Artificial Intelligence, Heuristic Frameworks and Tabu Search

Fred Glover

Center for Applied Artificial Intelligence, Graduate School of Business, University of Colorado, Boulder, CO, USA

This paper examines some of the characteristics of AI-based heuristic procedures that have emerged as frameworks for solving difficult optimization problems. Consideration of attributes shared to some degree by human problem solvers leads to focusing in greater detail on one of the more successful procedures, tabu search, which employs a flexible memory system (in contrast to 'memoryless' systems, as in simulated annealing and genetic algorithms, and rigid memory systems as in branch and bound and A* search). Specific attention is given to the short-term memory component of tabu search, which has provided solutions superior to the best obtained by other methods for a variety of problems. Our development emphasizes the principles underlying the interplay between restricting the search to avoid unproductive retracing of paths (by means of tabu conditions) and freeing the search to explore otherwise forbidden avenues (by aspiration criteria). Finally, we discuss briefly the relevance of a supplementary framework, called target analysis, which is a method for determining good decision rules to enable heuristics to perform more effectively.

INTRODUCTION

Heuristic approaches to optimization problems abound, and many claim some connection to artificial intelligence. Generally speaking, good heuristic procedures are based on ideas that can trace their origins equally to the fields of artificial intelligence and operations research. (This is not too surprising, since the two fields emerged from common beginnings.) Nevertheless, whether due to differences of emphasis or to differences in the supply of 'congenial metaphors', recent heuristic innovations have tended frequently to become aligned with AI. Hence in order to investigate what is current in heuristic ideas it is appropriate to examine procedures that have acquired some of the imprint of the AI domain.

Given the proliferation of heuristic procedures in this category, there is some challenge to identifying those that are more significant, or that at least embody principles that have widespread utility. Currently, four methods that are perceived as affiliated in some measure with the AI field have gained prominence as *frameworks* for solving difficult problems: neural networks, simulated annealing, genetic algorithms and tabu search.

Neural networks have claimed intriguing successes in pattern-recognition applications, but have generally performed less than impressively in optimization settings. They have demonstrated their primary value for problems whose structures can be exploited by processes likened to those of 'associative memory', and appear less well adapted (so far) to the solution of optimization problems in broader categories. While these approaches have a great deal that is inherently fascinating about them, particularly concerning the directions in which they may evolve, neural networks appear to show greatest promise in partnership with other methods, and are coming to rely on one of the other three heuristic frameworks to improve their effectiveness in a number of applications.

Simulated annealing and genetic algorithms draw on analogies to phenomena in the physical and biological sciences, respectively, and have the attractive feature of assured convergence under appropriate assumptions. It should be cautioned, however, that convergence in these procedures takes a less than impressive form, couched in probabilistic assertions. In simulated annealing, for example, optimality is guaranteed to be achieved with probability 1 after an infinite number of iterations. The guarantee offered by genetic algorithms is likewise probabilistic and refers only to certain classes of "undominated" solutions. Consequently, members of the AI and OR communities who are

0143–6570/90/050365–11$05.50

less religiously attached to mathematical demonstration (e.g. who would like to see convergence take place in the world we know) may prefer to place greater stock in the empirical performance of such approaches than in their theoretical underpinnings. Fortunately, a number of instances of effective empirical performance have been documented, attesting to the value of these procedures as practical tools.

From an AI standpoint, one of the more interesting features of simulated annealing and genetic algorithms is that they are virtually memoryless. Each operates by using a form of threshold to allow randomly sampled moves to alter current solutions, and to initiate subsequent iterations of search. Memory has no role except as implicit in the structure of solutions generated from one stage to the next as a result of progressively applied screening criteria.

Mathematically, there is a good reason for this. At our present level of mathematical development we are unable to provide theorem-proof demonstrations for the behavior of systems that embody memory, except in its most trivial or most rigid forms. Leading in popularity are the systems with no memory whatsoever, which simply involve rules for transforming a present state into a successor, closely followed in popularity by the 'highly structured' memory systems exemplified by branch and bound, A* search, and their relatives.

Beyond the appeal of being susceptible to mathematical analysis, memoryless systems have the attraction of appearing to effectively serve certain realms of physics and biology (at a stage of evolution that precedes the development of a complex brain), while rigid memory systems appeal to notions of orderliness (as manifested in what may be called a *search-by-bookkeeping* orientation).

To the extent that AI motivates a notion of intelligence that involves freer reign with the use of memory, however, we may conceive it worthwhile to explore frameworks that embrace more flexible memory structures. Mathematics may not yet be able to justify certain forms of intelligence, but we may suspect that this should not compel us to abandon such forms of intelligence in building solution methods.

This latter view provides the perspective adopted in this paper—a perspective that also underlies the development of the fourth heuristic framework, tabu search. Consequently, the class of approaches embodied within tabu search will be the chief focus of the material that follows. In addition, we will briefly offer a fifth framework for consideration, called *target analysis*, which is not a heuristic solution procedure but a form of learning approach designed to determine good variants of such solution procedures.

TABU SEARCH

Tabu search is a higher-level method, or metastrategy, for solving optimization problems. This technique is designed to be superimposed on any procedure whose operation can be characterized as performing a sequence of moves that lead the procedure from one trial solution (or solution state) to another. Each move is assumed to be selected from a set of currently available alternatives, and is susceptible to being evaluated by one or more functions that measure its relative attractiveness in some local sense. When the solution produced by the move is feasible, for example, the objective function value itself provides such a measure.

The well-known *hill-climbing heuristics* fall within the class of procedures susceptible to being embedded within tabu search. In general, a hill-climbing heuristic progresses from an initial feasible solution along a path that changes the objective function value in a uniformly descending or ascending direction (for minimization or maximization, respectively) until no further improvement of the objective function is possible by means of the available moves. At the stopping point, the solution obtained is a local optimum that, for combinatorial problems, very rarely is also global (i.e. rarely the best solution across the entire range of feasible possibilities). In this context, tabu search provides a guiding framework for exploring the solution space beyond points where an embedded heuristic would become trapped at a local optimum.

The most basic form of tabu search consists of introducing *tabu restrictions* that classify certain moves as forbidden, together with *aspiration criteria* capable of overriding the tabu status of moves (where appropriate). These activities have a time-dependent dimension that can be implemented by means of a short-term memory function. More elaborate tabu search procedures include intermediate and long-term memory functions to carry out additional strategic operations.

The success of the method has been noteworthy. Although, at present, tabu search is not nearly as widely studied as simulated annealing and genetic algorithms—and, correspondingly, more remains to be learned about the best ways to apply the method—there are already a number of problem settings where tabu search has been able to find solutions superior to the best results previously obtained by any method. In other cases, tabu search has been demonstrated to offer advantages in ease of implementation or in the flexibility to handle additional considerations (such as constraints of a form not encompassed by the original problem formulation).

Applications of tabu search where superior performance and/or greater adaptability have been reported cover a considerable spectrum, including employee scheduling (Glover and McMillan, 1986), machine scheduling (Laguna, 1989; Laguna *et al.*, 1989), maximum satisfiability problems (Hansen and Jaumard, 1987), space planning and architectural design (Glover *et al.*, 1985), computer channel balancing (Glover, 1989a), character recognition (Hertz and de Werra, forthcoming), convoy scheduling (Bovet, 1987), telecommunications path assignment (Ryan *et al.*, 1989), quadratic assignment problems (Skorin-Kapov, 1990), nonlinear covering problems (Glover, 1990), traveling salesman problems (Knox, 1989; Malek *et al.*, 1989a, b), flow-shop sequencing problems (Windmer and Hertz, forthcoming), job-shop scheduling problems (Eck, 1989), graph coloring and partitioning problems (Hertz and de Werra, 1987; Wendelin, 1988), maximum stable set problems (Friden *et al.*, 1989), and a variety of others (see, e.g., the surveys in Glover, 1989a, 1990; Hertz and de Werra, forthcoming).

In the domain of combinatorial problem solving there is still a long way to go, however, and the goal remains to do better tomorrow than today. Acknowledging this, the large body of positive results for a procedure that is just beginning to be studied suggests the tabu search framework may offer something of value in a number of the areas currently considered challenging. Viewed from the perspective of research, the new types of memory schemes appropriate to tabu search motivate the development of correspondingly new data structures and processing methods. Tabu search also invites new applications for cutting-plane theory (including the introduction of 'pseudo' cutting planes, which may be purged with the expiration of a short-term memory tenure), and new quests for probability theory in the versions of probabilistic tabu search. Some of the features of the method are elaborated in the following sections.

BASIC ELEMENTS

Tabu search may be viewed as a nested hierarchy of long-, intermediate- and short-term memory functions, with the short-term function constituting the core of the procedure. The short-term memory component of the method operates by selecting moves designed to progress quickly to a local optimum (seeking those with the highest evaluations, subject to trade-offs involved in the effort of identifying such moves), and then to go beyond the local optimum by forbidding moves with certain attributes (making them tabu). No concern is given to the fact that the best moves available may not improve the current solution. Instead, the method selects the moves with highest evaluations, from the set not classified as tabu, to drive the search into new regions. The process generates a trajectory that often includes a large portion of high-quality solutions, while periodically obtaining solutions better than the best found previously during the search. Each such pass of the short-term memory component continues until a specified number of iterations elapses since the best solution was last improved or until an overall cutoff limit is reached.

The intermediate and long-term memory functions of tabu search co-ordinate successive passes of the short-term memory component, or successive intervals of a given pass, to achieve goals that may be described as local intensification and global diversification of the search (Glover, 1989a, 1990). These memory functions operate as boosters to obtain solutions of still higher quality or to permit solutions of a given quality to be obtained more efficiently.

Because of the central role of the short-term memory component, we will focus on its operations in the remainder of this section. To provide a fuller understanding of how this component operates, a diagram of a single pass of this procedure is given in Fig. 1, adapted from Glover (1989b).

The word 'solution' as used in this figure admits a flexible interpretation. It can represent what is commonly called a *trial solution*, or even a *partial solution*; i.e. it may not satisfy all constraints or specify values for all variables. A form of evaluation criterion is used that permits different solutions (of whatever type) to be compared.

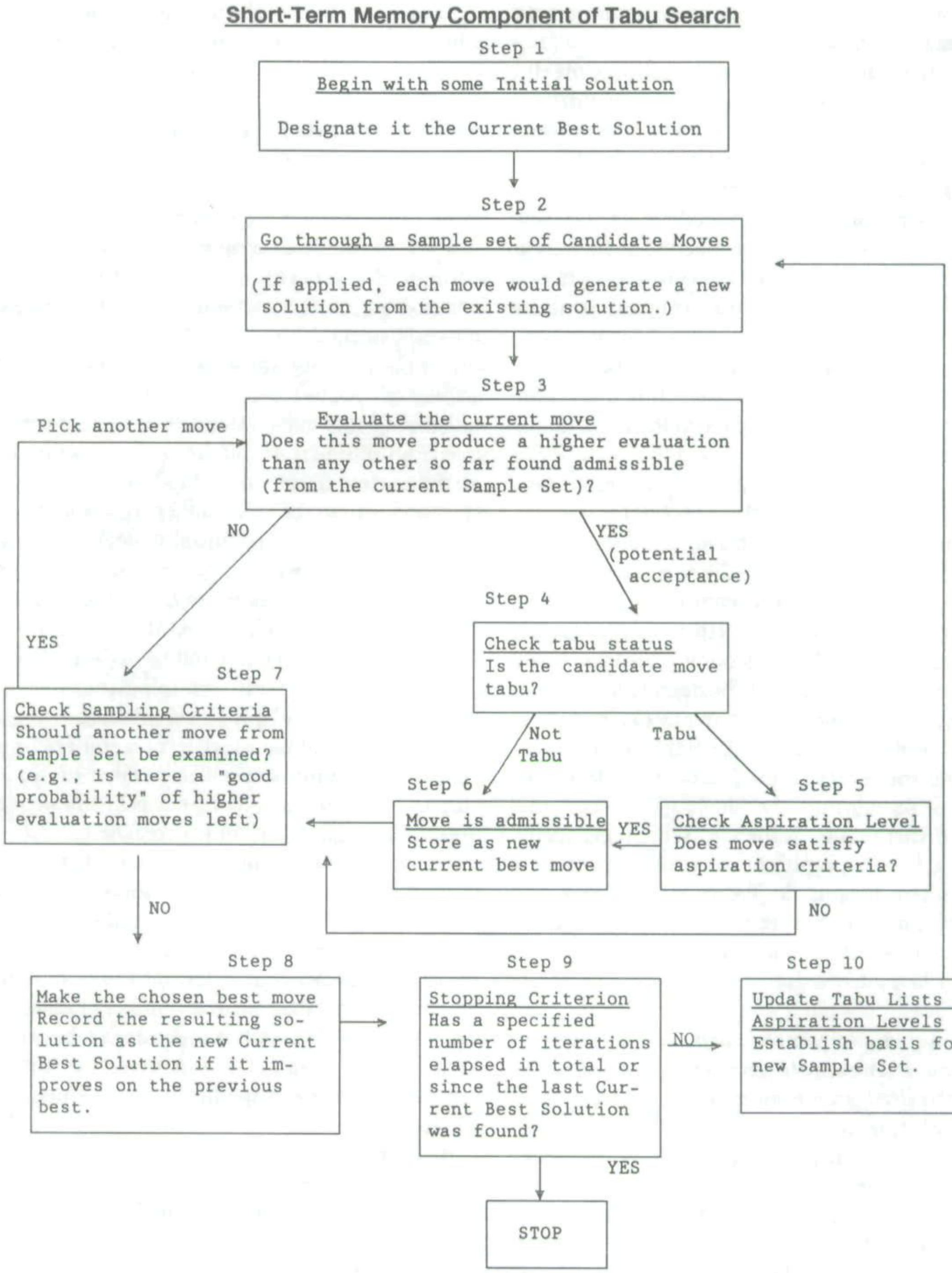
Short-Term Memory Component of Tabu Search
Step 1
Begin with some Initial Solution
Designate it the Current Best Solution
Step 2
Go through a Sample set of Candidate Moves
(If applied, each move would generate a new solution from the existing solution.)
Step 3
Pick another move
Evaluate the current move
Does this move produce a higher evaluation than any other so far found admissible (from the current Sample Set)?
NO
YES (potential acceptance)
Step 4
Check tabu status
Is the candidate move tabu?
YES
Step 7
Check Sampling Criteria
Should another move from Sample Set be examined? (e.g., is there a "good probability" of higher evaluation moves left)
Not Tabu
Tabu
Step 6
Move is admissible
Store as new current best move
YES
Step 5
Check Aspiration Level
Does move satisfy aspiration criteria?
NO
NO
Step 8
Make the chosen best move
Record the resulting solution as the new Current Best Solution if it improves on the previous best.
Step 9
Stopping Criterion
Has a specified number of iterations elapsed in total or since the last Current Best Solution was found?
NO
Step 10
Update Tabu Lists and Aspiration Levels
Establish basis for new Sample Set.
YES
STOP

Figure 1.

To give substance to the diagram, the functions performed by each of its steps will be identified. For concreteness, we describe how the procedure may be applied to the traveling salesman problem. The principles can readily be extrapolated to other contexts.

Step 1. A natural form of a 'solution' in the traveling salesman setting is a tour, i.e. a cycle that visits each node exactly once. An initial tour can be generated in a variety of ways, randomly or otherwise (allowing artificial edges with high costs where necessary) as by a simple construction process that grows a progressively enlarging chain until the last edge completes the tour.

Step 2. A Sample Set of candidate moves for the traveling salesman problem may consist, for example, of the standard '2-OPT' moves, or swaps, which result by dropping any two nonadjacent edges and adding the unique pair of edges that will produce a different tour. Where the set of such moves is large, the Sample Set may be chosen as a subset, as by subdividing the tour in various ways and allowing only those swaps that occur within given components of the subdivision. (An approach of creating sample sets by such subdivisions has proved highly effective in applications of tabu search to machine scheduling problems (Laguna *et al.*, 1989).)

Step 3. The evaluation of the current candidate move can be applied in the case of 2-OPT moves by comparing the lengths of the two added and two deleted edges to see if the resulting tour would be improved. To take the 'YES branch' from this step, leading to Step 4, the move must have a higher evaluation than those in the Sample Set so far found admissible, where 'admissible' is defined as a result of passing tests in subsequent steps. Until a move from the Sample Set is found that thus qualifies as admissible, and therefore provides a basis for comparison, all moves take the YES branch from the evaluation step (to see if they may in fact become admissible).

Step 4. This step embodies a key issue of the procedure, which is to establish a basis for deciding if a move being examined should be classified as tabu. To create this basis, the most straightforward application of tabu search maintains a *tabu list* that records selected attributes of each move made. For example, each element of the tabu list for 2-OPT moves could be a four-component vector whose first two entries identify the edges added by the move and whose last two entries identify the edges deleted.

The tabu list embodies one of the primary short-term memory functions of the procedure, which it executes by recording only the t most recent moves, where t is the parameter that identifies the 'size' of the list. Considerable success has resulted from strategies that keep t at a fixed value. However, recent experimentation (Taillard, 1990) discloses that better performance results by varying t within a chosen interval of values, remaining with a given value for approximately $2t$ consecutive iterations before choosing another.

To implement the procedure, an array denoted tabu-time (e) is created which identifies the (most recent) iteration when a move 'containing' a specified attribute e was made. Attributes generally are defined so that each has a natural complement or 'reverse' attribute. (e.g., the complement of an attribute that corresponds to adding a particular edge is the attribute that corresponds to dropping that edge.) Then the repetition of a move containing an attribute e is avoided, along with other associated moves, by classifying e tabu (hence forbidding its inclusion in future moves) as long as tabu-time (e) lies within t iterations of the current iteration. The reversal of a move is avoided similarly by classifying the complement f of e tabu as long as tabu-time (f) does not exceed the difference 'current iteration $-t$'. Avoiding move reversals is frequently a more effective strategy than avoiding move repetitions, although in some contexts (Glover, 1989a) it is appropriate to prevent both reversals and repetitions, according to the type of move employed.

The attributes of moves that are chosen to be recorded in the elements of the tabu list can be used in a variety of ways to define tabu status. Such differences are important and generate search paths with different characteristics. Generally it seems worthwhile to create a separate tabu list and tabu list 'size' t for each attribute class. For example, added edges and deleted edges can each have their own lists, and the size of the list for added edges (to prevent them from being subsequently deleted) should normally be somewhat smaller than for deleted edges (to prevent them from being subsequently added back), reflecting the fact that the number of edges contained in a traveling salesman tour is generally somewhat smaller than the number not in the tour. Best ranges for t (as in the approach that varies t between selected limits) typically lie in a proper subset of the interval from n/3 to 3n, where n is related to problem dimension (such as the number of nodes or edges of a graph, or a square root of this number). However, in some applications a simple choice of t in a range centered around 7 seems to be quite effective. In any case, good ranges are characteristically easy to identify and highly robust. Dynamic tabu list strategies that employ "moving gaps" are emerging that may prove even more successful (Hubscher and Glover, 1990). The relation between different tabu list structures and the types of restrictions for classifying moves tabu (as a function of selected move attributes) is an important area for research.

One goal of this application of short-term memory is to avoid a solution path that duplicates a sequence of solutions, i.e. to avoid cycling (loosely defined, since sometimes it is preferable to return to a previous solution to find an improved path leading away). Because the manner of cycle avoidance involved in the operation of the tabu list is highly flexible—much more so than branch and bound, for example—it does not give an absolute theoretical assurance that cycling is impossible. Nevertheless, from an empirical standpoint the tabu list performs this function highly effectively for appropriate values of t. (More advanced types of tabu lists, theoretically motivated, are now emerging (Glover, 1990).)

Step 5. Another key issue of tabu search arises when the move under consideration has been found to be tabu. If appropriate aspiration criteria are satisfied, the move will still be considered admissible in spite of its tabu classification. Roughly speaking, these criteria are designed to override tabu status if a move is 'good enough'. The condition 'good enough' must be sufficiently limiting to be compatible with the goal of preventing the solution process from cycling (in an appropriate sense).

Based on this motivation, a simple form of an aspiration-level check is to permit tabu status to be overridden if the solution produced would be better than the Current Best Solution. Another approach is to define an aspiration level $A(L)$ to be the length of the best tour that has ever been reached by a move from a tour of length L. Then, if a move applied to a tour of length L can produce a tour of length L^* better than $A(L)$ (i.e. $L^* < A(L)$), the solution process cannot duplicate a sequence followed before.

A method that has proved useful in some applications is to base the aspiration level on the same attributes that define tabu status, and to require a solution better than the one that gave rise to classifying such attributes tabu (Glover, 1989b; Malek *et al.*, 1989a). Another approach is to regard a tabu move admissible if its evaluation surpasses that of the best non-tabu move by a specified margin (which can depend, for example, on age on the tabu list, frequency of surpassing the best non-tabu move, etc.).

Steps 6 *and* 7. Following the aspiration-level check, either the move under examination is found admissible and is recorded as the new current best move at Step 6 or the process proceeds directly to Step 7 to check the sampling criteria. The latter step is a dynamic means of monitoring the definition of the Sample Set, and hence of choosing the 'best move' from this set at each iteration. Thus, for example, one form of dynamic sampling strategy is to stop as soon as a move of a specified level of attractiveness is found. If, at the extreme, this level is set to admit any move that creates an improvement in the current tour, then the result is the same as the classical heuristic hill-climbing procedure, until a situation is reached where no improving move exists. Tabu search, however, favors a more aggressive orientation, seeking the highest-quality move that can be expected for a given expenditure of search effort, subject to the restrictions imposed by the tabu conditions and moderated by the aspiration level.

The operation of checking the sampling criteria at Step 7 has an additional function. Except in those variations where tabu status is enforced by penalizing rather than forbidding certain moves, it is possible that no move in the Sample Set is admissible. Once such a condition is discovered, the operation of selecting another move is performed by temporarily relaxing the tabu restrictions to allow at least one of the tabu moves to be classified admissible. Alternatively, if the Sample Set has been restricted to a subset of moves potentially available, a new Sample Set may be created.

The issue of generating effective subsets of moves for consideration may be classified under the heading of *candidate list strategies*, and offers an extremely fertile area for investigation. As observed in Glover (1989b), candidate list strategies encompass a variety of well-known procedures, including simulated annealing, as simple special cases.

Steps 8–10. These steps are self-explanatory or involve considerations already discussed.

The preceding framework for the short-term memory component of tabu search can be applied to a wide range of settings. Although it leaves out some considerations that can be important in special contexts (see, e.g., Glover, 1989a, b), its principles are directly relevant to many combinatorial problems.

The variant of the procedure called *probabilistic tabu search* replaces the deterministic choice rules previously described, by assigning probabilities governing the acceptance of available moves which reflect their evaluations and tabu status, while allowing aspiration criteria to override these probabilities for moves that are sufficiently attractive.

Such probabilities characteristically are biased to encourage the selection of highest evaluation moves, again relying on candidate list strategies to help isolate good alternatives when the effort of examining the full set is laborious. Conditions for converging to an optimal solution with a probability of 1 have been established for this variant by Faigle and Kern, 1989, and are shown to be more general than the corresponding conditions for simulated annealing.

AN ILLUSTRATIVE EXAMPLE

To clarify the operation of the short-term memory component of tabu search diagrammed in the previous section, and to identify some of the options relevant to its application, this section traces the solution steps for applying the to a zero–one IP problem. The outcomes for each iteration are shown in Table 1.

A brief description of the problem objective and the types of moves selected, together with tabu restrictions and aspiration criteria, appears at the top of the table. Below this, in the main body of the table, the objective function value is given for the current solution at each iteration, followed by the moves with the three best evaluations. These moves are listed to disclose how the method operates when one or more of the highest evaluation moves are tabu. (Normally, the method would not undertake to sort the top evaluations, but would simply identify the best admissible move according to the tabu restrictions and aspiration criteria.)

In this example the move evaluations correspond to the x_0 values produced by the moves. Since the problem has a minimization objective, these x_0 values progress from smaller to larger as the moves progress from Best to Second-best to Third-best.

The current solution vector and the tabu list at each iteration are shown at the bottom of the table. Initially, the tabu list (which in this example con-

Table 1. Tabu Search Illustrated

Zero–One Integer Programming Problem

Objective: Minimize x_0 = function of integer variables $(x_1, x_2, \ldots, x_5)$
Move Type: Increase or decrease value of a selected variable by 1.
(Set the value to 0 or 1)
Tabu Restriction: Prevent a variable from decreasing if it was previously increased, or from increasing if it was previously decreased
Aspiration Criterion: Override the tabu status of a move if x_0 can be improved beyond the value it had when the move was made tabu
Choice: Pick the best move that is not tabu (or whose aspiration criterion is satisfied)

			Iteration		
	1	2	3	4	5
Current x_0	60	30	80	40	65
Best	$x_0=30$(*)	$x_0=60$	$x_0=30$	$x_0=60$	$x_0=20$ (*)
move	$x_3=1$	$x_3=0$ (T)	$x_1=1$ (T)	$x_1=1$ (T)	$x_4=1$
Second-best	$x_0=40$	$x_0=80$ (*)	$x_0=40$ (*)	$x_0=65$ (*)	$x_0=40$
move	$x_1=0$	$x_1=0$	$x_3=0$ (TA)	$x_2=1$	$x_2=0$ (T)
Third-best	$x_0=85$	$x_0=85$	$x_0=110$	$x_0=80$	$x_0=85$
move	$x_2=1$	$x_4=1$	$x_5=0$	$x_3=1$ (T)	$x_1=1$ (T)
.	.	.	.	.	.
.	.	.	.	.	.
Current solution	(1, 0, 0, 0, 1)	(1, 0, 1, 0, 1)	(0, 0, 1, 0, 1)	(0, 0, 0, 0, 1)	(0, 1, 0, 0, 1)
Tabu state and	·	$x_3=0$ (60)	$x_3=0$ (60)	$x_3=0$ (60)	$x_2=0$ (40)
aspiration	·	·	$x_1=1$ (30)	$x_1=1$ (30)	$x_1=1$ (30)
value	·	·	·	$x_3=1$ (80)	$x_3=1$ (80)

(*): Move selected.
(T): Tabu.
(TA): Tabu but passes aspiration test.

tains only three elements) contains no entries, as indicated by the black dots in the cells. Thereafter, the entries in the cells indicate both the tabu moves and, in parentheses, the aspiration values for x_0 which allow tabu status to be overridden. The entries of the table may be explained as follows.

Iteration 1. Starting from an initial solution with $x_0=60$, identified by the vector (1, 0, 0, 0, 1), the method examines the moves that change exactly one x_j value in this vector (from 0 to 1 or from 1 to 0). The moves yielding the three best values for x_0 are shown in the table as $x_3=1$, $x_1=0$ and $x_2=1$, which, respectively, give $x_0=30$, $x_0=40$ and $x_0=85$. (The first two moves improve the current x_0 value, while the third does not.) No moves are tabu, so the best move $x_3=1$ is selected, as indicated by the (*) symbol. The reverse move $x_3=0$ becomes tabu and is entered on the tabu list. The entry appears in the tabu list for Iteration 2 rather than for Iteration 1, however, since this list shows the tabu status for moves at the start of each iteration. The aspiration value of 60 is recorded with the tabu move, because $x_0=60$ occurs for the solution of Iteration 1, and this is the value to beat if the move $x_3=0$ is to be considered admissible.

Iteration 2. The solution obtained by setting $x_3=1$ in Iteration 1 appears in the column for Iteration 2, identified by (1, 0, 1, 0, 1) with $x_0=30$. The best move results by setting $x_3=0$ to yield $x_0=60$. (This discloses that a local optimum has been reached, because the best move does not lead to a solution better than the present solution with $x_0=30$). The tabu list shows, however, that the best move is tabu, and the associated aspiration value of 60 is not surpassed by the value $x_0=60$ of this move. (The move under consideration leads directly back to the solution of Iteration 1). Consequently, the symbol (T) appears in the cell to indicate the move's tabu status. In computer implementation, tabu status arrays such as the tabu-time (e) array permit such information to be determined directly without having to search a list of the form shown here.

The next best move, which yields $x_0=80$ for $x_1=0$, is not tabu and hence is selected. The reverse move, $x_1=1$, is entered in the next available position on the tabu list (appearing in Iteration 3 to disclose the list condition at the start of that iteration), together with the aspiration value of 30, corresponding to the fact that $x_0=30$ in the solution of Iteration 2.

Iteration 3. The best available move once again is tabu, and hence is not taken. In this case the second-best move also appears on the tabu list. This move, $x_3=0$, has an aspiration value of 60, while yielding $x_0=40$, and hence passes the aspiration test. The symbol (TA) identifies the move's admissibility in spite of being tabu, and this move is selected.

The tabu list is updated as before, yielding the new tabu list which appears on the column for Iteration 4. This list contains two entries for x_3, one for $x_3=0$ and one for $x_3=1$. This does not currently create a problem, since the entry for $x_3=0$ is irrelevant to the current solution (in which x_3 already is 0). In general, however, it would be preferable to erase such an irrelevant entry, i.e. replacing it by a black dot in the table, to avoid the possibility of future ambiguities.

A superior method exists for updating aspiration criteria of the type used in this example when the tabu status of a move is overridden. An approximation to this approach is to subtract an amount D from the aspiration values for all tabu moves recorded since the tabu move s that was overridden, where D is equal to the aspiration value for the tabu move recorded immediately after s, minus the aspiration value for s. If D is negative, then this step should only be applied to the aspiration value for the first tabu move after s.

Iteration 4. The solution reached at this iteration is another local optimum, evidenced by the fact that the best available move results in $x_0=60$, as contrasted with the current $x_0=40$. The second-best move, which is the highest evaluation move that is not tabu, is selected. Since the most recent tabu list entry occurred in the last position, the entry for the current move is made in the first position (to record the current tabu restriction $x_2=0$). This removes the tabu restriction previously recorded in the first position, rendering it inapplicable.

Iteration 5. The highest evaluation move, $x_4=1$, is admissible to be selected and hence becomes the current choice. (The second-highest evaluation move is also admissible as a result of passing the aspiration test.) The solution associated with this move yields $x_0=20$, which is better than any thus far produced, qualifying the solution as the new current best.

Additional columns of the table may be generated in a corresponding manner, given a subroutine capable of identifying evaluations (i.e. in this ex-

Table 2. Example problem (Solved by Tabu Search in Table 1)

Minimize $20x_1+25x_2-30x_3-45x_4+40x_5$

Subject to

$$\begin{aligned} x_1+x_2-x_3 \quad &+x_4+x_5\geqslant 1\\ x_1+x_2 \quad &-x_4+2x_5\geqslant 2\\ x_2+x_3 \quad &+x_5\leqslant 2\\ -x_2 \quad &+x_4+x_5\leqslant 1 \end{aligned}$$

All variables 0 − 1

Penalty for each unit of constraint violation:

70/unit for each of the first two constraints

100/unit for each of the last two constraints

ample, the x_0 values) that result for available moves.

Table 2 identifies the problem that was solved by the process illustrated in Table 1. (The solution with $x_0=20$ is in fact optimal for this problem.) This table was not provided earlier because of the difficulty of tracing relevant correspondences between the solution process and the problem data, and the effort of verifying the associated arithmetic calculations. Table 2 is instructive at this point as a means of disclosing how tabu search can be applied to a problem of this type in more than one way.

The initial solution of Table 1 is feasible for the problem of Table 2. The solution process illustrated in Table 1 is not based on requiring all subsequent solutions to be feasible, but instead allows consideration of infeasible moves, which are evaluated by imposing penalties on violating the constraints. These penalties, identified in Table 2, produce a solution trajectory in which one of the constraints becomes violated during the solution process. (Other penalties, leading to different trajectories, also could have been used.)

Combinatorial optimization problems are not always conveniently structured to assure a feasible path will exist between all feasible solutions, and hence some method of allowing infeasible solutions to be evaluated and visited is important. Besides employing evaluators that penalize infeasibilities in various ways, tabu search also provides an approach called strategic oscillation, which introduces additional tabu restrictions to compel the search to cross feasibility boundaries to various depths (Glover, 1986, 1989a). The strategic oscillation approach has been effective in applications that have included the solution of *p*-median (lock box) problems and large-scale employee scheduling problems (Glover, 1989a; Glover and McMillan, 1986).

From the perspective of the example problem this approach leads to consideration of a particular variation that invites further exploration. The strategic oscillation procedure often incorporates a shifting evaluation criterion that varies the emphasis on feasibility and optimality considerations (at different depths and on different sides of feasibility boundaries). A natural way to carry out this type of process is to use adaptive penalty values for the different problem constraints.

For any given set of such penalties the problem is effectively transformed into an integer goal programming problem. Consequently, the use of successive (implicit) transformations of this sort gives rise to what may be called a *tabu goal programming procedure*, where the word 'tabu' conveys not only the use of tabu lists to avoid cycling but the adaptive manipulation of penalties according to the objectives of the strategic oscillation element of tabu search. (Effective use of such penalties should reduce the size of the tabu list that otherwise might be employed.) This type of procedure offers interesting possibilities to be investigated in integer programming applications.

TARGET ANALYSIS

A careful consideration of heuristic solution procedures sooner or later encounters the challenge of determining more advanced evaluation measures than embodied in objective function values produced by available moves, as the preceding discussion underscores. In contexts where relaxation strategies are used, as in obtaining LP solutions as part of a method for solving integer programming problems, a true objective function value for a move (which results when the relaxed formulation is replaced by the original) may not be known or readily determined. A similar situation occurs in certain scheduling problems, where the implications of making a move do not become visible until the schedule approaches a completed state.

Target analysis (Glover, 1986; Glover and Greenberg, 1989; Glover and Laguna, 1989; Glover *et al.*, 1989) is a method that can be used to determine more effective decision rules in such situations. The principles of target analysis harmonize well with those of tabu search, and the method also can be applied in conjunction with many other procedures.

Its main features may briefly be sketched by viewing the approach as a five-phase procedure, as follows.

Phase 1 of target analysis is devoted to applying existing methods to determine optimal or near-optimal solutions to representative problems from a given class. This phase is straightforward in its execution, although a high level of effort may be expended to assure the solutions are of the specified quality.

Phase 2 uses the solutions produced by Phase 1 as targets, which become the focus of a new set of solution passes. During these passes, each problem is solved again, this time scoring all available moves (or a high-ranking subset) on the basis of their ability to progress effectively toward the target solution. (The scoring can be a simple classification, such as 'good' or 'bad'.) Choices may be biased during this phase to select moves that have high scores, thereby leading to the target solutions more quickly than the customary choice rules. Information generated during the solution effort, which may be useful in inferring these scores, is stored for later analysis.

Phase 3 constructs parameterized functions of the information recorded in Phase 2, with the goal of finding values of the parameters to create a *master decision rule*. This rule is designed to choose moves that score highly according to the outcomes of the second phase.

Phase 4 generates a mathematical or statistical model (such as a generalized goal programming or discriminate analysis model) to determine effective parameter values for the master decision rule. (The second, third and fourth phases are not entirely distinct, and may be iterative.) On the basis of the outcomes of the Phase 4, the master decision rule becomes the rule that drives the solution method. This rule itself may be evolutionary, i.e. it may use feedback of outcomes obtained during the solution process to modify its parameters for the problem being solved.

Phase 5 concludes the process by applying the master decision rule to the original representative problems and to other problems from the chosen solution class to confirm its merit. (The process can be repeated and nested to achieve further refinement.)

Target analysis has an additional important function. On the basis of the information generated during its application, and particularly during its final confirmation phase, the method produces empirical frequency measures for the probabilities that choices with high evaluations will lead to an optimal (or near-optimal) solution within a certain number of steps. By this means, target analysis can provide inferences concerning expected solution behavior, as a supplement to classical 'worst case' complexity analysis. These inferences can aid the practitioner by indicating how long to run a solution method to achieve a solution of desired quality (and with a specified empirical probability).

In combination with an effective heuristic framework such as tabu search, target analysis appears to offer a promising foundation for further advances in solving difficult optimization problems.

Acknowledgement

This research has been supported in part by the NASA Grant NAGQ-1388 of the Center for Space Construction and by the Air Force Office Scientific Research and the Office of Naval Research Contract # F49620-90-C-0033.

REFERENCES

J. Bovet, C. Constantin and D. de Werra (1987). A convoy scheduling problem. *Research Report ORWP* 87/22, Swiss Federal Institute of Technology in Lausanne, December.

B. T. Eck (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.

U. Faigle and W. Kern (1989). Some convergence results for probabilistic tabu search. Research Memorandum No. 882, Faculty of Applied Mathematics, University of Twente, The Netherlands, November.

C. Friden, A. Hertz and D. de Werra (1989). Stabulus: a technique for finding stable sets in large graphs with tabu search. *Computing* **42,** 35–44.

F. Glover (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* **13,** No. 5, 533–49.

F. Glover (1989a). Tabu search, part I. *ORSA Journal on Computing* **1,** No. 3, 190–206.

F. Glover (1989b). Candidate list strategies and tabu search. *CAAI Research Report*, University of Colorado, Boulder, July.

F. Glover (1990). Tabu search, part II. *ORSA Journal on Computing* **2,** No. 1, 4–32.

F. Glover and H. J. Greenberg (1989). New approaches for heuristic search: a bilateral linkage with artificial intelligence. *European Journal of Operational Research* **39,** No. 2, 119–30.

F. Glover and M. Laguna (1989). Target analysis to improve a tabu search method for machine scheduling. *Technical Report*, Advanced Knowledge Research Group, US West Advanced Technologies, Boulder, Colorado.

F. Glover and C. McMillan (1986). The general employee scheduling problem: an integration of management science and artificial intelligence. *Computers and Operations Research* **13,** No. 5, 563–93.

F. Glover, C. McMillan and B. Novick (1985). Interactive decision software and computer graphics for architectural and space planning. *Annals of Operations Research* **5,** C. Monma (ed) 557–73.

R. Glover, D. Klingman and N. V. Phillips (1989). A network-related nuclear plant model with an intelligence branch-and-bound solution approach. *Annals of OR: Linkages with Artificial Intelligence.*

P. Hansen and B. Jaumard (1987). Algorithms for the maximum satisfiability problem, RUTCOR Research Report RR #43-87, Rutgers, NJ, June.

A. Hertz and D. de Werra (1987). Using tabu search techniques for graph coloring. *Computing* **29,** 345–51.

A. Hertz and D. de Werra (forthcoming). The tabu search metaheuristic: how we used it. *Annals of Mathematics and Artificial Intelligence.*

R. Hubscher and F. Glover (1990). Tabu search for bin packing. CAAI Research Report 90-4, University of Colorado, Boulder, September.

J. Knox (1989). *The Application of Tabu Search to the Symmetric Traveling Salesman Problem*, PhD thesis, Graduate School of Business, University of Colorado, July.

M. Laguna (1989). A tabu search method for scheduling jobs on parallel processors. *Research Report*, Department of Mechanical Engineering, University of Texas-Austin, May.

M. Laguna, J. W. Barnes and F. Glover (1989). Scheduling jobs with linear delay penalties and sequence dependent setup costs using tabu search. *Research Report*, Department of Mechanical Engineering, The University of Texas-Austin, April.

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

M. Malek, M. Heap, R. Kapur and A. Mourad (1989b). A fault tolerant implementation of the traveling salesman problem. *Research Report*, Department of Electrical and Computer Engineering, University of Texas-Austin, May.

J. Ryan, S. Oliveira and G. Stroud (1989). A parallel version of tabu search and the path assignment problem. *Heuristics for Combinatorial Optimization* 1–24.

J. Skorin-Kapov (1990). Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing* **2:1,** 33–45.

E. Taillard (1990). Robust taboo search for the quadratic assignment problem. Research Report ORWP 90/10, EPFL, DMA, Lausanne, Switzerland, June.

Ch. Wendelin (1988). Graph partitioning with the aid of the tabu method. *Project Report*, Institute fur Hohere Studien, Vienna, Austria, June.

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

D. Woodruff and E. Zemel (1990). Hashing vectors for tabu search. Technical Report 90-08, Department of I.E. and M.S., Northwestern University, July.