Chapter 1

Multi-Start and Strategic Oscillation Methods – Principles to Exploit Adaptive Memory

A Tutorial on Unexplored Opportunities[♦]

FRED GLOVER
University of Colorado

Abstract: We propose approaches for creating improved forms of constructive multi-start and strategic oscillation methods, based on the search principles of *persistent attractiveness* and *marginal conditional validity*. These approaches embody adaptive memory processes by drawing on combinations of recency and frequency information, which can be monitored to encompass varying ranges of the search history. In addition, we propose designs for investigating these approaches empirically, and indicate how a neglected but important kind of memory called *conditional exclusion memory* can be implemented within the context of these methods.

[♦] Published as Chapter 1 of: *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, M. Laguna and J.L. Gonzales-Valarde, eds., Kluwer Academic Publishers, 2000, pp 1-24.

1. BACKGROUND.

1.1 Motivating Concerns and Multi-Start/Strategic Oscillation Links.

Recent studies have confirmed that intelligent uses of adaptive memory can be valuable for creating improved forms of multi-start methods (Fleurent and Glover, 1998; Laguna and Marti, 1998; Rolland, Patterson and Pirkul, 1998, Campos *et al*., 1999).

From a perspective sometimes noted in the tabu search (TS) literature, a multi-start approach can be viewed as an extreme version of a one-sided strategic oscillation approach. Applied to constructive neighborhoods, one-sided strategic oscillation operates by alternating constructive and destructive phases, where each solution generated by a constructive phase is dismantled (to a variable degree) by the destructive phase, after which a new constructive phase builds the solution anew. TS memory-based strategies can be applied in such settings to focus on deep oscillation patterns that destroy large parts of solutions during destructive phases (including the case where “large” = “all”).

The observations of this paper are accordingly offered as a basis for strategies that can be used both in multi-start methods and in strategic oscillation approaches. The basic strategies can also be joined with target analysis (Glover and Laguna, 1997) to identify subsets of variables in 0-1 problems that are sufficient to generate optimal solutions (or more precisely, sufficient to include all variables that receive values of 1). Consequently, these ideas are relevant to strategies for solving large 0-1 problems by reducing them to smaller and more tractable problems. Our following development, which is tutorial in nature, demonstrates how well-established principles can be advantageously put to new uses.

1.2 Classes of Problems.

It is useful to distinguish between problems where large numbers of decisions are sequence independent and those where most (or all) of the decisions must be made in a prescribed order (and where the options available for these decisions are highly restricted by this order). Problems of the first type, in which decisions can be made with little or no concern for sequential restraints, are illustrated by covering, multidimensional knapsacks, partitioning, independent set problems, p-median problems and telecommunication tree-star problems. Decisions can be made sequentially for such problems (which is the essence of a constructive process), and earlier decisions can profoundly affect the legitimacy and conditional quality of later decisions, but large numbers of the decisions could equally well be made in many different orders. Problems of the second type, which are governed by prior precedence restrictions, are illustrated by certain sequencing and scheduling problems where there is no way to evaluate some decisions until an appropriate prior set of decisions has been made.

The ideas at the focus of this paper are developed within the context of the first type of problems, where decisions are largely free of sequencing constraints. Nevertheless, with some added provisions they can be applied to the second type of problems also. For convenience, they will be discussed in relation to applications such as multi-dimensional knapsack and covering problems, where successive steps of a construction can be viewed as progressively making assignments of the form $x_j = 1$ for selected 0-1 variables x_j, understanding all other variables to receive values of 0.

2. NOTATION AND BASIC ASSUMPTIONS

Consider the problem

$$Maximize\ (or\ Minimize)\ f(x) :\ x \in X$$

and let N = {1,...,n} be the index set for *X*. We assume the condition $x \in X$ includes the requirement $x_j = 0$ or 1 for all $j \in N$. In addition, we define N(v) = $\{j \in N : x_j = v\}$, and call N(1) the *In-Set* and N(0) the *Out-Set*. Our terminology is motivated by the fact that many constructive processes can be visualized as successively adding elements (such as nodes or edges of a graph) to an In-Set to create a desired structure, corresponding to the mathematical representation of progressively setting $x_j = 1$ for selected variables.

The methods we examine can be interpreted as beginning with all elements of *N* being placed in a set *N*(#) of "undecided" elements (i.e., the assignment $x_j = \#$ does not commit x_j to be either 0 or 1). Then, constructively, we select various elements of *N*(#) to be placed in the In-Set, *N*(1). The consequence of selecting an element j of *N*(#) to add to *N*(1) may compel other elements of *N*(#) to be placed in *N*(0) or *N*(1) in order to take advantage of dominance considerations or to assure feasibility (including "objective function feasibility", which requires that a solution must be better than the best solution previously found).

2.1 The Assumption of Simple Dependency

The particular form of a constructive method we consider is based on assuming that feasibility, defined by $x \in X$, can be easily established by initially choosing any element $j \in N$ to add to $N(1)$. Following this, we assume that rules are known which make it possible to determine whether certain elements j are then compelled to go into $N(0)$ or $N(1)$. For example, such rules may arise from a requirement that all members of the In-Set must be elements of a tree, or members of a clique, etc. Moreover, at each step following the first, once compulsory assignments are made, we assume all elements remaining in $N(\#)$ are candidates to be chosen as the next j to add to $N(1)$. (A number of methods of this type can also be "reversed" to choose a j to go into $N(0)$ at each step, by flipping the variables so that $N(0)$ and $N(1)$ reverse roles.)

This "simple dependency" assumption, where knowledge of $N(1)$ immediately allows appropriate compulsory assignments to be identified so that all elements remaining in $N(\#)$ are legitimate candidates to be added to $N(1)$ (until a complete solution is obtained, allocating all remaining elements of $N(\#)$ to $N(0)$), leads to a *Simple Difference Rule*.

Briefly stated, the rule says that to make x'' different from x', we must simply choose at least one j from the Out-Set for x' to go into the In-Set for x'', and such a choice exists and is valid at the first step of constructing x'', and at each consecutive step until the moment when no elements of the Out-Set for x' remain in $N''(\#)$ or a complete solution is obtained. More formally, the rule can be expressed as follows.

Simple Difference Rule: Let $N'(v)$ and $N''(v)$ represent $N(v)$ defined relative to x' and x'', respectively, where x' is generated on a given constructive pass, and a new constructive pass is then initiated to generate the solution x''. Then, to assure x'' differs from x':

(1) It is necessary and sufficient to choose at least one element $j \in N''(\#) \cap N'(0)$ to be added to $N''(1)$.

(2) A choice of the form of (1) is always available and legitimate on the first step of constructing x'', and continues to be available and legitimate at each step until the set $N''(\#)$ no longer contains an element of $N'(0)$ (or until a complete solution is obtained).

This rule can be compounded to give conditions for x'' to differ from all previously generated solutions that lie in a specified set X', although memory structures are needed to determine differences in this case. The kinds of recency and frequency memory structures used in tabu search allow the general form of the Simple Difference Rule to be exploited in a convenient manner, while taking advantage of certain principles that the more rigid memory structures of branch and bound methods are unable to exploit. We identify a particularly useful type of such memory, called *conditional exclusion memory*, in section 6.

3. EVALUATIONS BASED ON PERSISTENT ATTRACTIVENESS.

The Principle of Persistent Attractiveness says that good choices derive from making decisions that have often appeared attractive, but that have not previously been made within a particular region or phase of search. That is, persistent attractiveness also carries with it the connotation of being "persistently unselected" within a specified domain or interval. (Illustrations of this principle are given in Chapter 5 of Glover and Laguna, 1997.) We take advantage of this principle in the present setting by creating measures of attractiveness for the purpose of modifying customary evaluations of constructive moves, i.e., evaluations used to select elements to add to the In-Set during a constructive solution pass. We first develop the basic ideas in a context that only uses memory in a rudimentary way. Later we introduce extended uses of memory to create a more advanced approach.

The attractiveness measures derive from a preliminary operation of creating a *component evaluator* E(s,r), where s is the current step of construction and r ranges over the ranks of the elements j chosen to be added to the In-Set on step s. The rank r of each such element is obtained by using a customary evaluation procedure. For example, if the evaluation procedure uses a "bang-for-buck" ratio as in a heuristic surrogate constraint strategy (Glover, 1965, 1977), then $r = 1$ corresponds to the best bang-for-buck ratio, $r = 2$ corresponds to the second best, and so forth. (In this setting, "bang" = the objective function coefficient taken as a numerator, and "buck" = the surrogate constraint coefficient taken as a denominator.) More simply, the ranks may be determined by ordering objective function changes produced by choosing the elements. We examine ranks r only up to a limit r^* which is somewhat less than n.

Likewise, we number the steps $s = 1, ..., s^*$, where s^* is the final step of adding an element j from $N(\#)$ to the In-Set, to result in creating a complete solution.

3.1 Illustration for Creating a Measure of Persistent Attractiveness.

We show how to use $E(s,r)$ to create a *Persistent Attractiveness Measure* PAM_j for each element j by means of a concrete example. We also demonstrate how the resulting measure PAM_j can be used to modify an ordinary evaluator, to allow a constructive pass that makes choices that would not be proposed by the evaluator under normal circumstances.

Assume for convenience that N is reindexed immediately after the first (or an arbitrary) constructive pass so that the sequence in which elements have been added to the In-Set on this pass is given by 1, 2, 3, ..., s^*. In other words, on each step $s = 1$ to s^*, the element j added to the In-Set is indexed so that $j = s$.

For our illustration, the evaluator $E(s,r)$ will take the simple form

$$E(s,r) = E'(s) + E''(r).$$

It is reasonable to suppose that early steps of the construction should influence the value of a persistent attractiveness measure more heavily than later steps (a supposition that will be clarified in subsequent observations). We embody such a "declining influence" effect in our example by stipulating that $E'(1)$ starts at 26, and then for each successive step s after the first, the value $E'(s)$ drops by 2 until reaching $E'(13) = 2$. In other words, we identify $s^* = 13$ as the number of steps required to obtain a complete solution (on the pass considered) and stipulate that $E'(s) = 28 - 2s$ for each step s from 1 to 13.

For simplicity in our example, we will limit r^* to 4, so that evaluations are created only for the 4 highest ranked choices on each step. Then we stipulate that the rank 1 choice is worth 4 points, the rank 2 choice is worth 3 points, etc., which yields $E''(r) = 5 - r$.

Thus the combined evaluator function $E(s,r) = E'(s) + E''(r)$ is given by

$$E(s,r) = 33 - 2s - r$$

Alternative forms for $E(s,r)$ will be considered later.

3.2 Creating a Persistent Attractiveness Measure PAM_j from $E(s,r)$.

To show how $E(s,r)$ can be used to create a Persistent Attractiveness Measure PAM_j , we depict the steps of a constructive pass in the Initial Table that appears below. We imagine $n = 20$, i.e., there are 20 0-1 variables whose best values we seek to approximate by a constructive solution process. Each row of the table corresponds to a step s of the constructive pass, and identifies the top ranked elements j that are candidates to be added to the In-Set at this step. In parenthesis beside each element j appears the $E(s,r)$ value that corresponds to this element on the current step. Because of the assumed indexing of the variables, the index $j = s$ is the best ranked choice (with rank = 1) at step s.

For example, the entries in row 1 of the following table, corresponding to the step $s = 1$, indicate that the elements j with the best ranks, in order from $r = 1$ to 4, are given by $j = 1$, 5, 8 and 2. Applying the formula $E(s,r) = 33 - 2s - r$, the $E(s,r)$ values shown in parentheses in this row start with E(1,1) = 30 for the top ranked element ($j = 1$), followed by $E(1,2) = 29$ for the 2nd ranked element ($j = 5$), then by $E(1,3) = 28$ for the 3rd ranked element ($j = 8$), and so forth.

In this illustration, proceeding from top to bottom in each column of the table, the successive values of $E(s,r)$ decrease by 2 units, while proceeding from left to right in each row of the table, the successive values of E(s,r) decrease by 1 unit. (The reason for this behavior is evident from the formula for $E(s,r)$, which gives s a multiple of −2 and r a multiple of −1.)

Initial Table

s(= j)	Top r* = 4 ranked indexes and $E(s,r)$ values			
1	1(30)	5(29)	8(28)	2(27)
2	2(28)	5(27)	3(26)	7(25)
3	3(26)	6(25)	9(24)	5(23)
4	4(24)	9(23)	14(22)	5(21)
5	5(22)	14(21)	6(20)	8(19)
6	6(20)	7(19)	8(18)	9(17)
7	7(18)	9(17)	14(16)	8(15)
8	8(16)	14(15)	16(14)	13(13)
9	9(14)	10(13)	15(12)	12(11)
10	10(12)	13(11)	15(10)	12(9)
11	11(10)	13(9)	15(8)	16(7)
12	12(8)	16(7)	13(6)	17(5)
13	13(6)	16(5)	17(4)	20(2)

From the values in this Initial Table, we compute the Persitent Attractiveness Measure PAM_j for each element *j* by simply summing the *E*(*s*,*r*) values attached to *j*. Thus, for example, since the index *j* = 1 has only the single *E*(*s*,*r*) value of 30 attached to it in the table, PAM_1 = 30. Similarly, since *j* = 2 has the two *E*(*s*,*r*) values 27 and 28 attached to it, PAM_2 = 55. Likewise, PAM_3 = 26 + 26 = 52, PAM_4 = 24, PAM_5 = 29 + 27 + 23 + 21 + 22 = 122, and so forth.

The illustrative formula we have chosen for *E*(*s*,*r*) tends to produce higher PAM_j values for elements j that are among the top ranked choices in earlier steps of the construction. This results from the -2 multiple for *s* in the formula for *E*(*s*,*r*), which causes *E*(*s*,*r*) to drop as *s* increases. A positive multiple for *s* would have the opposite effect, producing higher PAM_j values for elements *j* that are among the top ranked choices in later steps. In all cases, the more often an element *j* appears with a top rank, the greater its persistent attractiveness. Thus the PAM_j values constitute a combined form of recency and frequency information, weighted by attractiveness.

3.3 Using Persistent Attractiveness to Assess the Value of Choices Not Made.

For the purpose of analyzing the choices — to see the attractiveness of choices *not* made at various steps — we create a Persistent Attractiveness Table by duplicating the Initial Table, except that we replace the *E*(*s*,*r*) values by the PAM_j values. (Thus, each time an element *j* appears, the same PAM_j value also appears.) A large PAM_j value, as noted, generally indicates that a variable was often one of the top choices. The elements *j* from 14 to 20, which were never selected, also have associated measures of persistent attractiveness, since each was among the top four choices in at least one of the steps of construction.

The resulting Persistent Attractiveness Table appears below. In addition the table identifies two difference values D1 and D2 for each step s, where D1 = $Max(PAM_j) - First(PAM_j)$ and D2 = $2ndMax(PAM_j) - First(PAM_j)$, defining $Max(PAM_j)$ and $2ndMax(PAM_j)$ to be the two largest PAM_j values in the row, and $First(PAM_j)$ to be the first PAM_j value in the row (i.e., the PAM_j value for the element *j* with the top rank).

Thus, these difference values indicate the degree by which the persistent attractiveness of these Max and 2ndMax values exceeded the persistent attractiveness of the element actually chosen. (When the element chosen also has the highest persistent attractiveness value PAM_j , then $D1 = 0$ and $D2$ is negative.) Such information can be used in choice rules based on "marginal value" determinations.

Persistent Attractiveness Table

s (= j)	Top r* = ranked indexes and PAM_j values				$D1$	$D2$
1	1(30)	5(122)	8(96)	2(55)	92	66
2	2(55)	5(122)	3(52)	7(62)	67	7
3	3(52)	6(65)	9(95)	5(122)	70	43
4	4(24)	9(95)	14(74)	5(122)	98	71
5	5(122)	14(74)	6(65)	8(96)	0	-26
6	6(65)	7(62)	8(96)	9(95)	31	30
7	7(62)	9(25)	14(74)	8(96)	34	33
8	8(96)	14(74)	16(33)	13(45)	0	-20
9	9(95)	10(25)	15(30)	12(28)	0	-65
10	10(25)	13(45)	15(30)	12(28)	20	5
11	11(10)	13(45)	15(30)	16(33)	35	27
12	12(28)	16(33)	13(45)	17(9)	17	5
13	13(45)	16(33)	17(9)	20(2)	0	-12

3.4 Analysis of the Table.

In this section we raise a number of key issues about how to take advantage of the Persistent Attractiveness Table. The sections that follow then provide a deeper look at considerations relevant to handling these issues.

3.4.1 Creating New Passes Guided by PAM_j values.

Information from the Persistent Attractiveness Table suggests a number of changes that might be made in the choices of the first pass, and thus suggests the merit of initiating a new constructive pass that implements such changes. The resulting altered sequence in which the elements j are added to the In-Set will also alter the evaluations and rankings of elements. These new rankings provide information in addition to the information provided by the Persistent Attractiveness Table which can be taken into account to give an enlarged basis for selecting the elements to be added to the In-Set at each step.

Moreover, as a new sequence of choices is produced, new $E(s,r)$ values are also produced. This affords an opportunity to combine information from these values with the previous $E(s,r)$ and PAM_j values to produce a modified choice rule. We indicate two options for doing this (referring to the pass that generates the new $E(s,r)$ values as the "current" pass):

Option 1. Wait until the end of the current pass, so that all of the new $E(s,r)$ and new PAM_j values are known, before undertaking to make use of these values (to influence the next pass that follows the current pass);

Option 2. Keep partial (incomplete) new PAM_j values, which are updated at each step of the current pass, and use them to influence the choices made during the current pass.

By a partial PAM_j value we mean one that sums the $E(s,r)$ values attached to j at each step up to the current step s, without waiting until the final step $s = s^*$ to compute such values. In fact, Option 2 is also available on the first pass (i.e., the one that precedes the current pass). Accordingly, the partial PAM_j values, which are computed as the pass progresses, can be taken into account (in addition to rankings) for making choices at each step.

3.4.2 Selecting Steps Where Changes Are to Be Made.

There is also an issue of whether it can be useful to retain some portion of the previous constructive pass, and to make changes only after some "critical point" of the construction. From the standpoint of computation, a strategy that changes choices only at later steps (i.e., for larger values of s) has the advantage of reducing overall effort, since information from earlier steps remains unchanged, and fewer new steps need to be evaluated. In addition, evaluations made at later steps to rank the elements are likely to be more accurate (given the choices already made) because they are derived from a smaller residual problem, where fewer decisions remain to be executed.

On the other hand, changing choices at earlier steps allows the consequences of these changes to be considered throughout a larger range of decisions. That is, the influence of these changed decisions, which alter the rankings of various elements, operates for a larger number of steps and applies to a larger range of elements (since more elements remain to be selected at earlier steps). In considering the relevance of such tradeoffs, it is possible to take advantage of both early and late changes, designing a procedure that first changes only later choices (to gain the advantage of less effort), and then changes earlier choices (to gain the advantage of producing consequences that can be evaluated over a larger horizon).

3.4.3 Making Changes Independent of Previous Ordering

Instead of undertaking to distinguish between choices made early and late, the issue can be shifted to consider which decisions to retain and which to replace.

Specifically, if a given set of decisions is identified as "potentially replaceable" (hence, at least temporarily to be removed), while remaining decisions are to be retained on the next pass, then effort can be saved by treating the set of retained decisions as if they were made in a block before all other decisions. The determination of new evaluations and rankings can thus be restricted to elements not among those retained. In addition, by conceiving the retained decisions to precede all others, the consequences of such decisions are more fully represented than by an approach that amounts to "inserting" new decisions within some partially retained sequence.

This approach of selectively retaining some block of decisions that are treated as if prior to all others in fact constitutes an instance of strategic oscillation, as will be elaborated more fully in the next section.

3.4.4 Compensating for Incomplete Information.

Evidently the information of the Persistent Attractiveness Table is incomplete, as a result of depending on the sequence in which elements are selected on a given pass. More precisely, elements selected to be added to the In-Set during early steps are not present to be ranked over very many steps. Consequently, these elements do not have an opportunity to have many $E(s,r)$ values attached to them, and hence these elements may have smaller PAM_j values than elements added at later steps. The effect of this is especially pronounced for the element chosen on the first step, since only the single $E(s,r)$ value attached to it on this step becomes incorporated into its PAM_j value. This fact provides a motivation for giving $E(s,r)$ a form that assigns larger weight to choices at earlier steps. In addition, it suggests the value of: (a) reordering the earlier decisions, even if they are retained; (b) using the construction approach with a sequential fan candidate list strategy (Chapter 3 of Glover and Laguna, 1997), which can generate $E(s,r)$ values along different construction sequences, and thus provide additional information.

3.4.5 Alternative Parameters for Creating $E(s,r)$ Values.

The parameters used to create $E(s,r)$ values (and hence the PAM_j values) can clearly affect the analysis available from the Persistent Attractiveness Table. To illustrate some simple options, using the formula $E(s,r) = E'(s) + E''(r)$, we identify values for $E'(s)$ and $E''(r)$ as follows:

(i) $E'(s) = C' + M'(c' + s^* - s)$, where $C' \geq 0$, $c' \geq 1$ and the multiple M' can have any nonzero value. In the illustration of section 3.3, $M' = 2$, $C' = 0$ and $c' = 1$. (The value $1 + s^* - s$, for $c' = 1$, reverses the indexing of s, to run from s^* down to 1 as s runs from 1 to s^*. The constant c' can be chosen larger to make the values $c' + s^* - s$ more similar to each other.) Equivalently, this may be written $E'(s) = D' - M's$, which has the form for $E'(s)$ used in the numerical illustration of section 3.3. However, intuition about choosing D' may be improved by considering the representation of $E'(s)$ indicated here.

(ii) $E''(r) = C'' + M''(c'' + r^* - r)$, where $C'' \geq 0$, $c'' \geq 1$, and the multiple M'' is positive. In the illustration of section 3.3, $C'' = 0$ and $M'' = 1$. Comments similar to those of (i) apply to this formula.

To increase the differential effect produced by different values of s and r, the value $c' + s^* - s$ and the value $c'' + r^* - r$ can each be raised to a power greater than 1. Allowing for changes in these parameters can produce the basis for a multi-start approach that is guided by progressively modified PAM_j values. However, these are not the only relevant concerns. The value of r^* also can influence the PAM_j values, and the choice of r^* in the illustration of section 3.3 was somewhat arbitrary. (Note that s^*, in contrast to r^*, is not subject to being selected, and hence does not have to be considered.)

A more intelligent way to determine r^* is to allow it to vary during the constructive pass, so that it can be larger on earlier steps. An example of such an approach is to set $r^* = s^{**} + 1 - s$ on step s, where s^{**} is an advance estimate of s^*. (An ample value for s^{**} can be used, and then values of r greater than $s^* + 1 - s$ can be ignored at the later point when the PAM_j values are computed.) However, because work is involved in identifying the ranking of choices, an upper limit may be placed on r^*, perhaps even as small as 8 or 16. A lower limit can also be placed on r^*, such as $r^* \geq 3$.

4. GENERAL CHARACTER OF THE PERSISTENT ATTRACTIVENESS MEASURE AND ITS ANALYSIS.

We now examine issues raised in the preceding section from a broader perspective. Section 5 then examines underlying principles and more general concerns. (Additional considerations relevant for varying levels of implementation are also described in the Appendix.)

4.1 Notational Conventions and Structures for *E*(*s*,*r*) and PAM_j.

The notation used to define $E(s,r)$ and PAM_j oversimplifies the general situation. For example, as noted in section 3.4.5, r^* can vary depending on the step s. Also, independent of notation, the evaluation function that creates the rankings can change as the construction is applied. A familiar example occurs in the case of a multi-knapsack or covering problem, where on the last step (or last 2 steps, etc.) a bang-for-buck ratio may be amended to account for other factors, such as the greatest profit item that can maintain feasibility or the least cost item that can achieve feasibility.

For reasons suggested in sections 3.4.4 and 3.4.5, it appears relevant to create $E(s,r)$ to vary monotonically as a function of s. That is, for a given r, we may generally stipulate

$$E(1,r) \geq E(2,r) \geq \ldots \geq E(s^*,r).$$

However, we may also consider a reverse type of monotonicity based on s, where the inequalities above go in the reverse direction. Such an ordering may be introduced periodically for diversification purposes. In all cases it is appropriate to make $E(s,r)$ monotonic in r, so that on a given step s,

$$E(s,1) \geq E(s,2) \geq \ldots \geq E(s,r^*).$$

Issues of ranking disclose another limitation of the notation employed. First, since ranks $r = 1$ to r^* are based on an original evaluation, in some cases two consecutive ranks r and $r+1$ may correspond to identical original evaluations, and in this case we may let $E(s,r) = E(s,r+1)$. More generally, however, depending on the nature of the original evaluation, we may allow $E(s,r)$ to take fuller consideration of the relative magnitude of this evaluation for successively ranked choices (which makes $E(s,r)$ dependent upon step s, as is also true in the case of possible tied evaluations). Nevertheless, we have used the current notation because often the ranking of choices is important in determining values to assign to our special evaluation $E(s,r)$.

4.2 Maintaining Updated PAM_j Values.

Section 3.4.1 raises the issue of keeping an updated record of PAM_j values in the process of building the Initial Table, thus allowing these partial values to influence decisions even before completing the construction that provides a full solution. Such an approach can be used immediately on the first pass, or can be delayed until a later pass.

The calculation is simple. PAM_j is initialized to 0 for each j at the beginning of each pass. At a given step s of the current pass, once the top r^* choices are identified, then PAM_j is updated by identifying, for each r, the variable j that yields rank r, and then setting

$$PAM_j := PAM_j + E(s,r).$$

To combine this value with previous PAM_j values, several options are possible. Among the simpler options are to keep a running sum, $SumPAM_j$, which is initialized to 0 only at the start of the first pass (but not at the start of later passes), and then is updated exactly as PAM_j is updated, by identifying the matched j and r and setting

$$SumPAM_j := SumPAM_j + E(s,r).$$

Then $SumPAM_j$ can be changed into a mean value by dividing by the number of passes, or by the number of steps (accumulated from the beginning of the first pass), and so forth. A running value based on exponential smoothing can also be used.

5. CONDITIONAL EFFECTS OF CONSTRUCTIVE METHODS.

5.1 Principles and Inferences.

Since constructive methods make decisions sequentially, and the evaluation of potential decisions depends on those decisions made earlier, the effect of conditionality is one of the primary determinants of the effectiveness of such methods. For this reason it is useful to begin by identifying a principle that applies to constructive search methods in many types of applications.

Principle of Marginal Conditional Validity (MCV Principle). — As more decisions are made in a constructive approach (as by assigning values to an increasing number of variables), the information that allows these decisions to be evaluated becomes increasingly accurate, and hence the decisions become increasingly valid, conditional upon the decisions previously made.

The justification for the MCV principle is simply that as more decisions are made, the consequences of imposing them cause the problem to be more and more restricted (e.g., reduced in dimensionality). Consequently, future decisions face less complexity and less ambiguity about which choices are likely to be preferable.

This principle has long been known to apply to branch and bound methods, where variables are progressively assigned values by branching decisions.[1] In particular, a branch and bound method can be viewed as a repeated constructive heuristic, where the multiple passes of the heuristic are compelled to operate within a tree structure. The imposed tree structure has the advantage that all descendants of a given decision can be assured to inherit the restrictions that apply to their ancestors. On the other hand, this structure has the disadvantage of locking the search into a relatively rigid pattern, preventing flexible choices that might lead to good solutions much more readily.

Two evident outcomes of the MCV Principle are as follows.

Inference 1. Early decisions are more likely to be bad ones.

[1] Implications of the principle for creating dynamic branch and bound strategies are examined, for example, in Glover and Tangedahl, 1976.

Inference 2. Early decisions are likely to look better than they should, once later decisions have been made.

Inference 1 is an immediate consequence of the MCV Principle. Inference 2 results from the fact that later decisions which are chosen for their apparent quality manifest that quality in relation to the structure imposed by earlier decisions. Consequently, they are designed to "fit around" the earlier decisions, and thus are disposed to create a completed solution where the earlier decisions appear in harmony with those made later. (If, given later decisions, an earlier decision looks bad, then almost certainly it is bad. However, since later decisions are chosen to make the best of conditions created by earlier ones, if an earlier decision manages to look good in conjunction with those made subsequently, there is no assurance that it truly is good.)

These observations lead to the following additional inferences about constructive methods.

Inference 3. The outcome of a constructive method can often be improved by examining the resulting complete solution, where all decisions have been made, and seeing whether one of the decisions can now be advantageously replaced with a different one.

This inference is directly reinforced by the MCV Principle, because the outcome of changing a given decision — after a complete solution is obtained — has the benefit of being evaluated in the situation where all other decisions have been made. Therefore, in a conditional sense, the validity of this changed decision is likely to be greater (i.e., its evaluation is likely to be more accurate) than that of the decision it replaces — since the replaced decision was made at a point where only some subset of the full set of decisions had been made. Nevertheless, the scope of Inference 3 is inhibited by Inference 2. That is, the influence of conditional choices will tend to make decisions embodied in the current solution look better than they really are, given the other decisions made that tend to "support" them.

Inference 4. As a basis for doing better than standard types of improvement methods, it is useful to identify clusters of decisions that mutually reinforce each other and to find which of these decisions becomes less attractive when the reinforcement of its partners is removed.

Inference 4 motivates strategies for improvement methods that exploit clustering and conditional analysis (see Chapter 10 of Glover and Laguna, 1997). However, this inference can also be exploited within the context of a constructive method. It is particularly relevant to applying the Principle of Persistent Attractiveness.

5.2 Indicators of Persistent Attractiveness.

We can distinguish between two different indicators of persistent attractiveness in the conditional setting of constructive methods.

Indicator 1. A decision appears attractive for some number of decision steps, but is not made until a step that occurs somewhat after it first appears to be attractive.

Indicator 2. A decision appears attractive for some number of decision steps, but is never made.

In the case of Indicator 1, selecting the specified decision results in making a choice that is ultimately made anyway, but reinforces the focus on this "good choice" so that its implications can be generated, and hence exploited, at an earlier stage. Consequently, this creates an intensification effect relative to such attractive, but repositioned, decisions. By contrast, Indicator 2 clearly provides a foundation for diversification strategies, which drive the solutions to incorporate entirely new elements. However, while Indicator 1 may be useful to consider in strategic designs, the Simple Difference Rule implies Indicator 2 is more crucial to rely on. The following observations elaborate the relevance of these indicators and the manner in which they may be exploited.

5.3 Rationale for Using PAM_j Values to Generate Modified Decisions.

Any decision made at a later stage of a previous pass may change the appearance of attractiveness of other decisions if it is made earlier on the new pass. Given that the decision is part of the solution produced on the previous pass, if it is also part of the solution produced on the new pass (by making the decision earlier), then it also has a chance to influence the choice of other parts of the new solution. This opportunity was denied on the previous pass because the decision did not occur until a later point, and thus the resulting change affords an opportunity for the new solution to be better.

In addition, if the decision has been persistently attractive, there is an increased likelihood that it is a valid decision (i.e., a component of a high quality solution). Therefore, such a decision will offer additional advantages by being placed earlier, so that its consequences for evaluating other decisions can be appropriately taken into account.

From the standpoint of Indicator 2, the earlier that an "unmade decision" appeared attractive, the more likely it is that the decision should be considered attractive in an unconditional sense (since decisions that did not appear attractive until a later step acquired their attractiveness as a result of the decisions that preceded). This fact is relevant to defining the $E(s,r)$ values, or alternately to defining how they should be used to define the PAM_j values (as where the PAM_j values are produced by a rule other than by simple summing).

5.4 Additional Diversification.

Another principle derived from tabu search advocates the merit of making moves that are "influential", i.e., that cause significant changes. The characteristic of being influential is not sufficient in itself to warrant a move, however, because moves that are merely influential have no necessary virtue unless they are also linked in some way to solution quality.[2] This leads to considering the following indicator.

Influence/Quality Indicator. Identify a decision that appears attractive at some point during a given constructive pass especially during earlier steps of the pass. The decision should receive increased emphasis if its implementation would also change the attractiveness of other decisions, according to the degree that it changes such evaluations of attractiveness.

Evidently, a decision that rates highly by reference to such an indicator is one that that can create significant diversification in the solutions produced, by offering a chance to obtain a good solution that has a substantially different composition than the one obtained on the previous pass. Applying this type of indicator, however, may require somewhat more work than applying Indicators 1 and 2. Specifically, to know whether a given decision will change the attractiveness of other decisions requires that the decision tentatively be made, and then making the effort to examine its consequences.

On the other hand, this added effort may be avoided if an indirect strategy is used to indicate whether a decision is likely to change the evaluation of others. Such a strategy is based on the following analysis.

It is often likely that if making Decision *A* causes Decision *B* to become less attractive, then making Decision *B* will also cause Decision *A* to become less attractive. Thus, suppose Decision *A* appears attractive at a particular point, but upon making Decision *B* instead, Decision *A* now becomes significantly less attractive. Then Decision *A* may be considered an influential one (at least relative to Decision *B*). Consequently, an indirect way to identify potentially influential (yet potentially good) decisions is to look for those that were attractive at some (not-very-late) point on the previous pass, but were not selected, and which then later became significantly unattractive on this pass. These decisions have a notably

[2] Chapter 5, sections 5.1.1 and 5.1.2, of Glover and Laguna, 1997, discusses tradeoffs between influence and quality.

different effect than the decisions sought by the Indicators 1 and 2, and can be important for longer term diversification.

6. CONDITIONAL EXCLUSION MEMORY

An useful component of a memory design for search methods, especially for 0-1 problems, is a "conditional exclusion memory", which has generally been overlooked in the literature. Conditional EXclusion (CEX) memory is a combination of frequency memory and recency memory, that allows the effects of frequency to be isolated in a more intelligent way than in more primitive types of memory.[3]

The purpose of CEX memory is to allow a constructive procedure to generate a new solution by choosing $x_j = 1$ at each step (hence adding j to the In-Set) so that the resulting solution will not duplicate any solution previously generated, where for pragmatic and strategic purposes we restrict attention to solutions generated on the *p* most recent passes. (The value *p* is chosen to be a conveniently manageable but nevertheless effective number, based on the problem dimension and experience. E.g., p = 20 to 50 may work in a variety of applications.) Denote these *p* solutions by $x[1]$, ..., $x[p]$, from newest to oldest. The approach for using this memory is as follows.

CEX Method.

1. Keep a frequency record FR_j of the number of times the assignment $x_j = 1$ occurs in the solutions $x[1]$ to $x[p]$. (Hence, the *FR* vector is just the sum of these *p* solutions and can be updated at each step by setting $FR := FR + x[1] - x[p+1]$.)
2. Select a variable x_j to receive an assignment $x_j = 1$ by biasing its evaluation to favor a small frequency value FR_j. For example, if EV_j is a standard evaluation for $x_j = 1$, pick *j* to maximize EV_j subject to $FR_j \le MinFR + \Delta$, where *MinFR* is the minimum of the FR_j values

[3] This type of memory could also appropriately be called "Sequential EXclusion" memory, which provides a more interesting acronym. Observing that exclusion always implies inclusion, which results in an associated "Sequential INclusion" memory, leads inescapably to the conclusion that SEX is impossible without SIN.

and Δ is small. (Or choose j to maximize $EV_j/(1 + FR_j)$, etc.) Then set $x_j = 1$.

3. If $FR_j = 0$, a sufficient set of assignments $x_j = 1$ has been identified to assure that no solution from the collection $x[1]$ to $x[p]$ will be duplicated by the new solution. All remaining assignments can be made by any rule desired.
4. If $FR_j > 0$, redefine FR to be the sum of the solutions $x[h]$ whose jth component $x_j[h] = 1$. (If $FR_j > p/2$, the new sum can be computed more quickly by subtracting from FR the solutions $x[h]$ such that $x_j[h] = 0$.) Then return to step 2.

The CEX approach can be applied as well to special classes of solutions other than the p most recent solutions (such as the p most recent local optima or the p best solutions found within some chosen span of time). The approach clearly generates and exploits more refined information than a procedure designed to set $x_j = 1$ by giving preference to small frequency values FR_j of a single unadjusted FR vector.

The approach can also be generalized in several natural ways. For example, instead of representing the p most recent solutions, some of the vectors $x[h]$ can themselves be frequency vectors created by summing other solutions. Specifically, we may suppose $x[1]$ to $x[p1]$ consists of the $p1$ most recent solutions, but the vectors $x[p1+1]$ to $x[p]$ represent summed solutions, where p is allowed to vary to permit adjustments over time. E.g., $x[p]$ may be the sum of the first k solutions generated, $x[p–1]$ may be the sum of next k solutions generated, etc. Then as more solutions are generated, to keep p from growing too large, $x[p]$ is changed to be the sum of $x[p]$ and $x[p–1]$, and the other $x[h]$ vectors, excluding $x[p–1]$, are reindexed appropriately. Subsequent steps similarly merge other $x[h]$ vectors. (The rule for choosing which pair to merge next provides variation in the approach.) In this manner, the number of vectors recorded and manipulated can remain manageable, with a total effort on the same order as keeping track of p distinct solutions.

This generalized form of the CEX method, applying the same rules (steps 1-4) previously indicated, will still guarantee that no previous solution will be duplicated from the collection embodied in $x[1]$ to $x[p]$ under easily identified conditions. Specifically, the guarantee holds if some subset of $x[1]$ to $x[p1]$ has been removed and $FR^*_j = 0$, where FR^* is the modification of the original FR that results by removing only this subset (hence, not using the update to remove vectors $x[h]$ for $h > p1$). However, it is possible that these conditions will never be met, yet a few steps after removing all of $x[1]$ to $x[p1]$ from *FR* may still result in avoiding duplications. This fact enhances the utility of CEX memory in the general case.

7. CONCLUSION

The principles described in this paper, and the strategies proposed for exploiting them, offer a chance to create forms of multi-start methods that differ significantly from those considered in the past. Features that distinguish such methods from previous multi-start methods include the creation of measures to capture information about recency, frequency and attractiveness, which can be monitored and updated in adaptive memory structures, as used in tabu search. Thus, instead of simply resorting to randomized re-starting processes, in which current decisions derives no benefit from knowledge accumulated during prior search, specific types of information are identified that provide a foundation for systematically exploiting history.

The concept of persistent attractiveness plays a key role in deriving appropriate measures, and acquires particular relevance in consideration of conditional effects. In turn, these effects lead to inferences about the nature effective responses, which become translated into strategies that draw on associated indicators of quality and influence, and that take advantage of conditional exclusion memory.

The potential value of embedding such knowledge in multi-start methods is suggested by recent studies in which adaptive memory strategies have demonstrated the ability to create superior versions of multi-start methods. The observations of this paper are offered as a basis for developing more advanced forms of such memory-based strategies.

REFERENCES.

Amini, M., B. Alidaee and G. Kochenberger (1999). "A Scatter Search Approach to Unconstrained Quadratic Binary Programs," to appear in *New Methods in Optimization*, McGraw Hill.

Campos, V., F. Glover, M. Laguna and R. Marti (1999). "An Experimental Evaluation of a Scatter Search for the Linear Ordering Problem," Universitat de Valencia and University of Colorado.

Fleurent, C. and F. Glover (1998). "Improved Constructive Multistart Strategies for the Quadratic Assignment Problem," Research Report, University of Colorado, to appear in the *INFORMS Journal on Computing*.

Glover, F. (1965). "A Multiphase-dual Algorithm for the Zero-One Integer Programming Problem," *Operations Research*, 13, 879-919.

Glover, F. (1977). "Heuristics for Integer Programming Using Surrogate Constraints," *Decision Sciences*, 8, 156-166.

Glover, F. (1978). "Parametric Branch and Bound," *Omega*, Vol. 6, No. 0, 1-9.

Glover, F., A. Amini, G. Kochenberger, B. Alidaee (1999). "A New Evolutionary Scatter Search Metaheuristic for Unconstrained Quadratic Binary Programming," Research Report, University of Mississippi, Univesity, MS.

Glover, F. and M. Laguna (1997). *Tabu Search*, Kluwer Academic Publishers.

Glover, F. and L. Tangedahl (1976). "Dynamic Strategies for Branch and Bound," *Omega*, Vol. 4, No. 5, 1-6.

Laguna, M. and R. Marti (1998). "Local Search and Path Relinking for the Linear Ordering Problem," Research Report, University of Colorado.

Rolland, E., R. Patterson and H. Pirkul (1999). "Memory Adaptive Reasoning and Greedy Assignment Techniques for the CMST." In *Meta-Heuristics*: *Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I. Osman & C. Roucairol (eds.), Norwell, Massachusets: Kluwer Academic Publishers, pp. 487-498.

APPENDIX — *Considerations Relevant for Implementation*

We examine alternatives for implementing the solution principles discussed in this paper organized around a design of the following type:

(A) Initially, create a method to be as powerful as possible without concern for speed. (Seek to produce a method that obtains the highest quality solutions in the least number of iterations, without concern for how long an iteration takes.)

(B) Once a good approach is identified, determine how its speed may be improved.

(C) As a special exception, if an exceedingly simple variant of an approach emerges that is easy to implement, and if its outcomes offer a chance to gain insights into the design of a more complex approach, then the simple variant may be tested at once.

The type of implementation philosophy embodied in (A) clearly requires common sense in its interpretation. (For example, an iteration can always be defined to embody the execution of a complete solution method, hence reducing the number of iterations to 1!) Nevertheless, as a general principle (when "iteration" is defined in an appropriate way), the preceding design is useful for focusing effort on identifying the considerations that have the greatest impact before worrying about those that are subsidiary. It also has the utility of establishing limits and targets. If the "most powerful" version of an approach does not work well (regardless of allowing its iterations to consume more time than would be tolerated in practice), then there is no sense wasting time trying to develop an efficient version of the approach. (Look for a different approach instead.) On the other hand, if the most powerful version effectively finds its way to good solutions, then it provides a goal to be reached by more efficient versions (and a general foundation for developing such versions).

The exception of (C) is in recognition that the ultimate goal of any design is to gain as much information as possible as soon as possible about the nature of good decisions in order to exploit this information in the subsequent development process. Insights produced by a simple method may yield information that can be used in creating improved evaluations for a more advanced approach (i.e., the simple approach can become a "subroutine" of the advanced approach). However, (C) must be applied with extreme care, because it is always possible to see “easy alternatives”, and when one of these alternatives is implemented there is a great temptation to keep making marginal adjustments (in the hope of creating a version that works a little better). Such a process can become a costly detour.

In the case of constructive solution processes, invoking (A) and (B) as way of organizing an investigation of alternative methods suggests the following design:

(D1) Decisions gauged to be best (to change decisions that have gone before) should be placed early in the construction sequence so that their effects on other decisions can be identified.

(D2) The Simple Difference Rule should be relied on to compel changes that are necessary.

When considering (C), a naive (but interesting) way to apply (D2) immediately surfaces. This is simply to reverse the sequence obtained on a given pass, adopting the perspective that if the latest decisions are going to made anyway, then they may as well be sequenced first to see if the earlier decisions still receive evaluations that warrant including the same elements in the new pass. With high probability, this radically changed sequence will quickly yield very poor evaluations for next elements of the sequence (following the reverse order). When a "relative evaluation threshold" is exceeded — where the next element to be added looks bad enough compared to others available, using the standard evaluation — then the imposed reverse sequence should then be scuttled and the remainder of the sequence constructed using the standard evaluation. Upon obtaining the new sequence, the process can be completed.

Such an exceedingly simple approach, run for a number of passes, can be accompanied by monitoring to identify $E(s,r)$ values as a foundation for creating PAM_j values. But the apparent weakness of the approach is that the elements chosen at the end of a construction sequence are very likely to be "crack fillers" — elements that plug up the last holes in a solution structure left by the decisions preceding. In general, of course, the greater the number of preceding decisions, the greater the chance that the current decisions have little relevance except as a result of these antecedents, and hence a sequence reversal strategy is likely to be ineffective.

On the other hand, the PAM_j values give a way to modify such a strategy to become more effective. For example, if the component $E'(s)$ of $E(s,r)$ is changed so that its values are in ascending order, then the PAM_j values will tend to increase the attractiveness of elements assigned later in the sequence. Placing these later “attractive elements” first in the new sequence tends to follow the philosophy of reversing the sequence, but in a more subtle way.

Still simpler (in the spirit of (C)) is to go through the preceding sequence (just created) in reverse order, and choose an element to be next in the new sequence if its evaluation in this new sequence passes a threshold of desirability. (The threshold may be adaptive, taking into account the goal of not progressing too far before selecting an element to include.) Alternatively, the sequence can be divided into k segments, each containing n/k elements (rounded appropriately). Then, while progressing through these segments in reverse, pick the element from the segment that has the highest current evaluation in the new sequence. The selected element becomes the next element of that sequence (allowing a segment to be skipped if all its elements are sufficiently bad). The method will therefore tend to include k elements from the old sequence, in reverse order, to populate the new sequence.

In all such simplified variants, the process of monitoring $E(s,r)$ values continues, for the purpose of generating more advanced decision alternatives. Thus, these variants can be envisioned as a basis for creating subroutines to become part of a more advanced procedure.