Routing-scheduling flow shop problem

Michał Markowski

Politechnika Wrocławska Wydział Informatyki i Zarządzania

16 marca 2010
Contents

1 Scheduling problem
   - Problem formulation
   - Classes of scheduling problem

2 Flow shop problem with moving executors
   - Scheduling problem with moving executors

3 Flow shop problem with moving executors
   - Scheduling problem with moving executors
   - Example

4 Research
   - What has been done
   - Future research

5 Article
   - Publication
Scheduling problem

There is a set of jobs and executors. Executors must perform all of the jobs.
Problem formulation

Find a schedule $\gamma$, with respect to its constraints, which minimize a given criterion.
Criterion: the instant when last executor finish its job.
Classes of scheduling problem

Scheduling problems are divided into 2 classes, according to "usefulness" of executors.

- **Universal (parallel) executors**
- **Dedicated (specialized) executors**
  - *flow shop*
  - *open shop*
  - *job shop*
Executors are able to move!!!
Scheduling problem with moving executors

- \( H = \{1, 2, \ldots, H\} \), \( H \) - set of workstations and a number of workstations
- \( H + 1, \overline{H} = \{1, 2, \ldots, H, H + 1\} \) - depot and set of workstations including depot
- \( R = \{1, 2, \ldots, R\} \), \( R \) - set of executors and number of executors
- \( \hat{\tau}_{r,h} \) - the time for which activity \( h \) is performed by \( r \)
- \( \hat{\tau}_{r,g,h} \) - the time for which executor \( r \) drives up from workstation \( g \) to \( h \)
- \( \tau_{r,g,h} = \hat{\tau}_{r,h} + \hat{\tau}_{r,g,h} \) - task execution time including movement
- \( \gamma = [\gamma_{r,g,h}]_{g,h=1,2,\ldots,H+1}^{r=1,2,\ldots,R} \) - solution, where \( \gamma_{r,g,h} = 1(0) \) means if executor \( r \) performs task \( h \) after driving up from workstation \( g \).
Example of flow shop with moving executors - the routing scheduling flow shop

\[ H = 4, \text{depot} = 5, R = \{r1, r2, r3, r4\}. \]

Order of performing any task by executors: \{r1, r2, r3, r4\}
Decision result (task order): \{5, 3, 2, 4, 1, 5\}
What has been done

  Problem: routing scheduling flow shop
  Algorithm with $\frac{R+1}{2} Q^*$.

  Problem: routing scheduling open shop
  Algorithm with $\frac{R+1}{2} Q^*$.

  Problem: routing scheduling open shop
  Algorithm with $\frac{3}{2} Q^*$ for $R = 2$. 
Future research

- Branch and bound optimal algorithm.
- Better then Averbakh algorithm??.
- Implement and check Averbakh algorithm.
- Algorithms for solving routing FS with other criterion or constraints.
- Non-deterministic version
Definitions

- $D_i$ - deadline of job $i$
- $c_i$ - holding cost of job $i$
- $t_i$ - tardiness cost of job $i$
- $\sigma$ - partial schedule
- $Q(\sigma, j)$ - completion time of schedule $\sigma$ (partial schedule)
- $\sigma \oplus i$ - job $i$ appended to partial schedule
Criterion

\[ C_i = Q(\sigma \oplus i, m) \times c_i \text{ - weighted flowtime of job } i \]
\[ T_i = \max\{Q(\sigma \oplus i, m) - D_i; 0\} \text{ = weighted tardiness of job } i \]

Criterion of the problem:

\[ Z = \sum_{i=1}^{n} (C_i + T_i) \]
Heurystic

Heurystic formed of two construc algorithms. We have a null schedule, and we want to append whole set of jobs. We append jobs one by one using criterion for valuation of appended jobs.

2 criterion for appending jobs:

1. \[ \sum_{j=1}^{m} \left( Q(\sigma \oplus i, j) - (Q(\sigma, j) - c_i - t_i) \right) \]

2. \[ \frac{D_i}{c_i + t_i} \]
Jerzy Józefczyk and Michał Markowski.
Heuristic algorithms for solving uncertain routing-scheduling problem.

Jerzy Józefczyk and Michał Markowski.
Simulated annealing based robust algorithm for routing-scheduling problem with uncertain execution times.
INCOM, 2009.
(accepted).