A Dynamic Model of Tabu Search for the Job-Shop Scheduling Problem

Jean-Paul Watson, L. Darrell Whitley, and Adele E. Howe
Computer Science Department
Colorado State University
Fort Collins, CO 80523 USA
{watsonj,whitley,howe}@cs.colostate.edu

Abstract

Although tabu search is one of the most effective meta-heuristics for solving the job-shop scheduling problem (JSP), very little is known about why these algorithms work so well and under what conditions they excel. Our goal is to develop models of tabu search algorithms for the JSP that answer these and other related research questions. We previously [14] demonstrated that the mean distance between random local optima and the nearest optimal solution ($d_{lopt\text{-}opt}$) is highly correlated with problem difficulty for a well-known tabu search algorithm for the JSP introduced by Taillard [11]. In this paper, we discuss various shortcomings of the $d_{lopt\text{-}opt}$ model of problem difficulty and develop new models that correct for these deficiencies. We show that Taillard's algorithm can be accurately modeled using a surprisingly simple Markov chain, in that the model is a near-perfect predictor of problem difficulty. The Markov model also enables us to characterize the exact conditions under which different initialization methods can be expected to improve performance. Finally, we analyze the relationship between the Markov and $d_{lopt\text{-}opt}$ models.

Introduction

Taillard first demonstrated the effectiveness of tabu search algorithms for the job-shop scheduling problem (JSP) in 1989 [10]. Since then, researchers have introduced numerous improvements to Taillard's original algorithm [3, 8, 9]. This effort has been rewarded: tabu search algorithms are widely regarded as among the most effective approaches for generating high-quality solutions to the JSP [2, 5]. Yet, over the same time period, researchers have made little progress in developing a theoretical understanding of these algorithms. Specifically, little is known about why tabu search is so effective for the JSP and under what conditions we can expect such strong performance.

To gain insight into the behavior of tabu search algorithms for the JSP, we recently performed an extensive analysis of the relationship between various features of the fitness landscape (i.e., the underlying search space) and search cost for Taillard's algorithm [14, 13]. Our initial findings were largely negative: many features that are "known" to influence problem difficulty, such as the number of optimal solutions, the backbone size, fitness-distance correlation, and the mean distance between random local optima are in fact only weakly correlated with search cost. However, drawing from research on problem difficulty for local search in MAX-SAT, we showed that the mean distance between random local optima and the nearest optimal solution (which we denote $d_{lopt\text{-}opt}$) is highly correlated with the cost required by Taillard's algorithm to locate optimal solutions to the JSP. We also demonstrated that $d_{lopt\text{-}opt}$ accounts for much of the variance in the cost of locating sub-optimal solutions and differences in the relative difficulty of "square" versus "rectangular" JSPs.

The $d_{lopt\text{-}opt}$ model also has several shortcomings [14]. First, the model was developed and validated using small problem instances, leaving open the question of scalability. Second, despite good overall accuracy, model errors can exceed 1/2 an order of magnitude in search cost, and the model is least accurate for the most difficult problem instances. Third, and most importantly, the model provides no direct insight into the dynamic behavior of the underlying search process: it is unclear *why* $d_{lopt\text{-}opt}$ should be so highly correlated with search cost.

In this paper, we correct for these deficiencies by introducing a "dynamic" model that is based on an analysis of the run-time behavior of Taillard's algorithm. In doing so, we make the following contributions toward a theoretical understanding of tabu search algorithms for the JSP:

1. The accuracy of the $d_{lopt\text{-}opt}$ model can be significantly improved by considering the set of solutions visited by Taillard's algorithm *during* search.

2. Taillard's algorithm can be modeled using a surprisingly simple Markov chain whose states represent both the current distance from and the current search gradient relative to the nearest optimal solution. The Markov model is exceptionally accurate and accounts for nearly all of the variability in the cost required to locate optimal solutions to both small and large problem instances. The model also provides insight into the exact conditions under which different initialization methods can be expected to improve performance.

3. We identify a relationship between the Markov and $d_{lopt\text{-}opt}$ models, which enables us to account for why $d_{lopt\text{-}opt}$ is so highly correlated with search cost.

The remainder of this paper is organized as follows. First, in Section 1, we summarize the key features of Taillard's algorithm and discuss our experimental methodology. We summarize the key features of the $d_{lopt\text{-}opt}$ model in Section 2 and discuss shortcomings of the model. In Section 3, we show that the mean distance between solutions visited during search and the nearest optimal solution is more highly correlated with search cost than $d_{lopt\text{-}opt}$. We develop a dynamic Markov model of the behavior of Taillard's algorithm in Section 4 and use the resulting model to explore the conditions under which heuristic initialization can be expected to improve performance in Section 5. We conclude by discussing the implications of our results in Section 6.

1 Problem, Algorithm, and Methodology

We consider the well-known $n \times m$ static JSP [2], 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 each of the m machines in a 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 job i on machine $\pi_i(j)$ is denoted o_{ij} and is called an *operation*. An operation o_{ij} must be processed on machine $\pi_i(j)$ for an integral duration $\tau_{ij} \geq 0$. Once processing is initiated, an operation cannot be pre-empted, and concurrency is not allowed. For $2 \leq j \leq m$, o_{ij} cannot initiate processing until o_{ij-1} has completed. The scheduling objective is makespan minimization, i.e., to minimize the maximum completion time of the last operation of any job.

An instance of the $n \times m$ JSP is defined by the set of nm operation durations τ_{ij} and n job routing orders π_i. We define a *general* JSP as an instance generated by (1) sampling the τ_{ij} independently and uniformly from a fixed-width interval and (2) constructing the π_i from random permutations of the integers $[1..m]$. Most well-known JSP benchmark instances (available from the OR Library – www.ms.ic.ac.uk/info.html) are general JSPs; exceptions include instances such as `swv11-15` in which the π_i are more constrained (e.g., workflow or flowshop partitions).

Our analysis is based on a tabu search algorithm for the JSP introduced by Taillard [10, 11], which we denote $TS_{Taillard}$. We observe that $TS_{Taillard}$ is *not* the best available tabu search algorithm for the JSP: the algorithms of Nowicki and Smutnicki [8] and Barnes and Chambers [3] provide stronger overall performance. We chose $TS_{Taillard}$ because it is particularly amenable to analysis (for reasons discussed below) and serves as a basis for more advanced algorithms. Relative to $TS_{Taillard}$, state-of-the-art tabu search algorithms for the JSP employ more effective move operators and frequently re-intensify search around high-quality solutions. Instead of tackling state-of-the-art algorithms, our (pragmatic) approach is to first develop models of a basic algorithm ($TS_{Taillard}$) and to subsequently extend this model to account for algorithmic features found in state-of-the-art algorithms.

$TS_{Taillard}$ is a relatively "vanilla" implementation of tabu search and is based on the well-known *N1* move operator introduced by van Laarhoven et al. [12]. Taillard's original papers [10, 11] provide a detailed description of $TS_{Taillard}$. A key feature of $TS_{Taillard}$ is the dynamic tabu tenure, which is uniformly re-sampled from a fixed-width interval $[l_{min}, l_{max}]$ every 15 iterations. The combination of a dynamic tabu tenure (which prevents cycling) and the *N1* move operator (which guarantees that a path exists from an arbitrary solution to some optimal solution – see [12]) endows $TS_{Taillard}$ with a key property: for reasonable values of l_{min} and l_{max}, the algorithm is (at least empirically) guaranteed to eventually locate an optimal solution (see our discussion in [14]).

Our results are based on our own implementation of $TS_{Taillard}$, which deviates from Taillard's original algorithm in three respects. First, instead of initiating search from a lexicographic solution, we use a local optimum generated by applying steepest-descent (under the *N1* operator) to a random semi-active solution [7]; we investigate alternate initialization methods in Section 5. Second, we compute the makespan of neighboring solutions exactly; Taillard's original algorithm employed an estimation scheme. Third, we do not use the long-term frequency-based memory mechanism introduced by Taillard. Our modifications enable us to control for the possible impact of these mechanisms on the development of accurate behavioral models of $TS_{Taillard}$.

A key behavior of interest when analyzing $TS_{Taillard}$ is the cost required to locate *optimal* solutions to problem instances. For individual trials, this cost is naturally defined as the number of iterations c required to locate an optimal solution. Because $TS_{Taillard}$ is stochastic (due to the choice of initial solution, random tie-breaking when multiple "best" moves are available, and the tabu tenure), search cost is in fact a random variable with an approximately exponential distribution [11]. Consequently, we run $1,000$ independent trials of $TS_{Taillard}$ for a given instance, and define search cost as either the mean ($\overline{c}$) or median (c_{Q2}) number of iterations required to locate an optimal solution, depending on the context (estimates of c_{Q2} are more stable and used when possible).

As in our previous research [14], we develop and validate our models of $TS_{Taillard}$ using sets of 6×4 and 6×6 general JSPs containing $1,000$ instances each, with the τ_{ij} uniformly sampled from the interval $[1, 99]$. Additionally, due to recent advances in computing power, we are now able to assess model scalability using a set of one hundred 10×10 general JSPs (also generated by sampling the τ_{ij} uniformly from the interval $[1, 99]$). We also consider the following well-known 10×10 general JSP benchmark instances: `la16`-`la20` and `abz5`-`abz6`. We ignore several other well-known instances (e.g., Fisher and Thompson's infamous 10×10 instance and `orb01`-`orb10`) because the job routing orders are far more structured than that of a typical general JSP. We are unable to consider larger "rectangular" problem instances (i.e., instances with $n \gg m$) due to the large number of optimal solutions. For experiments involving our 6×4 and 6×6 problem sets, we set the l_{min} and l_{max} parameters of $TS_{Taillard}$ to 6 and 14, respectively; for our 10×10 problem set, we let $l_{min} = 8$ and $l_{min} = 14$. The models we develop are functions of the set of *all* optimal solutions to a given problem instance. We use Beck and Fox's [1] constraint programming algorithm to both compute the optimal makespan and to enumerate the set of optimal solutions for our test instances. Our models are also based on the notion of "distance" between pairs of solutions, which we take as the well-known *disjunctive graph* distance see [7].

2 A Summary and Critique of Prior Results

Previously, we analyzed the relationship between various fitness landscape features and problem difficulty for $TS_{Taillard}$ [14, 13]. We used regression methods to develop statistical models relating one or more of these features to the cost $log_{10}(c_{Q2})$ required to locate optimal solutions to 6×4 and 6×6 general JSPs. Because they are based on static (i.e., independent of meta-heuristic) features of the fitness landscape, we refer to these models as *static cost models*. The accuracy of a static cost model is naturally quantified as the model r^2, i.e., the proportion of the total variability in search cost accounted for by the model. We found that the accuracy of static cost models based on well-known features such as the number of optimal solutions, the backbone size, and the average distance between random local optima is only weak-to-moderate, with r^2 ranging from 0.22 to 0.53. In contrast, drawing from research on problem difficulty for local search in MAX-SAT, we demonstrated that a static cost model based on the mean distance between random local optima and the nearest optimal solution (which we denote $d_{lopt\text{-}opt}$) is significantly more accurate, yielding r^2 values of 0.8260 and 0.6541 for 6×4 and

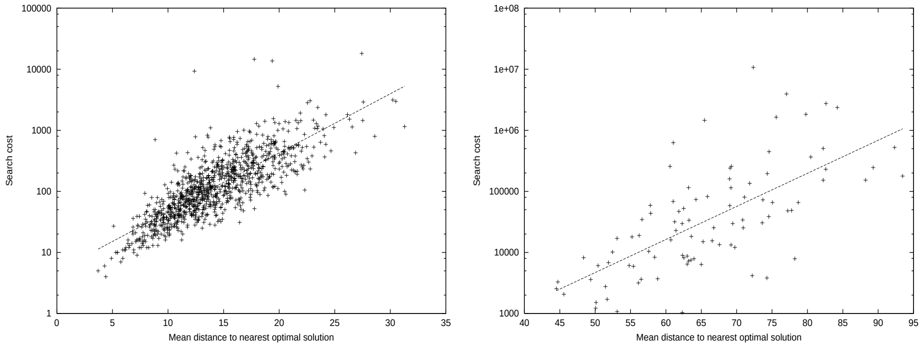

Figure 1: Scatter-plot $d_{lopt\text{-}opt}$ versus c_{Q2} for 6×6 (left figure) and 10×10 (right figure) general JSPs; the least-squares fit lines are super-imposed.

6×6 problem sets, respectively. For illustrative purposes, we provide a scatter-plot of $d_{lopt\text{-}opt}$ versus c_{Q2} in the left side of Figure 1. Additionally, we showed that $d_{lopt\text{-}opt}$ accounts for most of the variability in the cost required to locate *sub-optimal* solutions to these same problem instances (providing an explanation for the existence of "cliffs" in the search cost at particular offsets from the optimal makespan) and differences in the relative difficulty of square ($n/m = 1$) versus rectangular ($n/m > 1$)problem instances.

We also identified several shortcomings of the $d_{lopt\text{-}opt}$ model. First, as shown in the left side of Figure 1, the model is least accurate for the most difficult problems; of particular concern are the (albeit rare) instances that are associated with exceptionally large model residuals. Second, the model fails to account for roughly a third of the variability in $log_{10}(c_{Q2})$ for 6×6 instances. If high-residual instances are ignored, the residuals still vary over 1 to 1.5 orders of magnitude of c_{Q2}. Third, the differences in accuracy observed for the 6×4 and 6×6 instances raises concerns regarding scalability of the model to larger, more realistically sized problem instances. Fourth, and perhaps most importantly, the $d_{lopt\text{-}opt}$ model provides little direct insight as to *why* the mean distance between random local optima and the nearest optimal solution should be so highly correlated with search cost.

To assess the scalability of the $d_{lopt\text{-}opt}$ model, we computed $d_{lopt\text{-}opt}$ for the 92 instances of our 10×10 problem set with ≤ 50 million optimal solutions; the computation is intractable for the remaining 8 instances. For each instance, we use $5,000$ random local optima (generated by applying steepest-descent to random semi-active solutions) to estimate $d_{lopt\text{-}opt}$. We show a scatter-plot of $d_{lopt\text{-}opt}$ versus c_{Q2} for these instances in the right side of Figure 1. The r^2 value for the corresponding regression model is 0.4598, a 33% decrease in model accuracy relative to the 6×6 problem set. This result clearly demonstrates the failure of the $d_{lopt\text{-}opt}$ to scale to larger problem instances.

One obvious extension to our research would involve the development and analysis of more complex static cost models. However, we found in pilot experiments that more complex models appear to yield at best only marginal (less than 5%) improvements in accuracy. This result is consistent with our intuition: static cost models explicitly *ignore* the dynamic behavior of $TS_{Taillard}$. To develop truly accurate (i.e., $r^2 \geq 0.9$) cost models, we thought it necessary to account for the dynamic behavior of the algorithm under consideration. We now explicitly test this hypothesis, by developing both simple (Section 3) and complex (Section 4) cost models that are functions of the solutions visited by $TS_{Taillard}$ *during* search.

3 The Impact of Search Bias

Attractor basins of local optima in the JSP are surprisingly weak: they can be escaped with high probability by accepting one or two dis-improving moves and re-initiating a descent procedure [13]. Thus, search under $TS_{Taillard}$ is generally restricted to the sub-space of solutions containing both local optima and solutions very close (in terms of distance) to local optima. The intuition behind $d_{lopt\text{-}opt}$ is that it represents the size of this sub-space and as a consequence is highly correlated with search cost. However, the deficiencies of the $d_{lopt\text{-}opt}$ static cost model (specifically, the lack of scalability) suggest that either (1) $d_{lopt\text{-}opt}$ is not an entirely accurate indicator of the size of the local optima sub-space or (2) the size of the local optima sub-space is not completely indicative of search cost. We now focus on the first alternative, by developing a more accurate measure of the size of the local optima sub-space.

The $d_{lopt\text{-}opt}$ measure is a function of the distribution of the distance between *random* local optima and the nearest optimal solution (d_{opt}). Consider instead the distribution of d_{opt} for solutions visited by $TS_{Taillard}$ *during* search. Both distributions are typically Gaussian-like, although we have observed skewed distributions both with and without heavy tails. We provide examples of these distributions for two 10×10 instances in Figure 2. Although the two distributions are often identical (especially in 6×4 instances and to a lesser extent in 6×6 instances), they can also possess very different means and variances, as shown in the right side of Figure 2. Further, we commonly observe such discrepancies in our 10×10 instances – the same instances for which the $d_{lopt\text{-}opt}$ model is least accurate.

These observations led us to conjecture that the mean d_{opt} for solutions visited during search, which we denote $d_{tabu\text{-}opt}$, may be a more accurate indicator of the size of the local optima sub-space than $d_{lopt\text{-}opt}$. For a given instance, we compute $d_{tabu\text{-}opt}$ using a set of $100,000$ solutions visited by $TS_{Taillard}$ over a variable number of independent trials. Each trial is initiated from a random local optimum and terminated once an optimal solution is located; we impose the termination criterion because there exist optimal solutions from which no moves are possible under the *N1* operator [8]. We terminate the entire process, including the current trial, once we obtain $100,000$ samples.

Regression models of $d_{tabu\text{-}opt}$ versus $log_{10}(c_{Q2})$ yielded r^2 values of 0.8441 for our 6×4 instances and 0.7808 for our 6×6 instances; this corresponds to roughly 4% and 20% increases in accuracy over that of the $d_{lopt\text{-}opt}$ model, respectively. The scatter-plot for the 6×6 instances is shown in the left side of Figure 3. In either case, the absolute accuracy is remarkably high. The variable impact on the accuracy is due to frequency of instances in which the distributions of d_{opt} for random local optima and solutions visited during search are dissimilar. Model residuals are typically no larger than $1/2$ an order of magnitude, and we observe fewer and less extreme high-residual

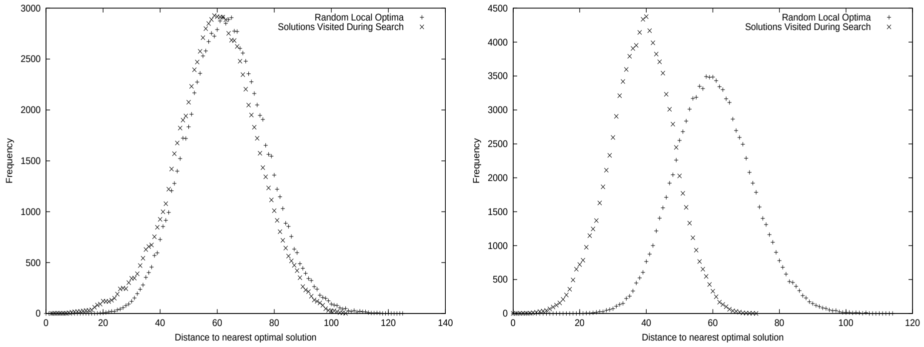

Figure 2: Histograms of the distance to the nearest optimal solution (d_{opt}) for (a) $100,000$ random local optima and (b) $100,000$ solutions generated by $TS_{Taillard}$ for two 10×10 general JSPs.

instances than under the $d_{lopt\text{-}opt}$ model. Finally, although beyond the scope of this paper, the $d_{tabu\text{-}opt}$ model provides similar improvements in the ability to predict the cost of locating sub-optimal solutions to these same problem instances.

We next assess the accuracy of the $d_{tabu\text{-}opt}$ model on the set of forty-two 10×10 instances with $\leq 100,000$ optimal solutions; the computation of $d_{tabu\text{-}opt}$ is intractable for the remaining instances. Given the relatively poor correlation between the number of optimal solutions and search cost [14], our selection criterion does *not* lead to a clean distinction between "easy" and "hard" problem instances; the hardest instance in our 10×10 problem set has approximately 1.5 *million* optimal solutions. However, on average, instances with $\leq 100,000$ optimal solutions are generally more difficult, with a median c_{Q2} of $65,710$, versus $13,291$ for instances with more than $100,000$ optimal solutions.

A regression model of $d_{tabu\text{-}opt}$ versus $log_{10}(c_{Q2})$ for these 10×10 instances yielded an r_2 value of 0.6641; we show the corresponding scatter-plot in the right side of Figure 3. The resulting r^2 represents an approximately 41% increase in accuracy over the $d_{lopt\text{-}opt}$ model. However, the model residuals typically vary from between $1/2$ and 1 order of magnitude, leaving a moderate proportion of the variability in search cost unexplained. The difference in model r^2 between the 6×6 and 10×10 problem sets is only ≈ 0.14. We have also annotated the scatter-plot with data for five of the seven 10×10 general JSPs found in the OR Library; both `la16` and `la17` have approximately 6.8 and 11.8 million optimal solutions, respectively, making computation of $d_{tabu\text{-}opt}$ intractable. The `abz5` and `la19` instances are known to be significantly more difficult than their respective counterparts (i.e., `abz6`, `la18`, and `la20`) for numerous local search algorithms, which is consistent given the observed differences in $d_{tabu\text{-}opt}$ [5].

Our results clearly demonstrate that the mean distance between solutions visited *during* search and the nearest optimal solution ($d_{tabu\text{-}opt}$) is largely responsible for the cost required by Taillard's algorithm to locate optimal solutions to general JSPs. However, two key issues remain. First, as shown by the difference in r^2 obtained for the 6×6 and 10×10 instances, there is still some evidence that the $d_{tabu\text{-}opt}$ model may

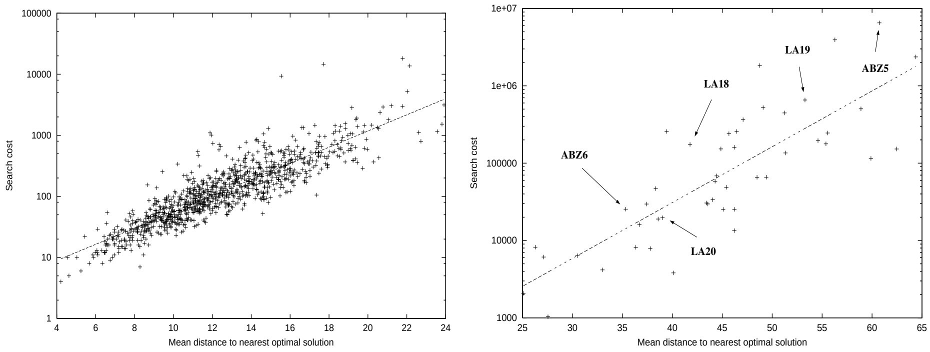

Figure 3: Scatter-plots of $d_{tabu\text{-}opt}$ versus search cost (c_{Q2}) for 6×6 (left figure) and 10×10 (right figure) general JSPs; the least-squares fit lines are super-imposed.

fail to scale to even larger problem instances. Second, as was the case with $d_{lopt\text{-}opt}$, it is unclear *why* $d_{tabu\text{-}opt}$ is so highly correlated with search cost. To address these issues, we now examine the dynamic behavior of $TS_{Taillard}$ in more detail.

4 A Dynamic Cost Model of Tabu Search

The dynamic behavior of memoryless local search algorithms (e.g., simulated annealing or iterated local search) can, at least in principle, be modeled using Markov chains: the set of feasible solutions is known and the transition probabilities between neighboring solutions can be computed. Although exact, such models require an exponential number of states for NP-hard problems and therefore provide little insight into the qualitative nature of the search process. The challenge is to develop "lumped" models, in which large numbers of solutions are grouped into individual states, yielding more tractable and consequently understandable Markov chains. A similar approach is possible when modeling tabu search, although we must additionally embed the contents of short-term memory into the state definition. We then face two questions: "What criterion do we use to aggregate solutions?" and "How do we model short-term memory?".

Given an objective of makespan minimization, we aggregate solutions based on their distance to the nearest optimal solution. To model the impact of short-term memory, we analyze how search progresses in terms of trends either toward or away from the nearest optimal solution. In Figure 4, we show a time-series of the distance to the nearest optimal solution for a random walk (left figure) and $TS_{Taillard}$ (right figure) for a 10×10 general JSP. As expected, the random walk exhibits minimal trending behavior. In contrast, $TS_{Taillard}$ exhibits distinct trending behavior, often maintaining the current search gradient for extended periods of time. Similar results hold in a limited sampling of our 6×4, 6×6, and 10×10 instances. This suggests that $TS_{Taillard}$'s short-term memory mechanism influences the search process by consistently biasing search either toward or away from the nearest optimal solution.

We then define a state $S_{i,grad}$ in our Markov model as a pair representing (1) the set of solutions distance i from the nearest optimal solution and (2) the current search

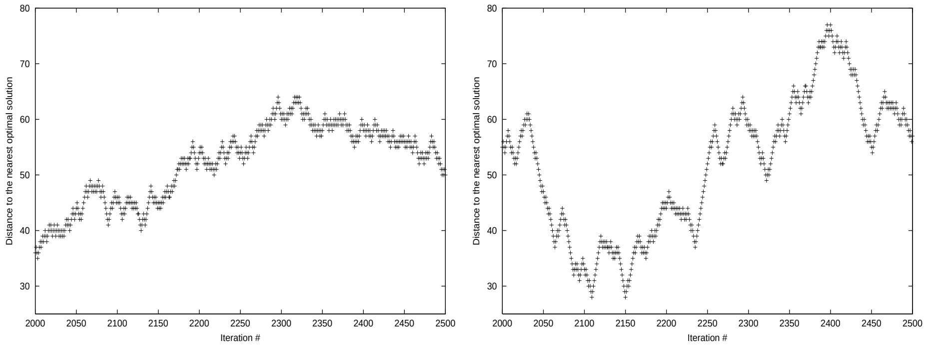

Figure 4: Time-series of the distance to the nearest optimal solution for the solutions visited by a random walk (left figure) and $TS_{Taillard}$ (right figure) for a 10×10 general JSP.

gradient *grad*. We define *grad* as the difference in d_{opt} for solutions from the current and previous iteration of $TS_{Taillard}$. Although $grad \in [-1, 0, 1]$, for clarity we denote these numeric values symbolically by *closer*, *equal*, and *farther*, respectively. In effect, we are modeling the impact of short-term memory as a *single* scalar (*grad*) and embedding this scalar into the state definition. Given a maximum possible distance of D from a solution to the nearest optimal solution, our Markov model consists of exactly $3 \cdot (D+1)$ states (the "$+1$" state represents the set of optimal solutions).

Next, we specify the transition probabilities between pairs of states $S_{i,grad'}$ and $S_{j,grad}$ in our model. Let the conditional probability $P(S_{i,grad'}|S_{j,grad})$ denote the probability of *simultaneously* altering the search gradient from *grad* to $grad'$ and moving from a solution at distance j from the nearest optimal solution to a solution at distance i away from the nearest optimal solution. The majority of these probabilities obviously equal 0: specifically, for any pair of states $S_{i,grad'}$ and $S_{j,grad}$ with $|i - j| > 1$, or when simultaneous changes in both gradient and distance to the nearest optimal solution are logically impossible, such as from state $S_{i,closer}$ to state $S_{i+1,closer}$. For each i such that $1 \leq i \leq D$, exactly 9 non-zero transition probabilities are possible:

- $P(S_{i-1,closer}|S_{i,closer})$, $P(S_{i,equal}|S_{i,closer})$, and $P(S_{i+1,farther}|S_{i,closer})$
- $P(S_{i-1,closer}|S_{i,equal})$, $P(S_{i,equal}|S_{i,equal})$, and $P(S_{i+1,farther}|S_{i,equal})$
- $P(S_{i-1,closer}|S_{i,farther})$, $P(S_{i,equal}|S_{i,farther})$, and $P(S_{i+1,farther}|S_{i,farther})$

The set of transition probabilities is also subject to the total-probability constraints:

- $P(S_{i-1,closer}|S_{i,closer}) + P(S_{i,equal}|S_{i,closer}) + P(S_{i+1,farther}|S_{i,closer}) = 1.0$
- $P(S_{i-1,closer}|S_{i,equal}) + P(S_{i,equal}|S_{i,equal}) + P(S_{i+1,farther}|S_{i,equal}) = 1.0$
- $P(S_{i-1,closer}|S_{i,farther}) + P(S_{i,equal}|S_{i,farther}) + P(S_{i+1,farther}|S_{i,farther}) = 1.0$

To complete our Markov model, we impose a reflective barrier at $i = D$ and an absorbing state at $i = 0$ by imposing the constraints $P(S_{D+1,farther}|S_{D,farther}) = 0$ and $P(S_{0,equal}|S_{0,equal}) = 1$, respectively.

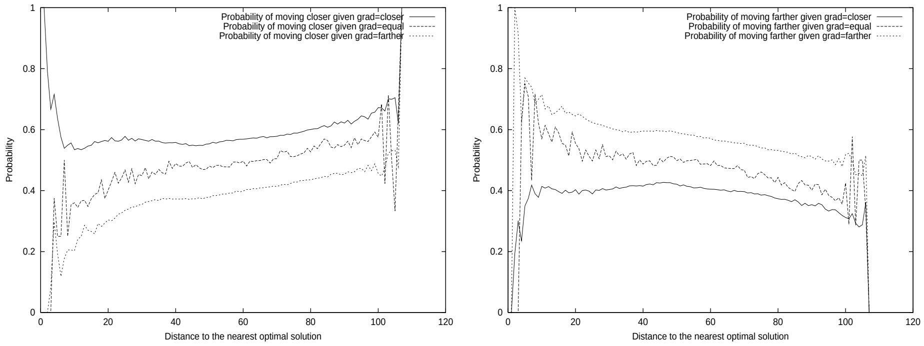

Figure 5: The transition probabilities for moving closer to (left figure) or farther from (right figure) the nearest optimal solution under *TS*$_{Taillard}$ for a 10×10 general JSP.

We estimate the set of transition probabilities for a given problem instance by analyzing the set of solutions visited by *TS*$_{Taillard}$ over a large number of independent trials. At each iteration $k \geq 1$ of each trial, we compute (1) the distance j from the current solution s to the nearest optimal solution and (2) the current search gradient *grad*, which is a function of solutions encountered in both the current (kth) and previous ($(k-1)$th) iterations. Given the neighboring solution s' of s selected for the next ($(k+1)$th) iteration, we then compute (3) the distance i from s' to the nearest optimal solution and (4) the search gradient *grad*$'$ given the solutions s' and s. We denote the total number of occurrences of state $S_{j,grad}$ by $\#(S_{j,grad})$ and the total number of occurrences of a successor state $S_{i,grad'}$ given a current state $S_{j,grad}$ by $\#(S_{i,grad'}|S_{j,grad})$; both quantities are tracked over all iterations of all trials. We execute *TS*$_{Taillard}$ until $\#(S_{i,closer}) \geq 50$ for $1 \leq i \leq d_{lopt\text{-}opt}$; individual trials are initiated from random local optima and terminated once an optimal solution is located. Because *TS*$_{Taillard}$ is empirically guaranteed to eventually locate an optimal solution and there is a non-zero probability of initiating a trial from a random local optimum that is distance $i \geq d_{lopt\text{-}opt}$ from the nearest optimal solution, the termination criterion will eventually be satisfied as the number of trials approaches ∞.

We compute estimates of the transition probabilities using the obvious formulas: e.g., $P(S_{i-1,closer}|S_{i,closer}) = \#(S_{i-1,closer}|S_{i,closer})/\#(S_{i,closer})$. Because $\#(S_{i,closer}) \geq 50$ is possible for $i > d_{lopt\text{-}opt}$, we take $D = X - 1$ where X is the minimal value satisfying $\#(S_{X,closer}) < 50$. Empirically, omitting states $S_{i,grad}$ with $i \gg d_{lopt\text{-}opt}$ appears to have little impact on model accuracy. We observe that our estimates of the transition probabilities are largely insensitive to both the random initial local optima and the sequence of local optima visited during individual trials: i.e., the statistics appear to be isotropic.

In Figure 5, we show the estimated probabilities of moving closer to (left figure) or farther from (right figure) the nearest optimal solution for a 10×10 general JSP; the probability of maintaining an *equal* search gradient is negligible ($p < 0.1$) for any i. We observe qualitatively similar results for all of our test instances. These results indicate that tabu search in the JSP can be viewed as a diffusion process with a central

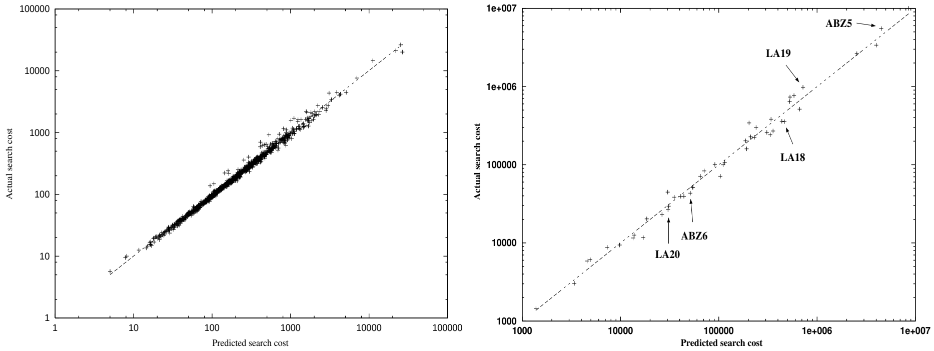

Figure 6: Scatter-plots of the observed versus predicted mean cost ($\overline{c}$) to locate an optimal solution under *TS_{Taillard}* for 6×6 (left figure) and 10×10 (right figure) general JSPs; the least-squares fit lines are super-imposed.

restoring force – the probability of moving closer to (farther from) the nearest optimal solution is proportional (inversely proportional) to the current distance from the nearest optimal solution. In other words, there is pressure toward solutions that are an "average" distance from the nearest optimal solution, which is consistent with the Gaussian-like histograms of d_{opt} observed for $TS_{Taillard}$ (as shown in Figure 2). The impact of short-term memory is also evident, in that the probability of continuing to move along the current gradient is very strong and exceeds 0.5 independently of i for nearly all problem instances. This bias accounts for the strong trending behavior observed in the time-series of d_{opt} for $TS_{Taillard}$, as shown in the right side of Figure 4. Finally, in contrast to the transition probabilities under a random walk, as $i \to 0$ the probability of continuing to move closer to the optimal solution actually rises, typically approaching 1 at some $i \geq 3$.

To validate our Markov model, we compare the predicted versus actual mean search cost $\overline{c}$ for our 6×4, 6×6, and 10×10 problem sets. For a given instance, we estimate the predicted $\overline{c}$ by repeatedly simulating the Markov chain defined by D, the set of states $S_{i,grad}$, and the estimated transition probabilities $P(S_{i,grad'}|S_{j,grad})$. Let v_i denote the mean number of iterations (with statistics taken over $10,000$ samples) required to achieve a state $S_{0,closer}$ given an initial state $S_{i,X}$. We set the initial value of X to equal either *closer* or *farther* with equal probability, given that the probability of maintaining an *equal* search gradient is negligible. Let δ be the integer nearest $d_{lopt\text{-}opt}$. We then define the predicted mean search cost ($\overline{c}$) as v_δ, i.e., search is initiated from solutions that are, on average, roughly distance $d_{lopt\text{-}opt}$ from the nearest optimal solution.

For our 6×4 and 6×6 instances, log_{10}-log_{10} regression models of the predicted versus actual $\overline{c}$ yielded r^2 values of 0.9941 and 0.9939, respectively; we show the corresponding scatter-plot for the 6×6 instances in the left side of Figure 6. Model accuracy is exceptionally high in both problem sets, and for all instances the predicted $\overline{c}$ is within a factor of 2 of the actual $\overline{c}$. To assess the scalability of our Markov model, we consider the subset of forty-two 10×10 instances with $\leq 100,000$ optimal solutions; the estimation of the transition probabilities is intractable for the remaining instances. For these instances, a log_{10}-log_{10} regression model of the predicted versus actual $\overline{c}$ yielded an r^2 value of 0.9877. We show the corresponding scatter-plot in the right side of Figure 6, which additionally includes the results for the benchmark instances `la18-la20` and `abz5-abz6`. These results clearly demonstrate that the behavior of Taillard's algorithm can be modeled with high fidelity as a simple one-dimensional random walk. In contrast to both the $d_{lopt\text{-}opt}$ and $d_{tabu\text{-}opt}$ models, the Markov model appears scalable. Finally, we note that our Markov model is equally successful in accounting for the variability in the cost of locating sub-optimal solutions to these same problem instances.

4.1 The Relationship Between the Models

In hindsight, the success of the $d_{lopt\text{-}opt}$ model was due to the fact that $d_{lopt\text{-}opt}$ and $d_{tabu\text{-}opt}$ are highly correlated for small problem instances. What remains is to establish a link between the $d_{tabu\text{-}opt}$ model and the lumped Markov model. As previously noted, the *qualitative* forms of the estimated transition probabilities (e.g., see Figure 5) are identical for all of the problem instances we examined. Any major differences are due to variability in D, which (like $d_{tabu\text{-}opt}$) can be viewed as a measure of the size of the local optima sub-space. We also observe that these transition probabilities are roughly symmetric around $D/2$ and that search in $TS_{Taillard}$ is necessarily biased toward solutions that are approximately distance $D/2$ from the nearest optimal solution. But the latter quantity is essentially equivalent to $d_{tabu\text{-}opt}$, and consequently $d_{tabu\text{-}opt} \approx D/2$. Thus, we believe the success of the $d_{tabu\text{-}opt}$ model is due to the fact that it estimates a key parameter (D) of the lumped Markov model.

5 The Impact of Initialization Method on Performance

Our models of $TS_{Taillard}$ are based on the assumption that search is initiated from a random local optimum. But can our models yield any insight into the impact of heuristic initialization on algorithm performance? Although researchers generally agree that high-quality initial solutions *can* improve the performance of tabu search algorithms (e.g., see [6]), the exact conditions under which improvements can be achieved, and the expected degree of improvement, are poorly understood. In this section, we explore a particular aspect of this broader issue: What impact do different initialization methods, both heuristic and random, have on the cost required by $TS_{Taillard}$ to locate *optimal* solutions to problem instances?

To validate our Markov model, we only computed v_δ: the mean number of iterations required to locate an optimal solution, given a starting point that is (modulo rounding) distance $d_{lopt\text{-}opt}$ from the nearest optimal solution. But what does our model predict if search is initiated from a solution that is either closer to or farther from the nearest optimal solution than δ? In Figure 7, we show plots of the predicted costs v_i over the full range of i for a 6×6 (left figure) and 10×10 (right figure) general JSP. For the 6×6 instance, search cost rises rapidly between $i = 3$ and $i = 10$, but only gradually increases once $i > 10$. In contrast, search cost for the 10×10 instance rises rapidly between $i = 2$ and $i \approx 15$, but is roughly *constant* (modulo the sampling noise) once $i > 15$. Even when $i = 2$, our model predicts that search cost is still significant:

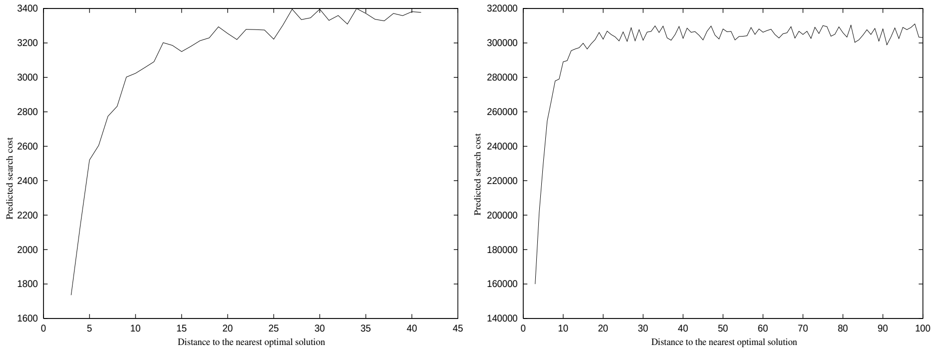

Figure 7: Predicted cost required by $TS_{Taillard}$ to locate an optimal solution, given an initial solution that is distance i from the nearest optimal solution, for two 6×6 (left figure) and 10×10 (right figure) general JSPs.

if the initial search gradient is not *closer*, search is driven toward solutions that are an "average" distance from the nearest optimal solution and any benefit of a favorable initial position is lost. We observed qualitatively identical behavior in a large sampling of our problem instances: for easy (hard) instances, the approach toward an asymptotic value as $i \to D$ is gradual (rapid).

Our Markov model predicts that the distance to the nearest optimal solution, and not the fitness, dictates the benefit of a particular initialization method. The distinction is especially key in the general JSP, where fitness-distance correlation is known to be comparatively weak [13]. In particular, our model predicts that an initialization method will at best have a minimal impact on search cost *unless* the resulting solutions are very close to the nearest optimal solution. To test this hypothesis, we analyzed the performance of $TS_{Taillard}$ using a variety of heuristic and random initialization methods. Following Jain et al. [6], we consider the following set of high-quality priority dispatch rules (PDRs), used in conjunction with Giffler and Thompson's procedure for generating active schedules[4]: *FCFS* (First-Come, First-Serve), *LRM* (Longest ReMaining work), *MWKR* (Most WorK Remaining), and *SPT* (Shortest Processing Time). We additionally considered both active and non-delay solutions [4] generated using random PDRs, which we respectively denote RND_{active} and $RND_{nondelay}$. We denote our baseline random semi-active solutions by $RND_{semiactive}$. Finally, we examined Taillard's original lexicographic solution method, denoted *LEXICO*, and the insertion procedure introduced by Werner and Winkler [15], which we denote by *WW*; the latter is one of the best constructive heuristics available for the general JSP [6]. As with $RND_{semiactive}$, we post-process the resulting solutions by applying a steepest-descent procedure to generate a local optimum.

For each initialization method, we computed $d_{lopt\text{-}opt}$ (by substituting the optima generated by a particular initialization method for random local optima) for the forty-two 10×10 instances with $\leq 100,000$ optimal solutions. With the exception of *LEXICO*, all of the methods we consider are stochastic. Consequently, we define $d_{lopt\text{-}opt}$ as the mean distance between $5,000$ random solutions and the nearest opti-

Table 1: The differences in both the mean distance to the nearest optimal solution ($d_{lopt\text{-}opt}$) and search cost ($Q2$) for various initialization methods, measured relative to random semi-active solutions ($RND_{semiactive}$).

Initialization Method				
FCFS	*LRM*	*MWKR*	*SPT*	$RND_{semiactive}$
$d_{lopt\text{-}opt}$				
58.49	97.41	97.94	64.97	70.92
Significance of Mean Difference in $d_{lopt\text{-}opt}$ relative to $RND_{semiactive}$				
$p < 0.0001$	$p < 0.0001$	$p < 0.0001$	$p = 0.1256$	$p = 1.0$
Percent Mean Difference in c_{Q2} relative to $RND_{semiactive}$				
1.76	2.32	2.94	1.55	0.0
Significance of Mean Difference in $log_{10}(c_{Q2})$ relative to $RND_{semiactive}$				
$p = 0.0594$	$p = 0.0836$	$p = 0.0727$	$p = 0.0769$	$p = 1.0$

Initialization Method				
LEXICO	RND_{active}	$RND_{nondelay}$	*WW*	$RND_{semiactive}$
$d_{lopt\text{-}opt}$				
49.25	64.68	58.55	53.10	70.92
Significance of Mean Difference in $d_{lopt\text{-}opt}$ relative to $RND_{semiactive}$				
$p < 0.0001$	$p < 0.0001$	$p < 0.0001$	$p < 0.0001$	$p = 1.0$
Percent Mean Difference in c_{Q2} relative to $RND_{semiactive}$				
1.44	1.07	0.06	-2.79	0.0
Significance of Mean Difference in $log_{10}(c_{Q2})$ relative to $RND_{semiactive}$				
$p = 0.5129$	$p = 0.5730$	$p = 0.5555$	$p = 0.3090$	$p = 1.0$

mal solution. We show the resulting $d_{lopt\text{-}opt}$ for each initialization method in Table 1. We also provide the p-values for the statistical significance of the mean difference in $d_{lopt\text{-}opt}$ between the various methods and $RND_{semiactive}$, which we obtained using a Wilcoxon signed rank test. With the exception of *SPT*, we observe significant differences in $d_{lopt\text{-}opt}$ between our baseline method ($RND_{semiactive}$) and the other initialization methods. Initially, this data appears to suggest that it may be possible to improve the performance of $TS_{Taillard}$ using initialization methods with low $d_{lopt\text{-}opt}$. However, the lowest absolute values of $d_{lopt\text{-}opt}$ (obtained using the *LEXICO* and *WW* methods) are still large. For our 10×10 instances, our Markov model predicts that these solutions are typically too far from the nearest optimal solution to have any impact on search cost (see the right side of Figure 7).

To test this hypothesis, we computed c_{Q2} using each initialization method (using $1,000$ independent trials of $TS_{Taillard}$) for each of the forty-two 10×10 instances with $\leq 100,000$ optimal solutions. We then computed the percent difference in c_{Q2} for each method relative to our baseline initialization method $RND_{semiactive}$; the results are shown in Table 1. We observe a *worst-case* deviation of less than 3%, and the best improvement (obtained under *WW*) is only 2.70%. Further, all observed discrepancies can be attributed to sampling error in the estimates of c_{Q2}, and in no case was the difference in search cost statistically significant (we provide the p-values resulting from a Wilcoxon signed-rank test in Table 1). The data clearly support the hypothesis predicted by our dynamic model: for difficult problems, the choice of initialization method has no significant impact on the performance of $TS_{Taillard}$.

The results presented in this section concern the impact of initialization method on the cost required by $TS_{Taillard}$ to locate *optimal* solutions to *difficult* problem instances. Although beyond the scope of this paper, our Markov model also predicts that initialization methods can significantly impact the cost of locating both optimal solutions to easy or moderate problem instances and good *sub-optimal* solutions to a wide range of problem instances. We have confirmed these predictions experimentally [13]. Finally, we note that our model says nothing about the impact of initialization method on the performance of tabu search algorithms that employ re-intensification, such as Nowicki and Smutnicki's algorithm [8]; we are currently investigating this issue.

6 Implications and Conclusions

Our results provide a significant first step toward understanding the behavior of tabu search algorithms. We have introduced a Markov model of the behavior of Taillard's tabu search algorithm for the JSP. Despite its simplicity, this model is able to predict the cost of locating optimal solutions with a remarkably high degree of precision. Further, our dynamic model indicates that Taillard's algorithm can be viewed as a straightforward one-dimensional random walk, with a bias toward solutions that are an average distance from the nearest optimal solution. In contrast to previous models based on static features of the fitness landscape, the dynamic model is also scalable and provides direct insight into the search process. We also identified a link between the static and dynamic models, providing a hitherto lacking explanation for the success of the static models. Finally, we used our dynamic model to gain some insight into the conditions under which different initialization methods can be expected to improve performance.

While our research has yielded a highly accurate behavioral model of $TS_{Taillard}$, several questions relating to generalization remain. Can our models be extended to more complex tabu search algorithms for the JSP? We are currently analyzing the impact of features found in several state-of-the-art tabu search algorithms, including "elite solution recovery" [8] and other forms of long-term memory, on our models of $TS_{Taillard}$. Can our models be extended to more realistic and/or different problem instances? We have already shown that our models extend to JSPs with workflow and flowshop partitions [13]. However, it is currently unclear whether our models apply to JSPs with more structured forms of the job processing times. Can our models be extended to tabu search algorithms for other combinatorial optimization problems? We have preliminary evidence that similar models hold for tabu search in the Quadratic Assignment Problem, although our results to date are by no means conclusive.

Acknowledgments

This work was sponsored 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.

References

[1] J.C. Beck and M.S. Fox. Dynamic problem structure analysis as a basis for constraint-directed scheduling heuristics. *Artificial Intelligence*, 117:31–81, 2000.

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

[3] J.B. Chambers and J.W. Barnes. New tabu search results for the job shop scheduling problem. Technical Report ORP96-10, Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin, 1996.

[4] B. Giffler and G. L. Thompson. Algorithms for solving production scheduling problems. *Operations Research*, 8:487–503, 1960.

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

[6] A.S. Jain, B. Rangaswamy, and S. Meeran. New and "stronger" job-shop neighborhoods: A focus on the method of Nowicki and Smutnicki (1996). *Journal of Heuristics*, 6:457–480, 2000.

[7] D.C. Mattfeld, C. Bierwirth, and H. Kopfer. A search space analysis of the job shop scheduling problem. *Annals of Operations Research*, 86:441–453, 199.

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

[9] F. Pezzella and E. Merelli. A tabu search method guided by shifting bottleneck for the job shop scheduling problem. *European Journal of Operational Research*, 120:297–310, 2000.

[10] É.D. Taillard. Parallel taboo search technique for the jobshop scheduling problem. Technical Report ORWP 89/11, DMA, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 1989.

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

[12] P.J.M van Laarhoven, E.H.L. Aarts, and J.K. Lenstra. Job shop scheduling by simulated annealing. *Operations Research*, 40:113–125, 1992.

[13] J.P. Watson. *Cost Models of Local Search Algorithms for Job Shop Scheduling*. PhD thesis, Colorado State University, In Preparation.

[14] J.P. Watson, J.C. Beck, A.E. Howe, and L.D. Whitley. Problem difficulty for tabu search in job-shop scheduling. *Artificial Intelligence*, In Press.

[15] F. Werner and A. Winkler. Insertion techniques for the heuristic solution of the job shop problem. *Discrete Applied Mathematics*, 58:191–211, 1995.