A Robust Meta-Hyper-Heuristic Approach to Hybrid Flow-Shop Scheduling

José Antonio Vázquez Rodríguez and Abdellah Salhi

Mathematical Sciences Department, University of Essex, Wivenhoe Park, Colchester, U. K. {javazq@gmail.com, as@essex.ac.uk}

Summary. Combining meta-heuristics and specialised methods is a common strategy to generate effective heuristics. The inconvenience of this practice, however, is that, often, the resulting hybrids are ineffective on related problems. Moreover, frequently, a high cost must be paid to develop such methods. To overcome these limitations, the idea of using a hyper-heuristic to generate information to assist a meta-heuristic, is explored. The devised approach is tested on the Hybrid Flow Shop (HFS) scheduling problem in 8 different forms, each with a different objective function. Computational results suggest that this approach is effective on all 8 problems considered. Its performance is also comparable to that of specialised methods for HFS with a particular objective function.

1 Introduction

Hybrid Flow Shops (HFS) are manufacturing environments in which a set of jobs must be processed in a series of stages with multiple parallel machines [2]. Given the NP-hard nature, [16], [21], of this problem, to be effective, generic methods rely on the information provided by specialised ones. Often, this information is decided by the objective being optimised. Consequently, the resulting hybrids are not as effective on problems with other objectives. This is a serious shortcoming since, in practice, it is desirable to have solution tools that are reliable on problems with different objectives.

Successful algorithms for HFS use meta-heuristics to schedule the first stage of the shop and a simple constructive procedure to schedule the rest, [29], [28], [34], [15], [26]. Genetic Algorithms (GA) have been used successfully in exploiting this idea. In [23] and [22], for instance, a Random Keys Genetic Algorithm (RKGA) performed well against many specialised heuristics and meta-heuristics such as the problem space-based search method, [24], on HFS problems with sequence dependent setup times. In [15], a GA with a permutation representation of the individuals, and many variants of the crossover operator, also performed well against several heuristics such as ant colonies, tabu search, simulated annealing, other GA’s and deterministic methods on...
HFS with sequence dependent setup times and machine eligibility. Note that, most of these methods and the ones reviewed in [33], [25] and [32], consider problems with makespan as the optimisation criterion, mainly.

A more recent investigation, [28], studied the performance of the generic combination GA and a constructive procedure. The constructive procedure is one of four. Each of these combinations, called Single Stage Representation Genetic Algorithm (SSRGA), was applied to 4 variants of the HFS problem, each in turn having a different objective function. Although the SSRGA variants performed well, there is some variance between the performances. This means that it is not easy to tell which SSRGA will be most suitable for a given problem before carrying out the actual solution.

In this chapter the idea of SSRGA was extended by allowing GA to evolve the heuristic or combination of heuristics to be used to schedule the stages posterior to the first one. The proposed approach uses GA to generate part of the solution to the original problem and also searches the space of heuristics for a combination of some or all of them to generate the remaining components of the solution. In this way, the problem of deciding which SSRGA to use on a specific problem is removed. Moreover, by allowing different heuristics to be used at different stages, the number of usable SSRGA’s increases given the different possible combinations of heuristics that can arise, each leading to a different SSRGA variant. The main virtue of the proposed approach is that specialisation occurs during the solution process and not before it. In other words, the specialisation is a function of the instance of the problem in hand rather than the general form of it. Moreover, this allows a high level of portability between related problems.

Heuristics that combine other heuristics are referred to as hyper-Heuristics (HH), [10]. A HH can be seen as a black box that takes as input a set of low level heuristics and a problem, [30]. At each decision point, the black box selects a low level heuristic and applies it. Note that in HH the high level heuristic acts, exclusively, on the space of low level heuristics, while the latter, are the ones that solve the original problem; see left part of Figure 1.

Hyper-heuristics are becoming popular because they are adaptable, effective on different problems of the same class, and yet relatively easy to implement. This is because most of the effort in implementing them goes into the low level heuristics which themselves are usually easy to implement. Because HH’s are adaptable, they have been successfully applied to many problems including personnel scheduling problems such as the sales summit scheduling problem, [10], [11], the project presentation scheduling problem, [13], the nurse scheduling problem, [12], and others, [8], [7]. They have also been applied in manufacturing environments such as job shop, [20], and open shop, [14], scheduling problems and in industrial cutting stock problems, [31]. Meta-heuristics have been adopted as the basis (high level heuristic) of HH. Tabu search was used in [3] and [6]; ant colonies in [4], [5]; and GA’s in [14], [20], [9], [19], [17], [18] and [31].