MODIFIED PIPELINING HYBRITIZATION OF JOB SHOP SCHEDULING

P.Paul Pandian, Sethu Institute of Technology, India
S.Saravana Sankar, Arulmigu Kalasalingam College of Engineering
P.Ven kumar , Kalasalingam University, India

Abstract

Hybridization involves generally genetic algorithm in a stage .Here instead of genetic algorithm, metaheuristics method Local search method, is applied as primary search routine, for tackling combinatorial search and optimization problems.The dispatching rule LPT is applied first, serving as a preprocessor. The local search methods are works on the iterative exploration of a solution space: at each iteration a local search algorithm start search from one solution to one of its neighbor. The method is analysis the job shop bench mark problems. The comparison of the performance measure is evaluated.

Keywords

Dispatching rule,LPT,local search, makespan, jobshop

Introduction

Scheduling problem can be defined as the task of associating one or several resources to activities over a certain time period. These problems are of particular interest both in the research community and in the industrial environment. They commonly arise in business operations, especiously in the areas of supply chain management, air flight crew scheduling and scheduling for manufacturing and assembling.

More over scheduling problems arise also in other organization such as schools (as to home as Calvin suggested) universities and hospitals.Generlly speaking, scheduling problems belong to the class of combinatorial optimization problems.

Makespan: The makespan, defined as max ($C1……C_n$), is equivalent to the completion time of the last job to leave the system. A minimum makespan usually implies a high utilization of the machines. It is denoted by $C_{max.}$

Dispatching rules

The Longest processing time first (LPT) rule assign longest processing jobs to machines and the shortest job is the last job to start its processing and also the last job to finish its processing. This rule used to place the shorter jobs toward the end of the schedule, where they can be used for balancing the loads. LPT rule is used for job shop problem and output machine sequence is used as population of local search.

Local search

Optimization techniques can be classified into two categories (1) local search (2) global search methods. A local method uses local information about the current set of data (state) to determine a promising direction for moving some of the dataset, which is used to form the next set of data.[1].The advantages of local search techniques is that they are simple and computationally efficient. The Local Search (by Stephan Kreipl) works only for Ordinary environment. It has a feature of stopping on demand. In other words, the user can choose the time period the algorithm is allowed to run, or even stop the algorithm at any moment and load the best schedule found so far.

Simulated annealing:

The one of local search method is simulated annealing .This neighborhood search method produced good results for combinatorial optimization problems. After introduced this method by Kirkpatrick et.al (1983) and by Cerny (1985), is being used to optimize traveling salesman problem, complex scheduling problem etc.Simulated annealing initially produce a solution randomly. Every stage current solution is chosen by comparing the neighborhood solution, if it is equal or less.cost.A new solution with a higher cost is accepted with a probability that decreases as the difference in the costs increases and as the temperature of the method decreases. Temperature is reduced periodically by a scheme; so that it reduced to zero as the problem continuses.As the temperature reaching zero the method gets a local optimum. This is because simulated annealing has performed many perturbations at higher temperatures which have pushed the search path into new areas, and so a better local optimum solution will be reached. Several authors Matsuo et al [21], van laarhoven et al, [27] DellAmico and Trubian [8] and Nowicki and Smutnicki [23] observed that the choice of a good initial solution is an important aspect of algorithms performance in terms of solution quality and computational time.

The design of neighborhood is a very important aspect of a local search procedure. For a single machine, a neighborhood of a particular schedule may be simply defined as all schedules that can be obtained by doing a single adjacent pair wise interchange. This implies that there are n-1 schedules in the neighborhood of the original schedule. A larger neighborhood for a single machine schedule may be defined by taking an arbitrary job in the schedule and inserting it in another position in the schedule , each job can be inserted in n-1 other positions. The entire neighborhood of a schedule in a more complicated machine environment is usually more complex.

An interesting example is a neighborhood designed for the job shop problem with the makespan as objective. A critical path in a job shop schedule is used to describe the neighborhood. At t=0 the process starts and finishes at t=$C_{max.}$ The completion time of each operation on a critical path is equal to the starting time of the next operation on that path. optimize the makes pan, sequences of the operations must be changed on the critical path. A simple neighborhood is as follows: The set of schedules whose corresponding sequences of operations on the machines can be obtained interchanging a pair of adjacent operations on the critical path of the current schedule .Note that, to interchange a pair of operations on the critical path, the operations must be on the same machine and belong to different jobs. If there is a critical path, then the number of neighbors within the neighborhood is at most the number of operations on the critical path, then the number of neighbors is the number of iterations on the critical path - 1.

Dorndorf and Pesch (1995) method that a framework should be constructed which navigates these local decisions through the search domain in order to determine a high quality global solution in a reasonable amount of time. The meta-heuristics or iterated local search algorithms having the framework local decisions made by myopic problem specific heuristics are guided beyond local optimality by an underlying metastrategy.

There are number of ways the neighborhood search can be done. To select schedules in the neighborhood at random is a simple method; the schedules are evaluated and decide which one to accept. The acceptance _rejection criterion is usually the design aspect that distinguishes a local search procedure the most. In simulated annealing, the acceptance –rejection criterion is based on the probalistic process, where as in tabu search it is based on a deterministic process.

Simulated annealing is a search process that has its origin in the fields of material science and physics. It was first developed as a simulation model for describing the annealing process of condensed matter. The simulated annealing procedure goes through a number of iterations. At iterations k of the procedure, there is a current schedule S_k as well as a best schedule found so far, S_o .For a single machine problem, these schedules are sequences of the jobs. Let D (S_k) and D (So) denote the corresponding values of the objective function. Note that D (S_k) ≥ D (So). D (So) is called aspiration ratio because the value is best schedule so far. .The algorithm, in its research for an optimal schedule, moves from one schedule to another. At iteration k, a search for a new schedule is conducted within the neighborhood of S_k.Candidate schedule, S_d is selected from the neighborhood, initially. This done by in an organized possible sequential or random way. If D(S_k) < D(So),a move is done, let S_{k+1}= S_k.If D(S_d) < D(So),then S_o= S_k.However, if that D(S_d) ≥ D(So),a move is made to S_d with probability

$$P (S_{k,} S_d) = \exp((D (S_k) - D (S_d))/\beta_k$$

With probably 1-P($S_{k,}S_d$) schedule S_d is rejected in favor of the current schedule, setting S_{k+1}= S_k.Schedule So does not change when it is better than schedule S_d.The controlling parameters are to be $\beta_1 \geq \beta_2 \geq \beta_3 \geq \ldots > 0$ referred to as cooling parameters. The parameter β_k is set to be c^k for some c between 0 and 1.The move to worse solution is possible in above description. The reason for allowing these moves is to give the procedure the opportunity to move away from a local minimum and after find a better solution. Because

β_k reduces with k, , in later iterations the acceptance probably for a non improving move is lower in search process.

Simulated Annealing Algorithm:

Step 1. Set k=1 and select β_1.

Select an initial sequence S_1 using some heuristic.

Set $S_o = S_1$

Step 2. Select a candidate schedule S_d from the neighborhood of S_k.

If D (So) < D (S_d) < D (S_k), then $S_{k+1} = S_k$ and go step 3.

If D(S_d) < D(S_o),set $S_o = S_{k+1} = S_d$ and go to step 3.

If D(S_d) < D(S_k),generate a random number U_k from a uniform distribution (0.1).

If $U_k \leq P(S_k, S_c)$, set S_{k+1} =Sc ; otherwise set $S_{k+1} = S_k$ and goto step3.

Step 3. Select $\beta_{k+1} \leq \beta_k$.

Increment k by 1.

If k =N then stop, otherwise go to step 2.

Modified Pipelining hybrids

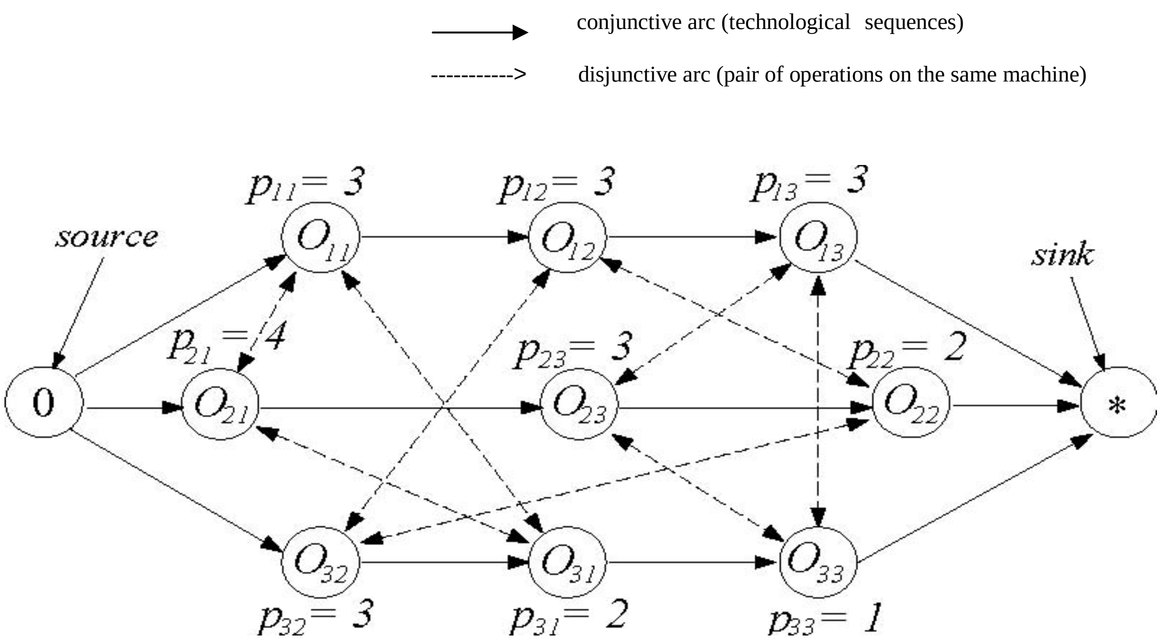
Pipeline hybrids are the most commonly used and simplest method, in which the genetic algorithm and some other optimization techniques are applied sequentially- one, generates data used by the other. Modified pipeline hybrids use local search method instead of genetic algorithm and LPT dispatching rule. Here local search is applied last like the second type of pipeline hybridization (primary search routine).

Problem Description

The $n \times m$ *minimum-makespan* general job-shop scheduling problem, known as the JSSP, can be described by a set of *n* jobs $\{J_j\}_{1 \leq j \leq n}$ which is to be processed on a set of *m* machines $\{M_r\}_{1 \leq r \leq m}$. Each job has a technological sequence of machines to be processed. The processing of job J_j on machine M_r is called the *operation* O_{jr}. Operation O_{jr} requires the exclusive use of M_r for an uninterrupted duration p_{jr}, its processing time. A *schedule* is a set of completion times for each operation $\{c_{jr}\}_{1\leq j\leq n, 1\leq r\leq m}$ that satisfies those constraints. The time required to complete all the jobs is called the *makespan Cmax*. The objective when solving or optimizing this general problem is to determine the schedule which minimizes *Cmax*. The JSSP is not only *NP*-hard, but it is one of the worst members in the class. An indication of this is given by the fact that one 10 × 10 problem formulated by the researchers Muth and Thompson remained unsolved for over 20 years. The JSSP can be formally described by a disjunctive graph $G = (V, C\cup D)$,where

- *V* is a set of nodes representing operations of the jobs together with two special nodes, a *source* (0) and a *sink* *, representing the beginning and end of the schedule, respectively.
- *C* is a set of conjunctive arcs representing technological sequences of the operations.
- *D* is a set of disjunctive arcs representing pairs of operations that must be performed on the same machines.

The processing time for each operation is the weighted value attached to the corresponding nodes. The following figure shows this in a graph representation for a typical 3x3 problem,



O_{ij} : an operation of job *i* on machine *j pij* : processing time of *Oij*

Figure :1 A disjunctive graph of a 3 × 3 problem

Job-shop scheduling can also be viewed as defining the ordering between all operations that must be processed on the same machine, i.e. to fix precedence between these operations. In the disjunctive graph model, this is done by turning all undirected (disjunctive) arcs into directed ones. A *selection* is a set of directed arcs selected from disjunctive arcs. By definition, a selection is *complete* if all the disjunctions are selected. It is *consistent* if the resulting directed graph is acyclic.

A schedule uniquely obtained from a consistent complete selection by sequencing operations as early as possible is called a *semi-active* schedule. In a semi-active schedule, no operation can be started earlier without altering the machining sequences. A consistent complete selection and the corresponding semi-active schedule can be represented by the same symbol *S* without confusion. The makespan *L* is given by the length of the longest weighted path from source to sink in this graph. This path *P* is called a *critical path* and is composed of a sequence of *critical operations*. A sequence of consecutive critical operations on the same machine is called a *critical block*.

The problem is to find the schedule that minimize the makespan the following constraints are subjected(a) the precedence of operations given by each job are to be respected, (b) each machine can perform at most one operation at a time and(c) the operations cannot be interrupted,(d) the operations cannot be repeated in the same machine(preemptive).

Let:

J= {1… n} mention the set of jobs;

M= {1… m} mention the set of machines;

V = {0, 1 …n+1} mention the set of operations, where 0, n=1 denote the start and end dummy operations, respectively.

A be the set of pair of operations constrained by the precedence relations as in (a);

V_k be the set of operations performed by the machine k;

P_v and t_v denote the processing time and (variable) start time of the operation v, respectively. The processing time of the 0 and n+1 operations is equal to zero, i.e $p_o=p_{n+1}=0$.

The problem can be stated as

Minimize t_{n+1}

Subject to

$t_j-t_i \geq p_i$ (i,j)€A,

$t_j-t_i \geq p_i$ V $t_i- t_j \geq p_j$, (i,j) € $E_k k$, € M,

$t_i \geq 0$, i €V

The first set of constraints represents the precedence relations among the operations of the same job , whereas the second set of constraints describes the sequencing of the operations on the same machine. These constraints impose that either $t_j-t_i \geq p_i$ or $t_i- t_j \geq p_j$

Any feasible solution of the problem (a) is called schedule.

Computational results

The bench mark problems are downloaded from the website of Prof Eric Taillard and OR Library.. The algorithms have been coded in C language and experiments have been run AMD sempron 600 MHz PC equipped with 35 GB of memory, running windows XP professional..

The method is tested on Taillard bench mark problems Adams, Balas , and Zawack 10×10 .These are large size problems in job shop environment. The experiment was carried out on instances namelyTa1, Ta2, Ta3, Ta4, Ta5, Ta6 Ta7, Ta8 Ta9, Ta10,of size15×15. abz5 and abz6.

The results of various instances taken into account and their average of makespan are found out at same number of iterations 200. They are tabulated in Table 1. In the table.1 'n 'denotes number of jobs; 'm' denotes number of machines, and $\check{C}_{max}$- denotes $_{mean}$ makespan.

Table 1.

Bench mark	Size n×m	LPT	Local Search	Modified Pipeline hybridization
Ta1	15×15	1542	1262	**1256**
Ta2	15×15	1517	1267	**1253**
Ta3	15×15	1428	1242	**1248**
Ta4	15×15	1391	1198	**1198**
Ta5	15×15	1490	1246	**1236**
Ta6	15×15	1421	1265	**1262**
Ta7	15×15	1542	1253	**1251**
Ta8	15×15	1567	1251	**1229**
Ta9	15×15	1547	1308	**1298**
Ta10	15×15	1592	1277	1282*

n-number of jobs;m –number of machines; $\check{C}_{max}$-mean makespan

All makespan values of modified pipeline hybridization are smaller than local search and LPT values except fot Ta4 and Ta10 . Makes pan is equal to local seach in case of Ta 4 .

Comparison

The result of Modified Pipeline hybridization is compared with Kuo-Huang, Ching-Joug Liao, ACOFT-MWR and ACOFT-TR methods (Ant colony optimization combined with taboo search for the job shop scheduling) published in Computers &Operations Research 2006.The comparisons shown in table 2. The makespan of Modified Pipeline hybridization is better than ACOFT-MWR and equal to ACOFT-TR method for instant Ta5 and for abz5 our proposed method give better result ,for abz6 the makespan is equal to ACOFT-MWR and ACOFT-TR.

Figure1 shows Gantt chart of optimum makespan of Modified Pipeline hybridization of abz5. In which m1, m2……..m10 denotes the machine numbers and j001.j002…….j010 denotes the jobs.

Table 2

.Bench mark	Size n×m	ACOFT-MWR	ACOFT-TR	Modified Pipeline hybridization
Ta5	15×15	1237.5	1236.9	**1236**
Abz5	10×10	1235.8	1235.8	**1185**
Abz6	10×10	943	943	**943**

Gantt Chart:

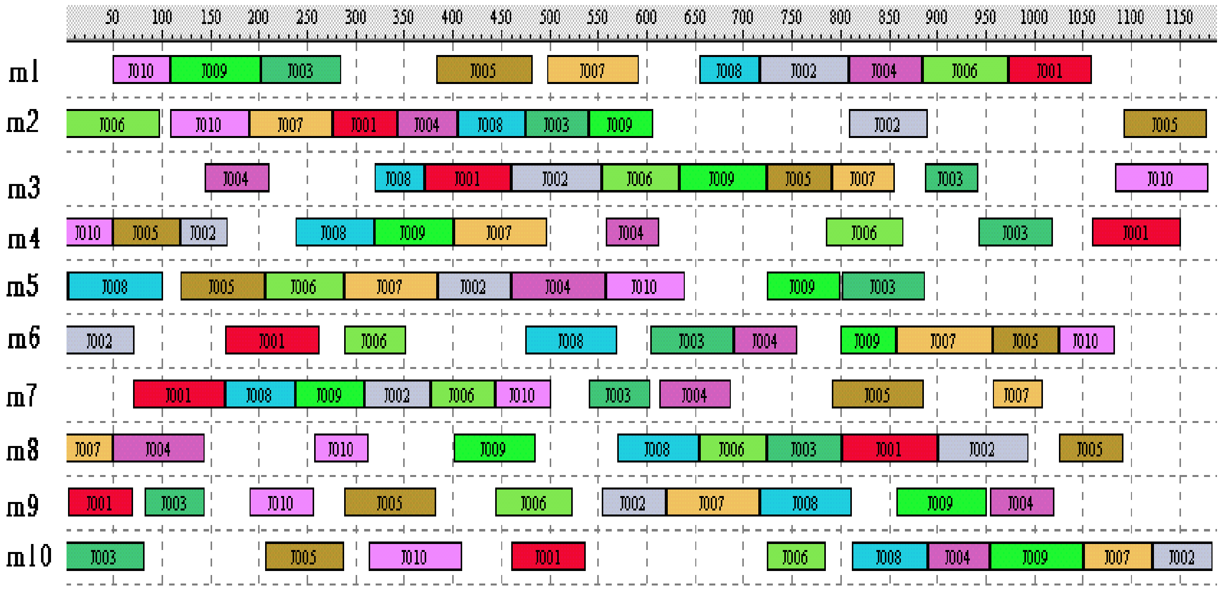


FIGURE 2: A instance of 10×10- makespan 1185

Design and Analysis: By Randomized complete block Design.

The significance of the difference between the mean of different samples tested

By randomized complete block **design**.. A table 3 is formed to analysis the experiments results. For example R.Pannerselvam et al[research methodology2006] In the table ,V_{ij} is the i th cputime of the jth problem size. The problem size is the factor which has effect on the responsible variable V_{ij}. Let it be factor B. b is the number of problem. This is also known as the number of levels/treatments of the factor b. n is the number of data under each problem. This is also known as the number of replications under each level of the factor B.

Fixed factor is used in this problem because a specific set of treatment of a factor is selected with certainty.

The model of the Latin square design $V_{ij}=\mu+Bi+Tj+Pk+e_{ijk}$

Where μ is the overall mean;

V_{ij},is the observation with respect to the j th treatment of the factor and ith block. ;

Tj is the effect of the jth treatment of the factor

Pk is the effect of the k block representing period and

e_{ijk} is the random error associated with the I th block and the jth treatment of the factor..

Table 3: Latin Square Design:

Source of variation	Degrees of freedom	Sum of squares	Mean sum of Squares	F ratio
Between treatments	2	15559830.4	7779915.2	**1.5**
Between blocks	2	15559830.4	7779915.2	**1.5**
Error	6	30914447	51522407.9	

In the table3, the value of the calculated F ratio for the treatment is 1.5.

In all above three cases the calculated value is less than respective table values.

Hence the null hypothesis, Ho is rejected.

Inference: This means that there is no significant difference in terms of makespan between different METHODS ACOFT-MWR, ACOFT-TR, Modified Pipeline hybridization.

Conclusions

With a lot of research work has been carried out on the study of dispatching rule local search in job shop scheduling , there have been relatively few attempts to study the makespan using with pipeline hybridization of dispatching rule local search in job shop scheduling. In this work for each category 12 bench mark problems is iterated. Nearly 120 iterations are carried out. Medium size like 10×10, large size problem like 15jobs and 15 machines problem can be solved using this program. Using LSD the difference in objectives is tested.

References

[1] A.S Jain, S.Meeran (1998)"Theory and Methodology Deterministic job shop scheduling: Past, Present and Future". European journal of operation Research.Vol.113 390-434.

[2] Baker, K.R, Introduction to Sequencing and Scheduling, Wiley, New York, 1974.

[3] Carlos Gomes Da Silva B.C,Jose Figueira A.C.Joao Climaco(2007)"Integerating partial optimization with scatter search for solving bi-criteria {0,1}-knapsack problems",European journal of operations Research ,Vol 177 1656-1677.

[4] E.Taillard, Theory and Methodology, Some Efficient Heuristic methods for the flow shop problem, European journal of operations research,1988.

[5] John F.Muth and Gerald L.Thomson(1963)"Industrial scheduling". Prentice Hall,Inc,Englewood Cliffs,New Jersey.

[6] John Yen and RezaLangari (2003)"Fuzzy Logic Intelligence, Control, and Information". Pearson Education Pte Ltd.

[7] Kuo-Ling Huang,Ching-JongLiao(2006)."Ant colony optimization combined with taboo search for the job shop scheduling problem". Computers and operations research vol 64.88-121.

[8 Luca Di Gaspero(2003)"Local search Techniques for scheduling problems:Algorithms and Software tools" Ph.D Thesis.

[9] Michael Pinedo(2002)"Scheduling Theory, Algorithms, and Systems "Prentice Hall Upper Saddle River, New Jersey.

[10] R.Panneerselvam (2006)." Research methodology", Prentice Hall of India Pvt Ltd