Approximation methods for total completion time with set-up times

Adawiyah A. Mahmood

University of Diyala
College of Science

Abstract

This research considers the problem of scheduling jobs on a single machine to minimize the objective function , the sum of completion time .The jobs partitioned into families , and a set-up time is necessary for scheduling the first job and when there is a switch in processing jobs from one family to jobs of another family . To solve this problem some known approximation methods are modified , namely the tree type heuristic (TTH) and tow local search methods descend method (DM) and simulated annealing method (SAM) . The performance of approximation methods can be tested on a large class of test problems.

Keywords: Scheduling , single machine , set-up time ,heuristics

الملخص: تناولنا في هذا البحث مسألة جدولة النتاجات على ماكنة واحدة لتصغير دالة الهدف وهي مجموع أوقات الإتمام(The sum of completion time). لقد قسمت النتاجات إلى F من العوائل وهناك وقت إعداد ضروري للماكنة عند جدولة أول نتاج وعند جدولة نتاج من عائلة تختلف عن عائلة النتاج الذي يسبقه . لحل هذه المسألة قمنا بتطوير بعض الطرائق التقريبية (Approximation methods) المعروفة وهي طريقة (TTHM) (Tree type heuristic method) وطريقتين من البحث المحلي (Local Search) وهما (DM) (Descent method) و (SAM) (Simulated annealing method). يمكن تطبيق هذه الطرائق التقريبية على عدد كبير من مسائل الاختبار .

1. Introduction

Approximation methods are an improvement techniques which seek good solutions at a reasonable computational cost [7] . The tree type heuristic method is used to find approximate solutions , by using a branch and bound method without backtracking .Local search methods work as starting with some feasible initial solution , a neighbour (i.e. a feasible solution) in some predefined neighbourhood is generated and then the objective function value of this generated neighbour is compared with that of the starting solution. By means of some acceptance criterion , it is decided which of both feasible solutions is selected to be the starting solution for the next neighbour generation [1].

2. Problem formulation

The scheduling groups of jobs on a single machine problem can be described as follows :

We are given N jobs that are divided into F families . Each family f ,for $1\leq f \leq F$,contains n_f jobs . Sometimes it is more convenient to refer to job (i,f) , which is the ith job in family f , for $1\leq i \leq n_f$. All jobs are available for processing at time zero, and are to be scheduled on a single machine . We let P_{if} denote the processing time of job (i,f) . A machine set –up time S_f is incurred whenever a job in family f is processed immediately after a job in a different family .Also , a set –up time S_f is required for processing the first job in the schedule.

Given a processing order of the jobs , completion time C_{if} of job(i,f) can be computed .Our object is to find a sequence that minimizes the objective function , the sum of completion time

($\sum_i C_i$) .

3. Tree type heuristic method (TTHM)

Although a branch and bound (BAB) method guarantees the finding of an optimal [5], a near optimal solution may result if some of the possibly optimum partial schedules have not been explored . This fact has been used to obtain near optimal solutions for many scheduling problems.

We apply a shortest processing time (SPT) rule , in which the jobs are sequenced in non-decreasing order of processing time at the top of the search tree to provide an upper bound (UB).We compute the lower bound (LB) for all the nodes by relaxation of constraints , usually , one node is chosen within each level of the tree .The method that we choose one node only to branch from within each level of the tree , according to
$H_{LB} = \min_{i} \{ LB(i) \}$, where LB(i) is the lower bound computed at every node i .

A newest active node search is then used to select a node from which to branch , say node j is selected and the LB at all the remaining nodes immediately below node j evaluated and so on by using a forward branching without backtracking .

The tree type heuristic (TTH) continues in a similar way whenever a complete sequence is obtained, this sequence is evaluated and the upper bound (UB) is altered if the new value IUB is less than the old one (i.e. if IUB < UB , then set UB = IUB).

4. Structure of neighbourhood search

A sequencing problem can be defined as a problem with a well defined solution space S i.e. the set of all feasible solutions of the problem , an objective function f(s) to evaluate an element s of S in order to find the element s* that minimize the problem such that :

$$f(s^*) = \min_{s \in S}\{f(s)\}$$

Neighbourhood search is an intuitive solution approach to this minimization problem . This approach usually starts with a known feasible solution , and it tries to improve this solution by making small changes to it. A solution that we obtain after making such a change to a solution s is called a neighbour of s.
The neighbourhood N(s) , a subset of S , is the set of all neighbours of s.

During the iterative process , one "moves" through the solution space S from neighbour to neighbour .A move is evaluated by comparing the objective function value of the current solution to that of its neighbour . If the former is larger , then we refer to the neighbor as an improving move ; if the latter is larger , it is a deteriorating move , if both are the same, then it is a neutral move. For sequencing problems , the "natural" representation of a solution is a permutation of the integers 1,2,…..,n with n the number of jobs .On this representation four basic neighbourhoods can be defined (Anderson et al.[2]).

Each is illustrated by considering a typical neighbour of the sequence (1,2,3,4,5,6,7)in a problem where there are seven jobs labeled1,2,…….,7.

1.Swap two jobs which is not adjacent .Thus , (1,2,6,4,5,3,7)is a neighbour of the solution.

2.Swap two adjacent jobs .Thus ,(1,3,2,4,5,6,7)is a neighbour of the solution. This type of swap neighbourhood called transpose .

3.Remove a job from one position in the sequence and insert it at another position (either before or after the original position).Thus, (1,4,2,3,5,6,7) and (1, 2, 3, 5,6,4,7)are both neighbours.

4.Move a subsequence of jobs from one position in the sequence and insert it at another position . Thus ,(1,4,5,2,3,6,7)is a neighbour. This type of insert neighbourhood called block insert.

The structure of neighbourhood search algorithm is as the following ([5],[6]):

- Initialization. Choose an initial feasible solution s to be the current solution and compute its objective function value f (s).
- Neighbour generation . Select a (feasible) neighbour $\bar{s}$ of the current solution s and compute its objective function value f $(\bar{s})$.
- Acceptance test. Test whether to accept the move from s to $\bar{s}$. If the move is accepted then $\bar{s}$ replaces s as the current solution ;otherwise ,s is retained as the current solution.
- Termination test. Test whether the algorithm`should terminate .If it terminates , output the best solution generated ;otherwise , return to the neighbour generation step.

5. Descent method(DM)

In a descent method (DM),only improving moves are allowed. A potential move is rejected if it is deteriorating or neutral . This method is a

-

simplest neighbourhood search algorithms that starts with an initial solution s and then continually searches its neighbourhood s to be a solution of better quality [8].More precisely , starting with an initial solution (perhaps chosen at random) ,a neighbour $\bar{s}$ in a specific neighbourhood is generated .If $\bar{s}$ has a better objective function value than s, the move from s to $\bar{s}$ is accepted and $\bar{s}$ is chosen as new initial solution. Then the search is continued until no neighbour leads to an objective function value improvement. Although (DM) is simple and quick to execute , but the drawback of it, is that the local minimum found ,it is not necessarily a global minimum .One way of improvement the solution is to run the descent method several times starting from different initial solutions , and take the best sequence as final solution.

To illustrate the above method , we present the following example :
Exmeple:
The problem of 4 jobs with the following data :-

i	1	2	3	4
pi	5	3	6	8

Assume the 4 jobs are divided into two families f1={ 1,2 } and f2={ 3,4 }, and with set-up times are S1=2 and S2=3 for the families f1 and f2 respectivily step (1) : Using the shortest processing time (SPT) rule in which the jobs are sequenced in non- decreasing order of Pi to obtain an initial current solution :

i	2	1	3	4
pi	3	5	6	8
Ci	5	10	19	27

And the objective function value is $\Sigma Ci = 61$.

Step (2) : Two jobs 2 and 3 which is not adjacent are sequenced in position 3 and 2 to obtain a neighbour (3,1,2,4) , for this neighbour we compute Σ Ci = 74 .
Step (3): For the last neighbour the improvement is not made , then the move from (2,1,3,4) to (3,1,2,4) is rejected . The procedure is then repeated from the beginning by repeating step (2) .
Step (4) : The search is continued until no neighbour leads to an objective function value improvement.

6. Simulated annealing method (SAM)

Simulated annealing (SA)is a method for obtaining good solutions to difficult optimization problems which has received much attention over the last few years [4].

The major difference between (SA) and (DM)is in the acceptance rule .In a SA method , improving and neutral moves are always accepted , while deteriorating moves are accepted according to a given probabilistic acceptance function .

So , to allow the search to continue from a local optimum , moves that increase the objective function value are accepted with a probability which becomes smaller when the size of this increase is greater . More precisely , let s be the initial current solution and s be the generated neighbour. Then the difference $\Delta = f(\bar{s}) - f(s)$
in the value of the objective function f is calculated .
When $\Delta \leq 0$, solution $\bar{s}$ is accepted as new starting solution for the next iteration. Otherwise , i.e. if $\Delta > 0$, the acceptance is made by the probabilistic function exp ($-\Delta / T$) , the parameter T is known as the temperature because it refers to the cooling in the physical process , where its value is relatively high in the initial stages of the search so that escaping from a local optimum is rather easy .
Then T gradually decreases until it will close to zero in the final stages . A review of this procedure is provided by Bank [3]and Glass[6].

7. Conclusions

This paper considers the problem of scheduling jobs on a single machine to minimize the objective function , the sum of completion time . Jobs are divided into families and a set – up time is required between consecutively processed jobs of different families . We have designed two local search methods DM and SAM for obtaining near optimal solutions without too much computational effort . Also TTHM generates a very good quality solutions but are computationally time consuming.

An interesting future research topic would involve experimentation with the machine scheduling problem

$$1 \mid S_f \mid \sum_{i=1}^{n} C_i^2 \quad .$$

References

[1] Anderson E .J., Glass C .A . and Potts C.N. , " Applications of local search in machine scheduling",March , (1995).
[2] Anderson E .J., Glass C.A.and Potts C.N. ,"Local search in combinatorial optimization",edited by E.H.L. Aarts and J.K. Lenstra , Wiley (1997).
[3] Bank J.and Werner F. ," Heuristic algorithms for unrelated parallel machine scheduling with a common due date , release dates and linear earliness and tardiness penalties", Working paper (2000).
[4] Eglese R.W.," Simulated annealing :A tool for operational research ", 46,271-281, (1990).
[5] Glass C.A. ,Potts C.N.and ShadeP. ,"Unrelated parallel machine scheduling using local search ", 20,2,41-52 ,(1994).
[6] Glass C.A. and Potts C.N. , "A comparison of local search methods for flow shop scheduling", 63,489-509,(1996).
[7] Reevs C.R. , "Modern heuristic techniques for combinatorial problems ", John Wiley and sons.Inc ., New York (1993).
[8] Vaessens R.J.M., Aarts E.H.L . and Lenstra J.K. , " A local search template", Computers Ops . Res ., 25 ,11,969-979, (1998).