ORSA Journal on Computing
Vol. 7, No. 4, Fall 1995

0899-1499/95/0704-0426 $01.25

Tabu Thresholding: Improved Search by Nonmonotonic Trajectories

FRED GLOVER / *US West Chair in Systems Science, Graduate School of Business, Box 419, University of Colorado, Boulder, Colorado 80309-0419; Email: fred.glover@colorado.edu*

(Received: October 1992; revised: November 1993; accepted: June 1994)

There is an appeal to methods like simulated annealing and threshold acceptance that operate by imposing a monotonically declining ceiling on objective function levels or degrees of disimprovement (treated probabilistically or deterministically). An alternative framework, embodied in tabu search, instead advocates a *nonmonotonic* form of control, keyed not only to the objective function but to other elements such as values of variables, direction of search, and levels of feasibility and infeasibility. This creates a more flexible search behavior and joins naturally with the use of memory-based strategies that are the hallmark of tabu search approaches. Embodied particularly in the strategic oscillation component of tabu search, this nonmonotonic control has been shown in a variety of studies to yield outcomes superior to those of simulated annealing and threshold acceptance. The question arises whether such an approach offers a sufficiently rich source of search trajectories to be relied upon as a primary guidance mechanism, with greatly reduced reliance on forms of memory customarily used in tabu search. To provide an easily implemented method of this type we propose a *tabu thresholding* approach, which joins prescriptions of strategic oscillation with a candidate list procedure derived from network optimization studies. The candidate list and tabu search philosophies are mutually reinforcing, and the computational advantages contributed by these elements, documented by studies cited in this paper, motivate a closer look at combining them. The result yields a method with a significant potential for variation and an ability to take advantage of special structure.

The basic ideas for solving optimization problems explored in this paper originate from two sources. The first source consists of simple but fundamental ideas from tabu search. The second consists of candidate list strategies proposed for network optimization procedures. These two sources contain strongly reinforcing elements that combine to produce a class of procedures we call *tabu thresholding* methods.

As a basis for our development, we express the mathematical optimization problem in the form

$$\text{(P) Minimize } c(x)$$
$$x \in X \subseteq R^n.$$

The function c(x) may be linear or nonlinear, and the condition $x \in X$ summarizes constraints on the vector x such as embodied in linear or nonlinear inequalities, including discrete restrictions (as where some components of x are required to take integer values).

The tabu thresholding methods we propose for (P) share a number of features in common with more complex forms of tabu search, but rely to a significantly reduced degree on the incorporation of memory structures. They specifically embody the principle of aggressively exploring the search space in a nonmonotonic way. The motivation for using a nonmonotonic search strategy, in contrast to relying on a undirectionally modified "temperature" parameter as in simulated annealing, derives from the following observation (Glover[16]).

> ... the human fashion of converging upon a target is to proceed not so much by continuity as by thresholds. Upon reaching a destination that provides a potential "home base" (local optimum), a human maintains a certain threshold—not a progressively vanishing probability—for wandering in the vicinity of that base. Consequently, a higher chance is maintained of intersecting a path that leads in a new improving direction.
>
> Moreover, if time passes and no improvement is encountered, the human threshold for wandering is likely to be increased, the reverse of what happens to the probability of accepting a nonimproving move in simulated annealing over time. On the chance that humans may be better equipped for dealing with combinatorial complexity than particles wandering about in a material, it may be worth investigating whether an "adaptive threshold" strategy would prove a useful alternative to the strategy of simulated annealing.

This observation is extended by noting the relevance of one of the basic components of tabu search, the strategic oscillation approach (Glover[15]), which seeks improved outcomes by

> ... creating a search pattern that resembles a series of pendulum swings... inducing oscillations around "ideal" values of various parameters (including the objective function value and measures of infeasibility)...

Subject classifications: Tabu search.

During the past several years, a number of implementations of the strategic oscillation component of tabu search have demonstrated the effectiveness of this nonmonotonic type of guidance. A summary of such applications appears in Table I. Each of these applications includes comparisons that establish performance results superior to those of earlier studies for the problems examined. Such comparisons are particularly interesting in the case where the objective function is used as the element of control (Osman[43]; Osman and Christofides[44]; Verdejo, Cunquero and Sarli[55]). In these applications the outcomes are shown to dominate those of simulated annealing, which bases its control on the same parameter. (Several of the studies using different parameters for strategic oscillation—in particular, those for quadratic assignment, vehicle routing, data integrity, mixed fleet VRP, graph partitioning, time deadline VRP, the P-median Problem and Traveling Purchaser Problem—also include comparisons to results of simulated annealing, or to other studies reporting such results, and likewise yield solutions of higher quality.) In addition, a new proposal of a non-monotonic thresholding method keyed to objective function levels, but using a deterministic guidance rule, has recently been reported by Hu, Kahng, and Tsau[28] to produce outcomes superior to those of the monotonic "threshold accepting" approach of Dueck and Scheurer.[6]

These results motivate a closer look at forms of strategic oscillation that are primarily based on controlling search by reference to the objective function. Acknowledging that other control elements may be more effective in some contexts (as suggested by the applications cited in Table I), the creation of guidance rules based on the objective function is particularly straightforward and therefore appealing.

Nonmonotonic guidance effects can be accomplished either deterministically or probabilistically, and in the following development we specifically invoke elements of probabilistic tabu search, which use controlled randomization to fulfill certain functions otherwise provided by memory. By this orientation, probabilities are designed to reflect evaluations of attractiveness, dominantly weighted over *near best* intervals, and additional control is exerted in the choice of the subsets of moves from which these intervals are drawn. These linked processes create an implicit tabu threshold effect that emulates the interplay of tabu restrictions and aspiration criteria in TS procedures that are designed instead to rely more fully on memory.

1. Preliminaries—Strategic Oscillation

To set the stage for the main considerations that follow, it is useful to sketch the ideas of strategic oscillation more fully. Strategic oscillation may be viewed as composed of two interacting sets of decisions, one at a macro level and one at a micro level. These decisions are depicted in Table II.

We discuss these two types of decisions in sequence.

1.1. Macro level decisions

The oscillation guidance function of the first macro level decision corresponds to the element controlled, as listed in the middle column for the applications cited in Table I. (More precisely, the function provides a measure of this element that permits control to be established.) In the second decision at the macro level, a variety of target levels are possible, and their form is often determined by the problem setting. For example, in an alternating process of adding or deleting edges in a graph, the target can be the stage at which the current set of edges creates a spanning tree. Similarly in a process of assigning (and "unassigning") jobs to machines, the target can be the stage at which a

Table I. Guidance by Strategic Oscillation

Application	Element Controlled	Reference
Quadratic Assignment	Tabu Restrictions	Battiti and Tecchiolli[2]
Transportation	Infeasibility Penalties	Cao and Uebe[3]
Operation Timetables	Penalty Measures	Costa[4]
Multidimensional Knapsack	Infeasibility Depth	Freville and Plateau[11]
Vehicle Routing	Nodes and Infeasibility	Gendreau, Hertz and Laporte[12]
Employee Scheduling	Number of Employees	Glover and McMillan[21]
Network Design	Platform (node) assignment	Glover, Lee and Ryan[20]
Data Integrity	Verification Infeasibility	Kelly, Golden and Assad[32]
Multilevel GAP	Assignments/Infeasibility	Laguna et al.[32]
Classroom Scheduling	Assignments/Infeasibility	Mooney and Rardin[38]
Vehicle Routing	Objective Function	Osman[43]
Capacitated Clustering	Objective Function	Osman and Christofides[44]
Mixed Fleet VRP	Vehicles and Infeasibility	Osman and Salhi[45]
Delivery Systems	Penalty Measures	Rochat and Semet[47]
Graph Partitioning	Partitioned Nodes	Rolland, Pirkul and Glover[48]
Time Deadline VRP	Infeasibility	Thangiah et al.[54]
Graph Design	Objective Function	Verdejo, Cunquero and Sarli[55]
P-Median Problem	Values of P	Voss[56]
Traveling Purchaser Problem	Markets	Voss[57]

Table II. Strategic Oscillation Decisions

Macro level decisions
1. Select an oscillation guidance function.
2. Choose a target level for the function.
3. Choose a pattern of oscillation.

Micro level decisions
1. Choose a target rate of change
   (for moving toward or away from the target level).
2. Choose a target band of change.
3. Identify aspiration criteria to override
   target restrictions.

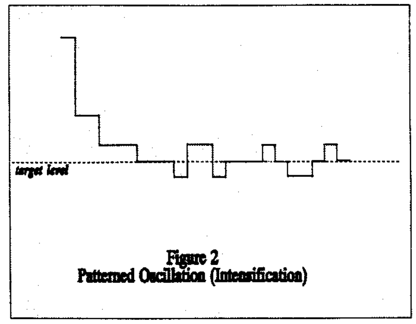

Figure 2
Patterned Oscillation (Intensification)

Figure 2. Patterned oscillation (intensification).

complete assignment is achieved. Each of these can be interpreted as special cases of the situation where the target constitutes a boundary of feasibility, approached from inside or outside. In some instances, the target may appropriately be adaptive, as by representing the average number of employees on duty in a set of best workforce schedules (Glover and McMillan[21]). Such a target gives a baseline for inducing variations in the search, and does not necessarily represent an ideal to be achieved.

The pattern of oscillation at a macro level deals with features such as the depth by which the search goes beyond the target in a given direction, and more generally the way in which the depth varies over time. An example of a simple type of oscillation is shown in Figure 1, where the search begins by approaching the target "from above" and then oscillates with a constant amplitude thereafter.

The pattern in Figure 1 is shown as a broken line rather than a smooth curve, to indicate that the search may not flow continuously from one level to the next, but may remain for a period at a given level. The diagram is suggestive rather than precise, since in reality there are no vertical lines, i.e., the guidance function does not change its value in zero time. Also, the dashed line for the target level should be interpreted as spanning an interval (as conveyed by using the term *target level* rather than *target value*). Similarly, each of the segments of the oscillation curve may be conceived as having a "thickness" or "breadth" that spans a region within which particular values lie.

Intensification: A type of oscillation pattern often employed in short term tabu search strategies adopts an aggressive approach to the target, in some cases slowing the rate of approach and spending additional time at levels that lie in the near vicinity of the target. When accompanied by a policy of closely hugging the target level once it is attained, the pattern is an instance of an *intensification strategy*. Such a pattern is shown in Figure 2.

In both Figures 1 and 2 the oscillations that occur above and below the target may be replaced by oscillations on a single side of the target. Over a longer duration, for example, the pattern may predominantly focus on one side and then the other, or may alternate periods of such a one-sided focus with periods of a more balanced focus.

Patterns that represent intensification strategies generally benefit by introducing some variation over the near to intermediate term. Figure 2 contains a modest degree of variation in its pattern, and other simple types of variation, still predominantly hugging the target level (and hence qualifying as intensification strategies), are shown in Figures 3 and 4. Evidently, intensification patterns with greater degrees of variability can easily be created (as by using pseudo randomization to sequence their components).

In general, intensification in tabu search refers to focusing the search more strongly on attractive regions, and typically takes one of two forms, based on recording and grouping sets of good solutions: (1) moving directly from a current solution to a good solution from the recorded set

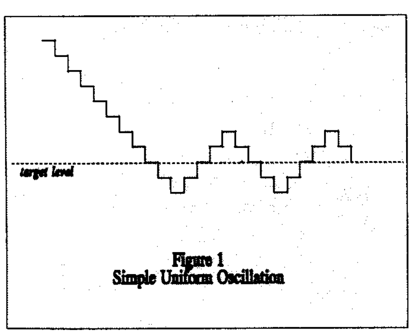

Figure 1
Simple Uniform Oscillation

Figure 1. Simple uniform oscillation.

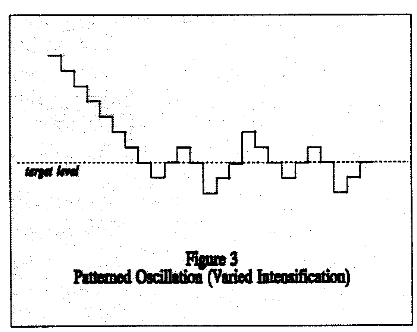

Figure 3
Patterned Oscillation (Varied Intensification)

Figure 3. Patterned oscillation (varied intensification).

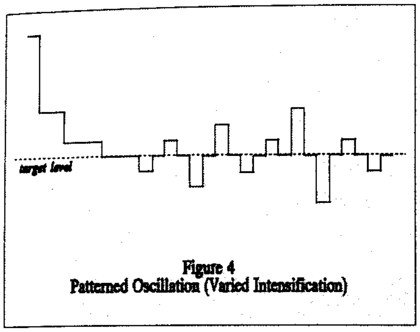

Figure 4. Patterned oscillation (varied intensification).

(which may include unvisited neighbors of other solutions previously visited); (2) modifying choice criteria to favor the inclusion of attributes of good solutions from a selected group. Variability is sometimes introduced by creating a group for this second approach whose elements are drawn from different solution clusters. The type of intensification illustrated in the preceding diagrams is a simple form that does not make recourse to memory, except to keep track of the current phase of a pattern being executed. Nevertheless, it embodies the idea of focusing on a region judged to be strategically important.

Intensification processes that alter choice rules to encourage the incorporation of particular attributes—or at the extreme, that lock such attributes into the solution for a period—can be viewed as designs for exploiting *strongly determined* and *consistent* variables. A strongly determined variable is one that cannot change its value in a given high quality solution without seriously degrading quality or feasibility, while a consistent variable is one that frequently takes on a specific value (or a highly restricted range of values) in good solutions. The development of useful measures of "strength" and "consistency" is critical to exploiting these notions, particularly by accounting for tradeoffs determined by context. However, straightforward uses of frequency-based memory for keeping track of consistency, sometimes weighted by elements of quality and influence, have produced methods with very good performance outcomes (e.g., Farvolden, Crainic and Gendreau[8]; Laguna et al.[36]; Woodruff and Rocke[60]; Porto and Ribeiro[46]).

These kinds of approaches are also beginning to find favor in other settings. For example, strategies introduced in genetic algorithms for sequencing problems, which use special forms of "crossover" to assure offspring will receive attributes shared by good parents, constitute a type of intensification based on consistency (Muhlenbein[40]; Whitley, Starkweather and Fuquay.[58]) Extensions of such procedures based on identifying elements that qualify as consistent and strongly determined according to broader criteria, and making direct use of memory functions to establish this identification, provide an interesting area for investigation.

Diversification: Longer term processes, following the type of progression customarily found beneficial in tabu search, incorporate diversification strategies into the oscillation pattern. In the present setting this translates into periodically creating more pronounced departures from the target level (as when gains from the search begin to diminish).

Extreme forms of diversification are illustrated by restart methods. Diversification sometimes is confused with randomization, but these processes embody somewhat different concepts and have demonstrably different consequences. The popular use of random restarting, for example, generally proves inferior to restarting that incorporates more systematic principles of diversification. In addition, diversification procedures that drive the search into new regions on a path that leads directly from the current region often provide advantages over restarting (see, e.g., Barnes and Laguna[1]; Battiti and Techiolli[2]; Kelly et al.[33]; Hubscher and Glover[29]; Woodruff[59]).

When oscillation is based on constructive and destructive processes, the repeated application of constructive phases (rather than moving to intermediate levels using destructive moves) similarly embodies an extreme type of oscillation that is analogous to a restart method. In this instance the restart point is always the same (i.e., a null state) instead of consisting of different initial solutions, and hence it is important to use choice rule variations to assure appropriate diversification, as discussed later. (The "greedy randomized" procedure of GRASP illustrates an approach based on such repeated constructions; see, e.g., Feo, Venkatraman and Bard[9]).

A connection can also be observed between an extreme version of strategic oscillation—in this case a relaxed version—and the class of procedures known as *perturbation* approaches (Glover[16]). An example is the "large-step simulated annealing" method (Martin, Otto and Felton[37]), that tries to drive an SA procedure out of local optimality by propelling the solution a greater distance than usual from its current location. Perturbation methods may be viewed as loosely structured procedures for inducing oscillation, without reference to intensification and diversification and their associated implementation strategies. Similarly, perturbation methods are not designed to exploit tradeoffs created by parametric variations in elements such as different types of infeasibility, measures of displacement from different sides of boundaries, etc. Nevertheless, at a first level of approximation, perturbation methods seek goals similar to those pursued by strategic oscillation.

The trajectory followed by strategic oscillation is imperfectly depicted in the figures of the preceding diagrams, insofar as the search path will not generally conform to precisely staged levels of a functional, but more usually will lie in regions with partial overlaps. (This feature is taken into account in the micro level decisions, examined in the next subsection.) Further, in customary approaches where diversification phases are linked with phases of intensification, the illustrated patterns of hovering about the target level are sometimes accompanied by hovering as well at other levels, in order to exploit a notion called the Proximate Optimality Principle (POP). According to this notion, good solutions at one level can often be found *close* to good solutions at an adjacent level. (e.g., only a modest number of steps will be required to reach good solutions at one level from those at another.) This condition of course depends on defining levels—and ways for moving within them—appropriately for given problem structures.

The challenge is to identify oscillation parameters and levels that will cause this potential relationship to become manifest. In strategies for applying the POP notion, the transition from one level to another normally is launched from a chosen high quality solution, rather than from the last solution generated before the transition is made (representing another feature difficult to capture in the preceding diagrams).

Path Relinking: The POP notion motivates the tabu search strategy called *path relinking*, which provides an important means for enhancing the outcomes of strategic oscillation. It also gives a further means for exploring the solution space more effectively than by recourse to approaches such as random restarting. In brief, path relinking keeps track of elite solutions generated during the search, and selects subsets of these solutions to serve as *reference solutions*. The procedure then generates paths in neighborhood space, starting from selected members of the reference solutions, and following a trajectory guided by the other reference solutions. New solutions are generated by choosing best moves subject to the influence of this guidance, making use of TS aspiration criteria to pursue exceptional alternatives that qualify as sufficiently attractive. The trajectories can extend beyond the region of the reference solutions, and the best solutions encountered are used as a basis for launching new searches by customary heuristic processes.

The path relinking approach gives a way to refine strategic oscillation by choosing reference solutions that constitute elite solutions from different levels, using them to determine new trajectories that pass through the target level. Elite solutions from the target level itself likewise provide a basis for path relinking processes to launch new searches at this level. (See Glover[16] for further details.)

The POP notion implies a form of connectivity for the search space that may be usefully exploited by this approach. That is, path relinking trajectories guided by elite solutions, whether deterministically or probabilistically, are likely to go through regions where new elite solutions reside, provided appropriate neighborhood definitions are used. The result is a type of focused diversification that is more effective than "sampling." Evidence of such a topology in optimization problems is provided by findings from Moscato[39] and Nowicki and Smutnicki.[42]

1.2. Micro level decisions.

The decision of selecting a target rate of change for strategic oscillation is placed at a micro level because it evidently involves variability of a particularly local form. This type of variability is particularly relevant when tabu search memory structures are set aside, since then it becomes necessary to provide a means of avoiding cycling, i.e., preventing the search from endlessly (and exclusively) revisiting a particular set of solutions. Figure 5 illustrates three alternative rates of change, ranging from mildly to moderately aggressive, where the current direction is one of "descent." Again the changes are shown as broken lines, to suggest they do not always proceed smoothly or uniformly. The issue of using target rates as a component of search strategy may seem evident, but its significance is often overlooked. (For example, in simulated annealing, all improving moves are considered of equal status.)

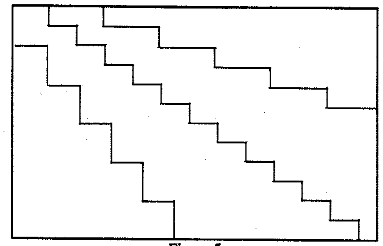
Figure 5
Target Rates of Change
(Micro Level)

Figure 5. Target rates of change (micro level).

When the parameter of oscillation is related to values of the objective function, tabu search normally prescribes aggressive changes, seeking the greatest improvement or least disimprovement possible (in a steepest descent, mildest ascent orientation: see, e.g., Hansen[23] and Hansen and Jaumard[24]). Such an approach applies chiefly to intensification phases and may be tempered or even reversed in diversification phases. More precisely, the notion of seeking aggressive (best or near best) changes is qualified in tabu search by specifying that the meaning of *best* varies in different settings and search phases, where rates of change constitute one of the components of this varying specification. This may be viewed as constituting another level of oscillation, which likewise is applied adaptively rather than subjected to monotonic control.

Accompanying the target rate of change is the micro level decision of choosing a target band of change, which sets boundaries on deviations from the target rate. The lines of the preceding diagrams, as noted earlier, should be interpreted as having a certain breadth (so that the lower limit of one segment may overlap with the upper limit of the next), and the target band of change is introduced to control this breadth.

Targeted rates and bands are not determined independently of knowledge about accessible solutions, but are based on exploring the current neighborhood to determine the possibilities available (hence bending the curves of Figure 5 by a factor determined from these possibilities). The meaning of this will be clarified in later discussions of candidate list strategies.

Finally, aspiration criteria at the micro level permit the controls previously indicated to be abandoned if a sufficiently attractive alternative emerges, such as a move that leads to a new best solution, or to a solution that is the best one encountered at the current oscillation level.

A template for strategic oscillation is given in the appendix for readers interested in additional details of this approach. In the next section, we begin by presenting a simple instance of this scheme in which the decisions about targets and levels are handled implicitly, using the objective function as the element of guidance.

2. The Method in Overview

The skeletal form of a tabu thresholding method in the context pursued here is easy to describe, and we state it first in overview without considering details of its component steps. The method may be viewed as consisting of two alternating phases, an *Improving Phase* and a *Mixed Phase*. The Improving Phase allows only improving moves, and terminates with a local optimum, while the Mixed Phase accepts both non-improving and improving moves.

Choices of moves in the two phases are governed by employing candidate list strategies to isolate subsets of moves to be examined at each iteration, and by a *probabilistic best* criterion. (The meaning of this criterion, which also has a deterministic counterpart, is explained in Section 5.) The thresholding effect of the choice criterion is further influenced by a *tabu timing parameter* t which determines the number of iterations of the Mixed Phase, analogous to maintaining a tenure for tabu status in a more advanced system. Control over t is exerted by selecting lower and upper bounds, L and U, between which t is permitted to vary randomly, or according to a selected distribution.

The form of the method is given as follows, starting from some initially constructed solution and retaining the best solution found throughout its operation.

Simple Tabu Thresholding Procedure in Overview

Improving Phase

(a) Generate a subset S of currently available moves, and let S* be the set of improving moves in S. (If S* is empty and S does not consist of all available moves, expand S by adjoining new subsets of moves until either S* is nonempty or all moves are included in S.)
(b) If S* is nonempty, choose a *probabilistic best* move from S* to generate a new solution, and return to (a).
(c) If S* is empty, terminate the Improving Phase with a locally optimal solution.

Mixed Phase

(a) Select a tabu timing parameter t (randomly or pseudo randomly) between lower and upper bounds L and U.
(b) Generate a subset S of currently available moves, and select a *probabilistic best* move from S to generate a new solution.
(c) Continue step (b) for t iterations, or until an aspiration criterion is satisfied, and then return to the Improving Phase.

Termination of the foregoing procedure occurs after a selected total number of iterations. The set S in these phases may consist of all available moves in the case of small problems, or where the moves can be generated and evaluated with low computational expense. In general, however, S will be selected by a candidate list strategy, as subsequently described, to assure that relevant (and different) subsets of moves are examined, in changing sequences.

The Mixed Phase may be expressed in a format that is more nearly symmetric to that of the Improving Phase, by identifying a subset S* of S that consists of moves satisfying a specified level of quality. However, the determination of S* can be implicit in the rules of the candidate list strategy for selecting S, and further can be controlled by the *probabilistic best* criterion, which itself is biased to favor choices from an elite subset of moves. We introduce S* as distinct from S in the Improving Phase to emphasize the special role of improving moves in that phase.

As noted, this overview procedure is a direct embodiment of the strategic oscillation approach, keying on the movement of the objective function rather than that of other functionals that combine cost and feasibility, or distances from boundaries or stages of construction. In spite of this narrowed focus, it will become apparent that a significant range of strategic possibilities present themselves for consideration, drawing on associated ideas from the tabu search framework.

As a prelude to ideas presented later, we observe that the method offers immediate strategic variability by permitting the Improving Phase to terminate at various levels other than a local optimum, passing directly to the Mixed Phase at each such point. In this instance, the Improving Phase need not rigidly adhere to a policy of expanding S to include all moves, when no improving moves can be found, and we subsequently give guidelines for alternative policies. At an extreme, by suitably controlling nonimproving moves, the method can operate entirely in the Mixed Phase, or alternately, the Improving Phase can be given a more dominant role and the Mixed Phase can be replaced by a Nonimproving Phase.

Temporarily disregarding these supplementary considerations, the simple tabu thresholding approach we have outlined is governed by three critical features: determining the subset S of candidate moves to be considered, defining the probabilistic best criterion for choosing among them, and selecting the bounds L and U that affect the duration of the Mixed Phase. We now turn to examining these features in detail.

3. Candidate List Strategies

Studies of linear and continuous optimization problems sometimes contain useful implications for solving nonlinear and discrete optimization problems. An instance of this is an investigation of linear network optimization approaches for selecting pivot moves in basis exchange algorithms (Glover et al.[19]). Two findings emerged that are relevant for our present purposes: first, a *best evaluation rule*, which always selects a move with the highest evaluation, produced the fewest total pivot steps (from a wide range of procedures examined); and second, a straightforward candidate list strategy proved notably superior in overall efficiency for problems of practical size (in spite of requiring more iterations). The first finding was consistent with outcomes of other related studies. The second finding, however, was contrary to the accepted folklore of the time.

The candidate list strategy underlying that second finding provides the starting point for our present concerns.

The basis of the network candidate list strategy was to subdivide the moves to be examined into subsets, one associated with each node of the network. To characterize the method in terms that provide insights into its general nature, we introduce some notation that will also be useful in the subsequent development. Let MOVE_SET(x) denote the set of all moves associated with a given solution x. An element in MOVE_SET(x), which we denote by MOVE(x), transforms x into a new solution. For example, in the network setting, MOVE(x) can represent a pivot move that generates a new basic solution from the current one. Each such move has an evaluation, denoted EVALUATION (MOVE(x)), which identifies its attractiveness for selection. (A higher evaluation corresponds to a "more attractive" alternative. Thus, if a move is evaluated by reference to the change in c(x), the negative of this change provides the evaluation for a minimization objective.)

For any arbitrary subset of moves in MOVE_SET(x), denoted MOVE_SUBSET(x), define BEST(MOVE_SUBSET(x)) to be the set of best moves in MOVE_SUBSET(x) (those that share the highest evaluation in this subset). The first finding of the network study previously cited can be expressed as saying that fewest total iterations resulted by choosing a move that belongs to BEST(MOVE_SET(x)).

To create a candidate list, divide MOVE_SET(x) into indexed subsets: MOVE_SUBSET(1, x), MOVE_SUBSET(2, x), ..., MOVE_SUBSET(m, x). For the network problem MOVE_SUBSET(i, x) identifies all moves associated with a given node $i \in M = (1, 2, \ldots, m)$; specifically, all pivot moves that introduce one of the arcs meeting node i into the network basis.

One more ingredient is needed to specify the method: an acceptance criterion that determines whether a given move under consideration is acceptable to be chosen. For this we require no added notation, but merely stipulate that a move is acceptable or unacceptable. In the network context, a move is acceptable if it is an improving move; i.e., if EVALUATION(MOVE(x)) indicates that the move will yield an improved solution (under conditions of nondegeneracy).

The subsets MOVE_SUBSET(i, x) for $i \in M$ operate as candidate lists of moves to be examined, because the procedure seeks a best move from each such subset, considered in turn, as a candidate for selection, rather than seeking a best move over the complete collection of moves in MOVE_SET(x). Strictly speaking, once a move from a given MOVE_SUBSET(i, x) has been selected and executed, the structure of moves in other subsets may be changed. However, the moves typically can be identified by reference to component attributes (or "associations") that permit the subset classifications to remain applicable.

With these preliminaries, the method may be described as follows.

Candidate List Strategy (CLS)

Step 0. (*Initialization.*) Start with the set REJECTED empty and set $i := 0$ (to anticipate examination of the first subset).

Step 1. (*Choose a candidate move from the current subset.*) Increment i by setting $i := i + 1$. (If i becomes greater than m, set $i := 1$.) Select CANDIDATE_MOVE to be a move that is a member of BEST (MOVE_SUBSET(i, x)).

Step 2. (*Execute the move or proceed to the next subset.*)

2A. If CANDIDATE_MOVE is acceptable: Execute the move to produce a new solution. Then reset REJECTED to be empty and return to Step 1.

2B. If CANDIDATE_MOVE is unacceptable: Add i to REJECTED. If REJECTED = M, stop (all subsets have been examined and none have acceptable candidate moves). Otherwise, return to Step 1.

The success of this simple strategy soon led to a number of variations. It was observed, for example, that defining MOVE_SUBSET(i, x) relative to nodes of the network could cause these sets to vary somewhat in size (in non-dense networks), thus making them "unequally representative." Improvements were gained by redefining these sets to refer to equal sized blocks of moves (in this case, blocks of arcs, since each nonbasic arc defines a move). This change also made it possible to choose the number of subsets m as a parameter, rather than compelling it to equal the number of nodes. Good values for m in the network setting were found to range robustly over an interval from 40 to 120, although it appeared to be better to put each successive block of m "adjacent" arcs into m different subsets (Glover et al.,[19] Mulvey[41]). In each instance, it also appeared preferable to retain the policy of examining all subsets before returning to the first. Extended variations of this basic candidate list strategy will be discussed later.

4. Tabu Search

A notion that is complementary to the candidate list ideas, and that provides a foundation for applying them more broadly, emerged in the late 1960's and early 1970's with the following stipulation. In order to solve problems from the domain of general discrete optimization (as contrasted to the domain of linear optimization, exemplified by networks) it is unnecessary to base acceptance of a move on the ability to create an improvement. Rather, measures of improvement may be allowed to fluctuate by introducing controls to prevent reversals of certain changes induced by the moves, such as increments in reference equation values, variable augmentation indexes and edge extensions (see e.g., Glover[13,14]).

The approaches that embodied this notion shared an element in common with the candidate list approaches: an aggressive strategy of seeking the best move at each step from a restricted (and dynamically updated) class of admissible alternatives. The provision that measures of improvement could fluctuate in the presence of such controls was later elaborated and adapted, by introducing special forms of flexible memory, to create the class of methods now known as tabu search methods.

For the approach of this paper, we limit attention to only a few of the basic elements of tabu search, as follows. Let TABU(x) denote a subset of moves in MOVE_SET(x) that are rendered tabu, i.e., that are currently forbidden to be chosen. (The usual memory based criteria for composing TABU(x) will be disregarded for our present purposes.) Let ASPIRE(x) identify a subset whose evaluations are sufficiently attractive (or whose attributes are sufficiently favored) to permit them to be chosen in spite of their tabu classification. Finally, let TABU*(x) be the set of *strictly tabu* moves, given by TABU*(x) = TABU(x) − ASPIRE(x).

There are three customary ways of handling tabu conditions.

(1) *Explicit tabu restrictions*: Let TRIAL_SET(x) denote a set of moves currently being examined to yield a candidate move (such as a collection of one or more of the subsets MOVE_SUBSET(i, x)). Then consideration of TRIAL_SET (x) is restricted to allow moves to be selected only from the set

$$\begin{aligned}&\text{ADMISSIBLE(TRIAL\_SET(x))}\\&\quad = \text{TRIAL\_SET(x)} - \text{TABU}^*(\text{x}).\end{aligned}$$

(2) *Tabu penalties and incentives*: The evaluation of a given move is modified according to whether the move is tabu or satisfies current aspiration criteria. We express this modification as the outcome of creating associated penalty and incentive values, PENALTY(MOVE(x)) and INCENTIVE (MOVE(x)), where the penalty value is positive if MOVE(x) belongs to TABU(x) and the incentive value is positive if MOVE(x) belongs to ASPIRE(x). Then EVALUATION (MOVE(x)) is replaced by

$$\begin{aligned}&\text{TABU\_EVALUATION(MOVE(x))}\\&\quad = \text{EVALUATION(MOVE(x))}\\&\qquad - \text{PENALTY(MOVE(x))}\\&\qquad + \text{INCENTIVE (MOVE(x))}\end{aligned}$$

Penalties and incentives in general reflect degrees of being tabu or of satisfying aspiration criteria, and may be incorporated other than by adding and subtracting explicit associated terms. (For example, they are often integrated to allow an incentive to exceed an associated penalty only under restricted conditions.)

(3) *Probabilities governing selection*: Probabilities are assigned as a function of TABU_EVALUATION(MOVE(x)), as identified in (2), so that moves with higher tabu evaluations receive higher probabilities of being selected. Generally, higher evaluations are disproportionately favored over lower ones. The probabilities sum to 1 over TRIAL_SET(x) (the set of moves currently considered), so that a candidate move is identified at each application.

It may be noted that the probabilities of (3) are not "acceptance" probabilities in the sense employed in Monte Carlo or simulated annealing processes, since they always select a candidate move. Moreover, where more than one candidate is generated, and none meet the current acceptance standard, a final choice again may be made by (3). We simplify this in our present development so that positive probabilities are given only to acceptable alternatives, making a secondary acceptance criterion unnecessary (except where this criterion is simply a requirement for improvement).

The tabu controls embodied in the three preceding instances have the following immediate goals: to avoid cycling, to provide flexibility to pursue the search aggressively, and to introduce a basic form of diversity into the search. We will refer to a solution trajectory generated by these principles as a tabu path, or T-path.

5. A Tabu Thresholding Procedure

In order to design a simple procedure that does not rely on memory to create a T-path, we start from the premise that such a path should adaptively determine its trajectory by reference to the regions it passes through. Thus, instead of obeying an externally imposed guidance criterion, such as a monotonically changing temperature, we seek to reinforce the tabu search strategy of favoring behavior that is sensitive to the current search state, accepting (or inducing) fluctuations while seeking best moves within the limitations imposed. Evidently, this must frequently exclude some subset of the most attractive moves (evaluated without reference to memory), because repeated selection of such moves otherwise may result in repeating the same solutions.

A basic tenet of probabilistic tabu search is that randomization, if strategically applied, can be used to perform certain problem-solving functions of memory (Glover[17]). In the present context, the probabilistic TS orientation suggests that choosing moves by reference to probabilities based on their evaluations (attaching high probabilities for selecting those that are *near best*) will cause the length of the path between duplicated solutions to grow, and this will provide the opportunity to find additional improved solutions, as with nonprobabilistic TS methods. By this same orientation, we are motivated to inject a probabilistic element into the manner of choosing move subsets in a candidate list approach, to create a reinforcing effect that leads to more varied selections.

As noted in the overview of the method described in Section 1, we seek to achieve these goals by dividing the solution process into two phases, an *Improving Phase* and a *Mixed Phase*. During the Improving Phase we employ the Candidate List Strategy (CLS) in exactly the form specified in Section 2, except for the introduction of a special procedure for scanning M; i.e., for identifying a sequence for examining the subsets MOVE_SUBSET(i, x), i ∈ M. Once a local optimum is reached by this phase, the Mixed Phase is activated, again applying CLS by reference to an approach for scanning M, in this case a variant of the strategy used in the Improving Phase. The two phases are alternated, retaining the best solution obtained, until a selected cutoff point is reached for termination.

The resulting method will first be sketched in an outline form similar to that of the overview method of Section 2, after which its components will be explained.

Tabu Thresholding Procedure (Specialized)

Improving Phase.

(a) Apply CLS by a Block-Random Order Scan of M, accepting a candidate move if it is improving.
(b) Terminate with a local optimum.

Mixed Phase.

(a) Select the tabu timing parameter t randomly or pseudo randomly between L and U.

(b) Apply CLS by a Full-Random Order Scan of M, automatically accepting a candidate move generated for each given $i \in M$ examined.

(c) Continue for t iterations, or until an aspiration criterion is satisfied, and then return to the Improving Phase.

The bounds L and U in the Mixed Phase can be set to values customarily used to bound tabu list sizes in standard tabu search applications. For example, setting L and U to bracket a simple function of the problem dimension, or in some instances setting L and U to constants, may be expected to suffice for many problems in the short term. (The same types of rules used to set tabu tenures in tabu search may give reasonable guidelines for the magnitude of these values.) Control of L and U over the longer term will be discussed later.

In the most direct case, the aspiration criterion for allowing early termination of the Mixed Phase can consist of seeking a solution better than the best previously found. It is to be noted that the Improving Phase may perform no iterations after leaving the Mixed Phase (i.e., the Mixed Phase may terminate with a local optimum, in which case it will be executed again).

We will specify two levels of implementing the method.

Level 0. Select a candidate by reference to BEST (MOVE_SET(i, x)), as specified in CLS.

Level 1. Select a candidate by replacing BEST with PROBABILISTIC_BEST.

Level 0 is a simplified form of the method that is relevant only for large problems.

We include it because it is fast and because it can be viewed as a component of the approach at the next level. The determination of PROBABILISTIC_BEST, which provides an explicit definition for the *probabilistic best* criterion discussed in Section 1, lies at the heart of the Level 1 procedure. We consider the two levels of implementation as follows.

Level 0 and Scanning M

The Level 0 version of the method operates precisely as the original CLS except for the procedures used to scan M. The method is completely determined by identifying the form of these procedures. In the original CLS and its variants, all scanning processes involve sampling without replacement; i.e., randomization is constrained to examine all elements of a set, in contrast to performing sampling that may revisit some elements before scanning others. (Strategies for revisiting elements with tabu search are allowed only by restricting the frequencies of such visits.) One way to apply the CLS prescription for scanning the indexes of M, i.e., for examining their associated subsets MOVE_SUBSET(i, x), $i \in M$, is to start completely fresh after each full scan, to examine the elements in a new order. This approach has the shortcoming that some subsets can become reexamined in close succession and others only after a long delay. We deem it preferable for each subset to be re-examined approximately m iterations after its previous examination. This leads to the following stipulation for the examination sequence of the Improving Phase.

Block-Random Order Scan: Divide M into successive blocks, each containing a small number of indexes relative to m (e.g., Min(m, 5) if $m < 100$, and otherwise no more than $m/20$). As a given block is encountered, reorder its elements randomly before examining them, thus changing the sequence of this portion of M. The index i of the CLS strategy thus identifies *positions* in M, rather than elements of M.

Unless the number of elements in a block divides m, which can be countered by varying the block sizes, the block-random order scan permits the resequenced elements gradually to migrate. On any two successive scans of M, however, a given element will be scanned approximately m moves after its previous scan.

The scanning procedure for the Mixed Phase is similar, and determines the initial conditions for the scan of the next Improving Phase.

Full-Random Order Scan: A full-random order scan corresponds approximately to selecting a block of size m; all elements are potentially re-ordered. (Once a local optimum is reached, such a re-ordering of M is conceived appropriate.) The simplest version of this scan is to randomly re-order M, subject to placing the last element that yielded an improving move in the Improving Phase at the end of M, and then scan the elements in succession. If $t > m$, the process reverts to a block-random order scan after all of M is examined.

This scan is resumed in the subsequent Improving Phase at the point where it is discontinued in the Mixed Phase. A preferable size for MOVE_SUBSET(i, x) in the Improving Phase may not be ideal for the Mixed Phase, and in general several such subsets may be combined at each iteration of the Mixed Phase as if they composed a single subset, for the purpose of selecting a current move. If the value $m - t$ is large, time may be saved by randomly extracting and re-ordering only the sets actually selected from M during the Mixed Phase, without randomly re-ordering all of M.

The combined effect of randomization and non-replacement sampling in the preceding scanning processes approximates the imposition of using tabu restrictions recency-based memory. The use of the block-random order scan in the Improving Phase (and in the Mixed Phase, once all elements of M have been examined), succeeds in preventing the selection of moves from sets recently examined, starting from each new local optimum. This will not necessarily prevent reversals. However, randomization makes it unlikely to generate a sequence consisting of moves that invariably correspond to reversals of previous moves, and hence acts as a substitute for memory in approximating the desired effect. If moves can be efficiently isolated and reassigned upon the initiation of the full-random order scan, the composition of moves within subsets also may be randomly changed.

The rudimentary Level 0 version of the method is now fully specified. The indicated rules for scanning M produce a candidate move at each iteration, identified as an element of BEST(MOVE_SUBSET(i, x)), exactly as in the original CLS. We next describe the criterion that replaces the use of BEST with PROBABILISTIC_BEST, to provide the Level 1 version of the procedure.

Level 1 (Probabilistic Candidate Selection)

The function PROBABILISTIC_BEST, represented notationally as a set containing the proposed candidate, is created as follows. A subset of r best (highest evaluation) moves is extracted from MOVE_SUBSET(i, x) (e.g., for $r = 10$ or $r = 20$). The current evaluations of these moves are then used a second time to generate probabilities for their selection. A small probability may be retained for choosing an element outside this set.

A method that finds r best elements from a set in significantly less time than classical methods (or alternatively, that finds "approximately" r best elements to a desired level of approximation) is given in Glover and Tseng.[22]

Selecting a Candidate

Once a set R of r best moves has been identified, a simple way to determine the probabilities for selecting an element from this set is as follows. Let R_MIN equal the smallest evaluation over R, and create a normalized evaluation.

$$\text{NORM(MOVE(x))} = \text{EVALUATION(MOVE(x))} + \epsilon - \text{R\_MIN},$$

where ϵ is given a value representing a "meaningful separation" between different evaluations (or is determined by more refined scaling considerations). Further, let NORM_SUM denote the sum of the normed evaluations over R. Then we may specify the probability for choosing a given MOVE(x) to be

$$\text{PROBABILITY(MOVE(x))} = \text{NORM(MOVE(x))/NORM\_SUM}$$

Note that the evaluation function itself may be modified (as by raising a simple function to different power) to accentuate or diminish the differences among its assigned values, and thus to produce different probability distributions.

In the Mixed Phase, a candidate move selected by this probabilistic criterion is automatically deemed acceptable for selection. Automatic acceptance also results for the Improving Phase, since only improving moves are allowed to compose the alternations considered. A special case occurs for the transition from the Improving Phase to the Mixed Phase, since the final step of the Improving Phase performs no moves but simply verifies the current solution is locally optimal. To avoid wasting the effort of the scanning operation of this step, an option is to retain a few best moves (e.g., the best from each of the r most recently examined MOVE_SUBSET's), and to use these to select a move by the PROBABILISTIC_BEST criterion, thus giving a "first move" to initiate the Mixed Phase.

By the preceding stipulations, the Level 1 version of the method also is fully specified.

6. Implementation Considerations and Enhancements

We examine ways to enhance the tabu thresholding method that maintain the feature of convenient implementation, considering the dimensions of efficiency and solution quality.

6.1. Efficiency Enhancements

Three straightforward approaches to accelerate the method invite consideration. First, and simplest, the method may examine a self-adjusting fraction f of the elements of M to seek an improving move in the Improving Phase. The value of f increases by steps to 1 (through a sequence such as .3, .5, 1) as the current objective function value falls within intervals progressively closer to the best value found or projected. This approach is relevant for problems involving large numbers of moves, and where improving paths are not typically "long and narrow." A learning procedure such as target analysis (Laguna and Glover[35]) can be used to determine how to set f as a function of other parameters.

The idea behind this approach is more generally to permit the method to stop short of achieving (or verifying) local optimality under conditions where a local optimum is not likely to be particularly attractive relative to an aspiration level (e.g., established by the best solution found). Factors allowing early termination of the search for an improving move include: the quality and distribution of moves examined during the scanning process, the distance of the current objective function value from the aspiration level, numbers of improving moves seen in recent iterations (and identities of better ones). The latter elements introduce memory considerations and lead to the issue of more refined candidate list strategies, discussed later.

The second approach for accelerating the method, which is often effective in tabu search implementations, consists of subdividing the moves into the set of those whose evaluations change after executing a specified move and the set of those whose evaluations do not change (Barnes and Laguna[1]; Hertz and de Werra[27]). The second set generally is much larger than the first. By recording move evaluations, and by updating only the relatively small number that change, efficiency can often be significantly increased. (This likewise involves some concession to the inclusion of memory, but it is memory of a very simple kind.)

Such an approach, when conditions exist that permit it to apply, will normally enable MOVE_SUBSET(i, x) to contain a somewhat larger number of elements then would otherwise be possible. This will reduce the total number of such subsets (sometimes to 1), and cause the PROBABILISTIC_BEST criterion of Level 1 to gain increased importance relative to the manner of scanning M.

This second approach also can give the basis for a parallel computation strategy. Specifically, moves are allocated to the subsets so that some collection of these subsets satisfies a noninterference property, which stipulates that moves of one member of the collection can be executed without affecting the moves (or evaluations) of other members. Then each member of the collection is processed in parallel for a chosen number of iterations, whereon the moves are re-allocated, or the collection otherwise is redefined, and the process repeats.

The third type of acceleration occurs by a screening approach, using a partial evaluation process to isolate potentially attractive moves before applying a full evaluation. Moves often contain identifiable components that must have an attractive evaluation if the complete move is to qualify as good. In graph problems, for example, moves that add and drop multiple pairs of edges generally can be screened into promising and less promising sets according to whether the first pair of edges added and dropped yields an improving evaluation. If the multiple pairs that define a given class of moves range over all combinations of component pairs, then consideration can be limited during an Improving Phase to moves with *improving first pairs.* This type of screening has been used effectively in a local descent process by Johnson,[31] and can be extended by requiring that the combined evaluation of the first two pairs must be improving, and so on.

In the more general setting of the Mixed Phase, where good moves may not be improving, we undertake to select moves that contain at least one *good component*—i.e., a component which, if not improving, is disimproving only to a limited degree. Generating thresholds to identify limits of this type, and thereby to screen moves by partial evaluations, can reduce the total time to examine moves from a polynomial time function to a linear time or even constant time function.

6.2. Solution Quality Enhancements

For difficult problems, it is appropriate to seek improved solutions by a tabu search principle applicable to the longer term. The basis of this principle is to organize moves into classes according to their ability to induce different kinds or degrees of change in the current solution, or in values of selected functions associated with the solution. In our present approach we exploit this organization by assigning members of a given class to a particular candidate list subset. A rule is then imposed that requires such subsets to be periodically selected as the source of candidate moves, particularly upon encountering transition conditions such as local optimality.

Motivation for applying this concept is provided by the following observation. Under normal circumstances, high evaluations will rarely be accorded to *high influence* moves, such as those that create large changes in the objective function or constraints (e.g., edge exchanges where one or more edges have a large weight). Nevertheless, high influence moves can be essential to trajectories that lead to improved solutions. This observation is commonly implemented in one of two ways: by giving preference to a high influence move whenever its evaluation lies acceptably close to the best evaluations of other moves (e.g., bounded by a requirement for improvement), and by periodically restricting consideration to high influence moves during a succession of iterations when no admissible moves are improving.

To provide a concrete illustration of a way to exploit this idea in the tabu thresholding method, we create move subsets of the form MOVE_SUBSET(i, x), $i \in M^*$, where M^* is an index set for one or more specific classes of high influence moves. Elements in the new subsets may duplicate elements found in original subsets, or may be based on different neighborhood structures that generate uncommon changes. When moves of a given class are collected in the same MOVE_SUBSET(i, x), the operation of CLS magnifies the likelihood that at least one will be selected. In addition, during the Mixed Phase, elements of M^* can be compelled to be included among the t elements selected from M.

To achieve this effect in a more tightly structured way, we suggest that MOVE_SUBSET(i, x) should be constituted to contain moves of generally higher (or no lower) influence than MOVE_SUBSET(i + 1, x), for all $i < m$. Then M is partitioned to subdivide these consecutively ordered subsets into a small number of groups (e.g., 2 to 4), to permit the method to focus on higher influence moves hierarchically. In particular, the Improvement Phase provides a basis for a staged process that operates on each group in turn as if it comprised all of M. By this design, the process cycles through elements from a given group until no improvement is possible, and then progresses to cycle through elements from the next group, ultimately returning to the first group after visiting the last. When none of the groups yield an improvement a local optimum is reached. The block-random order scan operates to change the order of elements within a group, but is not permitted to cross boundaries between groups and thereby change their compositions. During the Mixed Phase this process can consist of modifying the full-random order scan to give a probabilistic bias to favor the elements of M associated with higher influence moves. Again, the effect is to reorder elements within groups, but not to alter the composition of the groups themselves.

General tabu search methods typically incorporate frequency-based memory to isolate uncommonly selected alternatives as candidates for appropriate high influence moves (Skorin-Kapov and Vakharia[51]; Tailliard[53]; Laguna and Glover[35]). Provided features of such moves can be identified in advance, improved strategies undoubtedly may be formulated without reliance on memory. We acknowledge that a concession to a simple counting record can be useful to induce elements of M that are rarely or never chosen to be selected periodically, especially in the Mixed Phase.

7. Variations

Two primary variations of the tabu thresholding method likewise deserve consideration as a supplement to the alternatives for enhancing efficiency and solution quality. The first variation involves replacing the Mixed Phase by a Non-Improving or Disimproving Phase (depending upon whether improvement is interpreted as "better than" or "at least as good" in the Improving Phase). Such a change is analogous to increasing the strength of tabu restrictions, and accordingly may be expected to result in smaller preferred values for L and U than in the Mixed Phase. Recent theoretical results by Ryan[49,50] identify values for U in various combinatorial problem settings that assure a path to an optimal solution will exist, and may provide useful guidelines in the present context.

A progressive restriction approach can maintain L = 0 and slowly decrease U from a large value to 0. This exerts an indirect influence on the objective function variation, by controlling numbers of steps of nonimprovement (employing a tabu search type of restriction). To induce greater diversification, L and U periodically may be boosted to larger than customary values over the longer term.

The second chief variation is to modify the definition of the function PROBABILISTIC_BEST, providing an opportunity to encompass additional considerations in the determination of probabilities for selecting candidates. Differences in the evaluations used to generate these probabilities can be accentuated or attenuated by raising NORM (MOVE(x)) to a nonnegative power p, where accentuation occurs for p greater than 1 and attenuation occurs for p less than 1. (The value p = 0 yields complete randomization without regard for the evaluations.) Such norms are replaced by their new values before computing NORM_SUM. Allowing p to assume changing values, or progressively incrementing p from 0 to larger values, offers a way to compound this form of variation.

Again to provide a concrete example of relevant alternatives, the definition of PROBABILISTIC_BEST may be shifted differently in this variation by redefining

$$
\begin{aligned}
&\mathrm{NORM(MOVE(X))} \\
&\quad = \mathrm{MAX} + \epsilon - \mathrm{MEAN} \\
&\qquad - |\mathrm{EVALUATION(MOVE(X))} - \mathrm{MEAN}|
\end{aligned}
$$

where MAX and MEAN are the maximum and mean evaluations over the set of candidate moves, and ϵ is given as in the previous definition of NORM(MOVE(X)). Use of this alternative norm assigns higher probabilities to selecting evaluations closer to the mean, and hence *pulls away* from selecting the types of moves chosen under the original definition. However, the altered probabilities retain an element of aggressiveness, since they likewise are defined over a set of near best elements, and are restricted to improving moves during an Improving Phase. Such a modified probabilistic best criterion is appropriate for a longer term diversification strategy, periodically invoking the changed norm to introduce moves less frequently selected, without abandoning the quest for an aggressive search path.

8. More General Candidate Lists

The form of candidate list structure considered in the preceding sections is not appropriate for all types of problems. Other forms can be particularly relevant in cases where moves cannot be conveniently partitioned into subsets. An approach that preserves the orientation of the method previously described is the *abc candidate list strategy*, where the symbols a, b, and c refer to parameters for which $a \leqslant c$. In this approach, the move to be selected at each iteration is determined by generating at least a and at most c elements, stopping before reaching c elements if the b *best* elements generated satisfy a chosen aspiration condition. For example, during an Improving Phase, the aspiration condition may require that b of the moves generated are improving moves (possibly balancing the value of b with a desired level of improvement), where all improving moves qualify as candidates if the process reaches c elements without satisfying the requirement. To accord with earlier stipulations, we allow c to be the number of all moves in an Improving Phase, or a specified fraction f of these moves. (The factors for determining f, and hence c, can include the aspiration condition that gauges the quality of the b best moves currently generated.) Values of the parameters a, b, and c typically will be smaller in the Mixed Phase than in the Improving Phase, and the aspiration condition associated with b then will be less stringent.

When the problem and neighborhood structure permit, an abc candidate list strategy can be advantageously superimposed on the type of candidate list strategy that partitions moves into subsets. A benefit of retaining the MOVE_SUBSET structure within such an implementation is the ability to assure that fundamentally different move possibilities are examined in a successively exhaustive manner. In an implementation based on move influence, such a structure alternately can assure that moves of certain classes are examined accordingly to preferred sequences or frequency of selection.

As a final consideration in the construction of candidate lists, an *elite element* strategy can be applied by combining ideas from Mulvey[41] and Frendewey and Glover.[10] The basic theme is to periodically expend increased effort to generate a set of preferred moves, which then provides a source for selecting candidate moves during interim iterations. During the stage of increased effort, this type of method constructs a master candidate list composed of best members identified by examining all, or a large subset, of the moves available (as by applying an abc approach with large values of a and c). In the settings where special instances of this method have been implemented, reference is made to subsets of the form MOVE_SUBSET(i, x), selecting at most one move from any given subset, to create a master list that consists of at most 20 to 40 of the best such moves overall. A reasonable alternative is to select multiple moves from a given subset, e.g., up to 40/m, if m is not large. At each iteration, a best member is chosen from the master list until the highest evaluation falls below a chosen threshold (e.g., MIN + (MAX − MIN)/4, where MIN and MAX refer to evaluations of elements admitted to the master list during the step when it was most recently constructed). Then the master list is rebuilt and the process repeats.

In the tabu thresholding approach, such a master list can be the source of the r best candidates of a Level 1 approach. Block-random and full-random order scanning become irrelevant in this case, since a large number of the MOVE_SUBSETs will be scanned to construct the master list. Instead, the same screening used in the Level 1 approach can introduce probabilistic variation into the construction of the master list. In particular, a candidate to include on the master list can be selected probabilistically from each MOVE_SUBSET(i, x) by reference to the several best elements of this subset. With these provisions, the procedure continues to operate as specified by the general description in Section 2.

9. Parallel Processing Applications

The tabu thresholding approach is particularly relevant for parallel processing implementations. The notion of influential moves discussed in Section 6.2 motivates a staged parallel process that starts by constructing solutions out of higher influence elements (e.g., with larger cost, feasibility or interdependency effects) and then gradually introducing elements of lesser influence. Hierarchies of this type may be defined relative to nodes or edges in a graph, jobs or machines in a schedule, variables or constraints in a mathematical program, and so forth. The process effectively proceeds from a *coarse grain* level to a *fine grain* level, where elements introduced at later stages permit refinement of outcomes at earlier stages.

The tabu thresholding approach can be organized naturally for parallel processing to permit each such stage to be run asynchronously, i.e., without being interrupted for coordination. The aggressive search orientation of the approach is conducive to generating a number of good solutions early in the execution of each process, thus making it reasonable to establish a common time point to end the processes executed during a given stage and to establish conditions for the stage to follow. Although a chance to do better may result by running a stage longer, it is likely to be sufficient during earlier stages to allot only enough time to obtain a few good local optima. Later stages may run for progressively increasing durations to uncover refinements made possible by the new elements they introduce. Thus, upon establishing hierarchies of decreasing influence, the process may operate as follows:

Parellel Processing by Descending Influence Level

Step 1. Initiate q parallel solution streams, incorporating elements at the highest influence level.

Step 2. In each solution stream, retain up to the p best local optima encountered during the period allotted to the current stage.

Step 3. At the end of the stage, coordinate by selecting the q best solutions from the union of solutions retained by the separate streams.

Step 4. Assign one of the selected solutions to each processor (allowing duplications if the union of solutions does not contain q distinct members). Add the new elements to be considered at the next lower influence level, and return to Step 2 for the next stage, until all stages are completed.

During the final stage of the foregoing process, each stream simply saves its single best solution. The definition of stage may be broadened so that instead of terminating after all elements are incorporated, the method continues to execute some number of additional stages while operating on the full set of elements. (At the extreme, all elements may be introduced at the beginning, disregarding differences in influence levels.) Efficiency gains may be expected by culling out elements that have not been incorporated in any of the good solutions previously encountered, provided these elements have been available for incorporation for some minimum number of stages.

We suggest such a process be applied relative to different measures of influence, to determine which ones are heuristically more effective. Influence measures can be integrated with a standard measure of attractiveness, to give a composite measures representing a "bang for buck" criterion. More broadly, conditional measures of influence are relevant to staging the parallel process, where the preferred new elements to introduce at each stage depend on the composition of the best solutions thus far generated. For example, a small collection of new elements may be added to yield the best local extension of a current solution. This provides an option of operating each solution stream with its own preferred set of elements, where best candidate solutions to initiate the subsequent stage carry with them their own, progressively enlarging, element pools.

The probabilistic aspect of the tabu thresholding procedure contributes to the assurance that solutions from different streams will be different. These differences can be further accentuated by selecting solutions to be carried forward from one stage to the next according to criteria for identifying diverse subsets of good solutions (Glover[18]).

The exploitation of parallel processing by reference to stages based on influence measures also can be achieved by a parametric variant of the preceding approach. Beginning with a full set of elements, the approach alternately can be applied to create stages by an operation of parameterizing problem coefficients such as costs or capacities. This creates hierarchical differences that gradually become more refined at successive levels, until the coefficients ultimately receive their true values. The idea again is to proceed in descending levels of influence, from coarse grain to fine grain considerations.

The notion of staging solution processes relative to measures of influence, and embedding these stages in a parallel processing approach, is anticipated to accomplish more than the customary goal of accelerating solution time. If this form of exploiting influence measures is strategically sound, the outcome should also lead to generating solutions of superior quality.

Appendix: Further Connections to Strategic Oscillation

To put the observations of this paper in broader perspective, we identify a specific template for strategic oscillation that embraces the tabu thresholding procedure of Section 3, and show how the thresholding procedure arises from this template. We also subsequently indicate a simple memory structure for implementing strategic oscillation in some common types of problems.

We begin with a *goal function* $g(x)$ related to (and possibly the same as) the objective function $c(x)$. We also make use of an *oscillation function* $\theta(x)$, together with control limits α and β, with the purpose of choosing moves to minimize the goal $g(x)$ while maintaining $\alpha \leqslant \theta(x) \leqslant \beta$. The control limits α and β are parameters that are adjusted to produce oscillations in $\theta(x)$, subject to the tolerance

$\Delta = \beta - \alpha$. The oscillation is considered to have a *positive (negative) direction* during a sequence of steps in which α and/or β periodically increases (decreases), and neither of them decreases (increases).

The oscillation function $\theta(x)$ may represent a variety of elements, according to the focus of strategic control. For example, $\theta(x)$ can be a feasibility/infeasibility measure that indicates how far inside or outside the feasible region x lies, taking different signs on different sides of the feasibility boundary. Different types of feasibility, such as those relative to integer requirements and inequality requirements, can be normalized and combined by adaptively chosen weights, but they also may be treated "separately"; i.e., $\theta(x)$ can be a vector valued function and α and β vector parameters. (This applies as well to situations where $\theta(x)$ may measure a stage of construction.) Strictly speaking, $\theta(x)$ is not simply a function of x, but also of the current neighborhood of moves available from x. In tabu search, this means that $\theta(x)$ implicitly depends on search history.

The goal function g(x), and the moves that seek to minimize it, are designed to take account of the current state and direction of oscillation (the bounds on $\theta(x)$ and their direction of change). For example, depending on whether x is currently feasible, and whether the search is moving toward or away from a feasibility boundary, it is natural to employ different penalty or inducement measures to evaluate candidate moves. The parameters α and β need not be treated as rigid bounds on $\theta(x)$, but can be considered as target values for such bounds. The responsive control provided by such measures gives the search process a vitality that often is lacking in search without such an oscillation component.

A template for strategic oscillation based on these elements may be characterized as follows. The method begins by identifying the goal function g(x) and oscillation function $\theta(x)$, and by establishing initial values for the parameters α and β, together with a chosen direction of oscillation.

Strategic Oscillation Template

1. Select a move at each iteration by an evaluation that seeks to minimize g(x) subject to $\alpha \leqslant \theta(x) \leqslant \beta$.
2. Repeat Step 1 for a chosen number of iterations (until a criterion is satisfied to trigger a parameter change). Then update α and β by increasing or decreasing both, according to the current selected direction of oscillation.
3. Repeat Steps 1 and 2 for a chosen number of iterations until a criterion is satisfied to trigger a direction or tolerance change. Then reverse the direction and/or modify the tolerance of the oscillation.
4. Repeat steps 1–4 until satisfying a termination condition (such as an overall cut off limit).

In applying the foregoing process there is no need to restrict consideration to a single g(x), $\theta(x)$ pair, and hence the process may operate as a subroutine to a procedure that selects and periodically updates g(x) and $\theta(x)$.

The tabu thresholding procedure is evidently a rudimentary form of the strategic oscillation provided by this template, effectively choosing g(x) = c(x). Control is exerted through a $\theta(x)$ that is defined relative to the current state of search. That is, $\theta(x)$ may be conceived simply as an indicator function that takes the values 0 or 1 according to whether the search is in an Improving Phase or a Mixed Phase. (Hence $\theta(x)$ implicitly relies on history, as in customary tabu search applications, though in this case with no substantive recourse to memory.) The basis of strategic control lies in the transition between these two phases and in the associated processes for seeking moves to minimize g(x) as detailed in the paper.

Tabu Memory for Strategic Oscillation

To complement the focus on processes that make negligible recourse to memory, and to move from general to more specific concerns, we conclude by describing a simple type of memory for controlling strategic oscillation processes in commonly encountered classes of optimization problems. We refer specifically to problems where moves can be interpreted as "adding" and "removing" elements from a solution, which is represented by a zero-one vector, and where each solution is classified as lying either "below," "above," or "on" a *strategic oscillation boundary* (which corresponds to the target level of Section 2). For the class of problems considered, we specify that the boundary is reached from a solution *below the boundary* by progressively adding elements to the solution (changing values from 0 to 1), and from a solution *above the boundary* by progressively removing elements from the solution (changing values from 1 to 0). Further we stipulate that boundary solutions include all solutions necessary to be considered as candidates for optimal solutions.

A common example of such problems occurs in graph theory where the boundary corresponds to a state of complete construction, which is reached by progressively adding nodes or edges to a graph with too few elements, or by progressively removing nodes or edges from a graph with too many elements. A somewhat different type of example (with a more complex boundary) is the multidimensional knapsack problem, which has the property that a feasible solution can be "extended" by progressively setting variables equal to 1 until reaching a maximally feasible solution (which becomes infeasible if additional variables are set to 1), and an infeasible solution can be "reduced" by progressively setting variables equal to 0 until reaching a minimally infeasible solution (which becomes feasible if additional variables are set to 0). The boundary consists of maximally feasible solutions together with feasible solutions reached in one step from minimally infeasible solutions. (The maximally feasible solutions will suffice if the process of crossing from infeasibility to feasibility is accompanied by moving back to maximal feasibility before setting additional variables to 0.) Generalized covering problems provide an example with analogous feasibility properties, where the role of setting variables to 0 and 1 is reversed. It should be noted in each of these instances that the choice rules for implementing strategic oscillation will customarily differ when approaching the boundary from different directions, and also will appropriately become more "judicious" in close proximity to the boundary. These are important practical aspects that impart an enriched heuristic component to the search.

A memory structure to take advantage of these problems is as follows. We assume the oscillation process is two-sided, i.e., that it operates by moving to and then crossing the boundary from a given side, and then progresses for some number of steps before turning around and similarly approaching and crossing the boundary from the other side. (Modifications of subsequent comments to apply to a one-sided oscillation process will be apparent.)

Let $x(k)$ denote the kth boundary solution generated during this process, for $k = 1, \ldots, k^*$. The memory structure we employ maintains a record of the last t such boundary solutions, i.e., the solutions where $k^* - (t - 1) \leqslant k \leqslant k^*$. We further maintain a special record denoted TABU that is the sum of these last t solutions. That is, starting with TABU equal to the 0 vector, each time a new $x(k^*)$ is identified, set

$$\text{TABU} = \text{TABU} + x(k^*) - x(k^* - t),$$

disregarding the term $x(k^* - t)$ if $k^* \leqslant t$. (A circular list can be used to record a selected number t_0 of the most recent $x(k)$ solutions, allowing t_0 to be the largest t value considered relevant—as where a long term strategy of periodically varying t is employed—and moving around the circle to write each new solution $x(k^*)$ over the oldest solution $x(k^* - t_0)$ A simple offset pointer keeps track of the location of $x(k^* - t)$.)

To specify the rule for controlling the oscillation with this memory, we note the method has two turn-around points, an upper turn-around point where the method discontinues setting variables equal to 1 and prepares to set them equal to 0, and a lower turn-around point where the method discontinues setting variables equal to 0 and prepares to set them equal to 1.

First consider the lower turn-around point. We seek to impose the requirement that the first variable x_j set equal to 1 after reaching this point must not have received a value of 1 in any of the previous t boundary solutions. (Hence, if this requirement is satisfied the boundary solution generated next will be compelled to be different from each of these t previous boundary solutions.) This condition is met by designating x_j to be tabu, to forbid changing it from 0 to 1, if $\text{TABU}(j) > 0$, where TABU(j) denotes the jth component of TABU.

More generally, we undertake to assure as nearly as possible that the first r variables x_j set equal to 1 after reaching the lower turn-around point will not be among the tabu variables, for which $\text{TABU}(j) > 0$. This is accomplished by attaching a large penalty weight w to TABU(j) —i.e. creating the product $w \cdot \text{TABU}(j)$, which equals 0 if TABU(j) is 0 and equals a large value if TABU(j) is positive. The penalty value (product) is then subtracted from the choice rule evaluation customarily used, where the maximum evaluation identifies a preferred variable to set equal to 1. Once a boundary solution is reached the choice rule regains its normal form, even if r steps have not yet occurred since the turn-around point.

The targeted number of variables to make tabu by this process may be determined as follows. Begin with $r = 1$. Then, after a chosen number of iterations (e.g., after k^* has increased by the amount 2t), set $r = r + 1$. Continue to increment r after each such round of iterations until reaching a desired limit. Then set $r = 1$ again and repeat. (In general, the maximum value of r need not be large, and the changes in r can follow a graduated oscillation schedule for both increases and decreases.)

By contrast, the rule upon reaching the upper turn-around point operates to restrict the choice of variables to set equal to 0 so that the first r of these x_j will be variables with the largest values of TABU(j), while the remaining variables are implicitly tabu during these r steps. (The value of r for this upper turn-around point can differ from that for the lower turn-around point, independent of whether the turn-around points are chosen at different distances from the boundary.) The rule is implemented by creating an inducement to encourage setting a variable to 0, rather than a penalty to discourage setting a variable to 1. In particular, a large positive weight w again is used to create the product $w \cdot \text{TABU}(j)$, and this product is then added to the customary evaluation, whose maximum value in this case identifies an x_j that is preferable to set to 0.

In the case where the oscillation strategy lingers at the boundary by incorporating exchange moves (changing the values of two variables simultaneously in opposite directions), the process can be further controlled by customary forms of short term tabu search memory, as by employing an array tabu_time(j) that records the most recent step, or "time," the variable x_j changed its value. Such memory can be used not only to prevent the exchanges from cycling, but also to avoid reversing more than a desired number of moves, from among those executed during the first r steps after the last turn-around point.

Finally, intermediate and long term strategies can take advantage of these same structures, using two or more TABU arrays based on different choices of t. (A long term TABU array can be maintained as the sum of all boundary solutions encountered, or as an exponentially smoothed sum, without having to keep a record of past boundary solutions to update it.) Then a natural diversification strategy consists of periodically shifting away from a "smaller t" TABU array to a "larger t" array. Alternatively, the arrays can be used in concert, giving each its own weight to create penalty and inducement functions that combine the influence of memories spanning different time intervals. In such a combined approach, weights should be normalized by dividing each by its associated t value. More precisely, if TABU1 and TABU2 are two arrays based on $t = t_1$ and $t = t_2 > t_1$, and if w_1 and w_2 are weights to differentiate the influence of the t_1 most recent boundary solutions from the influence of the $t_2 - t_1$ solutions preceding, then TABU2 may appropriately receive a weight of $z_2 = w_2/(t_2 - t_1)$ while TABU1 correspondingly receives a weight of $z_1 = (w_1/t_1) - z_2$.

The disposition of human problem solvers to combine the influence of memories over varying horizons motivates a closer look at considerations of the form outlined. If the human disposition is worth emulating, we envision the merit of exploring conditional and adaptive ways of integrating time-differentiated memories within a search framework.

Acknowledgment

(Note: This paper is an extended version of the paper "Simple Tabu Thresholding," which appeared in Actes du LIPN: Optimization Combinatoire, CNRS et Université Paris-Nord October, 1990.) This research is supported in part by the Joint Air Force Office of Scientific Research and Office of Naval Research Contract No. F49620-90-C-0033, at the University of Colorado.

References

1. J.W. BARNES and M. LAGUNA, 1991. *A Review and Synthesis of Applications of Tabu Search to Production Scheduling Problems,* Technical Report Series, ORP91-01 Graduate Program in Operations Research, The University of Texas at Austin.
2. R. BATTITI and G. TECCHIOLLI, 1992. The Reactive Tabu Search, IRST Technical Report 9303-13, to appear in *ORSA Journal on Computing.*
3. B. CAO and G. UEBE, 1991. *Solving Transportation Problem with Nonlinear Side Constraints with Tabu Search,* Universität der Bundeswehr, Hamburg.
4. D. COSTA, 1990. Tabu Search Algorithm for Computing an Operational Timetable, ORWP90/9, to appear in *European Journal of Operations Research.*
5. U. DORNDORF and E. PESCEH, 1992. *Fast Clustering Algorithms,* INFORM and University of Limberg.
6. G. DUECK and T. SCHEURER, 1990. Threshold Accepting: A General Purpose Optimization Algorithm, *Journal of Computational Physics 90,* 161–175.
7. U. FAIGLE and W. KERN, 1992. Some Convergence Results for Probabilistic Tabu Search, *ORSA Journal on Computing 4:1,* 32–37.
8. J.M. FARVOLDEN, T.G. CRAINIC, and M. GENDREAU, 1993. Comparison of Tabu Search Approaches for the Fixed Charge Network Design Problem, paper presented at the Joint National ORSA/TIMS Meeting, Phoenix, AZ.
9. T.A. FEO, K. VENKATRAMAN and J. BARD, 1991. A GRASP for a Difficult Single Machine Scheduling Problem, *Computers and Operations Research 18:8.*
10. J. FRENDEWEY and F. GLOVER, 1983. Candidate List Strategies for GN and Simplex SON Methods, Management Science Report Series, MSRS 83-9, University of Colorado, Boulder.
11. A. FREVILLE and G. PLATEAU, 1986. Heuristics and Reduction Methods for Multiple Constraint 0-1 Linear Programming Problems, *European Journal of Operational Research 24,* 206–215.
12. M. GENDREAU, A. HERTZ and G. LAPORTE, 1991. A Tabu Search Heuristic for Vehicle Routing, CRT-777, Centre de Researche sur les transports, Universite de Montreal, to appear in *Management Science.*
13. F. GLOVER, 1967. A Pseudo Primal-Dual Integer Programming Algorithm, *Journal of Research of the National Bureau of Standards 71B:4,* 187–195.
14. F. GLOVER, 1972. Cut Search Methods in Integer Programming, *Mathematical Programming 3:1,* 86–100.
15. F. GLOVER, 1977. Heuristics for Integer Programming Using Surrogate Constraints, *Decision Sciences 8:1,* 156–166.
16. F. GLOVER, 1986. Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers and Operations Research 13,* 533–549.
17. F. GLOVER, 1989. "Tabu Search—Part I," *ORSA Journal on Computing 13,* 190–206.
18. F. GLOVER, 1991. Tabu Search for Nonlinear and Parametric Optimization with Links to Genetic Algorithms, Graduate School of Business, University of Colorado, Boulder, to appear in *Discrete Applied Mathematics.*
19. F. GLOVER, D. KARNEY, D. KLINGMAN and A. NAPIER, 1974. A Computational Study on Start Procedures, Basis Change Criteria and Solution Algorithms for Transportation Problems, *Management Science 20:5,* 793–813.
20. F. GLOVER, M. LEE and J. RYAN, 1992. Least-Cost Network Design for a New Service, *Annals of Operations Research 33,* 351–362.
21. F. GLOVER and C. MCMILLAN, 1986. The General Employee Scheduling Problem: An Integration of Management Science and Artificial Intelligence, *Computers and Operations Research 15:5,* 563–593.
22. F. GLOVER and F. TSENG, 1993. *Thresholding Methods for Finding Subsets of Best and Near Best Elements,* Graduate School of Business, University of Colorado, Boulder.
23. P. HANSEN, 1986. The Steepest Ascent, Mildest Descent Heuristic for Combinatorial Programming, presented at the Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy.
24. P. HANSEN and B. JAUMARD, 1990. Algorithms for the Maximum Satisfiability Problem, *Computing 44,* 279–303.
25. M. HASAN and I.H. OSMAN, 1993. Local Search Strategies for the Maximal Plana Layout Problem Graph, Report no. UKC/IMS/OR93/3, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK. Forthcoming in *International Transactions in Operational Research,* 1994.
26. A. HERTZ and D. DE WERRA, 1987. Using Tabu Search Techniques for Graph Coloring, *Computing 29,* 345–351.
27. A. HERTZ and D. DE WERRA, 1991. The tabu search metaheuristic: how we used it, *Annals of Mathematics and Artificial Intelligence 1,* 111–121.
28. T.C. HU, A.B. KAHNG and C.A. TSAO, 1995. *Old Bachelor Acceptance: A New Class of Non-Monotone Threshold Accepting Methods,* in this issue.
29. R. HUBSCHER and F. GLOVER, 1992. Applying Tabu Search with Influential Diversification to Multiprocessor Scheduling, Graduate School of Business, University of Colorado to appear in *Computers and Operations Research.*
30. B. JAUMARD, P. HANSEN and M. POGGI DE ARAGAO, 1989. Column Generation Methods for Probabilistic Logic, *ORSA Journal on Computing 3,* 135–148.
31. D. JOHNSON, 1990. Local Optimization and the Traveling Salesman Problem, *Proceedings of the 17th Annual Colloquium on Automata, Languages and Programming,* Springer-Verlag, Berlin, 446–461.
32. J.P. KELLY, B.L. GOLDEN and A.A. ASSAD, 1993. Large-Scale Controlled Rounding Using Tabu Search with Strategic Oscillation, *Annals of Operations Research 41,* 69–84.
33. J.P. KELLY, M. LAGUNA and F. GLOVER, 1992. A Study of Diversification Strategies for the Quadratic Assignment Problem, to appear in *Computers and Operations Research.*
34. M. LAGUNA, J.W. BARNES and F. GLOVER, 1991. Tabu Search Methods for a Single Machine Scheduling Problem, *Journal of Intelligent Manufacturing 2,* 63–74.
35. M. LAGUNA and F. GLOVER, 1993. Integrating Target Analysis and Tabu Search for Improved Scheduling Systems, *Expert Systems with Applications 6,* 287–297.
36. M. LAGUNA, J.P. KELLY, J.L. GONZALEZ-VELARDE and F. GLOVER, 1992. Tabu Search for the Multilevel Generalized Assignment Problem, Graduate School of Business and Administration,

University of Colorado at Boulder, to appear in *European Journal of Operations Research*.
37. O. MARTIN, S.W. OTTO and E.W. FELTON, 1992. Large-Step Markov Chains for TSP Incorporating Local Search Heuristics, *Operations Research Letters 11*, 219–224.
38. E. MOONEY and R. RARDIN, 1993. Tabu Search for a Class of Scheduling Problems, *Annals of Operations Research 41:1–4*, 253–278.
39. P. MOSCATO, 1993. An Introduction to Population Approaches for Optimization and Hierarchical Objective Functions: A Discussion on the Role of Tabu Search, *Annals of Operations Research 41*, 85–122.
40. H. MUHLENBEIN, 1991. Parallel Genetic Algorithms and Combinatorial Optimization, to appear in *SIAM Journal on Optimization*.
41. J. MULVEY, 1978. Pivot Strategies for Primal Simplex Network Codes, *Journal of the Association for Computing Machinery 25:2*, 266–270.
42. E. NOWICKI and C. SMUTNICKI, 1993. A Fast Taboo Search Algorithm for the Job Shop Problem, Report 8/93, Institute of Engineering Cybernetics, Technical University of Wroclaw.
43. I.H. OSMAN, 1993. Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem, *Annals of Operations Research 41*, 421–451.
44. I.H. OSMAN and N. CHRISTOFIDES, 1993. Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search, Report No. UKC/IMS/OR93/5, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK. Forthcoming in *International Transactions in Operational Research*, 1994.
45. I.H. OSMAN and S. SALHI, 1993. Tabu Search for the Mix Fleet Vehicle Routing Problem, Report No. UKC/IMS/OR93/8, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK. Presented at ORS-35 Annual Conference in York, July 1993.
46. S.C. PORTO and C. RIBEIRO, 1993. A Tabu Search Approach to Task Scheduling on Heterogeneous Processors Under Precedence Constraints, Monographia em Ciêcia da Computacão, No. 03/93, Pontificia Universidade Católica do Rio de Janeiro.
47. V. ROCHAT and A. SEMET, 1992. Tabu Search Approach for Delivering Pet Food and Flour in Switzerland, ORWP92/9.
48. E. ROLLAND, H. PIRKUL and F. GLOVER, 1993. *Tabu Search for Graph Partitioning*, Graduate School of Management, University of California at Riverside, Riverside, CA.
49. J. RYAN, 1992a. *Depth and Width of Local Optima*, Department of Mathematics, University of Colorado at Denver.
50. J. RYAN, 1992b. *The Depth and Width of Local Minima for Graph Colouring Problems*, Department of Mathematics, University of Colorado at Denver.
51. J. SKORIN-KAPOV and A. VAKHARIA, 1993. Scheduling a Flow-Line Manufacturing Cell: A Tabu Search Approach, *International Journal of Production Research 31:7*, 1721–1734.
52. P. SORIANO and M. GENDREAU, 1993. Diversification Strategies in Tabu Search Algorithms for the Maximum Clique Problem, Publication #940, Centre de Recherche sur les Transports, Universite de Montreal.
53. E. TAILLARD, 1991. Parallel Tabu Search Technique for the Job Shop Scheduling Problem, Research Report ORWP 91/10, Departement de Mathematiques, Ecole Polytechnique Federale de Lausanne.
54. S.R. THANGIAH, I.H. OSMAN, R. VINAYAGAMOORTHY and T. SUN, 1993. Genetic Algorithms for Vehicle Routing With Time Deadlines, Report no. UKC/IMS/OR93/10a, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK. Submitted to *American Journal of Mathematical and Management Sciences*.
55. V.V. VERDEJO, R.M. CUNQUERO and P. SARLI, 1993. *An Application of the Tabu Thresholding Technique: Minimization of the Number of Arc Crossings in an Acyclic Digraph*, Departamento de Estadistica e Investigacion Operative, Univerdsidad de Valencia, Spain.
56. S. VOSS, 1993a. *The P-Median Problem*, Technische Hochschule Dormstadt, Germany.
57. S. VOSS, 1993b. Dynamic Tabu Search Strategies for the Traveling Purchaser Problem, Technische Hochschule Darmstadt, Germany.
58. D. WHITLEY, T. STARKWEATHER and D. FUQUAY, 1989. Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator, *Proceedings of The Third International Conference of Genetic Algorithms*, Fairfax, VA, J.D. Schaffer (ed.) Morgan Kaufman, San Mateo, CA, p. 133.
59. D.L. WOODRUFF, 1993. Tabu Search and Chunking, working paper, University of California, Davis.
60. D.L. WOODRUFF and D.M. ROCKE, 1992. Computable Robust Estimation of Multivariate Location and Shape in High Dimension, to appear in the *Journal of the American Statistics Association*.
61. D.L. WOODRUFF and M.L. SPEARMAN, 1992. Sequencing and Batching for Two Classes of Jobs with Deadlines and Setup Times, *Production and Operations Management 1:1*, 87–102.