ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1994
Seria: AUTOMATYKA z.114 Nr kol. 1250

Czesław SMUTNICKI
Politechnika Wrocławska

ALGORYTM TABU SEARCH DLA PEWNEJ KLASY PROBLEMÓW SZEREGOWANIA NA JEDNEJ MASZYNIE

Streszczenie: Rozważany jest problem szeregowania zadań na jednej maszynie z terminami gotowości, terminami dostarczenia oraz kryterium minimalizacji maksymalnego terminu zakończenia wykonywania zadań. Zaproponowano nowy algorytm aproksymacyjny dla tego problemu oparty na technice poszukiwania tabu. Przedstawiono także wyniki analizy eksperymentalnej algorytmu.

TABU SEARCH ALGORITHM FOR A CLASS OF SINGLE-MACHINE SCHEDULING PROBLEMS

Summary: The problem of scheduling jobs with release times and delivery times on a single machine is considered. The new approximation algorithm based on the tabu search technique is presented. Results of the experimental analysis have also been devised.

АЛГОРИТМ ПОИСКА С ЗАПРЕЩЕНЯМИ ДЛЯ КЛАССА ЗАДАЧ СОСТАВЛЕНИЯ РАСПИСАНИЙ НА ОДНОЙ МАШИНЕ

Резюме: В статье рассматривается проблема упорядочения задач на одной машине со сроками готовноности, временами поставки и критерием оконьчания всех задач. Представлен новый аппроксимационный алгоритм, основан на методе поиска с запрещенями. Приведены тоже результаты экспериментального анализа.

1. The problem

The considered problem can be briefly described as follows. A set of jobs $N=\{1,2,...,n\}$ should be processed on a single machine. The machine can process at most one job at a time and preemption is not allowed. Each job j has a release time r_j, a processing time $p_j > 0$, and a delivery time q_j. The job must begin processing on the machine sometime after its release time, and its delivery begins immediately after processing has been completed. All jobs may be simultaneously delivered. The objective is to find a sequence of jobs which minimizes the time by which all jobs are delivered (denotation $1|r_j, q_j|C_{max}$).

The stated problem has been known in the literature for years as:

(i) that with release times, due dates instead of delivery times and maximum lateness criterion ($1|r_j|L_{max}$,) [10],

(ii) with non-bottleneck machines instead of release times and delivery times, [13]. Due to the forward-backward symmetry, the model (ii) has been studied more frequently than others.

The problem $1|r_j, q_j|C_{max}$ has received considerable attention in past twenty years and has been employed among others in scheduling jobs the on critical machine ([13]), in approximation algorithms for the job-shop problem ([1]), as a lower bound for the flow-shop and job-shop problems ([3],[5]).

2. Background and definitions

Let π be a permutation on N. Denote by $\pi(i)$ the element of N which is in position i of π. The permutation π defines a job processing order on the machine. The time by which all jobs are delivered is given by $C_{max}(\pi)=\max_{1\leq i\leq j\leq n}(r_{\pi(i)}+\sum_{k=i}^{j} p_{\pi(k)}+ q_{\pi(j)})$.
A permutation π^* which minimizes $C_{max}(\pi)$ over all permutations on N is optimal and let $C^*=C_{max}(\pi)$. Each pair of integers (i,j), $1\leq i\leq j\leq n$ is called a path in π. The path u=(a,b) which maximizes the right hand of formulae on $C_{max}(\pi)$ is called a critical path in π. The sequence $B=(\pi(a), \pi(a+1),...,\pi(b))$ is called a block in π. Block B contains b-a+1 jobs.

3. Solution methods

It has been shown in [14] that the problem is NP-hard in the strong sense, however, there exist polynomial algorithms for special cases [13]. Enumerative methods for solving the problem have been studied in [2],[16],[4],[9]. Results of computer tests pointed out Carlier's algorithm [4] as the most promising approach, especially for problems of large size. Although these algorithms have been tested on benchmarks up to 500 [6] and 1000 [4] jobs, their practical application own some limitations. Already some authors reported the phenomenon of explosion of calculations observed in Carlier's algorithm used in Adams' [1] shifted-bottleneck method.

Approximation algorithms have been studied in [23], [22] and [11]. Schrage proposed [23] an algorithm (S) which employs the following rule: whenever the machine is free and one or more jobs are available for processing, schedule an available job with largest delivery time. In [15] it has been proved that algorithm S generates schedules not longer than 2 times the length of the optimal schedule (2-approximation algorithm). An implementation of algorithm S which runs in O(nlogn) time is provided in [4]. Modifying the algorithm S, Potts has obtained a 3/2-approximation algorithm (P) which runs in $O(n^2 \log n)$ time [23]. Hall and Shmoys have proposed a modified version (HS) of Potts' algorithm which is a 4/3-approximation algorithm and also runs in $O(n^2 \log n)$ time [11]. This is the best approximation algorithm yet known. Very recently a quick 3/2-approximation algorithm (H) which runs in O(nlogn) time has been proposed in [18].

Approximation algorithms, beside conventional application, act also as auxiliary algorithms. For example Carlier's enumerative algorithm generates at every node of the search tree a complete solution using the algorithm S. The solution is applied to upper bound modification and tree generation (branching rule). Replacing S by other approximation algorithms in order to get better upper bound could imply fastest convergence to the optimal solution.

4. Tabu search

Tabu search (TS) is a metaheuristic designed for finding near-optimal solutions of combinatorial optimization problems [7],[9]. It belongs to the class of so-called improving methods and is composed of several specific elements called the move, neighborhood, searching strategy, tabu list, aspiration function, stopping rule. The move is a function which transforms a solution into another solution. The neighborhood of a given solution is a subset of solutions generated from this solution using moves. TS start from an initial solution. At each step the neighborhood of a given solution is searched through in order to find a neighbor, usually the best in the neighborhood. Next, the move which has conducted to the found neighbor is performed and the obtained new solution is set as the primal for the next step. In order to avoid cycling and become trapping in a local optimum there exists a circular tabu list which determines which moves are forbidden. However we can perform a forbidden move if the aspiration function evaluates its as profitable.

4.1. Moves and neighborhood

The neighborhood of π is generated by a set of moves. Among many types of moves considered in the literature the following three have been applied commonly: (i) exchange jobs placed at the a-th and the b-th position (we call it E-move), (ii) remove the job placed at the a- th position and put it at b-position (I-move). Results obtained for problems with C_{max} criterion [24],[19],[20] showed that I-moves are better than E-moves from both efficiency and effectiveness of TS point of view. Therefore only this type of moves will be consider further on.

Let $v=(a,b)$ be a pair of positions $a,b\in\{1,...,n\}$, $a\neq b$ in the permutation π. We define the new permutation π_v obtained from π by removing the job $\pi(a)$ from the position a and inserting it in position b as follows: $\pi_v=(\pi(1),...,\pi(a-1),\pi(a+1),...,\pi(b),\pi(a),\pi(b+1),...,\pi(n))$ if $a<b$ and $\pi_v=(\pi(1),...,\pi(b-1),\pi(a),\pi(b+1),...,\pi(a-1),\pi(a+1),...,\pi(n))$ if $a>b$. Each pair $v=(a,b)$ defines a move from π and let U be a set of such pairs. The neighborhood of a permutation π generated by a move set U will be denoted by $N(U,\pi) = \{\pi :v\in U\}$. The neighborhood generated by the move set $V=\{(a,b):b\notin\{a-1,a\},a,b\in\{1,2,...n\}\}$ is one of the biggest [25]. Note, that for a,b such that $|a-b|=1$ two moves $v=(a,b)$ and $v'=(b,a)$ yield the same permutations $\pi_v=\pi_{v'}$ (we call these moves equivalent). So, in order to avoid redundancy, V contains exactly one from each pair of equivalent moves. $N(V,\pi)$ has $(n-1)^2$ neighbors and satisfies so-called connectivity property: for any $\pi^0\in\Pi$ exists a finite sequence $\pi^0,\pi^1,...,\pi^r$ such that π^r is an optimal processing order and $\pi^{i+1}\in N(V,\pi^i)$, $i=0,...,r-1$.

Among few drawbacks of $N(V,\pi)$, the too high computational complexity of a single neighborhood searching seems to be dominant. This drawback can be eliminated using two approaches: (1) evaluate neighbors by lower bounds and do not calculate makespans (it is assumed that the computational complexity of a lower bound is essentially less than that of makespan) [24], (2) reduce the neighborhood size and calculate the makespan for each neighbor [19],[20]. Employing the second approach we propose some reduced neighborhoods based on so called block of jobs properties [9].

Consider subsets $\underline{W}(\pi),W(\pi),\overline{W}(\pi)\subset V$, which depend on π and $u=(a,b)$, and are defined as follows: $\overline{W}(\pi)=\{(i,j):\ i,j\in\{1,...,a-1\}\}$, $\overline{W}(\pi)=\{(i,j):\ i,j\in\{a+1,...,b-1\}\}$, $\overline{W}(\pi)=\{(i,j):\ i,j\in\{b+1,...,n\}\}$. The neighborhood $N(\underline{W}(\pi)\cup W(\pi)\cup\overline{W}(\pi),\pi)$ contains $\max\{0,a-1\}^2 + \max\{0,b-a-1\}^2 +\max\{0,n-b\}^2$ permutations obtained by I-moves limited to positions $1,...,a-1$, positions $a+1,...,b-1$ within block B (the internal part of the block), and positions $b+1,...,n$. By [9] the following property holds.

Property 1. For any permutation $\beta\in N(\underline{W}(\pi)\cup W(\pi)\cup\overline{W}(\pi),\pi)$ we have

$$C_{max}(\beta) \geq C_{max}(\pi). \square$$

In other words, moves from $V\setminus(\underline{W}(\pi)\cup W(\pi)\cup\overline{W}(\pi))$ are "more interesting" than those from $\underline{W}(\pi)\cup W(\pi)\cup\overline{W}(\pi)$, taking into account only the immediate improvements point of view.

Consider another neighborhood generated by the move set $Z(\pi)\subseteq V\setminus(\underline{W}(\pi)\cup W(\pi)\cup\overline{W}(\pi))$ defined as follows: $Z(\pi)=Z'(\pi)\cup Z''(\pi)$, where $Z'(\pi)=\{(a,j):\ a<j\leq b\}$ and $Z''(\pi)=\{(j,b):\ a<j<b\}$. The neighborhood $N(Z(\pi),\pi)$ contains $2(b-a-1)+1$ permutations obtained by some selected I-moves. More precisely, each job $\pi(j)$, $j=a+1,...,b-1$, is inserted only in two positions a and b in

π(some exceptions occur for jobs $\pi(a)$ and $\pi(b)$). By [9] the following property can be proved.

Property 2.
(a) For any permutation $\beta \in N(Z'(\pi),\pi)$ such that $r_{\pi(j)} \geq r_{\pi(a)}$ we have $C_{max}(\beta) \geq C_{max}(\pi)$.
(b) For any permutation $\beta \in N(Z''(\pi),\pi)$ such that $q_{\pi(j)} \geq q_{\pi(b)}$ we have $C_{max}(\beta) \geq C_{max}(\pi)$. □

Note, that only a part of the neighborhood $N(Z(\pi),\pi)$ is really interesting.

Now consider the neighborhood generated by the move set $X(\pi) \subseteq Z(\pi)$ defined as follows: $X(\pi)=X'(\pi) \cup X''(\pi)$, where $X'(\pi)=\{(a,j): a<j\leq b, r_{\pi(j)} < r_{\pi(a)}\}$ and $X''(\pi)=\{(j,b): a<j<b, q_{\pi(j)} < q_{\pi(b)}\}$. The neighborhood $N(X(\pi),\pi)$ contains at most $2(b-a-1)+1$ permutations from $N(Z(\pi),\pi)$, excluding these satisfying Property 2.

The next (fourth) neighborhood is generated by the move set $Y(\pi) \subseteq X(\pi)$, $|Y(\pi)|=2$, defined as follows: $Y(\pi)=Y'(\pi) \cup Y''(\pi)$, where $Y'(\pi)=\{(a,k): r_{\pi(k)} = \min_{a<j\leq b} r_{\pi(j)}\}$ and $Y''(\pi)=\{(k,b): q_{\pi(k)}=\min_{a\leq j<b} q_{\pi(j)}\}$. The neighborhood $N(Y(\pi),\pi)$ contains two permutations selected among these from $N(X(\pi),\pi)$.

Now we will give the detailed characteristics (redundancy, relations to others, size, connectivity) of the proposed neighborhoods.

Property 3. For any permutation $\pi \in \Pi$ we have
$N(Y(\pi,\varepsilon),\pi) \subseteq N(X(\pi,\varepsilon),\pi) \subseteq N(Z(\pi,\varepsilon),\pi) \subseteq N(V \backslash (W(\pi) \cup W(\pi) \cup \overline{W}(\pi)),\pi) \subseteq N(V,\pi)$. □

Property 4. The neighborhood $N(Z(\pi,\varepsilon),\pi)$ satisfies connectivity property. □

The proof of Property 4 can be done by analogy to that constructed for neighborhood NB of Dell'Amico and Trubian [6]. Note that the connectivity property is desired but is not obligatory. There are known methods based on neighborhoods without connectivity which numerically behave excellent [19].

4.2. Tabu list

The tabu list is a mechanism of preventing cycling during the search. Among variety of attributes stored in tabu list ([7]) we select, upon previous investigations [20], a pair of jobs as the most promising.

Let $T=(T_1,...,T_{maxt})$ be the tabu list of a fixed length maxt, where $T_j=(g,h)$ is a pair of jobs. The tabu list is initiated by zero elements $T_j=(0,0)$, $j=1,...,maxt$. Let $v=(a,b)$ be a move performed from π. This move receives status tabu and will be add to T (denotation $T \oplus v$) in the following way. we shift to the left list setting $T_j:=T_{j+1}$, $j=1,...,maxt-1$ and add at the end $T_{maxt}:=(\pi(a),\pi(a+1))$ if $a<b$ and $T_{maxt}:=(\pi(a-1),\pi(a))$ otherwise. We assume that a move $v=(a,b)$ from the permutation β cannot be performed (has status tabu) if: (i) $a<b$ and at least one pair $(\beta(j),\beta(a))$, $j=a+1,...,b$ is in T, (ii) $a>b$ and at least one pair $(\beta(a),\beta(j))$, $j=b,...,a-1$ is in T.

4.3. Searching strategy and aspiration function

Similarly as in [19] moves are classified into three categories: unforbidden (UF), forbidden but profitable (FP), forbidden and nonprofitable (FN). A forbidden move v from π is profitable if leads to the makespan less than the value $A(C_{max}(\pi_v))$, where A is so called aspiration function, [7]. Among many types of aspirations we choose the basic form defined in current iteration t by the function $A(x)=\min_{1\leq i<t} C_{max}(\pi^i)$, where π^i is the permutation which generates the neighborhood in the i-th iteration of the algorithm, $i=1,2,...,t$. The move to be performed is selected among UF- and FP- moves.

Beside conventional searching strategy, we employ also a specific strategy (called SFU), fixed upon experimental investigations, and based on the step-wise selection of moves from the move set $X(\pi)$. At the first the move set $Y(\pi)$ is tested, whether it contains an UF-move. If so then we select the move $v'\in Y(\pi)$ such that $C_{max}(\pi_{v'})=\min_{v\in Y(\pi)} C_{max}(\pi_v)$. This can be done in O(n) time. If $Y(\pi)$ do not contain any UF-moves then we check the move set $X(\pi)\setminus Y(\pi)$ and select the first UF-move. This also can be done in O(n) time. If none UF-move has been selected (very rare situation) then the move set $X(\pi)$ is scanning for possible existence FP-moves, among which the best (with minimal $C_{max}(\pi_v)$) is chosen. If all moves in $X(\pi)$ are FN we add to tabu the zero element (0,0), until a UF-move can be chosen.

5. Experimental analysis

The experimental analysis of the algorithm TS with relation to HS were carried out. Test instances were generated n the way described by Carlier [4]. For each n=50,100,150,.. ..,1000 and F=16,17,18,...,25 a sample of 20 instances were obtained; values r_j,p_j,q_j were chosen with uniform distribution between 1 and r_{max},p_{max},q_{max}, respectively. We set $p_{max}=50$, $r_{max}=q_{max}=nF$. Thus, 4000 instances were tested. The instances with F=18,19,20 were reported by Carlier as the hardest one. At the beginning, some tests were performed in order to select the best configuration of the tabu search method. Finally, Algorithm TS were run with maxt=8, maxiter=n/2 (the limit of the total number of iterations), neighborhood $N(Z(\pi),\pi)$, aspiration function A, searching strategy SFU, π^S as the starting solution. For all tested instances (100%) TS provides the best result ($C^{TS}\leq C^{HS}$), whereas only for 0.2% of cases provides strictly better result ($C^{TS}<C^{HS}$). Note that in the tested configuration TS has $O(n^2)$, whereas HS has original $O(n^2 \log n)$ computational complexity. It means that for instances with n=50,...,1000 TS runs 5...10 times faster than HS. The real speed up is higher due to the special technique of implementation. Algorithm TS essentially improves π^S in 96.5% of cases.

BIBLIOGRAPHY

1. Adams J., Balas E., Zawack D.: The shifting bottleneck procedure for job shop scheduling problem. Management Science 34(1988), 391-401.
2. Baker K.R., Su Z.S.: Sequencing with due dates and early start times to minimize tardiness. Naval Research Logistic Quarterly 21(1974), 171-176.
3. Bratley P., Florian M., Robillard P.: On sequencing with earliest starts and due dates with application to computing bounds for (n/m/G/F) problem. Naval Research Logistic Quarterly 20(1973), 57-67.
4. Carlier J.: The one-machine sequencing problem. European Journal of Operational Research 11 (1982), 42-47.
5. Carlier J., Pinson E.: An algorithm for solving the job-shop problem. Management Science 35(1989), 164-176.
6. Dell'Amico, M., Trubian, M.: Applying Tabu-Search to the Job-Shop Scheduling Problem. 1992 (to appear in Annals of Operations Research).
7. Glover, F.: Tabu Search. Part I. ORSA Journal of Computing 1 (1989) 190-206.
8. Glover, F.: Tabu Search. Part II. ORSA Journal of Computing 2 (1990) 4-32.
9. Grabowski J., Nowicki E., Zdrzalka S.: A block approach for single-machine scheduling with release dates and due dates. European Journal of Operational Research 26 (1986), 278-285.
10. Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 5(1979), 287-326.
11. Hall L.A., Shmoys D.B.: Jackson's rule for single-machine scheduling: Making a good heuristic better. Mathematics of Operational Research, to appear.
12. Lageweg B.J., Lenstra J.K., Rinnooy Kan A.H.G.: Minimizing maximum lateness on one machine: computational experience and some applications. Statistica Neerlandica 30(1976), 25-41.
13. Lenstra J.K.: Sequencing by enumerative methods. Mathematical Centre Tracts 69, Amsterdam, 1977.
14. Lenstra J.K., Rinnooy Kan A.H.G., Brucker P.: Complexity of machine scheduling problems. Annals of Discrete Mathematics 1 (1977), 343-362.
15. Kise H., Ibaraki T., Mine H.: Performance analysis of six approximation algorithms for the one-machine maximum lateness scheduling problem with ready times. Journal of the Operations Research Society of Japan 22(1979), 205-244.

16. MacMahon G.B., Florian M.: On scheduling with ready times and due dates to minimize maximum lateness. Operation Research 23 (1975),475-482.
17. Nowicki E., Smutnicki C.: On lower bounds on the maximum lateness on one machine subject to release date. Operation Research 24(1987) 106-110.
18. Nowicki E., Smutnicki S.: An approximation algorithm for the single-machine scheduling problem with release times and deliver times. Discrete Applied Mathematics 48 (1984) 69-79.
19. Nowicki, E. and Smutnicki, C.: A fast tabu search algorithm for the job shop problem. Report 8/93, Institute of Engineering Cybernetics, Technical University of Wroclaw, 1993.
20. Nowicki, E. and Smutnicki, C.: A fast tabu search algorithm for the flow shop problem. Report 13/94, Institute of Engineering Cybernetics, Technical University of Wroclaw, 1994.
21. Nowicki E., Zdrzalka S.: A note on minimizing maximum lateness in a one-machine sequencing problem with release dates. European Journal of Operational Research 23 (1986), 266-267.
22. Potts C.N.: Analysis of a heuristic for one-machine sequencing with release dates and delivery times. Operations Research 28 (1980), 1436-1441.
23. Schrage L.: Obtaining optimal solution to resource constrained network scheduling problem. unpublished manuscript, 1971.
24. Taillard, E.: Some efficient heuristic methods for flow shop sequencing. European Journal of Operational Research 47 (1990) 65-74.
25. Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research 64 (1993) 278- 285.

Recenzent: Prof.dr hab.inż. Jan Węglarz

Wpłynęło do Redakcji do 30.04.1994 r.

Streszczenie

Rozważany jest problem szeregowania zadań na jednej maszynie z terminami gotowości, terminami dostarczenia oraz kryterium minimalizacji maksymalnego terminu zakończenia wykonywania zadań. Zaproponowano nowy algorytm aproksymacyjny dla tego problemu oparty na technice poszukiwania tabu. Opisano szczegółowo podstawowe elementy metody: ruch, sąsiedztwo, listę tabu, funkcję aspiracji. Pokazano także pewne własności zaproponowanych sąsiedztw. W wyniku zastosowania zredukowanego sąsiedztwa opartego na pojęciu ścieżki krytycznej i tzw. eliminacyjnych własnościach bloku zadań zaproponowany algorytm działa znacznie szybciej od innych algorytmów stosowanych do tej pory do rozwiązywania postawionego problemu. Przedstawiono także wyniki analizy eksperymentalnej alorytmu na przykładach losowych do 1,000 zadań.