1000-9825/2001/12(01)0011-07 ©2001 Journal of Software 软 件 学 报 Vol. 12, No. 1

Solving Multiple Hoist Scheduling Problems by Use of Simulated Annealing*

YANG Guang-wen[1], JU Da-peng[1], ZHENG Wei-min[1], Kokin Lam[2]

[1](Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China);
[2](Department of Management Science, City University of Hong Kong, Hong Kong, China)
E-mail: {ygw,ju}@tsinghua.edu.cn
http://www.tsinghua.edu.cn
Received July 6, 1999; accepted December 20, 1999

Abstract: The multiple hoist schedule problem is a critical issue in the design and control of many manufacturing processes. When the hoist number and station number are very large, finding optimal schedule is very hard. In this paper, the mathematical model of multiple hoist schedule problem is discussed and a heuristic method about it is proposed by use of simulated annealing. A large number of examples of random case indicate that this heuristic method is very useful, and satisfactory solution can be obtained with small amount of computation.

Key words: multiple hoist schedule problem; simulated annealing; zone partition; simple cyclic schedule

Many industrial processes employ computer-controlled hoists for handling material. The hoists are programmed to perform a fixed sequence of moves repeatedly. It is called a Hoist Scheduling Problem. This problem has been a critical issue in the design and control of many manufacturing processes and is of increasing importance as more and more manufacturers become aware of the issue. Many researchers have worked in this area in the past two decades, but most previous approaches toward solving cyclic hoist problems have been limited to single-hoist or two-hoist cases[1,2]. Lam *et al.* have studied the multiple hoist schedule problem[3], and the method can only obtain a local minimum solution. Using simulated annealing technique, we propose a new heuristic method that is suitable for finding schedules for systems with multiple hoists.

The multiple hoist scheduling problem is defined as follows. There are $N+1$ workstations, $S_0, S_1, \ldots, S_N$, and M identical hoists that move jobs between stations. S_0, S_N and $S_i (1 \leqslant i \leqslant N-1)$ stand for the input buffer, output buffer and station i, respectively.

The following are some concepts of the problem.

move: A move between successive stations consists of three simple hoist operations: (1) lift a job from a station; (2) move the job to the next station; (3) lay the job on the station.

---

* This project is supported by the National High Technology Development Program of China under Grant No. 863-306-ZD11-01-9 (国家863高科技项目基金). **YANG Guang-wen** was born in 1963. He is an associate professor of the Department of Computer Science and Technology, Tsinghua University. His current research areas include design of algorithms, parallel computing. **JU Da-peng** was born in 1967. He is a lecturer of Department of Computer Science and Technology, Tsinghua University. His current research areas include parallel and distributed processing. **ZHENG Wei-min** was born in 1945. He is a professor of Department of Computer Science and Technology, Tsinghua University. His current research areas include parallel and distributed processing. **Kokin Lam** was born in 1957. He is an associate professor of Department of Management Science, City University of Hong Kong. His current research areas include combinational optimization and multiple hoist scheduling.

cycle: The time spent in performing the fixed sequence of moves in the repeated sequence of moves (the moves performed by M hoists follow a cyclic pattern).

l-degree schedule problem: Each move, $m_i (0 \leqslant i \leqslant N)$, is performed exactly l times in a cycle.

time window constraints: They include the following constraints:

processing time constraints: The processing time for a job at station i must be no less than L_i and no more than $U_i (0 \leqslant i \leqslant N)$, where L_i and U_i are given time limits.

traveling time constraints: There must be enough time for a hoist to travel between successive assigned moves.

operation assignment constraints: One hoist can move only one job, and each job requires only one hoist.

on line constraints: Once a job is removed from S_0, it must then be processed at each of N stations one after another without encountering any intermediate delay.

collision constraints: Any two hoists can not cross each other in a cycle.

Additional assumptions

(1) Jobs are identical and each job has to visit all stations in the order that stations are numbered.

(2) We only consider a 1-degree schedule problem, that is to say, exactly one job is removed from each station in a cycle, and therefore, one job enters and one job leaves the system in a cycle.

Let φ be a permutation which orders $N+1$ distinct numbers, $\{0,1,\ldots,N\}$, into a sequence: $\{\varphi(0),\varphi(1),\ldots,\varphi(N)\}$, such that $\varphi(i)=k$, $0 \leqslant i,k \leqslant N$. For a given sequence $\{m_{\varphi(0)},m_{\varphi(1)},\ldots\}$, $m_{\varphi(i)}$ is the i-th move to be performed in each cycle, $i=1,2,\ldots,N$, ($\varphi(i)=k$ means that the i-th move is moving the job in station k to station $k+1$). Let ψ be a permutation, which orders $N+1$ distinct numbers, $\{0,1,\ldots,N\}$, into a sequence: $\{\psi(0),\psi(1),\ldots,\psi(N)\}$, such that $\psi(i)=j$, ($0 \leqslant i \leqslant N$ and $1 \leqslant j \leqslant M$). $\psi(i)=j$ means that move $m_{\varphi(i)}$ is assigned to hoist j. We also introduce notation $T=\{0=T_{\varphi(0)} \leqslant T_{\varphi(1)} \leqslant \ldots \leqslant T_{\varphi(N)}\}$, such that $T_{\varphi(i)}$ is the time at which move $m_{\varphi(i)}$ starts in a cycle, $i=0,1,2,\ldots,N$. Each three-tuple (φ,ψ,T) defines a multiple hoist cyclic schedule. A cyclic schedule (φ,ψ,T) is feasible if and only if it satisfies the four constraints: time window constraints, operation assignment constraints, on line constraints and collision constraints.

For any feasible cyclic schedule (φ,ψ,T), the resulting cycle time $\Pi(\varphi,\psi,T)$ is the time required by M hoists to perform all their assigned moves. The multiple hoist cyclic schedule problem is thus to find a feasible cyclic schedule (φ^*,ψ^*,T^*), such that

$$\Pi(\varphi^*,\psi^*,T^*)=\min\{\Pi(\varphi,\psi,T) \quad \text{for all feasible } (\varphi,\psi,T)\}$$

The number of alternative schedules increases exponentially with N and M, so that in practice an exact optimal solution can be attempted on a multiple hoist schedule problem involving a few stations and hoists (in general only single-hoist or two-hoist problems are considered). For large M and N, a heuristic method is needed for finding the solutions of the scheduling problem. Heuristic method is designed to find a near-optimal solution (that is a satisfactory solution). We shall discuss a heuristic method for multiple hoist scheduling problem in the following.

1 Multiple Hoist Schedule Model with Zoned Partitions

We first consider mathematical model of the multiple hoist schedule problem with zoned partitions.

Let $P=\{\Omega_1,\Omega_2,\ldots,\Omega_M\}$ be an arbitrary zoned partition, where $\Omega_m (1 \leqslant m \leqslant M)$ denote the sequence of contiguous stations in the m th zone, and the cycle time with respect to φ,ψ,T,P is denoted by $\Pi(\varphi,\psi,T,P)$. In a zoned partition, contiguous stations are grouped into a zone, the boundary station in each partition can be conceptually considered as the dummy input buffer or output buffer and each zone is then exclusively assigned to a single hoist. In this way, the collision constraints are satisfied in nature. All the moves in partition Ω_m are finished by m-th hoist. The minimum cycle time is denoted by $\Pi(P)$ for given P. For each Ω_m, we consider the sub-problem with stations in Ω_m, and a single hoist m. As discussed above, the cycle time is denoted by $\Pi(\varphi_m,\psi_m,T_m,\Omega_m)$. Let

o_i be the time required by hoist to perform move m_i,

$c_{j,k}$ be the time to travel (when not carrying a job) from station j to station k for a hoist,

$$x_{i,j}=\begin{cases}1 & \text{if move } m_j \text{ immediately succeeds move } m_i,\\ 0 & \text{otherwise,}\end{cases}$$

$$Y_i=\begin{cases}1 & \text{if move } m_i \text{ is performed prior to move } m_{i-1} \text{ in the cycle,}\\ 0 & \text{otherwise,}\end{cases}$$

$$F_i=\begin{cases}1 & \text{if move } m_i \text{ is the first operation performed by a hoist in the cycle,}\\ 0 & \text{otherwise,}\end{cases}$$

$x_{i,j}, Y_i, F_i$ depend on the selection of (φ_m,ψ_m).

Consider the optimization problem (SchP):

$$\min \quad \Pi(\psi_m,\varphi_m,T_m,\Omega_m)$$

$$\text{s.t.} \quad L_{\varphi_m(i)}\leqslant(t\,\varphi_m(i)+\Pi(\varphi_m,\psi_m,T_m,\Omega_m)Y_{\varphi_m(i)})-(t_{\varphi_m(i)-1})\leqslant U_{\varphi_m(i)} \tag{1}$$

$$(t_{\varphi_m(i)}+(O_{\varphi_m(i)}+c_{\varphi_m(i)+1,\varphi_m(j)})x_{\varphi_m(i),\varphi_m(j)})x_{\varphi_m(i),\varphi_m(j)}+(x_{\varphi_m(i),\varphi_m(j)}-1)\Pi(\varphi_m,\psi_m,T_m,\Omega_m) \tag{2}$$
$$\leqslant t_{\varphi_m(j)}+\Pi(\varphi_m,\psi_m,T_m,\Omega_m)F_{\varphi_m(j)}$$

$$\sum_{\varphi_m(j)\in\Omega_m} x_{\varphi_m(i),\varphi_m(j)}=1$$
$$\sum_{\varphi_m(i)\in\Omega_m} x_{\varphi_m(i),\varphi_m(j)}=1 \tag{3}$$
$$\varphi_m(i),\varphi_m(k)\in\Omega_m,\ m=1,2,\ldots,M$$

In above constraints, Eqs. (1), (2), (3) stand for the processing time constraints, traveling time constraints and operation assignment constraints respectively. On line constraints are also satisfied according to Eqs. (1), (2), (3). For each partition P, SchP_m is a single hoist scheduling problem, and there always exist feasible solutions for problem $\text{SchP}_m\,(m=1,2,\ldots,M)$.

Using the algorithm of Armstrong[1], we can solve the problem SchP_m and obtain the solution $\Pi(\Omega_m)$ (the minimum cycle time for Ω_m).

Lam[3] put forward an algorithm for finding the minimum cycle time $\Pi(P)$:

Algorithm A

```
BEGIN
    Solving problem SchP_m (i=1,2,...,M)
    Π_min(P)=min{Π(Ω_i), i=1,2,...,M};
    Π_max(P)=max{Π(Ω_i), i=1,2,...,M};
    finished=false;
    Π(P)=Π_max(P);
    REPEAT
         BEGIN
              finished=true;
                FOR i=1,2,...,M DO
                  BEGIN
                      Add the constraint Π(φ_m,ψ_m,T_m,Ω_m)≥Π(P) to SchP_m and solve
                      the problem;
                      IF Π(Ω_i)>Π(P) THEN
                        BEGIN
```

```
                    Π(P)=Π(Ω_i);
                    finished=false;
                END
            END
        END
    UNTIL finished=true;
    Adjust the starting time of the first move for each zone and the optimal schedule is obtained;
END
```

2 Solving Multiple Hoist Schedule Problems by Using of Simulated Annealing

For any initial partition $P=\{\Omega_1,\Omega_2,...,\Omega_M\}$, using Algorithm A, we can obtain a minimum cycle time $\Pi(P)$, it is only a local optimal solution to multiple hoist schedule problem and highly sensitive to P_0(the initial partition). How to find a global optimal solution is a critical issue for the researchers.

In general, for a scheduling system of $N+1$ stations and M hoists, there are $N!\ (M!)^2M^{N-2M+1}$ alternative schedules. In practice, $N\gg M$, but the number of alternative schedules is still very large.

Using the annealing process, Metropolis[4] introduced a simple algorithm that can be used to provide an efficient simulation of a collection of atoms in equilibrium at a given temperature. It has been used in many fields[5]. Let E stand for the energy function of the system, T stand for the temperature. In each step of this algorithm, an atom is given a small random displacement and the resulting change, $\Delta E=E(T-\Delta)-E(T)$, in the energy of the system is computed. If $\Delta E\leqslant 0$, the displacement is accepted, and the configuration with the displaced atom is used as the starting point of the next step. The case $\Delta E>0$ is treated randomly; the probability that the configuration is accepted is $p(\Delta E)=\exp(-\Delta E/k_BT)$ (where k_B is Boltzmann constant, and we can use T to replace k_BT later). Random numbers uniformly distributed the interval (0,1) are a convenient means of implementing the random part of the algorithm. One such number is selected and compared with $p(\Delta E)$. If it is less than $p(\Delta E)$, the new configuration is retained; if not, the original configuration is used to start the next step. By repeating the thermal motion of atoms in thermal contact with a heat bath at temperature T, this choice of $p(\Delta E)$ results in that the system evolves to a Boltzmann distribution.

Annealing, as implemented by the Metropolis procedure, differs from iterative improvement. The transitions out of a local optimum are always possible at nonzero temperature.

According to annealing process, we can use the simulated annealing to multiple hoist schedule problem as follows.

Let $P=\{\Omega_1,\Omega_2,...,\Omega_M\}$ be a configuration. $\Pi(P)$ is the cost function. Introduce the temperature T. All the partitions are feasible. The simulated annealing algorithm is described as follows for multiple hoist schedule problem with zoned partition:

Algorithm B

```
BEGIN
    P_0={Ω_1,Ω_2,...,Ω_M};          /initial partition
    Using Algorithm A to find the solution Π(P_0) with respect to P_0;
    P*=P_0, Π*=Π(P_0)
    T=T_0;                          /initial temperature
    REPEAT
        REPEAT
            P=GENERATE(P_0);
```

Using Algorithm A to find the solution $\Pi(P)$ with respect to P;

IF $\Pi(P) \leqslant \Pi(P_0)$;

THEN $P_0 = P$;

$P^* = P_0$, $\Pi^* = \Pi(P_0)$

ELSE

IF $\exp\left(-\frac{\Pi(P) - \Pi(P_0)}{T}\right) > \text{Random}(0,1)$

THEN $P_0 = P$;

ELSE

CONTINUE;

UNTIL inner loop stops;

$T = \alpha^* T$ /temperature cooling schedule;

UNTIL stopping criterion is satisfied;

END

P^* and Π^* are the minimum zoned partition and minimum cycle time respectively so far obtained by Algorithm B.

Some notes about Algorithm B:

(1) The selection of initial partition P_0 and T_0:

Decompose $\{0,1,\ldots,N\}$ to M zones $\Omega_i (i=1,2,\ldots,M)$ such that each Ω_i has the same number of stations as much as possible, that is $|\Omega_j| = |\Omega_i|$, $|\Omega_i|+1$ or $|\Omega_i|-1$. We can also decompose $\{0,1,\ldots,N\}$ to M zones Ω_i $(i=1,2,\ldots,M)$ randomly.

According to the physical background of annealing, T_0 must be high enough to obtain global optimal solution. It can be selected according to experience.

(2) The definition of GENERATE(P_0)

In Algorithm B, GENERATE(P_0) is a map to generate new partition. It can be considered as obtaining a neighbor of partition P_0. We use four tactics:

a) Change the bottleneck zone (corresponding to a single-hoist problem, the cycle time is the largest among all zones), and the "smallest" zone (the cycle time of this zone is the smallest).

Suppose that $P_0 = \{\Omega_1^0, \Omega_2^0, \ldots, \Omega_M^0\}$, the boundary points of stations are $0 = k_0^0 \leqslant k_1^0 \leqslant \ldots k_{M-1}^0 \leqslant k_M^0 = N$, $\Pi(\Omega_j^0) = \min\{\Pi(\Omega_m^0) \mid m=1,2,\ldots,M\}$, and $\Pi(\Omega_i^0) = \max\{\Pi(\Omega_m^0) \mid m=1,2,\ldots,M\}$. We can obtain the new partition $P = \{\Omega_1, \Omega_2, \ldots, \Omega_M\}$ and the boundary points are: if $i<j$, then, $k_i = k_i^0 + 1$, $k_{j-1} = k_{j-1}^0 - 1$, and $k_l = k'_l (l \neq i, j-1)$; if $i>j$, then $k_{i-1} = k_{i-1}^0 - 1$, $k_j = k_j^0 + 1$, and $k_l = k_l^0 (l \neq i-1, j)$. We can obtain a new partition $P = \{\Omega_1, \Omega_2, \ldots, \Omega_M\}$ (the boundary points of Ω_i are k_{i-1} and k_i). Where the perturbation step length is selected as one, we can also select it randomly.

b) Perturb the partition P_0 randomly (that is to say we can change all the boundary points of P_0 randomly), and obtain the new partition P. The random step length may be 1 or more.

c) Only change the boundary points of bottleneck zone according to P_0. The randomly step length may be 1 or more.

d) Obtain the new partition P randomly (in this case, P is not related to P_0).

(3) The stopping criterion of algorithm

Inner loop stopping criteria: Under temperature T, suppose the search sequence of partition is $P^{(0)}, P^{(1)}, \ldots, P^{(i)}, \ldots$. Given an integer n_0, if $P^{(i)} = P^{(i+1)} = \ldots = P^{(i+n_0)}$ for some i, then the inner loop stops. We can also give the maximum iteration number in the inner loop.

Outer loop stopping criteria: For integers k, p_0, if the cycle time of the multiple hoist schedule problem is the same as temperature $T_k, T_{k+1}, \ldots, T_{k+p_0}$, then the stopping criterion is satisfied. We can also give the minimum value of the temperature in the outer loop.

(4) The changing manner of temperature

The changing manner of temperature is $T_{k+1} = \lambda T_k (0 < \lambda < 1)$.

In simulated annealing algorithm, the acceptance criterion (bad partition may be accepted) may let the iteration jump out from one local minimum area to another, to finally reach the global optimal area. The simulated annealing algorithm can get the global optimization solution of the problem. The determination of the upper boundaries of computation and complexity of the algorithm is very difficult. However, according to Fox's results[6], the time complexity of our method is often essentially linear with problem size.

3 Simulation and Conclusion

We have evaluated the performance of Algorithm B in terms of random case.

In the random case examples, we select L_i to be a random number $[a,b]$, $U_i = \alpha L_i$, where α is a constant. $c_{i,i+1}$ is a random number in $[c,d]$, $o_i = c_{i,i+1} + e$, and $c_{i,j} = \sum_{k=i}^{j-1} c_{k,k+1} (i < j)$. a,b,c,d, and e are constants. The examples indicate that the algorithm can solve multiple hoist scheduling problem efficiently, especially for the large scale problem. For a given N, with M increasing, the computing time increases, reaches an upper boundary, then decreases. The effective combination of four tactics for map GENERATE will accelerate the algorithm. Stopping criterion and annealing schedule are very important in determining a satisfactory solution.

We have discussed the multiple hoist scheduling problem, applied simulated annealing to it and proposed a heuristic method about the problem. The computation results state clearly that this method is very useful. We can also obtain the optimal partition zones in theory, but in practice, we can obtain a satisfactory solution and need less computing time depending on the selection of stopping criterion and annealing schedule. Large scale multiple hoist scheduling problem can be solved by the method efficiently. But there are some problems to be solved.

(1) How to control the process of the cooling schedule is very important in obtaining the optimal solution (or a satisfactory solution).

(2) The method of GENERATE procedure can be considered in many deterministic structures or random structures. The performance of the algorithm will be better by selecting an appropriate generation method.

References:

[1] Armstrong, R.D., Lei, L., Gu, S. A bounding schedule for deriving the minimal cycle time of a single hoist *n*-stage process with time-window constraints. European Journal of Operational Research, 1994,75:133~140.

[2] Kolen, A., Rinnooy, K.A., Trienekens, H. Vehicle routing with time windows. Operations Research, 1987,35:266~273.

[3] Kokin Lam, *et al*. A study of heuristics for bidirectional multiple hoist production scheduling system. International Journal of Production Economics, 1994,33:207~214.

[4] Gelatt K.S., Vecchi, M.P. Optimization by simulated annealing. Science, 1983,220:671~680.

[5] Liu Hsiao Chung, Huang Jeng sheng. Pattern recognition using evolution algorithms with fast simulated annealing. Pattern Recognition Letters, 1998,19(5):403~413.

[6] Fox, G.C. Physical computation concurrence. Practice and Experience, 1991,3(6):627~653.

利用模拟退火技术求解多 Hoist 调度问题

杨广文[1]， 鞠大鹏[1]， 郑纬民[1]， 林国健[2]

[1](清华大学 计算机科学与技术系，北京 100084)；

[2](香港城市大学 管理科学系，香港)

摘要：多 Hoist 调度在许多制造过程的设计与控制中是一个关键问题.当 Hoist 数与工作台数很大时，求解调度问题非常困难.建立了多 Hoist 调度问题的数学模型，并利用模拟退火算法提出了一种启发式求解方法.随机模拟的大量算例表明，该启发式方法十分有效，通过少量的计算可得到调度问题的一个满意解.

关键词：多 Hoist 调度问题；模拟退火算法；区域划分；简单循环调度

中图法分类号：TP391　　**文献标识码**：A