European Journal of Operational Research 000 (2014) 1–10

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Job-shop local-search move evaluation without direct consideration of the criterion's value

Boštjan Murovec*

University of Ljubljana, Faculty of Electrical Engineering, Tržaška cesta 25, SI-1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 21 June 2013
Accepted 26 August 2014
Available online xxx

Keywords:
Move evaluation
Local search
Job-shop
Makespan
Scheduling

ABSTRACT

This article focuses on the evaluation of moves for the local search of the job-shop problem with the makespan criterion. We reason that the omnipresent ranking of moves according to their resulting value of a criterion function makes the local search unnecessarily myopic. Consequently, we introduce an alternative evaluation that relies on a surrogate quantity of the move's *potential*, which is related to, but not strongly coupled with, the bare criterion. The approach is confirmed by empirical tests, where the proposed evaluator delivers a new upper bound on the well-known benchmark test yn2. The line of the argumentation also shows that by sacrificing accuracy the established makespan estimators unintentionally *improve on* the move evaluation in comparison to the exact makespan calculation, in contrast to the belief that the reliance on estimation degrades the optimization results.

1. Introduction

Local search (Vaessens, Aarts, & Lenstra, 1996) is a popular method for tackling the deterministic job-shop scheduling problem with the makespan criterion (Jain & Meeran, 1999). The strength of the approach depends on the properties of the chosen neighborhood (Mattfeld, 1996), which is usually either limited to *swap* moves (Laarhoven, Aarts, & Lenstra, 1992; Nowicki & Smutnicki, 1996, 2005; Watson, Howe, & Whitley, 2006; Beck, Feng, & Watson, 2011) or consists of *insertion* moves (Dell'Amico & Trubian, 1993; Balas & Vazacopoulos, 1998; Murovec & Suhel, 2004; Fernandes & Lourenco, 2008; Zhang, Li, Rao, & Guan, 2008b; Li, Pan, & Xie, 2010). Insertion neighborhoods in job-shop scheduling are discussed in Mattfeld (1996) and Grabowski and Wodecki (2005), whereas the swap neighborhood of Nowicki and Smutnicki (1996) is analyzed in Jain, Rangaswamy, and Meeran (2000); additional discussion is given in Watson et al. (2006). Jain et al. (2000) and Li, Pan, and Xie (2010) take into consideration several critical paths of a schedule.

Besides the choice of a neighborhood, the crucial issue of any local-search implementation is the selection of moves to be applied to a schedule in the course of the optimization. The omnipresent opinion is that the best way to evaluate the moves is to consider their resulting value of a criterion function. Criterion evaluation is time consuming and may account for the majority of the optimization time (Eikelder, Aarts, Verhoeven, & Aarts, 1997), which motivated Nowicki and Smutnicki (2005) to develop a method for an exact makespan determination without any recalculation of heads, whereas Mati, Dauzère-Pérès, and Lahlou (2011) further improve on this approach. So far, these methods are only applicable to swap neighborhoods.

The use of criterion *estimators* (Taillard, 1994; Dell'Amico & Trubian, 1993; Balas & Vazacopoulos, 1998; Grabowski & Wodecki, 2005; Braune, Zäpfel, & Affenzeller, 2013) is another technique for boosting performance, although estimators are considered a nuisance, and algorithm designers prefer to avoid them whenever feasible. Nowicki and Smutnicki (1996, 2005) preserve a reliance on an exact makespan calculation due to the low cardinality of their neighborhood. Eikelder, Aarts, Verhoeven, and Aarts (1997) regard the loss of accuracy due to the use of an estimation as a drawback. Jain, Rangaswamy, and Meeran (1998) combine an estimation with an exact calculation in order to alleviate the consequences of the loss of accuracy, despite their empirical tests that show a slight *improvement* of the estimation-obtained results in comparison to the use of an exact calculation. Similarly, Zhang, Shao, Rao, and Qiu (2008a) conclude that empirical tests fail to confirm the superiority of the exact evaluation in comparison to the estimation.

Despite the mentioned empirical hints, the belief in the superiority of an exact-criterion move evaluation remains firmly in place. In contrast, we reason that estimators in the role of move evaluators can better guide the local search into promising areas of the solution space than the exact criterion calculation does, since the former are capable of detecting certain prospects of moves, which are invisible to the latter. The situation resembles a chess game, where it is crucial not only to make moves that capture your opponent's pieces, but

* Tel.: +386 1 4768 265.
E-mail address: bostjan.murovec@fe.uni-lj.si

http://dx.doi.org/10.1016/j.ejor.2014.08.044
0377-2217/

longer-term merits also need to be taken good care of as well. The myopia is the main weakness of local search and makes optimization highly susceptible to entrap in dense local minima (Mattfeld, 1996). Resorting to an exact criterion for the move evaluation makes the situation *as worse as possible*, since in this way really *only the immediate* benefits of moves are taken into account.

On the other hand, the estimators are unintentionally capable of spotting at least some of the non-immediate benefits of the moves, due to the fact that they examine only a confined part of the schedule around a perturbation. Consequently, the beneficiary structural changes to a schedule have a chance of being detected, even when the isolated application of a move does not improve the criterion's value by itself. Hence, we postulate that estimators reduce the myopic nature of a local search, at least to some extent (Section 2).

It is possible to develop the idea further and disregard the criterion's value (exact or estimated) in favor of a move evaluation according to a surrogate value of the move's *potential*, by means of which the focus of the evaluation is shifted away from a criterion delivery to a genuine detection of the beneficiary properties of moves (Section 3). Such an evaluation is expected to better guide a local search into promising regions of the solution space, which is confirmed with empirical tests (Section 4) where our evaluator delivers a new upper bound on the well-known benchmark test yn2.

2. Reducing local-search myopia with estimators

Estimators have the ability to reduce the myopia of a local search in comparison to an exact criterion evaluation. Let us explain this idea with the aid of Fig. 1, which presents three schedules of a sample job-shop instance. The jobs that the operations belong to are denoted by letters from a to d, whereas the operations' processing sequences within their respective jobs are prescribed by the numbers following the letters. The operations on the selected critical path are boldfaced.

The makespan of the top schedule equals ten, which is the length of the critical path through the operations a1, a2, b1 and c2. For this schedule there is no swap or insertion move that could decrease the makespan by itself due to the existence of another critical path through the operations d1, d2, c1 and c2. The schedule is entrapped in a local minimum in the sense that the exact makespan in the role of a move evaluator cannot declare any move as being potentially improving. In order to reduce the makespan, the length of *all* the critical paths in the schedule must be reduced, which, in this case, cannot be done by *any* isolated move in *any* classically defined neighborhood.

On the other hand, the estimation of Taillard (1994) evaluates a swap move according to the maximum resulting path lengths through the swapped operations, whereas the rest of the schedule is ignored. The stated ignorance is precisely the *beneficiary* property that *enhances* the evaluation's insight. Due to this property, the swap of the operations a2 and b1 in the top schedule of Fig. 1 is correctly recognized as potentially improving, since it reduces the lengths of the paths through these two critical operations.

The middle schedule in Fig. 1 shows the result of applying the discussed move to the initial schedule. The resulting makespan is still ten. Nonetheless, the new longest path through the operation b1 passes through the nodes b1, a2 and c2, and weights seven. Similarly, the new longest path through a2 goes through the nodes a1, a2 and c2, and weights eight. The result of the estimation is the longest of the two lengths (eight), which is less than the initial makespan (ten), by means of which the estimator is able to declare the move as potentially improving.

The new schedule possesses only one critical path, so now the estimation as well as the exact calculation is able to recognize the swap of operations d2 and c1 as potentially improving. The result of its application is shown at the bottom of Fig. 1, where the resulting makespan equals eight.

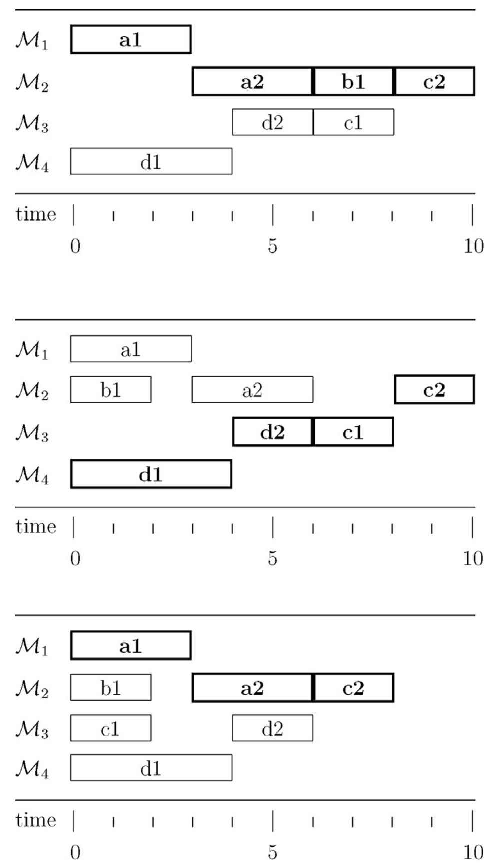

Fig. 1. Demonstration of the non-immediate usefulness of a move.

In the discussed example, the estimation of Taillard (1994) is able to directly guide the plain steepest-descent optimization from the initial to the final schedule. In contrast, the exact makespan calculation is entrapped in the local minimum and requires a metaheuristic mechanism to escape it. Hence, it is reasonable to conclude that the estimator of Taillard (1994) strengthens the intensification phase of the optimization by being able to remove a certain number of local minima from a solution's landscape because of its (unintentionally) less myopic move evaluation.

Similar examples can be constructed for other neighborhoods and estimators. Specifically, the estimator *lpath* of Dell'Amico and Trubian (1993), which extends the idea of Taillard (1994) to also cover insertion moves, is worth discussing. This estimator is still in the spotlight, despite being fairly old, since its results are as accurate as any achievable with a computational complexity of $O(n)$, where n is the number of operations with a changed machine processing order. Even the present job-shop set ups frequently still rely on the original idea of *lpath*, although certain modifications may be incorporated. For example, Braune et al. (2013) deal with the issue of a lower bound, by means of which the procedure *lpath* can be brought closer to the exact criterion calculation (by also increasing the computational demand).

The persistent attractiveness of the procedure *lpath* is primarily due to a ubiquitous belief in the superiority of move evaluation on the basis of a resulting criterion's value. This procedure achieves a fairly good accuracy (Balas & Vazacopoulos, 1998) with a modest computational demand. Lowering the complexity of the estimator results in a significant loss of accuracy, whereas an increase in the accuracy also noticeably raises the computational demand. Hence, the underlying idea of the procedure *lpath* will continue to be attractive as long as the ranking of moves is based on their criterion's value.