AN INTERACTIVE MULTICRITERIA OPTIMISATION APPROACH TO SCHEDULING

Martin Josef Geiger, Sanja Petrovic
Automated Scheduling, Optimisation & Planning Research Group
School of Computer Science & IT, University of Nottingham, UK
mjg@cs.nott.ac.uk, sxp@cs.nott.ac.uk

Abstract Scheduling problems overall assume that it is possible to identify stable criteria definitions measuring the quality of alternatives. In real world problems however, this does not necessarily have to be the case. Situations may change over time or even within the decision making process, and so may criteria and preferences of the decision maker.

The paper presents an interactive multicriteria guided optimisation framework for production scheduling. The methodology enables the decision maker to successively change the definitions of optimality criteria and his/her preferences. The methodology was tested on a real-world scheduling problem faced by the Sherwood Press Ltd, a printing company based in Nottingham, UK.

Keywords: Planning and Scheduling, Evolutionary Systems, Genetic Algorithms

1. Introduction

Scheduling in real-world manufacturing environments is a complex problem with a considerable amount of research activities in the fields of operations research, computer science and artificial intelligence, going back several decades (Brucker, 2001). The general problem of scheduling can be described to be a problem of assigning resources to tasks over time subject to a set of side constraints (e.g. resource capacity constraints, etc.) with the goal of optimising one or more objectives.

Over the years, various classes of scheduling problems have been investigated, and consequently different methods have been developed (Pinedo, 2002). Nevertheless, the impact of scheduling research on the real world problems has been limited. One of the reasons is that scheduling algorithms usually employ a single objective function which often fails to reflect preferences of the decision maker.

Multicriteria approaches to scheduling allow the integration of several, usually conflicting aspects or 'points of view' (Roy, 1996) that have to be considered simultaneously. Due to the complexity of most problems heuristic and metaheuristic techniques have been used for their solving with increasing popularity (Coello Coello et al., 2002).

This paper presents a methodology which enables the decision maker to change the set of optimality criteria and his/her preferences during the search process. The paper is organised as follows. Some concepts of multiobjective optimisation relevant for this paper are given in Section 2. Section 3 describes the production scheduling problem that was studied. Section 4 proposes a novel framework for interactive multicriteria optimisation, overcoming restrictions. The proposed methodology was tested on real-world data provided by Sherwood Press, a printing company based in Nottingham, UK, and the results are presented in Section 5. Conclusions are given in Section 6.

2. Concepts from multiobjective optimisation

The multicriteria decision making (MCDM) process involves four phases shown in Figure 1: formulation of a model which includes identification of optimality criteria and identification of the alternatives (the search space), search for the alternatives, the choice of the most preferable alternative and the execution of the selected solution. With each schedule $S \in \mathcal{S}$ a vector of objective functions $G(S) = \big(g_1(S), \ldots, g_z(S)\big)$ is associated.

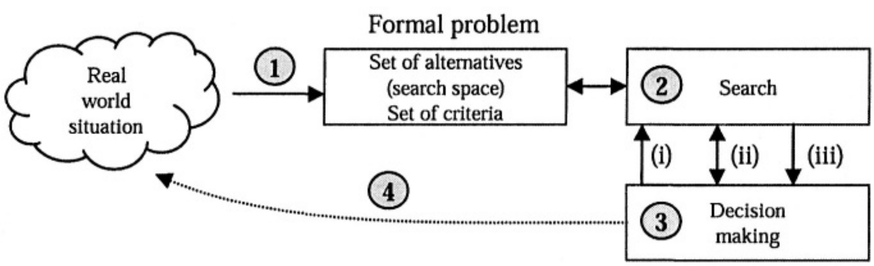

Figure 1. Multicriteria model building, search, and decision making process.

As criteria may be conflicting, the notion of optimality is interpreted in the sense of *Pareto optimality* (see e.g. Van Veldhuizen and Lamont, 2000), which is based on the concept of dominance relations among objective vectors. Without loss of generality, we assume in the further descriptions the minimisation of objective values.

DEFINITION 1 (PARETO DOMINANCE RELATION) *A vector of objective functions G(S) is said to dominate an vector* $G(S')$ *if and only if* $\forall i \;\; g_i(S) \leq g_i(S') \wedge \exists i \mid g_i(S) < g_i(S')$. *We denote the dominance of a vector G(S) over a vector* $G(S')$ *with* $G(S) \preceq G(S')$.

Using the dominance relation, the definition of Pareto optimality is derived as follows.

DEFINITION 2 (PARETO OPTIMALITY) *A solution S is said to be Pareto optimal if and only if* $\neg\exists S' \in \mathcal{S} \mid G(S') \preceq G(S)$. *The set of all Pareto optimal solutions is called the Pareto set P for which* $P \subseteq \mathcal{S}$ *holds.*

As visualised in steps 2 and 3 in Figure 1, the solving of the scheduling problem can now be seen in identifying a most preferred schedule $S^* \in P$, which itself is twofold. First, a Pareto optimal schedule is computed, which is $\mathcal{NP}$-hard if the scheduling problem for at least one criterion definition is $\mathcal{NP}$ (Ehrgott, 2000). Second, an element $S^* \in P$ has to be selected. Three general strategies are possible.

(i) *A priori.* The preferences of the decision maker are obtained before the search.

(ii) *Interactive.* The problem solving alters between search and decision making, successively revealing preferences.

(iii) *A posteriori.* After identifying the Pareto set *P*, the decision problem is solved using multicriteria decision aiding techniques (Vincke, 1992).

While a few approaches combine the set of criteria to an overall evaluation function (Allahverdi, 2003), most existing applications to multicriteria scheduling are a posteriori approaches of Pareto optimisation (Bagchi, 1999). Interactive applications are however comparably scarce (Hapke et al., 1998).

3. Problem statement

Machine environment

We investigated the scheduling problem faced by the Sherwood Press printing company which may be best characterised as a flexible job-shop problem with release dates and due dates (Błażewicz et al., 2001). The machine environment is characterised by a set $\mathcal{M} = \{M_1, \ldots, M_m\}$ of machines organised into disjunct working centres. Seven working centres have been identified to be of relevance for scheduling: printing, cutting, folding, cards insertion, embossing/debossing, gathering/stitching/trimming, and packaging.

As processing times depend on both the machine and the particular task, the working centres may be regarded as consisting of unrelated parallel machines. The availability of the machines changes over time. Some machines may also be operated on Saturdays while others are only available from Monday till Friday. Compared with problems known from literature, in the described problem shifts of the job floor have to be respected, allowing the assignment of tasks to machines only within a specific time window on each day.

Job characteristics

Scheduling in the investigated problem has to deal with a set $\mathcal{J} = \{J_1, \ldots, J_j, \ldots, J_n\}$ of jobs, each of them consisting of a set $J_j = \{T_{j1}, \ldots, T_{jk}, \ldots, T_{jt}\}$ of tasks. The tasks are ordered with respect to a technically required processing sequence which is known in advance. Associated with each job J_j is a release date r_j, a due date d_j, and a nonnegative weight w_j reflecting its relative importance for the decision maker. While the release date must not be violated, the due date constraints are desirable to satisfy but is is often impossible to find a solution which violates none of them.

Each task T_{jk}, being able to be processed on one or several machines of a certain working centre, has a quantity q_{jk} indicating the size of the task, e.g. the amount of sheets that have to be printed. Furthermore, processing times p_{ijk}, setup times s_{ijk} and cleanup times c_{ijk} for each task T_{jk} on machine M_i are given.

It has to be noticed that some tasks are not processed as a whole as they would exceed the capacity of the processing machine on a certain day. These tasks are split into smaller processing units called lots. In a general formulation, for each task T_{jk} exists a set of lots $\mathcal{L}_{jk} = \{L_{jk1}, \ldots, L_{jku}, \ldots, L_{jkl}\}$, $l \geq 1$. Here, a lot L_{jku} has a quantity q_{jku} such that

$$\sum_{u=1}^{l} q_{jku} = q_{jk} \quad j = \{1, \ldots, n\}, \; k = \{1, \ldots, t\} \tag{1}$$

While setup- and cleanup times are not dependent on the quantity of the lot, the processing time p_{ijku} of lot L_{jku} on machine M_i can be computed as

$$p_{ijku} = \frac{q_{jku}}{q_{jk}} p_{ijk} \tag{2}$$

In the studied problem, the decision about how to split tasks with longer processing times is not treated separately from the problem of finding a schedule, and the lot quantities q_{jkl} are additional decision variables.

Optimality criteria

Meeting the agreed due dates is an important goal. Taken the completion time C_j of job J_j, we are able to obtain the total weighted tardiness:

$$g_1(S) = \sum_{j=1}^{n} w_j T_j \tag{3}$$

where

$$T_j = \max\{0, C_j - d_j)\} \tag{4}$$

Although the splitting of tasks into lots may enable a parallel processing and a possible earlier completion, setup- and cleanup times are together with the organisational overhead accordingly increased. The second objective is the minimisation of the number of lots.

$$g_2(S) = \sum_{j=1}^{n} \sum_{k=1}^{t} | \mathcal{L}_{jk} | \tag{5}$$

Given the defined objective functions g_1 and g_2, the problem can now be treated as a vector optimisation problem.

$$\text{minimise } G(S) = \left(g_1(S), g_2(S)\right) \tag{6}$$

However, the definition of optimality criteria cannot be regarded as exhaustive due to two reasons:

1 In practice, the general objective functions aggregate tardiness and number of lots over a large number of jobs, implying the possibility that different schedules might have similar or even identical evaluations.

2 It may not be possible to formulate all desired objectives in the phase of the model construction, meeting the formal requirement of exhaustiveness of the set of criteria (Bouyssou, 1990). Some information might not be present from the very beginning but are discovered during the search and decision making process.

As a result, it may occur that none of the Pareto optimal solutions given a certain definition of criteria is preferred by the decision maker.

While existing multicriteria approaches already consider the problem of possibly changing preferences during search in interactive techniques, a methodology for changing optimality criteria, to our knowledge has not been proposed yet.

4. A novel framework for interactive multicriteria optimisation

In the proposed framework, the decision maker is allowed to redefine the set of criteria interactively during the search process in order to refine the notion of optimality according to the specific situation. Three cases are possible:

1 A new objective function is introduced.

2 An existing objective function is removed from the objective vector $G(S)$.

3 An existing evaluation function is altered. As an example, the weights w_j of the jobs may be changed during the search.

```
[1] Create initial population of solutions POP
[2] If no change in criteria definition is detected
[3]       Select schedule S ∈ POP
[4]       Create neighbouring solution S^nh using S
[5]       Update population with S^nh with respect to nondominance
[6] Else
[7]       Recompute weak nondominance relations in POP
[8]       Return to [2]
```

Figure 2. Multicriteria guided evolutionary algorithm.

Obviously, changes within the objective vector have an impact on the evaluation of the alternatives. A closer investigation reveals that the concept of Pareto optimality may not be sufficient to anticipate possible changes of the set of criteria. We therefore propose the concept of *weak nondominance* to be used.

DEFINITION 3 (WEAK NONDOMINANCE) *An objective vector G(S) is said to be weakly nondominated if and only if* $\neg\exists S' \in \mathcal{S} \mid \forall i \; g_i(S') < g_i(S)$.

With respect to Definitions 2 and 3, a Pareto optimal solution is also weakly nondominated but not vice versa. However, weakly nondominated solutions may become Pareto optimal if the definition of criteria is altered such that a criterion for which inequality within the objective vector holds is removed.

During the search, all weakly nondominated solutions are kept in an archive, which is successively updated. The introduction of conflicting criteria accordingly results in an archive having a larger cardinality.

An evolutionary algorithm for interactive scheduling

The proposed framework is based on an evolutionary algorithm. A population oriented approach has been chosen for implementation as a whole set of weakly nondominated solutions should be found simultaneously. As the pseudo code for the algorithm in Figure 2 shows, in the case of occurring changes of the optimality criteria the weakly nondominance relations among the individuals of the population are updated, resulting possibly in a removal of alternatives that do not meet Definition 3.

Lot-sizing

The splitting of larger tasks is crucial for the further assignment to the machines. Numerous small lots should be avoided, while tasks with longer processing times have to be devided such that they meet the availability time windows of the machines. A probabilistic decision rule has been used to decide

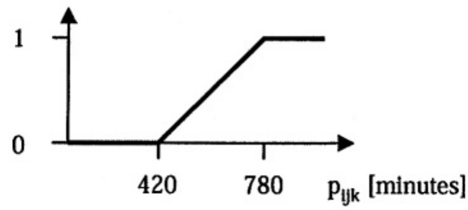

Figure 3. Probabilistic splitting of tasks depending on their processing time p_{ijk}.

whether tasks should be split into smaller processing units. For each task T_{jk}, a probability of splitting is derived depending on its processing tune p_{ijk} on machine M_i. With respect to the known daily capacity of the machines and the duration of the shifts, tasks are not split if their processing time is lower or equal to 7 hours (shift is 8 hours). Starting with 7 hours, the split probability is monotonically increasing up to a maximum value of 1 being reached at 13 hours which is depicted in Figure 3.

In the case of a splitting, a uniform number of splits between $\lceil \frac{p_{ijk}}{780} \rceil$ and $\lceil \frac{p_{ijk}}{420} \rceil$ is chosen.

Representation and decoding

The schedule encoding of the evolutionary algorithm consists of a set of job permutations, one for each machine. At the beginning of the optimisation procedure, lots are assigned randomly to technically possible machines and their sequences are randomly generated. As different assignments are possible, the permutations of the machines can have different elements (lots) and consequently can be of different length.

To obtain a schedule with start and end dates for the lots, the permutational representation is decoded using the approach of (Giffler and Thompson, 1960) for constructing active schedules avoiding cycles within the precedence graph of the schedule. Here, all lots are subsequently scheduled while conflicts for processing on the same machine are resolved with respect to the sequence in the permutation, giving leftmost occurring lots priority. An example of this representation is given in (Mattfeld and Bierwirth, 2004).

Operators

As different schedules might have different chromosome lengths, existing crossover techniques of combining two encodings are not applicable. Instead, a set of mutation operators is used, and at each iteration a neighbourhood solution is generated by applying one of the following operators with equal probability:

1 *Resplitting.* One of the splitted tasks is randomly chosen and the number of defined splits is changed within the given interval.

2 *Resequencing.* The position of a single lot on a particular machine is changed by means of a shift operator as described in (Reeves, 1999), shifting it forward of backward in the sequence.

3 *Reassignment.* A lot is removed from a machine and reassigned to a different machine from the set of machines appropriate for the task.

5. Results

The algorithm has been tested on a real world data set from Sherwood Press, containing the workload of four weeks (18 machines, 64 jobs, 218 tasks). In total 50 test runs have been performed starting with different initial populations, each containing 50 individuals, leading to an overall approximation of the Pareto set *P* as shown in Figure 4. In each test, 100,000 schedules have been computed, keeping the best found alternatives from generation to generation.

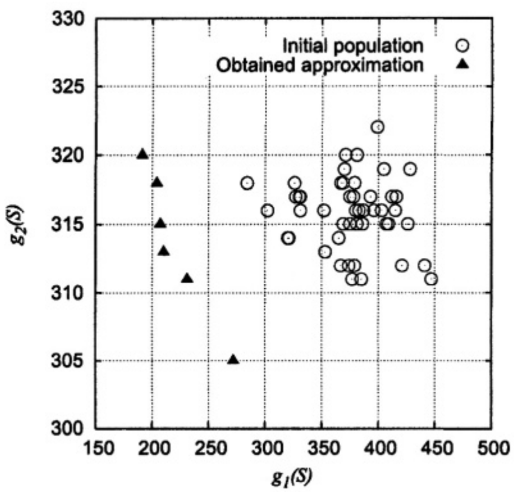

Figure 4. Results for the problem instance.

As suspected, the results show a tradeoff between the number of lots defined by the splitting procedure and the weighted tardiness. Numerous smaller lots are easier to schedule with respect to the total weighted tardiness. However, it has to be taken into consideration, that $g_1(S)$ does not discriminate between individual jobs but aggregates over their whole set, therefore allows a compensation of tardiness of different jobs.

In order to improve the quality of the schedules further, an additional objective function g_3 measuring the tardiness of a specific, highly important but late job was introduced during the search. It can be seen in Figure 5 that similar schedules with respect to the criterion g_1 are more clearly distinguishable by criterion g_3. While this aspect of evaluating the schedules is added to the set of criteria, the existing information of g_1 and g_2 is kept.

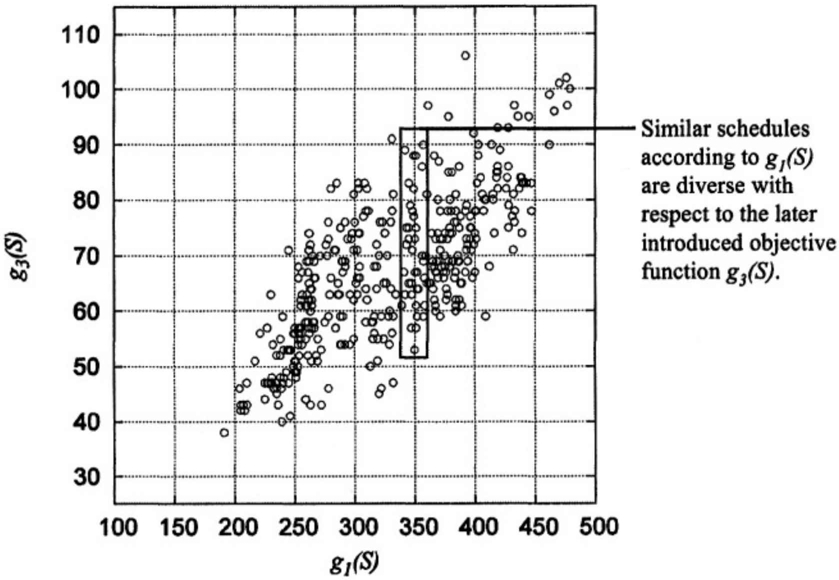

Figure 5. Comparison of schedules according to their g_1 and g_3 values.

The introduction of an additional criterion is the most direct way of expressing the importance of the mentioned job. Another possible action could be changing the weight of the selected job. However, it would affect the relative importance of jobs, and it might not be easy for the decision maker in practical situations with numerous jobs to find a proper weight adaptation in order to obtain the intended results. Also, the introduction of a new criterion does not allow compensation of tardiness of jobs.

6. Conclusions

A general approach for interactive multicriteria optimisation has been presented. An evolutionary algorithm has been proposed and applied to a problem from the printing industry. We believe that the successive introduction of criteria during the search is an important factor reflecting decision making in complex scheduling environments while guiding the search to preferred regions of the search space. Each 'point of view' is added or removed during the search and decision making procedure in a step-by-step procedure while maintaining transparency for the decision maker.

Apart from the application for scheduling in the printing industry, the methodology is of general use for complex decision problems where the relevant criteria are changing over time and have to be developed interactively by the decision maker.

Acknowledgments

The authors would like to thank their industrial collaborator Sherwood Press Ltd., Nottingham, and three anonymous referees for their helpful suggestions. This research is supported by the Engineering and Physical Sciences Research Council (EPSRC), UK, Grant No. GR/R95319/01.

References

Allahverdi, Ali (2003). The two- and m-machine flowshop scheduling problem with bicriteria of makespan and mean flowtime. *European Journal of Operational Research,* 147:373–396.

Bagchi, Tapan P. (1999). *Multiobjective scheduling by genetic algorithms.* Kluwer Academic Publishers, Boston, Dordrecht, London.

Błażewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., and Węglarz, J. (2001). *Scheduling Computer and Manufacturing Processes.* Springer, Berlin, Heidelberg, New York, 2nd edition.

Bouyssou, Denis (1990). Building criteria: A prerequisite for MCDA. In Bana E Costa, Carlos, editor, *Readings in Multiple Criteria Decision Aid,* pages 58–80. Springer, Heidelberg.

Brucker, Peter (2001). *Scheduling Algorithms.* Springer, Berlin, 3rd edition.

Coello Coello, Carlos A., Van Veldhuizen, David A., and Lamont, Gary B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic Publishers.

Ehrgott, Matthias (2000). *Multicriteria Optimization.* Springer, Berlin, Heidelberg, New York.

Giffler, B. and Thompson, G. L. (1960). Algorithms for solving production-scheduling problems. *Operations Research,* 8:487–503.

Hapke, Marciej, Jaszkiewicz, Andrzej, and Słowiński, Roman (1998). Interactive analysis of multiple-criteria project scheduling problems. *European Journal of Operational Research,* 107:315–324.

Mattfeld, Dirk C. and Bierwirth, Chrstian (2004). An efficient genetic algorithm for job shop scheduling with tardiness objectives. *European Journal of Operational Research,* 155:616–630.

Pinedo, Michael (2002). *Scheduling: Theory, Algorithms, and Systems.* Precentice Hall, Upper Saddle River, NJ, 2nd edition.

Reeves, Colin R. (1999). Landscapes, operators and heuristic search. *Annals of Operations Research,* 86:473–490.

Roy, Bernard (1996). *Multicriteria Methodology for Decision Aiding.* Kluwer Academic Publishers, Dordrecht.

Van Veldhuizen, David A. and Lamont, Gary B. (2000). Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation,* 8(2): 125–147.

Vincke, Philippe (1992). *Multicriteria Decision-Aid.* John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore.