Proceedings of the 1999 IEEE
International Symposium on Assembly and Task Planning
Porto, Portugal - July 1999

META-HEURISTICS FOR THE SINGLE-MACHINE SCHEDULING TOTAL WEIGHTED TARDINESS PROBLEM

Ana Maria Madureira
Instituto Superior de Engenharia do Porto (ISEP/IPP)
Departamento de Engenharia Informática
Rua de São Tomé, 4200 PORTO, Portugal
Phone: +351 - 2 - 8340500; Fax: +351 - 2 - 8321159
Email: anamadur@dei.isep.ipp.pt

Abstract

In this paper, some general features of single-machine scheduling problems are described, and some of their structural properties are used to design local search procedures based on alternative definitions of neighbourhoods.
In particular, the traditional idea of "exchanging the position of two jobs" is replaced by the idea of "exchanging jobs not apart more than a given number of positions (considered as a parameter of the algorithm)".
For generating initial solutions, some traditional priority rules were tested with some degree of randomisation, introducing in general, a positive effect in the performance of the algorithms.
Through a set of computational tests, the importance of the different parameters was evaluated, and their values for different meta-heuristic procedures (Tabu Search, and Randomised Local Search) were tuned. Though these tests have been exhaustive only for a given problem (Weighted Tardiness), the results already available show these approaches are robust and flexible, and that, in general, satisfactory solutions can be obtained in an efficient way.

Keywords: scheduling; local search; meta-heuristics.

1. Introduction

The planning of Manufacturing Systems involves frequently the resolution of scheduling problems with an important impact on the performance of manufacturing organisations.

The basic scheduling problem consists in sequencing a set of jobs for processing on a single resource or machine. We assume a deterministic context where the processing times and due dates are known and independent for the sequence. In this case an ordering of the jobs completely determines a schedule[2][6].

The study of the single-machine scheduling problems is still very important for several reasons, but the most relevant is that a good understanding of this problem provides a support to model the behaviour of a complex system. In this cases it is important to understand the working of the system components, and quite often the single-machine problem appears as an elementary component in a larger scheduling problem[2]. Sometimes the basic single-machine problem is solved independently, and then incorporates the result into the main problem. For example, in a multiple operation problem there is often a critical machine which processing capacity is lower than the necessary (bottleneck). The analysis and treatment of the bottleneck like a single-machine may determine the properties of the entire schedule.

The basic single machine problem is characterised by the following conditions:

- a set of n independent jobs ($j=1,...,n$) is available for processing at time zero and the job descriptors are known in advance;
- a machine is continuously available and is never kept idle while working is waiting;
- the set-up times for the jobs are independents of job sequence and can be included in processing times;
- and, once processing begins on a job, it is processed to completion without interruption.

Under these conditions exists a one-to-one correspondence between a sequence of these n jobs and a permutation of the job indices ($1, 2, ..., n$). The total number of different solutions to the single machine-scheduling problem is $n!$.

It could be considered for the jobs some additional constraints, for example, to consider a problem with release dates (the point of time at which the job becomes available for processing) or deadlines (the instant of time at which the processing job must be completed).

In this paper, the goal consists in evaluating the performance of some local search procedures (Tabu Search, and Randomised Local Search) on the resolution of the Weighted Tardiness problem. As a measure of algorithm performance it will be used the quality of the solution.

2. Weighted Tardiness Problem (WT)

Let us consider the problem of scheduling n jobs on a single machine that can process only one job at a time. For each job j ($j=1,...,n$), let p_j be its processing time, d_j its due date and w_j the penalty incurred for each unit

0-7803-5704-3/99/$10.00©1999 IEEE

of time in late. The processing of the first job begins at the instant of time t=0. The lateness of a job is given by T_j= max $\{t_j+p_j-d_j, 0\}$ where t_j is the start time of job j. The goal is to find a sequence that minimises the sum of *Weighted Tardiness*: **Min $\sum w_j T_j$, with T_j= Max $\{t_j+p_j-d_j, 0\}$**

This problem is NP-complete [13]. An optimal algorithm (enumerative methods) for this problem would require a significant computation time that grows exponentially with the problem size. Hence, only small sized instances can be solved in an efficient way. The WT problem can be solved more efficiently by applying, in a systematic way, a set of dominance rules (see Emmons' dominance rules[1]) that allow us either to fix the position of some jobs, or to settle some precedence relating pairs of jobs. For the general *Weighted Tardiness* problem several branch-and-bound procedures and dynamic programming techniques have been proposed (see [1][2][11]).

Local Search Meta-heuristics are a class of approximation algorithms that are considered to be efficient tools for solving hard combinatorial optimisation problems.

3. Local Search Meta-Heuristics

The planning of Manufacturing Systems involves frequently the resolution of a huge amount and variety of combinatorial optimisation problems with an important impact on the performance of manufacturing organisations. Examples of those problems are the sequencing and scheduling problems in manufacturing management, routing and transportation, layout design and timetabling problems. The classical optimisation methods are not enough for the efficient resolution of those problems or are developed for specific situations. The interest of the Meta-Heuristic approaches (Tabu Search, Simulated Annealing, Genetic Algorithms and Neural Networks) based on Local Search, is that they converge, in general, to satisfactory solutions in an effective and efficient way (computing time and implementation effort).

Local Search methods perform a blind search, since they only accept sequential solutions which produce reduction in the objective function value. Essentially, *Local Search* consists in moving from a solution to another one in the neighbourhood according to some definite rules. The sequence of solutions can be called a *trajectory* in the solution space. They depend heavily on initial solutions and the neighbourhood generation mechanisms. The main weakness of the basic algorithm is its inability to escape from local optimum.

In order to avoid some of the local search disadvantages, some general frameworks were developed creating new hybrids procedures by combining different concepts derived from: classical heuristics, artificial intelligence, learning process, natural selection, species evolution and neural systems. They have been successful in finding optimal or near-optimal solutions to high practical applications of optimisation problems in diverse areas better than classical procedures.

A *Meta-Heuristic* is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space using learning strategies to structure information (initial solution and the neighbourhood generation mechanisms) in order to find efficiently near-optimal solutions.

Tabu Search is a Local Search procedure introduced by Glover and designed for escaping from local optimum. It is based on the general tenets of intelligent problem solving. The process in which the Tabu Search algorithm seeks to transcend local optimality is based on an aggressive evaluation that chooses the best available solution at each iteration even when this solution may result in a degradation of objective function value. The recent moves are penalised [10].

The Simulated Annealing has its origins on the analogy between the annealing process of solids and the problem of solving combinatorial optimisation problems. At the beginning almost all solutions are accepted. Then, gradually, the "temperature" is decreased which means that one becomes more and more selective in accepting new solutions. At the end, only the solutions that improve the objective function value are accepted [10].

Genetic Algorithms were developed by Holland in the 70's and are an attempt to mimic the biological evolution process for discovering good solutions. They are based on a direct analogy to Darwinian natural selection and mutations in biological reproduction (see [4][9][10]). A Genetic Algorithm maintains a population of solutions throughout the search. It initialises the population with a pool of potential solutions to the problem and seeks to produce better solutions by combining the better of the existing ones through the use of genetic operators (selection, crossover and mutation).

The Neural Networks were originally designed for simulating the brain behaviour. The main technical interest of Neural Networks is their ability to process information with a high degree of parallelization. They have been used for dealing with classification problems where their learning capabilities as well as their computing power are determinant features[10]. The conception of Neural Networks designed for optimisation is due to Hopfield and Tank, so the general Neural Nets used in combinatorial optimisation are known as Hopfield nets.

4. Computational Tests

A set of software tools were developed to perform a computational study aiming at analysing and evaluating the performance of some local search heuristics, on resolution of single-machine scheduling problems, with special concern on minimisation of Weighted Tardiness problem [7][8]. As a measure performance it was used the quality of the solution.

4.1 Initial solution generating heuristics

In order to evaluate the impact or a possible influence of the initial solution on the quality of the final solution some computational tests have been performed with Local Search algorithm on instances of the Weighted Tardiness problem with 20 and 30 jobs (WT20A, WT20B, WT20C, WT30A e WT30B) presented in [12].

Due to the random component incorporated in some of those procedures and on generation of sub-neighbourhoods, the results, for each instance of the problems are referred to 12 runs of the Local Search algorithm.

It was necessary to make some other options, and have been defined as neighbourhood generating mechanism the exchanging jobs not apart more than 25% the number of jobs. Considering the significant dimension of the problems, it was decided to work with sub-neighbourhoods constituted by 25% of the solutions randomly obtained in the neighbourhood of the problem. The criteria defined to select the best 75% solutions have as purpose to avoid eventual extreme behaviours of the algorithm, considering the important degree of randomisation.

Table 1 - Influence of the initial solution generating heuristics

Problem		WT20A	WT20B	WT20C	WT30A	WT30B
Optimal value		137	329	425	1911	404
SPT						
best value		171	387	467	1964	534
average value		238.5	468	573.1	2123.2	631
average from best 75% solutions	a)	201.2	429.2	553.1	2072.3	596.8
deviation from a) to the optimal		46.88%	30.46%	30.14%	8.44%	47.72%
EDD						
best value		147	329	451	1944	469
average value		162.3	420.3	532.2	2070.8	516.4
average from best 75% solutions	a)	154.9	385	491.1	2040.1	497.3
deviation from a) to the optimal		13.06%	17.02%	15.56%	6.76%	23.1%
D_j/W_j						
best value		147	405	478	1979	446
average value		282.3	428.7	527.3	2047.9	562.1
average from best 75% solutions	a)	202.2	409.3	500.9	2040.1	552
deviation from a) to the optimal		47.61%	24.42%	17.86%	6.76%	36.63%
RND						
best value		147	411	442	1924	460
average value		262.5	509.75	527.6	2015.8	521.9
average from best 75% solutions	a)	204.7	450.1	498.2	1984.8	499.3
deviation from a) to the optimal		49.39%	36.81%	17.23%	3.86%	23.6%
REDD 4 pairs						
best value		149	366	445	1948	445
average value		195.2	466.8	510.2	2048.9	500.1
average from best 75% solutions	a)	177.1	435.3	496.9	1996.3	482.1
deviation from a) to the optimal		29%	32.3%	16.9%	4.46%	19.3%

In this work, two types of heuristics procedures for generating an initial solution will be considered:

- deterministic (for each priority rule correspond one single solution or sequence);
- randomised (allow the generation of different initial solutions).

With the deterministic procedures we try take advantage from the specific characteristics of the objective function, we studied the following priority rules: SPT (Shortest Processing Time) where the jobs are sequencing in order of non-decreasing processing times; EDD (Earliest Due Date) where the jobs are sequencing in order of non-decreasing due dates and the dj/wj ratio where the jobs are sequencing in order of non-decreasing dj/wj ratio.

In the randomised procedures it will be considered a rule called by RND (Random) where the order of the jobs on the initial solution is defined randomly, and a random version from the priority rule EDD, which is called by REDD (Randomised Earliest Due Date). The idea of introducing some degree of randomisation on a EDD solution arises from the trial to simulate a intermediary situation between a "good" solution generating in a deterministic way and a random

solution, trying take advantages from EDD solution and simultaneously avoid local optimal. The random component of the REDD rule is introduced by exchange pairs of jobs randomly chosen.

From the general analysis results it was verified that the initial solutions does not induce significantly the quality of the final solution. It is not clear that if we use a better initial solution, whose value is more near optimal, we obtain a better solution than if we start from a worst initial solution.

After all, and at the impossibility of finding a better job ordering procedure it was tried to identify a rule with a satisfactory performance in most of cases. It was used the minimisation of the deviations from the average value to the global optimum for the best 75% of solutions of each one of the instances as a comparative criteria.

Though it is possible to notice some advantage in the use of the EDD rule in most of the instances, the results are not completely conclusive possibly due to the randomisation degree introduced on the algorithm (see table 1).

The results show that are obtained slowest deviations from average value to the global optimum on the instances WT20A, WT20B, WT20C with a initial solution generated by EDD rule, on the WT30A with RND rule and on WT30B with REDD rule by exchanging 4 pairs of jobs. Some general conclusions about these results are:

- the worst final solutions were obtained with the initial solutions constructed by SPT ordering;
- there is no advantage in the use of d_j/w_j ratio;
- a EDD solution presented some advantage in most instances and in some cases, quality improvements can be obtained in introducing some degree of randomisation (REDD rule).

4.2 Generating neighbourhood mechanisms

Neighbourhood is the set of solutions generated from each one of problem solutions by the application of certain generation mechanism.

For scheduling problems, that involves permutations, several generating mechanisms have been used, such as, the adjacent pairwise interchange [2][3], the general pairwise interchange and the exchange of three jobs.

In this work, to control the compromise efficiency-effectiveness, essential for the good performance of the algorithm, a intermediate situation was considered in the case of the general pairwise interchange, "exchanging jobs not apart more than a given number of positions (considered as a parameter of the algorithm)".

In order to evaluate the influence of neighbourhood generating mechanisms on the quality of the final solution, computational tests (see table 2) have been performed using Adjacent Pairwise Interchange and Pairwise Interchange with a distance (not apart more than 25% of problem size) mechanisms, considering sub-neighbourhoods with same size.

Table 2 - The influence of the generating neighbourhood mechanisms

Problem	WT20A	WT20B	WT20C	WT30A	WT30B
Optimal value	**137**	**329**	**425**	**1911**	**404**
Adjacent Pairwise Interchange					
best value	137	386	465	2113	599
worst value	559	784	1300	2709	1174
average value	232.4	498.5	671.5	2259.05	752.3
deviation from average	69.64%	51.52%	58.00%	18.21%	86.21%
Exchange 2 jobs (apart=25%n)					
best value	147	341	431	1990	452
worst value	297	660	1072	2207	607
average value	178.65	434.8	573.1	2090.85	532.45
deviation from average	30.40%	32.16%	34.85%	9.41%	31.79%

It is possible to verify that the mechanism of Pairwise Interchange (with a maximum distance between each other) seems to be more effective, because it generates a bigger neighbourhood; although, with a sub-neighbourhood with the same size, the probability to escape to the local optimum could be bigger (possibly due to more diversification).

Considering the times involving in evaluating solutions, it is not possible to use the complete neighbourhood from a solution, an alternative way is to use sub-neighbourhoods. The sub-neighbourhood size N' (number of sequences) is defined by a percentage $N'=\%N$, where N is the neighbourhood size from x_n. The N' sequences are randomly chosen from its neighbourhood $N(x_n)$.

Some computational tests have been performed and the conclusion was that sub-neighbourhoods composed by 25% of the neighbourhood solutions present satisfactory results.

4.3 Comparative evaluation of meta-heuristics performance

It was proposed a meta-procedure called Random Local Search (RNDLS) that is a extension of Local Search (LS) procedure and consists in running the original Local Search algorithm many times with random initial solutions. The best overall solution is the result of the process (see [5][8]).

Through a set of computational tests, the importance of the different parameters was evaluated, and their values for meta-heuristic procedures (Tabu Search, and Randomised Local Search) were tuned.

At the beginning the goal was to evaluate the influence of the random component associated to the generation of the initial solution on the algorithm performance. For each one of the instances of the WT problem, the RNDLS algorithm has been executed with initial solutions generated by a random procedure and by REDD rule. There is a significant variability on the quality of the obtained results, that could be due to the strong random component introduced and to some extreme behaviours from the algorithm. Though, the results are not completely conclusive, it is possible to conclude that the use of initial solutions generated by the REDD rule can reduce this effect, allowing the algorithm to converge faster to better solutions.

In order to define the attributes that characterise a tabu move, and at the same time the length of the tabu list, some computational tests have been performed on instances of the WT problem using Tabu Search (TS) method. Due to the remarkable degree of randomisation of the algorithm it has become difficult to take absolute and definitive conclusions about the values of those parameters, considering that the obtained computational results depends from the random component introduced either on generation of the initial solution (the solution is constructed by REDD rule) or on the neighbourhood generation (the neighbourhood elements are randomly chosen). It is possible to conclude that better solutions were generally obtained with a tabu list that stores the pairs of jobs involved on exchanging of positions, and there is some advantage on using tabu lists with length 7 or near to 7 (see [8]).

Finally, the performance of the RNDLS and Tabu Search algorithms has been compared on resolution of the instances above mentioned for the minimisation Weighted Tardiness problem. In order to analyse the obtained results, performance measures were computed: the best value and the deviation from the best value to the optimal. The optimal values from instances of problems with 50 e 60 jobs (that were randomly generated using a software tool described in [8]) are not known, so in these cases only the best value is considered.

The definition of parameters of both algorithms have been made, in order to obtain identical computational efforts to permit the comparison in effectiveness (quality of solutions).

Table 3 - Comparative study of the meta-heuristics performance

Problem	WT20A	WT20B	WT20C	WT30A	WT30B	WT50A	WT60B
Optimal value	**137**	**329**	**425**	**1911**	**404**	**-**	**-**
Random Local Search							
best value	137	342	435	1936	415	218	948
deviation from optimal	0%	3.95%	2.35%	1.3%	2.72%		
Tabu Search							
best value	137	329	425	1911	427	205	873
deviation from optimal	0%	0%	0%	0%	5.69%		

Both algorithms obtained a good performance on resolution of the mentioned instances, however, the Tabu Search procedure has presented better solutions in most of the instances, and in some cases, the Tabu Search found the optimal solution (WT20A, WT20B, WT20C, WT30A, WT50A and WT60B). The Randomised Local Search found a better solution on WT30B instance.

The difference in effectiveness between the TS and RNDLS algorithms are least significant, as we can see on the table 3, the deviations from the best value to the optimal, for both algorithms are inferior to 6%.

However the Randomised Local Search has the following features and advantages when compared with Tabu Search:

- the possibility to obtain solutions with good quality (near to optimal and to the obtained solutions from Tabu Search), more quickly and with a smaller computational effort;
- is easy to implement;
- does not need a great parameter tuning effort.

The set of performed tests suggests the global interest in the construction of meta-algorithms based on original algorithms repetition, allowing a progressive

improvements of the solutions, in most of the cases with least computational effort.

5. Conclusions

More than developing algorithms with unquestionable practice utility, the main purpose of this work was to illustrate on simple scheduling problems, the potentiality of the meta-heuristic approaches. Though the computational results obtained are not completely conclusive, they show that on this problem (where exists natural structures neighbourhoods: adjacent pairwise interchange and exchanging jobs not apart more than a given number of positions), these approaches are robust and flexible, and that, in general, it is possible to obtain satisfactory solutions in an efficient way.

The obtained results show the interest of using algorithms with a strong random component to construct meta-procedures that consists in repeat those algorithms so many times as the computation time permits.

Another positive aspect from this work is that it permits to build a general approach to the designing and evaluation of meta-heuristics in others scheduling problems.

A possible developing of this work could be the handling of another objective functions (that include, for instance, set-up costs and set-up times) or another restrictions (for instance, with job precedence restrictions). In particular, those extensions will permit to construct multi-criteria analysis procedures to integrate possibly on a Decision Support System, where the user interface could play an important role.

Another direction relates to the intelligent optimisation of the dynamic scheduling problem in distributed manufacturing systems. In this perspective, we pretend the development of a manufacturing system architecture that combine the strengths of local search meta-heuristics (Genetic Algorithms, for example) with Intelligent Manufacturing Systems and Artificial Intelligence concepts.

References

[1] Abdul-Razaq, T.S., C.N. Potts and L.N. Van Wassenhove, *A survey for the Single-Machine Scheduling Total Weighted Tardiness Scheduling Problem*, Discrete Applied Mathematics, n°26, 235-253, 1990.

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

[3] Croce, F. D., *Generalised Pairwise Interchanges and Machine Scheduling*, European Journal of Operational Researchs, Vol 83, 310-319, 1995.

[4] Davis, Lawrence, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[5] Feo, T.A., M.G. Resende, *A Greedy Randomised Adaptative Search Procedure for Maximum Independent Set*, first draft, 1989.

[6] French, S., *Sequencing and Scheduling: An introduction to the Mathematics of the Job-Shop*, Ellis Horwood, Chichester, 1982.

[7] Madureira, Ana M., *"Aplicações de Meta-Heurísticas em Problemas de Sequenciamento"*, Msc Thesis, Faculty of Engineering, University of Porto, 1995 (in Portuguese).

[8] Madureira, Ana M., *Meta-heuristics for Single-Machine Scheduling Problems*, IMS-WG Workshop 4, Nancy-France, 1998.

[9] Michalewicz, Zbigniew, *Genetic Algorithms + Data Structures = Evolution Programs*, Third Revised and Extended Edition, Springer-Verlag Berlin Heidelberg, NewYork, 1992.

[10]Pirlot, M., *General Local Search Heuristics in Combinatorial Optimisation: a tutorial, JORBEL*, vol. 32, Bruxelas, 1992.

[11]Rinnooy Kan, A.H.G., B.J. Lageweg and J.K. Lenstra, *Minimizing Total Costs in One-Machine Scheduling*, Operations Research, Vol 23, n° 5, 908-927, 1975.

[12]Sousa, J.P. and L.A. Wolsey, *A time indexed formulations of non-preemptive single-machine scheduling problems*, Mathematical Programming, Vol. 54, n°3, 353-367, 1992.

[13]Sousa, J.P., *Time indexed formulations of non-preemptive single-machine scheduling problems*, Ph. D. Thesis, Faculté des Sciences Appliquées, Université Catholique de Oluvain, Belgium, 1989.