Ernst-Moritz-Arndt-University Greifswald

Faculty of Law and Business

Discussion Papers in Economics and Business Administration

Sequencing CONWIP flow-shops: Analysis and heuristics

JOSE M. FRAMIÑAN[1]
RAFAEL RUIZ-USANO[2]
RAINER LEISTEN[3]

Discussion Paper 15/98

December 1998

[1] Departamento de Organizacion Industrial y Gestion de Empresas, Escuela Superior de Ingenieros, University of Seville, Camino de los Descubrimientos s/n. 41092 SEVILLE, SPAIN
Tel: Int + 34+54487214 Fax: Int + 34+54487329
E-mail: jose@pluto.us.es

[2] Departamento de Organizacion Industrial y Gestion de Empresas, Escuela Superior de Ingenieros, University of Seville, Camino de los Descubrimientos s/n. 41092 SEVILLE, SPAIN

[3] Production Management, Faculty of Law and Business, University of Greifswald, Fr.-Loeffler-Str 70, D-17489 GREIFSWALD, GERMANY
Tel: Int + 49+3834-862490 Fax: Int + 49+3834-862489
E-mail: leisten@rz.-uni-greifswald.de

ABSTRACT

In this paper, we address the backlog sequencing problem in a flow-shop controlled by a CONWIP production control system with the objective to minimise the makespan. We characterise the problem and analyse its similarities and differences with the permutation flow-shop problem. A comparison of some well-known flow-shop heuristics is carried out, and a simple and fast dispatching rule is proposed. Regarding the more simple and faster heuristics, the proposed dispatching rule outperforms those commonly used for the permutation flow-shop problem.

Keywords: Scheduling, Sequencing, Flow-Shop, Constant Work in Process (CONWIP), Heuristics, Dispatching Rules

ZUSAMMENFASSUNG

In diesem Arbeitspapier wird das Reihenfolgeproblem für die Einsteuerung von Aufträgen in ein Fertigungssystem (backlog sequencing) bei Reihenfertigung (Flow-Shop) behandelt, wenn das System durch eine 'Konstantes Arbeitsvolumen in der Fertigung'-Steuerung (CONstant Work In Process = CONWIP) geführt wird. Als Zielfunktion wird die Minimierung der maximalen Durchlaufzeit (C_{max}) angenommen. Das Problem wird zunächst charakterisiert. Seine Ähnlichkeiten und Unterschiede zum klassischen Reihenfolgeproblem bei Reihenfertigung mit konstanter Auftragsfolge auf allen Maschinen (Permutation-Flow-Shop) werden analysiert. Bekannte Heuristiken für den Permutation-Flow-Shop werden auf den CONWIP-Flow-Shop übertragen und verglichen. Eine einfache und schnelle Prioritätsregel wird vorgeschlagen. Wird diese Regel mit einfachen und schnellen Heuristiken für den Permutation-Flow-Shop verglichen, schneidet sie in Simulationsuntersuchungen besser als letztgenannte ab.

1. Introduction

The production control system CONWIP -acronym for CONstant Work In Process- (Spearman *et al.* 1989) is a pull system that appears to share the benefits of Kanban while being applicable in a wider range of situations. The key point of CONWIP is that it does not limit the single station's WIP or its buffer size, but the total WIP in the system. Several studies (e.g. Spearman *et al.* 1989, Lambrecht and Seagert 1990, Chang and Yih 1994, Gstettner and Kuhn 1996, and Bonvik *et al.* 1997) indicate that in certain production scenarios, this control system is preferable to others.

The detailed flow control mechanism of CONWIP is extensively discussed by Hopp and Spearman (1995). Basically, it can be described as follows. When a job order arrives to a CONWIP line, a card is attached to the job, provided there are cards available at the beginning of the line. Otherwise, the job must wait in a backlog. When a job is processed at the final station, the card is dropped off and released back to the beginning of the line, where it may be attached to the next job waiting in the backlog. Under no circumstances is a job allowed to enter into the line without a corresponding card. Intermediate buffers are established between two consecutive stations, driven by a FCFS (First Come First Served) discipline. Planning a CONWIP system therefore consists of two problems: a) Determining the number of cards, and b) sequencing jobs in the backlog.

2. Scheduling in a CONWIP system

Spearman *et al.* (1990) propose a hierarchical production planning framework for CONWIP. The two CONWIP-specific modules described in this framework are the WQS (WIP and Quota Setting) module and the SB (Sequencing and Batching) module. In the previous, a production quota for the production period and the card count – that is, the maximum WIP - are fixed while in the latter the sequence of the backlog is established once the WQS module has determined the number of cards to be used.

Referring to the decisions taken in the WQS module, several contributions have been reported (see, e.g. Dar-El *et al.* 1992, Spearman and Zazanis 1992, Hopp *et al.* 1993, and Gstettner and Kuhn 1996).

Sequencing the backlog in some cases can be reduced to the sequencing problem on a single machine (Hopp and Spearman 1995) and algorithms are available for these cases (for a review on single machine sequencing problem, see e.g. Morton and Pentiço 1993). Woodruff and Spearman (1992) addressed the sequencing problem in a CONWIP line with sequence dependent set-up times. They develop a taboo search approach that maximises the value of the work selected minus holding and set-up costs.

However, in the general case where the bottleneck machine is sequence-dependent, the sequencing problem cannot be reduced to a one-machine problem and the problem of scheduling in a CONWIP system must be specifically addressed (Spearman *et al.* 1990). It should be mentioned that this general case is supposed to be a common situation in many CONWIP environments, since one of its advantages over kanban is that it can handle a wider range of different products (Hopp and Spearman 1995).

On this general CONWIP sequencing problem, very little work has been reported. Duenyas (1994) studies the most suitable dispatching rules for a CONWIP line by using network queues. Tardiff (1995) designs an MRP-C based framework for sequencing in a CONWIP system. Finally, Herer and Masin (1997) formulate a mathematical programming version of the problem of sequencing the backlog in a CONWIP system with the objective of minimising total costs.

In this paper we address the backlog sequencing problem in a CONWIP system. In the CONWIP framework described by Spearman *et al.* (1990), sequencing is a static issue, since the backlog is generated by an MPS (Master Production Schedule) taking into account customers demand, so the composition of the backlog is known at the beginning of the production period. It is supposed that the number of cards to be employed in the system has been fixed within the previous WQS module, and thus there is no way to schedule the entrance of jobs into the system once the jobs are arranged in the backlog. This means that a job in the backlog will enter whenever there is a card available given it has been sequenced first in the queue in front of the system.

In this context, a plausible objective for sequencing the backlog seems to be minimising the makespan because:

(1) Meaning of the backlog in a CONWIP system: that is, the set of jobs that are going to be processed within the next production period. Thus, unless a high priority is assigned to a specific job, the completion of all jobs is a major issue. If different priorities are assigned to the jobs, a relevant objective could be minimisation of the weighted flow time.

(2) According to Schonberger (1984) and Spearman *et al.* (1990), the production period for a pull system in general - and particularly for a CONWIP system - must be divided into a regular production time in which the production quota can be achieved and a catch-up time which provides a time-buffer to reach the production quota if, due to unforeseen circumstances (e.g. machine breakdowns or lack of raw materials), it cannot be accomplished within the regular production time. Usually, the catch-up period is used for housekeeping, preventive maintenance, etc. In this context, makespan minimisation has the effect of ensuring the production quota to be reached within the regular production time. On the long run, it also provides a basis for a reduction of the regular production time and for shortening lead times.

(3) When generated by an MPS, a deadline for all jobs in the backlog is imposed by the production period. This means that it has been previously checked, at least approximately on a final product level, that all jobs might be accomplished within the production period. Then, due date oriented objectives might be not as relevant. Besides, CONWIP being a pull system, the customers demand is served from the finished goods inventory. In this context, Berkley (1992) observes that it is not clear how the notion of due dates might be applied, since released cards correspond to demands that have already occurred. However, if the backlog results immediately from customer demand, or if the immediate satisfaction of the demand of a specific job is required, then the minimisation of tardiness or weighted tardiness may be important objectives as well.

(4) Makespan minimisation has been found to reduce scheduling costs significantly (see e.g. Gupta and Dudek 1971, and Panwalkar *et al.* 1973).

Formally, the problem addressed here is the sequencing of the backlog for a CONWIP line with m stations. The backlog is composed of n jobs allows to have different processing times job by job and station by station. The maximum WIP allowed is given by NT and it is equal to the number of cards. The objective is the minimisation of the makespan. In the following, we will name this the 'CONWIP sequencing problem'. An instance of this problem can be described by n, m and NT, along with the processing times of each job in each station.

Other assumptions made in the formulation of this problem include those habitual in the classic scheduling literature: Simultaneous availability of all jobs and all stations, deterministic processing times, etc. For a complete list of these assumptions, see e.g. Dudek and Teuton (1964).

According to the standard triple notation for scheduling problems introduced by Graham *et al.* (1979), the CONWIP flow shop scheduling can be characterised as $F_m|conwip|C_{max}$. Note that the *prmu* attribute in the notation might be omitted, since the FCFS discipline of the intermediate buffers in the system restricts the space of solutions to permutations.

The CONWIP sequencing problem also might be interpreted as a special case of the more general flow-shop sequencing problem with limited buffer storage. The buffer constrained flow-shop sequencing problem has two extreme cases: the unconstrained sequencing problem (i.e. no buffer constraints which might be interpreted as infinite intermediate buffers) and the zero or no-wait sequencing problem. In the two latter the intermediate buffers are zero, but the no-wait sequence problem is even more restrictive in the sense that the starting time of a job on the first machine has to be chosen such that the job does not have to wait for its processing on any machine. The unconstrained sequencing problem has been studied extensively, and a good reference for zero and no-wait problems is provided by Hall and Sriskandarajah (1996).

However, the intermediate case in which buffers are different from zero and not infinite has been addressed by relatively few researchers for very specific purposes (Dutta and Cunningham 1975, Reddi 1976, Papadimitriou and Kanellakis 1980, Leisten 1990, Logendran and Sriskandarajah 1993, and Sharadapriyadarshini and Rajendran 1997), despite it is considered the most realistic situation in many manufacturing environments (Wismer 1972, and Reddi and Ramamoorthy 1972). Besides, this type of problems is considered in principle to be more difficult than the extreme cases (Morton and Pentico 1993).

With the exception of Dutta and Cunningham, who develop a dynamic programming procedure to generate optimal solutions with the objective of minimising the maximum completion times for the two-station problem with limited buffers, the rest of the papers are mainly devoted to the development of heuristics for the problems.

Maybe the most complete work in this direction is done by Leisten, who compares several heuristics for the m-station flow-shop sequencing problem with different intermediate buffer sizes. The tested heuristics include buffer-constrained specific (including Dutta and Cunningham, Reddi, and Papadimitriou and Kanellakis and two new heuristics suggested in the paper) as well as $F_m|prmu|C_{max}$ heuristics. It turns out that the NEH heuristic (Nawaz *et al.* 1983), employed for the unconstrained buffer problem, performs well according to the objective of minimising makespan. These results suggest that some well-known $F_m|prmu|C_{max}$ heuristics may be suitable also for the CONWIP sequencing problem, which is connected to the buffer-constrained problem with respect to regarding buffer capacities within the system. However, in contrast to fixed buffer sizes between each two machines, the CONWIP problem supposes a fixed number of jobs within the entire system being either processed on a machine or waiting for their next operation.

3. The structure of the problem

To obtain a first insight into the CONWIP-problem, the empirical distributions of all possible makespans obtained by complete enumeration of 1 000 instances of seven machines and seven jobs have been calculated. This may reveal similarities and differences between the CONWIP-problem and the $F_m|prmu|C_{max}$ problem, although only qualitative conclusions might be extrapolated from these instances. Each instance has been generated using random processing times (integers between one and 99) and all possible makespans have been obtained for a card count ranging from two to seven. Note that for one card, each sequence is optimal while for seven cards the problem is identical to $F_7|prmu|C_{max}$. The number of machines and jobs has been chosen as is in order to obtain the results within a reasonable computation time.

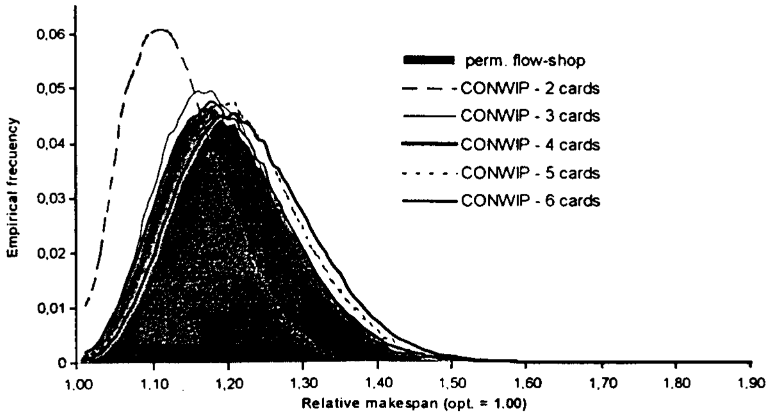

Figure 1. Empirical distributions of the makespans in a seven-machine, seven-job setting, depending on the number of cards

From figure 1, it can be seen that there are basically three patterns of curves. The first is obtained for a very low card count – two cards -, and shows a small dispersion of the makespan. Finding a good solution is rather easy: 1% of the solutions are at most 1% above the optimum, and a random solution has a 0.5 probability of being at most less than 12% above the best makespan. The second pattern is reached for an intermediate number of cards (four cards is the representative in the example). In this case finding a near-optimum solution becomes difficult: only 0.08% of the solutions are at most 1% above the optimum. Here, the makespan distribution reaches its flattest point and chances to obtain randomly a relative good schedule are lower than in any other case. Finally, the third pattern is given by the permutation flow-shop case, and it is easier to find a good solution than in the previous pattern (0.24% of the makespans are at most 1% above the optimum). The rest of the curves seem to represent transitions between two of the three previously commented patterns For instance, the CONWIP six cards curve nearly fits the flow-shop makespan distribution.

However, from a practical outlook it seems there should be a focus primarily onto the second pattern: that is when the card count is neither very low nor close to the number of jobs The first problem is easy to solve and even a 'bad' sequence does not have as much impact on the system performance. For the latter case, we can use numerous techniques developed for the $F_m|prmu|C_{max}$ problem. Besides of discussing the second pattern itself, an additional problem therefore arises from determining the borderline between the second pattern, i.e. the 'real' CONWIP-problem, and the third pattern, i.e. the ordinary permutation flow shop case.

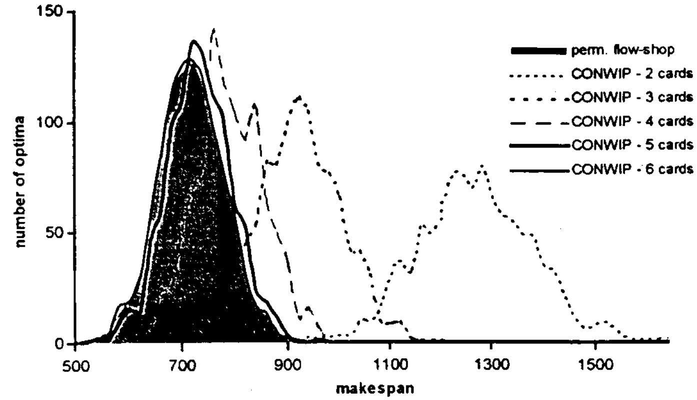

Figure 2: Empirical distributions of the optimal makespans

In figure 2, the distributions of the optimal makespan for the 1 000 problem instances are shown. The distributions are almost symmetric. While for six cards instances, the optimal values of the makespan coincide with those obtained for the permutation flow-shop, the situation is different when the number of cards is lower. A more exhaustive analysis by inspection of the vectors of the optimal job sequences confirms that these vectors, within this small numerical study, are identical for the six cards problem and for the flow-shop problem, and almost the same in most of the instances of the five cards problem as compared with the flow-shop problem. However, the analysis reveals that, in general, the vector of the optimal solutions for two, three and four cards are different as compared with those obtained for the ordinary permutation flow- shop case.

The main outcome of this small numerical study is that the $F_m|conwip|C_{max}$ problem, in general, is different from the $F_m|prmu|C_{max}$ problem. Not only the values of the objective function are different (which is obvious, since the use of cards enlarges the completion times), but also the optimal sequences and the structure of the space of solutions are different. This indicates that suitable methods for the $F_m|prmu|C_{max}$ problem have at least to be checked whether they also work for $F_m|conwip|C_{max}$. If not, special approaches for the CONWIP-problem have to be developed.

4. Comparison of heuristics

In order to evaluate different heuristics, a set of test problems has been generated. As mentioned before, the problem characteristics in the $F_m|conwip|C_{max}$ problem are given by n, m and NT together with the processing time of each job on each machine.

Although the ratio n/m might have some influence on the results, we have found no particular setting for this in the design of the experiments contained in papers related to the permutation flow-shop problem. Indeed, this ratio varies from very low values ($\in [0.2, 1]$ from the instances generated by Nawaz *et al.* 1983, and Zegordi *et al.* 1995) to relatively high values ($\in [4/3, 4]$ taken from the test-bed of Reeves, 1995). On the other hand, more extensive problem collections cover wider intervals of this ratio: in Taillard (1990) it ranges from one to 25, while in Ho and Chang (1991) the range is from 0.175 to 37.5.

Adopting an intermediate position, we have chosen the most common values for n and m found in the previously cited papers. Thus, the instances have been developed as follows: $(n,m) \in \{(10,10), (15,10), (20,10), (20,20), (30,10), (30,20)\}$. With respect to the values for NT, we tried to catch those problems close to the second pattern discussed in the previous section, although we have no guarantee that the instances will behave in that way. Indeed, after running the experiments we realise that some of the problems may respond better to the third pattern (closer to $F_m|prmu|C_{max}$), but we keep them as representatives of a 'real' situation in a CONWIP system. Therefore, from a total set of $NT \in \{5,8,10,12,15,18,20,25\}$, we exclude those combinations of (n, m) and NT in which the number of cards was equal or higher than n (that is, the $F_m|prmu|C_{max}$ case) and those in which NT was very low with respect to n in order to avoid the easy instances corresponding to the first pattern of problems presented in the previous section.

For a given triplet of n, m and NT, ten instances have been generated with random processing times ranging from one to 99. This distribution of the processing times is known to produce difficult scheduling problems (Campbell *et al.* 1970, and Dannenbring 1977). In total, the test-bed contains 180 instances.

The tested heuristics are those due to Palmer (1965), Gupta (1971), CDS (Campbell *et al* 1970), RA or Rapid Access (Dannenbring 1977), and the previously mentioned of NEH. The three latter perform makespan calculation in several consecutive steps, so the card number limitation must be explicitly taken into account. We will call them CONWIP-CDS, CONWIP-RA and CONWIP-NEH to indicate that in the evaluation of partial schedules, the available number of cards has been taken into account. For the Palmer and Gupta heuristics, no specific adaptation is required, since both are based in the construction of indices, taking into account exclusively the processing times of the jobs.

Besides the above mentioned heuristics, we also test the PF (Profile Fitting) heuristic, introduced by McCormick *et al.* (1989) and proposed in Pinedo (1995) for the flow-shop with limited intermediate storage problem. In this heuristic, the first job of the sequence is the one with the smallest sum of processing times. This job generates a profile determined by its departure times from every machine. The next job to be sequenced is chosen among the remaining jobs in order to fit the profile of the previous job (that is, to minimise the idle and blocking time in the machines during the processing of this job). The procedure is repeated until all jobs are arranged. This heuristic has been also adapted for a CONWIP system in the calculations of the profiles of the jobs taking into account the availability of cards, so in the following it will be named CONWIP-PF.

To compare the relative efficiency of the heuristic, an upper bound of each instance has been found by long runs (about 10 000 objective function evaluations) of a standard simulated annealing algorithm. The quality of the solutions is represented by the percentage deviation of this upper bound. In table 1, the mean percentage deviation from the best known solution of each type of problem is presented.

The results shown in table 1 reveal that the NEH heuristic is by far the most suitable among the tested heuristics. The poor performance of the CONWIP-RA heuristic is surprising to a certain extent, since it is usually considered a rather suitable heuristic for the $F_m|prmu|C_{max}$ problem when short computation times are required (Taillard, 1990). This reinforces the idea already expressed in section 3 that suitable methods for the $F_m|prmu|C_{max}$ problem may not work well for $F_m|conwip|C_{max}$.

Problem size (n/mNT)	CONWIP-NEH	CONWIP-CDS	Palmer	Gupta	CONWIP-PF	CONWIP-RA
10/10/5	4.619	5.209	12.310	13.274	15.665	19.860
10/10/8	1.919	4.718	11.883	12.528	14.767	26.471
15/10/5	7.070	8.133	19.726	16.790	21.841	22.116
15/10/8	6.175	8.269	18.593	16.613	24.524	26.098
15/10/10	4.251	9.030	14.964	16.793	22.789	27.932
15/10/12	3.811	9.468	14.075	16.512	22.837	28.635
20/10/10	5.683	8.348	18.130	18.571	24.732	27.660
20/10/15	5.523	8.244	14.877	18.500	26.265	29.000
20/10/18	5.523	8.244	14.877	18.500	26.100	29.000
30/10/10	5.575	10.594	19.854	21.300	22.796	28.783
30/10/20	3.518	10.132	13.850	18.534	23.391	32.673
30/10/25	3.523	10.137	13.855	18.539	22.769	32.679
20/20/10	28.375	28.606	40.169	42.797	36.083	45.860
20/20/15	13.752	15.283	23.555	25.823	26.476	35.201
20/20/18	10.165	14.693	20.519	24.925	26.543	34.518
30/20/10	6.704	8.955	15.858	18.358	20.856	19.986
30/20/20	3.746	8.695	13.712	21.093	24.167	28.385
30/20/25	3.518	9.258	14.027	21.719	23.374	29.057
Mean:	**6.858**	**10.334**	**17.491**	**20.065**	**23.665**	**29.106**

Table 1: Mean percentage deviation of the best solution (upper bound by simulated annealing approach) for the heuristics

A second experiment tries to find reasons for the superior behaviour of the NEH heuristic. This heuristic can be basically divided into two phases: first the jobs are arranged with respect to descending sums of their processing times. Within the second phase, a job sequence is constructed by evaluating the partial schedules originating from the initial order of the first phase: Supposing a sequence already determined for the first k jobs, job $k+1$ is placed in front of the first job of this sequence, between the first and the second job, ..., and finally between the k-1st and the kth job of the given sequence of the previously placed k jobs. Out of these $k+1$ sequences, the sequence yielding the minimum completion time for the $k+1$ jobs is kept as relative sequence for these first $k+1$ jobs given by phase one. Then, job $k+2$ from phase 1 is considered analogously and so on until all jobs have been sequenced. Once a precedence order between two jobs has been established in a partial schedule, this order is maintained for the rest of the evaluations. Thus, the initial order of the jobs bounds the solutions that are going to be explored. The number of subsequences to be evaluated in the NEH-heuristic is rather large as compared with other heuristics

Therefore, also for the CONWIP-problem setting it can be analysed whether the excellent performance of the CONWIP-NEH heuristic is based on the large number of evaluations, the initial sequence of jobs generated in phase one, or on the combination of both.

Therefore, in an additional experiment, we try to evaluate the contribution of the initial order of jobs to the CONWIP-NEH heuristic's performance. To do this, we have run the CONWIP-NEH heuristic's phase 2 starting from a randomly generated initial order of jobs, replacing the initial order generation according to sums of processing times. The results indicate to what extent the final sequence in a NEH heuristic is caused by the initial order of the jobs (phase 1) and how much is due to the evaluations of the partial sequences (phase 2).

Problem size n/m/NT	CONWIP-NEH	NEH	RANDOM-NEH
10/10/05	4.619	9.136	6.452
10/10/08	1.919	2.997	2.553
15/10/05	7.070	16.115	8.172
15/10/08	6.175	13.353	6.267
15/10/10	4.251	5.859	4.339
15/10/12	3.811	3.811	4.587
20/10/10	5.683	10.845	5.573
20/10/15	5.523	5.523	3.608
20/10/18	5.523	5.523	3.608
30/10/10	5.575	8.843	5.941
30/10/20	3.518	3.518	4.670
30/10/25	3.523	3.523	4.675
20/20/10	28.375	36.753	30.641
20/20/15	13.752	21.832	14.697
20/20/18	10.165	10.207	10.545
30/20/10	6.704	13.093	8.034
30/20/20	3.746	5.296	4.620
30/20/25	3.518	3.518	4.309
Mean:	**6.858**	**9.986**	**7.405**

Table 2: Mean percentage deviation of the best solution for the heuristics

Besides, the NEH heuristic is by far the best constructive heuristic known for the permutation flow-shop sequencing problem (Taillard 1990), although its main shortcoming is the time required to arrive to the solution, compared with other heuristics. This is because the NEH heuristic is $O(n^3 m)$, while Gupta's, RA and Palmer's are $O(n \log(n) + n m)$, and CDS is $O(n m^2 + m n \log(n))$. As mentioned before, in CONWIP sequencing, it is not possible to know *a priori* whether the optimal solution of an instance is effectively constrained by the number of cards (pure CONWIP) or it is not. Thus, it might be that in some cases the good performance of NEH is due to the fact that the problem instance is not a 'true' CONWIP setting. To take this into account, for each problem we have run the original NEH heuristic – that is, without taking into account the available number of cards -, afterwards we have evaluated this solution with explicitly taking the number of cards into account in the time computation. The results are shown in table 2.

In total, the performance of the CONWIP-NEH and the RANDOM-NEH are rather similar, indicating that the performance of CONWIP-NEH is due mainly to the evaluation of the enormous number of partial sequences and not to the initial order of the jobs. Surprisingly, in some cases, the RANDOM-NEH even outperforms CONWIP-NEH. Further experiments carried out on Taillard's library (Taillard 1993) for the $F_m|prmu|C_{max}$ problem show that the previous conclusions can be extended to this case.

Comparing NEH and CONWIP-NEH, it is easy to see that on one hand the latter dominates the first and on the other hand the latter obtains its best results whenever both heuristics coincide, that is, when the number of cards is sufficient for the NEH heuristic to obtain the same solution without taking into account the WIP limitation. This roughly means that the best CONWIP-NEH results are obtained when dealing with non WIP-constrained instances. CONWIP-NEH's performance decreases for the cases with lower card count. Although the reasons are not clear yet, this might indicate that in a $F_m|conwip|C_{max}$ problem it is not primarily important to obtain good partial sequences that minimise the partial makespans. Instead, it will be more desirable that the jobs – particularly the first ones - keep a pace through the system that can regularly release cards for the incoming jobs. In other words, these partial schedules must minimise the partial sum of the flow times at least for the first n - NT jobs entering the system, due that they have to provide the cards for the last n - NT jobs that have no cards from the beginning. Further research has to be carried out to confirm or refuse this hypothesis, although partially available results suggest this expectation to be appropriate.

Finally, in order to provide a fast heuristic for the CONWIP sequencing problem, based on the above analysis we propose a simple dispatching rule. It operates as follows: first, calculate the sum of the processing times of each job in the system. Second, arrange the jobs in ascending order. Finally, allocate the jobs alternatively in the beginning or in the end of the schedule, allocating the jobs with higher sum of processing times in the centre of the schedule. Note that this can be achieved by placing the shortest processing time job at the beginning of the sequence or at the end. Consequently, two possible sequences might be originated in this manner, being one the inverse of the other. In the proposed 'Centre' dispatching rule, we evaluate both of them and we chose the one with the best value of the makespan.

For instance, if the ascending order of the sums of their processing times of a certain problem instance with six jobs is (1,2,3,4,5,6), the results of the dispatching rule are the schedules (1,3,5,6,4,2) and (2,4,6,5,3,1).

The rationale for this procedure is as follows: in a CONWIP system it is important that the jobs entering in the first places have a short processing time, since the cards that are attached to them must be used for further jobs. This is a critical condition for the first jobs, since a delay of the first jobs is propagated along the sequence. On the other hand, entering the jobs with higher processing times at the end causes the makespan to be enlarged, and most of the time gained within processing the first jobs is spoiled by large processing times at the end of the schedule. Thus for a CONWIP system, the location of a job with high processing time is less harmful with respect to the makespan of the sequence if it is in the middle of the sequence.

n/m/NT	Palmer	Gupta	CONWIP-RA	Centre	SPT	LPT	SIRO
10/10/05	14.849	13.274	19.860	7.403	17.494	22.103	19.322
10/10/08	11.001	12.528	26.471	13.829	19.990	20.387	17.726
15/10/05	21.359	16.790	22.116	11.572	17.758	20.460	24.187
15/10/08	19.288	16.613	26.098	16.614	22.085	24.841	28.126
15/10/10	14.817	16.793	27.932	20.894	24.631	26.290	28.357
15/10/12	13.181	16.512	28.635	21.540	25.287	26.966	29.025
20/10/10	18.761	18.571	27.660	20.196	25.561	27.019	29.458
20/10/15	14.055	18.500	29.000	21.573	26.355	27.403	29.525
20/10/18	14.055	18.500	29.000	21.573	26.355	27.403	29.525
30/10/10	17.902	21.300	28.783	16.753	24.000	23.481	27.826
30/10/20	11.962	18.534	32.673	20.110	25.276	24.944	27.441
30/10/25	11.968	18.539	32.679	20.115	25.282	24.950	27.447
20/20/10	39.899	42.797	45.860	33.483	42.054	43.968	48.167
20/20/15	24.448	25.823	35.201	27.607	28.336	31.462	31.581
20/20/18	20.469	24.925	34.518	27.686	28.234	31.468	31.178
30/20/10	16.706	18.358	19.986	13.672	17.517	17.538	21.829
30/20/20	13.732	21.093	28.385	20.862	24.666	23.425	24.030
30/20/25	14.064	21.719	29.057	21.494	25.320	24.067	24.677
Mean:	**17.362**	**20.065**	**29.106**	**19.832**	**24.789**	**26.010**	**27.746**

Table 3: Mean percentage deviation of the best solution for the dispatching rules

We have tested this heuristic on the previously described set of problems along with other common dispatching rules. For comparison we have selected the SPT (Shortest Processing Times) rule, the LPT (Longest Processing Times) rule, and the SIRO (Service In Random Order) rule. We also compared the results with those obtained for the fastest heuristics for the flow-shop sequencing problem, i. e. RA, Gupta's and Palmer's heuristics.

The results of this experiment are shown in table 3. Note that CPU times are not reported here since differences are neglectable, although those provided by Gupta and Palmer heuristics are slightly higher than for the others.

From table 3, it is clear that the central distribution of the higher processing times jobs is the most efficient of the four dispatching rules solely based on the processing times (SPT, LPT, Centre) and the SIRO rule. It behaves better than CONWIP-RA, and similarly to Palmer, being much faster than the latter.

Although on the overall, the Palmer heuristic behaves better than the central dispatching rule, this situation changes for those problems with lower number of cards. In table 4, a subset of 60 problems is shown corresponding to those instances in which the number of cards is the lowest for a given number of jobs and stations. In these cases, the central distribution of the jobs outperforms the other heuristics, including Palmer's. This suggests that the central dispatching rule is particularly suitable when the number of cards is lower, that is, in the most pure CONWIP instances.

n/m/NT	Palmer	Gupta	CONWIP-RA	Centre	SPT	LPT	SIRO
10/10/05	14.849	13.274	19.860	7.403	17.494	22.103	19.322
15/10/05	21.359	16.790	22.116	11.572	17.758	20.460	24.187
20/10/10	18.761	18.571	27.660	20.196	25.561	27.019	29.458
30/10/10	17.902	21.300	28.783	16.753	24.000	23.481	27.826
20/20/10	39.899	42.797	45.860	33.483	42.054	43.968	48.167
30/20/10	16.706	18.358	19.986	13.672	17.517	17.538	21.829
Mean:	**21.579**	**21.848**	**27.378**	**17.180**	**24.064**	**25.762**	**28.465**

Table 4: Mean percentage deviation of the best solution for the heuristics (only instances with the lowest NT for a given n and m are selected)

5. Conclusions

The results of the experiments reveal that well-known heuristics successfully used for the $F_m|prmu|C_{max}$ problem cannot be extended to the CONWIP sequencing problem. This might be seen as a confirmation that $F_m|prmu|C_{max}$ and $F_m|conwip|C_{max}$ are, in general, different problems, as it is derived from the results obtained in section 3.

As far as simple heuristics are concerned, CONWIP-RA heuristic, which is considered to offer very good results with short computation times in the $F_m|prmu|C_{max}$ problem, is clearly outperformed by the other heuristics. For more elaborate heuristics, it appears that the behaviour of the NEH heuristic, which offers the best results among the tested heuristics, is partly due to the numerous partial evaluations of sequences, and partly due to the fact that some problem instances are not WIP-constrained.

Finally, a simple dispatching rule is devised for the CONWIP sequencing problem. This rule has proven to be particularly suitable for those instances with lower card count. In these situations, the rule also clearly outperforms the more common fast heuristics used for the $F_m|prmu|C_{max}$ problem.

The study presented here can be extended definitely: As far as the relation of *n*, *m* and *NT* as well as the structure of the processing times are concerned, the border line between simple flow shop behaviour of a problem or a problem instance respectively and the relevance of the CONWIP constraints can be explored. Additionally, heuristics and dispatching rules exploiting the CONWIP problem structure in more detail might be developed. We plan work on these aspects. However, the study presented here gives a first insight into the structure of the sequencing problem of CONWIP problem settings.

References

Berkley, B. J., 1992, A review of the Kanban production control research literature. *Production and Operations Management*, **1**, 393-411

Bonvik, A. M., Couch, C., and Gershwin, S. B., 1997, A comparison of production-line control mechanisms. *International Journal of Production Research*, **35**, 789-804

Campbell, H. G., Dudek, R. A. and Smith, M. L., 1970, A heuristic algorithm for the n-job, m-machine sequencing problem. *Management Science*, **16**, B630-B637

Chang, T. M., and Yih, Y., 1994, Generic kanban system for dynamic environment. *International Journal of Production Research*, **32**, 889-902

Dannenbring, D. G., 1977, An evaluation of flow-shop sequence heuristics. *Management Science*, **23**, 1174-1182

Dar-El, E. M., Herer, Y. T., and Masin, M., 1992, CONWIP-based production lines with multiple bottlenecks: performance and design implications. Technical report, Israel Institute of Technology, Haifa, Israel.

Duenyas, I., 1994, A simple release policy for networks of queues with controllable input. *Operations Research*, **42**, 1162-1171

Dudek, R. A., and Teuton, O. F., 1964, Development of m stage decision rule for scheduling n jobs through m machines. *Operations Research*, **12**, (3).

Dutta, S. K., and Cunningham, A., 1975, Sequencing two-machine flow-shops with finite intermediate storage. *Management Science*, **21**, 989-996

Graham, R. L, Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G., 1979, Optimisation and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, **5**, 287-326

Gstettner, S. and Kuhn, H., 1996, Analysis of production systems kanban and CONWIP. *International Journal of Production Research*, **34**, 3253-3273

Gupta, J. N. D., 1971, A functional heuristic algorithm for the flowshop scheduling problem. *Operational Research Quarterly*, **22**, 39-47

Gupta, J. N. D., and Dudek, R. A., 1971, Optimality criteria for flow-shop scheduling. *AIIE Transactions*, **3**, 199-205

Hall, N. G. and Sriskandarajah, C., 1996, A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, **44**, 510-525

Herer, Y. T. and Masin, M., 1997, Mathematical programming formulation of CONWIP based production lines and relationships to MRP. *International Journal of Production Research*, **35**, 1067-1076

Ho, J. C. and Chang, Y. L., 1991, A new heuristic for the n-job, m-machine flow-shop problem. *European Journal of Operational Research*, **52**, 194-202

Hopp, W. J. and Spearman, M. L., 1995, *Factory Physics* (Chicago, USA; Irwin).

Hopp, W. J., Spearman, M. L., and Duenyas, I., 1993, Economic Production Quotas for pull manufacturing systems. *IIE Transactions*, **25**, 71-79

Lambrecht, M. and Seagert, A., 1990, Buffer Stock Allocation in Serial and Assembly Type of Production Lines. *International Journal on Production Management*, **10**, 47-61

Leisten, R., 1990, Flow-shop sequencing problems with limited buffer storage. *International Journal of Production Research*, **28**, 2085-2100

Logendran, R., and Sriskandarajah, C., 1993, Two-machine group scheduling problem with blocking and anticipatory set-ups. *European Journal of Operational Research*, **69**, 467-481

McCormick, S. T., Pinedo, M. L., Shenker, S., and Wolf, B., 1989, Sequencing in an assembly line with blocking to minimize cycle time. *Operations Research*, **37**, 925-936

Morton, T. E., and Pentico, D. W., 1993, *Heuristic Scheduling Systems* (NY; Willey Interscience)

Nawaz, M., Enscore, E. E., and Ham, I., 1983, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *OMEGA*, **11**, 91-95

Palmer, D. S., 1965, Sequencing jobs through a multistage process in the minimum total time: a quick method of obtaining a near-optimum. *Operational Research Quarterly*, **16**, 101-107

Panwalkar, S. S., Dudek, R. A., and Smith, M. L., 1973, Sequencing research and the industrial scheduling problem. Proceedings on the symposium on theory of scheduling and its applications, New York, pp. 29-38

Papadimitriou, C., and Kanellakis, P., 1980, Flow-shop scheduling with limited temporary stage. *Journal of the Association for Computing Machinery*, **27**, 533-549

Pinedo, M., 1995, *Scheduling: theory, algorithms and systems* (Englewood Cliffs, NJ; Prentice Hall).

Reddi, S. S., 1976, Sequencing with intermediate storage. *Management Science*, **23**, 216-217

Reddi, S. S., and Ramamoorthy, C. V., 1972, On the flow-shop sequencing problem with no wait in process. *Operational Research Quarterly*, **23**, 323-331.

Reeves, C. R., 1995, A genetic algorithm for flowshop sequencing. *Computers Operational Research*, **22**, 5-13

Schonberger, R. J., 1984, *World class manufacturing: the lessons of simplicity applied* (NY, The Free Press).

Sharadapriyadarshini, B., and Rajendran, C., 1997, Scheduling in a kanban controlled flow-shop with dual blocking mechanism and missing operations for part-types. *International Journal of Production Research*, **35**, 3133-3156

Spearman, M. L., Hopp, W. J., and Woodruff, D. L., 1990, A hierarchical control architecture for Constant Work-In-Process (CONWIP) production systems. *Journal of Manufacturing and Operations Management*, **2**, 147-171

Spearman, M. L., Woodruff, D. L., and Hopp, W. J., 1989, CONWIP: a pull alternative to kanban. *International Journal of Production Research*, **28**, 879-894

Spearman, M. L., and Zazanis, M. A., 1992, Push and pull production systems: issues and comparisons. *Operations Research*, **40**, 521-532

Taillard, E., 1990, Some efficient heuristic methods for the flow-shop sequencing problem. *European Journal of Operational Research*, **47**, 65-74

Taillard, E., 1993, Benchmark for basic scheduling problems, *European Journal of Operational Research*, **64**, 278-285

Tardiff, V., 1995, Detecting scheduling infeasibilities in multi-stage, finite capacity production environments, PhD. Thesis, Northwestern University, USA.

Wismer, D. A., 1972, Solution of the flow-shop sequencing problem with no intermediate queue. *Operations Research*, **20**, 689-697

Woodruff, D. L., and Spearman, M. L., 1992, Sequencing and batching for two classes of jobs with due dates and set-up times. *Journal of Production and Operations Management*, **1**, 87-102

Zegordi, S.H., Itoh, K., and Enkawa, T., 1995, Minimizing makespan for flow-shop scheduling by combining simulated annealing with sequencing knowledge. *European Journal of Operational Research*, **85**, 515-531

Reddi, S. S., and Ramamoorthy, C. V., 1972, On the flow-shop sequencing problem with no wait in process. *Operational Research Quarterly*, **23**, 323-331.

Reeves, C. R., 1995, A genetic algorithm for flowshop sequencing. *Computers Operational Research*, **22**, 5-13

Schonberger, R. J., 1984, *World class manufacturing: the lessons of simplicity applied* (NY, The Free Press).

Sharadapriyadarshini, B., and Rajendran, C., 1997, Scheduling in a kanban controlled flow-shop with dual blocking mechanism and missing operations for part-types. *International Journal of Production Research*, **35**, 3133-3156

Spearman, M. L., Hopp, W. J., and Woodruff, D. L., 1990, A hierarchical control architecture for Constant Work-In-Process (CONWIP) production systems. *Journal of Manufacturing and Operations Management*, **2**, 147-171

Spearman, M. L., Woodruff, D. L., and Hopp, W. J., 1989, CONWIP: a pull alternative to kanban. *International Journal of Production Research*, **28**, 879-894

Spearman, M. L., and Zazanis, M. A., 1992, Push and pull production systems: issues and comparisons. *Operations Research*, **40**, 521-532

Taillard, E., 1990, Some efficient heuristic methods for the flow-shop sequencing problem. *European Journal of Operational Research*, **47**, 65-74

Taillard, E., 1993, Benchmark for basic scheduling problems, *European Journal of Operational Research*, **64**, 278-285

Tardiff, V., 1995, Detecting scheduling infeasibilities in multi-stage, finite capacity production environments, PhD. Thesis, Northwestern University, USA.

Wismer, D. A., 1972, Solution of the flow-shop sequencing problem with no intermediate queue. *Operations Research*, **20**, 689-697

Woodruff, D. L., and Spearman, M. L., 1992, Sequencing and batching for two classes of jobs with due dates and set-up times. *Journal of Production and Operations Management*, **1**, 87-102

Zegordi, S.H., Itoh, K., and Enkawa, T., 1995, Minimizing makespan for flow-shop scheduling by combining simulated annealing with sequencing knowledge. *European Journal of Operational Research*, **85**, 515-531

Ernst-Moritz-Arndt-Universität Greifswald

Rechts- und Staatswissenschaftliche Fakultät

Wirtschaftswissenschaftliche Diskussionspapiere

Bisher erschienen:

1/97 Ole Janssen/Carsten Lange: "Subventionierung elektronischer Geldbörsen durch staatliche Geldschöpfungsgewinne"

2/97 Bernd Frick: "Kollektivgutproblematik und externe Effekte im professionellen Team-Sport: "Spannungsgrad" und Zuschauerentwicklung im bezahlten Fußball"

3/97 Frauke Wilhelm: "Produktionsfunktionen im professionellen Mannschaftssport: Das Beispiel Basketball-Bundesliga"

4/97 Alexander Dilger: "Ertragswirkungen von Betriebsräten / Eine Untersuchung mit Hilfe des NIFA-Panels"

1/98 Volker Ulrich: "Das Gesundheitswesen an der Schwelle zum Jahr 2000"

2/98 Udo Schneider: "Der Arzt als Agent des Patienten – Eine Anwendung der Principal-Agent-Theorie auf die Arzt-Patient-Beziehung"

3/98 Volker Ulrich/Manfred Erbsland: "Short-run Dynamics and Long-run Effects of Demographic Change on Public Debt and the Budget"

4/98 Alexander Dilger: "Eine ökonomische Argumentation gegen Studiengebühren"

5/98 Lucas Bretschger: "Nachhaltige Entwicklung der Weltwirtschaft: Ein Nord-Süd-Ansatz"

6/98 Bernd Frick: "Personal-Controlling und Unternehmenserfolg: Theoretische Überlegungen und empirische Befunde aus dem professionellen Team-Sport"

7/98 Xenia Matschke: "On the Import Quotas on a Quantity-Fixing Cartel in a Two-Country-Setting"

8/98 Tobias Rehbock: "Die Auswirkung der Kreditrationierung auf die Finanzierungsstruktur der Unternehmen"

9/98 Ole Janssen/Armin Rohde: „Einfluß elektronischer Geldbörsen auf den Zusammenhang zwischen Umlaufsgeschwindigkeit des Geldes, Geldmenge und Preisniveau"

10/98 Stefan Degenhardt: „The Social Costs of Climate Change - A Critical Examination“

11/98 Ulrich Hampicke: „Remunerating Conservation: The Faustmann-Hartmann Approach and its Limits“

12/98 Bretschger, Lucas: „Dynamik der realwirtschaftlichen Integration am Beispiel der EU-Osterweiterung“

13/98 Burchert, Heiko: „Ökonomische Evaluation von Telematik-Anwendungen im Gesundheitswesen und Schlußfolgerungen für ihre Implementierung“

14/98 Dilger, Alexander: The Absent-Minded Prisoner