



HAL
open science

Discrepancy search for solving flexible scheduling problems

Abir Ben Hmida, Mohamed Haouari, Marie-José Huguet, Pierre Lopez

► **To cite this version:**

Abir Ben Hmida, Mohamed Haouari, Marie-José Huguet, Pierre Lopez. Discrepancy search for solving flexible scheduling problems. 12th International Workshop devoted to Project Management and Scheduling (PMS 2010), Apr 2010, Tours, France. 4p. hal-00462415

HAL Id: hal-00462415

<https://hal.science/hal-00462415>

Submitted on 9 Mar 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discrepancy Search for solving flexible scheduling problems

Abir Ben Hmida³, Mohamed Haouari³, Marie-José Huguet^{1,2}, and Pierre Lopez^{1,2}

¹ CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France

² Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France
e-mails: huguet@laas.fr, lopez@laas.fr

³ Ecole Polytechnique de Tunisie, Unité ROI, La Marsa, Tunisia
e-mails: bh_abir@yahoo.fr, mohamed.haouari@ept.rnu.tn

Keywords: Scheduling, Flexible Job Shop, Hybrid Flow Shop, Two-Stage Hybrid Flow Shop, Discrepancy Search.

1. Flexible Scheduling Problems: Problem statements

Flexible scheduling problems under study are Flexible Job Shop (FJS) and Hybrid Flow Shop (HFS), which generalize traditional scheduling problems. For flexible scheduling problems it is desired to process operations on a machine chosen among a set of available ones. The objective is to find a schedule which minimizes the makespan. The FJS problem is more computationally difficult than the Job Shop problem, since it presents a further decision level besides the sequencing one, *i.e.*, the job assignment. This problem is known to be strongly NP-hard, even if each job has at most three operations and there are two machines (Garey *et al.*, 1976). Brandimarte (1993) was the first to use a decomposition approach based on Tabu Search (TS) for the FJS problem. Hurink *et al.* (1994) and Barnes and Chambers (1996) developed TS algorithms to solve the FJS problem. Dauzère-Pérès and Paulli (1997) defined a new neighbourhood structure for the problem where there was no distinction between reassigning and resequencing an operation, and proposed a TS algorithm. Mastrolilli and Gambardella (2000) improved the previous work and presented two neighbourhood functions. Their TS is the well-known efficient approach for solving FJS problems. Pezzella *et al.* (2008) presented a genetic algorithm (GA) in which a mix of different strategies for generating the initial population, selecting individuals for reproduction, and reproducing new individuals is presented. Gao *et al.* (2008) studied the FJS with a multi-objective approach. They developed a hybrid genetic algorithm (hGA) based on the integrated approach for this problem. Their algorithm is the well-known competitive genetic algorithm for solving the FJS.

In the HFS there is a set of L stages, machines used at each stage are identical, and successive operations of a job have to be processed serially through the L stages. Solving the HFS problem consists in assigning a specific machine to each operation of each job as well as sequencing all operations assigned to each machine. The HFS problem is NP-hard even if it contains two stages and when there is, at least, more than one machine at a stage (Gupta, 1988). Most of the literature has considered the case of only two stages, denoted by 2-HFS. Several methods were proposed for solving both general HFS and 2-HFS, most of them are based on specific lower bounds. We have already proposed an efficient adaptation of discrepancy-based method in (Ben Hmida *et al.*, 2007) for the HFS and we only consider in this abstract the 2-HFS problem.

In this paper, we propose specific adaptations of the Climbing Depth-bounded Discrepancy Search (CDDS) method developed in (Ben Hmida *et al.*, 2007) to solve two new flexible scheduling problems (2-HFS and FJS). To the best of our knowledge, the use of discrepancy-based methods was only used for solving the Hybrid Flow Shop (Ben Hmida *et al.*, 2007).

The remainder of the abstract is organized as follows. Section 2 introduces the principles of CDDS and some adaptation for the problems under study. Section 3 presents CDDS performance through different well-known benchmarks.

2. Adaptation of discrepancy-based search approach

Discrepancy-based methods are tree-search methods based on the concept of discrepancy to expand the search tree. Limited Discrepancy Search (LDS), proposed in (Harvey, 1995), starts from initial variable instantiations suggested by a given heuristic and successively explores

branches with increasing discrepancies from it, *i.e.*, by changing the instantiation of some variables. Basically, a discrepancy occurs if the choice of a variable does not follow the rank of the ordering heuristic (the initial instantiation). The method stops when a solution is found (if such a solution does exist) or when an inconsistency is detected (the tree is entirely expanded). The concept of discrepancy was first introduced for binary variables. For taking into account non-binary variables we consider (based on experimental studies) that choosing the first ranked value leads to 0 discrepancy, while choosing all other ranked values implies 1 discrepancy. The main drawback of LDS is to be highly redundant: during the search for solutions with k discrepancies, solutions with 0 to $k - 1$ discrepancies are revisited. To avoid this, several improvements of LDS were proposed. One of them consists in applying discrepancy first at the top of the tree to correct early mistakes in the instantiation heuristic; this yields the Depth-bounded Discrepancy Search method (DDS) proposed in (Walsh, 1997).

Climbing Discrepancy Search (CDS) is another improvement of LDS. It is a local search method that adapts the concept of discrepancy to find a good solution for combinatorial optimization problems (Milano and Roli, 2000). It starts from an initial solution delivered by a given heuristic. Then, a sequence of neighbourhoods including nodes with discrepancy equal to 1, 2, ..., k_{max} respectively, are consecutively explored. When a leaf with an improved value of the objective function is found, the reference solution is updated, the number of discrepancies is reset to 0, and the process for exploring the neighbourhoods is restarted. The idea of applying discrepancies only at the top of the search tree can be also joined with CDS algorithm to limit the search tree expansion. Therefore, we use Climbing Depth-bounded Discrepancy Search (CDDS) (Ben Hmida *et al.*, 2007) which combines both the CDS and the DDS methods. With this method, one can restrict neighbourhoods to be visited by only using discrepancies on variables at the top of the tree.

The efficiency of the discrepancy-based methods is firstly based on dedicated instantiation heuristics. For the studied problems there are two kinds of variables (for the operations selection and for the resources assignment) and the instantiation heuristics are defined for both kinds of variables. (1) *Selection of operations*: for the FJS problems (as for the HFS ones), the priority is firstly given to the operation with the Earliest Start Time. In case of ties, for the HFS problem, we have considered several alternatives. For the FJS problem, we only consider the operation belonging to the job having the Earliest Due Date. For 2-HFS, we use an extension of Johnson's rule to schedule the first stage and the Earliest Start Time-Longest Job Duration rule for the second one. (2) *Assignment of machines to operations*: For all of the problems, the operation previously chosen is assigned to the machine such that the task completes as soon as possible (Earliest Completion Time). After these instantiations, a constraint propagation mechanism updates the starting time of successor operation and maintains the availability date of the chosen resource.

Despite the two kinds of variables, for 2-HFS (as for HFS), discrepancies are only applied on job selection variables because machines are identical at each stage. For solving FJS we consider discrepancies on the two kinds of variables. To limit the tree search expansion, we have proposed to introduce the computation of lower bounds (LB) in CDDS for solving HFS problems. For solving the 2-HFS problem, we also propose to introduce in CDDS a lower bound based on Shortest-Processing-Time rule developed in (Haouari and M'Hallah, 1997). For the FJS problems, we do not find efficient LBs, thus we propose to explore different neighbourhoods based on block notion characteristics which consider pairs of critical operations (Jurisch, 1992). These neighbourhoods imply that discrepancies are not applied to all problem variables. Only some relevant ones, chosen by using the block notion, are considered. Therefore, using neighbourhoods prunes the process of discrepancies by focusing on promising variables with regards to the considered criteria.

3. Computational experiments

The proposed variants of CDDS was coded in C and implemented on an Intel Core 2 Duo 2.9 GHz Personal Computer with 2GB of RAM. We set the maximum CPU time to 15 s for all test instances except for DPdata instances (see below) for which it is set to 200 s because of the huge values of operation processing times. If no optimal solution was found within time limit, then the

search is stopped and the best solution is output as the final schedule. The depth of discrepancy in our method is fixed at 7 from the top of the tree; this number has been experimentally chosen. The mean relative error is calculated as follows: $100\% \times (C_{max} - LB) / LB$.

For solving FJS problems, we define four neighbourhoods for CDDS (denoted respectively by N1, N2, N3, and N4) and we keep the best result on these four neighbourhoods. We considered the following instances: BRdata is a set of 10 problems from (Brandimarte, 1993); BCdata is a set of 21 problems from (Barnes and Chambers, 1996); DPdata is a set of 18 problems from (Dauzère-Pérès and Paulli, 1997) and HUdata is a set of 129 problems from (Hurink *et al.*, 1994).

We first compare the efficiency of the four neighbourhoods over all instances in terms of Mean Relative Error (MRE) and Computer-Independent CPU time in seconds (CPU). We note that CDDS-N4 and CDDS-N2 outperform other strategies in terms of quality of solutions but they spend a little more time. Indeed, they both present a mean relative error of 7.46% within about 780 s (comparatively to 7.52% for CDDS-N1 and 7.47% for CDDS-N3, and to 751 s for CDDS-N1 and 768 s for CDDS-N3). This result corresponds to the use of a dynamic rule (Earliest Completion Time) in both N2 and N4, which provides an additional time to calculate new starting times and new assignments.

Table 1. Comparison of the MRE of algorithms CDDS, GA, TS, and hGA for solving FJS

Instances	Nb	CDDS	GA	TS	hGA	dev. (CDDS,GA)	dev. (CDDS,TS)	dev. (CDDS, hGA)
BRdata	10	14.98	17.53	15.14	14.92	-2.55	-0.16	0.06
BCdata	21	22.54	29.56	22.53	22.61	-7.02	0.01	-0.07
DPdata	18	1.94	7.63	2.01	2.12	-5.69	-0.07	-0.18
Hurink Edata	43	2.32	6.00	2.17	2.13	-3.68	0.15	0.19
Hurink Rdata	43	1.34	4.42	1.24	1.19	-3.08	0.10	0.15
Hurink Vdata	43	0.12	2.04	0.095	0.082	-1.92	0.03	0.04

Results of Table 1 show that the proposed CDDS method outperforms GA of (Pezzella *et al.*, 2008) for the optimization of all types, TS of (Mastrolilli and Gambardella, 2000) on BRdata and DPdata, and hGA of (Gao *et al.*, 2008) on BCdata and DPdata. But hGA and TS algorithms both worked better than the proposed CDDS algorithm on HUdata.

For solving the 2-HFS problem, we considered 1680 instances generated in the same way as in (Lee and Vairaktarakis, 1994) which are composed of three sets (560 instances per set): (1) In set A the processing times are drawn randomly either from the discrete uniform distribution in $[1, 20]$ for the first stage and in $[1, 40]$ for the second stage. (2) In set B the processing times on the first stage were drawn randomly from the discrete uniform distribution in $[1, 40]$ and in $[1, 20]$ for the second one. (3) In set C the processing times on both stages were drawn randomly from the discrete uniform distribution in $[1, 40]$.

First, we propose to compare solutions obtained by the CDDS variant dedicated for 2-HFS (denoted by $CDDS^2$) with the CDDS variant developed initially for solving the general hybrid flow shop problem (denoted by $CDDS^1$). Results given by each algorithm are compared with LBs used in the B&B algorithm developed in (Haouari *et al.*, 2006).

$CDDS^2$ presents a tiny mean relative error ($\leq 0.20\%$) considering all instances and it solved to optimality 90% of problems belonging on set A, 91% of problems of set B, and 86% of problems of set C. In comparison $CDDS^1$ has a higher mean relative error (between 0.41% and 0.89%).

$CDDS^2$ is also compared with TS method developed in (Haouari and M'Hallah, 1997) and Table 4 summarizes obtained results. As shown in Table 2, $CDDS^2$ outperforms the tabu search method in all instances, except the instances of the subclass (4, 2) of Set B. It is however assumed that $CDDS^2$ is very efficient for all problem sizes. In most problems, the $CDDS^2$ procedure yields optimal or very near-optimal solutions.

Relating to the results obtained from the computational study, we conclude that the proposed variants of CDDS provide efficient approaches that are comparable with the state of the art.

Table 2. Performance comparison between CDDS and TS sets for solving 2-HFS

n		Performance on Set A			Performance on Set B			Performance on Set C			Average MRE
		(2, 4)	(4, 4)	(4, 2)	(2, 4)	(4, 4)	(4, 2)	(2, 4)	(4, 4)	(4, 2)	
20	CDDS	0.95	0.26	0.00	0.03	0.14	0.73	0.00	1.48	0.05	0.40
	TS	2.90	1.20	0.35	0.92	5.72	0.13	0.56	3.43	1.22	1.83
30	CDDS	0.92	0.10	0.00	0.00	0.11	0.96	0.07	1.45	0.02	0.40
	TS	1.43	0.85	0.06	0.57	3.10	0.05	0.27	1.45	1.46	1.03
40	CDDS	0.21	0.05	0.00	0.00	0.05	0.28	0.02	0.46	0.01	0.12
	TS	0.96	0.43	0.12	0.5	1.57	0.12	0.34	1.08	0.89	0.67
50	CDDS	0.15	0.06	0.00	0.00	0.02	0.37	0.00	0.88	0.02	0.16
	TS	0.54	0.30	0.02	0.26	1.09	0.04	0.20	0.95	0.42	0.42
100	CDDS	0.06	0.02	0.00	0.00	0.02	0.03	0.00	0.22	0.01	0.04
	TS	0.19	0.15	0.02	0.11	0.39	0.01	0.07	0.41	0.18	0.17
Average MRE	CDDS	0.46	0.10	0.00	0.01	0.07	0.48	0.02	0.90	0.02	0.22
	TS	1.20	0.59	0.11	0.47	2.37	0.07	0.29	1.46	0.83	0.82

References

- Barnes J.W. and Chambers J.B., (1996). "Flexible job shop scheduling by tabu search". Graduate Program in Operations and Industrial Engineering, The University of Texas at Austin, *Technical Report Series, ORP96-09*.
- Ben Hmida A., Huguet M.-J., Lopez P., and Haouari M., (2007). "Climbing Discrepancy Search for solving the hybrid Flow shop". *European Journal of Industrial Engineering*, 1(2):223–243.
- Brandimarte P., (1993). "Routing and scheduling in a flexible job shop by tabu search". *Annals of Operations Research*, 41:157–183.
- Dauzère-Pérés S. and Paulli J., (1997). "An integrated approach for modelling and solving the general multiprocessor job-shop scheduling problem using tabu search". *Annals of Operations Research*, 70:281–306.
- Garey, M.R., Johnson, D.S., and Sethi, R., (1976). "The complexity of flow shop and job shop scheduling". *Mathematics of Operations Research*, 1:117-129.
- Gao J., Sun L., and Gen M., (2008). "A hybrid genetic and variable neighbourhood descent algorithm for flexible job shop scheduling problems". *Computers & Operations Research*, 35(9):2892–2907.
- Gupta J.N.D. (1988). "Two-stage hybrid flowshop scheduling problem", *Journal of the Operations Research Society*, Vol. 39, pp.359–364.
- Haouari M., Hidri L., and Gharbi A., (2006). "Optimal scheduling of a two hybrid flow shop". *Mathematical Methods of Operations Research*, 64:107–124.
- Haouari M., and M'Hallah R., (1997). "Heuristic algorithms for the two-stage hybrid flowshop problem". *Operations Research Letters*, 21:43–53.
- Harvey W.D., (1995). "Nonsystematic backtracking search". *PhD thesis*, University of Oregon.
- Hurink E., Jurisch B., and Thole M., (1994). "Tabu search for the job shop scheduling problem with multi-purpose machines". *Operations Research-Spektrum*, 15:205–215.
- Jurisch B., (1992). "Scheduling jobs in shops with multi-purpose machines". *PhD dissertation*, Fachbereich Mathematik/Informatik, Universität Osnabrück.
- Mastrolilli M. and Gambardella L.M., (2000). "Effective neighbourhood functions for the flexible job shop problem". *Journal of Scheduling*, 3:3–20.
- Milano M. and Roli A., (2002). "On the relation between complete and incomplete search: an informal discussion". *Proceedings CPAIOR'02*, p.237–250, Le Croisic, France.
- Pezzella F., Morganti G., and Ciaschetti G., (2008). "A genetic algorithm for the flexible job-shop scheduling problem". *Computers & Operations Research*, 35(10):3202–3212.
- Walsh T., (1997). "Depth-bounded Discrepancy Search". *Proceedings IJCAI-97*, p.1388–1395, Nagoya, Japan.