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*

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 solutions 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 when considering structured, non-random JSPs. Our results are likely to be useful in future research on models of problem difficulty for local search in SAT, as local search cost in both SAT and the JSP is largely dictated by the same features of the search space. 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 domains other than job-shop scheduling.

Key words: Problem Difficulty, Job-Shop Scheduling, Local Search, Tabu Search

* Corresponding author. E-mail: watsonj@cs.colostate.edu

[1] The authors from Colorado State University were sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-00-1-0144. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

[2] This work was performed while the second author was employed at ILOG, SA.

1 Introduction

The job-shop scheduling problem (JSP) is widely acknowledged as one of the most difficult NP-complete problems encountered in practice. Nearly every well-known optimization or approximation technique has been applied to the JSP, including linear programming, Lagrangian relaxation, branch-and-bound, constraint satisfaction, local search, and even neural networks and expert systems [1]. Most recent comparative studies of algorithms for solving the JSP conclude that local search algorithms provide the best overall performance on the set of widely-used benchmark problems; for example, see the recent surveys by Blażewicz et al. [2] or Jain and Meeran [1]. Within the broad class of local search algorithms, the strongest performers are typically derivatives of tabu search [2] [1] [3], the sole exception being the guided local search algorithm of Balas and Vazacopoulos [4]. The power of tabu search for the JSP is perhaps best illustrated by Nowicki and Smutnicki's algorithm [5], which is capable of solving a notoriously difficult benchmark problem, Fisher and Thompson's infamous 10×10 instance [6], in less than a minute on now-dated hardware. 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 superior 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 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 models of problem difficulty for tabu search algorithms for the JSP. Further, only one group of researchers, Mattfeld et al. [7], has analyzed the link between problem difficulty and local search for the JSP in general.

In contrast, models of problem difficulty for local search do exist for many other well-known NP-complete problems. The majority of these models consider the Boolean Satisfiability Problem (SAT), and the most recent models account for much of the variance in local search cost observed in a particular class of random problem instances commonly known as Random 3-SAT [8] [9] [10]. All of these models are based on particular features of the search space, and hypothesize that a particular feature, or set of features, is largely responsible for the cost required by a local search algorithm to locate an optimal solution to a problem instance. The hypothesis is generally tested via linear or multiple regression methods, with the regression r^2 value quantifying the accuracy of the resulting model. We refer to models developed using this methodology as *descriptive cost models* of local search; the goal of such models is to account for a significant proportion (ideally all) of the variance in local search cost observed for a set of problem instances.

Descriptive 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 solutions and the nearest optimal solution. Clark et al. [8] introduced the first descriptive cost model for SAT, and demonstrated that the logarithm of the number of optimal (i.e., satisfying) solutions accounts for a significant proportion of the variability in local search cost. In SAT, the *backbone* of a problem instance is the set of Boolean variables that have the same truth value in all optimal solutions. Both Parkes [9] and Singer et al. [10] demonstrated that the size of the problem backbone is positively correlated with the cost of local search in SAT. More recently, Singer et al. [10] demonstrated that local search cost in SAT is positively correlated with the mean distance between random solutions and the nearest optimal solution.

Descriptive cost models for problems other than SAT have received relatively little attention, the sole exception being the related and more general Constraint Satisfaction Problem (CSP) [8]. However, the factors underlying the descriptive cost models for SAT are very intuitive, and this intuition extends beyond SAT to many other NP-complete problems, including the JSP; for example, most researchers would be surprised if the number of optimal solutions did *not* influence local search cost. However, the search spaces of the JSP and SAT are qualitatively dissimilar, and local search algorithms for SAT differ in many ways from tabu search algorithms for the JSP. For example, local search algorithms for SAT have a strong stochastic component, while tabu search algorithms for the JSP are much more deterministic. Consequently, it is unclear a priori whether the descriptive cost models for SAT can be leveraged in an effort to understand problem difficulty for tabu search in the JSP.

In this paper, we develop a descriptive cost model of tabu search in the JSP, drawing heavily from the existing descriptive cost models of local search in SAT. The resulting model accounts for a significant proportion of the variance in the cost of finding optimal solutions to random JSPs. We then use the model to explain two well-known but poorly-understood qualitative observations regarding problem difficulty in the JSP, and identify two important limitations of the model. More specifically, our research makes the following contributions:

(1) We show that the search space features known to influence the cost of local search in SAT, specifically the number of optimal solutions ($|optsols|$) and the mean distance between random solutions and the nearest optimal solution ($d_{lopt\text{-}opt}$), also influence the cost of locating optimal solutions using tabu search in the JSP. Further, the *strength* of the influence of these two factors is nearly identical in both problems. As in SAT, we find that $d_{lopt\text{-}opt}$ has a much stronger influence than $|optsols|$ on search cost in the JSP, and ultimately accounts for a significant proportion of the variance in search cost observed for a set of identically-sized problem instances. This result was somewhat unexpected given the differences between the search spaces and local search algorithms of the JSP and SAT.

(2) Our experiments indicate that for JSPs with moderate to large backbones, the correlation between backbone size and the number of optimal solutions is

extremely high. As a direct consequence, for these problems, backbone size provides no more information than the number of optimal solutions, and vice versa: one of the two factors is necessarily redundant. Given the recent surge of interest in the link between backbone size and problem difficulty, the strong one-to-one correspondence between these two factors was completely unanticipated.

(3) In contrast to Singer et al. [10], we find *no* interaction effect between the backbone size and $d_{lopt\text{-}opt}$. Further, we find that descriptive cost models based on either multiple factors or interacting factors are no more accurate that the simple model based solely on $d_{lopt\text{-}opt}$.

(4) A simple extension of the $d_{lopt\text{-}opt}$ descriptive cost model accounts for most of the variance in the cost of finding *sub-optimal* solutions to the JSP. This extension is the first quantitative model of the cost of locating sub-optimal solutions to any NP-complete problem, and provides an explanation for the existence of 'cliffs' in the cost of finding sub-optimal solutions of varying quality [11].

(5) For some time, researchers have observed that 'square' JSPs are generally more difficult than 'rectangular' JSPs. We show that this phenomenon is likely due to differences in the distribution of $d_{lopt\text{-}opt}$ for the two problem types. For square JSPs, the proportion of problem instances with large values of $d_{lopt\text{-}opt}$ is substantial, while most instances of rectangular JSPs have very small values of $d_{lopt\text{-}opt}$.

(6) We identify two important limitations of the $d_{lopt\text{-}opt}$ model. First, we show that model accuracy is inversely proportional to problem difficulty, and is exceptionally poor for very high-cost problem instances. Second, we demonstrate that the $d_{lopt\text{-}opt}$ model is significantly less accurate when we consider more structured JSPs, specifically those containing workflow partitions.

Because local search cost in both the JSP and SAT is influenced by the same search space features, our results for the JSP also identify likely deficiencies in the descriptive cost models of local search in SAT. Specifically, we conjecture that the following also hold in SAT: (1) the tight correspondence between backbone size and the number of optimal solutions, (2) the extreme inaccuracies of the $d_{lopt\text{-}opt}$ model on very high-cost problem instances, and (3) the degradation in the accuracy of the $d_{lopt\text{-}opt}$ model for structured problem instances.

Although tabu search algorithms have been successfully applied to a number of NP-complete problems, very little is known in general about which search space features influence problem difficulty, and to what degree. Our research provides a preliminary answer to this question for one particular problem, the JSP, and only for a relatively simple form of tabu search. Consequently, our results may be useful to researchers developing models of problem difficulty for tabu search in NP-complete problems other than the JSP, or for models of more advanced tabu search algorithms for the JSP.

In the following section, we briefly review existing models of problem difficulty and identify the sub-set of models that form the basis of our analysis. In Section 3, we define the job-shop scheduling problem and introduce the tabu search algorithm used in our experiments; the section concludes with a discussion of prior work on problem difficulty for local search in the JSP. In Section 4, we develop a model of the cost required by tabu search to find optimal solutions to the JSP. Section 5 details two important applications of the resulting model: accounting for the cost of locating *sub-optimal* solutions to the JSP, and providing an explanation for differences in the relative difficulty of square versus rectangular JSPs. In Section 6, we expose two important limitations of the model, both of which suggest new directions in research on models of problem difficulty. Finally, we conclude by discussing the implications of our results in Section 7.

2 Models of problem difficulty

One key lesson from the research into models of problem difficulty is the often 'universal' nature of these models, in that they typically apply to a wide range of NP-complete problems. For example, phase transitions have been observed in problems ranging SAT to Graph K-colorability [12]. Similarly, the distribution of local optima in many problems exhibits a 'Big-Valley' structure, for example in both the Traveling Salesman and Graph Bi-Partitioning Problems [13]. Given the pervasiveness of these phenomena, the obvious first step in our research is to determine whether existing models of problem difficulty can be extended to the JSP. However, before investigating particular models of problem difficulty, we first consider the following questions:

- What type of information should our model provide?
- What existing models provide this type of information?
- What is our success criteria?

Our immediate goal is to develop an understanding of existing tabu search algorithms for the JSP; subsequently, we intend to leverage such knowledge to improve the performance of existing algorithms. A detailed understanding of how an algorithm interacts with the search space is clearly required to propose enhancements in a principled manner. Consequently, our goal is to produce quantitative models that relate search space features to search cost; we refer to such models as *descriptive cost models*. Good descriptive cost models account for a significant proportion of the variance in search cost observed for a set of problem instances.

Most models of problem difficulty do not share our goal of relating search space features to search cost, and as a consequence generally fail to account for any significant proportion of the variability in search cost. Within the AI community, phase transitions [12] are the dominant model of problem difficulty. Phase transition models partition the 'universe' of problem instances into a large number of sub-classes, and are able to account for mean differences in sub-class difficulty. However, the variance in problem difficulty within a sub-class is unaccounted for, and is typically largest in the most difficult sub-classes (i.e., those near the transition region). For example, the local search cost in the most difficult sub-class of 100-variable Random 3-SAT instances varies over 5 orders of magnitude [8]. Outside of AI, the most widely studied models of problem difficulty are correlation length and the Big-Valley local optima distribution. A correlation length model measures the 'smoothness' of a search space by analyzing the autocorrelation of the time-series of solution quality produced by a random walk. However, the correlation length for a large number of problems is *strictly* a function of the problem size (e.g., the number of cities in the Traveling Salesman Problem) [14]. Consequently, correlation length models fail to account for *any* of the often large variance in problem difficulty observed for an ensemble of fixed-size problem instances. Similarly, Big-Valley local optima distributions are found in both easy and hard problems; the model generally fails to account for the relative difficulty of individual problem instances.

To date, researchers have only produced descriptive cost models of local search in SAT and the closely related Constraint Satisfaction Problem. Descriptive 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 solutions and the nearest optimal solution. In Section 4, we define each of these features, discuss the properties of existing descriptive cost models that are based on these features, and analyze the applicability of these features to descriptive cost models of tabu search in the JSP.

Because our descriptive cost models are either linear or multiple regression models (see Section 4), model accuracy is naturally quantified by the r^2 value of the regression model [15]. We can also quantify worst-case model accuracy by analyzing the magnitude of the residuals under the regression model. Ultimately, our goal is to produce a descriptive cost model with (1) $r^2 \geq 0.8$ and (2) the actual search cost varying no more than $1/2$ an order of magnitude from the predicted search cost. Although somewhat arbitrary, any descriptive cost model satisfying these two criteria would conclusively identify those search space features that largely dictate the cost of tabu search in the JSP. Further, more stringent criteria are likely to leave insufficient room for measurement error. Finally, as we discuss in Section 4, the task of producing descriptive cost models satisfying the two proposed criteria is sufficiently challenging.

3 The job-shop scheduling problem, local search, and problem difficulty

We now introduce the job-shop scheduling problem and detail the specific tabu search algorithm that forms the basis of our analysis. We then briefly review prior research on problem difficulty and job-shop scheduling.

3.1 The job-shop scheduling problem

We consider the well-known $n \times m$ static job-shop scheduling problem (JSP), in which n jobs must be processed exactly once on each of m machines. Each job i ($1 \leq i \leq n$) is routed through the m machines in some pre-defined order π_i, where $\pi_i(j)$ denotes the jth machine ($1 \leq j \leq m$) in the routing order. The processing of a job on a machine is called an *operation*, and the processing of job i on machine $\pi_i(j)$ is denoted by o_{ij}. An operation o_{ij} must be processed on machine $\pi_i(j)$ for an integral duration of $\tau_{ij} > 0$. Once processing is initiated, an operation cannot be pre-empted, and concurrency is not allowed. Finally, for $2 \leq j \leq m$, o_{ij} cannot begin processing until o_{ij-1} has completed processing.

A solution s to an instance of the $n \times m$ JSP specifies a processing order for all of the jobs on each machine, and implicitly specifies an earliest start time $est(x)$ and earliest completion time $ect(x)$ for each operation x [16]. Although a number of objective functions have been defined for the JSP, most research considers the problem of makespan minimization [2]. The *makespan* $C_{max}(s)$ of a solution s is the maximum earliest completion time of the last operation of any job: $C_{max}(s) = max(ect(o_{1,m}), ect(o_{2,m}), ..., ect(o_{n,m}))$. We denote the optimal makespan of a problem instance by C^*_{max}. The decision problem of finding a solution to the JSP with a makespan less than or equal to some constant L is known to be NP-complete for $m \geq 2$ and $n \geq 3$ [17]. Further, the JSP is widely regarded as one of the most difficult NP-complete problems encountered in practice [2] [1].

As discussed in Section 3.3, extraction and manipulation of the critical paths of a solution s is a key component of tabu search algorithms for the makespan minimization form of the JSP. A *critical path* of a solution s consists of a sequence of operations $o_1, o_2, ..., o_l$ such that (1) $est(o_1) = 0$, (2) $ect(o_l) = C_{max}(s)$, and (3) $est(o_i) = ect(o_{i-1})$ for $1 \leq i \leq l$, where $ect(o_0) = 0$ by convention. The operations o_i are known as *critical operations*. A *critical block* consists of a contiguous sub-sequence of operations on a critical path that are processed on the same machine. A solution s may possess more than one critical path. If multiple critical paths exist, they may share common sub-sequences of critical operations.

3.2 Generating problem instances

An instance of the $n \times m$ JSP is uniquely defined by the set of nm operation durations τ_{ij} and the n job routing orders π_i ($1 \leq i \leq n$ and $1 \leq j \leq m$). Typically, the τ_{ij} are independently and uniformly sampled from a fixed-width interval, typically $[1, 99]$ (e.g., see Taillard [18]). Most often, the job routing orders π_i are produced by generating independent random permutations of the integers $[1..m]$. We refer to problem instances in which both the τ_{ij} and π_i are independently and uniformly sampled as *general* JSPs.

Well-known specializations of the JSP impose non-random structure on the job routing orders. One such specialization we consider in Section 6.2 are JSPs with *workflow*. In workflow JSPs, the set of machines is typically divided into two equal-sized partitions containing machines 1 through $m/2$ and $m/2+1$ through m, respectively, and every job must be processed on all machines in the first partition before any machine in the second partition. Within the partitions, the job routing orders are produced by generating independent random permutations of the integers $[1..m/2]$ and $[m/2+1..m]$, respectively.

3.3 Algorithm description

The analyses in Sections 4 through 6 are based on a tabu search algorithm for the JSP introduced by Taillard [16], which we denote *TS*$_{Taillard}$. We note that *TS*$_{Taillard}$ is *not* the best available tabu search algorithm for the JSP; both the tabu algorithms by Nowicki and Smutnicki [5] and Dell'Amico and Trubian [19] provide stronger overall performance. Rather, we have selected *TS*$_{Taillard}$ for three reasons. First, *TS*$_{Taillard}$ provides reasonable performance of the set of widely-used benchmark problems, and out-performs many other local and constructive search algorithms for the JSP. Second, high-performance tabu search algorithms are generally much more complex than *TS*$_{Taillard}$, complicating analysis. Instead of tackling the most complex algorithms first, our goal is to develop a descriptive cost model for a straightforward implementation of tabu search in the JSP, and then to *systematically* assess the influence of more complex algorithmic features on the descriptive cost model of the basic algorithm. Third, as we now discuss, certain features of *TS*$_{Taillard}$ make it particularly amenable to analysis, especially in comparison to more advanced tabu search algorithms for the JSP.

At the core of any local search algorithm is a move operator, which defines the set of solutions that can be reached from the current solution; elements of this set are called *neighbors* of the current solution. In the JSP, neighbors are generally produced by re-ordering the sequence of operations on a critical path; only through such re-ordering is it possible to produce a neighbor with a makespan better than that of the current solution [20]. The first move operator for the JSP was introduced by van Laarhoven et al. [20], and is often denoted by $N1$. The neighborhood of a solution s under the $N1$ move operator is generated by swapping the order of all distinct pairs of adjacent operations on the same critical block in s. An important property of $N1$ is that it induces search spaces that are provably *connected*, in that it is always possible to move from an arbitrary solution to a global optimum. Consequently, it is possible to construct a local search algorithm based on $N1$ that will eventually locate an optimal solution, given a sufficiently large run-time. Hoos [21] refers to algorithms with this property as being *probabilistically approximately complete*, or PAC; the probability of the algorithm locating an optimal solution approaches 1 as the run-time approaches ∞. A primary reason we consider $\mathit{TS}_{\mathit{Taillard}}$ in our analysis is that it is based on the $N1$ operator. Further, $\mathit{TS}_{\mathit{Taillard}}$ is, at least empirically, PAC: in producing the results discussed in Sections 4 through 6, no trial of $\mathit{TS}_{\mathit{Taillard}}$ failed to locate an optimal solution. In contrast, many of the more advanced tabu search algorithms for the JSP use move operators that induce disconnected search spaces, and are consequently not PAC: e.g., Nowicki and Smutnicki's algorithm. Our primary goal is to model the cost of locating optimal solutions to the JSP, and as we discuss later in this section, the measurement of this cost is straightforward only if an algorithm is PAC.

The descriptive cost models we consider in Section 4 are based in part on search space features that involve distances between pairs of solutions; for example, the average distance between local optima or the mean distance between random solutions and the nearest optimal solution. Ideally, the distance between two solutions is defined as the minimum number of applications of a particular move operator that are required to transform one solution into the other. Unfortunately, computation of this measure is generally intractable, and operator-independent measures are typically substituted. The most widely used operator-independent distance measure in the JSP is defined as follows [7]. Let $\mathit{preceeds}_{i,j,k}(s)$ be a Boolean-valued function indicating whether job i is processed before job j on machine k in a solution s. The distance $D(s_1, s_2)$ between two solutions s_1 and s_2 to an $n \times m$ JSP instance is then given by:

$$\sum_{i=1}^{m} \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} \mathit{preceeds}_{i,j,k}(s_1) \oplus \mathit{preceeds}_{i,j,k}(s_2) \tag{1}$$

where the symbol $\oplus$ denotes the Boolean XOR operator. We denote the normalized distance $2D(s_1, s_2)/mn(n-1)$ by $\overline{D}(s_1, s_2)$; clearly, $0 \leq \overline{D}(s_1, s_2) \leq 1$. Another important property of the $N1$ operator is the fact that Equation 1 provides a relatively tight lower bound on the number of applications of the $N1$ move operator to transform solution s_1 into solution s_2. In contrast, similar lower bounds or even estimates for more advanced JSP move operators, including that used by Dell'Amico and Trubian's algorithm, are currently unknown.

$TS_{Taillard}$ is a relatively 'vanilla' implementation of tabu search [22]. As with most tabu search algorithms for the JSP, recently swapped pairs of jobs are prevented from being re-established for a particular duration, called the tabu tenure. In each *iteration* of $TS_{Taillard}$, all $N1$ neighbors are generated. The neighbors are then classified as tabu (the pair of jobs was recently swapped) or non-tabu, and the best non-tabu move is taken; ties are broken randomly. All runs are initiated from randomly generated local optima, produced using a standard steepest-descent algorithm initiated from a random 'semi-active' solution [16]. The only long-term memory mechanism is a simple aspiration criterion, which over-rides the tabu status of any move that results in a solution that is better than any encountered during the current run. As indicated by Taillard ([16], p. 100), long-term memory is only necessary for problems that require a very large (> 1 million) number of iterations, which is not the case for the test problems considered in our analysis. The only parameters of $TS_{Taillard}$ involve computation of the tabu tenure, which is uniformly sampled from the interval $[6, 14]$ every 15 iterations. Empirically, $TS_{Taillard}$ fails to be PAC without such a dynamic tabu tenure, or if the tabu tenure is sampled from a smaller interval (e.g., $[5, 10]$).

The cost required to solve a given problem instance using $TS_{Taillard}$, or any PAC algorithm, is naturally defined as the number of iterations required to locate an optimal solution. However, the number of iterations is stochastic (with an approximately exponential distribution [16]), due to both the randomly generated initial solution and random tie-breaking when more than one 'best' non-tabu move is available. Consequently, we define the local search cost for a problem instance as the median number of iterations required to locate an optimal solution over 5000 independent runs, which we denote $cost_{med}$. This contrasts with SAT, in which researchers have reported that only 1000 samples are required to produce relatively stable estimates of the median [21] [10].

3.4 Prior research on problem difficulty in the JSP

A number of qualitative observations regarding the relative difficulty of various types of JSPs have emerged over time, from a wide variety of sources [1]:

(1) For both general and workflow JSPs, 'square' ($n/m \approx 1$) problems are generally more difficult than 'rectangular' ($n/m \gg 1$) problems.
(2) Given fixed n and m, workflow JSPs are generally more difficult than general JSPs.
(3) Given fixed n and m, relative problem difficulty is largely algorithm independent.

Clearly, any such model needs at least to be consistent with, and should ultimately provide explanations for, each of these three observations.

Large differences in the difficulty of square versus rectangular JSPs are easily illustrated by considering the best-known makespans and lower bounds for the problems in Taillard's benchmark suite, which are available from the OR-Library [23]. Specifically, the optimal makespans of all the relatively small 20×20 and 30×20 instances are currently unknown, while optimality has been established for all but two of the larger 50×20 and 100×20 instances, although the search spaces are astronomically larger in the latter instances. Taillard [16] studied the impact of changes in the ratio of n/m on search cost for $TS_{Taillard}$. His experiments demonstrated that for $n/m \geq 6$, the growth in the cost of locating optimal solutions grows *polynomially* with increases in n and m, despite an exponential growth in the size of the search spaces. In contrast, for problems with $n/m \approx 1$, the search cost grows exponentially with increases in n and m, as expected given the proportionate growth in the size of the search space. Although intuitive explanations have been proposed for why the growth in problem difficulty changes with increases in n/m, a complete understanding of this phenomenon remains elusive. No research has analyzed the changes in search space features as n/m is varied, which is of particular interest when developing descriptive cost models of local search.

The second observation stems largely from computational experiments on two sets of 50×10 general and workflow JSPs introduced by Storer et al. [24]: a large number of algorithms for the JSP have more difficulty finding high-quality solutions to the instances with workflow. Further, the optimal makespans of all the general instances are known, while optimality has only been established for one of the workflow instances. The sole quantitative study of problem difficulty in the JSP is due to Mattfeld et al. [7], and is largely devoted to providing an explanation for the differences in relative difficulty of general and workflow JSPs. Mattfeld et al.'s study also provides a possible explanation for why genetic algorithms generally perform poorly on the JSP. Mattfeld et al. identified significant differences in the search spaces of Storer et al.'s general and workflow instances, specifically by demonstrating that the extension of the search space (as measured by the average distance between random local optima) is larger in workflow JSPs than in general JSPs. This suggests a cause for the relative differences in search cost between the two problem types. Similar differences were observed for two other quantitative search space measures: entropy and correlation length.

Finally, the third observation results from the fact that easy (difficult) benchmark problem instances are easy (difficult) for *all* search algorithms, including those based on branch-and-bound, constraint programming, and local search. A causal basis for this phenomenon is lacking, although we hypothesize an explanation for the class of local search algorithms in Section 4.5.

4 Modeling the cost of locating optimal solutions

We now introduce and analyze several models of the cost required by $TS_{Taillard}$ to find optimal solutions to general JSPs. Each model is based on a specific feature of the search space: the number of optimal solutions, the backbone size, the average distance between local optima, and mean distance between random solutions and the nearest optimal solution. Similar models have been considered in other contexts, primarily SAT, and we analyze their ability to model the cost of tabu search for general JSPs. We then consider models based on aggregations of these factors, specifically analyzing the impact of additive and interaction effects.

Because the models we adapt were originally developed for other NP-complete problems such as SAT, their applicability to the JSP is unclear a priori. For example, the SAT search space is dominated by plateaus of equally-fit quasi-solutions, each containing an identical, small number of unsatisfied clauses. The main challenge for local search is to either find an exit from a plateau to an improving quasi-solution, or to escape the plateau by accepting a short sequence of dis-improving moves [25]. In contrast, the JSP search space is dominated by local optima with variable-sized and variable-depth attractor basins, which local search algorithms must either escape or avoid. Further, local search algorithms for SAT are largely stochastic, while tabu search algorithms such as $TS_{Taillard}$ are largely deterministic. On the other hand, the features underlying these models are very intuitive, and would appear to influence the difficulty of local search in a wide range of NP-complete problems.

In this section, we demonstrate that despite strong differences in both search space topologies and local search mechanisms, adaptations of the SAT descriptive cost models do yield an accurate descriptive cost model of search cost in $TS_{Taillard}$ for the general JSP. Specifically, we show that (1) only the mean distance between random local optima and the nearest global optimum accounts for a substantial proportion of the variance in local search cost, (2) simultaneous consideration of other features, through the addition of either additive or interaction terms, does not enhance the accuracy of this model, and (3) the correlation between the number of optimal solutions and the backbone size is extremely high, with one factor providing no more information than the other.

The material presented in this section is a significant extension of an analysis we previously reported [26]. In our prior work, we directly replicated the methodology introduced by Singer et al. [10], and demonstrated that the SAT descriptive cost models also applied to $TS_{Taillard}$ for the general JSP. In this section, instead of controlling for various features a priori, we adopt a different methodology. Instead, we explicitly focus on model accuracy for ‘typical’ instances of the general JSP. The new methodology also enables our new insights concerning the existence of interactions between the various search space features.

Finally, as discussed in Section 2, we quantify the accuracy of each descriptive cost model using linear or multiple regression techniques. Unless otherwise noted, the assumptions concerning model errors (e.g., the errors are normally distributed and homogeneous) are approximately satisfied, and the F-statistics are significant at $p < 0.0001$. For instances where the regression assumptions are not satisfied, we additionally report the non-parametric Spearman's rank correlation coefficient.

4.1 Test problems

For a number of reasons, models of problem difficulty are generally produced by considering relatively small problem instances. We develop our descriptive cost models using 6×4 and 6×6 general JSPs; we selected these two groups because they represent rectangular and square JSPs, respectively (see Section 3.4). For both groups, we generated 1000 instances using the procedures discussed in Section 3.2. Three of the four models we consider require computation of *all* optimal solutions to a problem instance, which can number in the tens of millions for 6×4 and 6×6 general JSPs. Further, $cost_{med}$ for each problem instance is defined as the median search cost over 5000 independent runs of $TS_{Taillard}$, which requires considerable CPU time for even small JSPs. Consequently, extensive analysis of descriptive cost models on larger general JSPs is currently impractical. For each of the 2000 problem instances, we used an independent implementation of Beck and Fox's [27] constraint-directed scheduling algorithm to compute the optimal makespan, the backbone size, and the set of optimal solutions. Finally, we note that the distribution of $log_{10}(cost_{med})$ is approximately normal for both problem groups, with any deviation due to the presence of a few very high-cost problem instances.

4.2 The number of optimal solutions and search cost

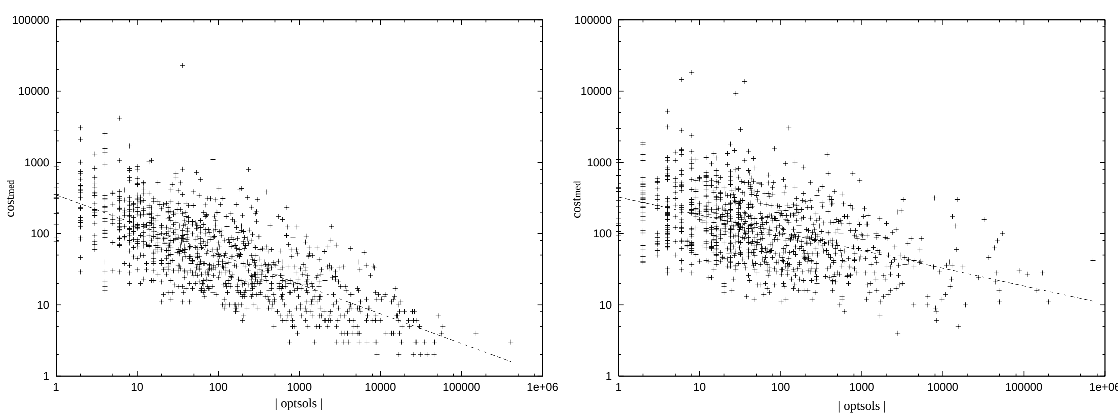

Fig. 1. Scatter-plots of $log_{10}(|optsols|)$ versus $log_{10}(cost_{med})$ for 6×4 (left figure) and 6×6 (right figure) general JSPs; the least-squares fit lines are super-imposed. The r^2 values for the corresponding regression models are 0.5307 and 0.2231, respectively.

The first descriptive cost model we consider is based on the number of optimal solutions to a problem instance, which we denote $|optsols|$. Intuitively, a decrease in the number of optimal solutions should yield an increase in local search cost. This observation formed the basis of the first descriptive cost model of local search in both SAT and CSP, first introduced by Clark et al. [8] (and later refined by Singer et al. [10]). Clark et al. demonstrated a relatively strong negative log_{10}-log_{10} correlation between the number of optimal solutions and search cost for three local search algorithms, with r-values ranging anywhere from -0.77 to -0.91. However, the model failed to account for the large cost variance observed for problems with small numbers of optimal solutions, where model residuals varied over three or more orders of magnitude. We have also observed very similar behavior for $TS_{Taillard}$ in the general JSP [26].

We show scatter-plots of $log_{10}(|optsols|)$ versus $log_{10}(cost_{med})$ for 6×4 and 6×6 general JSPs in Figure 1. The r^2 values for the corresponding regression models are 0.5365 and 0.2223, respectively. Although the model errors are clearly heterogeneous [8] [26], the results are consistent with the computed rank correlation coefficients (-0.7277 and -0.4661, respectively). In comparing the results for the 6×4 and 6×6 general JSPs, it is important to note the large difference in size of the search spaces: 2^{60} versus 2^{90}, respectively. Consequently, although the range of $log_{10}(|optsols|)$ is nearly identical in both cases, the relative number of optimal solutions is, on average, much smaller for the 6×6 general JSP. Given that the accuracy of the $|optsols|$ model is poor for instances with small numbers of optimal solutions, the discrepancy between the r^2 values of the 6×4 and 6×6 general JSPs can be explained by noting that the frequency of instances with relatively small numbers of optimal solutions is larger in square general JSPs [26].

The results presented in Figure 1 indicate that for typical general JSPs, a descriptive cost model based on $|optsols|$ is relatively inaccurate, accounting for roughly 50% of the variance in search cost in the *best* case. In the general JSP, as $n/m \to \infty$, the frequency of problem instances with a large number of optimal solutions increases. By extrapolation, we would then expect the accuracy of the $|optsols|$ model to increase as $n/m \to \infty$. In contrast, the accuracy of the model appears worst for the most difficult class of general JSP (i.e., those with $n/m \approx 1.0$), with model residuals varying over 2 to 3 orders of magnitude.

4.3 Backbone size and search cost

Recently, researchers have introduced several models of problem difficulty that are based on the concept of a backbone. Informally, the *backbone* of a problem instance is the set of solution attributes that have identical values in *all* optimal solutions to the instance. For example, in SAT the backbone is the set of Boolean variables that have identical values in all optimal (i.e., satisfying) assignments; in the TSP,

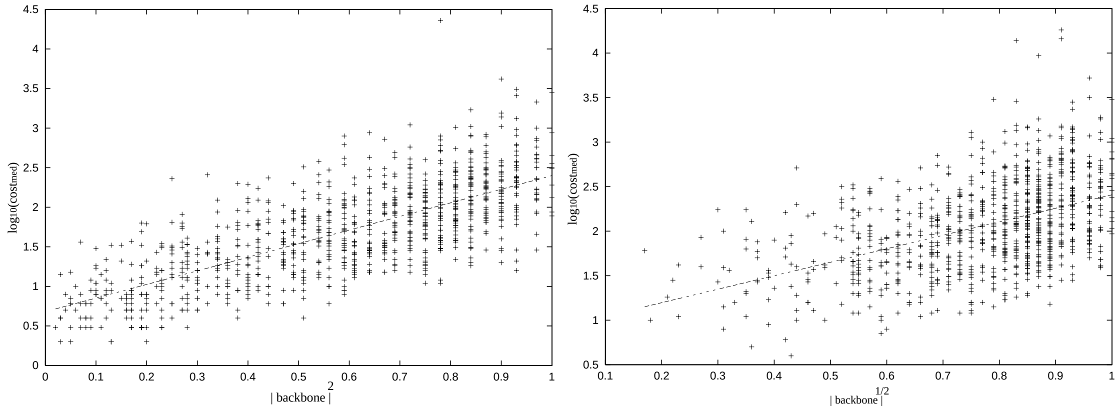

Fig. 2. Scatter-plots of $|backbone|^2$ versus $log_{10}(cost_{med})$ for 6×4 (left figure) and 6×6 (right figure) general JSPs; the least-squares fit lines are super-imposed. The r^2 values for the corresponding regression models are 0.5307 and 0.2231, respectively.

the backbone consists of the set of edges common to all optimal tours. The recent interest in backbones stems largely from the discovery that backbone size (as measured by the *fraction* of solution attributes appearing in the backbone) is correlated with search cost in SAT (e.g., see Monasson et al. [28]). Specifically, Parkes [9] showed that large-backboned SAT instances begin to appear in large quantities in the critical region of the phase transition (for a more detailed investigation into the relationship between backbone size and the SAT phase transition, see Singer et al. [10] or Singer [29]). Similarly, Achlioptas et al. [30] demonstrated a rapid transition from small to large-backboned instances in the phase transition region. While researchers have demonstrated a correlation between backbone size and problem difficulty in SAT, none of the papers cited directly measure the *strength* of this correlation.

Only Slaney and Walsh [31] have studied the influence of backbone size on search cost in problems other than SAT. Focusing on constructive search algorithms, they analyze the cost of both finding an optimal solution and proving optimality in a number of NP-complete problems, including the TSP and the number partitioning problem. For these two problems, Slaney and Walsh report a weak-to-moderate correlation between backbone size and the cost of finding an optimal solution (0.138 to 0.388). No studies to date have directly quantified the correlation between backbone size and problem difficulty for local search algorithms, and this relationship has only been qualitatively explored for SAT.

The definition of a backbone clearly depends on how solutions are represented. The most common solution encoding used in local search algorithms for the JSP, including $TS_{Taillard}$, is the *disjunctive graph* [32]. In the disjunctive graph representation, there are $n(n-1)/2$ Boolean 'order' variables for each of the m machines, each of which represents a precedence relation between a distinct pair of jobs on a machine. Consequently, we define the backbone of a JSP as the set of Boolean order variables that have the same value in all optimal solutions. We define the backbone size in the JSP as the fraction of the possible $mn(n-1)/2$ order variables that are fixed to the same value in all optimal solutions, which we denote by $|backbone|$.

Following Slaney and Walsh, our initial analysis considered the influence of $|backbone|$ on $log_{10}(cost_{med})$. We observed a significant quadratic component in the relationship, and the linear term in the quadratic regression model is statistically insignificant. We show scatter-plots of $|backbone|^2$ versus $log_{10}(cost_{med})$ for 6×4 and 6×6 general JSPs in Figure 2. The r^2 values for the corresponding regression models are 0.5307 and 0.2331, respectively. As with the $|optsols|$, the model errors are heterogeneous, although the results are consistent with the computed rank correlation coefficients (0.7275 and 0.4701, respectively). In both instances, the r-values (0.7285 and 0.4828, respectively) are significantly larger than that reported by Slaney and Walsh for constructive search algorithms. Further, we found absolutely no evidence that the most difficult instances possess medium-sized backbones, as conjectured by Achlioptas et al. [30] for SAT.

Of more interest is the exceptionally close correspondence between the r^2 values of the $|backbone|$ and $|optsols|$ models; the absolute differences for the 6×4 and 6×6 general JSPs are only 0.0058 and 0.0108, respectively. Upon closer examination, this phenomenon is due to an extremely high correlation between $|backbone|^2$ and $log_{10}(|optsols|)$: -0.9337 and -0.9103 for 6×4 and 6×6 problems, respectively. Within each problem group, the correlation is near-perfect for instances with large backbones, and gradually decays as $|backbone| \rightarrow 0.0$. Our results indicate that, somewhat surprisingly, for problem instances with moderate-to-large backbones, the backbone size is essentially a proxy for the number of optimal solutions, and vice-versa. From the standpoint of models of problem difficulty for reasonably difficult general JSPs (i.e., those with moderate-to-large backbones), the two features are redundant. In retrospect, this observation is not surprising given what is implied by a large backbone – as more order variables are fixed, fewer solutions can satisfy the constraints of the backbone. We conjecture that a similar phenomenon occurs in SAT.

4.4 The average distance between local optima and search cost

Search in algorithms with a strong bias toward local optima, such as tabu search and certain hybridized genetic algorithms, is largely constrained to the sub-space of local optima. Consequently, we would expect search cost in these algorithms to be at least somewhat correlated with the size of this sub-space. A similar observation led Mattfeld et al. [7] to consider whether differences in the size of this sub-space could account for relative differences in the difficulty of general and workflow JSPs. Using Equation 1 (presented in Section 3.3), Mattfeld et al. define the size of the local optima sub-space as the average normalized distance $\overline{D}(s_1, s_2)$ between distinct random pairs of local optima; we denote this measure by $\overline{loptdist}$. Mattfeld et al. did not directly analyze the ability of $\overline{loptdist}$ to account for the variance in search cost

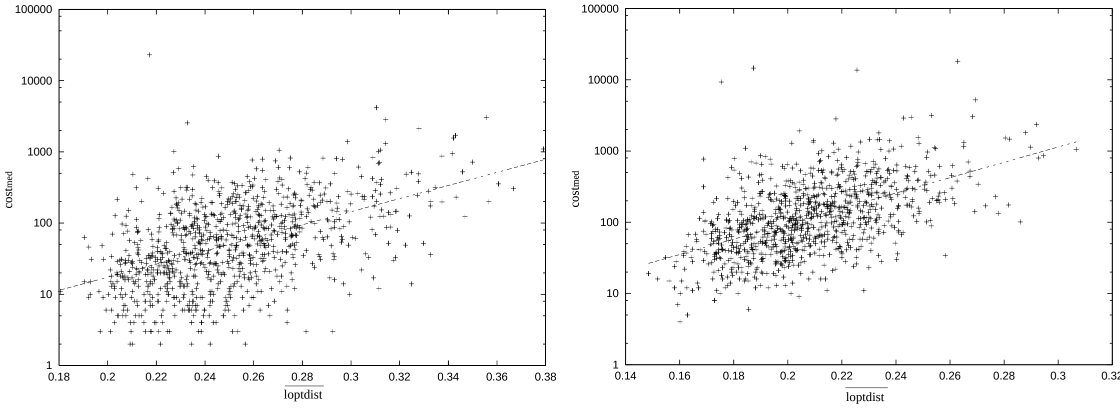

Fig. 3. Scatter-plots of $log_{10}(\overline{loptdist})$ versus $cost_{med}$ for 6×4 (left figure) and 6×6 (right figure) general JSPs; the least-squares fit lines are super-imposed. The r^2 values for the corresponding regression models are 0.2415 and 0.2744, respectively.

observed for different instances of a given problem size.

For each of our general JSPs, we compute $\overline{loptdist}$ using a set of 5000 random local optima produced using the steepest-descent procedure documented in Section 3.3. Scatter-plots of $\overline{loptdist}$ versus $log_{10}(cost_{med})$ for 6×4 and 6×6 general JSPs are shown in Figure 3. The r^2 values for the corresponding regression models are 0.2415 and 0.2744, respectively. These results confirm the intuition that the size of the local optima sub-space is correlated with the cost of finding optimal solutions under $TS_{Taillard}$, albeit more weakly than either $|optsols|$ or $|backbone|$ in 6×4 general JSPs (i.e., r^2 values of 0.2415 versus 0.5365 and 0.5307, respectively). The strength of the correlation is roughly identical to that of $|optsols|$ and $|backbone|$ for 6×6 general JSPs (i.e., r^2 values of 0.2744 versus 0.2223 and 0.2331, respectively). Finally, in contrast to both $|optsols|$ and $|backbone|$, the strength of the $\overline{loptdist}$ model is largely insensitive to relatively small changes in the problem dimensions.

To summarize, the descriptive cost model based on $\overline{loptdist}$ fails to account for a significant proportion of the variance in the cost of $TS_{Taillard}$ in the general JSP. Further, the models based on both $|optsols|$ and $|backbone|$ are at least as accurate as the $\overline{loptdist}$ model. Finally, in Section 6.2 we re-visit and ultimately refute Mattfeld et al.'s original claim regarding the ability of differences in $\overline{loptdist}$ to account for differences in the difficulty of general and workflow JSPs.

4.5 The distance between initial solutions and the nearest optimal solution and search cost

In both the JSP and SAT, the accuracy of the $|optsols|$ model decreases as the number of optimal solutions approaches 0. Analogously, the $|backbone|$ model is more accurate on problem instances with small backbones. Singer et al. [10] recently introduced a descriptive cost model for SAT that largely corrects for these deficien-

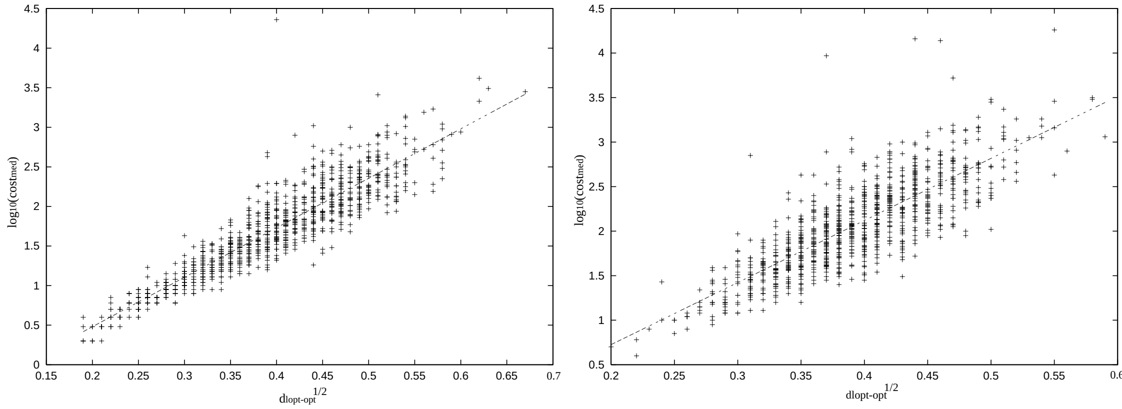

Fig. 4. Scatter-plots of $\sqrt{d_{\textit{lopt-opt}}}$ versus $log_{10}(cost_{med})$ for 6×4 (left figure) and 6×6 (right figure) general JSPs; the least-squares fit lines are super-imposed. The r^2 values for the corresponding regression models are 0.826 and 0.6541, respectively.

cies. Local search algorithms for SAT, such as GSAT or Walk-SAT [21], quickly locate sub-optimal *quasi-solutions*, in which relatively few clauses are unsatisfied. These quasi-solutions form a sub-space that contains all optimal solutions, and is largely interconnected; once a solution in this sub-space is identified, local search is typically restricted to this sub-space. This observation led Singer et al. to hypothesize that the distance between the first quasi-solution encountered and the nearest optimal solution largely dictates the cost of local search in SAT.

A obvious analog of quasi-solutions in SAT are local optima in the JSP. For each of our general JSPs, we generated 5000 random local optima using the steepest-descent procedure documented in Section 3.3. We then computed the mean normalized distance between the resulting local optima and the nearest optimal solution using Equation 1 (presented in Section 3.3); we denote the result by $d_{\textit{lopt-opt}}$. The distances are normalized to enable comparisons between 6×4 and 6×6 general JSPs; Singer et al. did not perform normalization because the problem size is held constant in their experiments. An initial regression model of $d_{\textit{lopt-opt}}$ versus $log_{10}(cost_{med})$ indicated a slight curvature in the residual plots for small values of $d_{\textit{lopt-opt}}$, which is corrected via substitution by the term $\sqrt{d_{\textit{lopt-opt}}}$. Scatter-plots of $\sqrt{d_{\textit{lopt-opt}}}$ versus $log_{10}(cost_{med})$ for 6×4 and 6×6 general JSPs are shown in Figure 4. The r^2 values for the corresponding regression models are 0.826 and 0.6541. As we discuss in detail below and in Section 6.1, the model errors are heterogeneous; however, the results are consistent with the computed rank correlation coefficients(0.9162 and 0.8072, respectively).

Clearly, the $d_{\textit{lopt-opt}}$ model is significantly more accurate than any of the $|optsols|$, $|backbone|$, or $\overline{loptdist}$ models. In both 6×4 and 6×6 general JSPs, there is strong evidence that the model residuals are heterogeneous, generally growing larger with increases in $\sqrt{d_{\textit{lopt-opt}}}$. Consequently, the $d_{\textit{lopt-opt}}$ model is less accurate for problem instances with large $d_{\textit{lopt-opt}}$, or equivalently, with large $cost_{med}$. Singer et al. report a similar phenomenon in the $d_{\textit{lopt-opt}}$ model for SAT, and these results are consistent with our prior research on local search cost in the general JSP [26].

The discrepancy between the r^2 values for 6×4 and 6×6 general JSPs is due to two factors. First, there are more very high-cost 6×6 instances (e.g., those with $log_{10}(cost_{med}) > 3.5$), and large model residuals are typically associated with such instances. Second, although the range of $d_{lopt\text{-}opt}$ is nearly identical in 6×4 and 6×6 general JSPs, the relative frequency of instances for which $\sqrt{d_{lopt\text{-}opt}} \leq 0.3$ is much larger in 6×4 general JSPs (161 versus 67). We further analyze the relationship between the $d_{lopt\text{-}opt}$ model and very high-cost general JSPs in Section 6.1 and consider the influence of the ratio of jobs to machines (n/m) on the accuracy of the $d_{lopt\text{-}opt}$ model in Section 5.2.

To summarize, the descriptive cost model based on $d_{lopt\text{-}opt}$ accounts for a substantial proportion of the variance in the cost of $TS_{Taillard}$ in typical general JSP. With few exceptions, the model residuals vary over roughly 1 to 1.5 orders of magnitude in 6×4 and 6×6 problems, respectively; the improvement is substantial in comparison to the residuals for the models based on either $|optsols|$, $|backbone|$, or $\overline{loptdist}$. Finally, the $d_{lopt\text{-}opt}$ model is also consistent with the observation that hard (easy) problem instances tend to be hard (easy) for all local search algorithms, as discussed in Section 3.4. Intuitively, if the distance between random solutions and the nearest optimal solution for a particular problem instance is very large, we would expect the instance to be difficult for *any* algorithm based on local search, as search in these algorithms clearly progresses in small increments.

4.6 *Multiple-feature models and search cost*

Table 1
The correlation (Pearson's) between search space features for 6×4 general JSPs.

	$log_{10}(\|optsols\|)$	$\|backbone\|$	$\overline{loptdist}$	$d_{lopt\text{-}opt}$
$log_{10}(\|optsols\|)$	1.0	-0.921	-0.039	-0.751
$\|backbone\|$	-0.921	1.0	0.006	0.722
$\overline{loptdist}$	-0.039	0.006	1.0	0.571
$d_{lopt\text{-}opt}$	-0.751	0.722	0.571	1.0

We now consider whether a descriptive cost model that is more accurate than the $d_{lopt\text{-}opt}$ model can be produced by considering combinations of additive and/or interaction effects of the four search space features we considered thus far. We proceed via well-known multiple regression methods. Ideally, the independent variables in a multiple regression model are highly correlated with the dependent variable, but not with each other; if the independent variables are highly correlated, they are said to be collinear. Collinearity is known to cause difficulties for model selection techniques in multiple regression, in part because the regression coefficients are not unique, making interpretation very difficult [15]. In Table 1, we show the correlation for 6×4 general JSPs between the four search space features that serve as the independent variables in our multiple regression model. Similar correlations hold for 6×6 general JSPs, indicating a high degree of collinearity among the four search space features we have considered. Finally, when the sample size is large, terms may be statistically significant due to high power, but in reality may have very small practical effect: i.e., dropping these terms yields a negligible reduction in the model r^2.

We first consider multiple regression models with only additive terms. In both 6×4 and 6×6 general JSPs, the models resulting from forward selection, backward elimination, and step-wise model selection methods [15] [33] were very different, as expected given collinear independent variables and a large sample size. However, the $d_{\textit{lopt-opt}}$ term was present in all of the resulting models, and was consistently the most statistically significant term. For 6×4 and 6×6 general JSPs, the best multiple regression models we obtained yielded r^2 values of 0.8296 and 0.6589, respectively; further, the r^2 values for all models were very similar. Given that the corresponding r^2 values for the basic $d_{\textit{lopt-opt}}$ model are 0.8260 and 0.6541, we conclude that the addition of the $\overline{\textit{loptdist}}$, $|\textit{optsols}|$, and $|\textit{backbone}|$ features fails to enhance the accuracy of the $d_{\textit{lopt-opt}}$ model. Similarly, we found *no* statistically significant interaction terms. Further, the r^2 values for any models with interaction terms were no larger than those obtained by models without interaction terms.

Interestingly, although Singer et al. control for backbone size in their experiments, they do not explicitly indicate whether an interaction effect between $d_{\textit{lopt-opt}}$ and $|\textit{backbone}|$ was observed. However, their results do suggest a lack of interaction effect, in that the slopes of the $d_{\textit{lopt-opt}}$ model are largely homogeneous across a wide range of backbone sizes and clause-to-variable ratios (e.g., see Singer et al. (2000), Table 2, p. 249); the intercepts are slightly more variable, which is likely due in part to the presence of high-residual problem instances.

4.7 A note on backbone robustness

In addition to introducing the $d_{\textit{lopt-opt}}$ model for SAT, Singer et al. also posited a causal model to account for the variance in $d_{\textit{lopt-opt}}$ observed for different problem instances. Their model is based on the notion of *backbone robustness*. A SAT instance is said to have a *robust* backbone if a substantial number of clauses can be deleted before the backbone size is reduced by at least half. Conversely, an instance is said to have a *fragile* backbone if the deletion of just a few clauses reduces the backbone size by half or more. Singer et al. argue that “backbone fragility approximately corresponds to how extensive the quasi-solution area is” ([10], p. 251), by noting that a fragile backbone allows for large $d_{\textit{lopt-opt}}$ because of the sudden drop in backbone size, while $d_{\textit{lopt-opt}}$ is necessarily small in problem instances with robust backbones.

As evidence of this hypothesis, Singer et al. measured a moderate (≈ -0.5) negative correlation between backbone robustness and the log of local search cost for large-backboned SAT instances. Surprisingly, this correlation degraded as the backbone size was decreased, leading to the hypothesis that “finding the backbone is less of an issue and so backbone fragility, which hinders this, has less of an effect” ([10], p. 254), although this conjecture was not explicitly tested. We have previously reported very similar results for general JSPs [26]. As indicated in Section 4.5 and more fully in Section 6, we have since discovered relatively serious deficiencies in the $d_{lopt\text{-}opt}$ model, and feel it is somewhat premature to posit causal hypotheses before the source of these deficiencies is completely understood. As a consequence, we have failed to pursue further analyses of backbone robustness in the JSP.

5 Applications of the $d_{lopt\text{-}opt}$ model

The analyses presented in Section 4 demonstrate that the $d_{lopt\text{-}opt}$ descriptive cost model accounts for a substantial proportion of the variance in the cost of finding optimal solutions to typical general JSPs under the $TS_{Taillard}$ algorithm. Further, more complex models that consider $d_{lopt\text{-}opt}$ in conjunction with backbone size, the number of optimal solutions, and the size of the search space failed to yield even marginal improvements. In this section, we additionally show that the $d_{lopt\text{-}opt}$ model accounts for both (1) a substantial proportion of the variance in the cost of finding *sub-optimal* solutions to typical general JSPs under $TS_{Taillard}$ and (2) differences in the relative difficulty of general JSPs with different job-to-machine ratios.

5.1 Modeling the cost of locating sub-optimal solutions

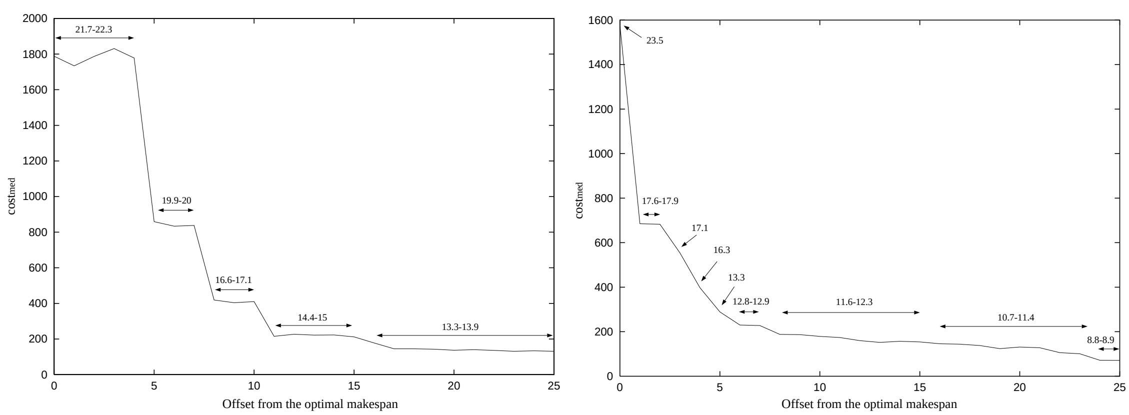

Fig. 5. The offset x from the optimal makespan C^*_{max}, $0 \leq x \leq 25$, versus the cost $cost_{med}(x)$ required to locate a solution with $C_{max} \leq C^*_{max} + x$ for two 6×6 general JSPs. The numeric annotations indicate either $d_{init-T(x)}$ for specific a x, or the range of $d_{init-T(x)}$ over a contiguous sub-interval of x.

Because they are incomplete, local search algorithms are only used to find solutions to satisfiable SAT instances, where the evaluation of the global optima is known, and is equal to the total number of clauses m. Such a priori knowledge leads to the obvious termination criterion: keep searching until a global optimum is located. Consequently, analyses of problem difficulty for local search in SAT only consider the cost required to locate globally optimal solutions. For most NP-complete problems, however, the evaluation of the global optima is not known a priori. Armed only with the knowledge that larger run-times generally lead to higher-quality solutions, local search practitioners generally use the following termination criterion: allocate as much CPU time as possible, and return the best solution found.

Although larger run-times generally yield higher-quality solutions, the relationship is typically discontinuous, non-linear, or both. Often, small or moderate increases in run-time fail, on average, to improve solution quality; for example, Stützle ([11], p. 47) notes that in the Traveling Salesman Problem "...instances appear to have 'hard cliffs' for the local search algorithm, corresponding to deep local minima, which are difficult to pass.". Similar observations have been reported for a variety of NP-complete problems, including the JSP. Another manifestation of this phenomenon has been observed by several researchers, including ourselves. Here, multiple independent trials of a particular local search algorithm typically yield sub-optimal solutions that can be partitioned into a very small number of sub-sets (often 1), with each sub-set containing solutions with identical evaluations.

One simple way to visualize these and other similar observations is to plot the cost required to achieve a solution with an evaluation of *at least* $C^*_{max} + x$ over a wide range of $x \geq 0$. In Figure 5, we provide examples of such plots for two moderately difficult 6×6 general JSPs. In both plots, the offset from the optimal makespan x is varied from 0 to 25, and the median cost (over 5000 independent runs of $TS_{Taillard}$) required to find a solution with an evaluation of at least $C^*_{max} + x$ is computed for each x, and is denoted by $cost_{med}(x)$. In the left side of Figure 5, we see a typical example of a problem instance with discrete jumps in search cost at specific sub-optimal makespans, with plateaus in search cost in between the jump points. In the right side of Figure 5, we show a problem instance for which the decay in search cost is generally more gradual; a large, discontinuous jump in search cost occurs only between $x = 0$ and $x = 1$.

As shown in Section 4, the $d_{lopt\text{-}opt}$ descriptive cost model accounts for a significant proportion of the variance in the cost of finding optimal solutions to general JSPs using $TS_{Taillard}$. Intuitively, this cost is large if $TS_{Taillard}$ is, on average, initiated from solutions that are very distant from the nearest optimal solution. We conjecture that this intuition extends to *any* sub-set of solutions, including sub-optimal solutions; we would expect local search cost to be proportional to the distance between the initial solutions and the nearest target solution. As evidence of this conjecture, we consider a set $T(x)$ containing all solutions with a makespan between C^*_{max} and $C^*_{max} + x$, $x \geq 0$, and denote the mean distance between random local optima and the nearest solution in the set $T(x)$ by $d_{init-T(x)}$; as in the computation of $d_{lopt\text{-}opt}$, the statistics are taken over 5000 independent samples. We have annotated the plots in Figure 5 with the computed $d_{init-T(x)}$, $0 \leq x \leq 25$. In both instances, (1) large jumps in search cost are clearly coincide with large jumps in $d_{init-T(x)}$, (2) intervals of roughly constant search cost correspond to contiguous sub-intervals of x with nearly identical values of $d_{init-T(x)}$, and (3) gradual drops in search cost coincide with gradual drops in $d_{init-T(x)}$. Consequently, we hypothesize that $d_{init-T(x)}$ accounts for a significant proportion of the variance in the cost of finding *both* optimal and sub-optimal solutions to typical general JSPs using $TS_{Taillard}$.

To test this hypothesis, we computed $cost_{med}(x)$ and $d_{init-T(x)}$ for both our 6×4 and 6×6 general JSPs, varying x from 1 to 25. Finding solutions to 6×4 and 6×6 general JSPs with $C_{max} > C^*_{max} + 25$ is generally easy for $TS_{Taillard}$, with $cost_{med}(25) \leq 100$ in all but a few cases. Under this methodology, we are effectively creating 25 derivatives of each problem instance (one for each value of x), which results in new 'sub-optimal' 6×4 and 6×6 problem groups, each with $25\,000$ instances. For many of the derivative instances, especially those produced using large x, $cost_{med}(x) = 0$, or equivalently $d_{init-T(x)} \approx 0.0$. We observed 1293 6×4 zero-cost instances, and 60 6×6 zero-cost instances; in both cases, the zero-cost instances are excluded in the following analysis.

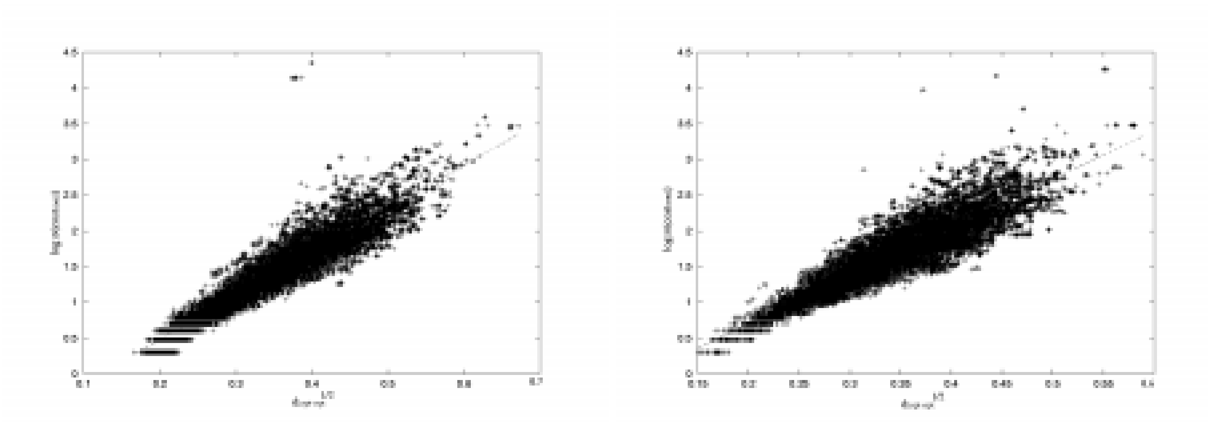

Fig. 6. Scatter-plots of $\sqrt{d_{lopt\text{-}opt}}$ versus $log_{10}(cost_{med})$ for the sub-optimal 6×4 (left figure) and 6×6 (right figure) general JSP problem groups; the regression lines are super-imposed. The r^2 values for the corresponding regression models are 0.8866 and 0.8252, respectively.

In Figure 6, we show scatter-plots of $\sqrt{d_{lopt\text{-}opt}}$ versus $log_{10}(cost_{med})$ for the sub-optimal 6×4 and 6×6 problem groups; the r^2 values for the corresponding regression models are 0.8866 and 0.8252, respectively. Clearly, the $d_{lopt\text{-}opt}$ model accounts for most of the variance in the cost of finding sub-optimal solutions to typical general JSPs using $TS_{Taillard}$. We observed larger r^2 values in the sub-optimal 6×4 and 6×6 problem groups than for the corresponding problem groups analyzed in Section 4.5: 0.8866 versus 0.8260 for the 6×4 problems and 0.8252 versus 0.6541 for the 6×6 problems. We explain the greater accuracy of the $d_{lopt\text{-}opt}$ on the sub-optimal problem groups by noting that the proportion of instances with small

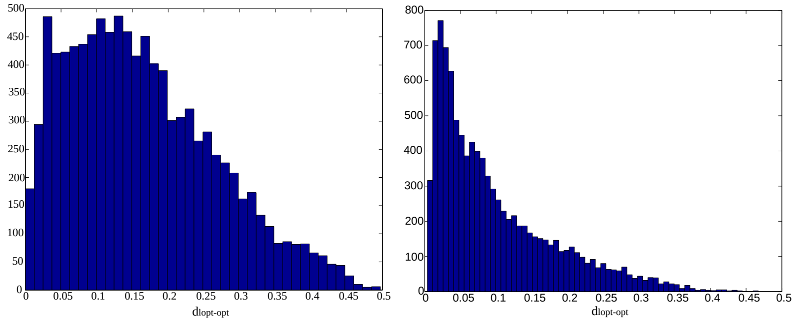

Fig. 7. Histograms of $d_{\textit{lopt-opt}}$ for $10\,000$ 4×3 (left figure) and 7×3 (right figure) general JSPs.

values of $d_{\textit{lopt-opt}}$ is larger in the sub-optimal problem groups, which corresponds to the region where the $d_{\textit{lopt-opt}}$ model is most accurate.

We conclude by noting that the $d_{\textit{lopt-opt}}$ model provides the first quantitative explanation for ‘cliffs’ in local search cost observed at particular sub-optimal evaluations: abrupt changes in local search cost occur when there are abrupt changes in $d_{\textit{lopt-opt}}$. Similarly, the plateaus observed in Figure 5 occur because solutions on the plateau are equi-distant from random local optima; $\textit{TS}_{\textit{Taillard}}$ is equally likely to encounter any of the solutions on the plateau, given a fixed run-time. Similarly, gradual increases in search cost occur when slightly better solutions are only marginally farther from random local optima.

5.2 Explaining differences in the relative difficulty of square versus rectangular JSPs

Given the accuracy of the $d_{\textit{lopt-opt}}$ model for both 6×4 and 6×6 general JSPs, it is natural to consider whether or not differences in the distribution of $d_{\textit{lopt-opt}}$ for problems with different ratios of n/m can account for the empirical observation that square JSPs are generally more difficult than rectangular JSPs.

Fixing $m = 3$, we generated $10\,000$ general JSPs for $n = 4$ through $n = 7$; although we initially considered larger values of n, the huge number of *optimal* solutions ($>$ 1 billion in many cases) prevented us from efficiently computing $d_{\textit{lopt-opt}}$. We show histograms of $d_{\textit{lopt-opt}}$ for 4×3 and 7×3 general JSPs in Figure 7. In 4×3 general JSPs, the right-tail mass of the distribution is substantial (e.g., for $d_{\textit{lopt-opt}} \geq 0.3$), especially in comparison to the distribution for 7×3 general JSPs, where instances with $d_{\textit{lopt-opt}} \geq 0.3$ are quite rare. We have also generated similar histograms for general JSPs with $n/m < 1$, where the distribution mass continues to shift toward 0.5.

Although not entirely conclusive, our results provide relatively strong evidence that the right-tail mass of the $d_{\textit{lopt-opt}}$ distribution vanishes as $n/m \to \infty$, suggesting a cause for the empirical observation that square JSPs are generally more difficult than rectangular JSPs. Further, we hypothesize that the shift from an exponential to polynomial growth in search cost at $n/m \approx 6$ is due to the disappearance of any significant mass in the right tail of the $d_{lopt\text{-}opt}$ distribution. However, due to the huge number of optimal solutions in problem instances with $n/m \geq 4$, we are currently unable to empirically test this hypothesis. Finally, we note that the accuracy of the $d_{lopt\text{-}opt}$ model should further improve as $n/m \rightarrow \infty$, due to the increasing frequency of instances with small values of $d_{lopt\text{-}opt}$. Consequently, from the standpoint of models of problem difficulty, only general JSPs with $n/m \approx 1.0$ warrant significant attention in the future.

In a previous paper [26], we argued that a shift in the distribution of $|backbone|$, and not $d_{lopt\text{-}opt}$, was responsible for differences in the relative difficulty of square versus rectangular JSPs. While our original observation still holds (i.e., that the proportion of instances with small backbones grows as $n/m \rightarrow \infty$), we have chosen to re-cast our original results in terms of the most accurate descriptive cost model available, which is based on $d_{lopt\text{-}opt}$.

6 Limitations of the $d_{lopt\text{-}opt}$ model

Although the $d_{lopt\text{-}opt}$ descriptive cost model largely accounts for the cost of finding both optimal and sub-optimal solutions to typical general JSPs using $TS_{Taillard}$, and provides an explanation for the differences in the relative difficulty of general JSPs with different job-to-machine ratios, the model is by no means perfect. As discussed in Section 4.5, the $d_{lopt\text{-}opt}$ model is less accurate for problem instances with large values of $d_{lopt\text{-}opt}$ (or, equivalently, large $cost_{med}$), and consequently fails to account for roughly 35% of the cost variance in our 6×6 general JSPs.

In this section, we identify two additional limitations of the $d_{lopt-opt}$ model. First, we conclusively demonstrate that the accuracy of the $d_{lopt-opt}$ model is exceptionally poor for very high-cost general JSPs (we provided some preliminary evidence for this conclusion in Section 4.5). Second, we show that the $d_{lopt-opt}$ model is unable to account for a significant proportion of the variance in the cost of finding optimal solutions of more structured JSPs: e.g., workflow JSPs. Although both of the results presented in this section are clearly 'negative', we feel it is important to identify and report such deficiencies, as research into why the $d_{lopt-opt}$ model fails in these circumstances is likely to lead to more general and accurate descriptive cost models in the future.

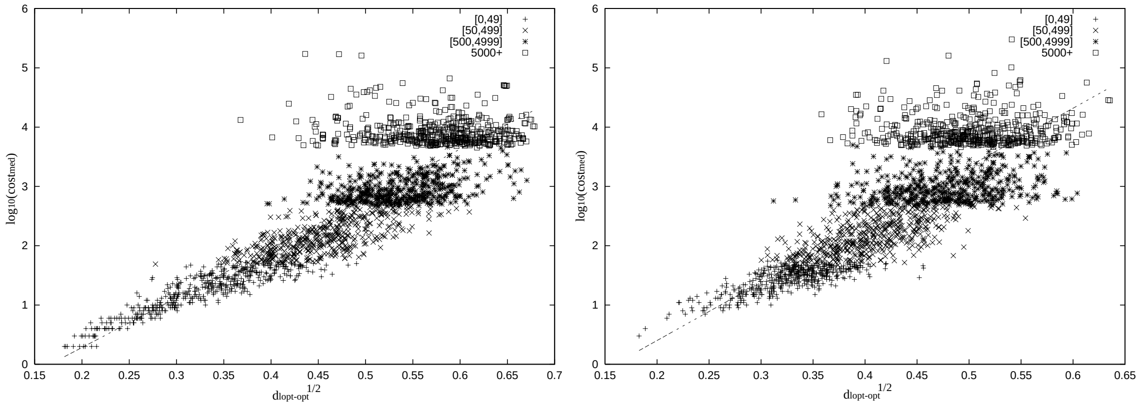

Fig. 8. Scatter-plots of $\sqrt{d_{lopt\text{-}opt}}$ versus $log_{10}(cost_{med})$ for easy ($cost_{med} \in [1, 49]$), moderate ($cost_{med} \in [50, 499]$), high ($cost_{med} \in [500, 4999]$), and very high-cost ($cost_{med} \in [5000, \infty]$) 6×4 (left figure) and 6×6 (right figure) general JSPs; the least-squares fit lines are super-imposed. The r^2 values for the corresponding regression models are 0.7742 and 0.6820, respectively.

6.1 *Modeling search cost in exceptionally hard general JSPs*

In Section 4.5, we provided evidence that the $d_{lopt\text{-}opt}$ model is less accurate for problem instances with large values of $d_{lopt\text{-}opt}$, or equivalently, large $cost_{med}$. Of particular concern are the rare, very high-cost ($cost_{med} \geq 10000$) instances appearing in both sides of Figure 4; in all but one case, these instances possess the largest residuals under the corresponding regression model. To determine whether large model residuals are typically associated with very high-cost general JSPs, we created groups of 6×4 and 6×6 general JSPs with equal proportions of problem instances over the range of $cost_{med}$. Specifically, we sub-divided the range of possible $cost_{med}$ values into the following four contiguous sub-intervals: $[1, 49]$, $[50, 499]$, $[500, 4999]$, and $[5000, \infty]$. These intervals qualitatively correspond to easy, moderate, difficult, and very difficult problem instances, respectively. For both 6×4 and 6×6 general JSPs, we then produced 500 instances belonging to each sub-interval using a generate-and-test procedure.

We provide scatter-plots of $\sqrt{d_{lopt\text{-}opt}}$ versus $log_{10}(cost_{med})$ for the two resulting problem groups in Figure 8. The r^2 values for the corresponding regression model are 0.7742 and 0.6820, respectively. First, we note that because the high-cost and very high-cost instances reside in the right-tail of the $log_{10}(cost_{med})$ distribution, the large relative frequencies of problem instances with $cost_{med}$ near the lower bounds of the corresponding sub-intervals was expected. In both problem groups, we observe a substantial reduction in the accuracy of the $d_{lopt\text{-}opt}$ model for high-cost ($500 \leq cost_{med} \leq 4999$) instances. For very high-cost instances ($cost_{med} \geq 5000$), the degradation in accuracy is far more extreme, such that $\sqrt{d_{lopt\text{-}opt}}$ provides almost no information about $cost_{med}$. These results clearly reinforce the deficiencies of the $d_{lopt-opt}$ model discussed in Section 4.5: accuracy is inversely proportional to both $d_{lopt\text{-}opt}$ and $cost_{med}$. As a direct consequence, although we are now able to account for a significant proportion of the variance in search cost for 'typical' general JSPs, an understanding of the search space properties that make certain problems exceptionally difficult for $TS_{Taillard}$ remains elusive.

Several researchers have reported situations in which problems that are exceptionally difficult for one algorithm are much easier for other algorithms [34] [35]. To date, this phenomenon has only been observed in constructive search algorithms, and occurs when one algorithm makes a particular sequence of decisions that yields a very difficult sub-problem [34]. Although this phenomenon has *not* been observed for local search in any NP-complete problem, it does raise an obvious question: "Is the exceptional difficulty of our very high-cost general JSPs algorithm-independent?". To informally answer this question, we solved both the 500 very high-cost and 1000 'typical' 6×6 (i.e., those considered in Section 4) instances using two local search algorithms other than $TS_{Taillard}$, and a constructive heuristic search algorithm. Specifically, we considered the following local search algorithms: (1) Nowicki and Smutnicki's state-of-the-art tabu search algorithm [5] and (2) van Laarhoven et al.'s simulated annealing algorithm [20]. We selected Nowicki and Smutnicki's algorithm because it uses a much more powerful move operator than $TS_{Taillard}$, and in contrast incorporates advanced long-term memory mechanisms (see Section 3.3); van Laarhoven et al.'s algorithm provides a well-known alternative local search paradigm to tabu search. The constructive algorithm we consider is Beck and Fox's constraint-directed scheduling algorithm [27], which was selected because it shares little in common with local search algorithms for the JSP. In all three cases, the search cost (as measured by the median search cost over 1000 independent trials for the two local search algorithms, and the number of nodes visited by the constructive algorithm) was significantly higher in the very high-cost instances. We therefore conclude that it appears that the difficulty of our very high-cost general JSPs is algorithm-independent.

Finally, we conjecture that the failure of the $d_{lopt\text{-}opt}$ model to account for local search cost in very difficult problem instances also extends to SAT. Although Singer et al. do not provide scatter-plots of $d_{lopt\text{-}opt}$ versus $log_{10}(cost_{med})$ for high-cost problem instances (e.g., those with large backbones), their analysis does indicate that the accuracy of the $d_{lopt\text{-}opt}$ model is inversely proportional to backbone size (e.g., see Singer et al. (2000), Table 2, p. 249), and as a consequence, to $cost_{med}$ (as in the general JSP, local search cost and backbone size are positively correlated in SAT). Further, very high-cost SAT instances possess the largest residuals under Singer et al.'s model of backbone robustness (e.g., see Singer et al. (2000), Figure 11, p. 255), which in turn is highly correlated with $d_{lopt\text{-}opt}$.

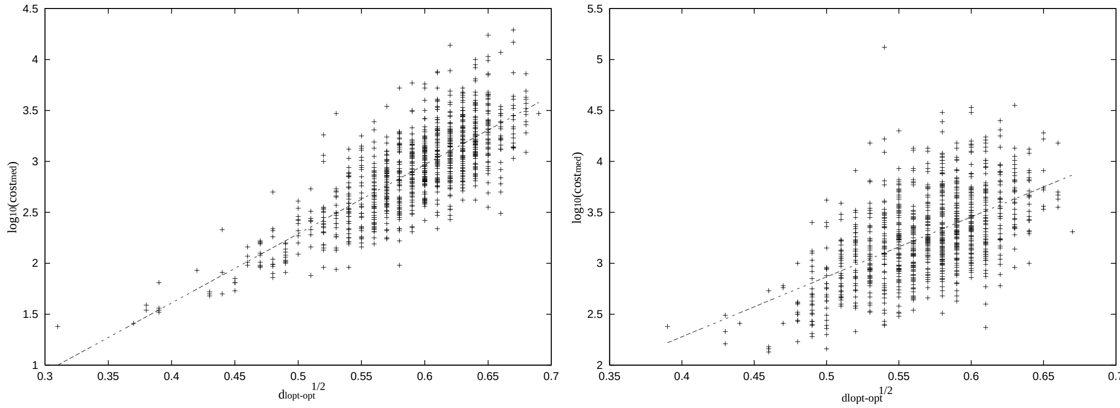

Fig. 9. Scatter-plots of $\sqrt{d_{lopt\text{-}opt}}$ versus $log_{10}(cost_{med})$ for 6×4 (left figure) and 6×6 (right figure) workflow JSPs; the least-squares fit lines are super-imposed. The r^2 values for the corresponding regression models are 0.6082 and 0.3049, respectively.

6.2 Modeling search cost in JSPs with workflow

In the JSP and SAT, the primary problem constraints are the job routing orders π_i and the disjunctive clauses, respectively. Most widely used benchmark suites consist of problem instances in which these constraints are, in expectation, completely random. An important issue is then generalization: real-world problems have non-random constraints, and it is unclear whether descriptive cost models developed for random problems are extensible to problems with more structured constraints. To study the effect of non-random constraints on the accuracy of the descriptive cost models examined in Section 4, we apply the same analysis to JSPs with workflow– which impose a simple structure on the job routing orders. All results are produced using groups of 6×4 and 6×6 workflow JSPs, each containing 1000 problem instances; the details of the problem generation process are discussed in Section 3.2.

We first consider the results for our 6×4 workflow JSPs. A scatter-plot of $\sqrt{d_{lopt\text{-}opt}}$ versus $log_{10}(cost_{med})$ is shown in the left side of Figure 9. The r^2 value for the corresponding regression model is 0.6082, which is roughly 75% of the r^2 value observed for the same model for 6×4 general JSPs (see Section 4.2). In contrasting the left sides of Figures 4 and 9, it is clear that the presence of workflow partitions greatly increases the relative frequency of instances with large values of $d_{lopt\text{-}opt}$, which partially explains the reduction in the observed r^2. Workflow partitions also have a substantial impact on the accuracy of the other models considered in Section 4. For example, we observed an r^2 value of only 0.0016 for the $\overline{loptdist}$ model, in contrast to 0.2415 for our general JSPs: the extent of the search space has no bearing on search cost in 6×4 workflow JSPs using $TS_{Taillard}$. In contrast to the results for $d_{lopt\text{-}opt}$, we observed roughly a 20% *increase* in the r^2 values for the $|optsols|$ and $|backbone|$ models, to 0.6155 and 0.6107, respectively. Consequently, the accuracy of the $|optsols|$, $|backbone|$, and $d_{lopt\text{-}opt}$ models is nearly identical. Finally, we note that strong correlation between $log_{10}(|optsols|)$ and $|backbone|^2$ is maintained ($r = 0. - 0.8936$) in the 6×4 workflow JSPs.

Next, we consider the results for our 6×6 workflow JSPs; a scatter-plot of $\sqrt{d_{lopt\text{-}opt}}$ versus $log_{10}(cost_{med})$ is shown in the right side of Figure 9. Here, we see a further reduction in the accuracy of the $d_{lopt\text{-}opt}$ model; the r^2 value is over 50% less than that observed for the same model for 6×6 general JSPs, dropping from 0.6541 to 0.3049. As with 6×4 workflow JSPs, the reduction in accuracy is partially due to dramatic increases in the relative frequency of problem instances with very large values of $d_{lopt\text{-}opt}$. Similarly, the correlation between $\overline{loptdist}$ and $log_{10}(cost_{med})$ is insignificant ($r^2 = 0.002983$), and we observed an increase in the accuracy of the $|optsols|$ and $|backbone|$ models (to r^2 values of 0.3345 and 0.2974, respectively); the relatively strong correlation between $log_{10}(|optsols|)$ and $|backbone|^2$ was also maintained ($r = -0.8346$).

Our results also cast serious doubt on Mattfeld et al.'s assertion that differences in the relative difficulty of general and workflow JSPs are due to differences in $\overline{loptdist}$. While we observed statistically significant differences between the mean $\overline{loptdist}$ of general and workflow JSPs (e.g., 0.2080 versus 0.3465 in 6×6 JSPs), we also computed a relatively weak correlation between $\overline{loptdist}$ and $log_{10}(cost_{med})$ for general JSPs – for workflow JSPs, the correlation between these same variables is almost 0. Statistically significant mean differences between general and workflow JSPs also exist for $|optsols|$, $|backbone|^2$, and $\sqrt{d_{lopt\text{-}opt}}$. Further, each of these models is at least as accurate as the $\overline{loptdist}$ model for *both* general and workflow JSPs. Consequently, we believe that any of the $|optsols|$, $|backbone|$, and $d_{lopt\text{-}opt}$ models provide *at least* an equally likely explanation as the $\overline{loptdist}$ model for the differences in relative difficulty between general and workflow JSPs.

Finally, we note that our results provide the first solid evidence that the descriptive cost models for random and structured problem instances may in fact be quite different; no research to date has considered the impact of problem structure on the descriptive cost models for SAT. Given the strong similarities between the descriptive cost models of the JSP and SAT, we conjecture that existing descriptive cost models for SAT, because they are based on random problem instances, will fail to account for a significant proportion of the variability in search cost observed for *structured* SAT instances.

7 Conclusion

Drawing from research on problem difficulty in SAT, we demonstrated that the $d_{lopt\text{-}opt}$ descriptive cost model accounts for a significant proportion of the variance in the cost of finding optimal solutions to general JSPs using a straightforward tabu search algorithm. This result was somewhat unexpected, given strong differences in both the search space topologies and local search algorithms for the general JSP and

SAT. Further, the accuracy of the model is nearly identical in both problems. In the course of our analyses, we also encountered several other important, unanticipated results: (1) backbone size and the number of optimal solutions are largely redundant features of the search space, (2) there is no significant interaction effect between $d_{lopt\text{-}opt}$ and any of the other search space features we considered, and (3) multiple-factor models do not significantly improve upon the accuracy of the $d_{lopt\text{-}opt}$ model.

We then used the $d_{lopt\text{-}opt}$ model to explain two additional phenomenon involving problem difficulty in the JSP. First, we showed that the $d_{lopt\text{-}opt}$ model also accounts for a significant proportion of the variance in the cost of finding *sub-optimal* solutions to general JSPs. The resulting extension provides an explanation for the discontinuous jumps in search cost observed at particular offsets from the optimal makespan. Second, we demonstrated that strong differences in the distributions of $d_{lopt\text{-}opt}$ provide a possible explanation for differences in the relative difficulty of square versus rectangular JSPs: problem instances with large values of $d_{lopt\text{-}opt}$ are common in square JSPs, but are relatively rare in rectangular JSPs.

Finally, we showed that the $d_{lopt\text{-}opt}$ model has limitations. First, our analyses indicated that the accuracy of the model is inversely proportional to the magnitude of $d_{lopt\text{-}opt}$, or equivalently, the difficulty of the problem instance. We also found that the accuracy is exceptionally poor on relatively rare, very high-cost problem instances. Second, we demonstrated that the accuracy of the $d_{lopt\text{-}opt}$ model is significantly worse for a particular class of structured JSPs – those with workflow partitions.

We selected $TS_{Taillard}$ precisely because it serves as a baseline for more advanced algorithms, such as the state-of-the-art algorithm of Nowicki and Smutnicki, which employs a more advanced move operator and makes more extensive use of long-term memory mechanisms. With a relatively accurate descriptive cost model of Taillard's algorithm, we can begin to *systematically* assess the influence of these more advanced features on the descriptive cost model. One inherent limitation of our analysis is that it is only directly applicable to tabu-like search algorithms for the JSP. Because descriptive cost models are tied to specific algorithms, it seems likely that other factors are responsible for local search cost in algorithms such as iterated local search or genetic algorithms, which are based on principles quite different from tabu search. At the same time, it seems likely that variations on the basic $d_{lopt\text{-}opt}$ model may account for the cost of tabu search in other NP-complete problems.

Because the descriptive cost models for both SAT and the JSP are very similar, it also seems likely that our results will be useful to researchers working on models of problem difficulty for local search in SAT. For example, our analyses indicate that the backbone size and the number of optimal solutions are largely redundant, that simultaneous consideration of number of optimal solutions, backbone size, the average distance between local optima fail to improve the accuracy of the basic

$d_{lopt\text{-}opt}$ model, and that the accuracy of the $d_{lopt\text{-}opt}$ model is exceptionally poor on very high-cost problem instances. We conjecture similar observations hold in SAT. Similarly, we showed that the descriptive cost models for random and structured problem instances can be very different. If similar results hold in SAT, they would provide some evidence that the best algorithms for solving random instances may be based on different principles than the best algorithms for solving structured instances.

References

[1] A. Jain, S. Meeran, Deterministic job-shop scheduling: Past, present and future, European Journal of Operational Research 113 (1999) 390–434.

[2] J. Błażewicz, W. Domschke, E. Pesch, The job shop scheduling problem: Conventional and new solution techniques, European Journal of Operational Research 93 (1996) 1–33.

[3] R. Vaessens, E. Aarts, J. Lenstra, Job shop scheduling by local search, INFORMS Journal on Computing 8 (3) (1996) 302–317.

[4] E. Balas, A. Vazacopoulos, Guided local search with shifting bottleneck for job-shop scheduling, Management Science 44 (2) (1998) 262–275.

[5] E. Nowicki, C. Smutnicki, A fast taboo search algorithm for the job shop problem, Management Science 42(6) (1996) 797–813.

[6] H. Fisher, G. Thompson, Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules, Prentice-Hall, Englewood Cliffs, New Jersey, 1963, Ch. 15, pp. 225–251.

[7] D. C. Mattfeld, C. Bierwirth, H. Kopfer, A search space analysis of the job shop scheduling problem, Annals of Operations Research 86 (1999) 441–453.

[8] D. A. Clark, J. Frank, I. P. Gent, E. MacIntyre, N. Tomov, T. Walsh, Local search and the number of solutions, in: Proceedings of the Second International Conference on Principles and Practices of Constraint Programming (CP-96), 1996, pp. 119–133.

[9] A. J. Parkes, Clustering at the phase transition, in: Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97), 1997, pp. 340–345.

[10] J. Singer, I. P. Gent, A. Smaill, Backbone fragility and the local search cost peak, Journal of Artificial Intelligence Research 12 (2000) 235–270.

[11] T. Stützle, Local search algorithms for combinatorial problems – analysis, improvements, and new applications, Ph.D. thesis, Darmstadt University of Technology (1999).

[12] T. Hogg, B. A. Huberman, C. P. Williams, Phase transitions and the search problem, Artificial Intelligence 81 (1–2) (1996) 1–15.

[13] K. D. Boese, A. B. Kahng, S. Muddu, A new adaptive multi-start technique for combinatorial optimization, Operations Research letters 16 (1994) 101–113.

[14] C. M. Riedys, P. F. Stadler, Combinatorial landscapes, Tech. Rep. 01-03-014, The Santa Fe Institute (2001).

[15] R. L. Ott, An Introduction to Statistical Methods and Data Analysis, Duxbury Press, Belmont, California, 1993.

[16] E. D. Taillard, Parallel taboo search techniques for the job shop scheduling problem, ORSA Journal on Computing 6 (2) (1994) 108–117.

[17] M. R. Garey, D. S. Johnson, R. Sethi, The complexity of flowshop and jobshop scheduling, Mathematics of Operations Research 1 (2) (1976) 117–129.

[18] E. D. Taillard, Benchmarks for basic scheduling problems, European Journal of Operational Research 64 (1993) 278–285.

[19] M. Dell'Amico, M. Trubian, Applying tabu search to the job-shop scheduling problem, Annals of Operations Research 41 (1993) 231–252.

[20] P. van Laarhoven, E. Aarts, J. Lenstra, Job shop scheduling by simulated annealing, Operations Research 40 (1992) 113–125.

[21] H. H. Hoos, Stochastic local search – methods, models, applications, Ph.D. thesis, Darmstadt University of Technology (1998).

[22] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Boston, MA, 1997.

[23] J. Beasley, OR-library: Distributing test problems by electronic mail, Journal of the Operational Research Society 41 (11) (1990) 1069–1072.

[24] R. H. Storer, S. D. Wu, R. Vaccari, New search spaces for sequencing problems with application to job shop scheduling, Management Science 38 (1992) 1495–1509.

[25] J. Frank, P. Cheeseman, J. Stutz, When gravity fails: Local search topology, Journal of Artificial Intelligence Research 7 (1997) 249–281.

[26] J.-P. Watson, J. C. Beck, A. E. Howe, L. D. Whitley, Toward an understanding of local search cost in job-shop scheduling, in: A. Cesta (Ed.), Proceedings of the Sixth European Conference on Planning, Springer-Verlag, 2001.

[27] J. C. Beck, M. S. Fox, Dynamic problem structure analysis as a basis for constraint-directed scheduling heuristics, Artificial Intelligence 117 (2) (2000) 31–81.

[28] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, Determining computational complexity for characteristic 'phase transitions', Nature 400 (1998) 133–137.

[29] J. Singer, Why solutions can be hard to find: A featural theory of cost for a local search algorithm on random satisfiability, Ph.D. thesis, University of Edinburgh (2000).

[30] D. Achlioptas, C. Gomes, H. Kautz, B. Selman, Generating satisfiable problem instances, in: K. Ford (Ed.), Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000), AAAI/MIT Press, 2000, pp. 256–261.

[31] J. Slaney, T. Walsh, Backbones in optimization and approximation, in: B. Nebel (Ed.), Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01), Morgan Kaufmann, 2001, pp. 254–259.

[32] J. Błażewicz, E. Pesch, M. Sterna, The disjunctive graph machine representation of the job shop scheduling problem, European Journal of Operational Research 127 (2000) 317–331.

[33] P. R. Cohen, Empirical Methods for Artificial Intelligence, The MIT Press, 1995.

[34] B. Smith, S. Grant, Sparse constraint graphs and exceptionally hard problems, in: C. Mellish (Ed.), Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), 1995.

[35] I. P. Gent, T. Walsh, Easy problems are sometimes hard, Artificial Intelligence 70 (1–2) (1994) 335–345.