Introduction to Job Shop Scheduling Problem

Qianjun Xu

Oct. 30, 2001

Description of Job Shop Scheduling

- A finite set of n jobs
- Each job consists of a chain of operations
- A finite set of m machines
- Each machine can handle at most one operation at a time
- Each operation needs to be processed during an uninterrupted period of a given length on a given machine
- Purpose is to find a schedule, that is, an allocation of the operations to time intervals to machines, that has minimal length

Formal Definition of JSS

- Job set $J = \{ j_1, j_2, \ldots j_n\}$
- Machine set $M = \{m_1, m_2, \ldots m_m\}$
- Operations $O = \{o_1, o_2, \ldots o_n\}$ $O_i = \{o_{i1}, o_{i2}, \ldots o_{im_i}\}$
- Each operation has processing time $\{\tau_{i1}, \tau_{i2}, \ldots \tau_{im_i}\}$
- On O define A, a binary relation represent a precedence between operations. If $(v, w) \in A$ then v has to be performed before w.
- A induce the total ordering belonging to the same job; no precedence exist between operations of different jobs.

Formal Definition of JSS cont.

- A schedule is a function $S: O \rightarrow IN \cup \{0\}$ that for each operation v defines a start time S(v).
- A schedule S is feasible if

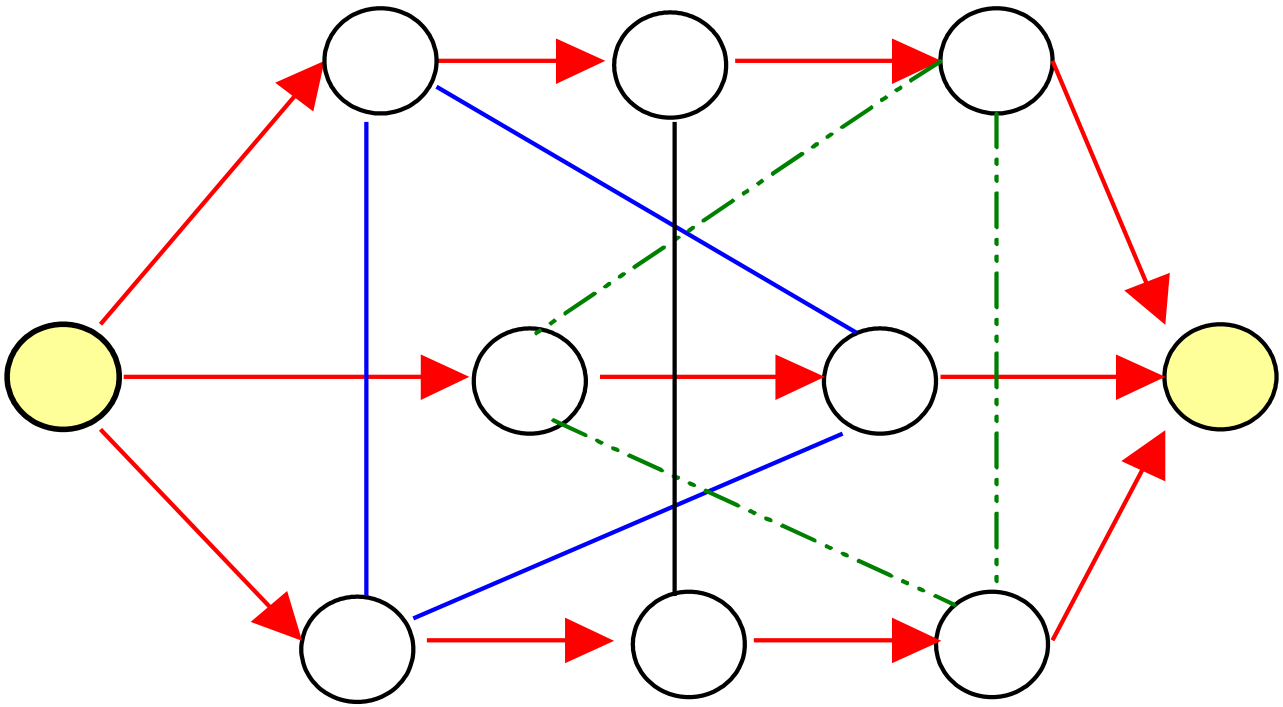
$$\forall v \in O: \qquad S(v) \geq 0$$
$$\forall v, w \in O,\ (v,w) \in A: \qquad S(v) + \tau(v) \leq S(w)$$
$$\forall v, w \in O,\ v \neq w,\ M(v) = M(w): \ S(v) + \tau(v) \leq S(w)\ or$$
$$S(w) + \tau(w) \leq S(v)$$

- The length of a schedule S is $len(S) = \max_{v \in O}(S(v) + \tau(v))$
- The goal is to find an optimal schedule, a feasible schedule of minimum length., min(len(S)).

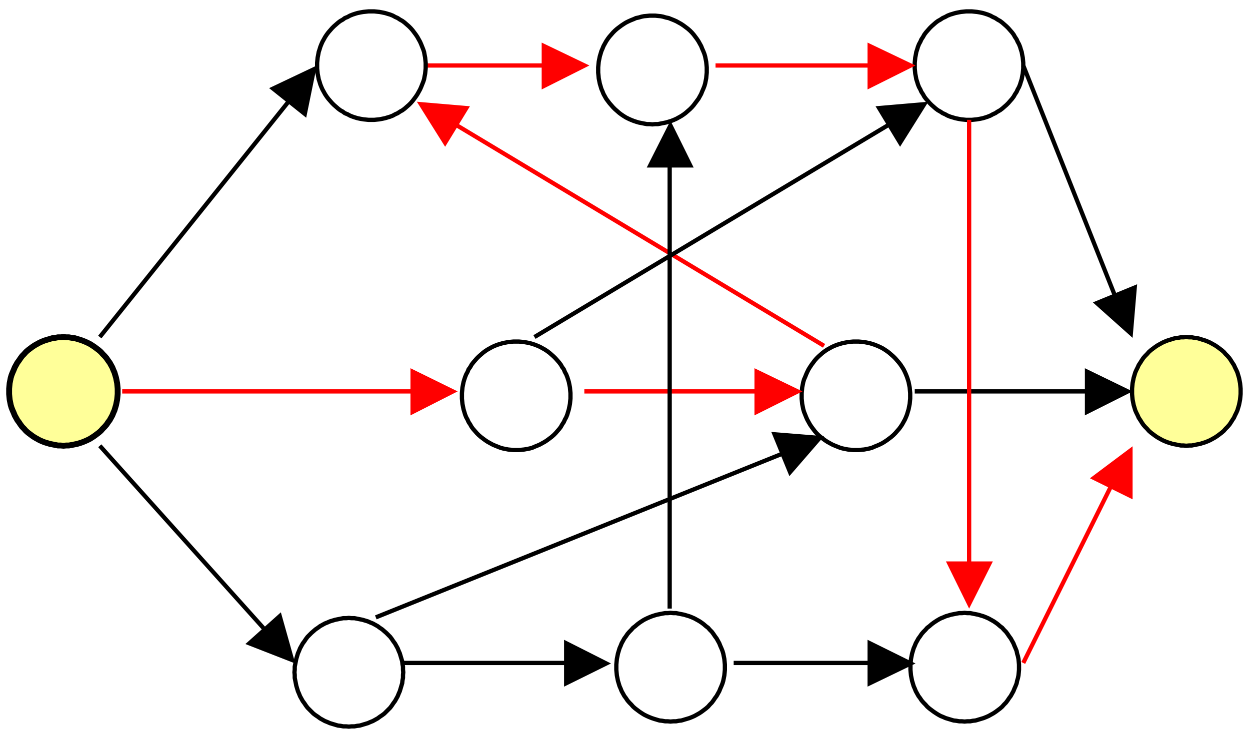
Disjunctive Graph

- An instance of the JSS problem can be represented by means of a **disjunctive graph** G=(O, A, E).
- The vertices in O represent the operations
- The arcs in A represent the given precedence between the operations
- The edge in $E = \{(v, w) \mid v, w \in O,\ v \neq w,\ M(v) = M(w)\}$ represent the machine capacity constraints
- Each vertex v has a weight, equal to the processing time $\tau(v)$

Example of Disjunctive Graph



Disjunctive Graph with Edge Orientations



Disjunctive Graph Cont.

- Finding an optimal feasible schedule is equivalent to finding an orientation E' that minimizes the longest path length in the related digraph.

Why JSS Problem

- It is considered to be a good representation of the general domain and has earned a reputation for being notoriously difficult to solve
- JSS is considered to belong to the class of decision problems which are NP
- Lenstra et al(1977) Show that
  - 3*3 problem
  - N*2 instance with no more than 3 operations per job
  - N*3 problem with no more than 2 operations per job
  - N*3 problem where all operations are of unit processing time

Belong to the set of NP instances.

Methods to Solve JSS

- **Mathematical Formulations:**
  - mixed integer linear programming (1960)
- **Branch and Bound**
- **Approximation Methods**
  - Priority dispatch rules
  - Bottleneck based heuristics
  - Artificial intelligence(constraint satisfaction approach, neural networks)
  - Local search methods

Branch and Bound

- Using a dynamically constructed tree structure represents the solution space of all feasible sequences
- Search begins at topmost node and a complete selection is achieved once the lowest level node has been evaluated
- Each node at a level p in the search tree represent a partial sequence of p operations
- From an unselected node the branching operation determines the next set of possible nodes from which the search could progress
- The bounding procedure selects the operation which will continue the search and is based on an estimated LB and currently best achieved UB. IF at any node the estimated LB is found to be greater than the current best UB, this partial selection and all its subsequent descendants are disregarded.

Priority Dispatch Rules

- At each successive step all the operations which are available to be scheduled are assigned a priority and the operation with the highest priority is chosen to be sequenced.
- Usually several runs of PDRs are made in order to achieve valid results.

Constraint Satisfaction Approach

- Aiming at reducing the effective size of the search space by applying constraints that restrict the order in which variables are selected and the sequence in which possible values are assigned to each variable
- Constraint propagation
- Backtracking
- Variable heuristic
- Value heuristic

Neural Networks

- Hopfield networks
- Back-error propagation networks

Local Search Method

- Configurations: a finite set of solutions.
- Cost function to be optimised.
- Generation mechanism, generating a transition from one configuration to another.
- Neighborhood, N(x), is a function which defines a simple transition from a solution x to another solution by inducing a change.
- Selection of neighborhood:
  - chose the first lower cost neighbor found;
  - select the best neighbor in the entire neighborhood;
  - Choose the best of a sample of neighbors.

Summary

- The definition of JSS
- The disjunctive graph
- The methods to solve JSS