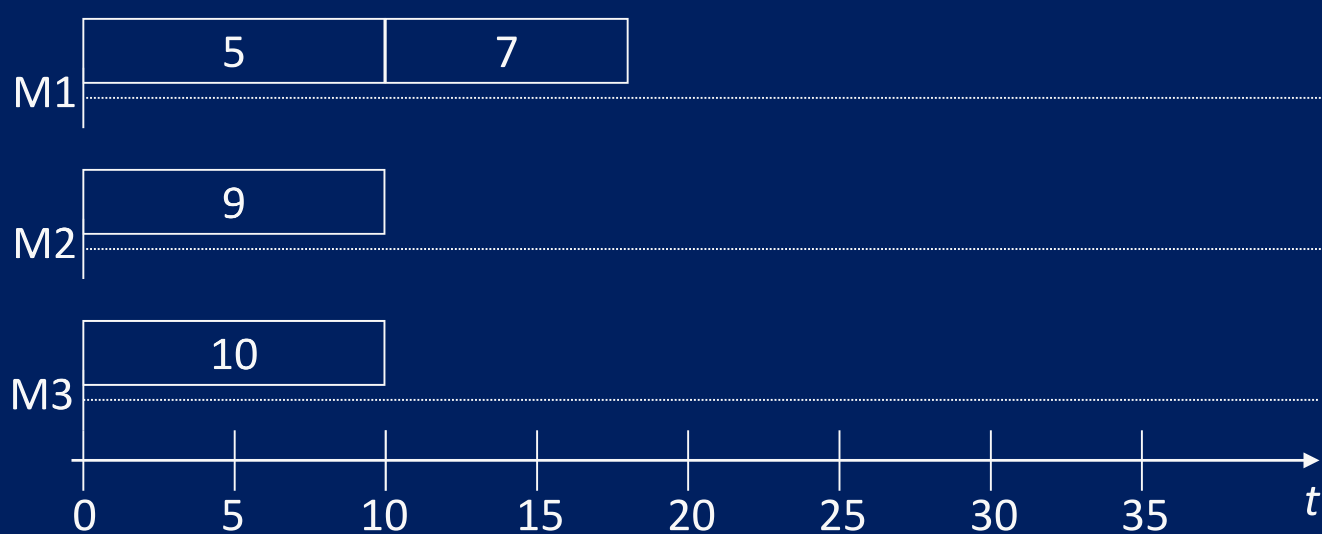

$$C_{M1}= 10$$
$$C_{M2}= 10$$
$$C_{M3}= 10$$



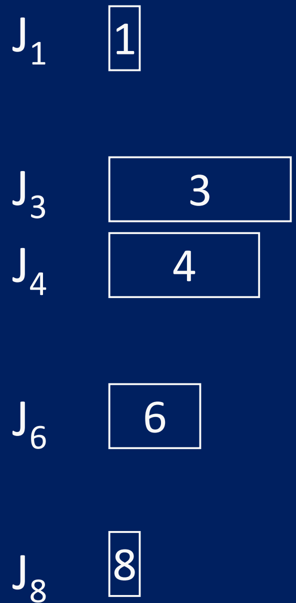

$$LPT\{\ 2, 3, 4, 6, 1, 8\ \}$$

$$\min\{\ C_{M1}, C_{M2}, C_{M3}\ \} \Rightarrow M^*$$

$$S_{M1} = \{\ 5, 7\ \}$$
$$S_{M2} = \{\ 9\ \}$$
$$S_{M3} = \{\ 10\ \}$$



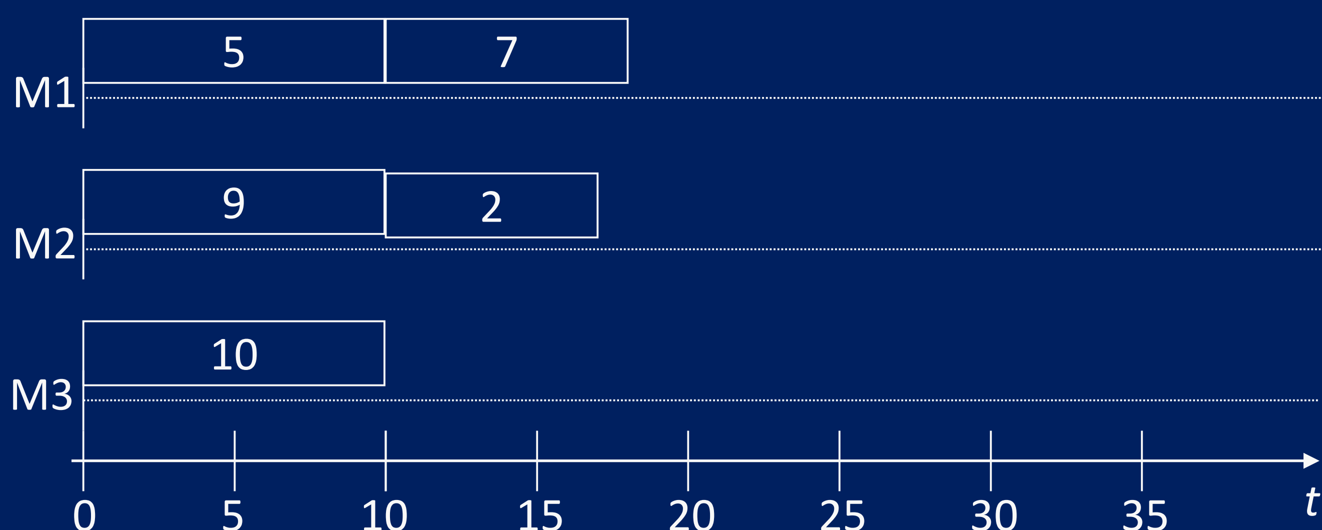

$$C_{M1} = 18$$
$$C_{M2} = 10$$
$$C_{M3} = 10$$




$$LPT\{\ 3,\ 4,\ 6,\ 1,\ 8\ \}$$

$$\min\{\ C_{M1},\ C_{M2},\ C_{M3}\ \} \Rightarrow M^*$$

$$S_{M1}=\{\ 5,\ 7\ \}$$
$$S_{M2}=\{\ 9,\ 2\ \}$$
$$S_{M3}=\{\ 10\ \}$$




$$C_{M1}= 18$$
$$C_{M2}= 17$$
$$C_{M3}= 10$$

$J_1$ 1

$$LPT\{\ 4,\ 6,\ 1,\ 8\ \}$$

$\Downarrow$

$J_4$ 4

$$\min\{\ C_{M1},\ C_{M2},\ C_{M3}\ \} \Rightarrow M^*$$

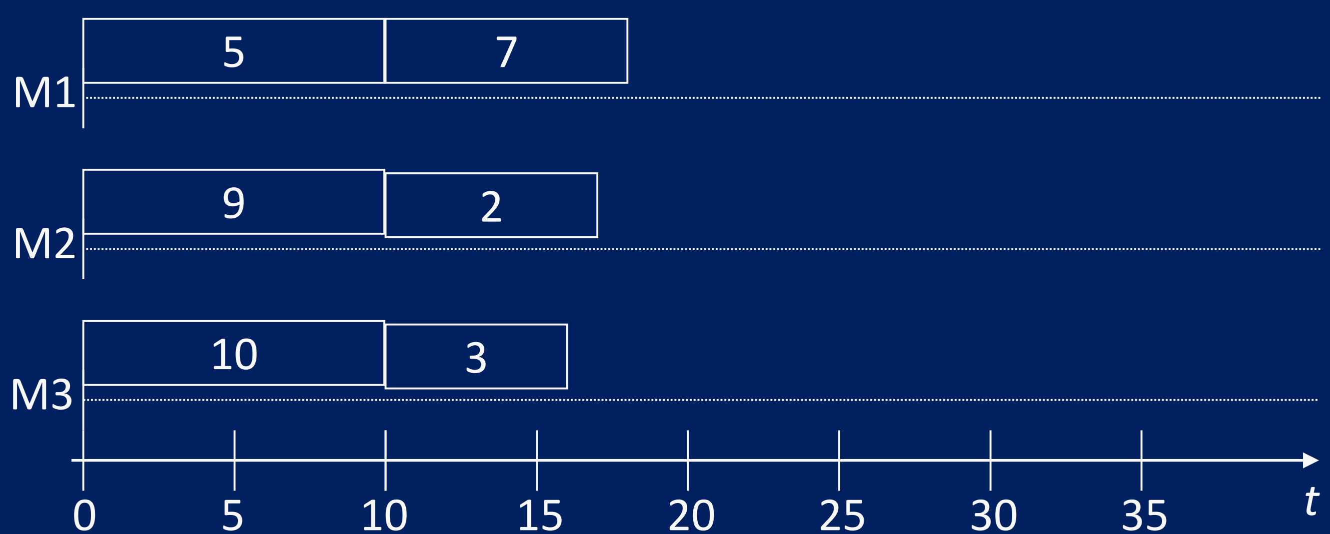
$J_6$ 6

$$S_{M1}=\{\ 5,\ 7\ \}$$

$$S_{M2}=\{\ 9,\ 2\ \}$$

$$S_{M3}=\{\ 10,\ 3\ \}$$

$J_8$ 8




$C_{M1}= 18$

$C_{M2}= 17$

$C_{M3}= 16$

$J_1$ $\boxed{1}$

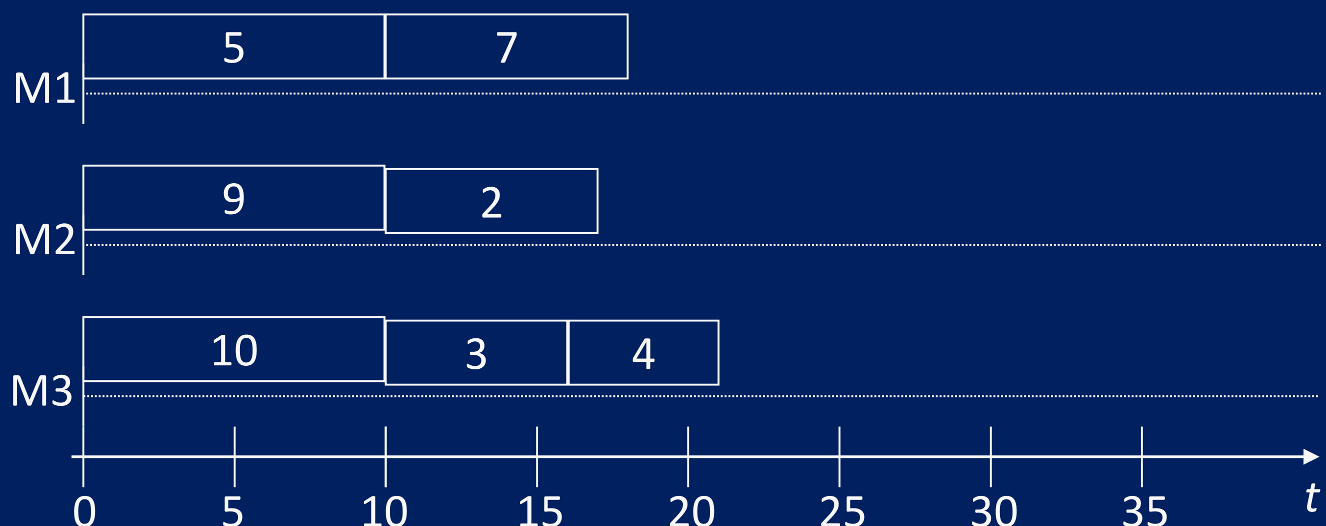
$$\mathrm{LPT}\{\ 6,\ 1,\ 8\ \}$$

$$\min\{\ C_{M1},\ C_{M2},\ C_{M3}\ \} \Rightarrow M^*$$

$J_6$ $\boxed{6}$

$J_8$ $\boxed{8}$

$$S_{M1}=\{\ 5,\ 7\ \}$$
$$S_{M2}=\{\ 9,\ 2\ \}$$
$$S_{M3}=\{\ 10,\ 3,\ 4\ \}$$




$C_{M1}= 18$

$C_{M2}= 17$

$C_{M3}= 21$

$J_1$ $\boxed{1}$

LPT{ 1, 8 }

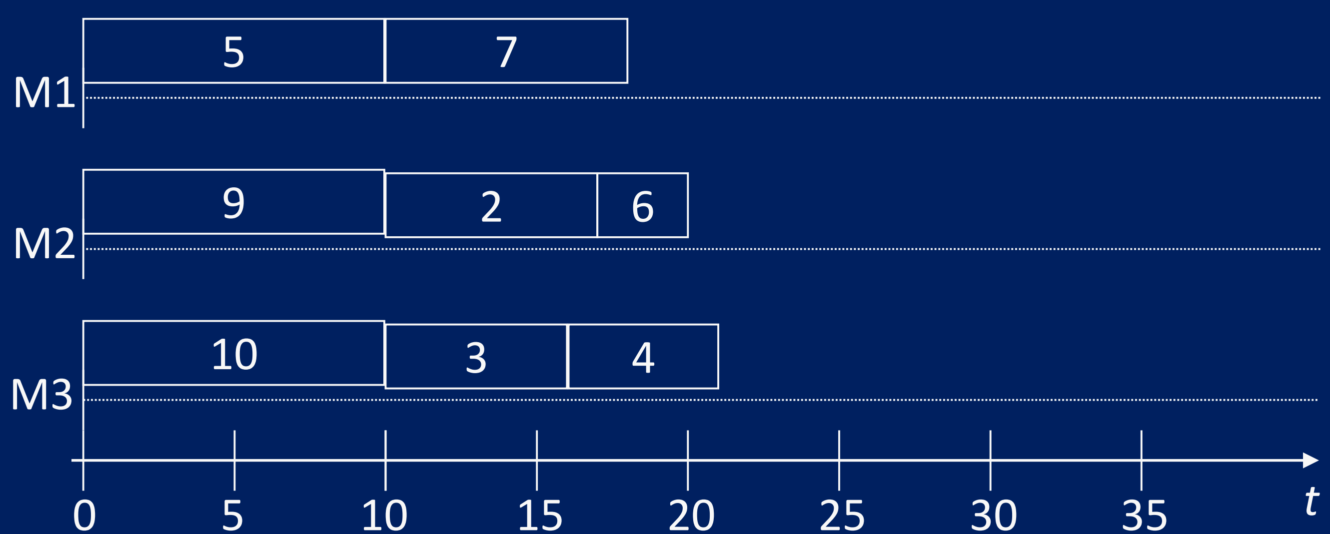
$\min\{ C_{M1}, C_{M2}, C_{M3} \}$ ⇒ $M^*$

$J_8$ $\boxed{8}$

$S_{M1}=\{ 5, 7 \}$

$S_{M2}=\{ 9, 2, 6 \}$

$S_{M3}=\{ 10, 3, 4 \}$




$C_{M1}= 18$

$C_{M2}= 20$

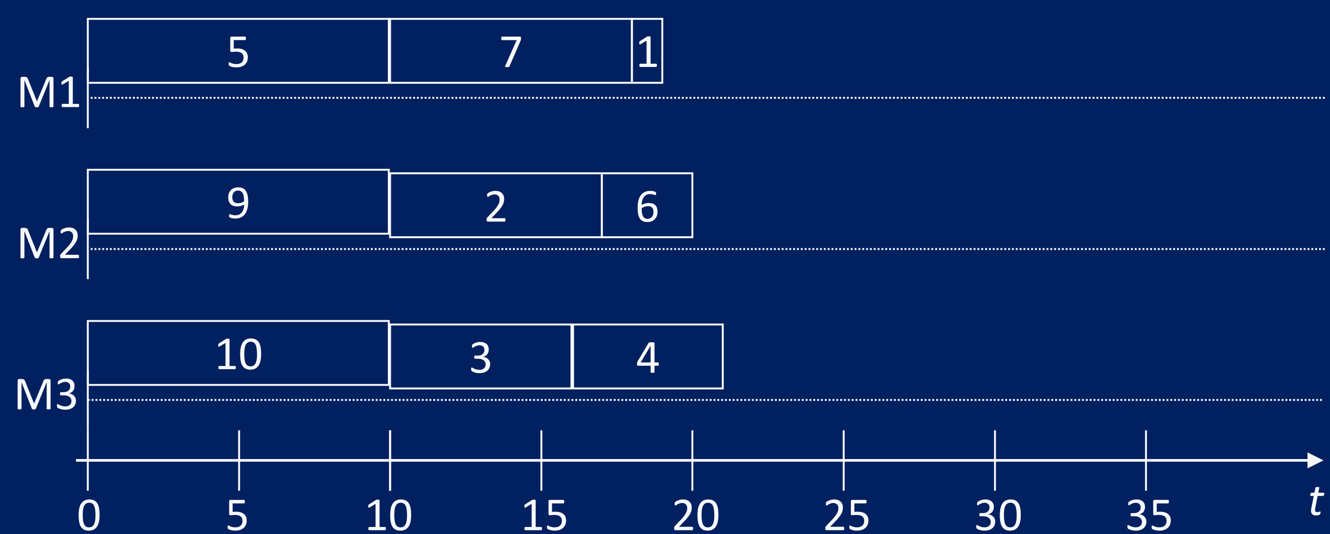
$C_{M3}= 21$

LPT{ 8 }

$$\min\{ C_{M1}, C_{M2}, C_{M3} \} \Rightarrow M^*$$

$$S_{M1}=\{ 5, 7, 1 \}$$
$$S_{M2}=\{ 9, 2, 6 \}$$
$$S_{M3}=\{ 10, 3, 4 \}$$

$J_8$ $\boxed{8}$




$C_{M1}= 19$

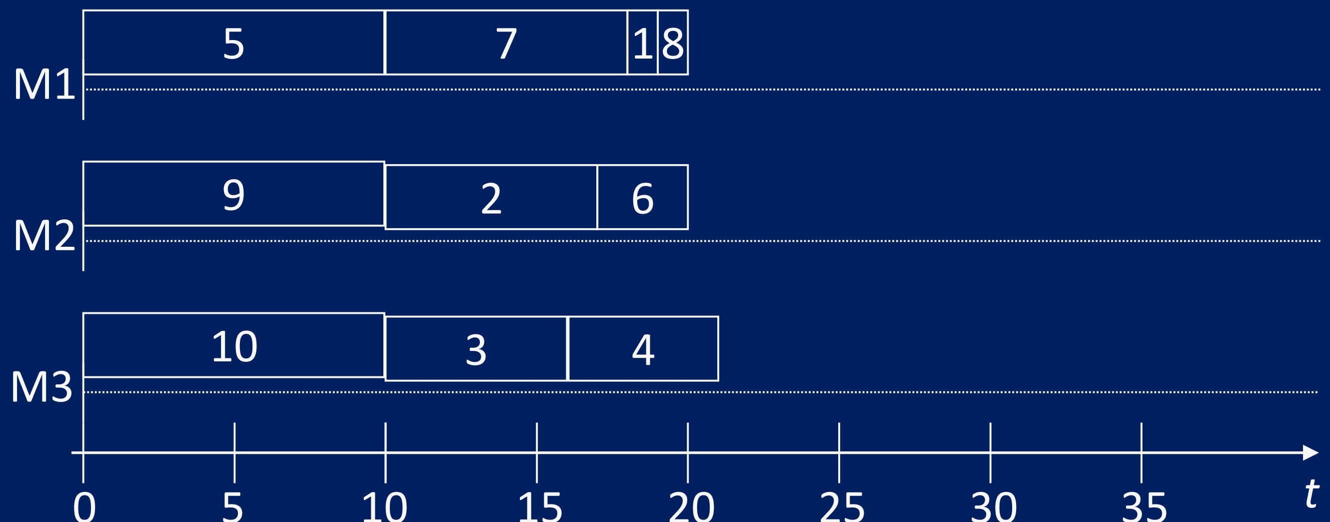
$C_{M2}= 20$

$C_{M3}= 21$

$$\mathrm{LPT}\{\ \}$$

$$\min\{ C_{M1}, C_{M2}, C_{M3} \} \Rightarrow M^*$$

$$S_{M1}=\{ 5, 7, 1, 8 \}$$
$$S_{M2}=\{ 9, 2, 6 \}$$
$$S_{M3}=\{ 10, 3, 4 \}$$




$$C_{M1}= 20$$
$$C_{M2}= 20$$
$$C_{M3}= 21$$

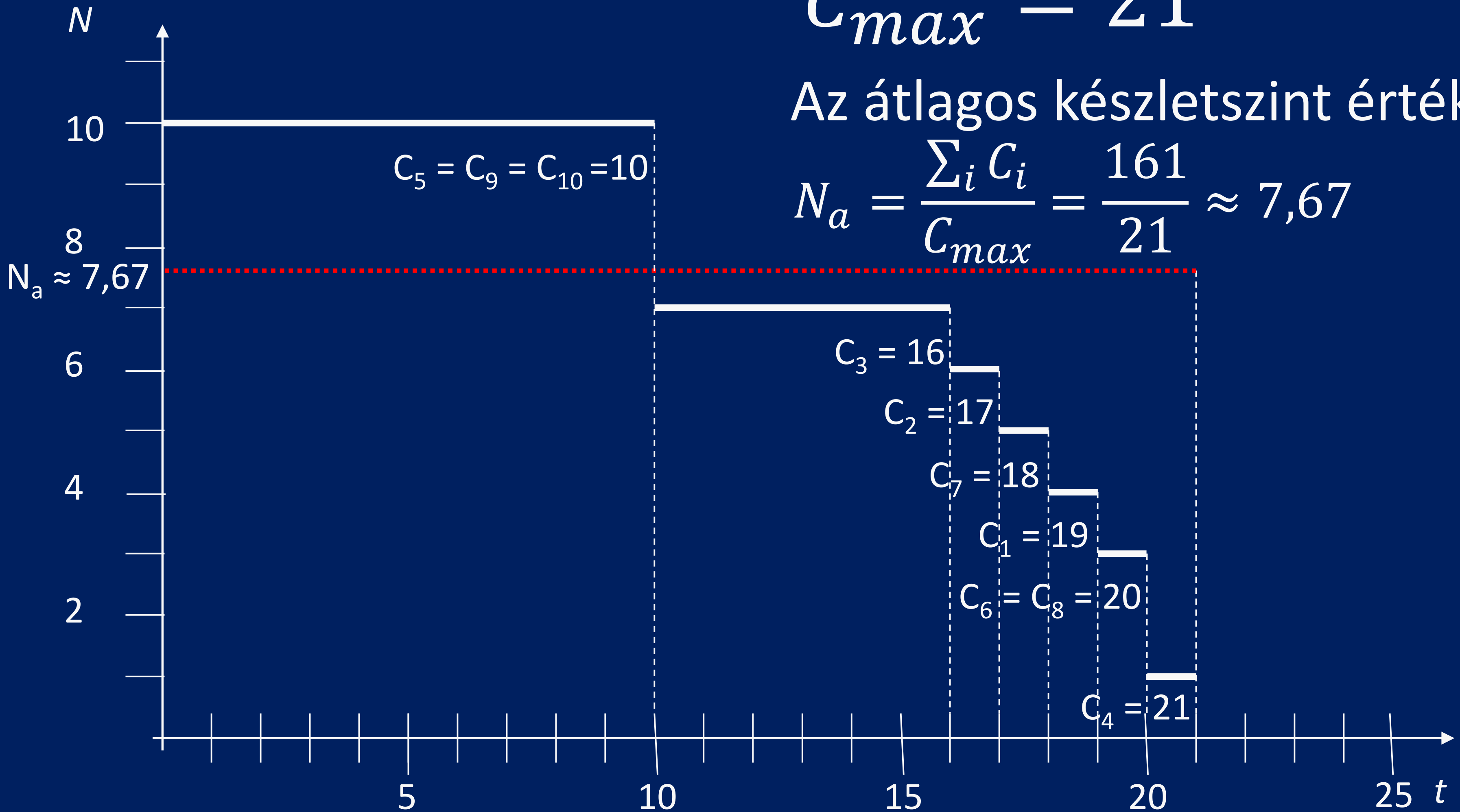
# Készlet-diagram

A célfüggvény értéke:

$$C_{max} = 21$$

Az átlagos készletszint értéke:

$$N_a = \frac{\sum_i C_i}{C_{max}} = \frac{161}{21} \approx 7{,}67$$



# Párhuzamosan működő erőforrások ütemezése EDD+LIST algoritmus alkalmazásával

# Az ütemezés célja

- A legnagyobb késés minimalizálása

  $L_i = C_i - d_i$

  $L_{max} = max\{L_1, \ldots, L_i , \ldots, L_N\} = max_i \{ L_i \}$

  $L_{max} \rightarrow min$

Az ütemezési modell formális leírása:

$$P \mid d_i \mid L_{max}$$

# Az EDD+LIST algoritmus

# Az EDD+LIST algoritmus

EDD+LIST algoritmus:

- 1. fázis: Előkészítés EDD algoritmussal.
  Rendezzük a munkákat a határidők szerint nemcsökkenő sorrendbe (Earliest Due Date, EDD).
  Ennek az a célja, hogy először a korábbi határidejű munkákat helyezzük el az ütemtervben, majd aztán a későbbi határidejűeket.
  2. fázis: Ütemezés LIST algoritmussal.
  Az EDD szerint rendezett munkákat egyesével tesszük be az ütemtervbe úgy, hogy mindig a lista elejéről a még be nem ütemezettek közül a legkorábbi határidejű munkáról hozunk döntést. A soron következő munkát ahhoz a géphez rendeljük hozzá, amelyik a legkorábban felszabadul a terhelései alól. A beütemezendő munkát mindig a kiválasztott gép ütemezési vektorának a végére helyezzük el.

# Heurisztikus megoldás

- A $P|d_i|L_{max}$ ütemezési feladat általános esetben nem oldható meg egzakt módon polinomiális futási idejű algoritmussal. Ezért heurisztikus megoldási módszert használunk, amely gyorsan előállít egy jó közelítő megoldást, de az optimális megoldást nem tudja garantálni.

Az EDD+LIST heurisztikus módszer alapja:

- Az $1|d_i|L_{max}$ és a $P||C_{max}$ feladat megoldási módszerének kombinálása.

# Megjegyzések

1. Az EDD+LIST algoritmus ***felépítő jellegű***: az LPT+LIST algoritmushoz hasonlóan.
2. A feladat megoldására ***kereső algoritmusok***at is alkalmazhatunk. (Lásd később!)

# Párhuzamosan működő erőforrások ütemezése tartalékidő-orientált algoritmus alkalmazásával

# Erőforrás-korlátos ütemezési modell

A feladat jellemzői:

- N számú egymástól független munka: $J_i$ (i=1, 2, …, n)
  - legkorábbi indítási időpont ($r_i$)
  - legkésőbbi befejezési időpont ($d_i$)
- szerelő szakmunkások (erőforrások)
  - erőforrás-rendelkezésre állást definiáló lista:
    kezdési időpontok szerint növekvő sorrendbe rendezett, átlapolódás nélküli fix hosszúságú időintervallum-szakaszok
  - az időintervallumokban külön-külön előírt számú munka végezhető el
- ***cél***: a határidő túllépés maximális értéke a lehető legkisebb legyen ($L_{max} \rightarrow$ min)

# Probléma-transzformáció

Továbbfejlesztett párhuzamos gépes ütemezési feladat

1. Az erőforrások rendelkezésre-állási időintervallumait folyamatosan, decimális egészekkel sorszámozzuk → lépések ($s$).
2. A műveleti idők egységnyi értéket (lépést) vesznek fel ($p_i = 1$).
3. A munkák időadatait átalakítjuk lépésekre:
   - $r_i$ → a befogadó műszakot követő műszak lépésszáma
   - $d_i$ → a befogadó műszak sorszáma
   - A teljesítés időpontját ($C_i$) a befogadó műszak sorszámával kifejezett alakban keressük.

# Probléma-transzformáció (folyt.)

4. ($L_i = C_i - d_i$) → a munka késését is egységnyi lépésben mérjük.
5. Szerelő szakmunkások → párhuzamosan működő virtuális erőforrások
   - egy erőforrás egyszerre csak egy munkán dolgozhat
   - egy munkán egyszerre csak egy erőforrás dolgozhat
   - a virtuális erőforrások száma lépésenként eltérő lehet: $P(s)$

A transzformált feladat:

$$P(s) \mid p_i=1;\ r_i=egész;\ d_i=egész \mid Lmax$$

*Időtartalék-orientált algoritmus a*
*$P(s)$ | $p_i=1$; $r_i$=integer; $d_i$=integer | $L_{max}$ feladathoz*

```
{  Rendezzük a J_i (i=1, 2, …, N_J) munkákat r_i szerint nem csökkenő sorrendbe;
   item ← 1;
   while (item ≤ N)
     {  s ← r_item;
        while (P(s) < 1)
          {   s ← s + 1;
              if (s > s_max) Kilépés megvalósítható megoldás nélkül;
          }
        R ← {J_i | J_i nem ütemezett és r_i ≤ s};
        mach ← 1;
        while (R nem üres)
          {  Válasszuk ki a legkisebb d_i határidővel rendelkező J_i munkát az R-ből;
             R ← R \ {J_i};
             Ütemezzük a J_i-t a mach.-edik gépre az s műszakban;
             C_i ← s;
             L_i ← C_i - d_i;
             T_i ← max(0, L_i);
             item ← item + 1;
```

//folytatás a következő lapon

```
//folytatás
 if (mach + 1 ≤ P(s)) mach ← mach + 1;
            else
              {   mach ← 1;
                  s ← s + 1;
                 while (P(s) < 1)
                   {  s ← s + 1;
                       if (s > s_max) Kilépés megvalósítható megoldás nélkül;
                   }
                 R ← R ∪ {J_i | J_i nem ütemezett és r_i ≤ s}
              }
          }
      }
   Visszatérés az elkészített optimális megoldással;
}
```

# Eredmények

- Az algoritmus a $C_i$ értékek kiszámításával egyben elő is állítja a keresett megoldást:
  - A $J_i$ munkát az $s = C_i$ sorszámú műszakban kell elvégezni.
- Minimális késés: $L_{max} \rightarrow min.$
- Polinomiális futási idő.

| Ji | Ri | Di |
|---|---|---|
| J1 | 0 | 1 |
| J2 | 1 | 1 |
| J3 | 2 | 3 |
| J4 | 0 | 3 |
| J5 | 1 | 1 |
| J6 | 1 | 2 |
| J7 | 0 | 3 |
| J8 | 0 | 3 |
| J9 | 0 | 3 |

# Illusztratív példa

| s | P(s) | Terhelés | Ütemezett munkák | | | |
|---|---|---|---|---|---|---|
| 0 | 4 | 4 | J1 | J4 | J7 | J8 |
| 1 | 2 | 2 | J2 | J5 | | |
| 2 | 3 | 3 | J6 | J9 | J3 | |

Rendezett lista: Ri

| Ji | Ri | Di |
|---|---|---|
| J1 | 0 | 1 |
| J4 | 0 | 3 |
| J7 | 0 | 3 |
| J8 | 0 | 3 |
| J9 | 0 | 3 |
| J2 | 1 | 1 |
| J5 | 1 | 1 |
| J6 | 1 | 2 |
| J3 | 2 | 3 |

| s=0 | | |
|---|---|---|
| J1 | 0 | 1 |
| J4 | 0 | 3 |
| J7 | 0 | 3 |
| J8 | 0 | 3 |
| J9 | 0 | 3 |

Beütemezzük: J1, J4, J7, J8

| s=1 | | |
|---|---|---|
| J9 | 0 | 3 |
| J2 | 1 | 1 |
| J5 | 1 | 1 |
| J6 | 1 | 2 |

Beütemezzük: J2, J5

| s=2 | | |
|---|---|---|
| J9 | 0 | 3 |
| J6 | 1 | 2 |
| J3 | 2 | 3 |

Beütemezzük: J6, J9, J3

# Eredmény

| Ji | Ri | Di | Ci | Li | |
|---|---|---|---|---|---|
| J1 | 0 | 1 | 0 | -1 | Siet |
| J2 | 1 | 1 | 1 | 0 | Időben |
| J3 | 2 | 3 | 2 | -1 | Siet |
| J4 | 0 | 3 | 0 | -3 | Siet |
| J5 | 1 | 1 | 1 | 0 | Időben |
| J6 | 1 | 2 | 2 | 0 | Időben |
| J7 | 0 | 3 | 0 | -3 | Siet |
| J8 | 0 | 3 | 0 | -3 | Siet |
| J9 | 0 | 3 | 2 | -1 | Siet |

# Köszönöm a figyelmet!

Az előadásvázlat elérhető az alábbi webcímen:

http://ait.iit.uni-miskolc.hu/~kulcsar/serv07.htm