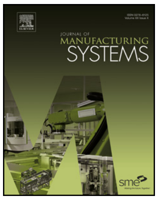
Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys



Handling ties in heuristics for the permutation flow shop scheduling problem

Dragan Vasiljevic, Milos Danilovic*

Department of Operations Management, Faculty of Organizational Sciences, University of Belgrade, Serbia

ARTICLE INFO

Article history:
Received 3 July 2012
Received in revised form 26 August 2014
Accepted 21 November 2014
Available online 19 December 2014

Keywords:
Scheduling
Constructive heuristic
Permutation flow shop
Makespan
NEH heuristic

ABSTRACT

The NEH heuristic, as the most efficient procedure for the flow shop scheduling problem is based on constructing a sequence of jobs by iteratively inserting the non-scheduled jobs into a current subsequence. The initial phase of NEH, in which an initial order (priority order) of jobs is defined, and the insertion procedure, usually cause a high number of ties. Unlike the sort of ties in the insertion phase, the ties in the initial phase are not uniquely defined by the definition of NEH. This results in an inaccuracy in most of the large number of published experimental results on this topic. The experimental research, presented in this paper confirms the importance of the inclusion of the information about the sort of ties in the initial phase in any experimental result related to NEH. The conclusion, obtained by this study, is that the range of the objective values for different sorts of ties is often greater than the improvements, published in literature. This allowed us to construct a very simple algorithm that outperforms published NEH improvements, maintaining NEH's exceptional efficiency. The proposed algorithm also uses the information about the ties in the insertion phase to improve the objective value. The permutation flow shop problem primarily concerns the makespan objective, but the main conclusions can be applied to other flow shop problems as well.

1. Introduction

The permutation flow shop problem (PFSP) with the makespan objective is one of the most thoroughly studied production scheduling problems (reviews in [1–5]). The PFSP determines the order of processing jobs, from the set $J = \{1, 2, ..., n\}$ of n independent jobs, over a set of m machines to optimize a given criterion, when all of the jobs have the same machine sequence. The processing sequence on the first machine is maintained throughout the remaining machines. The criteria most commonly studied in literature is the minimization of the total completion time, makespan ($C_{\max}$), and the corresponding problem is denoted as $F|prmu|C_{\max}$, [6]. In the *PFSP*, solutions are represented by the permutation of n jobs, so there are $n!$ possible sequences and the problem is proved to be NP-complete if m is higher than two [7].

The PFSP was first studied by Johnson in 1954 [8] and since then, an overwhelming number of papers propose different procedures trying to determine better job sequences. The efforts have been devoted to finding high-quality solutions in a reasonable computational time. A breakthrough in these efforts was obtained with the NEH heuristics [9], which is commonly regarded as the best heuristic for solving $F|prmu|C_{\max}$ (see in this regard [10–12,36]). NEH heuristic provides the initial sequence for all best metaheuristics (Agarval et al. [13]; Ekskioglu et al. [14]; Grabowski and Wodecki [15]; Haq et al. [16]; Jarboui et al. [17]; Kalczynski and Kamburowski [18]; Laha and Mandal [19]; Liao et al. [20]; Liu et al. [21]; Nowicki and Smutnicki [22]; Onwubolu and Davendra [23]; Pan et al. [24]; Ruiz and Stutzle [25]; Tasgetiren et al. [26], Ribas et al. [12], Fernandez-Viagas and Framinan [11], Dong et al. [27,28]).

NEH heuristic consists of two simple phases. First, NEH finds the priority order by sorting the jobs according to their non-increasing total processing times. Later, the first unscheduled job in this order is inserted in the best position among all possible positions of the current subsequence of already scheduled jobs. Makespan objective for the PFSP is particularly interesting for the researchers because Taillard [30] proposed an acceleration that enables that the complexity of the insertion phase is reduced from $O(n^2m)$ to $O(nm)$. In that way, the overall complexity of the NEH is reduced to $O(n^2m)$.

Usually, through the execution of NEH, ties can occur: in the priority order – jobs with the same total processing times in the initial ordering sequence; whereas in the insertion phase ties occur in different subsequences with the same best partial makespan. In known literature, much more attention is focused on the second

* Corresponding author. Tel.: +381 69 889 3236; fax: +381 11 3283 647.
E-mail address: danilovicm@fon.rs (M. Danilovic).

http://dx.doi.org/10.1016/j.jmsy.2014.11.011
0278-6125/© 2014 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

type of ties, proposing insertion tie-breaking rules. The *NEH* job priority order, π_p, was shown by Framinan et al. [31] to be superior to 177 different examined orders. Following this analysis, a large number of papers were published suggesting a new priority order and introducing tie-breaking rules in the insertion phase. In their well-known paper, Kalczynski and Kamburowski [18] suggested a better priority order combined with a simple tie-breaking in the insertion phase, securing optimality in the two-machine case and improving the general performance. Dong et al. ([27] for makespan and [28] for flowtime) proposed a starting order based on the mean and the variance of the processing times of the jobs, combined with a specific mechanism for tie-breaking in the insertion phase. According to [29], the tie-breaking mechanism proposed in this paper presents the best results that are tested on Taillard test bed. Fernandez-Viagas and Framinan [11] present a new tie-breaking mechanism based on an estimation of the idle times of the different subsequences.

All published work regarding NEH can be summed up or divided into two categories: the first attempts to improve the value of the attained makespan while keeping the high efficiency of the original procedure; whereas the second category proposes metaheuristic improvements which would advance the initial schedule, obtained through the NEH method, improving makespan without taking up too much additional CPU time. The only way of comparing the suggested improvements is by way of experimental cross-referencing using known benchmark instances. In the case of PFSP, in almost all cases, results are tested on the Taillard [32] benchmark set of 120 instances, rarely on the Watson et al. [33] benchmark suite consisting of 14,000 random-type and structured-type instances based on features found in some real-world scheduling problems.

The focus of our work is exploring the possibilities of improving makespan values by implementing simple changes to the NEH procedure. Before attempting any analysis on the subject, the four main advantages of using the NEH need to be emphasized, which classify it as the most effective heuristic, unmatched in the contemporary field of study:

1. NEH formulation is fundamentally simple, which is a significant advantage for objective experimental testing;
2. Exceptional efficiency; on computers of average capacity, NEH needs, for the largest Taillard instances, only twelve hundredths of a second. Studies have shown that in reality PFSP is a dynamic rather than a static problem, which indicates that the program execution time is a significant parameter;
3. CPU time spent on NEH is directly related to *n* and *m* only; it does not depend on the distribution of the job durations on machines. This allows the CPU time spent by NEH to be taken as the basis for comparison to other algorithms;
4. In this exceptionally short time interval, NEH determines the job schedule with the makespan value below 6% over the optimal values on Taillard instances, while on the Watson instances the deviations are even lower. It should be mentioned that for the largest Taillard instances the deviations are less than 3%. Therefore, all published work on this subject tries to make a 3% makespan improvement, thus compromising the three prior key advantages of the NEH procedure.

The above mentioned advantages of NEH impose strict criteria which need to be met by the NEH improvements. Seeing as how most improvements are related to an attempt to improve the makespan values, it is important that the authors explain their position regarding the compromising effect this has on the first three advantages of the NEH procedure. The particularity of this problem is that the range of possible makespan improvements is very limited (on average less than 3%), thus, a very precise and objective evaluation is considered a precondition. Unfortunately, almost all published work on the subject contains a certain imprecise and subjective element:

- *Ununified coding of NEH procedure*: in different articles, reported CPU time spent by the NEH procedure can differ by two orders of magnitude on similar hardware equipment. For example, the average CPU time in [34] on 500 × 20 Taillard test instances is 13.21 s, while the time using the same hardware for a code implemented in this study is twelve hundredths of a second. The exceptional difference in values is a consequence of inadequate coding of Taillard acceleration, which, as a consequence, has a repetition of bad coding in every cycle. To make matters worse, the time intervals arrived at by using this faulty process are used to make conclusions, and in the same study [34] the time of 13.21 s is taken as confirmation for their overall acceleration of the process. As a contribution for solving this type of incoherency, our study gives a detailed NEH algorithm with Taillard acceleration which is very simple and that, as far as our research indicates, yields results, in significantly shorter time than other known and published results;
- *Imprecise definition of the NEH procedure itself*: the NEH priority order, π_p, implies a non-increasing sequence for the total processing times. In the case of ties, π_p depends on the implemented sorting method. Different π_p usually result in different makespan values, obtained through the NEH procedure. Considering that a makespan value thus acquired is relevant for comparison to NEH, imprecision and partiality is obvious. It should be said that some newer studies tacitly adopted π_p in which the ties are mutually ordered in an increasing sequence of job labels. This however, does nothing to further objectivity because reference values thus acquired can be compared to improvements that implement a different sort procedure. This is why it is necessary, when presenting experimental findings, to clearly state the type of sorting and to compare the results to relevant values obtained with the same sorting procedure. This study shows that the deviations of makespan values obtained in the NEH procedure for different ties sequences in π_p are in the range of the best published NEH improvements;
- *Different reference values for the same benchmark instances*: for Taillard instances the best obtained results are updated on public sites. Various authors use different referent values information that can be either outdated or inaccurate.

Consequently, the reader is confounded upon finding, in the most prominent works of this field, different reference values used for further comparison. As an example, Table 1 gives us the average reference values used in the three previously referenced studies. The performance measure is the relative percentage increase as follows:

$$\Delta[\pi] = \frac{C_{\max}[\pi] - C_{\max}[\pi*]}{C_{\max}[\pi*]} \times 100\%, \tag{1}$$

where π^* are optimal or the best known sequences, and π are the sequences found by NEH. It should be stated that the average values shown here are for grouped test instances, the differences during individual testing of instances are even greater. The solutions are: Fernandez-Viagas and Framinan [11], denoted as FF, Dong et al. [27], denoted as *D* and Ruiz and Stutzle [25], denoted as RS.

The experimental results of analyzing the influence of sorting π_p, conducted in this study, as well as the exceptional efficiency of the well-programmed NEH procedure enable a very simple improvement of the NEH procedure which outperforms known improvements, yet do not compromise any of the three above mentioned advantages of the original NEH. This improvement also includes another type of ties, ties in the insertion phase, but unlike other known studies, without the use of tie-breaking rules.