Tabu Search: A Tutorial

FRED GLOVER

Center for Applied Artificial Intelligence
University of Colorado
Boulder, Colorado 80309-0419

Tabu search is a "higher level" heuristic procedure for solving optimization problems, designed to guide other methods (or their component processes) to escape the trap of local optimality. Tabu search has obtained optimal and near optimal solutions to a wide variety of classical and practical problems in applications ranging from scheduling to telecommunications and from character recognition to neural networks. It uses flexible structures memory (to permit search information to be exploited more thoroughly than by rigid memory systems or memoryless systems), conditions for strategically constraining and freeing the search process (embodied in tabu restrictions and aspiration criteria), and memory functions of varying time spans for intensifying and diversifying the search (reinforcing attributes historically found good and driving the search into new regions). Tabu search can be integrated with branch-and-bound and cutting plane procedures, and it has the ability to start with a simple implementation that can be upgraded over time to incorporate more advanced or specialized elements.

Tabu search is a metaheuristic that can be superimposed on other procedures to prevent them from becoming trapped at locally optimal solutions. The method can

0091-2102/90/2004/0074$01 25

PROGRAMMING INTEGER ALGORITHMS, HEURISTICS

INTERFACES **20:** 4 July–August 1990 (pp. 74–94)

be used to guide any process that employs a set of moves for transforming one solution (or solution state) into another and that provides an evaluation function for measuring the attractiveness of these moves. (Examples of moves are changing the value assigned to a variable, adding or deleting an element from a set, interchanging the position of two jobs on a machine, and executing a pivot step. The form of the guidance provided by tabu search is highly flexible and often motivates the creation of new types of moves and evaluation criteria to take advantage of its adaptability to different problem structures and strategic goals.

Although still in an early stage of development, tabu search has enjoyed a number of successes. In a variety of problem settings, it has found solutions superior to the best previously obtained by alternative methods. In other cases, it has demonstrated advantages in ease of implementation or in the ability to handle additional considerations (such as constraints not encompassed by an original problem formulation).

A partial list of tabu search applications follows:

- Employee scheduling [Glover and McMillan 1986],
- Maximum satisfiability problems [Hansen and Jaumard 1987],
- Character recognition [Hertz and de Werra 1987],
- Space planning and architectural design [Glover, McMillan, and Novick 1985],
- Telecommunications path assignment [Oliveira and Stroud 1989],
- Probabilistic logic problems [Jaumard, Hansen, and Poggi de Aragao 1989],
- Job shop scheduling [Eck 1989],
- Neural network pattern recognition [de Werra and Hertz 1989],
- Machine scheduling [Laguna, Barnes, and Glover 1989b],
- Convoy scheduling [Bovet forthcoming],
- Quadratic assignment problems [Skorin-Kapov 1989],
- Network topology design [Lee 1989],
- Computer channel balancing [Glover 1989],
- Traveling salesman problems [Knox 1989; Malek, Guruswamy, Owens, and Pandya 1989; Malek, Heap, Kapur, and Mourad 1989],
- Graph coloring [Hertz and de Werra 1987],
- Graph partitioning [Wendelin 1988],
- Nonlinear covering [Glover 1986],
- Maximum stable set problems [Friden, Hertz, and de Werra 1989a, 1989b], and
- Flow shop sequencing [Windmer and Hertz forthcoming].

A brief sampling of the outcomes of these studies suggests the potential value of tabu search applied in different settings. Glover and McMillan's [1986] employee scheduling investigation solved problems whose integer programming formulations involved one to four million variables, requiring 22–24 minutes on an IBM PC microcomputer to obtain solutions within 98 percent of an upper bound on optimality. Jaumard, Hansen, and Poggi de Aragao [1989] investigated the problem of determining the consistency of probabilities specifying whether given collections of clauses are true, with extensions to include probability intervals, conditional probabilities, and least perturbations to achieve satisfiability. By integrating a tabu search approach with an exact zero to one nonlinear programming procedure for generating columns of a master linear program, they readily solved problems with up to 140 variables and 300 clauses, approximately tripling on each dimension the size of problems previously solved. In their space planning study, Glover, McMillan, and Novick [1985] applied tabu search to subset clustering problems corresponding to zero-one mixed integer programs with over 25,000 variables and 50,000 constraints, obtaining solutions in less than one minute on a V77 minicomputer. The resulting system has been implemented to improve the architectural design of several large space planning firms.

Tabu search has found solutions superior to the best previously obtained by alternative methods.

By incorporating tabu search in a neural network application, de Werra and Hertz [1989] reduced the number of false attractors (or parasite states) for a visual pattern recognition problem by 80 percent, while requiring only 50 learning trials out of a potential 500,000. In their machine scheduling application Laguna, Barnes, and Glover [1989] obtained optimal solutions to all test problems for which optimality could be verified by specialized branch-and-bound methods and obtained solutions within a few percent of an optimality bound for larger problems (which the branch-and-bound methods could not handle). In a flowshop sequencing study, Windmer and Hertz [forthcoming] compared tabu search to a broad range of specialized heuristics and obtained solutions superior to the best found by any of the other methods for about 90 percent of the test problems. Skorin-Kapov's [1989] quadratic assignment study yielded the best known solutions for problems taken from the literature, while requiring less CPU time than previously reported. The method was used to find a solution superior to the best known for a classical benchmark problem [Steinberg 1961] and obtain solutions whose quality always equalled or surpassed that of solutions obtained by simulating annealing (an outcome also shared by the maximum satisfiability, graph coloring, and traveling salesman studies of Hansen and Jaumard [1987], Hertz and de Werra [1987], and Malek, Guruswamy, Owens, and Pandya [1989]).

These and a variety of other applications of tabu search are surveyed in Glover [1989, 1990], and Hertz and de Werra [forthcoming].

Overview

Tabu search is founded on three primary themes: (1) the use of flexible attribute-based memory structures designed to permit evaluation criteria and historical search information to be exploited more thoroughly than by rigid memory structures (as in branch-and-bound and A* search) or by memoryless systems (as in simulated annealing and other randomized approaches); (2) an associated mechanism of control—for employing the memory structures—based on the interplay between conditions that constrain and free the search process (embodied in tabu restrictions and aspiration criteria); and (3) the incorporation of memory functions of different time spans, from short term to long term, to implement strategies for intensifying and diversifying the search. (Intensification strategies reinforce move combinations and solution features historically found good, while diversification strategies drive the search into new regions.)

The core of tabu search is embedded in its short-term memory process, and many of the strategic considerations underlying this process reappear, amplified in degree but not greatly changed in kind, in the longer-term memory processes.

Short-Term Memory and Aggressive Search

The short-term memory of tabu search constitutes a form of aggressive exploration that seeks to make the best (highest evaluation) move possible, subject to requiring available choices to satisfy certain constraints (Figure 1). These constraints, embodied in the tabu restrictions, are designed to prevent the reversal, or sometimes repetition, of certain moves—by rendering selected attributes of these moves forbidden (tabu). The primary goal of the tabu restrictions is to permit the method to go beyond points of local optimality while still making high quality moves at each step.

Without such restrictions, the method could take a "best" move away from a local optimum (in this case, making a nonimproving move) and then conceivably at the next step fall back into the local optimum by taking the best move available at that point. In general, the tabu restrictions are intended to prevent such cycling behavior and more broadly to induce the search to follow a new trajectory if cycling in a narrower sense occurs (that is, revisiting some earlier solution). These restrictions do not operate in an isolated manner but are counterbalanced by the application of aspiration criteria.

Determining the Best Candidate

A critical step, which embodies the aggressive orientation of short-term memory, is choosing the best admissible candidate (Figure 2). First, each of the moves of the candidate list is evaluated in turn. (The issues of creating and updating candidate lists, which are particularly relevant for layer problems, are discussed in Glover [1989a].) In many settings, the evaluation of a move can be based initially on the change produced in the objective function value (that is, the difference between the objective function values for the solutions before and after applying the move). In other cases, where the ramifications of the move are less easily determined or where not all variables are currently assigned values, the evaluation may be based on generating relaxed or approximate solutions or may simply utilize local measures of attractiveness (as in local decision rules for job-shop scheduling). However, as the search progresses, the form of the evaluation employed by tabu search becomes more adaptive, incorporating reference to intensification and diversification concerns.

The core of tabu search is embedded in its short-term memory process.

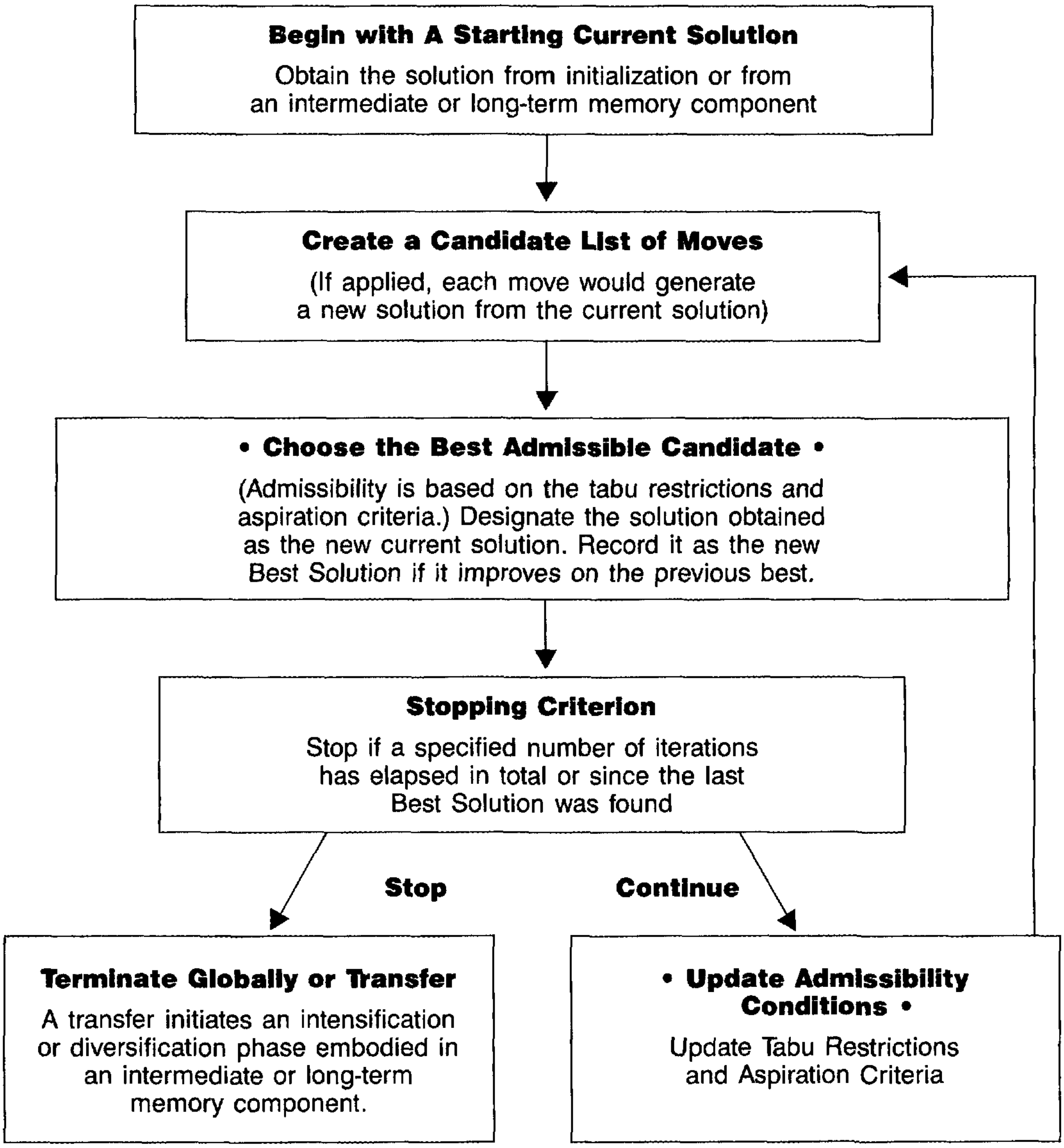

Figure 1: Tabu search short-term memory component.

Because the number of moves classified tabu will generally be small relative to the number available, and assuming the expense of evaluating a move is not great, it is usually preferable to check first whether a given move has a higher evaluation than its admissible predecessors before checking for tabu status. Checking tabu status is the first step in screening for admissibility. If the move is not tabu, it is immediately accepted as admissible; otherwise, the aspiration criteria are given an opportunity to override the tabu status, providing the move a second chance to qualify as admissible.

Examining the next move can be

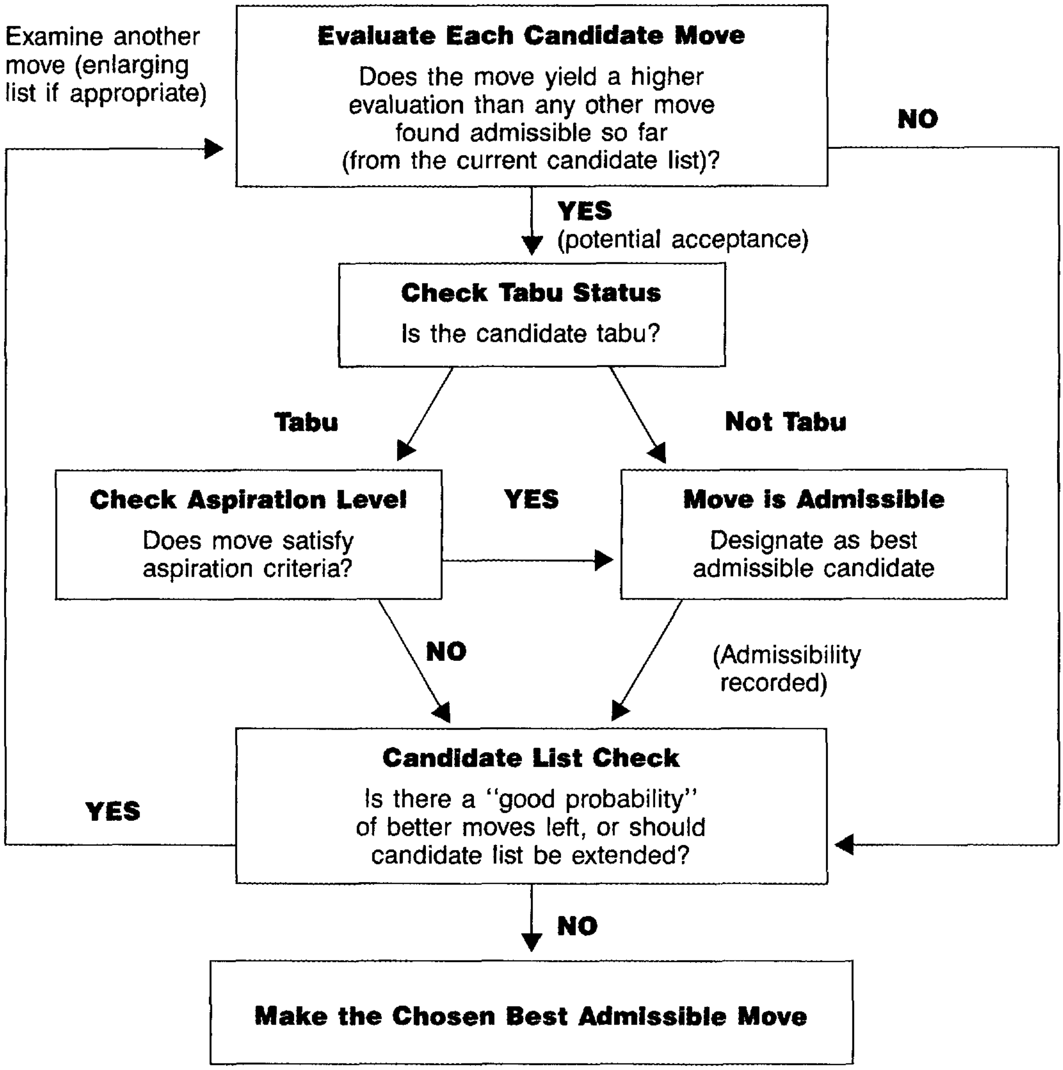

Figure 2: Selecting the best admissible candidate.

embedded in a candidate list strategy. In some cases, if the tabu restrictions and aspiration criteria are sufficiently limiting, none of the available moves will qualify as admissible. A "least inadmissible" move is saved to handle such a possibility and is chosen if no admissible alternatives emerge.

A Simple Illustration of Tabu Search

Consider a minimum cost spanning tree problem that includes constraints that prohibit certain edges from appearing in the tree together, or that allow some edges to appear only if certain other edges also appear. (Without these constraints, the problem could be solved by a straightforward greedy algorithm—for example, iteratively adding the least cost edge that doesn't create a cycle with previous edges until a complete spanning tree results.)

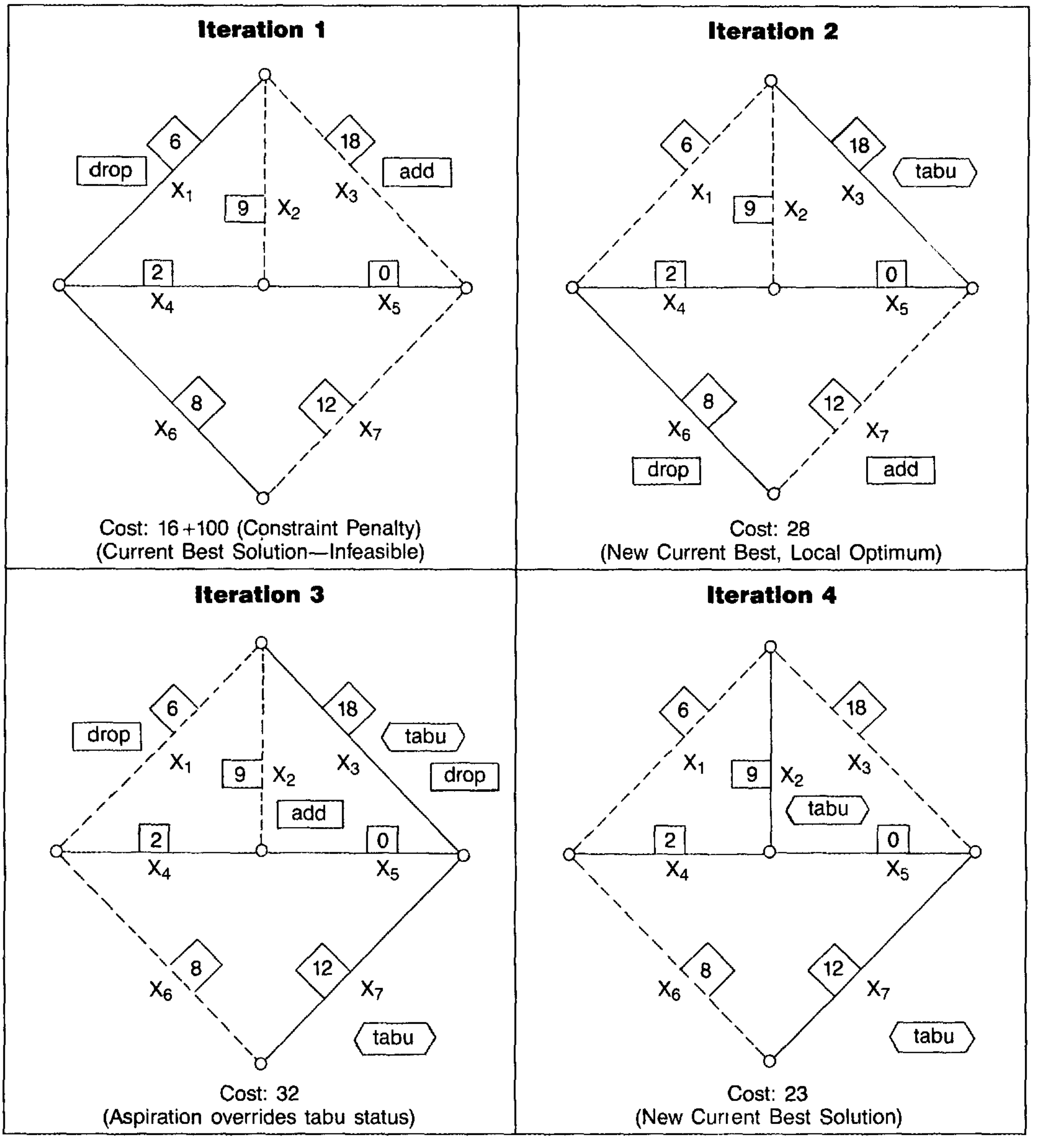

Figure 3: Illustrated solution: minimum-cost trees. The choice rule is select the least-cost admissible "edge swap." The tabu restriction is forbid dropping one of the two most recently added edges (these edges are designated tabu). The aspiration criterion is override the tabu restriction if the swap produces a new "current best solution." (Constraints: $X_1 + X_2 + X_6 \leq 1$, $X_1 \leq X_3$. Violation penalty = 50.)

The example problem is based on a graph that consists of five nodes, hence whose (spanning) trees consist of four edges. Four diagrams of the graph appear in Figure 3, corresponding to four successive iterations of tabu search. Edges of the current tree are shown by solid lines, and remaining edges are shown by dotted lines. The costs of the edges are indicated in the attached boxes and the "names" of the edges ($x_1, x_2, \ldots$) appear immediately opposite. The edge names have an auxiliary role as zero-one variables, where each x_j, $j = 1, \ldots, 7$, is defined by

$$x_j = \begin{cases} 1 \text{ if edge } x_j \text{ is in the tree} \\ 0 \text{ if edge } x_j \text{ is not in the tree.} \end{cases}$$

By this device, the constraints on the tree edges can be expressed in a simple form:

$$x_1 + x_2 + x_6 \leq 1$$

$$x_1 \leq x_3.$$

The first constraint says that at most one of the three edges x_1, x_2 and x_6 is permitted to be in the tree, while the second constraint says that the edge x_1 is allowed in the tree only if edge x_3 is also in the tree.

To permit the evaluation of trees in which these constraints are violated, we have introduced an arbitrarily chosen penalty cost of 50 for each unit of violation. Thus, in the diagram for Iteration 1 in Figure 3, the cost of the tree is indicated to consist of two parts: 16, which is the sum of the edge costs for the tree, and 100, which represents a penalty cost of 50 for a unit violation of each of the two constraints. (Each constraint is violated by one unit because the value of the left side of the constraint exceeds the value of the right side by 1.) The starting tree of Iteration 1 is the minimum cost spanning tree obtained by disregarding the problem constraints, although any other tree could also be used.

To apply tabu search to the example, we have elected to use the standard "edge swap" move that consists of adding an edge and dropping another edge to transform the current tree into a new tree. Such a move is characterized by the fact that the dropped edge always lies in the cycle created by the edge that is added. (For example, on Iteration 1, selecting x_2 to be added would result in dropping either x_1 or x_4, while selecting x_7 to be added would result in dropping x_4, x_5 or x_6.) Following the usual choice rule of the short-term memory component of tabu search, the move selected is an admissible move with the highest evaluation, that is, an admissible move that produces a new tree with the smallest cost (including reference to penalty costs for violating constraints).

To define a tabu restriction, we have singled out the added edge to be the move attribute to be assigned a tabu status (at the moment it is introduced into the tree). This in turn imports a tabu classification to moves that contain the edge by the restriction of forbidding a future move to drop the edge as long as it remains tabu. In the example, we permit only two edges to be tabu at any given time, that is, each added edge remains tabu for two iterations and then is removed from the tabu list (whose length = 2), freeing it from its tabu status.

The aspiration criterion we have selected to override tabu status is the simple one that allows the current move to include a tabu edge if the resulting tree is better than the best tree produced so far.

To trace the operation of tabu search, we will discuss each of the Figure 3 iterations in turn.

Iteration 1: Among the current alternatives for adding and dropping edges to create a new tree, the move that yields the best cost change is to add x_3 and drop x_1. This eliminates the violations of both of the constraints on allowable edges, reducing the penalty term from 100 to 0 while increasing the remaining component of cost from 16 to 28 (adding 18-6, the difference in the costs of the added and dropped edges).

Iteration 2: By the chosen rule for identifying the move attribute to be made tabu, the edge x_3 added by the move of Iteration 1 acquires a tabu status, thereby in turn imparting a tabu status to moves that drop this edge. Among the moves remaining, the best cost change is created by adding edge x_7 and dropping edge x_6. (The currently admissible moves that appear to produce a better cost tree also result in violating a constraint, incurring a penalty that gives them an inferior evaluation.) This move is also more attractive than the tabu moves in the present case, illustrating that the tabu restrictions do not always affect the preferred choice. The selected move also worsens the cost of the tree, however, indicating that the current tree is a local optimum, since no available move leads to a better solution. (In some implementations, tabu lists are not activated until a first local optimum has been reached.)

Iteration 3: Edge x_7, added to the tree by the move of Iteration 2, joins x_3 in becoming tabu. At this point, a new situation emerges. The best of the available moves is to add edge x_2 and drop edge x_3, a move that normally would be disallowed since x_3 is tabu. However, the move satisfies the aspiration criterion by producing a tree with a better cost than obtained so far, and consequently the move is selected as indicated.

Iteration 4: Edge x_2 joins edge x_7 to constitute the two most recently added edges and hence is designated tabu. (If x_3 had not been removed from the tree on the preceding move, this edge would still have been released from its tabu status since the tabu list consists of only two elements in this illustration.) The current tree is a new local optimum and also is the new current best.

The move with the highest evaluation is now to add x_3 and drop x_2, but this move is tabu (since edge x_2 is tabu) and fails to satisfy the aspiration criterion. (In fact, if chosen, this move would reverse the move most recently made and return to the tree of Iteration 3.) The move that adds x_3 and drops x_5 is the best move that also qualifies as admissible and is the one selected. On the next iteration, not shown, the two tabu edges would consist of x_2 and x_3, and the method would continue in this fashion until a desired iteration cutoff was reached.

In the preceding example, the tree obtained at Iteration 4 in fact constitutes a global optimum. Without a supplementary process for generating bounds or conducting some other type of verification, however, this outcome would not be known. From a practical standpoint, the absence of a theoretical guarantee of finding (and verifying) optimal solutions usually does not constitute a limitation for combinatorial optimization problems, since such guarantees are based on the unrealistic assumption that exponentially large amounts of computational effort are permissible. On the other hand, tabu search can also be integrated with methods containing optimality guarantees to improve their performance, providing an avenue of research that merits further investigation. (An effective integration of tabu search with branch and bound has been carried out by Friden, Hertz, and de Werra [1989b], and the introduction of cutting planes—including pseudo cuts that can be discarded with the expiration of associated tabu tenures—constitutes an associated area that warrants exploration.)

Related Considerations and Preliminary Guidelines

The preceding example leads directly to considerations for creating additional types of tabu restrictions and aspiration criteria. Instead of a tabu restriction preventing added edges from being dropped, for example, it would be equally possible to prevent dropped edges from being added or to prevent both types of reversals simultaneously. It is worthwhile to consider the conditions under which different alternatives may be preferable.

In applications involving drop-add moves, the number of elements available to be dropped from a solution (represented by edges of a tree, elements of a set, or variables of a basis, and so forth) are typically somewhat fewer than the number available to be added. When this occurs, a tabu restriction that prevents previously dropped elements from being added back to a solution allows a greater degree of flexibility than a restriction preventing added elements from being dropped. Experimental evidence indicates this type of flexibility is generally desirable, leading to our first guideline.

Guideline 1: When tabu restrictions are based on a single type of move attribute, it is generally preferable to select an attribute whose tabu status less rigidly restricts the choice of available moves. This guideline deserves to be qualified under certain circumstances. It appears preferable to avoid (or supplement) a tabu restriction whose degree of flexibility encourages the generation of consecutive solutions that are all accessible to a common earlier solution. For example, in a job sequencing application, a restriction preventing a job from returning to a previous position may still allow the job to move to other positions reachable from the initial position. The phenomenon of repeatedly shuffling among these neighboring alternatives generally constitutes an inefficient search process and hence should be countered by an additional tabu restriction (enforced for a smaller number of iterations) requiring the job not to move away from a position just reached. (The removal of such a restriction can be made conditional on repositioning another job that creates a new alternative for the given job.) A similar phenomenon is possible when restricting attention to a single type of move attribute from the class of drop-add moves; we will provide a guideline for handling this issue.

An important feature of tabu search is the ability to locate a robust range of tabu list sizes by preliminary empirical testing for a given class of problems that give the best results for any particular attribute and associated tabu restriction. As a result it is easy to verify experimentally the type of attributes and tabu restrictions that perform most effectively.

In several of the early applications of tabu search, the best tabu list sizes consistently fell in the interval from five to 12, with seven representing a highly effective value. This outcome has encouraged some speculation about connections between preferred tabu list sizes and the number of items normally retained in human short-term memory. An interesting supposition, for example, is that evolution may have discovered that a short-term memory in the neighborhood of seven elements is effective for problem solving. (It may be preferable in some systems, for example, those biologically derived, to forego a more extensive short-term memory and instead devote more machinery to abstraction processes such as "chunking"—analogous to identifying relevant move attributes in tabu search.) More recently, experimentation has uncovered applications where preferred tabu list sizes lie in intervals related to problem dimension instead of being linked to the magic number seven. As a general principle, tabu restrictions that are more stringent, as measured by the degree to which they limit the range of admissible moves, lead to somewhat smaller values for best tabu list sizes than restrictions that are less stringent. (Human beings no doubt select types of attributes for creating abstractions that are inherently balanced with the size of short-term memory to enable effective problem solving. Since people selected the attributes for defining the first tabu restrictions based on their intuition about elements that should be included, it is probably not surprising that the associated tabu list sizes took the values they did.)

Such speculations aside, the tabu search processes illustrated are susceptible to elaborations. Prominent among these is the use of multiple tabu lists, each devoted to a particular type of attribute.

Guideline 2: Incorporate separate but parallel lists for different attribute types, where the sizes of these lists reflect the relative differences in constraining the number of available moves by the tabu restrictions associated with these attribute types.

This guideline is particularly relevant to moves whose attribute types are differentiated by dropping and adding elements to sets but also has application to the use of dependent attributes, such as values of the objective function or of selected partial sums of variables. Dependent attributes can be created strategically by functions designed for this purpose, analogous to the use of hashing functions to avoid duplicate references in data base searches, as suggested by Hansen and Jaumard [1987]. An exploration by Woodruff and Spearman [1990] discloses the potential merit of this approach, but also suggests, in common with the current guidelines, that the form of dependent attributes relevant to searching solution spaces involves considerations beyond those encountered by customary applications of hashing.

To date, multiple tabu lists have been used primarily to maintain separate lists for different solution processes or to prevent repetitions as well as reversals. Hence, Guideline 2 invites future exploration. When different types of attributes are handled in this fashion, they can be given varying weights, depending on their classification and age, to determine the tabu status of moves that contain them. The recent development of dynamic tabu list processes [Glover 1990] examines such alternatives within a more formal context.

Aspiration Criteria

The minimum cost tree illustration discussed earlier treats only the most primitive form of aspiration criteria, and other possibilities warrant consideration. In general, each selected attribute of a move (such as the added and dropped edges of the example) can have one or more aspiration criteria of its own, based on the best solution that includes (or excludes) that attribute as a member. Like tabu restrictions, aspiration criteria can be given "tenures," that is, made time dependent.

In this type of approach, an aspiration criterion can be applied to a tabu attribute during the period that it remains tabu, overriding its tabu status if a solution is obtained that improves on the one immediately before creating this status (to avoid returning to this previous solution). Such aspiration criteria can have subtle consequences and ideally should be updated by special rules.

Intermediate and long-term memory operate primarily as a basis of strategies for intensifying and diversifying the search.

The relevant concerns can be illustrated by a simple example. Consider a minimum cost zero-one integer programming problem where moves consist of changing the values of variables from 0 to 1 or from 1 to 0. Suppose the variable x_1 is changed from 0 to 1, starting from a solution with a cost of 100, and then in turn x_2 is changed from 0 to 1, now starting from a solution with a cost of 120. (The aspiration criteria for these two moves thus will permit them to be reversed only by producing costs better than 100 and 120, respectively.) Subsequently, if x_1 is changed from 1 back to 0 (for example, by satisfying its aspiration criterion), the effect as far as the move for x_2 is concerned is the same as if the move changing x_1 from 0 to 1 had never occurred. Specifically, it may now be possible to change x_2 from 1 back to 0, yielding a cost of 100 (permitted by the aspiration cost of 120), with the potential result of duplicating the solution that inaugurated the move for x_1. In short, the reversal of the move for x_1 implies the aspiration criterion for x_2 should be changed from 120 to 100, and associated adjustments should be made for aspiration values of moves that followed the move for x_2. (These adjustments may be based on an approximating rule such as decreasing all such aspiration values by 20, or by amounts that diminish from 20 to 0, since the aspiration value for the most recent move is guaranteed to be accurate.)

A simple use of such aspiration criteria that involves no updating (other than to initiate and terminate aspiration levels at the start and end of their tenures) has proved effective in application to the traveling salesman problem [Knox and Glover 1989]. Together with the use of multiple tabu lists incorporating different attributes, this provides an area open to fruitful investigation and leads to the next guideline.

Guideline 3: Embody the treatment of aspiration criteria in an attribute-based framework analogous to that used to define tabu restrictions (employing the same or different attribute types).

Intermediate and Long-Term Memory: Intensification and Diversification Trade-offs

In many applications, the short-term

memory component by itself has produced solutions superior to those found by alternative procedures, and the use of longer-term memory in these cases has been bypassed. However, longer-term memory can be important for obtaining best results for hard problems [Malek, Guruswamy, Owens, and Pandya 1989; Skorin-Kapov 1989]. The modular form of the process makes it easy to create and test the short-term memory component first and then to incorporate the remaining components if additional refinement is desired.

Intermediate and long-term memory operate primarily as a basis of strategies for intensifying and diversifying the search. In fact, the fundamental elements of intensification and diversification strategies are already present in the short-term memory component of tabu search, since a short-term memory tabu list has an intensification role by temporarily locking in certain locally attractive attributes (those belonging to moves recently evaluated to be good), while it also has a diversification role by compelling new choices to introduce (or exclude) attributes that are not among those recently discarded (or incorporated).

The fact that different attributes, such as add and drop attributes, can create different types of intensification and diversification effects provides a further argument for creating parallel tabu lists (of different sizes) for handling such attributes—that is, their associated tabu restrictions—in concert.

An Example of Longer-Term Concerns

We will discuss an example that discloses the relevance of longer-term memory, emphasizing its role in creating a diversification strategy. Once again we refer to a minimum cost tree problem attended by additional constraints. We assume that the tabu restrictions, tabu list size, and aspiration criteria are the same as in the earlier example.

The beginning solution of Figure 4 is obtained by solving a minimum cost spanning tree problem without reference to the two added constraints, $x_9 \leq x_7$ and $x_3 + x_7 \leq 2x_5$. The second constraint (which effectively stipulates that edges x_3 and x_7 cannot be in the tree unless edge x_5 also is in the tree) is violated by two units, hence incurs a penalty cost of 100. As moves are made leading away from this initial solution, it is more attractive to drop edges x_3 and x_7 than to introduce the high cost edge x_5. Also, once a feasible solution is obtained, from which x_5 is excluded, edge x_5 remains unattractive as a candidate to be introduced. The limitation we have accepted for tabu list size is insufficient, moreover, to cause a move that adds x_5 to rise to the top of the nontabu alternatives. (A larger list size in general would not remedy this type of situation, because it can render other good moves harder to find. Also, at some point, such a larger list no longer qualifies as a form of short-term memory.)

Thus, the tabu search procedure that relies only on the short-term memory component fails to discover the right move to reach the optimal solution—that is, it fails to induce sufficient diversification to drive the process into an appropriate new region. For a problem of this simplicity, there are a variety of ways to escape from the trap that prevents access to the optimal solution (for example, employing a move evaluator based on Lagrangian considera-

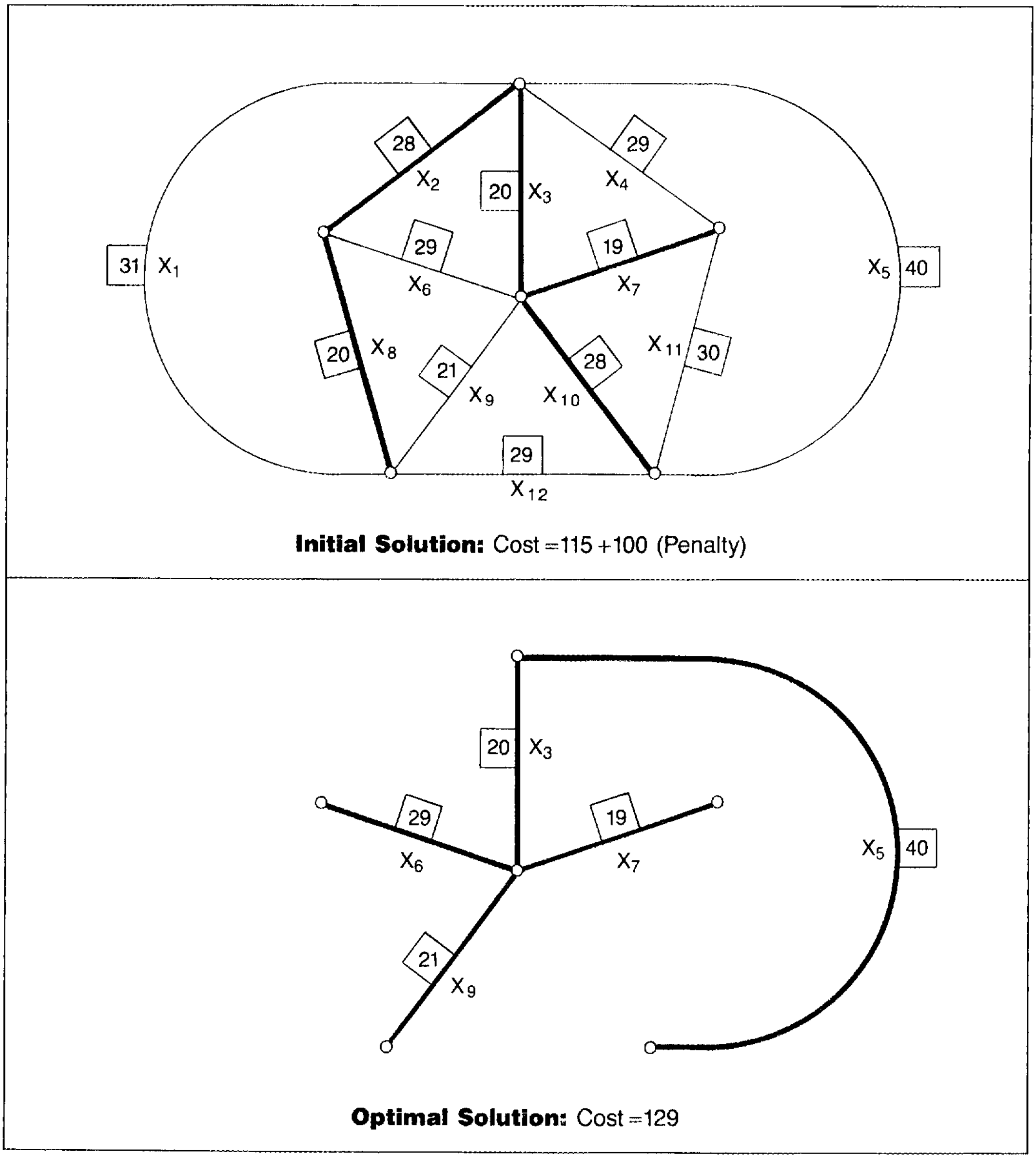

Figure 4: Relevance of longer-term memory and diversification minimum-cost tree problem. Added constraints are $X_9 \leqslant X_7$, $X_3 + X_7 \leqslant 2X_5$. (Unit violation penalty = 50.)

tions, or incorporating slightly more complex types of moves). However, the principle illustrated by the example remains applicable in settings where easy remedies are unavailable.

In the present climate of problem-solving methodologies, an instinctive response in this type of situation is to turn to randomization as an attempt to uncover an effective move by the operation of good fortune. Such an approach is certainly possible, but it is also a more haphazard means of achieving diversity than an approach based on strategic use of memory.

Moreover, randomization loses the counterbalancing effect of continuing to pursue good moves, following more than blind selection, in addition to incorporating a diversification objective. (A partial compromise is embodied in the probabilistic variant of tabu search, which generates probabilities for selecting moves and, like the deterministic form, establishes priorities based both on move evaluations and the tabu search memory structures. Faigle and Kern [1989] show that determining probabilities in this way has a mathematical as well as intuitive justification.)

Links to a Learning Process

An intermediate term memory procedure incorporates features of both intensification and diversification results by establishing a historical standard for differentiating the quality of alternative moves. A particular evaluator, such as one based on the change in objective function values, can vary in its accuracy of identifying good moves, depending on the current solution (or search state). This point has been brought home dramatically by the learning approach called target analysis [Glover and Greenberg 1989], which offers a useful means for developing evaluators to support the intensification and diversification strategies of tabu search.

The basis of the target analysis approach is to invest extensive preliminary effort to determine optimal or near optimal solutions, called target solutions, to representative problems from a given class. (Such effort is allowed to be considerably greater than would be employed on a routine solution attempt.) Subsequently, during a sequence of follow-up phases, these problems are re-solved using the target solutions to evaluate the evaluators. This is done by creating scores that rate the moves by their ability to lead to the target solutions, basing these scores on the change produced by the moves in the discrepancy between the current solutions and the target solutions. These scores then disclose when different potential evaluators succeed or fail in identifying good moves and lead to identifying information that can be used to create improved evaluators.

In a study applying target analysis to a tabu search method for machine scheduling Glover and Laguna [1989] found that the standard objective function evaluator worked well in identifying the quality of moves on iterations where moves existed that improved the current solution, but performed poorly when such moves were absent. This knowledge was used to create an intensification strategy that maintained history of good move attributes, characterized by their membership in the best solutions found in the past. These attributes (expressed in this case as relative positions of tasks scheduled on a machine) then became the basis for creating an alternative proxy target evaluator, which supplemented the standard evaluator when all admissible moves were nonimproving. The result was to obtain best known solutions more quickly for smaller problems and to obtain new solutions of higher average quality for larger problems.

By extension, such an approach can be elaborated to incorporate elements of diversification as well as intensification, with an ability to shift the emphasis between the two. This can be accomplished by assigning high ratings to attributes contained in moves that received good evaluations under conditions where such evaluations were reliable, as determined by target analysis. (Such ratings include a consistency factor, according to how often particular attributes belong to high evaluation moves.) Since not all high evaluation moves that qualify as reliable are selected during the solution process, some of their associated attributes do not become (or rarely become) incorporated into the solutions generated. Accordingly, this leads to the following guideline.

Guideline 4: To combine the diversification and intensification goals, create a rating system by reference to target analysis, maintaining records of highly rated attributes and of how often these attributes appear in solutions generated. On iterations where conditions derived from target analysis disclose standard evaluations to be less reliable, supplement these evaluations by favoring attributes with high historical ratings that have less frequently occurred in previous solutions.

In applying the preceding guideline, the relative stress on attribute ratings versus the frequency of being incorporated into solutions (by belonging to moves actually selected) provides a means for exploring a range of intensification/diversification trade-offs.

One of the possible ways of implementing Guideline 4 may be illustrated as follows: move attributes may be divided into six frequency classes according to whether these attributes (1) often occur in good (or very good) solutions; (2) often occur in poor solutions but rarely in good solutions; (3) often occur in moves to add the attribute to the current solution, where these moves receive evaluations that are high, but not high enough to be chosen; (4) often occur in moves to drop the attribute from the current solution, where these moves similarly receive evaluations insufficiently attractive to be chosen; (5) often occur in the solutions actually generated during the search process (whether good or bad); (6) often do not occur in solutions generated.

Class (1) and (2) attributes can be used to support intensification goals by selecting moves to add and drop such attributes, respectively, from solution. Class (3) and (4) attributes combine the elements of intensification and diversification by these same respective strategies. Finally, in reverse, moves that drop class (5) attributes and add class (6) attributes serve to emphasize diversification concerns. Target analysis may be used to define the thresholds implied by terms such as *often* and *good*, instead of resorting to arbitrary choices or calibration efforts based on trial and error. Other types of classifications are of course possible, including those that involve conditional relationships. The use of attribute-based memory in tabu search leads naturally to the parallel concept of attribute-based evaluations, as embodied in the foregoing longer-term memory strategies.

Diversification Based on Move Distance

A diversification strategy is particularly relevant in situations where the best solutions can be reached only by crossing certain humps, involving the choice of moves with inferior evaluations. To identify appropriate moves to negotiate such humps, a memory function can be created to classify the relative attractiveness of moves within a given distance class.

The notion of move distance derives

from the fact that some moves create greater changes in the current solution than others. For example, in a scheduling application, a move that transfers a task to a new sequence location several positions away from its current position involves a greater move distance than one that transfers a task to an adjacent location. (Such a distance measure may be based on elements other than position, for example, the sum of processing times of intervening tasks or the effect on secondary elements whose relationship may be altered by the repositioning.) Similarly, in an integer programming context, the degree to which a move changes the relative feasibility (or infeasibility) of certain constraints or alters the value of certain dependent variables can be made the basis for defining a distance measure.

Such a measure derives its significance from the following anticipated correlation: moves that involve greater distances are likely to entail greater cost, as determined by a standard evaluator, than moves that cover smaller distances. This implies that large distance moves are likely to appear relatively unattractive by such evaluators and hence also are likely to be among the moves rarely chosen, which may be necessary to cross humps to better solutions. Accordingly, historical information can be used to determine when an evaluation for a large distance (or intermediate distance) move is in fact attractive for members of its distance class, regardless of how the evaluation compares to the evaluation of smaller distance moves. Selecting preferred moves from infrequently sampled classes therefore provides a useful form of diversification.

Moves that induce greater solution changes should characteristically be applied when standard evaluators lose their effectiveness (in the simplest case, where they fail to offer direct improvement). Moves of lesser impact may accordingly be assigned the function of locally tuning solutions generated by larger distance moves.

Target analysis again provides a means for refining such a strategy, leading to a determination of such issues as (1) how far above the historical class average (or how close to the historical class best) a particular evaluation should be to qualify as preferred; (2) when the evaluation of a larger distance move should be considered superior to the evaluation of a smaller distance move; (3) what length of time a tabu condition should operate to prevent moves in different distance classes from being reversed. More precisely, parameters such as highest, average, and lowest evaluations for moves of a given class can be generated at each iteration and then accumulated over an interval of test iterations to identify extreme and central values for each (and for such associated quantities as spreads between these parameters). This information can be exploited by using scores derived from target analysis to establish relationships between the single iteration parameters and the historical statistics to identify which moves should qualify as good. This approach may be expressed as follows:

Guideline 5: Devote a limited number of preliminary iterations to identify historical statistics for move evaluations in each distance class, and apply relationships from target analysis to obtain effective choices at subsequent iterations (by comparing the values of associated single iteration parameters to these statistics).

This guideline can be applied to generate tabu sizes, or more broadly tabu measures, that can vary from iteration to iteration. A relevant variation of the distance concept in such an application is to consider moves that not only influence the current solution, but that provide gateways to other moves, that is, whose selection opens up alternatives not frequently accessible during the previous course of search.

These notions can have useful implications for parallel processing. Specifically, larger distance moves often will be more difficult to evaluate accurately than smaller distance moves, precisely because they engender or create the opportunity for greater change. However, it may be possible to identify a small set of preferred large distance moves, at least one of which will be good (in a sense established by target analysis), without precisely knowing which move should qualify as the good one. Applying parallel processing to examine the consequences of these preferred alternatives provides a more refined basis for selecting among them. Such an approach is exemplified in a geometric application in the multi-leveling procedure of Ron [1988]. In cases where it may be hard to return to states once accessible by alternate moves not taken, such a use of parallel processing can prove particularly beneficial. (Glover [1990] discusses memory and guidance structures to facilitate the use of return strategies, where these are appropriate, in the context of move restructuring.)

Diversification and Restarting

One form of diversification strictly involves longer term memory considerations. A common means of attempting to improve the performance of heuristic methods is to restart the solution process from different solutions generated randomly or by a set of favored starting heuristics. The generation of new starting solutions is a key area to be explored by a more systematic diversification strategy based on a long-term memory component.

A straightforward approach has been found highly effective in application to the traveling salesman problem [Malek, Guruswamy, Owens, and Pandya 1989] and the quadratic assignment problem [Skorin-Kapov 1989]. The basis of this approach is a frequency-derived strategy similar to that suggested in Guideline 4 for intermediate term intensification and diversification, but which focuses more specifically on producing initial solutions as different as possible from the solutions generated throughout the previous history of the search process [Glover 1986, 1989]. To do this, a count is maintained for each solution element or assignment (the assignment of a job to a position, or of a value to a variable, and so forth) identifying the number of times this element or assignment occurs over all solutions previously encountered. When a new starting solution is generated, the frequency counts are used to penalize the selection of their associated elements, thus favoring moves that introduce elements excluded from (or less frequently incorporated in) earlier solutions. Such a strategy may also be activated without restarting, driving the search away from its present vicinity for a specific distance or a selected number of iterations. These observations may be summarized as follows:

Guideline 6: For longer-term diversification, employ frequency-derived penalties to drive the search away from solutions previously encountered either by restarting or by progressing from the current solution.

Applications to date, which have centered on the restarting aspect of the preceding guideline, have utilized only a single restart, disclosing that even a limited form of the approach can be effective. It will be challenging to use the lessons learned from such applications to generate diverse starting points for parallel processing, with successive communication between parallel streams to capture appropriate intensification/diversification trade-offs.

There are special contexts, as in certain types of scheduling or loading problems, where the search process should preferably consist of successive waves of construction from alternative levels or stages (including the stage that entails reconstructing the entire solution) and where tabu restrictions should appropriately focus on preventing repetitions rather than reversals. Such settings lead to interdependent tabu lists that are similarly staged [Glover 1990] (generally based on attributes expressing sequential dependence), providing a search process that is particularly suited to the application of candidate list strategies [Glover 1989]. Frequency-derived penalties likewise can be applied to these settings in order to influence diversification on wider time scales.

Finally, an associated area for examination consists of integrating the frequency-derived penalties (which induce a form of tabu status) with the recency-derived penalties that result by amplifying the influence of a tabu perturbation function. The integration of both recency- and frequency-derived factors in human long-term memory suggests that determining an effective combination of the two may prove better than either in isolation. The preferential treatment of higher evaluation elements in memory (by incorporating intensification concerns) also provides a quality-derived dimension to memory, automatically supplementing the dimensions based on recency and frequency.

Conclusion

The rapidly growing and highly effective applications of tabu search suggest the useful potential of this approach and its underlying principles. At the same time, it is apparent that the studies to date have only taken the first steps in exploring this potential. Many more applications remain to be undertaken, and many new possibilities for refining the basic processes of the method remain to be tested.

At the most basic levels, the ability to launch simple implementations of tabu search with relatively small effort and to build on these as desired makes the approach convenient for carrying out preliminary investigations. As additional refinements are undertaken, the use of learning procedures, such as target analysis, provide an opportunity to more fully exploit the two key polarities within tabu search—embodied in the interplay between tabu restrictions and aspiration criteria, and between intensification and diversification strategies. These efforts should lead to increasingly effective variations and should open up new areas of research and implementation.

Acknowledgment

This work has been supported in part under NASA Contract NAGW-1388 with the Center for Space Construction at the University of Colorado, Boulder. I will be happy to supply an expanded version of this paper on request.

References

Bovet, J.; Constantin, C.; and de Werra, D forthcoming, "A convoy scheduling problem," *Discrete Applied Mathematics.*

Eck, B. T. 1989, "Good solutions to job shop scheduling problems via tabu search," research report, Department of Industrial Engineering and Operations Research, Columbia University, New York, May.

Faigle, U. and Kern, W. 1989, "Some convergence results for probabilistic tabu search," research memorandum No. 882, Faculty of Applied Mathematics, University of Twente, The Netherlands, November.

Friden, C.; Hertz, A ; and de Werra, D. 1989a, "Stabulus: A technique for finding stable sets in large graphs with tabu search," *Computing,* No. 42, pp. 35–44.

Friden, C.; Hertz, A.; and de Werra, D. 1989b, "TABARIS: An exact algorithms based on tabu search for finding a maximum independent set in a graph," working paper, Swiss Federal Institute of Technology, Lausanne, February.

Glover, F. 1986, "Future paths for integer programming and links to artificial intelligence," *Computers and Operations Research,* Vol. 13, No. 5, pp. 533–549.

Glover, F. 1989, "Tabu search, Part I," *ORSA Journal on Computing,* Vol. 1, No. 3, pp. 190–206.

Glover, F 1990, "Tabu search, Part II," *ORSA Journal on Computing,* Vol. 2, No. 1, pp. 4–32.

Glover, F. 1989, "Candidate list strategies and tabu search," CAAI Research Report, University of Colorado, Boulder, July.

Glover, F. and Greenberg, H. J. 1989, "New approaches for heuristic search. A bilateral linkage with artificial intelligence," *European Journal of Operational Research,* Vol. 39, No 2, pp. 119–130.

Glover, F. and Laguna, M 1989, "Target analysis to improve a tabu search method for machine scheduling," technical report, Advanced Knowledge Research Group, US West Advanced Technologies, Boulder, Colorado, September.

Glover, F. and McMillan, C. 1986, "The general employee scheduling problem: An integration of management science and artificial intelligence," *Computers and Operations Research,* Vol. 13, No. 5, pp. 563–593.

Glover, F.; McMillan, C.; and Novick, B. 1985, "Interactive decision software and computer graphics for architectural and space planning," *Annals of Operations Research,* Vol. 5, ed. C. Monma, pp. 557–573.

Glover, R.; Klingman, D.; and Phillips, N. V. 1989, "A network-related nuclear plant model with an intelligence branch-and-bound solution approach," *Annals of Operations Research Linkages with Artificial Intelligence.*

Hansen, P. and Jaumard, B. 1987, "Algorithms for the maximum satisfiability problem," RUTCOR Research Report RR#43-87, Rutgers University, New Brunswick, New Jersey.

Hertz, A. and de Werra, D. 1987, "Using tabu search techniques for graph coloring," *Computing,* Vol. 29, pp. 345–351.

Hertz, A. and de Werra, D. forthcoming, "The tabu search metaheuristic: How we used it," *Annals of Mathematics and Artificial Intelligence.*

Jaumard, B.; Hansen, P.; and Poggi de Aragao, M. 1989, "Column generation methods for probabilistic logic," Gerard, G-89-40, McGill University, November.

Knox, J. 1989, "The application of tabu search to the symmetric traveling salesman problem," PhD diss., Graduate School of Business, University of Colorado, July.

Knox, J. and Glover, F. 1989, "Comparative testing of traveling salesman heuristics derived from tabu search, genetic algorithms and simulated annealing," Center for Applied Artificial Intelligence, University of Colorado, September.

Laguna, M.; Barnes, J. W.; and Glover, F. 1989a, "Scheduling jobs with linear delay penalties and sequence dependent setup costs using tabu search," research report, Department of Mechanical Engineering, University of Texas-Austin, April.

Laguna, M.; Barnes, J. W.; and Glover, F.

1989b, "A tabu search method for scheduling jobs on parallel processors," research report, Department of Mechanical Engineering, University of Texas-Austin, November.

Lee, M 1989, "Least-cost network topology design for a new service using tabu search," *Heuristics for Combinatorial Optimization*, Section 6, pp. 1–18.

Malek, M.; Guruswamy, M.; Owens, H.; and Pandya, M. 1989, "Serial and parallel search techniques for the traveling salesman problem," *Annals of OR: Linkages with Artificial Intelligence*.

Malek, M.; Heap, M.; Kapur, R.; and Mourad, A. 1989, "A fault tolerant implementation of the traveling salesman problem," research report, Department of Electrical and Computer Engineering, University of Texas-Austin, May.

Oliveira, S. and Stroud, G. 1989, "A parallel version of tabu search and the path assignment problem," *Heuristics for Combinatorial Optimization*, Section 4, pp 1–24.

Ron, D. 1988, "Development of fast numerical solvers for problems in optimization and statistical mechanics," PhD diss., Department of Applied Mathematics, The Wiezmann Institute of Science, Rehovot, Israel.

Ryan, J., ed. 1989, "Final report of mathematics clinic," *Heuristics for Combinatorial Optimization*, sponsored by US West Advanced Technologies and the University of Colorado-Denver, June.

Skorin-Kapov, J. 1989, "Tabu search applied to the quadratic assignment problem," *ORSA Journal on Computing*, Vol. 2, No. 1 (May), pp. 33–45.

Steinberg, L. 1961, "The backboard wiring problem: A placement algorithm," *SIAM Review*, Vol. 3, pp. 37–50.

Wendelin, C. 1988, "Graph partitioning with the aid of the tabu method," project report, Institute für Hohere Studien, Vienna, Austria, June.

de Werra, D. and Hertz, A 1989, "Tabu search techniques: A tutorial and an application to neural networks," *OR Spectrum*, Vol. 11, pp. 131–141.

Windmer, M. and Hertz, A. forthcoming, "A new approach for solving the flowshop sequencing problem," *European Journal of Operational Research*.

Woodruff, D. L. and Spearman, M. L. 1990, "Sequencing and batching for two classes of jobs with due dates and setup times," working paper, Northwestern University, March.