Problem difficulty for tabu search in job-shop scheduling ☆

Jean-Paul Watson [a,*,1], J. Christopher Beck [b,2], Adele E. Howe [a,1], L. Darrell Whitley [a,1]

[a] *Department of Computer Science, Colorado State University, Fort Collins, CO 80523-1873, USA*
[b] *Cork Constraint Computation Centre, University College Cork, Cork, Ireland*

Received 12 February 2002

Abstract

Tabu search algorithms are among the most effective approaches for solving the job-shop scheduling problem (JSP). Yet, we have little understanding of why these algorithms work so well, and under what conditions. We develop a model of problem difficulty for tabu search in the JSP, borrowing from similar models developed for SAT and other NP-complete problems. We show that the mean distance between random local optima and the nearest optimal solution is highly correlated with the cost of locating optimal solutions to typical, random JSPs. Additionally, this model accounts for the cost of locating sub-optimal solutions, and provides an explanation for differences in the relative difficulty of square versus rectangular JSPs. We also identify two important limitations of our model. First, model accuracy is inversely correlated with problem difficulty, and is exceptionally poor for rare, very high-cost problem instances. Second, the model is significantly less accurate for structured, non-random JSPs. Our results are also likely to be useful in future research on difficulty models of local search in SAT, as local search cost in both SAT and the JSP is largely dictated by the same search space features. Similarly, our research represents the first attempt to quantitatively model the cost of tabu search for *any* NP-complete problem, and may possibly be leveraged in an effort to understand tabu search in problems other than job-shop scheduling.
© 2002 Elsevier Science B.V. All rights reserved.

1. Introduction

The job-shop scheduling problem (JSP) is widely acknowledged as one of the most difficult NP-complete problems encountered in practice. Nearly all well-known optimization and approximation techniques have been applied to the JSP, including linear programming, Lagrangian relaxation, branch-and-bound, constraint satisfaction, local search, and even neural networks and expert systems [19]. Most recent comparative studies of techniques for solving the JSP conclude that local search algorithms provide the best overall performance on the set of widely-available benchmark problems; for example, see the recent surveys by Błażewicz et al. [6] or Jain and Meeran [19]. Within the class of local search algorithms, the strongest performers are typically derivatives of tabu search [6,19,34], the sole exception being the guided local search algorithm of Balas and Vazacopoulos [2]. The power of tabu search for the JSP is perhaps best illustrated by considering the computational effort required to locate optimal solutions to a notoriously difficult benchmark problem, Fisher and Thompson's infamous 10×10 instance [12]: Nowicki and Smutnicki's algorithm [22] requires only 30 seconds on a now-dated personal computer, while Chambers and Barnes' algorithm [9] requires less than 4 seconds on a moderately powerful workstation. In contrast, a number of algorithms for the JSP still have significant difficulty in finding optimal solutions to this problem instance.

Despite the relative simplicity and excellent performance of tabu search algorithms for the JSP, very little is known about *why* these algorithms work so well, and under what conditions. For example, we currently have no answers to fundamental, related questions such as "Why is one problem instance more difficult than another?" and "What features of the search space influence search cost?". No published research has presented problem difficulty models of tabu search algorithms for the JSP. Further, only one group of researchers, Mattfeld et al. [20], have analyzed the link between problem difficulty and local search for the JSP in general.

In contrast to the JSP, problem difficulty models do exist for several other well-known NP-complete problems, such as the Traveling Salesman Problem (TSP) and the Boolean Satisfiability Problem (SAT). Models of local search cost in SAT have received significant recent attention, and are able to account for much of the variability in problem difficulty observed for a particular class of random problem instances commonly known as Random 3-SAT [10,24,27]. The SAT models relate individual features of the search space to search cost, and model accuracy is generally quantified as the r^2 value of the corresponding linear regression model. We refer to such models as *static cost models* of local search; the goal of such models is to account for a significant proportion (ideally all) of the variability in local search cost observed for a set of problem instances. The 'static' modifier derives from the fact these models are largely independent of particular algorithm dynamics, relying instead on static features of the search space.

Static cost models of local search in SAT are based on three search space features: the number of optimal solutions, the backbone size, and the mean distance between random